ULB UNIVERSITÉ LIBRE DE BRUXELLES

DI·fusion

Dépôt Institutionnel de l'Université libre de Bruxelles /
Université libre de Bruxelles Institutional Repository
Thèse de doctorat/ PhD Thesis

Citation APA:

Brohée, S. (2008). *Etude bioinformatique du réseau d'interactions entre protéines de transport ches les Fungi* (Unpublished doctoral dissertation).

Université libre de Bruxelles, Faculté des Sciences – Sciences biologiques, Bruxelles.

Disponible à / Available at permalink : https://dipot.ulb.ac.be/dspace/bitstream/2013/210432/15/0e52a8b3-982c-4616-9287-1e1e0b0cb703.txt

(English version below)

Cette thèse de doctorat a été numérisée par l'Université libre de Bruxelles. L'auteur qui s'opposerait à sa mise en ligne dans DI-fusion est invité à prendre contact avec l'Université (di-fusion@ulb.be).

Dans le cas où une version électronique native de la thèse existe, l'Université ne peut garantir que la présente version numérisée soit identique à la version électronique native, ni qu'elle soit la version officielle définitive de la thèse.

DI-fusion, le Dépôt Institutionnel de l'Université libre de Bruxelles, recueille la production scientifique de l'Université, mise à disposition en libre accès autant que possible. Les œuvres accessibles dans DI-fusion sont protégées par la législation belge relative aux droits d'auteur et aux droits voisins. Toute personne peut, sans avoir à demander l'autorisation de l'auteur ou de l'ayant-droit, à des fins d'usage privé ou à des fins d'illustration de l'enseignement ou de recherche scientifique, dans la mesure justifiée par le but non lucratif poursuivi, lire, télécharger ou reproduire sur papier ou sur tout autre support, les articles ou des fragments d'autres œuvres, disponibles dans DI-fusion, pour autant que :

- Le nom des auteurs, le titre et la référence bibliographique complète soient cités;
- L'identifiant unique attribué aux métadonnées dans DI-fusion (permalink) soit indiqué;
- Le contenu ne soit pas modifié.

L'œuvre ne peut être stockée dans une autre base de données dans le but d'y donner accès ; l'identifiant unique (permalink) indiqué ci-dessus doit toujours être utilisé pour donner accès à l'œuvre. Toute autre utilisation non mentionnée ci-dessus nécessite l'autorisation de l'auteur de l'œuvre ou de l'ayant droit.

---------- **English Version** ----------

This Ph.D. thesis has been digitized by Université libre de Bruxelles. The author who would disagree on its online availability in DI-fusion is invited to contact the University (di-fusion@ulb.be).

If a native electronic version of the thesis exists, the University can guarantee neither that the present digitized version is identical to the native electronic version, nor that it is the definitive official version of the thesis.

DI-fusion is the Institutional Repository of Université libre de Bruxelles; it collects the research output of the University, available on open access as much as possible. The works included in DI-fusion are protected by the Belgian legislation relating to authors' rights and neighbouring rights. Any user may, without prior permission from the authors or copyright owners, for private usage or for educational or scientific research purposes, to the extent justified by the non-profit activity, read, download or reproduce on paper or on any other media, the articles or fragments of other works, available in DI-fusion, provided:

- The authors, title and full bibliographic details are credited in any copy;
- The unique identifier (permalink) for the original metadata page in DI-fusion is indicated;
- The content is not changed in any way.

It is not permitted to store the work in another database in order to provide access to it; the unique identifier (permalink) indicated above must always be used to provide access to the work. Any other use not mentioned above requires the authors' or copyright owners' permission.

29/09/08

D 03624

Etude bioinformatique du réseau d'interactions entre protéines de transport chez les *Fungi*

Sylvain Brohée
http ://www.bigre.ulb.ac.be/ sylvain/
Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe)
Laboratory of Genome and Network Biology
Université Libre de Bruxelles, Belgium
http://www.bigre.ulb.ac.be/

28 septembre 2008

Universite Libre de Bruxelles
003398781

Table des matières

I Introduction 7

1 Les membranes cellulaires et protéines associées 9
1.1 Caractéristiques et structure des membranes biologiques . . . 9
1.1.1 Structure des membranes . . . 9
1.1.2 Fluidité des membranes . . . 11
1.1.3 Protéines associées aux membranes biologiques . . . 11
1.2 Perméabilité sélective et perméases . . . 12
1.2.1 Diffusion simple . . . 12
1.2.2 Diffusion facilitée . . . 12
1.2.3 Transport actif . . . 12

2 Interactions entre protéines 17
2.1 Introduction . . . 17
2.1.1 Généralités . . . 17
2.1.2 Interactions physiques ou fonctionnelles . . . 17
2.2 Interactions physiques et complexes protéiques . . . 17
2.2.1 Interactions entre protéines membranaires . . . 18
2.2.2 Regulation transcriptionnelle des complexes . . . 19
2.2.3 Méthodes de détection des complexes protéiques à large spectre . . . 21
2.2.4 Le projet *Associoport* . . . 25
2.3 Interactions fonctionnelles entre protéines . . . 25
2.3.1 Approches informatiques de détection d'interactions fonctionnelles . . . 27

3 Les réseaux biologiques 29
3.1 Biologie de systèmes complexes . . . 29
3.1.1 Réductionnisme et complexité . . . 29
3.1.2 Appréhender des phénomènes biologiques complexes . . . 29
3.2 La théorie des graphes . . . 30
3.2.1 Formalisme des graphes . . . 30
3.2.2 Propriétés générales des réseaux biologiques . . . 33
3.2.3 Quelques types de réseaux courament étudiés en biologie . . . 37
3.3 Clustering de graphe . . . 42
3.3.1 Définitions . . . 42
3.3.2 Applications en biologie . . . 42
3.3.3 Clustering de graphes biologiques . . . 44
3.4 Outils informatiques pour l'étude de réseaux biologiques . . . 46
3.4.1 Outils avec interface graphique . . . 47
3.4.2 Outils en ligne de commande . . . 48

4 Objectifs 51

II Résultats 53

5 Développement d'outils consacrés à l'analyse de graphes 55
5.1 Historique des outils . . . 55
5.2 Description des articles présentés . . . 57
5.2.1 Présentation du serveur Web de *NeAT* . . . 57
5.2.2 Protocole d'utilisation de quelques outils de *NeAT* . . . 66

6 Ré-implémentation de YTPdb 81
6.1 Bases de données de transporteurs . . . 81
6.2 Specificités de YTPdb . . . 81
6.2.1 Contenu . . . 83
6.2.2 Accès et interface . . . 83
6.2.3 Conclusions et perspectives . . . 83

7 Evaluation de méthodes de clustering de graphe 85
7.1 Intérêt de l'évaluation . . . 85
7.2 Description de l'article présenté . . . 85
7.2.1 Description des algorithmes évalués . . . 86
7.2.2 Principe de l'évaluation . . . 86
7.2.3 Résultats . . . 87
7.2.4 Conclusion . . . 87
7.3 Erratum . . . 107

8 Etudes de données expérimentales 111
8.1 Introduction . . . 111
8.2 Procédure bioinformatique . . . 111
8.2.1 Estimation de la qualité des données . . . 111
8.2.2 Traitement statistiques des données . . . 114
8.2.3 Analyse du réseau d'interactions . . . 117
8.3 Conclusion . . . 126

9 Profils phylogénétiques 127
9.1 Méthodes . . . 127
9.1.1 Installation de génomes fongiques . . . 127
9.1.2 Recherche des protéines orthologues . . . 127
9.1.3 Etude des profils phylogénétiques . . . 128
9.1.4 Evaluation . . . 129
9.2 Evaluation de notre approche . . . 129
9.2.1 *Fungi* . . . 129
9.2.2 Fungi, Bactéries et Archae . . . 132
9.2.3 Filtrage des organismes . . . 133
9.3 Interactions fonctionnelles impliquant des transporteurs . . . 133
9.4 Conclusion . . . 136

10 Réseaux de corégulation 137
10.1 L'approche des empreintes phylogénétiques . . . 137
10.1.1 Découverte de motifs . . . 137
10.1.2 Dyad-analysis . . . 137
10.1.3 Les empreintes phylogénétiques . . . 139
10.2 Description de l'article présenté . . . 141
10.2.1 Méthode utilisée . . . 141
10.2.2 Evaluation du réseau de co-régulation . . . 142

10.2.3 Conclusion . . . 142
10.3 Etude de la co-régulation des transporteurs . . . 180
10.3.1 Réseau induit par les transporteurs . . . 180
10.3.2 Etude des groupes de voisins des transporteurs orphelins . . . 183
10.3.3 Etude expérimentale du gène *YHL008C* . . . 183
10.4 Conclusion . . . 188

III Conclusion générale et Persepectives **191**

11 Conclusion générale et perspectives **193**
11.1 Etude des interactions entre transporteurs de la levure . . . 193
11.2 Etudes de réseaux biologiques . . . 195

IV Annexes **197**

12 Informations supplémentaires sur les sujet abordés dans la thèse **199**
12.1 Aperçu biochimique de la formation de complexes protéiques . . . 199

13 Liste des *Fungi* installés dans RSAT et utilisés dans notre approche de génomique comparative **201**

14 Glossaire **203**

Première partie

Introduction

Chapitre 1

Les membranes cellulaires et protéines associées

Les cellules de tous les organismes présentent une ou plusieurs membranes biologiques. Parmi celles-ci, la membrane plasmique est la démarcation entre une cellule et son environnement. Les cellules procaryotiques (bactéries et archaebactéries) ne présentant pas de compartimentation interne ne possèdent que celle-ci. A l'inverse, les cellules des organismes eucaryotes présentent plusieurs compartiments sous-cellulaires entourés par des membranes biologiques dont la structure et la composition restent globalement semblables quelque soit l'organisme et l'organite.

1.1 Caractéristiques et structure des membranes biologiques

1.1.1 Structure des membranes

Dès la fin du XIXème siècle, Ernest Overton avait observé que les substances liposolubles pénétraient plus rapidement les cellules que les substances non liposolubles et en déduisit que les membranes se composaient de lipides. Quelques années plus tard, l'analyse de membranes isolées à partir de globules rouges démontra que ces membranes étaient formées de phospholipides et de polypeptides [1].

Les phospholipides sont des lipides possédant un groupement phosphate, ils présentent donc une partie hydrophile et une partie hydrophobe (molécule amphiphile). On distingue les glycérophospholipides (Figure 1.3) [2], plus courants et les sphyngolipides dont la structure est basée sur le céramide.

Les polypeptides sont des chaînes plus ou moins longues d'acides aminés reliés entre eux par des liaisons peptidiques. Les polypeptides constituent les protéines qui sont composés d'un ou plusieurs polypeptides reployés. Ils sont

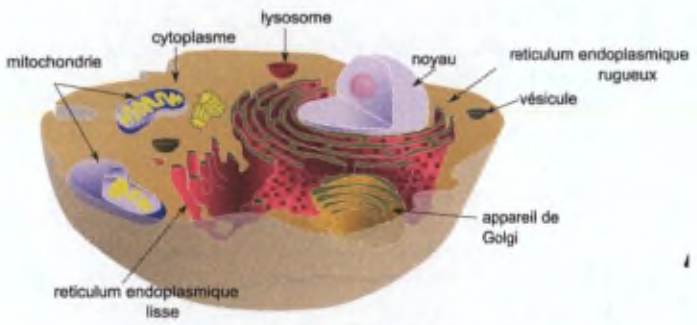


FIG. 1.1 – Membranes et organelles d'une cellule eucaryote. Source Wikipedia

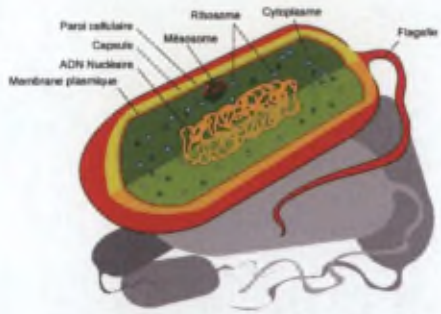


FIG. 1.2 – Structure d'une cellule procaryote. Source Wikipedia

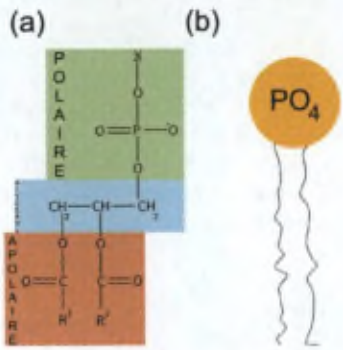


FIG. 1.3 – Représentations schématiques d'un glycérophospholipide. (a) Structure chimique simplifiée d'un glycérophospholipide formé d'un glycérol lié à deux acides gras apolaires et à un groupement phosphate polaire. Les glycérophospholipides varient entre eux selon la nature des acides gras et le radical (X) lié au groupement phosphate. Adapté de Wikipedia. (b) Représentation schématique classique des glycérophospholipides représentant la tête polaire et les deux queues hydrophobes.

synthétisés à partir de l'information génétique codée par le génome et remplissent la grande majorité des fonctions au sein de l'organisme (structure, métabolisme, transport, etc).

Sur cette base, on proposa au début du XXème siècle que les phospholipides s'associent en une double couche constituant une limite stable entre deux compartiments aqueux. En effet, la disposition des molécules abritait de l'eau les queues hydrophobes tout en y exposant les têtes hydrophiles.

La question fut posée de savoir comment les polypeptides étaient associés à ces membranes. Il avait été observé en effet que ces protéines membranaires, contrairement à celles dissoutes dans le milieu cellulaire aqueux (cytosol) comportaient elles aussi une partie hydrophobe et une partie hydrophile. Ainsi, après de multiples controverses, le modèle de la mosaïque fluide fut avancé dans les années 1970. Selon ce modèle, les protéines membranaires sont insérées dans la double couche de lipides permettant aux parties hydrophiles d'en émerger pour entrer en contact avec le milieu aqueux [3] (Figure 1.4a). Ce modèle a subi d'importantes modifications dans les années qui ont suivi. En effet, avec le modèle de la mosaïque fluide apparaît l'idée que les protéines membranaires flottent aléatoirement à faible concentration dans une bicouche lipidique très étendue. Premièrement, il est apparu que la densité de protéines dans la membrane est très importante et que celles-ci ne sont pas dispersées aléatoirement mais au contraire sont présentes dans des régions denses en protéines parfois sous forme de complexes séparés par des régions plus riches en lipides. Ensuite, il semble que l'épaisseur de la membrane varie fortement selon les protéines et les lipides qui la composent. Enfin, il apparait que les protéines occupent une partie non négligeable, bien que non encore quantifiée, de la surface de la membrane d'une façon qui favorise les interactions avec d'autres composants extra (autres cellules,

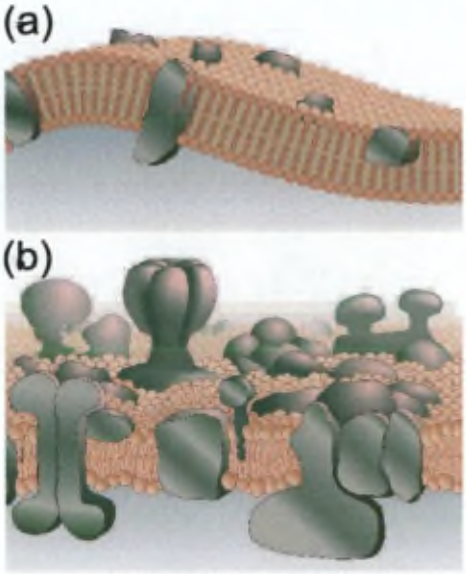


FIG. 1.4 – Représentation schématique de structure de la membrane plasmique. (a) Modèle de la mosaïque fluide [3]. (b) Mise à jour du modèle. Tiré de [4].

protéines, etc) ou intra-cellulaires (cytosquelette, organelles, etc) [4] (Figure 1.4b).

1.1.2 Fluidité des membranes

La fluidité des membranes est nécessaire à leur bon fonctionnement. Ainsi, dans des conditons extrêmes, lorsqu'une membrane se solidifie, sa perméabilité change et les protéines qui y sont insérées peuvent devenir inactives [5].

Les membranes ne sont pas des couches figées où les molécules sont maintenues en place mais présentent au contraire une certaine fluidité. Les lipides et certaines des protéines dérivent assez rapidement latéralement au sein de la membrane. A l'inverse, le passage spontané des phospholipides d'une couche à l'autre est beaucoup plus improbable car la partie hydrophile de la molécule devrait traverser le centre hydrophobe de la membrane (ce mouvement peut cependant être favorisé par certaines enzymes [6]). Enfin, les mouvements des protéines dans les membranes ne sont font pas anarchiquement en raison de contraintes stériques principalement. Il sont surtout le gait le fait de protéines *voyageant* entre différents assemblages protéiques plus complexes [4].

1.1.3 Protéines associées aux membranes biologiques

Comme évoqué précédemment, les cellules eucaryotes présentent plusieurs types de membranes en plus de la membrane plasmique. Celles-ci délimitent les différentes parties de la cellule et permettent leur spécialisation fonctionnelle.

Si la structure de base, la double couche de phospholipides, est conservée pour toutes ces membranes, les protéines et les lipides qui y sont insérés peuvent grandement différer et permettent la spécialisation des compartiments sous-cellulaires.

On distingue deux types de protéines associées aux membranes. D'une part, les *protéines intramembranaires*, qui pénètrent assez profondément la membrane ou la traversent complètement, afin que leur partie hydrophobe soit entourée par les parties hydrocarbonées des phospholipides et d'autre part, les *protéines périphériques* qui ne pénètrent pas dans la membrane mais qui y sont rattachées.

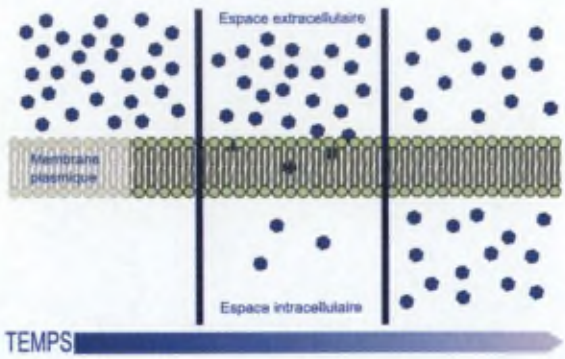


FIG. 1.5 – Diffusion simple au travers de la membrane plasmique

On retiendra que ces protéines peuvent être des récepteurs, des transporteurs, ou encore jouer un rôle enzymatique ou structurel. Dans la suite, nous nous pencherons plus spécialement sur les protéines de membranes permettant le transport de solutés aux travers des membranes biologiques.

1.2 Perméabilité sélective et perméases

Le centre hydrophobe de la double couche de phospholipides ne permet pas le passage des ions et des molécules polaires hydrophiles. Cependant, les molécules hydrophobes comme les hydrocarbures se dissolvent dans la membrane et la traversent aisément. Ainsi, le passage des molécules polaires neutres (saccharides, etc), des ions (Na+, H+) ou encore des molécules chargées n'est possible que grâce aux protéines présentes dans les membranes.

1.2.1 Diffusion simple

La diffusion simple, un phénomène physique passif, est la diffusion dans la membrane plasmique de molécules solubles dans la bicouche phospholipidique capables de traverser directement la membrane. Cette diffusion se fait dans le sens des concentrations fortes vers les concentrations faibles, jusqu'à équilibre des concentrations de part et d'autre de la membrane (Figure 1.5).

1.2.2 Diffusion facilitée

La diffusion facilitée est un mécanisme par lequel certaines substances hydrophiles sont capables de traverser les membranes hydrophobes afin d'entrer ou de sortir de la cellule ou des compartiments cellulaires. En fait, ces molécules évitent le contact avec la double couche lipidique en passant par des protéines intramembranaires : les *protéines de transport*, aussi appelées *transporteurs* ou *perméases*.

A l'inverse de la diffusion simple, la diffusion facilitée (et le transport actif que nous évoquerons dans la suite) présente les caractéristiques des mécanismes enzymatiques. En effet, ce transport peut être **saturé** (la vitesse de diffusion dépend du nombre de perméases présentes à la membrane) et est **spécifique** (les perméases ne sont capables de transporter qu'une seule molécule ou des molécules du même type). Chez la levure, les transporteurs d'hexoses (gènes *HXT*) forment une large famille de transporteurs utilisant la diffusion facilitée [7].

1.2.3 Transport actif

Le transport actif implique le transfert d'un soluté contre le gradient de concentration du compartiment le moins concentré en soluté vers le compartiment le plus concentré par des transporteurs.

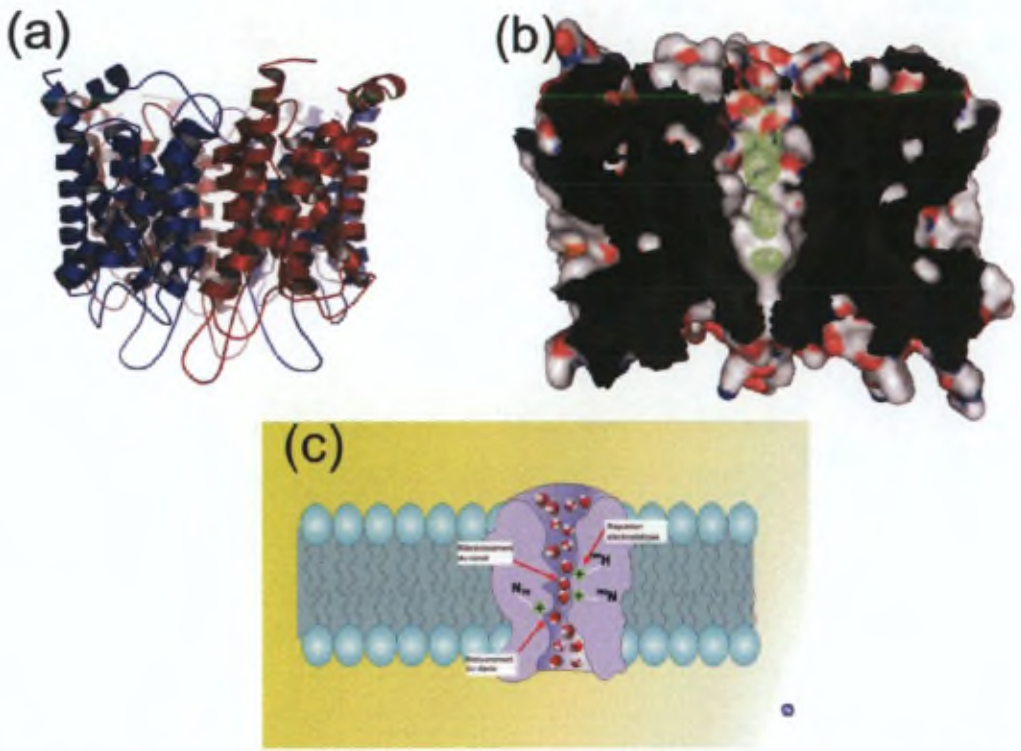


FIG. 1.6 – Exemple de canal membranaire : aquaporine-1 humaine (AQP1) permettant le passage de molécules d'eau au travers des membranes biologiques. (a) Structure cristallographique de AQP1. Le canal est formé par un homotétramère et chacune des sous-unités polypeptidiques de la protéine est représentée par une couleur différente (source Wikipedia obtenu à partir de 1J4N (PDB [8]) et générée par pyMOL). (b) Coupe longitudinale de la molécule d'aquaporine permettant la visualisation du canal hydrophile permettant le passage des molécules d'eau [9]. (c) Coupe schématisée d'une aquaporine. Des phénomènes stériques et électriques permettent le passage spécifique des molécules d'eau dans un étranglement (source Wikipedia).

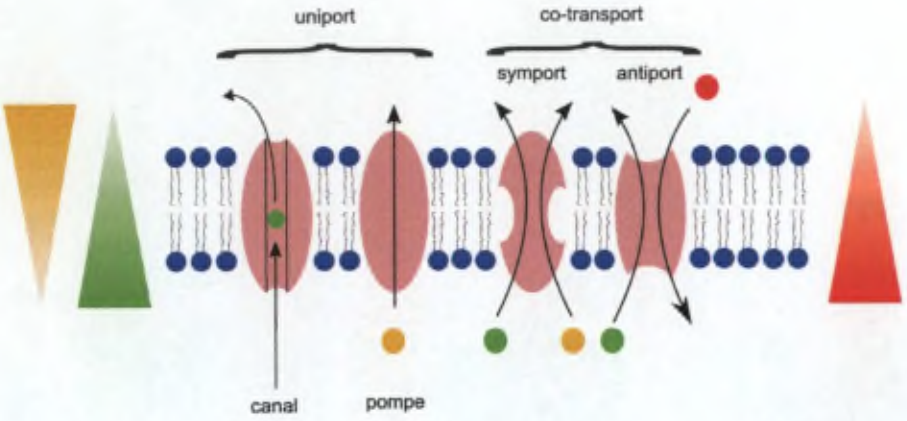


FIG. 1.7 – Types de transporteurs. Les flèches colorées représentent les gradient de concentration dans différents solutés représentés par les diques de mêmes couleurs. Les transporteurs de type uniport (canaux et pompes) ne transportent qu'un seul composé à la fois. Les canaux permettent aux composés de traverser les membranes biologiques du compartiment le plus concentré en ce métabolite vers celui le moins concentré (diffusion facilitée). A l'inverse, les pompes et peuvent transporter un soluté du compartiment le plus concentré vers le moins concentré en ce métabolite en hydrolisant l'ATP ou en utilisant un gradient de concentration favorable.

TAB. 1.1 – Nombre et importance relatives des protéines de transport de chacune des classes principales chez différents organismes modèles. Adapté de [10]

Types de transporteurs	*S.cerevisiae*			*E.coli*			*A.thaliana*			*O.sativa*			*C.elegans*			*D.melanogaster*			*H.sapiens*			*N.crassa*		
	#	% trans	% gènes	#	% trans	% gènes	#	% trans	% gènes	#	% trans	% prot	#	% trans	% prot	#	% trans	% gènes	#	% trans	% gènes	#	% trans	% gènes
Canaux	22	6.85	0.37	15	4.25	0.35	169	16.19	0.52	144	12.85	0.72	240	35.06	0.90	180	28.17	0.91	353	43.96	0.82	14	3.95	0.14
Pompes	70	21.81	1.19	72	20.40	1.68	173	17.51	0.57	245	21.86	1.23	73	11.13	0.29	91	14.24	0.46	99	12.33	0.25	63	17.80	0.64
Phosphotransférase	0	0.00	0.00	29	8.22	0.68	1	0.01	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
Transporteurs secondaires (uniport / symport / antiport)	223	69.47	3.79	233	66.01	5.43	635	64.27	2.08	721	64.32	3.61	349	53.20	1.37	350	56.18	1.82	337	41.97	0.78	271	76.55	2.76
Non classifiées	6	1.87	0.10	4	1.13	0.09	20	2.02	0.07	11	0.98	0.06	4	0.61	0.02	9	1.41	0.05	14	1.74	0.03	6	1.69	0.06
Total	321		5.46	353		8.22	988		3.24	1121		5.61	656		2.57	630		3.23	803		1.87	354		3.61
Nombre total de gènes codants	5880			4294			30480			20000			25537			19759			42937			9817		

On distingue différents types de transporteurs membranaires permettant le transport actif ou la diffusion facilitée : d'une part les perméases de type uniport qui ne transportent qu'un seul soluté au travers de la membrane (les canaux et pompes, notamment) et d'autre part les transporteurs secondaires qui transportent deux solutés simultanément (co-transport). C'est le cas des perméases de type symport qui transportent deux solutés simultanément et dans la même direction et les perméases de type antiport qui transportent deux solutés simultanément mais dans des directions opposées. La Figure 1.7 et le Tableau 1.1 illustrent l'importance relative des différents types de transporteurs que nous allons décrire dans la suite dans plusieurs organismes modèles.

Le transport de solutés à l'encontre du gradient de concentration n'est possible que grâce à l'utilisation de l'énergie métabolique (ATP) ou à l'utilisation par les transporteurs secondaires des gradients de concentration existants.

1.2.3.1 Utilisation d'ATP

Comme pour d'autres formes de travail cellulaire, l'ATP (adénosine tri-phosphate) peut fournir l'énergie nécessaire au transport par une perméase d'un soluté de l'autre côté d'une membrane.

Les transporteurs utilisant l'ATP sont parfois regroupés sous le terme de *pompe*. Chez la levure, nous citerons l'exemple des perméases Pma1p et Pma2p qui utilisent l'ATP afin d'exporter des protons à l'extérieur de la cellule maintenant ainsi le pH cytosolique et le gradient électrostatique de part et d'autre de la membrane plasmique [11].

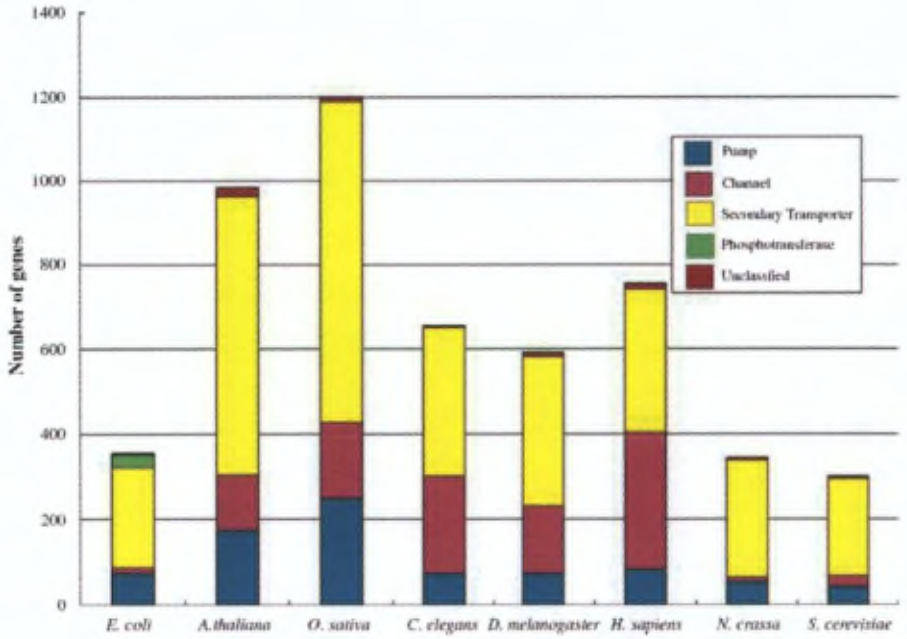


FIG. 1.8 – Nombre de gènes codant chacun des types de transporteurs chez différents organismes entièrement séquencés. On distingue trois classes de transporteurs : les pompes ATP-dépendantes, les canaux et les transporteurs secondaires. Tiré de [10].

1.2.3.2 Transporteurs secondaires

Il existe de nombreux exemples de transport membranaires où l'énergie est fournie par un ion ou une autre molécule qui suit son gradient électrochimique. Il s'agit d'un mécanisme de co-transport qui couple un canal ionique à une pompe membranaire en utilisant l'énergie de l'un pour activer l'autre. L'énergie vient donc du gradient électrochimique, entretenu entre autres par des pompes utilisant l'ATP. On peut donc considérer que c'est bien l'ATP qui a fourni l'énergie, de manière indirecte, d'où le terme de transport primaire pour désigner les pompes ATPasiques et de transport secondaire pour les transports couplés.

Chez la levure, une large majorité des perméases sont des transporteurs secondaires (Figure 1.8, Table 1.1). Parmi ceux-ci, on retrouve notamment les perméases d'acides aminés qui sont vraisemblablement des perméases de type symport protons [12].

1.2.3.3 Les perméases

Les perméases constituent une classe fonctionnelle de protéines parmi les plus importantes dans la grande majorité des organismes. En effet selon les prédictions, cette catégorie de protéines représente jusqu'à 15% du protéome chez certaines bactéries, entre 3% et 5% chez les Fungi et 2% chez l'homme (Figure 1.8, Table 1.1).

Ces protéines occupent une position centrale dans le métabolisme cellulaire et revêtent une importance capitale pour l'industrie pharmaceutique. En effet, certaines affections génétiques sont dues à un dysfonctionnement des transporteurs (voir par exemple [13, 14], ou pour une liste exhaustive [15]). En outre, ces perméases peuvent être ciblées par des molécules pharmaceutiques et il est essentiel de comprendre leur fonctionnement pour délivrer efficacement un médicament au sein l'organisme [16].

La localisation membranaire de ces protéines rendant leur étude plus difficile, elles sont généralement moins bien caractérisées que les protéines solubles.

Parmi près de 6000 gènes qui codent pour des protéines chez la levure *Saccharomyces cerevisiae*, plus 300 codent pour des perméases [12, 17, 10] et pour près de la moitié, aucune fonction ou substrat n'a pu être identifié (Table 1.1).

Chapitre 2

Interactions entre protéines

2.1 Introduction

2.1.1 Généralités

Longtemps, les biologistes ont considéré les cellules comme un gigantesque conteneur où des molécules diffusaient librement et où les réactions entre celles-ci n'étaient dues qu'au hasard des rencontres. Aujourd'hui, on sait par exemple que la plupart des processus cellulaires de base (réplication et transcription de l'ADN, dégradation des protéines, etc) sont menés par des assemblages d'au moins 10 protéines, chacun de ces assemblages interagissant eux-mêmes avec d'autres complexes protéiques [18, 19]. Par ailleurs, de nombreux complexes de plus petite taille ont aussi été identifiés et couvrent une large gamme de fonctions cellulaires allant des voies de biosynthèse aux transports entre compartiments cellulaires (Figure 2.1).

2.1.2 Interactions physiques ou fonctionnelles

Dans la littérature, il est fréquemment fait mention d'interactions entre protéines sans qu'une limite ne soit cependant bien établie entre les différents types d'interactions existantes. Intuitivement, le terme *interaction protéique* ne devrait regrouper que les interactions produites par un contact physique entre les surfaces de plusieurs protéines. Cependant, un nombre important de méthodes informatiques de prédiction d'interactions ont été élaborées et les prédictions effectuées par ces méthodes utilisent le terme d'interactions entre protéines à un niveau plus large dès lors qu'il s'agit en fait de *relations* entre protéines. Ces relations peuvent être très différentes : inclusion dans le même complexe protéique, le même compartiment sous-cellulaire ou la même voie métabolique, la co-expression ou encore la co-évolution [21].

Dans la suite, nous parlerons d'*interaction physique* pour décrire les interactions entre protéines appartenant au même complexe protéique (même si les protéines ne se touchent pas réellement dans la cellule). Le terme d'*interactions fonctionnelles* sera utilisé pour tout autre type de relations entre protéines. Nous décrirons quelques unes des méthodes permettant de les inférer dans la deuxième partie de ce chapitre.

2.2 Interactions physiques et complexes protéiques

Les interactions entre polypeptides se retrouvent à tous les niveaux cellulaires. Dans les cascades de transductions de signaux, des protéines kinases activent d'autres kinases en s'y fixant et en leur ajoutant un groupement phophate [22], certaines voies métaboliques sont constituées d'une suite d'enzymes interagissant physiquement afin d'en effectuer les différentes étapes (complexe formé par Trp2p et Trp3p, deux enzymes de la voie de biosynthèse du tryptophane chez la levure)[23], les cellules s'accrochent les unes aux autres grâce à l'interaction extracellulaire de protéines d'adhérence [24], certaines perméases comme le transporteur de cuivre, Ctr3p ne sont actives que sous forme de complexes [25], etc.

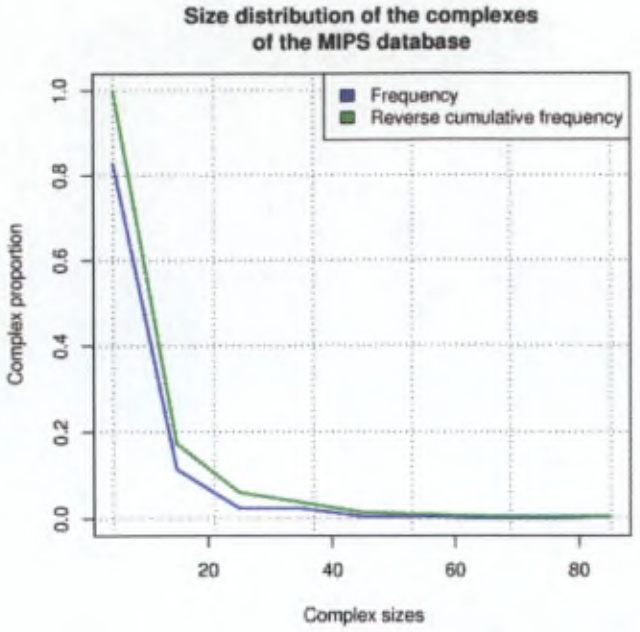


FIG. 2.1 – Distribution du nombre de protéine par complexe dans la base de données MIPS [20].

Certaines de ces interactions sont *stables* (on parle de complexe stable ou permanent), c'est-à-dire que les protéines sont toujours présentes assemblées dans la cellule, généralement pour accomplir une fonction biologique commune. C'est le cas des gros complexes évoqués plus haut (protéasome 26s, etc) (Figure 2.2) mais aussi de complexes plus petits comme le complexe trimérique Ctr3p déjà évoqué. D'autres interactions sont dites *transitoires* : les protéines ne sont pas systématiquement associées. A titre d'exemple, on cite souvent le cas des protéines impliquées dans les cascades de signalisation [21].

2.2.1 Interactions entre protéines membranaires

Un nombre croissant d'études mettent à jour l'importance des interactions protéine-protéine dans la fonction et la régulation des protéines de transport.

A titre d'exemple, on peut observer que la majorité des structures disponibles dans la base de données de structures connues de protéines transmembranaires, sont en fait des oligomères (*http ://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html*).

Chez la levure, on peut citer les exemples des transporteurs de cuivre Ctr3p [25] et de GDP-mannose Vrg4p [26] connus pour former des homo-oligomères. Par ailleurs, il semblerait que les transporteurs mitochondriaux forment des dimères (bien que ce soit toujours controversé) [27], et que les canaux de la famille des aquaporines s'assemblent sous forme de tétramères [12]. Par ailleurs, on a pu démontrer chez les plantes et la levure que les transporteurs d'ammonium de la famille *MEP* présentaient des interactions physiques ou fonctionnelles [28, 29].

Enfin, les perméases présentent aussi des interactions transitoires avec d'autres protéines au cours de leur sécrétion. Citons par exemple le cas des perméases d'acides aminés qui interagissent de façon transitoire avec Shr3p, une protéine membranaire de l'appareil de Golgi requise spécifiquement pour l'empaquetage de ces perméases dans les vésicules de sécrétion [30].

Chez la levure, un certain nombre d'études à grande échelle ont été menées afin de détecter systématiquement les interactions entre protéines soit au moyen de la méthode du double hybride [31, 32, 33], soit en identifiant les composants d'un grand nombre de complexes protéiques par spectrométrie de masse [34, 35, 36, 37]. Les méthodes à

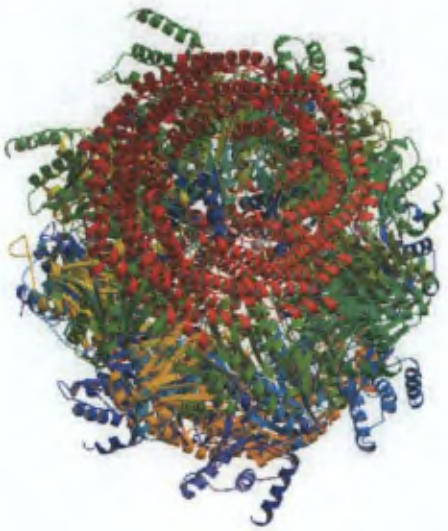


FIG. 2.2 – Représentation schématique des différents polypeptides formant le protéasome 26s chez le trypanosome. Chaque polypeptide composant le complexe présente une couleur différente. Extrait de la PDB (structure 1FNT).

grande échelle seront décrites dans la suite.

Cependant, les données relatives aux perméases sont largement sous-représentées dans les études de caractérisation à grande échelle de l'interactome [38]. Outre les problèmes liés à l'utilisation de techniques à large spectre, ceci peut s'expliquer par la nature hydrophobe et la localisation membranaire de ces protéines, et donc par une plus grande difficulté de manipulation.

2.2.2 Regulation transcriptionnelle des complexes

2.2.2.1 Expression génétique et transcription

L'ADN contient l'information génétique nécessaire à la fabrication des protéines qui sont les principaux acteurs cellulaires. La modulation de leur synthèse, en partant des gènes codés par l'ADN revêt une importance capitale pour la cellule. En effet, selon la situation, la cellule réagit différemment en *allumant* ou en *éteignant* les gènes de façon à répondre aux exigences environnementales ou développementales. Notons qu'il existe des gènes codants pour des ARN qui ne seront pas exprimés par la suite sous forme de protéines (ARN de transfert, ARN ribosomal) et que nous n'aborderons pas ici.

On peut définir l'**expression d'un gène** comme le processus par lequel l'information contenue dans ce gène est décodée pour amener l'assemblage d'une protéine. Elle implique plusieurs étapes schématisées à la Figure 2.3.

Classiquement, une perturbation environnementale amène une réponse cellulaire qui entraîne une cascade d'interactions biochimiques jusqu'à l'activation d'un facteur transcriptionnel. Celui-ci pénètre à l'intérieur du noyau et se fixe à une courte séquence sur l'ADN, appelée élément cis régulateur. Ce sont ces facteurs de transcription qui induisent ou inhibent la transcription. On considère généralement que l'élément cis-régulateur est situé en amont du gène et qu'il permet à l'ARN polymérase, le complexe enzymatique responsable de la transcription, de se placer au niveau du début de la séquence à transcrire. Cependant, de nombreuses études ont mis au jour l'existence d'éléments cis-régulateurs en aval voire au sein du gène même.

La **transcription** est un processus biologique qui consiste en la copie des séquences génétiques d'ADN en ARN. Suite à cette étape de transcription, un ARN pré-messager est produit. On estime que c'est à ce niveau que s'effectue principalement la régulation génétique, c'est-à-dire le contrôle de l'expression des gènes. Elle dépend principalement

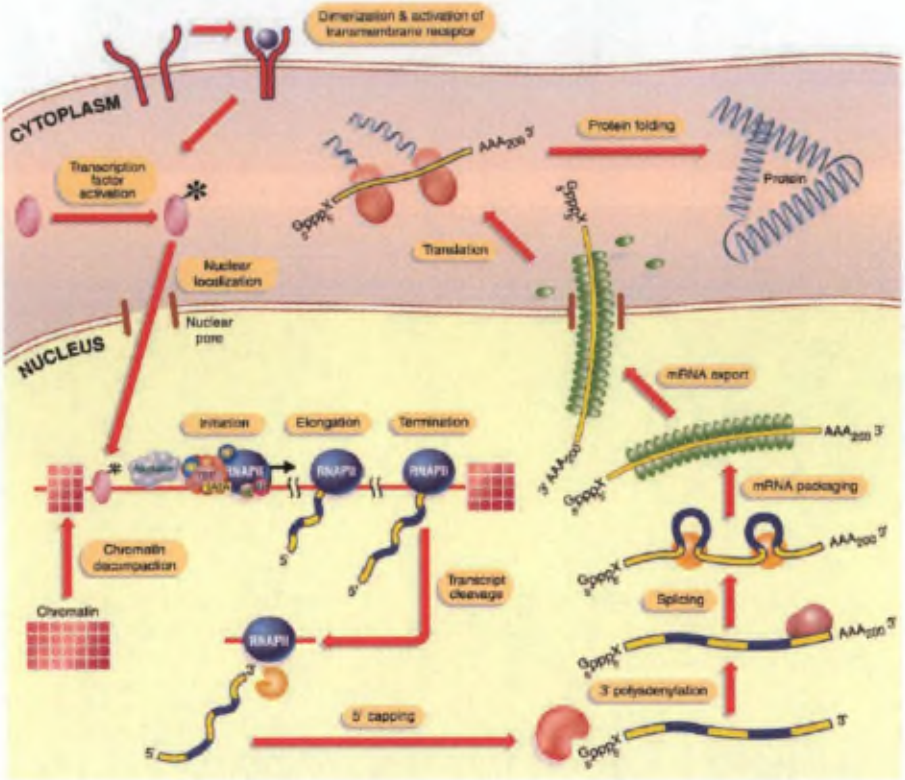


FIG. 2.3 – Vision classique de l'expression d'un gène eucaryote. Tiré de [42].

du facteur de transcription qui amène une activation ou une inhibition de la séquence du gène en en facilitant la fixation du complexe de transcription ou au contraire en l'empêchant.

Chez les eucaryotes, l'ARN produit après la transcription subit une étape de maturation (ajout d'une coiffe en 5' et d'une queue poly-A en 3', épissage des introns) avant d'être empaqueté et exporté hors du noyau.

Le ribosome assure ensuite la traduction de l'ARNm en polypeptide et le libère dans le cytoplasme où celui-ci adoptera sa structure tri-dimensionnelle définitive et subira éventuellement des modifications post-traductionnelles (phosphorylation, acylation, clivage, etc) avant d'arriver à sa forme finale. Le cas des protéines membranaires est un peu différent. En effet, pour la plupart, ces protéines empruntent la voie de sécrétion. Dans ce cas, le ribosome s'associe au réticulum endoplasmique et les domaines transmembranaires du peptide naissant sont directement orientés et insérés dans la bicouche lipidique du réticulum [39]. Les modifications post-traductionnelles ont lieu dans le réticulum endoplamsique ou par la suite dans l'appareil de Golgi. Les protéines membranaires, toujours incluses dans les membranes, sont ensuite envoyées dans des vésicules de sécrétion issus de l'appareil de Golgi qui fussionnent finalement avec la membrane de destination (plasmique ou d'un autre organite). Notons que les protéines de la membrane mitochondriales n'empruntent pas la voie de sécrétion. Le ribosome est associé à la membrane mitochondriale et les protéines sont insérées dans ou l'une des membranes mitochondriales par l'intermédiaires des complexes TIM et TOM [40].

Notons que si les différentes étapes décrites ici ont l'air bien compartimentées, plusieurs études ont mis en évidence que ces différentes phases présentent des liens fonctionnels et physiques étroits. Par exemple, l'ARN polymerase est nécessaire au recrutement des enzymes impliquées dans la maturation du pré-ARNm [41, 42].

2.2.2.2 Régulation transcriptionnelle des complexes protéiques

L'expression des polypeptides qui composent les complexes protéiques est finement régulée au niveau transcriptionnel. En effet, les protéines de certains complexes qualifiés de stables ou de permanents par certains auteurs sont toujours exprimées ou réprimées ensemble. Il s'agit généralement de gros complexes nécessaires aux fonctions essentielles des cellule comme le ribosome ou le protéasome. Curieusement, il existe aussi des complexes dont les protéines ne sont pas toujours toutes co-exprimées ou co-réprimées voire ne le sont jamais (dans les conditions étudiées jusqu'à présent). Or pour interagir, les protéines doivent nécessairement être présentes en même temps dans la cellule. Ceci pourrait être expliqué par le fait que la régulation transcriptionnelle n'est pas le seul mécanisme de régulation de l'expression des protéines. En effet, les ARNm codant pour les protéines d'un complexe ou les protéines du complexe elles-mêmes pourraient présenter des temps de vie différents et il ne serait donc pas nécessaires que leur gène soit transcrit en même temps. C'est le cas notamment des protéines formant les pores nucléaires qui ne semblent pas co-régulées au niveau transcriptionnel vraisemblablement parce que les protéines qui le composent sont très stables, en raison de quoi la répression des gènes met beaucoup de temps à affecter la quantité de protéines présentes dans la cellule [43, 44].

2.2.3 Méthodes de détection des complexes protéiques à large spectre

Avant que la totalité des séquences génomiques de nombreux organismes ne soit rendue disponible, les interactions physiques entre protéines étaient testées individuellement. La dernière décennie a vu le développement et l'application de méthodes dites à *haut débit* (aussi appelées à *large spectre* ou à *grande échelle*) qui permettent, en théorie, d'identifier tous les complexes protéiques au sein d'un organisme [45, 46].

Les approches à grande échelle principalement utilisées et sont le double hybride [47] et la co-immunoprécipitation couplée à une identification par spectrométrie de masse [48, 49].

2.2.3.1 Double hybride

Cette méthode permet de détecter une interaction entre deux protéines chimériques portant chacune un fragment de l'activateur transcriptionnel de levure Gal4p. Comme d'autres facteurs transcriptionnels, ce facteur de transcription présente une nature modulaire et est constitué d'une part d'un domaine de liaison à l'ADN (*binding domain*, *BD*) et d'un domaine d'activation (*activation domain*, *AD*). Le domaine de liaison permet la reconnaissance et la fixation du facteur à des séquences spécifiques localisées dans le promoteur des gènes codant pour les protéines impliquées dans le métabolisme du galactose tandis que le domaine d'activation recrute l'ARN polymerase qui induira la transcription de ces gènes (Figure 2.4a).

Tester une interaction entre deux protéines par cette méthode consiste dès lors à exprimer ces deux protéines fusionnées au domaine de liaison et au domaine d'activation dans deux souches haploïdes de levure différentes capables de fusionner en une souche diploïde. Si les deux protéines interagissent, le réassemblage du facteur transcriptionnel active la transcription d'un gène rapporteur dans le noyau cellulaire.

Cette technique est applicable à grande échelle par plusieurs approches : soit en testant chaque paire de gènes individuellement [51, 31, 33], Figure 2.4b) soit en utilisant des librairies de clones. On peut ainsi croiser un clone exprimant une protéine fusionnée au domaine d'activation à une librairie de clones exprimant chacune des protéines fusionnées au domaine de liaison ([51, 31, 50], Figure 2.4d) ou croiser deux librairies de clones exprimant tous les gènes fusionnés d'une part au *BD* et d'autre part à l'*AD* ([32], Figure 2.4c). Les plasmides exprimés par les clones capables de pousser sur le milieu de sélection sont ensuite séquencés afin de détecter les paires interagissant.

Le double hybride en levure présente certains inconvénients. Premièrement, comme seulement deux protéines peuvent être traitées à la fois, il n'est pas possible de mettre en évidence des liaisons coopératives qui impliquent des partenaires intermédiaires. Par ailleurs, ce système est limité par un taux important de faux positifs (interactions détectées mais non valides biologiquement) imputables à l'activation des gènes rapporteurs fortuitement ou par les protéines appâts. Le double hybride présente également des faux négatifs (des interactions qui bien que testées ne sont pas détectées) qui sont notamment le fait du mauvais reploiement des protéines de fusion. Par ailleurs, les interactions non détectées peuvent aussi être dues au fait que l'interaction doit être testée dans le noyau cellulaire, et ce même pour des interactions entre protéines membranaires insolubles [38, 52]. Enfin, les protéines exprimées n'ont pas subi de modifications post-traductionnelles, ce qui exclut les interactions dépendantes de ces modifications [53].

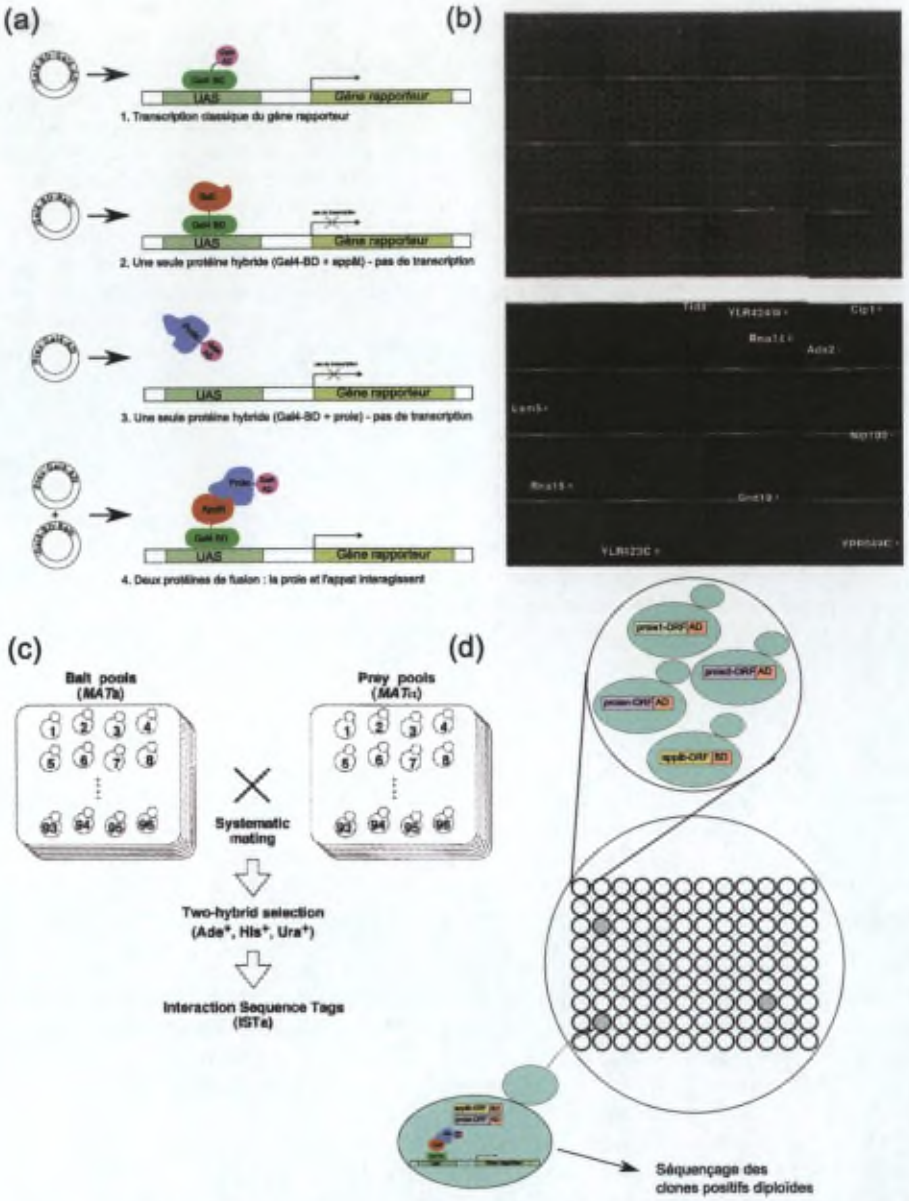


FIG. 2.4 – Technique du double hybride classique. (a) Principe. L'interaction dans le noyau entre deux protéines (protéine *appât* ou *bait* et la protéine *proie* ou *prey*) exprimant chacune une partie de Gal4 (domaine de liaison à l'ADN, *BD* et domaine d'activation, *AD*) reconstitue un activateur fonctionnel ce qui permet l'expression de gènes rapporteurs, généralement nécessaires à la complémentation d'une auxotrophie, c'est à dire, à la croissance sur un milieu minimum. (b) Première expérience de double hybride à grande échelle sur la levure [31] où chaque interaction est systématiquement testée. Dans cet exemple, chaque puits contient le gène *PCF11* fusionné au *AD* (appât) et est croisé avec une souche exprimant un gène différent fusionné au *BD* (proie). En dessous, on observe que seuls quelques clones diploïdes ont poussé sur le milieu de sélection permettant la détection de protéines interagissant avec Pcf11p. Tiré de [31]. (c) Seconde expérience de double hybride à grande échelle [32]. Tous les gènes de levures ont été individuellement fusionnés au *BD* et l'*AD* et divisés ensuite dans des réservoirs de 96 clones. Chacun des réservoirs a été croisé l'un avec l'autre et sélectionné pour l'activation de plusieurs gènes rapporteurs. Les plasmides des clones sélectionnés ont ensuite été séquencés pour déterminer les paires de protéines interagissantes. (d) Dans cette expérience, une souche exprimant un gène fusionné au *BD* (appât) est croisé avec un ensemble de librairies de souches exprimant chacune un gène différent fusionné au *AD*. Dans certains cas, on observe une croissance (puits grisés), les plasmides contenus dans ces clones doivent être séquencés pour connaître le gène interagissant avec la protéine appât. Adapté de [50].

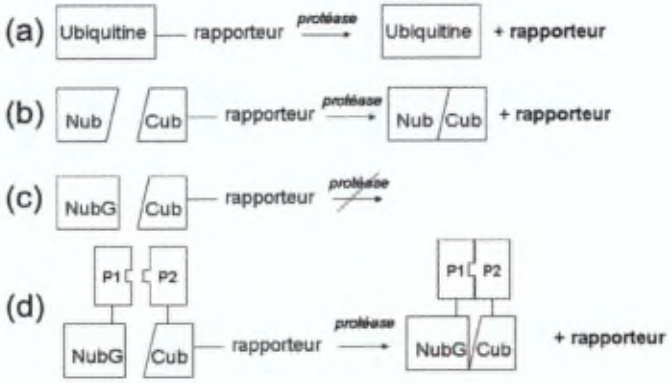


FIG. 2.5 – Principe de la méthode du split-ubiquitin. L'ubiquitine est une petite protéine de 76 acides aminés, extrêmement conservée dans le monde vivant. Son rôle principal consiste à marquer une protéine pour la dégradation en s'y liant. (a) Si l'ubiquitine est exprimée sous forme d'une protéine de fusion avec une protéine rapporteur à son extrémité carboxy-terminale, elle est reconnue par des protéases spécifiques qui induisent le clivage de la liaison entre les deux protéines. Grâce au rapporteur, le clivage peut être visualisé. (b) La protéine de fusion peut être exprimée en deux parties une partie amino-terminale (Nub) et une partie carboxy-terminale (Cub) fusionnée à une protéine rapporteur. Nub et Cub sont ré-assemblés dans la cellule et les protéases spécifiques clivent la liaison entre l'ubiquitine et le rapporteur. (c) Lorsque l'on utilise NubG une version mutée de Nub, le ré-assemblage ne se fait pas spontanément. (d) Si deux protéines P1 et P2 qui interagissent sont fixées à NubG et Cub, alors le contact entre NubG et Cub est forcé et la molécule d'ubiquitine est reconstituée. Cette méthode permet donc de tester l'interaction entre les deux protéinesã : si P1 et P2 n'avaient pas interagi, le rapporteur n'aurait pas pu être visualisé.

De nombreuses variations du double hybride ont été proposées pour contourner les inconvénients de cette méthode. Par exemple, le double hybride en bactérie qui permet l'expression de protéines léthales pour les eucaryotes et ne requiert pas une localisation nucléaire de la paire testée [54]. Une autre méthode, le *split-ubiquitin* permet de tester l'interaction sur son site naturel en reconstituant une petite protéine, appelée ubiquitine [55, 56, 57] (Figure 2.5) .

2.2.3.2 Co-immunoprécipitation couplée à une identification par spectrométrie de masse

La co-immunoprécipitation est basée sur l'observation que la plupart des interactions qui ont lieu *in vivo* sont maintenues si la cellule est lysée dans des conditions non dénaturantes. Si une protéine X est immunoprécipitée en utilisant un anticorps *anti-X*, alors la protéine Y qui forme un complexe stable *in vivo* avec X précipite elle aussi *in vitro*. Les membres du complexe protéique sont ensuite séparés sur un gel d'électrophorèse dénaturant et digérés par trypsine. Ces étapes peuvent être entièrement automatisées à l'aide de robots. Les peptides trypsiques sont ensuite extraits et analysés par spectrométrie de masse (Figure 2.6).

La spectrométrie de masse vise à identifier la composition chimique d'un échantillon en le fragmentant en particules chargées et en mesurant le rapport entre la charge et la masse de ces particules en les passant entre des champs électriques et magnétiques.

Plusieurs méthodes de spectrométrie de masse peuvent être utilisées. Par exemple, le MALDI-MS (pour matrix-assisted laser desorption/ionization mass spectroscopy) qui consiste à comparer les masses des fragments ionisés à des valeurs de masses moléculaires prédites de peptides générés par digestion trypsique théorique [37, 35, 36]. Une autre méthode, le MS/MS (spectroscropie de masse en tandem) consiste à séparer les peptides selon leur taille lors d'une première étape de spectrométrie et à envoyer les peptides ionisé dans un second spectromètre. La technique dite de "Mass matching" ou empreinte de fragmentation vise ensuite à comparer les masses mesurées des peptides fragmentés avec des masses théoriques de fragmentation générées à partir de base de données protéiques [37, 34].

Cette approche est assez peu efficace pour les protéines transmembranaires. En effet, avant de pouvoir précipiter ces protéines, il est nécessaire de solubiliser les membranes (par l'utilisation de détergents) tout en préservant les

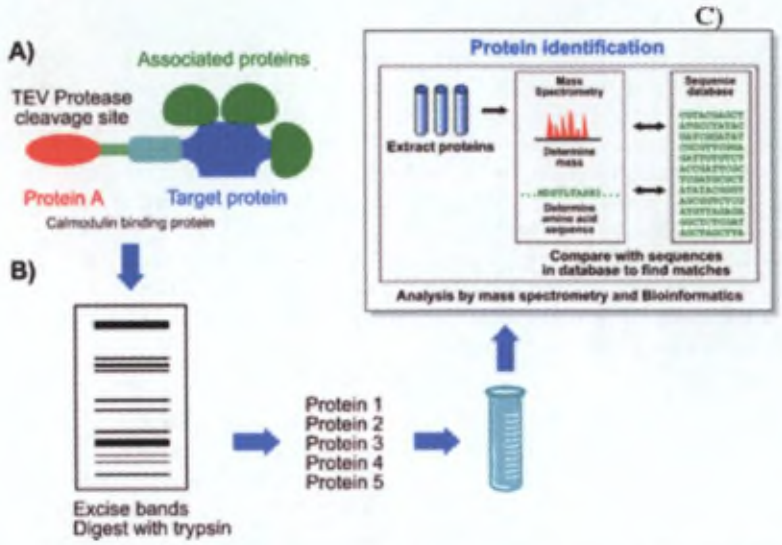


FIG. 2.6 – Analyse des complexes protéiques par spectrométrie de masse à grande échelle. (A) Marquage d'une protéine appât. Les protéines appâts sont systématiquement précipitées avec leurs protéines associées sur une colonne d'affinité. (B) Les complexes protéiques sont dissociés par électrophorèse, ce qui permet de séparer toutes les protéines du complexe en fonction de leur masse. Les protéines sont excisées du gel, digérées par trypsine et analysées par spectrométrie de masse (C) Des algorithmes bioinformatiques sont utilisés pour identifier les protéines en fonction de leur spectre. Tiré de [45]

interactions. Les études évoquées plus haut rapportent cependant chacune plusieurs cas d'interactions impliquant des protéines membranaires.

2.2.4 Le projet *Associoport*

2.2.4.1 Description du projet

Comme nous l'avons déjà souligné, les connaissances concernant les interactions entre protéines membranaires sont assez lacunaires. Afin de combler ce manque d'informations, le projet Associoport, financé par l'U.E., a été mis sur pieds par un consortium de laboratoires européens. Ce projet consiste en une étude systématique des interactions entre transporteurs membranaires au moyen d'une technique novatrice de double hybride : la méthode du *split-ubiquitin* dont nous avons déjà décrit le principe (section 2.2.3.1 p. 23). Grâce à une collaboration avec le laboratoire du Professeur Bruno André (co-promoteur de cette thèse), nous avons eu accès à ces données et leur analyse constituera une partie importante des Résultats.

2.2.4.2 Procédure expérimentale

Pour rappel, la méthode du split-ubiquitin consiste à exprimer les deux protéines pour lesquelles on suspecte une interaction, fusionnées aux parties amino et carboxy-terminales de l'ubiquitine (Nub et Cub, respectivement). Cub est elle-même exprimée fusionnée au PLV, un activateur transcriptionnel artificiel. En cas d'interaction entre les deux protéines testées, Cub et Nub sont rassemblées et forment une molécule d'ubiquitine reconnue par des protéases spécifiques qui vont cliver la liaison peptidique entre l'ubiquitine et le PLV. Le facteur transcriptionnel peut alors pénétrer dans le noyau et activer la transcription des gènes rapporteurs *ADE2* et *HIS3* dont le produit permet la biosynthèse de l'adénine et de l'histidine et donc la croissance sur un milieu carencé en ces deux métabolites (milieu sélectif). Notons que l'expression de Cub-PLV est sous contrôle du promoteur du gène *MET25* et donc partiellement réprimé par la méthionine (Figure 2.7).

Dans le cadre du projet Associoport, les interactions entre les transporteurs membranaires ont été étudiées chez deux organismes modèles : la levure (300 transporteurs) et la plante *A. thaliana* (450 transporteurs). L'étude des interactions chez la levure a été menée dans les laboratoires des professeurs Bruno André (Université Libre de Bruxelles) et Eckhard Boles (Université de Franfort). 279 souches exprimant les protéines membranaires couplées à la partie carboxy-terminale de l'ubiquitine ont été croisées avec 279 autres souches exprimant ces mêmes protéines mais fusionnées cette fois à la partie amino-terminale de l'ubiquitine, chaque protéine fusionnée à Cub étant testée contre 279 protéines fusionnées à Nub sur trois plaques de 96 puits. Différents croisements ont été effectués sur trois milieux différents : un milieu sélectif (milieu minimum de type YNB pour *Yeast Nitrogen Base*, carencé en adénine et en histidine, un milieu non sélectif où ces deux acides aminés étaient ajoutés au milieu et un milieu partiellement répressif ou de la méthionine a été ajoutée au milieu sélectif. D'autre part, au sein des laboratoires des Professeurs Jose Luis Revuelta (Université de Salamanque) et Wolf Frommer (Université de Stanford), des expériences similaires ont été menées entre les transporteurs membranaires de *A. thaliana*. Dans le cadre de cette thèse, nous ne nous sommes intéressés qu'aux perméases de *S.cerevisiae*.

L'étude et l'interprétation de ces données sera effectuée dans la partie Résultats de la thèse.

2.3 Interactions fonctionnelles entre protéines

L'accumulation de données génomiques a permis l'élaboration de nouvelles approches visant à prédire des interactions entre protéines à partir de l'analyse des seules séquences. Bien que certains auteurs se plaisent à laisser plâner le doute quant à la nature des interactions prédites, celles-ci sont principalement des interactions fonctionnelles, impliquées dans une fonction cellulaire commune. Cependant,il peut arriver qu'une interaction fonctionnelle prédite corresponde en fait à une interaction physique.

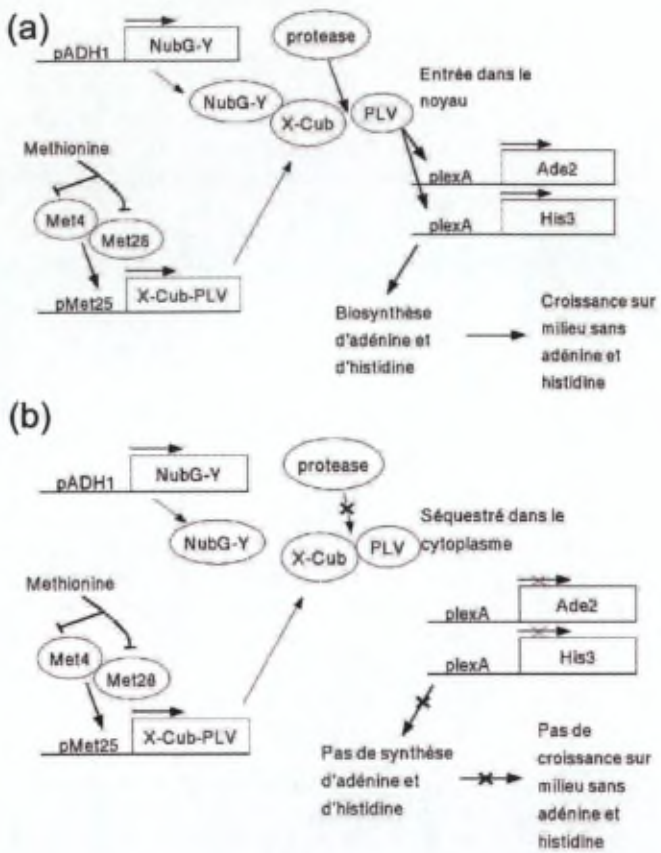


FIG. 2.7 – Principe de la méthode du split ubiquitin mise au point pour le projet Associoport [57]. (a) Deux gènes (X et Y) sont fusionnés respectivement aux parties carboxy (Cub) et amino (NubG) -terminale de l'ubiquitine. Sous contrôle du promoteur du gène *MET25*, la transcription du gène fusionnée à Cub est partiellement inhibée en présence de méthionine par l'intermédiaire des répresseurs transcriptionnels Met4p et Met28p. L'expression du gène fusionné à Nub, sous contrôle du promoteur du gène *ADH1* est constitutive. Dans le cas d'une interaction entre les produits des gènes X-Cub et Nub-Y, une molécule d'ubiquitine est reconstituée. Celle-ci sera reconnue par des protéases spécifiques qui vont libérer l'activateur de transcription PLV. PLV pénétrera dans le noyau et activera la transcription des gènes *ADE2* et *HIS3* nécessaires à la production d'adénine et d'histidine, permettant ainsi la croissance sur un milieu carencé en ces deux métabolites. (b) Lorsqu'il n'y a pas d'interaction entre X et Y, la molécule d'ubiquitine n'est pas reconstituée ni reconnue par les protéases spécifiques. PLV reste fusionné à X-Cub et ne peut pénétrer dans le noyau afin d'activer la transcription d'*ADE2* et *HIS3*. En principe, aucune croissance n'est détectée sur un milieu carencé en adénine et en histidine.

2.3.1 Approches informatiques de détection d'interactions fonctionnelles

2.3.1.1 Profils phylogénétiques

L'hypothèse sur laquelle se base cette approche est que les protéines reliées fonctionnellement, évoluent de manière (anti-)corrélée, et ont donc des orthologues dans le même sous-ensemble d'organismes. Des gènes toujours présents ou absents ensemble ont donc plus de chances de présenter une interaction fonctionnelle. Par cette méthode, chaque gène se voit associer un profil de présence / absence au sein d'un génome et on suppose que les gènes présentant des profils semblables ou inverses, c'est-à-dire, qui sont chaque fois présents ou absents ensemble (ou ne le sont jamais) interagissent fonctionnellement [58, 59] (Figure 2.8b).

2.3.1.2 Fusions de gènes

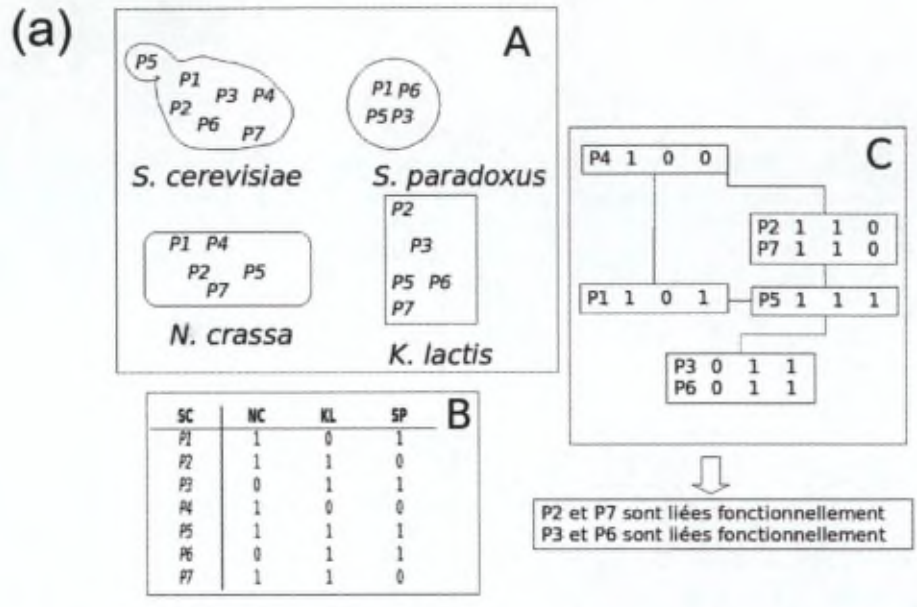
Certaines protéines impliquées dans un même processus (ou interagissant physiquement) ont des orthologues exprimés sous la forme d'une protéine fusionnée dans un autre organisme. C'est le cas, par exemple des *Gyrase A* et *B* de Escherichia coli qui correspond à la *topoiseomérase II* chez *S.cerevisiae*. Sur base de cette observation, Marcotte *et al* ont proposé la méthode des **fusions de gènes** qui consiste à rechercher deux protéines exprimées par deux gènes différents dans un génome mais qui existent sous la forme d'une seule protéine multi-domaine codée par un seul gène dans un autre génome [60]. Dans cette première étude, les auteurs affirment que leur méthode permet à la fois d'inférer la fonction de protéines mais aussi de prédire des interactions physiques. Cependant, ils ne valident avec succès que la prédiction d'interactions fonctionnelles et ne décrivent que quelques d'interactions physiques valides prédits par leur méthode (voir aussi [61], Figure 2.8b).

2.3.1.3 Voisinage de gènes

Il existe aussi des méthodes basées sur l'analyse du voisinage des gènes. L'idée est que des gènes dont les produits de fonction similaires sont localisés proches les uns à côté des autres de manière à leur assurer une régulation conjointe (comme dans le cas des opérons procaryotiques) [62, 63].

2.3.1.4 Co-evolution ligand-récepteur

Des méthodes basées sur la co-évolution ligand-récepteur ont été proposées. Ces méthodes partent du principe que des protéines qui interagissent doivent co-évoluer pour préserver leur interaction et que si une protéine est affectée par une mutation, alors la protéine qui interagit avec elle doit l'être aussi [64, 65].



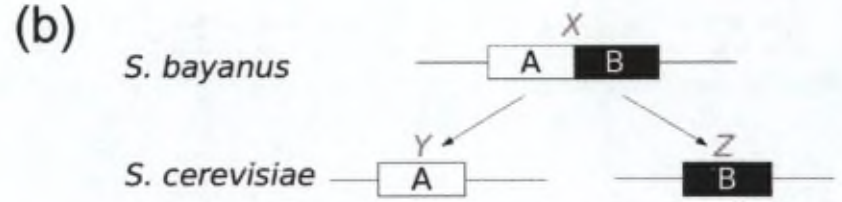


FIG. 2.8 – Méthodes bioinformatiques permettant de prédire des interactions fonctionnelles entre polypeptides. (a) Représentation schématique de la méthode des profils phylogénétiques. Construction d'un profil (B) indiquant quels génomes (A) codent pour un homologue de chacune de protéines (P1 à P7) de *S.cerevisiae*. (C) Clustering afin de déterminer quels protéines partagent le même profil. Les protéines de profil identique sont placés dans la même boîte, tandis que les profils ne différant que pour un seul organisme sont reliés par une ligne. Adapté de [58] (b) Principe des fusions - fissions de protéines. Le gène *X* de *S.bayanus*, une levure proche de *S.cerevisiae*, code pour une protéine formée des domaines A et B. Or chez *S.cerevisiae*, deux gènes différents (respectivement *Y* et *Z*) codent pour *A* et *B*. On peut dès lors émettre l'hypothèse raisonnable que *A* et *B* interagissent au moins fonctionnellement chez *S.cerevisiae*

Chapitre 3

Les réseaux biologiques

3.1 Biologie de systèmes complexes

3.1.1 Réductionnisme et complexité

Le réductionnisme, initié par Descartes au XVIIème siècle est une notion qui touche un grand nombre de domaines. Cette approche prône la compréhension de systèmes complexes en ne les considérant que comme un ensemble de constituants plus fondamentaux.

En biologie, l'analyse réductionniste consiste à étudier un système en séparant ses parties et en les caractérisant. Cette approche prévaut encore aujourd'hui et a déjà permis un nombre considérable de découvertes. Afin de comprendre "la vie", phénomène complexe par excellence, les biologistes ont étudié les phénomènes biochimiques existant au sein des cellules et des organismes. Par exemple, on détermine la fonction d'une protéine en invalidant le gène codant pour celle-ci et en observant le phénotype qui en résulte. Cependant, la vie au sein d'une cellule provient d'une suite d'interactions entre les divers constituants qui la composent.

En 1941, en étudiant l'incapacité de croissance sur milieu minimum de souches de la moisissure *Neurospora crassa* rendues auxotrophes par mutations aux rayons X, Beadle et Tatum émirent la théorie selon laquelle un gène est responsable d'une enzyme qui réalise une fonction [66]. Même si cette découverte est généralement présentée comme celle qui posa les bases de l'approche réductionniste en biologie moléculaire, leurs auteurs reconnaissaient déjà que les composants des systèmes génétiques étaient vraisemblablement connectés entre eux de façon complexe.

En effet, en raison de leur complexité, les systèmes biologiques ne peuvent être totalement caractérisés en n'étudiant leurs constituants que séparément. De nombreux phénomènes (biologiques ou autres) ne s'expliquent pas par la somme de leurs composants mais bien de la manière dont ceux-ci s'assemblent. Les interactions entre les parties du système ainsi que l'influence de l'environnement permettent d'observer de nouvelles propriétés, dites *émergentes* qui sont absentes dans les composants isolés. Afin d'étudier ces propriétés, le système doit donc être appréhendé dans son ensemble (Figure 3.1).

Ainsi, la simple connaissance des composants d'un complexe multiprotéique ou la description des enzymes impliquées dans une voie métabolique ne révèleront le plus souvent leur signification fonctionnelle que lorqu'on sera arrivé à saisir leur agencement spatial ou temporel [67].

3.1.2 Appréhender des phénomènes biologiques complexes

La biologie consiste en partie en la caractérisation des organismes vivants en analysant d'une part leurs constituants et d'autre part les relations entre ceux-ci. Par exemple, on connait de nombreuses protéines capables de se lier à d'autres pour former un complexe ce qui leur confèrera une fonction qu'aucun des membres du complexe ne pourrait effectuer seul. Dans un organisme, il existe des milliers voire des millions de combinaisons d'états (activé, dégradé, etc) et d'interactions entre protéines et il est impossible de les appréhender sans les considérer à un certain niveau d'abstraction [68].

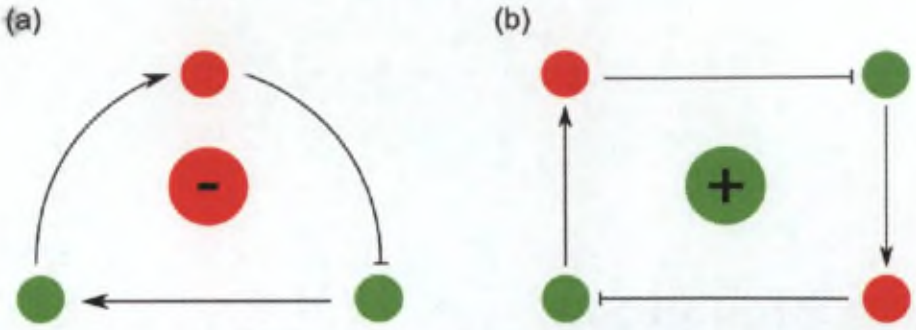


FIG. 3.1 – Illustration de la complexité et des propriétés émergentes. (a) Machinerie de complexité réduite constituée de 3 éléments en interaction deux à deux. Si l'un des agents a un effet négatif et les autres positifs, le motif global constitue une boucle négative. (b) Si un éventuel quatrième composant a échappé à l'attention et que celui-ci a un effet négatif, le système devient une boucle positive [67].

En *bioinformatique* (l'étude de phénomènes biologiques par des moyens informatiques), on représente souvent ces interactions entre les composants d'un système sous forme d'un *graphe*. Cette méthode d'abstraction permet ainsi d'appliquer des méthodes issues d'autres domaines scientifiques (algèbre, statistiques, ...) aux phénomènes biologiques étudiés. Bien que le terme *réseau* ait une toute autre définition en mathématique (voir glossaire), dans le cadre de ce travail, nous utiliserons indifféremment les termes de réseaux ou de graphes.

3.2 La théorie des graphes

La première trace de l'utilisation de la théorie des graphes remonte au XVIIIème siècle avec les travaux d'Euler (Figure 3.2). Le but de cette théorie est d'étudier les relations d'une structure complexe en la représentant sous forme de points (appelés noeuds) et de connexions (arêtes) entre ceux-ci.

3.2.1 Formalisme des graphes

La plupart des systèmes complexes (de la cellule vivante à Internet) dont dérivent certaines propriétés émergentes peuvent être représentés à un niveau plus abstrait par une série de noeuds interconnectés par des arêtes, chaque arête représentant une interaction entre deux éléments. Ensemble, les noeuds et les arêtes constituent un graphe (Figure 3.3a). Les noeuds et arêtes peuvent par ailleurs présenter des attributs comme des couleurs, des poids (Figure 3.3b), des étiquettes. Par exemple, en biologie, il est assez classique de représenter les interactions entre protéines par un graphe : les protéines étant représentées par les noeuds et les interactions par les arêtes entre les noeuds. Un poids, reflétant par exemple la force de l'interaction ou le degré de confiance accordé à cette interaction peut être associées aux arêtes de ce type de graphe.

3.2.1.1 Définition mathématique

Plus formellement, dans la théorie des graphes, un graphe G, est défini comme étant un couple (V, E) où :

- V est un ensemble fini de sommets ou noeuds,
- E est un ensemble de paires de sommets, les arêtes.

Le sous-graphe H de G est un graphe dont les ensembles d'arêtes et de sommets E et V sont des sous-ensembles de ceux de G.

FIG. 3.2 – Les Sept Ponts de Königsberg. La ville de Königsberg (aujourd'hui Kaliningrad) est construite autour de deux îles reliées entre elles par un pont et six ponts relient le continent à l'une ou l'autre des deux îles. Le problème consiste à déterminer s'il existe ou non une promenade dans les rues de Königsberg permettant, à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont, et de revenir à son point de départ. Euler démontra qu'une telle promenade n'existe pas en modélisant les chemins possibles dans la ville par un graphe, introduisant ainsi la théorie des graphes. La forme du graphe peut être altérée (arêtes courbes ou droites, position des noeuds, etc) sans changer le graphe lui même tant que les arêtes entre les noeuds ne changent pas. Euler réalisa que le problème pouvait être résolu en considérant le nombre de voisins des noeuds (leur *degré*). Dans ce graphe, trois noeuds ont un degré de 3 et un a un degré de 5. Euler prouva mathématiquement que le parcours désiré dans ce graphe n'est possible que si le graphe est connecté et que s'il y a exactement zéro ou deux noeuds de degré impairs. Or le graphe correspondant à Königsberg a 4 noeuds de degrés impair.

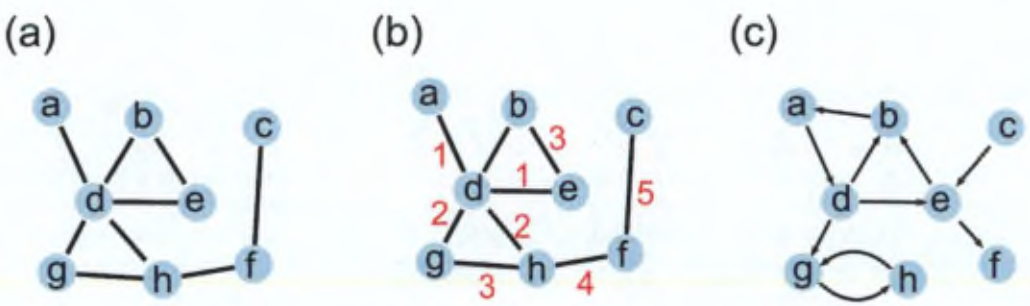
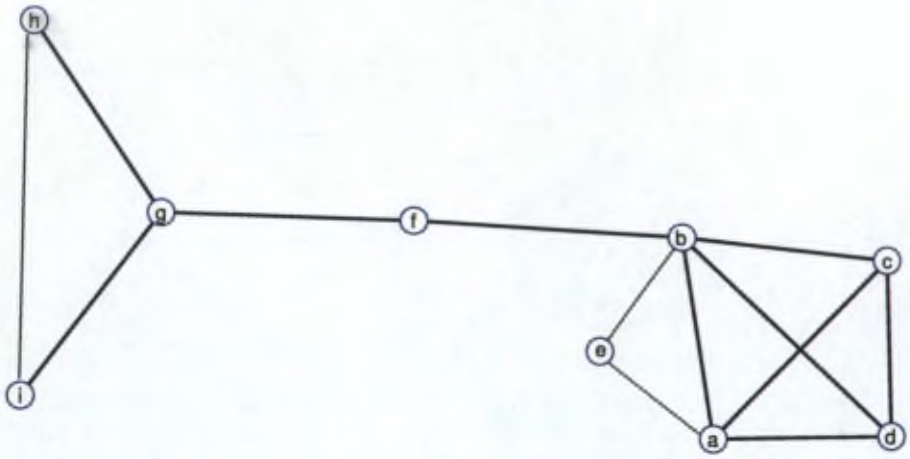


FIG. 3.3 – Représentation graphique classique de graphes. Les noeuds sont représentés par des points et les arêtes par des liens entre les noeuds. Chacun des noeuds de ce graphe a un nom. (a) Graphe simple. (b) Graphe pondéré sur les arêtes, les poids correspondent aux chiffres rouges. (c) Graphe dirigé mais non pondéré.



	Degré	Max nb arêtes entre voisins	Nb arêtes entre voisins	Coef clustering
a	4	6	4	0.67
b	5	10	4	0.4
c	3	3	3	1
d	3	3	3	1
e	2	1	1	1
f	2	1	0	0
g	3	3	1	0.33
h	2	1	1	1
i	2	1	1	1

FIG. 3.4 – Représentation schématique d'un graphe non pondéré non dirigé et calcul du degré et du coefficient de clustering (tableau). Par exemple, le noeud a a quatre voisins, et donc le sous graphe formé de a et de ses voisins (b, c, d et e) peut compter au maximum 6 arêtes, or il n'en compte que deux, le coefficient de clustering est donc de 2/3. Le sous-graphe a, b, c, d forme une clique puisque chacun des noeuds est connecté à l'ensemble de ses voisins.

3.2.1.2 Graphes dirigés

Alors que dans un graphe non dirigé, les arêtes n'ont pas de direction définie, il existe des graphes dirigés qui présentent des liaisons entre les noeuds dont la direction est définie 3.3c). Dès lors, une connexion entre les noeuds a et b n'est plus équivalente à une connexion entre b et a et on ne parle plus d'arêtes mais d'*arcs*. Les graphes dirigés peuvent être utilisés en biologie pour représenter des relations non réciproques comme l'activation d'une protéine par une autre dans un réseau de transduction de signal ou deux étapes enzymatiques successives dans un réseau de voies métaboliques.

3.2.1.3 graphe biparti

Un graphe *biparti* est un graphe comportant deux type de noeuds et dont les arcs ne relient entre eux que des noeuds de types différents. Plus formellement, un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V telle que chaque arête ait une extrémité dans U et l'autre dans V.

3.2.1.4 Degré

Le degré d'un noeud est le nombre d'arêtes qui sont liées à ce noeud (Figure 3.4). Dans les graphes dirigés, on distingue le degré entrant (*in-degree*) et le degré sortant (*out-degree*) qui sont respectivement le nombre d'arcs arrivant au noeud ou provenant d'un noeud.

3.2.1.5 Clique et coefficient de clustering

Une clique est un graphe dont chaque noeud est connecté à l'ensemble des autres noeuds du graphe. Une clique de N noeuds compte donc $N(N-1)/2$ arêtes dans un graphe non dirigé et $N(N-1)$ arcs dans un graphe dirigé. Si le graphe peut contenir des boucles (des arêtes ayant le même noeud de départ que d'arrivée) alors une clique non dirigée de N noeuds comptera $N^2/2$ arêtes et une clique dirigée comptera N^2 arcs (Figure 3.4).

Formellement, le coefficient de clustering C d'un noeud est le rapport entre le nombre d'arêtes existant entre les voisins de ce noeud et le nombre d'arêtes maximal qu'il pourrait y avoir entre ces voisins. Le coefficient de clustering d'un noeud quantifie à quel point l'entourage de ce noeud est proche de former une clique.

3.2.2 Propriétés générales des réseaux biologiques

La dernière décennie a vu l'émergence d'études consacrées à la topologie des réseaux biologiques ou autres. Malgré la grande diversité de types réseaux considérés dans ces études (réseaux biologiques, internet, sociaux), certaines caractéristiques qui leur semblent communes ont pu être dégagées.

Barabasi et ses collaborateurs sont les premiers à avoir publié des études de biologie faisant appel à la théorie des graphes. Leurs travaux, souvent contredits, ne présentent généralement aucune validation et ne tiennent pas toujours compte des réalités biologiques. Néanmoins, ces travaux continuent d'influencer les études dédiées aux réseaux biologiques et nous nous devons de les évoquer dans le cadre de ce travail. Nous nous efforcerons cependant de les considérer avec un esprit critique.

3.2.2.1 Loi de puissance

La topologie d'un graphe peut être caractérisée par la probabilité $P(k)$ qu'un noeud ait k arêtes et de nombreuses études ont été consacrées à la distribution des degrés des noeuds de graphes biologiques.

Dans un graphe où les arêtes sont distribuées aléatoirement entre les noeuds, c'est-à-dire où chaque paire de noeuds présente autant de chances d'être reliée par une arête (graphe aléatoire de type *Erdös-Renyi*) (Figure 3.5Aa), la distribution de $P(k)$ suit une distribution de *Poisson*. En effet, la courbe $P(k)$ en fonction du degré montre un pic important à $k = k_{mean}$ (où k_{mean} est le degré moyen) et décroît pour $k > k_{mean}$ et $k < k_{mean}$ (Figures 3.5Ab et 3.6).

Dans leurs travaux, Barabasi et ses collaborateurs affirment que pour les réseaux qu'ils ont étudiés (Figure 3.5Ba), la distribution des degrés ne suit pas cette règle et $P(k)$ diminuait en suivant une loi de puissance lorsque la connectivité k augmente : $P(k) = k^{-\gamma}$. Le graphique de $P(k)$ en fonction de k avec les axes en échelle logarithmique montre la droite de pente $-\gamma$ (Figure 3.5Bb). Ils appellent les graphes de ce type *power law* [69, 70].

Les réseaux suivant une loi de puissance sont non uniformes puisqu'ils sont composés d'une part d'un grand nombre de noeuds relativement peu connectés et d'autres part de quelques noeuds hautement connectés (appelés *hubs*) qui maintiennent l'ensemble du réseau.

Les réseaux de type power law présentent une autre propriété intéressante : l'effet de *petit monde* (*small world effect*). Quelque soit la taille du réseau considéré, l'existence des hubs permet de relier n'importe quelle paire de noeuds en un petit nombre d'étapes [70]. Bien sûr, cela n'implique pas que ces cours chemins recèlent une quelconque validité biologique.

On dit des réseaux dont la distribution des degrés des noeuds suit une loi de puissance qu'ils sont *libre d'échelle* (ou *scale-free* en anglais). Dans l'absolu, cela signifie que la distribution observée reste de forme identique quelle que soit l'échelle à laquelle on la considère (voir glossaire). Ainsi, par exemple, si dans un graphe dont la distribution des degrés des noeuds suit une loi de puissance, on trouve 5 fois plus de noeuds de degré 1 que de degré 2, on trouvera 5 fois plus de noeuds de degré 10 que de degré 20. Il peut être démontré que seuls les graphes dont la distribution de degrés suit une loi de puissance peuvent satisfaire la propriété de liberté d'échelle. Dans ces réseaux, il n'y a pas de

moyenne significative de liens. Notons que cette propriété ne s'applique qu'à la distribution des degrés des noeuds et que d'autres propriétés sont différentes selon l'échelle [71, 72].

Le paradigme de la liberté d'échelle pour les graphes biologiques se doit d'être tempérées. En effet, dans leur premier article dédié à l'étude de réseaux métaboliques, Jeong et al [70] s'appuient sur les figures des distributions théoriques de Poisson (Figure 3.5Ab) et de la loi de puissance (Figure 3.5Bb) pour affirmer que la distribution des degrés des graphes métaboliques suit bien une loi de puissance car la courbe théorique ressemble plus à cette distribution (Figure 3.7). Or, d'une part, les figures auxquelles ils comparent la distribution des degrés n'ont pas la même échelle (linéaire pour la Figure 3.5Ab et logarithmique pour la Figure 3.5Bb) et d'autre part, pour que la distribution de Poisson ait la forme de la Figure 3.5Ab, sa moyenne doit être située aux alentours de 20 (Figure 3.6(c)) alors qu'une distribution de Poisson dont la moyenne correspond au degré moyen de leur graphe métabolique (environ 2) présente une distribution totalement différente (Figure 3.6(a),(b)). La Figure 3.8 représente la distribution des degrés dans le graphe métabolique de *E.coli* (courbe noire) et son ajustement avec une distribution de Poisson (courbe bleue) et une loi de puissance (courbe verte). Dès lors que l'on compare ce qui est comparable, nous ne pouvons observer sans test statistique que la distribution des degrés suit plus une distribution de Poisson qu'une loi de puissance.

La présence de hubs et la distribution des degrés selon une loi de puissance a considérablement influencé nombre d'études de biologie et plusieurs articles exploitent ces propriétés pour établir ou justifier leur modèles évolutifs. Par exemple, selon Barabasi et ses collaborateurs, l'existence des graphes scale-free s'explique par deux processus conjoints : la croissance et l'attachement préférentiel. Au cours de l'évolution, les réseaux biologiques se sont agrandi en acquerrant de nouveaux noeuds (nouveau gènes issus de duplication, capacité à traiter de nouveaux substrats, etc). Cependant, ces nouveaux noeuds ne sont pas liés aux autres de manière aléatoire mais ont eu tendance à lier les noeuds déjà plus fortement connectés [73]. Cette affirmation, très générale et non validée, est cependant difficilement envisageable pour certains types de graphes (graphes métaboliques).

Il se pourrait cependant que la nature scale-free des graphes biologiques soit un effet de l'échantillonage du réseau complet. En effet, il est fréquent que seule une fraction des interactions entre les noeuds (quelque soit le réseau) ne soit connue et que cette fraction soit trop limitée pour rendre compte de la topologie globale du réseau [74].

Par ailleurs, plusieurs études récentes indiquent que la loi de puissance n'est sans doute pas la plus adaptée pour modéliser le réseau. Parmi celles-ci, citons Khanin et Wit [75] qui ont prouvé stastiquement que plusieurs graphes biologiques qualifiés de scale-free uniquement sur base de la ressemblance de leur distribution de degrés avec la distribution théorique ne l'étaient pas. Il s'agit donc d'être particulièrement critique envers les hypothèses basée sur le dogme non fondé de la nature scale free des réseaux biologiques.

3.2.2.2 Réseaux modulaires

Barabasi *et al* [73] notent que dans les graphes biologiques qu'ils ont étudiés, le coefficient de clustering moyen élevé C_{mean} est élevé en comparaison avec celui de graphes aléatoires. De plus, C_{mean} est indépendant de la taille du réseau. Cela indiquerait que les graphes biologiques présentent un nombre plus elevé qu'attendu de groupes de noeuds hautement interconnectés. L'existence de nombreux modules au sein de réseaux biologiques dérive vraisemblablement du fait que les nombreux composants cellulaires sont interconnectés dynamiquement afin de fonctionner efficacement.

Cependant, l'existence de ce coefficient de clustering élevé pose question car il implique que la plupart des noeuds appartiennent à des modules densément interconnectés et devraient donc approximativement présenter le même nombre d'arêtes. Ceci va à l'encontre de la nature scale-free du réseau. En fait, dans les réseaux biologiques, les modules de noeuds hautement interconnectés et les hubs coexistent naturellement et présentent une organisation hiérarchique, ce qui indique que les modules ne sont pas indépendants et se combinent pour former un réseau hiérarchique. La Figure 3.5Ca montre un exemple de ce type de réseau en assemblant entre eux plusieurs petits modules de 4 noeuds ($k = 6$, $C = 3/4$) qui sont assemblés en plusieurs modules de 16 noeuds ($k = 13$, $C = 2/13$). Ces modules de 16 noeuds sont eux-mêmes combinés en modules de 64 noeuds ($k = 40$, $C = 2/40$). On observe ainsi que plus le degré est élevé, plus le coefficient de clustering est petit contrairement aux autres graphes non modulaires [76].

3.2.2.3 Motifs

Les sous-graphes d'un réseau peuvent présenter des motifs d'interconnexions qui sont significativement surreprésentés. Ainsi, si l'on considère tous les sous-graphes d'un certain nombre de noeuds dans un réseau donné, il

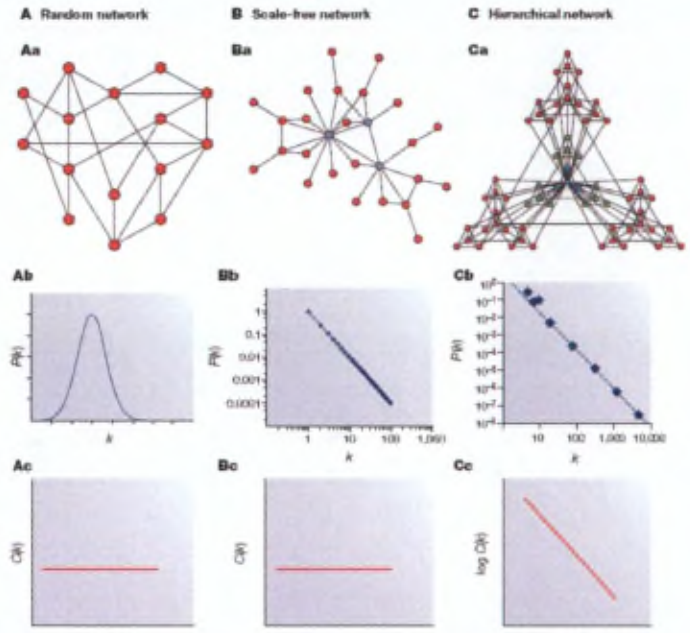


FIG. 3.5 – Fonction de distribution des degrés (deuxième ligne) et coefficient de clusering (troisième ligne) en fonction du degré dans trois types de réseaux. (A) Réseaux aléatoires de types Erdos-Renyi. (B) Réseau scale-free. (C) Réseau scale-free hierarchique. Tiré de [73]. Notons que les axes sont parfois linéaires ou logarithmiques selon les cas (voir discussion).

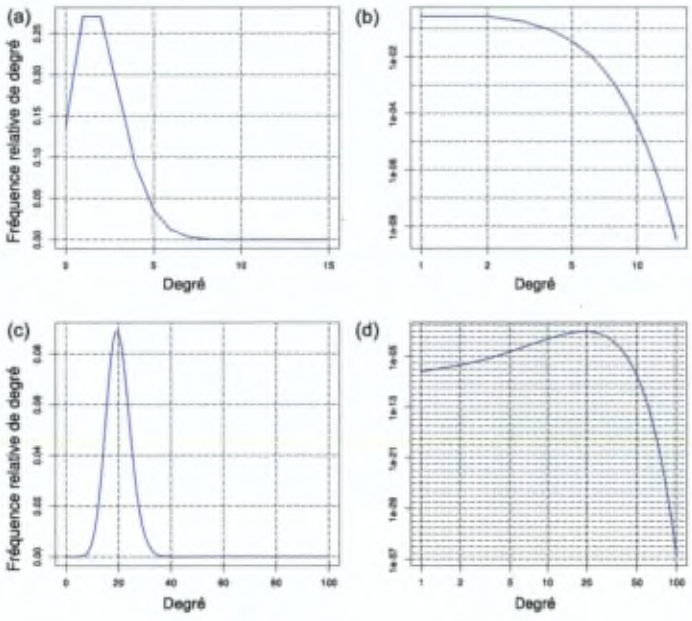


FIG. 3.6 – Distribution théorique de Poisson avec une moyenne de 2 (a-b) et de 20 (c-d) sur des axes linéaires (a-c) ou logarithmiques (b-d)

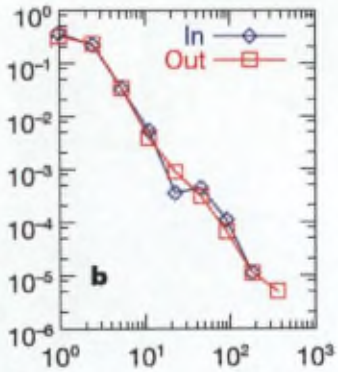


FIG. 3.7 – Distribution des degrés entrant et sortant (sur axes logarithmiques) d'un graphe constitué de voies métaboliques pour *E.coli* présenté par [70].

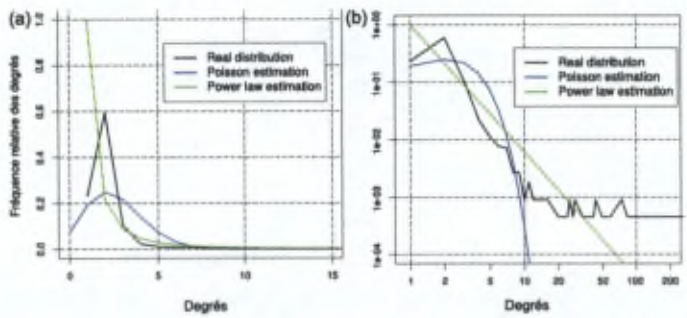


FIG. 3.8 – Ajustement de distribution normale et suivant une loi de puissance sur le graphe des voies métabolique de E.coli (a) Axes non logarithmiques (b) Axes logarithmiquess.

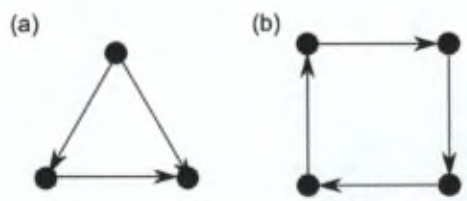


FIG. 3.9 – Exemples de motifs. (a) Motif de rétroaction positive à trois noeuds. Ce motif où un régulateur contrôle un autre régulateur et les deux régulateurs contrôlent un gène cible est statistiquement surreprésenté dans les réseaux de régulations. (b) Motifs de rétroaction positive à quatre noeuds.

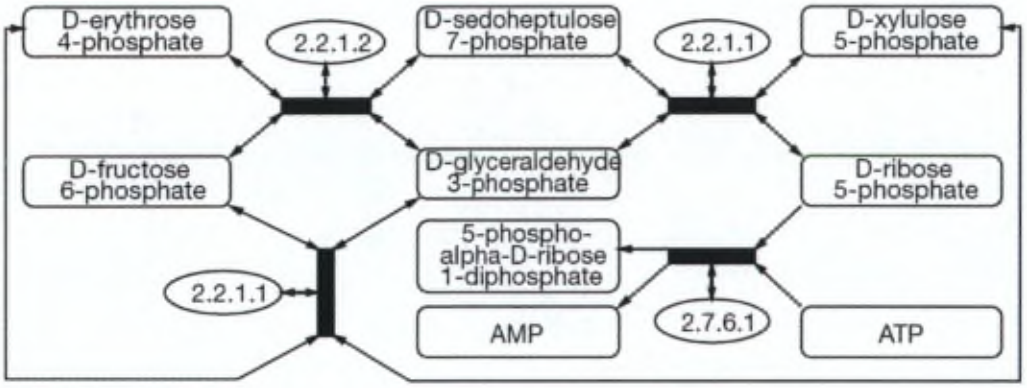


FIG. 3.10 – Représentation de parties du graphe métabolique. (a) Modèle utilisé par Jeong *et al* [83] pour la représentation du graphe métabolique de la bactérie *Escherichia coli*. Chaque métabolite est représenté comme un noeud du graphe relié par des complexes temporaires de substrats auxquelles sont reliées les enzymes permettant la réaction, représentées par leur numéro d'identification E.C. [84]. Tiré de [83].

peut s'avérer que certains types de sous-graphes, que l'on appelle alors *motifs* sont sur-représentés selon le type de graphe considéré. Par exemple, la boucle de rétroaction positive à 3 noeuds (Figure 3.9(a)) est très représenté dans les graphes dirigés de modélisation de la régulation transcriptionnelle ou dans les réseaux de neurones et à l'inverse, des motifs de rétroaction positives à 4 noeuds (Figure 3.9(b)) sont les plus fréquents dans les réseaux électriques [77].

3.2.2.4 Robustesse

Une caractéristique importante de nombreux systèmes complexes est leur robustesse, c'est-à-dire leur capacité à répondre à des changements externes ou internes tout en maintenant globalement un bon fonctionnement.

Grâce aux hubs, on peut retirer aléatoirement une grande proportion des noeuds et conserver cependant une interconnection dense entre les noeuds restants. En effet, puisqu'un graphe scale-free compte principalement des noeuds fort peu connectés, ce sont généralement ces derniers qui sont retirés lorsque on en retire aléatoirement. Cependant, cette robustesse ne repose que sur les hubs et la disparition de quelques noeuds très connectés (attaque ciblée) suffit à diviser le réseau en petits sous-réseau isolés [69].

Enfin, il a été constaté que contrairement à ce qui est observé dans d'autre réseaux, les hubs dans les réseaux biologiques sont d'une part moins connectés entre eux et d'autre part partagent moins de voisins avec d'autres hubs que ce que l'on pourrait attendre au hasard. Ce petit nombre de connexions entre hubs (parfois appelé *anti-corélation* ou *désassortiment*) pourrait permettrait une résistance accrue du réseau aux perturbations. En effet, si une perturbation affecte un noeud et ses voisins proches (parmi lesquelles un hub), celle-ci ne serait pas propagée aux autres hubs qui pourraient amplifier la perturbation en la transmettant à leurs nombreux voisins [78]. Notons que cette observation a déjà été mise plusieurs fois en doute et re-confirmée [79, 80, 81]

3.2.3 Quelques types de réseaux courament étudiés en biologie

Comme nous l'avons déjà indiqué plus haut, bon nombre de processus biologiques complexes peuvent être modélisés avec des réseaux. Dans la suite, nous allons présenter et décrire succintement quelques uns parmi les types de réseaux biologiques les plus courament analysés dans la littérature scientifique [82].

3.2.3.1 Les réseaux métaboliques

Cela fait longtemps que des disciplines aussi diverses que l'ingéniérie ou les sciences sociales utilisent la théorie des graphes pour analyser et modéliser leurs données. Cependant, l'application la théorie des graphes à des ques-

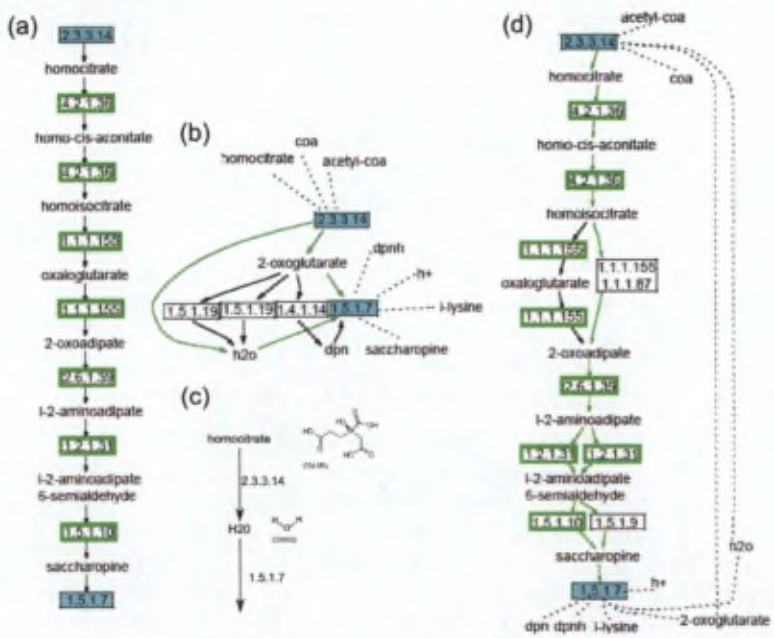


FIG. 3.11 – Inférence de la voie de biosynthèse de la lysine chez *S.cerevisiae* par la recherche des plus courts chemins dans un graphe biparti contenant des noeuds substrats et des noeuds réactions (rectangles) et constitué de l'ensemble des voies métaboliques annotées. Intégration des 5 plus courts chemins retournés par l'algorithme de recherche de chemin décrit dans [85, 86], en utilisant la première et la dernière réaction de la voie annotée comme noeud de début et de fin (rectangles bleus). Les chemins les plus court ont des arcs verts. (a) Voie métabolique annotée. (b) Voies métaboliques inférées dans un graphe non pondéré comparable à celui utilisé par Jeong *et al* [83]. (c) Voie métabolique non biologiquement valide correspondant au chemin le plus court de (b). L'homocitrate est converti en eau en une seule réaction. (d) Les chemin le plus légers obtenus dans le graphe pondéré en fonction du degré des noeuds emprunte 5 des 7 réactions intermédiaires de la voie métabolique annotée. Adapté de [86].

FIG. 3.12 – Représentation d'un réseau d'interaction protéine-protéine chez la levure *S.cerevisiae* provenant des premières expériences de double hybride [31, 32]. Il illustre que quelques noeuds fortement connectés (hubs) suffisent à maintenir le réseau et que les hubs ne sont généralement pas liés l'un à l'autre. Les noeuds sont colorés en fonction de l'effet que provoque la délétion du gène codant pour leur protéine : rouge : délétion léthale ; vert : délétion non léthale ; orange : croissance ralentie ; jaune : effet inconnu.

tions biologiques est apparue assez récemment avec les travaux de Barabasi et de ses collaborateurs sur les réseaux métaboliques (voir notamment [87, 70, 76]). Le graphe choisi pour la modélisation est biparti et dirigé. En effet, le graphe compte des noeuds *réactions* et des noeuds *substrat* et les noeuds *réactions* sont dupliqués pour représenter les sens directs et inverses des réactions réversibles [83] (Figure 3.10). Barabasi et ses collaborateurs établirent certaines des caractéristiques essentielles propres aux réseaux biologiques que nous avons discutées précédemment (loi de puissance, petit monde, etc) à partir de ce modèle [87, 70, 76].

Selon les auteurs, une recherche des chemins les plus courts entre les noeuds du réseau métabolique permet de retrouver les voies métaboliques. Cette idée fut cependant fort controversée par la suite. En effet, il s'est avéré que les voies métaboliques inférées à partir de ces graphes ne correspondent généralement à aucune réalité biologique car les chemins considérés au sein des graphes empruntent généralement les hubs représentant des métabolites fréquemment retrouvés comme co-facteur dans de nombreuses réactions (eau, ATP, NADP, etc) (Figure 3.11b,c) [88, 89, 90].

Pour résoudre ce problème, Croes *et al* [90] ont modélisé le réseau de voies métaboliques au moyen d'un graphe biparti dirigé et pondéré où des noeuds substrats pondérés selon leur degré sont reliés par les arcs au noeuds réactions. Le parcours du graphe en empruntant les noeuds densément connectés pesant plus lourd qu'en empruntant les noeuds moins connectés, l'algorithme de recherche de chemins les plus courts aura tendance à éviter les noeuds fortement connectés. Une évaluation a montré ques les voies métaboliques inférées par cette approche présentent une précision bien supérieure à celles obtenues avec un graphe non pondéré (Figure 3.11d) [90, 86].

3.2.3.2 Les réseaux d'interaction entre protéines

Comme nous l'avons mentionné précédemment, un certain nombre d'études à grande échelle, chez la levure principalement [31, 32, 34, 35, 33, 36, 37, 91] mais aussi chez d'autres organismes modèles [51] et l'humain [50] ont

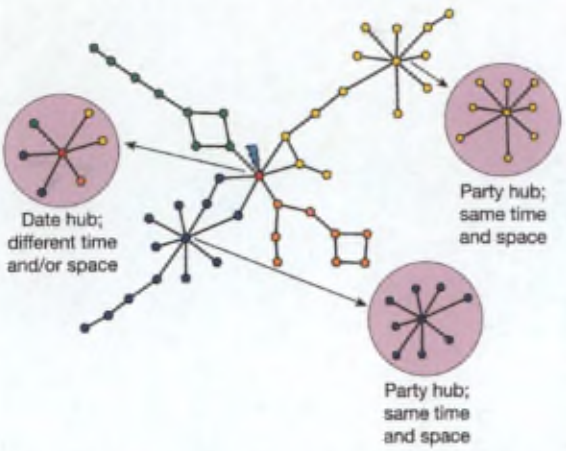


FIG. 3.13 – Date hubs et party hubs. Dans ce réseau d'interaction shématique, les protéines sont colorées selon la similarité de leur profil d'expression avec leurs partenaires. Les party hubs sont hautement corélés et interagissent avec tous leurs voisins au même moment. Les partenaires des date hubs présentent une co-expression plus restreinte correspondant à des interactions ayant lieu à des endroits ou à des moments différents. Tiré de [97].

été menées selon différentes approches afin de détecter les interactions physiques entre protéines. Cela a permis de construire des réseaux représentant l'*interactome*, c'est-à-dire, l'ensemble des interactions physiques entre protéines au sein d'un organisme (Figure 3.12)

Après avoir étudié les réseaux métaboliques, Jeong *et al* se sont penchés sur les réseaux d'interactions physiques entre protéines [83]. Dans cette étude, les auteurs observent que parmi les protéines hautement connectées, on compterait plus de protéines dont la délétion du gène s'avèrerait létale pour l'organisme (protéines dite essentielles). Il semblerait aussi que ces protéines essentielles aient un coefficient de clustering plus élevé [92]. Par ailleurs, les hubs évolueraient à une vitesse moins importante que les autres protéines vraisemblablement parce que l'interaction avec de nombreuses autres protéines imposerait des contraintes structurelles et donc évolutives plus élevées [93]. Ces constatations, bien qu'effectuées par différents groupes indépendants, ont cependant été mises en doute par d'autre études en utilisant des jeux de données qualifiés de plus sûrs par leurs auteurs [94]. Plusieurs explications ont été avancées afin d'expliquer l'essenssialité des hubs. Les hubs rendraient possibles les interactions entre les autres protéines moins connectées du réseau [83]. D'autres chercheurs ont proposé que les protéines ne sont essentielles que parce qu'elles impliquées dans des interactions essentielles. Les interactions essentielles étant distribuées de manières uniforme dans le réseau, les hubs auraient dès lors plus de chances d'être essentiels simplement parce qu'ils présentent plus d'interactions [95]. Enfin, tout récemment, une étude a démontré que les hubs étaient généralement concentrés au sein de modules de protéines densément interconnectées enrichis en protéines essentielles [96].

Certaines études ont permis d'effectuer une distinction entre les hubs sur base de la corelation de leur expression avec celle de leurs voisins. On distinguera les *party hubs* qui sont exprimés en même temps que tous leurs voisins et les *date hubs* qui ne montrent pas de corélation évidente au niveau de l'expression avec leurs partenaires [97]. Cette distinction suggère une certaine organisation de la modularité des réseaux d'interactions. En effet, les date hubs qui semblent plus important pour la conservation de la topologie du réseau constitueraient un lien entre différents modules fonctionnels. A l'inverse, les party hubs seraient des protéines présentes dans des modules fonctionnels distincts et qui bien qu'indispensables à la survie de la cellule seraient moins importantes pour l'organisation du protéome [97] (Figure 3.13). Les date hubs, impliqués dans différentes fonctions cellulaires, présentent une vitesse d'évolution plus importante que les party hubs [98], ceci peut être mis en relation avec la manière dont les protéines sont soumises à la sélection : une fois que les domaines protéiques sont établis, ils restent intacts mais il est fréquent que de nouvelles combinaisons de domaines établis créent de nouvelles protéines. Notons que la distinction date hubs / party hubs est toujours controversée puisque elle est observable ou non selon les jeux de données qualifiés tour à tour de plus sûrs ou de plus complets par leurs auteurs [99, 81, 100]

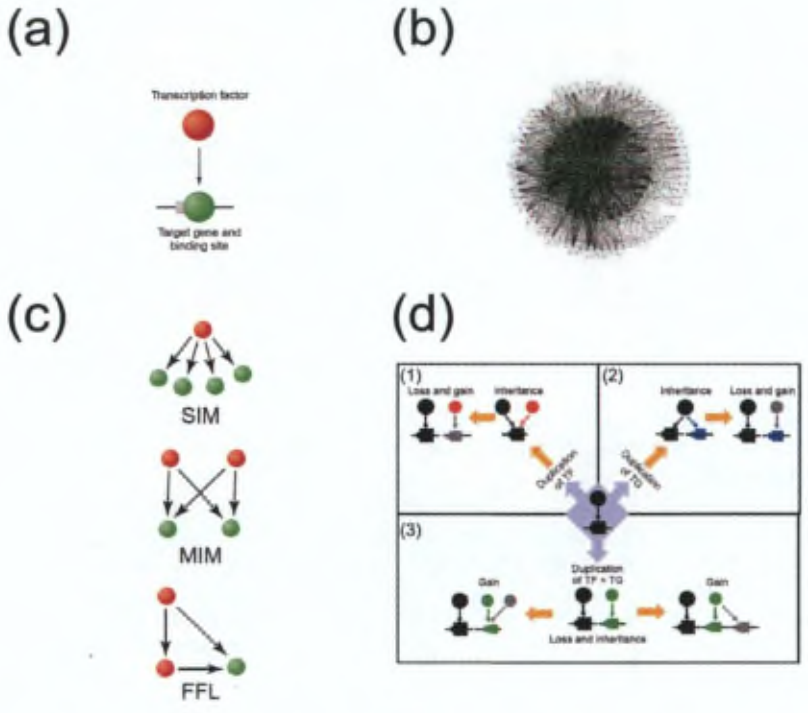


FIG. 3.14 – Réseaux de régulations transcriptionnelles. (a) L'unité de base du réseau comprend le facteur de transcription et son gène cible et l'interaction de régulation entre eux deux. (b) Réseau de régulation transcriptionnelle complet. (c) Exemple de motifs de régulations communs dans le réseau regulation de la levure. (c) Motifs sur-représenté dans réseau de régulation. SIM : single input motif ; MIM : multiple input motif ; FFL : feed forward loop. Tiré de [101]. (d) Croissance par duplication du réseau de régulation. (1) Duplication du régulateur ; (2) Duplication de la cible ; (3) Duplication du régulateur et de la cible. Le réseau peut ensuite évoluer par acquisition d'interactions de régulations ou perte d'interactions existantes [102].

3.2.3.3 Réseaux de régulation transcriptionnelle

Dans un organisme, les relations de régulation entre les facteurs de transcriptions et leurs gènes cibles peuvent être représentées par un graphe dirigé dans lequel les régulateurs et les cibles représentent les noeuds et les interactions de régulations les arcs [101] (Figure 3.14a,b).

Plusieurs études topologiques ont été menées sur ces réseaux. Tout d'abord, au niveau global, il apparaît que dans les réseaux étudiés la distribution des degrés entrant (nombre de facteurs de transcription par cible) semble suivre une loi exponentielle. C'est-à-dire que contrairement aux réseaux scale-free, il existe un nombre moyen significatif de régulateurs par gène. En effet, 93% des gènes sont régulés par un à quatre facteur de transcription. The observed distribution of in-degrees for target genes, in part, possibly reflect the limits on the number of TFs binding to a promoter region due to steric-hindrance between bound TFs. A l'inverse, la distribution des degrés sortant présente une distribution plus proche de la loi de puissance (bien qu'aucun test statistique n'ait été effectué pour l'affirmer [75]). Comme ces gènes représentent la majorité, l'inactivation (par mutation aléatoire, par exemple) a peu de chances d'affecter la régulation d'autres gènes. A l'inverse, l'inactivation d'un des gènes hautement connectés modifierait profondément la structure du réseau. D'ailleurs, un nombre élevé de gènes essentiels à la survie de la cellule sont des gènes situés en amont du réseau de régulation [103].

Ensuite, il apparaît que les réseaux de régulations sont constitués de modules fortement interconnectés. Beaucoup de modules interconnectés sont intégrés hiérarchiquement les uns dans les autres. Ainsi, on a pu montrer que plusieurs

motifs caractéristiques (Figure 3.14c) [104] se chevauchent et partagent donc les mêmes facteurs de transcription et gènes cibles. Cette organisation permettrait une réponse rapide par l'intermédiaire des hubs qui la propageraient rapidement à plusieurs modules densément connectés [105].

L'utilisation de réseaux a entre autres permis de quantifier les mécanismes d'évolution de la régulation transcriptionnelle. En effet, il apparaît qu'environ 90% des relations facteurs-cible au sein du réseau sont apparues suite à la duplication d'un facteur de transcription ou du gène cible. Les relations issues de duplication proviennent alors soit de l'héritage de la relation, soit par l'acquisition de nouvelles interactions régulatrices [102] (Figure 3.14d).

3.2.3.4 Autres réseaux étudiés en biologie

De plus en plus de phénomènes biologiques sont modélisés à l'aide de réseaux et il serait très difficile d'en restituer l'entièreté.

Citons les réseaux de domaines protéiques. Ceux-ci sont construits en reliant dans un réseau les domaines que l'on retrouve au sein de la même protéine dans un même organisme. Ces réseaux de type scale-free ont entre autre permis de mettre en évidence que les protéines des organismes plus complexes avaient tendance à présenter plus de voisins au sein du réseau que les protéines d'organismes moins complexes, ce qui semble indiquer que les protéines des organismes supérieurs sont plus modulaires [106]

Par ailleurs, des études ont porté sur des réseaux composés de voies de transduction de signaux. Ceux-ci présentent une distribution de type scale-free à la différence qu'il y a plus de noeuds de degré 2, 3 ou 4 que de noeuds de degré 1 vraisemblablement parce que les voies de transductions ne sont pas linéaires mais sont le compromis entre plusieurs signaux envoyés par différents récepteurs qui peuvent envoyer des signaux différents [107].

Enfin, d'autres réseaux très divers que nous n'aborderons pas ici ont été modélisés et étudiés : par exemple des réseaux médicament-protéine cibles [108], des réseaux reliant des acides aminés au sein de protéines [109] ou encore des réseaux de neurones [110].

3.3 Clustering de graphe

3.3.1 Définitions

Le clustering est un domaine de l'analyse statistique qui consiste à assigner des éléments à des groupes (ou *clusters*) de façon non supervisée, c'est-à-dire, sans connaissance préalable des groupes ni des éléments. Les regroupements se font généralement sur base d'une mesure de distance calculée en fonction des différentes caractéristiques de chacun des éléments, un clustering optimal visant à minimiser les différences intra-cluster et à maximiser les différences inter-clusters (Figure 3.15(a)). Il existe de nombreuses métriques de distances différentes et le choix peut grandement altérer le résultat de clustering. Lorsque à l'issue du clustering, chaque élément est attribué à un et un seul groupe, on parle de *partitionnement*. Le clustering est une méthode de classification non supervisée, c'est-à-dire que l'algorithme n'a pas au préalable *appris* à reconnaître les éléments qu'il doit classifier. Néanmoins, certaines approches nécessitent la spécification préalable du nombre de groupes que doit retourner l'algorithme. Par ailleurs, presque toutes les méthodes de clustering présentent des paramètres qui peuvent grandement modifier les résultats du clustering. Ces paramètres doivent être choisis judicieusement selon le type de données que l'on manipule [111].

3.3.2 Applications en biologie

En biologie de nombreuses études ont recours au clustering pour la classification automatique de données.

De nombreuses études ont recours au clustering pour la classifications de séquences homologues en familles de gènes. La classification des familles de gènes est un domaine important en biologie permettant entre autres l'annotation automatique de nouvelles séquences génétiques et l'identification de nouveaux domaines protéiques (voir notamment [112, 113, 114]).

Le clustering est très couramment utilisé dans le cadre d'expériences de biopuces à ADN [115]. Cette technique permet de mesurer le degré d'expression de plusieurs dizaines de milliers de gènes à la fois dans différentes conditions. Classiquement, dans ce cas, le clustering permet de classifier les gènes présentant des profils d'expression semblables selon les conditions, ces gènes présentant de bonnes chances d'être impliqués dans des fonctions similaires.

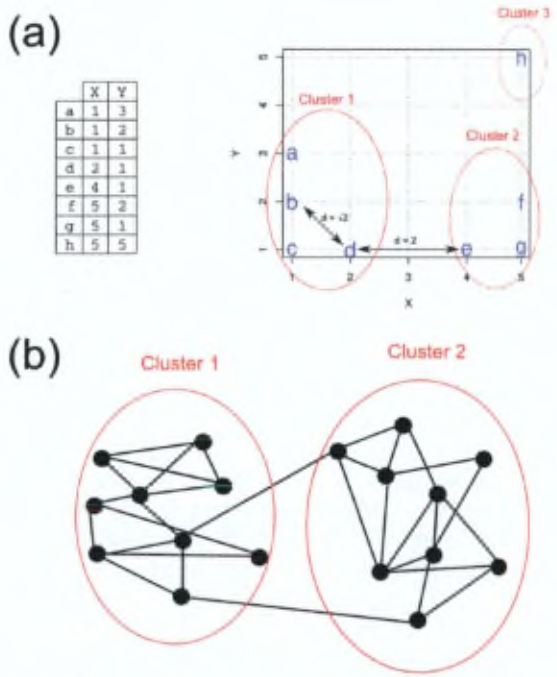


FIG. 3.15 – Le clustering. (a) Clustering d'éléments définis selon deux variables X et Y. Selon une métrique de distance d, les différents éléments sont attribués à des classes distinctes. Dans ce cas-ci, d correspond à la distance euclidienne calculée pour deux points x et y définis par n variables : $d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$. (b) Clustering de graphe. Les noeuds plus densément inter-connectés entre eux qu'avec les autres éléments du réseaux regroupés au sein du même cluster.

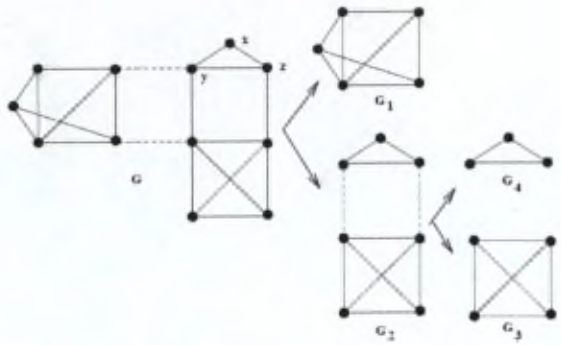


FIG. 3.16 – Algorithme de coupe minimale appliqué au graphe G. A chaque étape, les arêtes correspondants à la coupe minimale sont représentées en pointillés. Tiré de [121].

3.3.3 Clustering de graphes biologiques

Une des applications du clustering dans la théorie des graphes consiste à définir des groupes de noeuds densément interconnectés (clusters ou encore modules), c'est-à-dire présentant plus de connexions entre eux qu'ils n'en présentent avec les autres noeuds du graphe (Figure 3.15(b)).

Le clustering de graphe est notamment appliqué aux réseaux d'interactions protéiques afin d'extraire les complexes protéiques. En outre, la présence de protéines dont le rôle est inconnu au sein d'un cluster densément interconnecté de protéines de fonctions similaires peut révéler de nouveaux indices quant à leur fonction.

Dans la suite, nous allons présenter quelques unes de ces approches ainsi que leurs domaines d'applications [116].

3.3.3.1 Clustering hiérarchique

Le clustering hiérarchique effectue séquentiellement une série de partitionnements sur base d'une métrique de distance en partant soit d'un seul cluster contenant tous les élément soit d'une série de n clusters qui ne contiennent qu'un seul élément. On distingue ainsi les méthodes *agglomératives* qui procèdent en fusionnant itérativement les n objets dans des clusters de plus en plus grands des méthodes *divisives* qui subdivisent un premier cluster contenant les n éléments en une série de clusters imbriqués de plus en plus petits jusqu'à ce que chaque cluster ne contienne plus qu'un seul seul élément.

La métrique la plus intuitive est la longueur du chemin le plus court entre les noeuds du graphe. Cependant, à cause des hunbs, beaucoup de noeuds sont équidistants. Pour contourner ce problème, plusieurs solutions ont été envisagées comme choisir *manuellement* le partitionnement le plus probable parmi plusieurs solutions équivalentes (mais cette solution se révèle difficile pour les graphes de grande taille. D'autres auteurs ont proposé de calculer toutes les solutions équivalentes et d'utiliser comme nouvelle métrique de distance pour une deuxième étape de clustering le nombre de fois où les protéines sont groupées dans le même cluster [117].

D'autres métriques de distances ont aussi été envisagées comme la distance de Czekanovksi-Dice [118] ou la significativité statistique de partenaires d'interactions en commun [119].

Notons que le clustering hiérarchique a aussi été appliqué à un graphe métabolique biparti en regroupant à chaque étape les noeuds présentant le degré les plus faible avec leurs voisins, contournant ainsi l'effet agglomérateur des hubs. Ce clustering a permis d'observer une organisation hiérarchique du métabolisme plaçant les fonctions les plus importantes (synthèse d'ATP, maintien des concentrations ioniques) au sommet de la hiérarchie et plus bas les voies biochimiques individuelles (synthèse, transport) [120].

3.3.3.2 Algorithmes de coupes minimales (minimum cut)

Une coupe est la partition des noeuds d'un graphe dans deux ensembles (ou clusters). La valeur d'une coupe est le nombre d'arêtes inter-cluster ou la somme de leurs poids dans le cas d'un graphe pondéré.

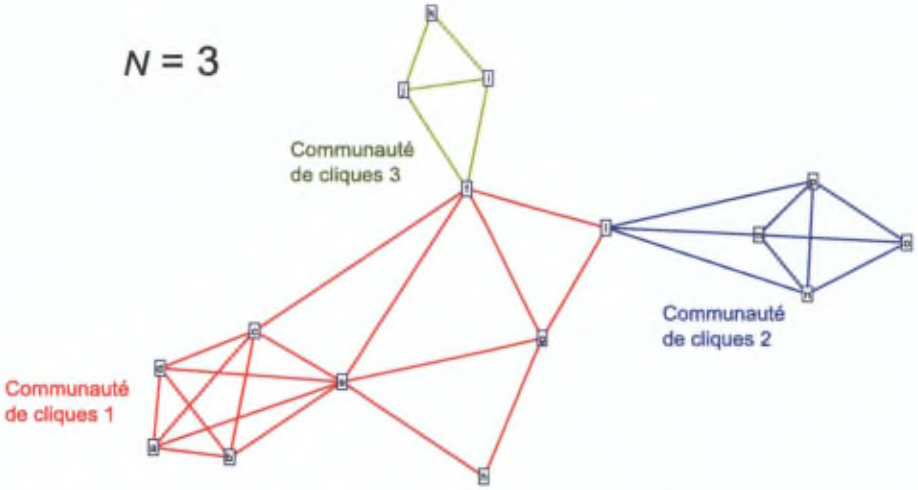


FIG. 3.17 – Recherche de communautés de cliques de taille $N = 3$. Chaque communauté est représentée dans une couleur différente.

Dans une première étape, l'algorithme trouve la coupe minimale qui permettra de séparer le graphe en deux clusters en minimisant le nombre d'arêtes inter-clusters. Chacun des deux sous-graphes sera ensuite traité de la même manière de façon à obtenir en tout quatre sous-graphes et ainsi de suite jusqu'à ce que les clusters ne contiennent plus qu'un élément ou jusqu'à ce qu'une condition d'arrêt (qui peut différer selon l'algorithme) soit atteinte [121, 122] (Figure 3.16).

Cette approche a été appliquée notamment pour la classification de profils d'expression de gènes et la découverte de modules connectés au sein de réseaux d'interactions protéiques.

3.3.3.3 Détection de cliques

La recherche de modules densément connectés dans un graphe peut aussi être exprimée comme une recherche de *quasi-clique*. Notons que la recherche de cliques dans un graphe est un problème récurrent en théorie des graphes. Il s'agit en effet d'un problème sans solution efficace dont le traitement nécessite des ressources plus importantes en fonction de la taille du réseau (problème de type *NP-complet*).

Certains algorithmes partent de la découverte de cliques pour découvrir des régions densément connectées au sein d'un graphe. C'est le cas notamment de CFinder qui crée des communautés de cliques de taille N donnée en reliant toutes cliques de taille N *adjacentes*. Deux cliques de taille N sont qualifiées d'adjacentes si elles partagent au moins $N - 1$ noeuds [123, 124] (Figure 3.17). Les communautés de cliques obtenues au moyen de cet algorithme présentant parfois des tailles trop importantes (surout pour des petits valeurs de N), d'autres auteurs ont eu l'idée de rechercher itérativement des communautés de cliques de taille $N + 1$ au sein de communautés de cliques de taille N jusqu'à l'obtention de communautés dont la taille ne dépasse pas une taille donnée S [125].

3.3.3.4 Algorithmes d'optimisation

L'optimisation consiste à déterminer la meilleure solution parmi l'ensemble des solutions réalisables. En effet, comme nous l'avons mentionné à la section précédente, les algorithmes de recherche de cliques peuvent être assez lents selon la taille du graphe. Pour cette raison, beaucoup d'algorithmes de clustering sont des *heuristiques*, c'est-à-dire des algorithmes qui fournissent rapidement une solution réalisable mais pas nécessairement optimale.

Par exemple, dans le cas d'un clustering, les noeuds sont attribués au hasard à un nombre donné de clusters et la qualité de cette première solution de clustering est estimée à l'aide d'un score. Par la suite, les noeuds sont aléatoirement échangés entre les clusters et les mouvements permettant d'obtenir un meilleur score sont conservés. A

titre d'exemple, l'optimisation peut jouer sur une minimisation du nombre d'arêtes inter-cluster, de la distance entre les noeuds au sein d'un cluster, sur une maximisation du coefficient de clustering au sein des clusters, etc [126, 127, 128].

Ces algorithmes présentent deux défauts majeurs. Tout d'abord, il s'agit d'heuristiques, c'est-à-dire des méthodes approximatives qui pour un même problème de départ retournent des solutions différentes selon le partitionnement de départ et les mouvements effectués par la suite. Par ailleurs, les mouvements aléatoires peuvent réduire l'ensemble des solutions vers une solution suboptimale éloignée de la véritable solution optimale. Pour contourner ces problèmes, il est d'usage d'utiliser plusieurs fois l'algorithme sur le même jeu de données et de choisir la solution optimale parmi les différentes solutions. Par ailleurs, certains mouvements n'optimisant pas le score sont parfois permis afin d'éviter les éventuelles solutions suboptimales.

3.3.3.5 Autres algorithmes

D'autres algorithmes de clustering ont été appliqué en biologie et certaines seront encore discutées dans la partie *Résultats* de cette thèse.

MCL MCL est un algorithme de clustering très populaire en biologie et qui a déjà été appliqué dans de nombreux domaines allant de la biologie (famille de gènes, complexes protéiques, données d'expression) à la reconnaissance d'images. L'idée derrière MCL est assez intuitive. Si l'on se représente les arêtes comme un ensemble de routes entre les noeuds et que du dessus, on observe des personnes marchant aléatoirement sur celles-ci, il est vraisemblable que les personnes présentes dans une région densément connectée ne s'en échapperont que difficilement, il y a donc plus de personnes sur ces chemins. MCL simule un grand nombre de marcheurs aléatoires (c'est à dire des flux de marcheurs), il augmente les flux là où ils sont déjà forts et les diminue lorsqu'ils sont plus faibles par des opérations simples. En répétant ce processus, on obtient finalement des régions où le flux est important séparées par des régions ou le flux est devenu nul [129, 113].

MCODE Cet algorithme courament utilisé pour la découverte de complexes protéiques dans les réseaux d'interaction consiste en plusieurs étapes. Tout d'abord, il calcule un poids pour chaque protéine du réseau. Ce poids est représentatif de la connectivité entre les voisins de ce noeud. Ensuite, lorsque les poids sont calculés, un cluster est composé à partir d'un noeud source présentant le poids le plus élevé. Ce cluster est ensuite complété en traversant le réseau et en y ajoutant les noeuds voisins de poids supérieur à une certaine valeur correspondant à un pourcentage donné de la valeur du noeud source. Cette deuxième étape est répétée en utilisant comme noeud source les noeuds n'appartiennent pas encore à un cluster.

Le résultat de cet algorithme n'est pas un partitionnement. En effet, d'une part tous les noeuds ne sont pas regroupés dans un cluster et d'autre part les noeuds peuvent appartenir à plusieurs clusters ou à aucun [130].

Fuzzy clustering A l'exception de MCODE et de CFinder, la grande majorité des algorithmes de clustering que nous avons décrits jusqu'ici produisent des partitionnements. Cependant, pour certains types de données, l'assignement d'un noeud à un seul cluster peut empêcher de rendre compte des relations multiples entre un noeuds et ses différents voisins. Par exemple, il n'est pas rare qu'une protéine fasse partie de complexes différents.

Le fuzzy clustering consiste, en partant d'une partition, à calculer le degré d'appartenance de chaque noeud à chaque cluster. Ce degré d'appartenance d'un noeud à un cluster se mesure généralement comme la proportions d'arêtes du noeud qui sont atteignent le cluster. Cette approche en deux étapes (partitionnement initial suivi du calcul des degrés d'appartenance des noeuds) a été appliquée récemment au laboratoire pour étudier un graphe où les noeuds représentent les phages et les arêtes pondérées illustrent la similarité entre les phages (nombre d' orthologues communs). Le fuzzy clustering a permis d'établir une classification de phages et de détecter les phages mosaïques résultant de la fusion de plusieurs type de phages [131].

3.4 Outils informatiques pour l'étude de réseaux biologiques

Parallèlement aux études sur les graphes biologiques, une série d'outils permettant leur étude bioinformatique ont été développés. Dans la suite, nous allons énumérer quelques unes de ces ressources parmi les plus populaires ou les

plus utiles. Tous ces outils sont des logiciels gratuits, généralement libres (ou open source, c'est-à-dire donc le code est librement et gratuitement accessible et réutilisable) [132, 46]

D'emblée une séparation peut être effectuée entre les outils accessibles en ligne de commande (mode console) et ceux disponibles à travers une interface graphique. Si la seconde solution est à première vue plus intuitive et permet de ne pas rebuter le biologiste dépourvu de connaissances approfondies en informatique, elle présente néanmoins certains désavantages. Notamment, l'accès aux outils par le biais unique d'une interface graphique rend quasiment impossible toute automatisation de processus, par exemple dans le cas d'une validation où l'on voudrait estimer l'espérance aléatoire de certaines propriétés du réseau en réitérant un grand nombre de fois la même mesure sur un grand nombre de réseaux générés aléatoirement.

3.4.1 Outils avec interface graphique

3.4.1.1 Cytoscape

Cytoscape est la plateforme dédiée aux réseaux biologiques la plus courament utilisée en bioinformatique car elle cumule plusieurs avantages. Ce programme à code ouvert permet notamment la visualisation de réseaux d'interactions ou de chemins biochimiques. En effet, il incluet une série d'algorithmes de *layout* qui permettent disposer les noeuds d'un graphe de façon à faciliter leur visualisation. Il permet aussi de travailler avec des réseaux hétérogènes via l'intégration de différents types de données au réseau (profils d'expression de gène, localisation, etc). Dans sa version de base, Cytoscape est surtout dévolue à la visualisation des réseaux. Cependant, sa grande force réside dans les *plugins* additionnels téléchargeables via son site web notamment. Ces plugins, écrits par de nombreux groupes différents, lui ajoutent de multiples possibilités d'analyse et d'annotation (analyse topologique, clustering, annotation, etc). Ecrit en Java et fonctionnant donc sur toutes les plateformes, il nécessite une installation et est téléchargeable sur *http ://www.cytoscape.org* [133].

3.4.1.2 VisANT

VisANT est une plateforme écrite en Java et dévolue à la visualisation et l'analyse de réseaux biologiques. Comme Cytoscape, il permet de combiner avec le réseau des données d'annotation, directement téléchargeables au travers de son interface. Par ailleurs, VisANT calcule les caractéristiques topologiques globales du réseau (coefficient de clustering, degré, ...). Comme VisANT est un programme très léger, il est accessible directement sans installation (comme applet ou via Java Web Start) ou avec installation préalable à l'adresse suivante : *http ://visant.bu.edu/* [134].

3.4.1.3 yED

yED est outil Java gratuit qui est presque uniquement dédié à la visualisation de réseaux. Les fonctions de *layout* de yED, basées sur le même code que celles de Cytoscape permettent néanmoins la spécification de plus de paramètres et les représentations graphiques sont réalisées rapidement et efficacement. Notons que bien que gratuit, le code de yED n'est pas disponible. Il est téléchargeable à l'adresse *http ://www.yworks.com/en/products_yed_about.html*. Les résultats de cette thèse montrant des graphes ont généralement été générés au moyen de yED et de Cytoscape (dans une moindre mesure).

3.4.1.4 Biolayout Express 3D

BioLayout Express3D est un outil écrit en code ouvert en Java (et donc multi-plateforme) permettant la visualisation et le clustering de graphes. Au départ, ces outil a été implémenté pour visualiser au sein d'un graphe les relation de co-régulation de gènes à partir de données micro-array. Ses principaux avantages sont d'une part l'intégration de l'algorithme de clustering MCL et d'autre part la visualisation de graphes en 3D. Malheureusement, contrairement à ce qui est vanté dans ses caractéristiques, il ne permet pas de traiter des graphes de tailles raisonnables (2000 noeuds et 30000 arêtes) pourtant aisément pris en charge par Cytoscape ou yED. Il est téléchargeable à partir de *http ://www.sanger.ac.uk/Teams/Team101/biolayout/b3d.html* [135].

3.4.1.5 Network Workbench

Network Workbench est un outil puissant écrit en Java. Cet outil semble un peu moins évident à prendre en main que les outils précédemment décrits et de de même, la visualisation n'en constitue pas le point fort. Cependant, il permet le traitement de gros réseaux et le calcul efficace de nombreuses caractéristiques topologiques et surtout la génération de graphes aléatoires selon différents modèles (Erdös-Renyi, scale-free, ...). Par ailleurs, Network Workbench est extrêment bien documenté. Il est téléchargeable à l'adrese *http ://nwb.slis.indiana.edu/.*

3.4.1.6 Pajek

Pajek est un outil rapide et puissant, écrit en Delphi, qui permet une énorme variété de traitements différents. Par ailleurs, il n'est pas axé principalement sur les graphes biologiques et leur visualisation. Un avantage de cet outil par rapport à ceux précédemment évoqués est qu'il peut aussi être utilisé en ligne de commande via l'implémentation de *macros* permettant ainsi son utilisation intensive. Malheureusement, d'une part cet outil est peu intuitif et d'autre part, il n'est prévu pour fonctionner que sous le système d'exploitation Windows (même s'il est assez simple de l'émuler sous un autre système d'exploitation). On peut télécharger Pajek à cette adresse : *http ://pajek.imfm.si/doku.php* [136].

3.4.2 Outils en ligne de commande

On ne compte plus le nombre d'articles qui décrivent un nouvel outil permettant d'extraire de l'information, d'effectuer du clustering de graphes ou encore d'afficher un réseau en mettant en avant certaines qualités et qui fonctionnent en ligne de commande. Dans la site, je décrirai succintement quelques uns des outils génériques d'analyse de réseaux et utilisables en ligne de commande.

3.4.2.1 GraphViz

GraphViz est un ensemble d'outils à code ouvert, créés par les laboratoires de recherche d'AT&T, qui manipulent des graphes définis à l'aide de scripts suivant le langage DOT. Graphviz est principalement utilisé pour la visualisation des graphes. Le principal inconvénient de GraphViz est sa relative lenteur pour l'affichage de gros graphes. On peut le télécharger du site *http ://graphviz.org/.*

3.4.2.2 NeAT

NeAT, développé au sein de notre laboratoire est une suite d'outils utilisables soit en ligne de commande soit au moyen d'une interface web. NeAT est aussi accessible au travers de services web, ce qui signifie qu'il est possible de le faire fonctionner en ligne de commande sur une machine distante du serveur sans installation préalable. Presque totalement écrit en Perl, NeAT permet néanmoins le traitement efficace de graphes de taille assez importante. Une partie entière des résultats de cette thèse sera consacrée à NeAT (*http ://rsat.ulb.ac.be/neat/*) [137].

3.4.2.3 tYNA

Comme NeAT, tYNA est une suite d'outils de traitement de graphes, implémentée en Java et accessible à la fois en ligne de commande, via des services web ou moyen d'une interface web. tYNA et NeAT ont des fonctionalités assez complémentaires. Leur accessibilité via les services web permet donc de combiner plusieurs types d'opérations [138].

3.4.2.4 Les librairies

En informatique, les librairies consistent en un ensemble de fonctions informatiques dont l'utilisation se fait au travers d'un autre programme. Ainsi, un utilisateur peut combiner selon ses besoins les fonctions contenues dans une ou plusieurs librairies. Cependant cette modularité a un prix car leur utilisation nécessite des connaissances en programmation, même si certaines facilités sont parfois prévues. Certaines librairies à code ouvert sont dédiées au traitement efficaces de graphes.

La librairie Boost Graph La librairie Boost Graph consiste en un ensemble de fonctions informatiques écrites en C++ permettant une large variété de traitements sur de gros graphes et ce, très rapidement. Très complète et très efficace, elle est cependant assez difficile à utiliser et à installer. L'utilisation des ressources de la librarie Boost peut se faire de différentes manières, soit en écrivant des programmes compilés en C++, ou au moyen de scripts écrits en différents langages comme Perl ou Python ou encore en utilisant le programme de statistiques R (*http ://www.boost.org/doc/libs/1_35_0/libs/g*

La librairie igraph Cette librairie écrite en C ressemble fort à la librairie Boost et a pour principal avantage d'être plus facile à installer. Cependant, elle ne gère que les graphes non pondérés. Les fonctions implémentées dans igraph sont utilisables au moyen de programmes développés en C, de scripts Python ou en utilisant le programme de statistiques R. (*http ://cneurocvs.rmki.kfki.hu/igraph/*).

La librairie JUNG JUNG est une libraire écrite en Java relativement facile à utiliser. Elle n'est utilisable qu'au moyen de programmes Java et est moins rapide que les librairies précédemment évoquées (*http ://jung.sourceforge.net/*). Elle fournit cependant un grand nombre d'outils permettant le traitement de graphes. Certaines de ses fonctions sont par exemple utilisée par le programme Network workbench, évoqué précédemment pour la visualisation de réseaux.

Chapitre 4

Objectifs

Comme nous l'avons indiqué dans l'Introduction, bien que les protéines associées aux membranes soient d'une importance cruciale pour la cellule, les données les concernant sont fragmentaires, notamment au point de vue de la formation de complexes entre ces protéines.

L'objectif global de notre travail consiste à combler cette lacune et à préciser les interactions entre protéines membranaires chez *S.cerevisiae* et plus précisément, entre les transporteurs membranaires. Cette étude sera basée pour commencer sur les données expérimentales d'interactions entre transporteurs collectées dans le cadre du projet Associoport (voir Introduction).

1. **Les données issues du projet Associoport sont-elles fiables ?**

Premièrement, nous aurons besoin d'évaluer la qualité des données délivrées par le projet Associoport. Pour cela, nous étudierons le comportement des témoins (positifs et négatifs) et le comportement de l'ensemble des tests de croissance. Ensuite, les données seront standardisées et filtrées de façon à ne conserver que celles qui sont hautement significatives.

2. **Quelles informations biologiques pertinentes puis-je retirer de ces interactions ?**

Les interactions filtrées seront étudiées de façon globale en les considérant sous forme d'un réseau d'interactions. Ce réseau sera étudié en lui appliquant plusieurs approches issues de la théorie des graphes pour répondre à différentes questions biologiques. Par exemple, au sein du réseau, les voisins d'un noeud présentent-ils une fonction similaire ? Est-il possible par des méthodes de clustering de détecter au sein du réseau d'interactions des modules de protéines densément connectées et présentant une fonction similaire ? La découverte de groupes de voisins ou de groupes de protéines fortement inter-connectées nous donnerait de précieuses indications quant à l'existence de complexes par exemple ou nous permettrait d'assigner des fonctions aux perméases dont la fonction est encore inconnue (dans le cas où une telle perméase fairait partie d'un groupe homogène de protéines). Nous nous demanderons aussi comment le réseau Associoport se compare-t-il avec d'autres jeux de données de la littérature ?

Pour répondre à ces questions, nous aurons besoin de plusieurs outils. Tout d'abord une base de données consacrées au protéines membranaires et bien annotée. Ainsi, nous allons réimplémenté *YTPdb*, une base de données relationnelle dédiée à l'annotation et à la classification des perméases chez la levure [17]. Ensuite, nous aurons besoin d'outils informatiques rapides consacrés à l'analyse des graphes biologiques. Pour cela, nous utiliserons *NeAT*, une suite d'outils développés en grande partie pendant la thèse et qui nous ont permettront de traiter aisément les nombreux réseaux utilisés dans notre étude. Enfin, pour choisir quelle méthode de clustering appliquer et avec quels paramètres parmi les nombreuses méthodes décrites dans la littérature, nous nous livrerons à une évaluation comparative de plusieurs de ces méthodes.

3. **Est-il possible par des méthodes de génomique comparative de prédire des interactions fonctionnelles entre les perméases ?**

Il est vraisemblable que les résultats que nous obtiendrons à la suite de ces analyses soient mitigés. En effet, il est connu que les expériences à grande échelle donnent des résultats globalement bien moins bon que les approches traditionnelles où une seule interaction est testée à la fois [52, 38, 139]. Nous allons donc appliquer des approches

bioinformatiques de génomique comparative qui ne sont plus basées sur l'expérience et évaluer les performances de ces méthodes. Les deux méthodes que nous allons envisagées sont tout à fait complétaires puisque d'une part, nous la comparaison de profils phylogénétiques et d'autre part nous allons inférer un réseau de co-régulation génétique au moyen d'une nouvelle méthode développée au laboratoire pendant cette thèse. Nous pourrons alors rendre compte des performances de ces méthodes pour la détection d'interactions fonctionnelles entre tous les gènes mais aussi entre les gènes de perméases.

Deuxième partie

Résultats

Chapitre 5

Développement d'outils consacrés à l'analyse de graphes

5.1 Historique des outils

Comme tout ce travail de thèse est presque entièrement basé sur l'analyse, l'inférence, l'évaluation et la caractérisation de réseaux biologiques de tailles importantes, nous avons développé une série d'outils informatiques consacrés à leur analyse. Ces outils sont assez efficaces et génériques (applicables à une grande variété de réseaux biologiques ou non). Ils sont utilisables en ligne de commande afin d'en permettre une utilisation intensive et via une interface web intuitive pour les utilisateurs non informaticiens.

Au début de ma thèse, quelques d'outils permettant la gestion des graphes (notamment la conversion entre les formats) étaient déjà inclus dans *RSAT*, la suite d'outils dédiée à l'analyse des éléments cis-régulateurs développés au laboratoire [140, 141, 142]. Malheureusement, ces quelques outils se sont avérés assez lents au vu des tailles des réseaux que nous avions à traiter. Leur implémentation très conviviale était basée sur un un recours intensif aux objets (chaque noeud, chaque arête constituait un objet), ce que le langage Perl gère difficilement. Nous avons donc ré-implémenté complètement la classe informatique principale de gestion des graphes en utilisant un minimum d'objets et en utilisant des structures de données plus simples (tableaux, tableaux indicés, etc).

Sur base de cette nouvelle classe et des classes de *RSAT* pré-existantes, nous avons ré-écrit certains des programmes de manière plus efficace et développé plusieurs autres nouveaux outils (Table 5.1).

Notre suite ne contenait alors que des outils nécessaires à l'analyse de réseaux (affichage, comparaison de plusieurs réseaux, recherche de chemins, ré-échantillonage aléatoire des arêtes, etc). Nous l'avons ensuite diversifiée en y ajoutant d'autres outils (dont certains existaient déjà au sein de *RSAT*) consacrés à l'analyse des classes et clusters ou en développant des programmes permettant de faire le lien entre classification et réseaux. Dans la suite, nous utiliserons le terme de cluster pour désigner un ensemble d'éléments regroupés par un algorithme (par exemple, un module de protéines densément interconnectées retrouvé au sein d'un graphe d'interactions par un algorithme de clustering sur graphe). Une classe désignera un ensemble d'éléments appartenant à une même catégorie (par exemple une classe fonctionnelles comme les transporteurs d'acides aminés ou les protéines de la membrane vacuolaire). Un ensemble de classes constitue une classification et ensemble de cluster un résultat de clustering.

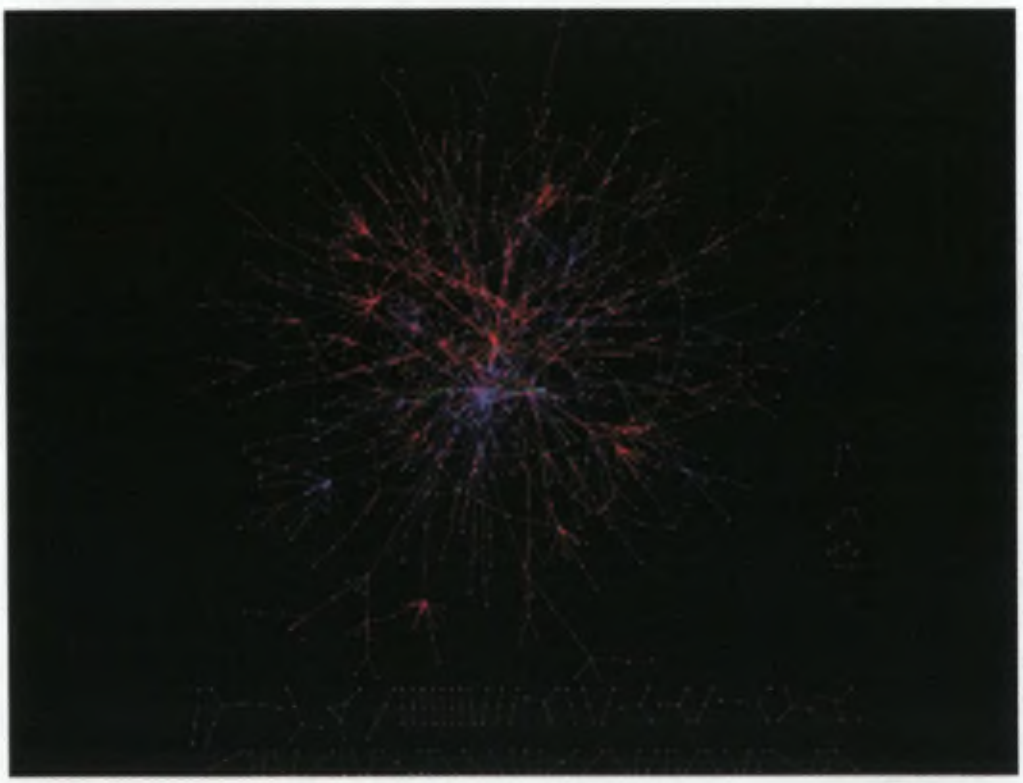
L'ensemble de ces outils, baptisé *NeAT* (pour **Ne**twork **A**nalysis **T**ools) a ensuite été rendue accessible sur Internet ainsi que par services web et nous y avons encore ajouté certains des outils de clustering de graphes évoqués dans notre introduction (RNSC, MCL et fuzzy clustering) ainsi que les outils évalués et développés par Karoline Faust dans le cadre sa propre thèse de doctorat qui porte sur l'inférence de voies métaboliques pertinentes dans un graphe métabolique. Tous les outils présents sur notre site web sont interconnectés de sorte que les résultats produits par un outil peuvent servir de données de départ à un autre outil. En outre, un soin tout particulier a été apporté à la documentation de chacun des programmes de *NeAT* aussi bien pour leur utilisation sur le Web qu'en ligne de commande.

Les outils de *NeAT* seront utilisés tout au long de notre analyse.

ISSN 0305-1048

Nucleic Acids Research

VOLUME 36 WEBSERVER ISSUE JULY 1, 2008
www.nar.oxfordjournals.org



Now Open Access
No barriers to access – all articles freely available online

OXFORD JOURNALS

FIG. 5.1 – Couverture de Nucleic Acids Research Database Issue (Juillet 2008) montrant un des résultats obtenus au moyen du programme *compare-graphs* de *NeAT*. Cette figure montre le graphe résultant de l'union des graphes représentant les deux premières expériences de double hybride à grande échelle chez la levure par Uetz *et al* [31] et Ito *et al* [32]. Les arêtes rouges représentent les interactions détectées par Ito uniquement, les arêtes violettes par Uetz uniquement et les arêtes vertes représentent les interactions communes aux deux études.

Programme	Premier développeur	Ré-implémentation
convert-graph	Jacques van Helden	Oui
display-graph	Sylvain Brohée	
compare-graphs	Gilles Vanderstocken, Jacques van Helden et Sylvain Brohée	
random-graph	Sylvain Brohée	
graph-topology	Jacques van Helden	Oui
alter-graph	Sylvain Brohée	
pathfinder	Jimenez et Marzal [143]	
MCL	Stijn van Dongen [113, 129]	
RNSC	Andrew King [126]	
graph-cliques	Sylvain Brohée	
graph-neighbours	Jacques van Helden	Oui
graph-cluster-membership	Gipsi Lima-Mendez	
graph-get-clusters	Sylvain Brohée	
compare-classes	Jacques van Helden	
contingency-stats	Jacques van Helden	
roc-stats	Rekin's Janky et Jacques van Helden	

TAB. 5.1 – Liste des outils de *NeAT* et de leurs développeurs

5.2 Description des articles présentés

NeAT a fait l'objet de deux publications. La première pour l'édition annuelle *Nucleic Acids Research* consacrée aux serveurs Web et la seconde consiste en un protocole d'utilisation de quelques outils dans la revue *Nature Protocols*.

5.2.1 Présentation du serveur Web de *NeAT*

Sylvain Brohée*, Karoline Faust, Gipsi Lima-Mendez, Olivier Sand, Rekin's Janky, Gilles Vanderstocken, Yves Deville and Jacques van Helden. NeAT : a toolbox for the analysis of biological networks, clusters, classes and pathways. Nucleic Acids Res. 2008 July 1 ; 36(Web Server issue) : W444-W451 [137].

Dans cet article, nous décrivons les outils développés ou intégrés au serveur web de *NeAT*. Nous y distinguons trois classes d'outils :

(*a*) les outils dédiés à l'analyse de réseaux (comparaison, calcul de statistiques topologiques, génération de réseaux aléatoires selon différents modèle, altération du réseau, affichage, recherche de chemins, etc) ;

(*b*) les outils dédiés aux clusters, aux classes fonctionnelles et aux classifications (comparaisons de l'intersection entre deux classifications, et calcul de tables de contingences, découverte de cliques, obtention des voisins de chaque noeud, clustering de graphe, fuzzy clustering) ;

(*c*) les outils faisant le lien entre réseaux et clusters (mapping ou extraction des arêtes intra-clusters)

En outre, nous présentons aussi deux outils supplémentaires. L'un permettant de dessiner des courbes de ROC (courbes courament utilisées dans les évaluations et représentant le taux de vrais positifs (sensibilité) en fonction du taux de faux négatifs) et l'autre permettant de télécharger des sous-réseaux de la base de données intégrative d'interactions fonctionnelles STRING [144].

Dans la suite de l'article, nous décrivons brièvement l'implémentation, la documentation et la disponibilité de nos outils.

W444–W451 *Nucleic Acids Research, 2008, Vol. 36, Web Server issue* *Published online 4 June 2008*
doi:10.1093/nar/gkn336

NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways

Sylvain Brohée[1,*], Karoline Faust[1], Gipsi Lima-Mendez[1], Olivier Sand[1], Rekin's Janky[1], Gilles Vanderstocken[1], Yves Deville[2] and Jacques van Helden[1]

[1]Laboratoire de Bioinformatique des Génomes et Réseaux (BiGRE), Université Libre de Bruxelles (ULB), Boulevard du Triomphe, CP263, B-1050 Bruxelles and [2]Department of Computing Science and Engineering, Université catholique de Louvain (UCL), Place Sainte Barbe, 2. B-1348 Louvain-la-Neuve, Belgium

Received January 31, 2008; Revised April 24, 2008; Accepted May 10, 2008

ABSTRACT

The network analysis tools (NeAT) (http://rsat.ulb.ac.be/neat/) provide a user-friendly web access to a collection of modular tools for the analysis of networks (graphs) and clusters (e.g. microarray clusters, functional classes, etc.). A first set of tools supports basic operations on graphs (comparison between two graphs, neighborhood of a set of input nodes, path finding and graph randomization). Another set of programs makes the connection between networks and clusters (graph-based clustering, cliques discovery and mapping of clusters onto a network). The toolbox also includes programs for detecting significant intersections between clusters/classes (e.g. clusters of co-expression versus functional classes of genes). NeAT are designed to cope with large datasets and provide a flexible toolbox for analyzing biological networks stored in various databases (protein interactions, regulation and metabolism) or obtained from high-throughput experiments (two-hybrid, mass-spectrometry and microarrays). The web interface interconnects the programs in predefined analysis flows, enabling to address a series of questions about networks of interest. Each tool can also be used separately by entering custom data for a specific analysis. NeAT can also be used as web services (SOAP/WSDL interface), in order to design programmatic workflows and integrate them with other available resources.

INTRODUCTION

During the last decade, large-scale biological studies produced huge amounts of data that reveal various layers of molecular interaction networks: protein interactions, transcriptional regulation, metabolic reactions, signal transduction, etc.

Graphs (in the mathematical sense) have been used to represent, study and integrate such biological networks. By definition, a mathematical graph is a set of nodes (generally represented as dots) that are connected by edges (lines between dots). Edges may be enriched by several features, e.g. a direction (an edge from node *A* to node *B* is distinct from an edge from *B* to *A*), a color, a type and a weight (a value is associated to the edges).

Such edges and nodes provide convenient ways to represent biological features. For example, in a protein–protein interaction network, a node represents a polypeptide and an edge indicates the existence of a physical interaction between two polypeptides (1). A weight can optionally be put on edges to reflect the strength of interactions. In 'compound-centric' metabolic networks, nodes represent metabolites and the directed edges represent the enzymes used to convert a metabolite into another one (2). The metabolic networks may also be represented as bipartite graphs, i.e. a network with two distinct types of nodes (one for compounds and one for reactions), where edges must always link a node of one type to a node of the other type (3,4). Similarly, graphs can be used to represent regulatory relationships (5,6) and transduction pathways (7). Network biology is emerging as a very fertile field, as reflected by the rapidly increasing pace of relevant publications (8,9).

Despite the ever-increasing availability of data that may be represented as networks, large-scale analyses should be considered with caution, for several reasons. Firstly, high-throughput data are notoriously noisy (presence of false positives) and incomplete (10,11). In addition, some interaction networks have been characterized by several independent studies, which are providing complementary subsets of the data. Important efforts will thus be required to extract reliable information from the ever-increasing amount of data.

Specialized tools are required to extract and compare information obtained from multiple data sources, and

*To whom correspondence should be addressed. Tel: +32 02 6505434; Fax: +32 02 6505425; Email: sylvain@scmbb.ulb.ac.be

© 2008 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

apply various statistical parameters treatments to describe and understand networks properties. For this purpose, we developed the *Network Analysis Tools* (*NeAT*), a set of modular software tools supporting a large variety of operations on networks and clusters. The web interface provides a convenient and intuitive access to the tools and allows to thread user-provided data sets through typical analysis work flows, in order to interpret their networks. The NeAT programs may be grouped in three categories: tools for manipulating graphs (graph comparison, randomization, alteration, visualization, etc.), tools for analyzing clusters (or, equivalently, classes) (cluster comparison, etc.) and tools that establish the link between networks and clusters (graph clustering, graph–cluster mapping, etc.).

NeAT DESCRIPTION

Figure 1 and Table 1 present the collection of tools available in NeAT as well as their input and output types. On the website, each tool is accessible via the menu on the left panel of the web page (Figure 3, inset).

As shown in Table 1, NeAT tools can be broadly grouped in three categories: *network tools* perform various operations on one or several graphs, *cluster tools* are mainly dedicated to comparisons between clusters and *network–clusters* tools make the connection between networks and clusters.

We will briefly describe the function of each tool together and discuss some typical application. Further information and examples of utilization can be found in the cited literature.

NETWORK TOOLS

Network topology

Several statistics have been defined to characterize global topological properties of a network. It has been shown that these topological properties distinguish biological networks from random networks. Noticeably, it is often stated that the distribution of degree (the number of edges connected per nodes) follows a power-law distribution (12). The program *graph-topology* computes the degree of each node of a graph, which can then be analyzed either as a full result table or visualized as a XY plot (Figure 2). *Graph-topology* also computes the betweenness (i.e. the proportion of shortest going through a node) and the

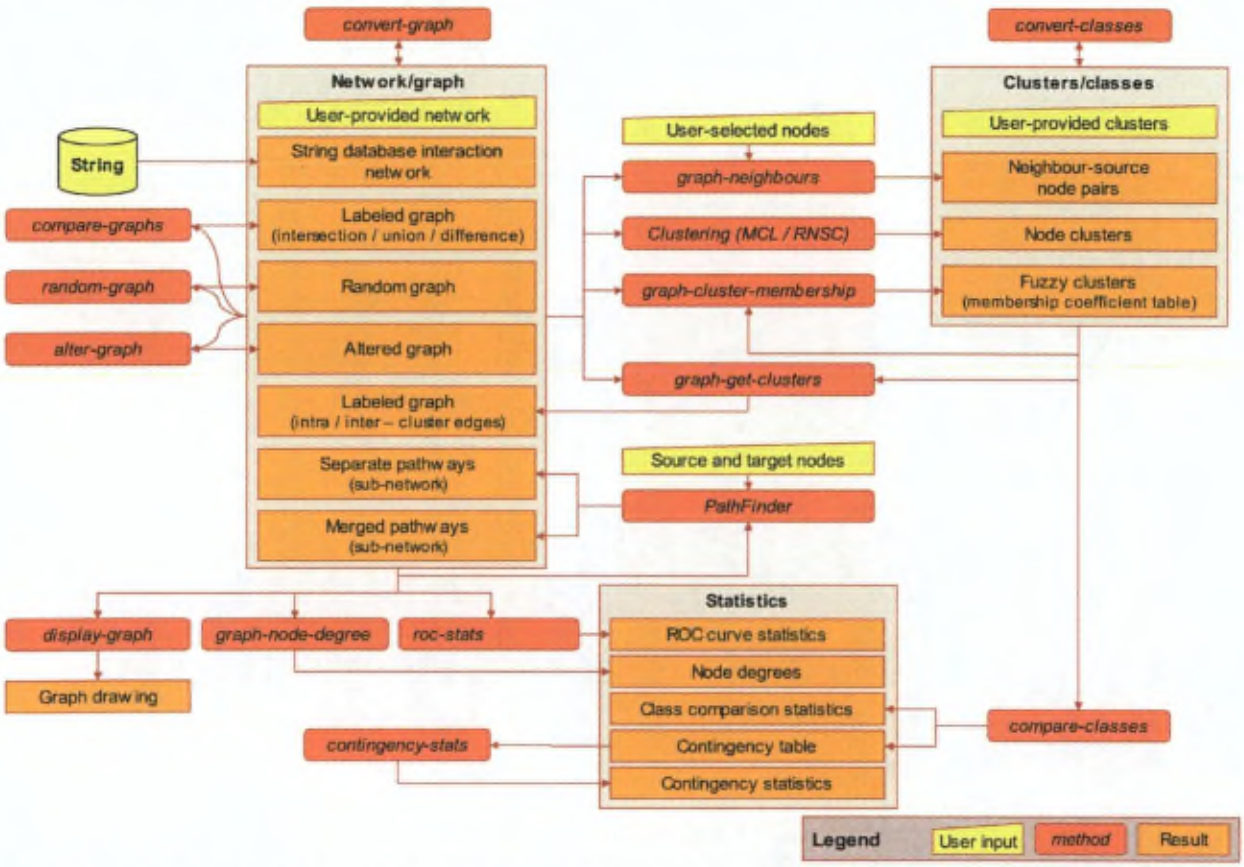


Figure 1. Flow chart of the tools and data types supported on NeAT. Trapezoidal boxes represent user-provided input, rounded boxes programs and rectangles results.

Table 1. Description of the programs available in NeAT

Program	Description	Input	Output
Network tools			
convert-graph	Converts a graph from a format to another one, position the nodes and changes the edge colors and width according to its weight	A network in a given format	A network in the requested format
display-graph	Draws a network graphical representation	A network	A figure in the requested format
compare-graphs	Computes the intersection, the union or the difference of two networks	Two networks to be compared	A network (intersection, union, difference)
random-graph	Generates random graphs either from an existing graph or from scratch according to different randomization procedure.	A graph or a list of node names or nothing	A randomized network
graph-topology	Calculates the degree, betweenness and closeness of each node and specifies if this node is a source or a target node	A network, (list of nodes for which the degree has to be computed)	A table the requested centrality statistics of each requested node
alter-graph	Alters a graph either by adding or removing edges or nodes (targeted removal or not)	A network	An altered network
Pathfinder	Finds the *k*-shortest path between a set of source nodes and a set of target nodes	A network and the list of source and target nodes	A table of pathway or a network composed of the set of pathways
String dataset download	Downloads a subset of the network of the String database (40)	A list of nodes for which you want to know the neighbors in String	The neighbors of the nodes your entered in and the edges between them.
Network clusters tools			
MCL (34,36) *and RNSC* (37)	Finds the densely connected subsets of the graph	A network	A list of clusters
graph-clique	Extract al cliques from a graph	A network	A list of cliques
graph-neighbours	Extracts from a graph the neighborhood of a set of seed nodes	A network, (a list of seed nodes)	Clusters of neighbor-source node pairs
graph-cluster-membership	Maps a clustering result onto a graph and compute the membership degree between each node and each cluster, on the basis of edges linking this node to the cluster	A network, clustering results	A tab-delimited membership table, where each row represents a node and each column a cluster. Entries are the membership degree of the node.
graph-get-clusters	Compares a graphs with clusters. Extracts the intra-clusters edges or map the clusters on the network	A network	An edge-labeled network
Clusters tools			
compare-classes	Compares two class files (the query file and the reference file). Each class of the query file is compared to each class of the reference file.	One or two cluster files	For each comparison, the number of common elements and comparison statistics or a contingency table
contingency-stats	Study of a contingency table	A contingency table	Statistics according to ref. (26)
Others tools			
roc-stats	Calculates and draws ROC curve	Scored results associated with validation labels	For each score value, the derived statistics (Sn, PPV, FPR), which can be further used to draw a ROC curve.

Input Parameters between brackets are optional.

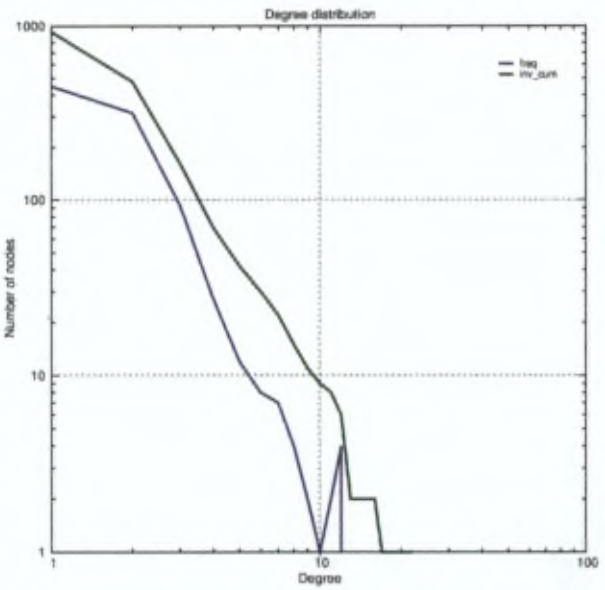


Figure 2. Node degree distribution of a yeast protein interaction network obtained from two-hybrid data. The distribution was computed with the program *graph-topology* and plotted on log scales for both the abscissa and ordinates. The linear shape of the curve on the log–log graph suggests that this network follows a power-law distribution of degree. Color code : blue, absolute frequency; green, reverse cumulative frequency.

closeness (i.e. the mean shortest distance of a node to all others) of each node in the network.

Node neighborhood

Starting from one or several nodes of interest, the program *graph-neighbours* collects neighbor nodes up to a user-specified distance. Neighborhood analysis can be for example applied to predict the function of an unknown polypeptide by collecting its neighbors with known function in a protein interaction network ('guilty by association') (13).

Network comparison

The program *compare-graphs* computes the intersection, the union and/or the difference between two input networks and estimates the statistical significance of the overlap (Figure 3, inset).

These basic operations between graphs can serve for many other tasks: the union can be used to integrate networks at different layers (e.g. metabolism, transduction signal and transcriptional regulation), the intersection to select interactions with evidences in two distinct experiments, the differences to select interactions detected by one method and missed by another one. A typical example of application is to estimate the relevance of a protein–protein interaction network obtained by some high-throughput experiment, by comparing it with a manually curated network [e.g. BioGrid or MIPS databases data (14,15)].

Evaluation of predicted networks using receiver operating characteristic (ROC) curves

The program *roc-stats* is typically used as a postanalysis program after a network comparison between predicted and annotated networks. It takes as input a set of scored results associated with validation status (positives or negatives) and computes, for each threshold on the score, the derived statistics: true positive rate (TPR, also called sensitivity), positive predictive value (PPV), false positive rate (FPR) and accuracy.

Those statistics are also further used to draw different graphical plots showing the performance as a function of the score threshold or allowing performance comparisons (precision recall and ROC curves).

ROC curves show the fraction of the true positives (TPR) versus the fraction of the (FPR) and are often used to compare the predictive performance of different programs (16).

Path finding in a network

Biochemical interactions form intricate networks, where a multitude of pathways can be used to join two nodes of interest. The search of optimal paths (minimizing the number of steps, or the distance, or some weight) has a long tradition in graph theory. Path finding algorithms have been applied to uncover signal transduction pathways from protein–protein interaction networks (17–19) or metabolic pathways in metabolic networks, respectively (20–22). Recently, we evaluated the performance of a k-shortest path finding algorithm for metabolic pathway inference and found that the correspondence between inferred and annotated pathways can be crucially improved by setting an appropriate weighting on the nodes of the metabolic network, in order to penalize highly connected compounds (4).

The NeAT interface includes a general k-shortest path finding algorithm, that supports searches from a set of (one or several) source nodes to a set of target nodes (23). Node weights can either be specified in the input graph or computed automatically according to node degree (4).

Network randomization and alteration

Random graphs are extremely useful to analyze statistical properties of graphs and to validate theoretical models (24–27). The significance of some properties observed in a biological network (e.g. node degree, clustering coefficient, network diameter, etc.) can be estimated by measuring the distribution of probability of the same property in a large set of random networks. Random networks can also be used to observe the behavior of a given algorithm (e.g. clustering) in absence of biological information.

The program *random-graph* supports different procedures to randomize a network, which can then be submitted to the same workflows in the same way as a real biological network. Random graphs can be generated from the scratch, according to an Erdös-Renyi model. Alternatively, random graphs can be generated by permuting the edges between the nodes of a given input graph. This randomization preserves the degree of each

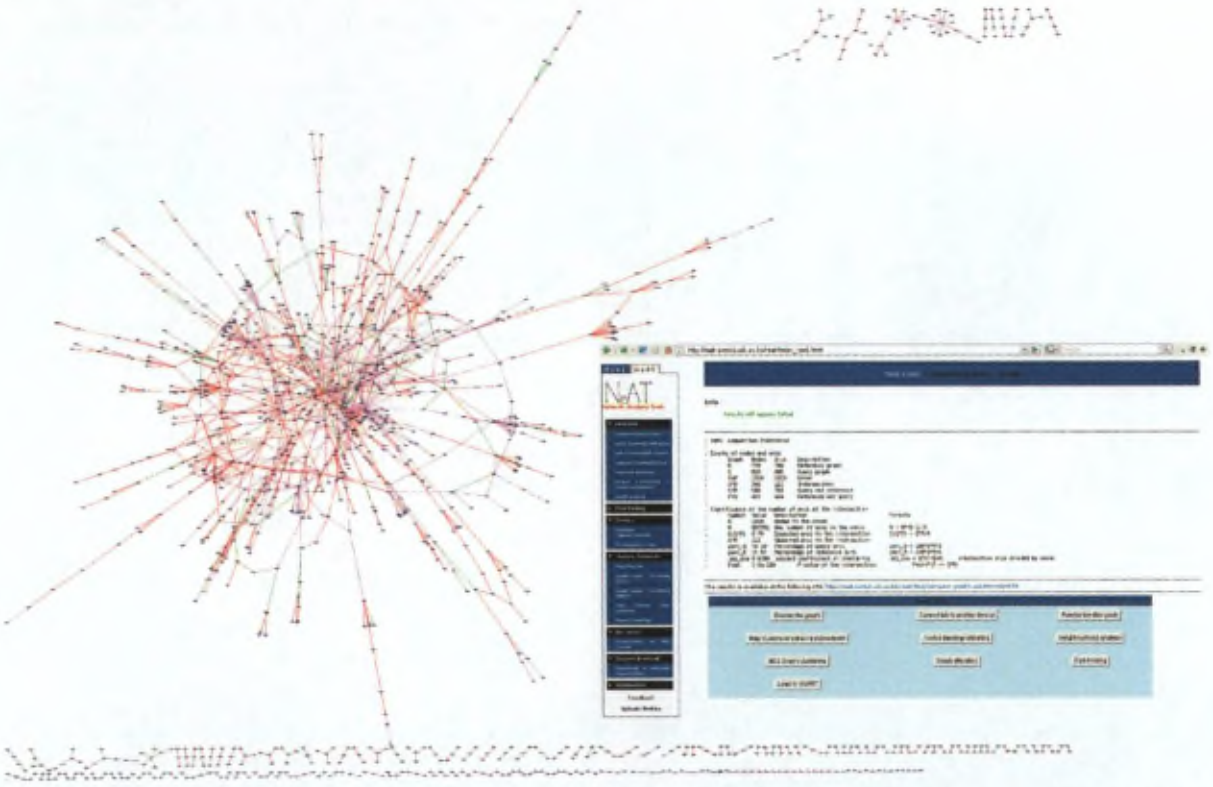


Figure 3. The *compare-graphs* result. Main figure: result of the comparison between two large-scale yeast protein interaction networks obtained by the two-hybrid method (41,42). The networks were compared using *compare-graphs* and displayed with yED. Edge color code: green, edges present in both networks (intersection); red, edges present in Ito's data set only; violet, edges present in Uetz' dataset only. Inset: comparison statistics, including an estimation of the significance of the intersection between the network comparison, based on the hypergeometric distribution.

node. A third mode of randomization preserves the degree distribution of the input graph, without preserving the degree of individual nodes.

Another tool, *alter-graph*, performs a partial randomization of a given input graph, by combining two operations: random addition and/or deletion of nodes and/or edges. Altered graphs are particularly useful to study the robustness of procedures to the presence of noise (node/edge additions) or to missing information (node/edge deletions). This tool was used in our comparative assessment of four graph-based clustering algorithms (26).

Network display

NeAT includes a tool called *display-graph*, which generates static images of an input network. Such drawings are convenient for a quick inspection of the results from the web browser, especially when dealing with large graphs. However, the cost of this speed is that the layout is rather rudimentary and the resulting image is static.

For more sophisticated layouts and for a dynamical manipulation of the drawing, NeAT is also able to load a network directly into the VisANT graph editor via Java Web Start (28).

For more advanced visualization facilities, we recommend specialized graph editors like yED Graph Editor (http://www.yworks.com/en/products_yed_about.html) and Cytoscape [(29), http://www.cytoscape.org]. To this purpose, the tool *convert-graph* permits to export any network resulting from NeAT to the GML format (http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html) which is supported by both editors.

CLUSTER TOOLS

NeAT also presents a series of tools allowing to study clusters or classification (functional classes) (Table 1, clusters tools). For example, the program *compare-classes* can study if among the clusters of highly connected nodes extracted from a graph via some clustering algorithm, some overlap with biological relevant classes [e.g. gene ontology classes (30)] exists. This program also allows to create a contingency table that can be further analyzed via the *contingency-stats* application.

NETWORK–CLUSTER TOOLS

Network clustering

Various algorithms have been implemented to extract clusters (i.e. groups of densely connected nodes) from biological networks. Clustering algorithms are often used

Figure 4. Comparison between a network and a set of classes. Mapping of the yeast protein complexes stored in MIPS database (15) on a large-scale interaction data set obtained by coimmunoprecipitation followed by mass spectrometry experiments (39). The mapping and coloring was performed with *graph-get-clusters*, and the image generated with the graphical editor *yED*. Intercluster edges (edges between nodes that do not belong to the same complex) are displayed in gray. Intracluster edges (edges between nodes belonging to the same complex) are colored with cluster-specific colors (one color for each protein complex).

in biology in order to extract coherent groups of nodes from networks : detection of protein complexes (26,31–33), of protein families (24), extraction of co-expressed clusters from in co-expression networks (35), etc.

The graph-based clustering algorithms MCL (34,36) and RNSC (37) have been shown to obtain good performances for extracting protein complexes from protein interaction networks (26). These algorithms can deal with large graphs and are very efficient in time. For these reasons, we included them in the NeAT tool suite.

Moreover, NeAT also includes a tool that discovers cliques (fully connected set of nodes) in networks.

From partitions to fuzzy clusters

As many clustering algorithms, MCL or RNSC partition the graphs into nonoverlapping clusters: each node is assigned to one and only one cluster. However, in some types of biological data, a single assignment may fail to represent multiple relationships between a node and various types of neighbors (for example, a protein may be part of different complexes).

Some graph-based clustering algorithms support multiple assignment and nonassigned nodes (i.e. fuzzy clustering), but the tuning of their parameters is sometimes delicate and the results can sometimes be weaker than those of a partitioning algorithm.

To keep the best of both worlds, an approach is to first run a partitioning algorithm and to postprocess its result by measuring *a posteriori* the membership between each node and each cluster of the partition. The membership of a node to a cluster is the proportion of edges from this node that reach that cluster. If the graph is weighted, the membership can take edge weights into account.

This two-step approach has been used to perform a reticulate classification of phages and detect mosaic phages resulting from fusions between other phage genomes (38). The program *graph-cluster-membership* takes as input a graph and a clustering result, and returns a node/cluster table indicating the degree of membership of each node to each cluster.

On the NeAT site, clustering results can automatically be launched to the *graph-cluster-membership* form. *graph-cluster-membership* can easily be adapted to be combined with other graph-based clustering algorithms.

Mapping of classes onto network

NeAT program *graph-get-clusters* then allows to extract or to map node clusters onto the network. A first function of such a mapping is to visualize the coherence of protein clusters or functional classes in the context of the network. Figure 4 displays a typical example of *graph-get-clusters* results, where known protein complexes (15) have been mapped onto a yeast protein interaction network obtained by high-throughput co-immunoprecipitation experiments (39). Edges between proteins belonging to annotated structural complexes have been colored according to their cluster (complex) membership. This helps the user to visualize the position of complexes in the interaction network.

DOCUMENTATION

NeAT programs are documented at various levels. Firstly, a manual is accessible from each query form, providing a systematic description of the parameters. Second, DEMO buttons automatically fill the query form with predefined examples (data sets + parameter values), in order to give the intuition of the result returned by the tools on a typical situation. Third, NeAT contains a tutorial, where users can learn using the tools on the basis of concrete biological data sets.

IMPLEMENTATION AND AVAILABILITY

Unless otherwise specified, all interaction data sets available in the NeAT demonstrations and the tutorials were downloaded from the BioGrid database (14) (http://www.thebiogrid.org/).

Moreover, NeAT includes a tool allowing to download and precisely filter subsets of the String database. This database contains protein interaction data obtained by integrating known and predicted interactions from a variety of sources (40).

Except for the path finding and the graph layout algorithms, all NeAT programs were developed in Perl and can be used as stand-alone applications on UNIX-based systems (tested on Linux + Mac OSX). The stand-alone version is freely available for academic users upon request (see *Informations* on the NeAT website).

The large majority of NeAT tools allows the treatment of graphs with several thousands of nodes and several tens of thousands of edges in a reasonable time. Typical published biological networks (a few thousands of nodes and tens of thousands of edges) are treated within seconds. However, some tools may be slower (cliques discovery (NP-hard), betweenness and closeness computation), but the execution time stays reasonable (minutes).

The web site (http://rsat.scmbb.ulb.ac.be/neat/) is free and open to all users and there is no login requirement.

NeAT programs are also accessible as web services (interface SOAP/WSDL), which allows to design programmatic workflows and integrate NeAT tools with various remote resources (databases and software tools). Actually, our website is itself a client for the web services, which guarantees a constant care for maintaining functional web services.

CONCLUSION

The Network Analysis Tools provide bioinformaticians and biologists with a set of web tools that can be combined to efficiently perform the main graph operations (comparison, node degree computation, clustering, etc.) used in today's network biology. As all programs can be integrated in workflows, either on our website or via SOAP web services, users can easily use them to study the topology of a network of interest, discover densely connected groups of nodes, compare these groups to some reference classification and run negative control by submitting randomized graphs to the same analysis.

With the increasing number of studies involving biological networks, we are confident that the NeAT web server will be useful to biologists in general and to network bioinformaticians in particular.

ACKNOWLEDGEMENTS

Pathfinder is a Java wrapper around the REA algorithm written by Jimenez and Marzal in C. K.F. acknowledge these developers for putting their code in public. The NeAT java wrapper was inspired by a Python wrapper written by Pierre Schaus and Jean-Noël Monette at UCL (INGI), who are acknowledged for having helped K.F. K.F. would also like to thank the former aMAZE team for their support and help. The clustering algorithm MCL was developed by Stijn Van Dongen (Sanger Institute, UK). We are thankful to Stijn for making his code publicly available, for allowing us to integrate it in our web server, and for his extensive feed-back on the NeAT web interface. The clustering algorithm RNSC was developed by Andrew King (McGill Institute, Canada). We are thankful to Andrew King for making his code publicly available and for allowing us to integrate it in our web server. We thank Zhenjun Hu for helping us to integrate VisANT in NeAT and Morgane Thomas-Chollier (Université Libre de Bruxelles, Belgium) for her help in the website design and in the implementation of php scripts. We also acknowledge the following colleagues for having tested NeAT and for their constructive comments: Cei Abreu (Sanger Institute, UK), Bruno Contreras Moreira (University of Saragossa, Spain), Alexandre Irrthum (Université de Liège, Belgium), Philippe Gautier (MRC Edimburg, UK), Nicolas Simonis (Harvard Medical School, Boston, USA) and Shuye Pu (Hospital for Sick Children, Toronto, Canada). S.B. and R.J. are recipients of a PhD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA). K.F. is supported by The Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). G.L.M. is a fellow from the Fonds Xenophilia, ULB. The laboratory is a member of the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The BiGRe laboratory is supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet). Funding to pay the Open Access publication charges for this article was provided by the Région Wallonne de Belgique (TransMaze project 415925).

Conflict of interest statement. None declared.

REFERENCES

1. Jeong,H., Mason,S.P., Barabàsi,A.L. and Oltvai,Z.N. (2001) Lethality and centrality in protein networks. *Nature*, **411**, 41–42.
2. Jeong,H., Tombor,B., Albert,R., Oltvai,Z.N. and Barabàsi,A.L. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

3. Gagneur,J., Jackson,D.B. and Casari,G. (2003) Hierarchical analysis of dependency in metabolic networks. *Bioinformatics*, **19**, 1027–1034.
4. Croes,D., Couche,F., Wodak,S.J. and van Helden,J. (2006) Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.*, **356**, 222–236.
5. Luscombe,N.M., Babu,M.M., Yu,H., Snyder,M., Teichmann,S.A. and Gerstein,M. (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, **431**, 308–312.
6. Babu,M.M., Luscombe,N.M., Aravind,L., Gerstein,M. and Teichmann,S.A. (2004) Structure and evolution of transcriptional regulatory networks. *Curr. Opin. Struct. Biol.*, **14**, 283–291.
7. Fukuda,K. and Takagi,T. (2001) Knowledge representation of signal transduction pathways. *Bioinformatics*, **17**, 829–837.
8. Deville,Y., Gilbert,D., van Helden,J. and Wodak,S.J. (2003) An overview of data models for the analysis of biochemical pathways. *Brief Bioinform.*, **4**, 246–259.
9. Huber,W., Carey,V.J., Long,L., Falcon,S. and Gentleman,R. (2007) Graphs in molecular biology. *BMC Bioinform.*, **8 (Suppl 6)**, S8.
10. von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.
11. Sprinzak,E., Sattath,S. and Margalit,H. (2003) How reliable are experimental protein-protein interaction data? *J. Mol. Biol.*, **327**, 919–923.
12. Yook,S.-H., Oltvai,Z.N. and Barabasi,A.-L. (2004) Functional and topological characterization of protein interaction networks. *Proteomics*, **4**, 928–942.
13. Letovsky,S. and Kasif,S. (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19 (Suppl 1)**, i197–i204.
14. Breitkreutz,B.-J., Stark,C., Reguly,T., Boucher,L., Breitkreutz,A., Livstone,M., Oughtred,R., Lackner,D.H., Bähler,J., Wood,V. *et al.* (2008) The biogrid interaction database: 2008 update. *Nucleic Acids Res.*, **36 (Database issue)**, D637–D640.
15. Mewes,H.W., Amid,C., Arnold,R., Frishman,D., Guldener,U., Mannhaupt,G., Munsterkotter,M., Pagel,P., Strack,N., Stumpflen,V. *et al.* (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.*, **32 (Database issue)**, D41–D44.
16. Janky,R. and van Helden,J. (2008) Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution. *BMC Bioinform.*, **9**, 37.
17. Scott,J., Ideker,T., Karp,R.M. and Sharan,R. (2006) Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, **13**, 133–144.
18. Bebek,G. and Yang,J. (2007) Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC Bioinform.*, **8**, 335.
19. Rahman,S.A., Advani,P., Schunk,R., Schrader,R. and Schomburg,D. (2005) Metabolic pathway analysis web service (pathway hunter tool at cubic). *Bioinformatics*, **21**, 1189–1193.
20. van Helden,J., Gilbert,D., Wernisch,L., Schroeder,M. and Wodak,S. (2001) Applications of regulatory sequence analysis and metabolic network analysis to the interpretation of gene expression data. In O.Gascuel and M.-F.Sagot (eds), *Computational Biology : First International Conference on Biology, Informatics, and Mathematics, JOBIM 2000. LNCS Vol. 2066*, Springer, Montpellier, pp. 155–172.
21. van Helden,J., Wernisch,L., Gilbert,D. and Wodak,S.J. (2002) Graph-based analysis of metabolic networks. *Ernst Schering Res. Found.Workshop*, **38**, 245–274.
22. Croes,D., Couche,F., Wodak,S.J. and van Helden,J. (2005) Metabolic pathfinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.*, **33 (Web Server issue)**, W326–W330.
23. Jimenez,V. and Marzal,A. (1999) Computing the k shortest paths: a new algorithm and an experimental comparison. *Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999)*, **1668**, 15–29.
24. Spirin,V. and Mirny,L.A. (2003) Protein complexes and functional modules in molecular networks. *Proc. Natl Acad. Sci. USA*, **100**, 12123–12128.
25. Han,J.-D.J., Dupuy,D., Bertin,N., Cusick,M.E. and Vidal,M. (2005) Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotechnol.*, **23**, 839–844.
26. Brohée,S. and van Helden,J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinform.*, **7**, 488.
27. Milenkovic,T., Lai,J. and Przulj,N. (2008) Graphcrunch: a tool for large network analyses. *BMC Bioinform.*, **9**, 70.
28. Hu,Z., Ng,D.M., Yamada,T., Chen,C., Kawashima,S., Mellor,J., Linghu,B., Kanehisa,M., Stuart,J.M. and DeLisi,C. (2007) Visant 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res.*, **35 (Web Server issue)**, W625–W632.
29. Shannon,P., Markiel,A., Ozier,O., Baliga,N.S., Wang,J.T., Ramage,D., Amin,N., Schwikowski,B. and Ideker,T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
30. Gene Ontology Consortium. (2008) The Gene Ontology project in 2008. *Nucleic Acids Res.*, **36 (Database issue)**, D440–D444.
31. Sharan,R., Ulitsky,I. and Shamir,R. (2007) Network-based prediction of protein function. *Mol. Syst. Biol.*, **3**, 88.
32. Krogan,N.J., Cagney,G., Yu,H., Zhong,G., Guo,X., Ignatchenko,A., Li,J., Pu,S., Datta,N., Tikuisis,A.P. *et al.* (2006) Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, **440**, 637–643.
33. Pereira-Leal,J.B., Enright,A.J. and Ouzounis,C.A. (2004) Detection of functional modules from protein interaction networks. *Proteins*, **54**, 49–57.
34. Enright,A.J., Dongen,S.V. and Ouzounis,C.A. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.
35. Lattimore,B.S., van Dongen,S. and Crabbe,M.J.C. (2005) Genemcl in microarray analysis. *Comput. Biol. Chem.*, **29**, 354–359.
36. Van Dongen,S. (2000) Graph clustering by flow simulation. *Ph.D. Thesis*. Centers for Mathematics and Computer science (CWI), University of Utrecht.
37. King,A.D., Przulj,N. and Jurisica,I. (2004) Protein complex prediction via cost-based clustering. *Bioinformatics*, **20**, 3013–3020.
38. Lima-Mendez,G., van Helden,J., Toussaint,A. and Leplae,R. (2008) Reticulate representation of evolutionary and functional relationships between phage genomes. *Mol. Biol. Evol.*, **25**, 762–777.
39. Gavin,A.-C., Aloy,P., Grandi,P., Krause,R., Boesche,M., Marzioch,M., Rau,C., Jensen,L.J., Bastuck,S., Dümpelfeld,B. *et al.* (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.
40. von Mering,C., Jensen,L.J., Kuhn,M., Chaffron,S., Doerks,T., Krüger,B., Snel,B. and Bork,P. (2007) String 7-recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.*, **35 (Database issue)**, D358–D362.
41. Uetz,P., Giot,L., Cagney,G., Mansfield,T.A., Judson,R.S., Knight,J.R., Lockshon,D., Narayan,V., Srinivasan,M., Pochart,P. *et al.* (2000) A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.
42. Ito,T., Chiba,T., Ozawa,R., Yoshida,M., Hattori,M. and Sakaki,Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA*, **98**, 4569–4574.

5.2.2 Protocole d'utilisation de quelques outils de *NeAT*

Sylvain Brohée*, Karoline Faust*, Gipsi Lima-Mendez, Gilles Vanderstocken and Jacques van Helden. Network Analysis Tools : from biological networks to clusters and pathways. Nature protocols. 2008 September 18 ; 3 : 1616 - 1629. [145].
Dans cet article, nous proposons la marche à suivre sur le serveur web de *NeAT* afin de réaliser quatre manipulations courantes de graphes biologiques.

(*a*) **Clustering de graphe**

Nous décrivons comment appliquer l'algorithme de clustering *MCL* à un graphe d'interactions entre protéines (extrait de la base de données STRING) afin d'en extraire les groupes de noeuds densément interconnectées et correspondant éventuellement à des complexes protéiques. Les clusters retournés par *MCL* sont utilisés comme données d'entrée dans la suite du protocole.

D'abord, nous expliquons comment visualiser ces clusters au sein du graphe (visualisation ou extraction des arêtes intra-clusters) soit au moyen de *NeAT* ou d'autres outils de visualisation. Ensuite, nous montrons comment calculer pour chaque noeud son degré d'appartenance à un cluster (fuzzy clustering). Enfin, pour observer si les complexes retournés par *MCL* correspondent à des complexes biologiques, nous exposons la manière de comparer les clusters de protéines avec un ensemble de complexes annotés pour calculer la significativité des intersections.

(*b*) **Comparaison de réseaux**

La deuxième partie du protocole décrit comment calculer, afficher et estimer la significativité statistique de l'union de deux réseaux. Les réseaux que l'on compare sont le réseau d'interaction déjà utilisé dans la première partie et un réseau de relations de létalité synthétique (chaque gène indépendamment n'est pas nécessaire à la survie de la cellule, mais la combinaison de deux mutations conduit à la mort cellulaire).

Nous montrons que malgré que l'intersection entre les deux réseaux semble très faible, celle-ci se révèle néanmoins significative.

(*c*) **Contrôle négatif**

En bioinformatique, il est fréquent que l'on publie les résultats d'une méthode sans préciser son espérance aléatoire, c'est à dire, les résultats que l'on aurait obtenu si l'on avait appliqué la méthode sur un jeu de données ne contenant pas d'information (témoin négatif).

La partie suivante de notre protocole explique comment réaliser un tel témoin négatif en utilisant *NeAT* pour ré-échantilloner les arêtes du réseau de départ et enjoint les lecteurs à ré-éditer les étapes effectuées dans les étapes précédentes pour vérifier que les résultats obtenus n'étaient pas dûs au hasard.

(*d*) **Recherche de chemin**

La dernière partie du protocole consiste à utiliser le programme de recherche des plus courts chemins installés et évalués par Karoline Faust au sein du graphe d'interaction utilisé dans la première partie du protocole. Le cas d'étude du protocole consiste à retracer une voie de signalisation induisant la division cellulaire entre les protéines Ras2p et Tec1p.

Le protocole contient par ailleurs une description des formats sous lesquels les graphes sont généralement encodés ainsi que des différentes métriques calculées par nos outils pour quantifier la significativité d'une intersection (de réseaux ou de classifications).

PROTOCOL

Network Analysis Tools: from biological networks to clusters and pathways

Sylvain Brohée, Karoline Faust, Gipsi Lima-Mendez, Gilles Vanderstocken & Jacques van Helden

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine, CP 263, Boulevard du Triomphe, B-1050 Bruxelles, Belgium. Correspondence should be addressed to J.v.H. (Jacques.van.Helden@ulb.ac.be).

Published online 18 September 2008; doi:10.1038/nprot.2008.100

Network Analysis Tools (NeAT) is a suite of computer tools that integrate various algorithms for the analysis of biological networks: comparison between graphs, between clusters, or between graphs and clusters; network randomization; analysis of degree distribution; network-based clustering and path finding. The tools are interconnected to enable a stepwise analysis of the network through a complete analytical workflow. In this protocol, we present a typical case of utilization, where the tasks above are combined to decipher a protein–protein interaction network retrieved from the STRING database. The results returned by NeAT are typically subnetworks, networks enriched with additional information (i.e., clusters or paths) or tables displaying statistics. Typical networks comprising several thousands of nodes and arcs can be analyzed within a few minutes. The complete protocol can be read and executed in ~1 h.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

INTRODUCTION

This is the last article in a series of four protocols for the analysis of regulatory sequences with the Regulatory Sequence Analysis Tools[1] (http://rsat.ulb.ac.be/rsat/) and biological networks with the Network Analysis Tools (NeAT)[2] (http://rsat.ulb.ac.be/neat/). The first article[3] presents a protocol to predict the location of binding sites for transcription factors whose specificity is already known (pattern matching). In the second article[4], we describe a protocol for the *ab initio* discovery of biological signals in biological sequences (pattern discovery). The third article[5] shows how to write scripts to automate the analysis on multiple clusters of genes using Web services. In this article, we describe a workflow for deciphering biological networks by combining network comparison, module identification and path finding. This protocol can be executed independently of the three other ones.

Network biology is emerging as a new field in biology, due to the increasing availability of genome-scale data sets of molecular interactions, such as those resulting from high-throughput technologies (e.g., protein interactions, regulatory networks and metabolome). Extracting relevant information from this huge amount of data requires dedicated tools. These data sets are commonly represented as graphs (or networks), where nodes represent molecules, and arcs their interactions. This representation eases data integration and makes it possible to apply well-known network algorithms to analyze the data.

In this protocol, we show how large biological networks can be explored by combining a set of modular tools accessible via a Web interface named NeAT. We describe hereafter the typical questions that can be addressed.

Network topology. It has been observed that some topological properties distinguish biological networks from random networks[6,7]. Noticeably, the distribution of degree (the number of arcs connected per nodes) is often claimed to follow a power-law distribution[6,8]. The tool ***graph-topology*** can be used to analyze the degree distribution of any kind of network.

Network comparison. Given two networks (i.e., protein–protein interaction networks from two different experiments), one would like to analyze their degree of overlap. This question is answered by the tool ***compare-graphs***, which computes the intersection, union and difference between two networks and estimates the statistical significance of the overlap.

Node neighborhood. Given a protein, gene or another node in a biological network, it is of interest to identify its direct or indirect neighbors in this network. This is the task of the tool ***graph-neighbours***, which returns the neighbors of a query node in a network up to a certain distance. This tool can be applied for instance on protein–protein interaction networks to retrieve the interaction partners of a given protein.

Cluster analysis. Various algorithms have been implemented to extract clusters (i.e., groups of densely connected nodes) from biological networks[9–13]. Among those, ***MCL***[14] algorithm has been shown to obtain good performances for extracting protein complexes from protein interaction networks[15–17]. In addition, this algorithm can deal with large graphs and is very efficient in time. For these reasons, we included MCL in the NeAT suite. The clusters resulting from MCL or other methods can be compared to some reference groups (e.g., functional classes) with the program ***compare-classes*** and mapped onto networks with ***graph-get-clusters***. Upon partitioning with MCL, each node belongs to only one cluster. However, sometimes the assignment of a node to a single cluster is an over-simplification of the biological data, for example, a protein may be part of different protein complexes. In those cases, it would be better to describe how much each node is related to the different clusters. The program ***graph-cluster-membership*** postprocesses a clustering result and calculates the membership as the proportion of edges (or weight) linking each node to each cluster. The node–cluster relationships are described as a membership matrix, where each row represents a node and each column a cluster.

Path finding. Given a biological network and two nodes of interest, a common task is to find a biological meaningful path connecting those nodes in the network. For instance, path-finding algorithms are applied to uncover signal transduction or metabolic pathways in protein–protein interaction or metabolic networks, respectively[18–21]. Recently, we evaluated the performance of a *k*-shortest

path-finding algorithm for metabolic pathway inference and found high accuracies if appropriate weights are set on the network[22].

Network randomization. Negative controls are essential to estimate the relevance of the results. The tool ***random-graph*** proposes different procedures to randomize a network, which can then be submitted to the same workflows as the original network.

Network alteration. To test the robustness of analytic methods to the presence of noise, or to the incompleteness of information, the tool ***alter-graph*** allows to modify an existing network by random addition or deletion of nodes and/or edges.

Network display. NeAT includes a tool called ***display-graph***, which generates static images of the input networks. Such drawings are convenient for a quick inspection of the results from the Web browser. For more sophisticated layouts and for a dynamical manipulation of the drawing, we recommend graph editors such as yEd (http://www.yworks.com/en/products_yed_about.html) or Cytoscape[23]. The tool ***convert-graph*** permits to export any network analyzed with NeAT into Graph Modeling Language (GML) (http://www.infosun.fim.uni-passau.de/Graphlet/GML/gml-tr.html), a file format supported by both editors.

Figure 1 depicts the way in which the NeAT can be connected. We suggest the reader to follow this flow chart progressively during the execution of the protocol.

Comparison to other graph analysis tool suites

A large variety of graph analysis tools exist. We may classify them in three categories: (i) libraries that can only be used programmatically, for example, Boost (http://www.boost.org/), igraph (http://cneurocvs.rmki.kfki.hu/igraph/index.html) or JUNG (http://jung.sourceforge.net/); (ii) stand-alone tools with graphical user interface (GUI) (Pajek[24], Network Workbench (http://nwb.slis.indiana.edu), BiologicalNetworks[25], VisANT[26], yEd, Cytoscape, etc.) and (iii) tools with GUI available via the Web such as tYNA (http://tyna.gersteinlab.org/tyna/) or CABiNeT (http://mips.gsf.de/genre/proj/CABiNet/).

Usually, the libraries offer generic graph algorithms, whereas stand-alone or Web-based tool suites are often specialized. For instance, VisANT, Cytoscape (with its plugins) and BiologicalNetworks focus on the analysis and display of biological networks, whereas yEd offers a flexible interface for the display, layout and edition of general-purpose graphs, but is equipped with limited analysis functions. Pajek and Network Workbench are stand-alone tools for generic graph analysis. Cytoscape (with plugins) and BiologicalNetworks allow in addition retrieval and integration of biological networks.

We describe hereafter some of the advantages and current limitations of NeAT.

Main advantages. The main advantages of NeAT are

(1) NeAT supports a variety of modular tools, which can either be used separately, or combined in a workflow. These tools include a number of unique features (fuzzy clustering, Web access to MCL and RNSC, *k*-shortest paths with multiple start and end nodes, statistical comparison of classes and clusters, etc.) that are currently not available in other packages.
(2) The programs are designed to enable treating very large graphs (several thousands of nodes) without excessive cost in memory or time.
(3) Although most of the analyses can also be performed in specialized software packages such as R, the NeAT Web site offers a user-friendly access for biologists who are not familiar with programming languages.
(4) NeAT can be run on command line, either by installing it locally or by calling Web services. This is not the case of the other stand-alone and Web-based tools (a notable exception is Pajek). The programmatic access (either as stand-alone application or as Web services) allows one to automate the executions of workflows for multiple data sets, which would require hundreds or thousands of manual operations with conventional GUIs or on a Web site. To our knowledge, there is only one other network tools suite enabling workflows, namely tYNA. NeAT and tYNA are complementary: NeAT supports path-finding, graph-based clustering (MCL, RNSC and fuzzy clustering), network randomization and cluster comparisons, whereas tYNA includes tools to find motifs in networks. Because both tools support Web services, they can be easily combined in workflows, either by programming client scripts or using GUIs such as Taverna[27].
(5) NeAT may be used for any kind of network, but it was developed with biological networks in mind. The tools have been extensively tested on a variety of biological networks (protein–protein interaction networks[17], evolutionary networks[28] and metabolic networks[21,22]). Extensive evaluation is rarely reported for other biological network tools suites.

Main limitations. NeAT essentially provides facilities for the analysis of networks, clusters and pathways, but is not focused on

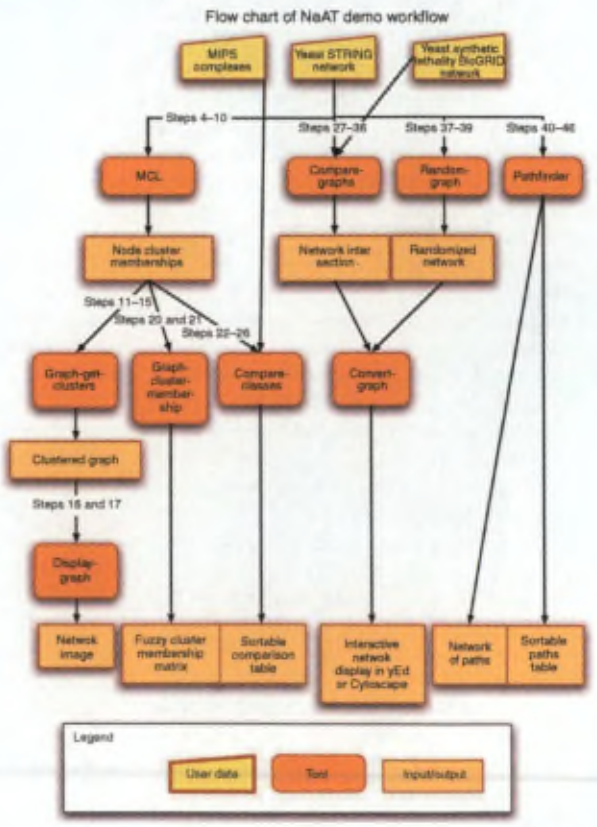


Figure 1 | Flow chart of the data, tools and results described in this protocol. Yellow represents the data set, orange the tool and light brown the results.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

the problem of network visualization. This limitation, however, is easily circumvented by installing some specialized visualization software: all graphs generated by NeAT can be exported to several formats, including GML, which can be loaded with Cytoscape, yEd and VisANT, and DOT, which can be loaded with Graphviz.

To summarize, NeAT addresses the needs of researchers interested in the analysis of biological networks. Some tools may require background knowledge (e.g., MCL, fuzzy clustering), whereas others are intuitive and easy to use (e.g., graph conversion and comparison).

For the user with experience in programming, NeAT can be run on command line or within workflow management environments such as Taverna. Otherwise, the user may access NeAT via its Web interface, guided by tutorials and demos. To our knowledge, no other biological network tools suite exists that combines all the features of NeAT.

Other applications of this protocol

For the sake of consistency, the cases treated in this protocol are restricted to protein interaction networks. The tools available in NeAT can also be used to analyze other types of biological networks representing other types of interactions, for example, regulation, signal transduction, metabolic reactions, and ecology. The fuzzy clustering approach was initially conceived to address the problem of classifying phage genomes while taking into account the frequent exchanges of genetic material between them[28]. The *k*-shortest path-finding algorithm has previously been applied to infer relevant pathways in metabolic networks[29,30].

MATERIALS

EQUIPMENT

- This protocol describes an online tool. The only requirement is a computer with Internet connection. Optionally, you can install yEd (http://www.yworks.com/) or Cytoscape[23] for visualization
- Sample interaction networks, which can be obtained from various biological databases. As examples, we cite the following:
  - STRING[31] (http://string.embl.de/), a database integrating seven different types of evidences for physical and/or functional interactions between proteins: experimental evidences, phylogenetic profiles ('co-occurrence'), gene fusion/fission, synteny ('neighborhood'), coexpression, text mining and a data set called 'database', regrouping several criterion selected by the STRING annotation team
  - BioGRID[32] (http://www.thebiogrid.org/), a database of protein and genetic interactions including >116,000 curated interactions from yeast, *Caenorhabditis elegans*, drosophila and human
  - BioCyc[33] (http://www.biocyc.org/) or KEGG[34] (www.genome.jp/kegg/), the two main metabolic pathway databases
- The data required for the study cases treated in this protocol is available in the data repository site: http://rsat.ulb.ac.be/nedt/. All the networks used to illustrate this protocol were taken from the yeast *Saccharomyces cerevisiae*. We selected various networks representing diverse types of interactions between biological molecules (protein interactions, metabolism, protein complexes, genetic interactions, etc.)
  - Protein–protein interactions (physical and functional). From the STRING database[31], we extracted a subset labeled as 'database' by the STRING team. Under this label, they regrouped different types of protein–protein interactions and metabolic relationships (Jensen L., personal communication). This network contains 1,237 nodes representing proteins, and 11,027 edges representing a mixture of physical and functional protein–protein interactions. The interactions between two proteins are considered symmetrical; it is an undirected graph. The network is stored in data repository, a tab-delimited text file named *yeast_string_database_graph_names_undirected.tab*
  - The ***synthetic lethality network*** was extracted from the BioGRID database[32]. It represents genes (2,353 nodes) whose individual deletion is viable, but whose paired deletions (12,419 edges) are lethal
  - Protein complexes. The file *mips_complexes_names.tab* describes the collection of protein complexes annotated in the MIPS database[35]. Complexes detected only by high-throughput experiments were discarded from the data set. The first column of the file gives the gene name, the second column the complex name and the third column the gene identifier. In total, the file contains 1,121 distinct proteins forming 243 distinct complexes. Note that a protein can belong to several complexes
  - Signal transduction pathway. As study case for the path finding, we take a yeast signal transduction pathway mentioned by Scott and colleagues[18]. This pathway, known to regulate filamentous growth in yeast, starts with RAS2 and ends with TEC1. The authors attempt to recover this pathway with a path-finding algorithm based on color coding[18]. We will try to recover it using *Pathfinder*
- Incompatibility between file formats is a constant problem in bioinformatics. To facilitate the use of the Web site, most tools support several among the most popular formats used to describe networks. A description of the supported format is given in **Box 1**

BOX 1 | MAIN GRAPH FORMATS

The ***tab-delimited format*** is a convenient and intuitive way to encode a graph. Each row represents an edge and each column an attribute of this edge. The two column fields are the source and target nodes. If the graph is directed, the source node is the node from which the arc leaves and the target node is the node to which the arc arrives. Logically, in undirected graph, the columns containing the source and the target nodes may be swapped. Orphan nodes can be included by specifying a source node without target. Some additional edge attributes (weight, label, color) can be placed in additional columns. The tool *Pathfinder* extends this format by supporting any number of attributes on nodes or edges. Check the *Pathfinder* help page for more details.

The ***GML format*** allows for the specification of additional layout and display attribute, such as node position, as well as the color, the label and the width of nodes and edges. A Graph Modeling Language (GML) file is made up of nested key-value pairs. The most popular graph editors (such as Cytoscape and yEd) support GML as input format.

The ***DOT format*** is a plain text graph description language. DOT files can be loaded in the programs of the suite Graphviz (http://www.graphviz.org/). It is a simple way of describing graphs in a human- and computer-readable format. Similar to GML, DOT supports various attributes on nodes (i.e., color, width, label).

Several tools also accept adjacency matrices as input. An ***adjacency matrix*** is an $N \times N$ table (with N the number of nodes), where a cell $A[i,j]$ indicates the weight of the edge between nodes i and j (or 1 if the graph is unweighted).

The Network Analysis Tools program ***convert-graph*** facilitates the handling of these formats by supporting interconversions between various input (tab, gml, adjacency) and output formats (tab, dot, gml, adjacency).

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

PROCEDURE

Downloading a sample network

1| We will show on an example workflow how the different tools of NeAT can be combined to analyze a network taken from the STRING database. Open a connection to the data repository for this protocol (see EQUIPMENT).

2| Download the network file *yeast_string_database_graph_names_undirected.tab* on your computer. It is described in a tab-delimited file that contains five columns. Each row represents one interaction between two genes or between their products. As described in **Box 1**, the two first columns indicate the name of the ***Source*** and ***Target*** genes/proteins of the source and target nodes. The third column contains a score ranging from 0 to 900, which reflects the reliability of the indications available for this interaction. Higher scores represent more reliable interactions. In this case, the score is higher if an interaction is found several times in different data set. The columns 4 and 5 contain the gene identifiers corresponding to the gene names in columns 1 and 2.

3| Open a connection to the NeAT Web server: http://rsat.ulb.ac.be/neat/.

Cluster analysis

4| *Extracting clusters from the network with MCL (Steps 4–10).* We will first apply graph-based clustering to detect groups of highly interconnected nodes in the sample network. For this, we will use the MCL algorithm, a fast unsupervised clustering algorithm based on simulation of flows in graphs[14]. In the menu from the left panel, click on the link ***MCL clustering*** to open the MCL query form.

5| Click on the ***Browse...*** button, and choose the file containing the network (e.g., *yeast_string_database_graph_names_undirected.tab* for the study case discussed here).

6| Specify the columns containing the source, target and (optionally) weight attributes of the tab-delimited file. In our example file, the source and target columns are by default 1 and 2 so we only have to add the weight column: ***Weight column*** = 3. Note that if you want to work with the gene identifiers instead of the gene names, you could have used value 4 and 5 in the source and target column fields. However, this is not recommended in this protocol as in the following we will only work with gene names.
▲ CRITICAL STEP The weighting of edges strongly affects the MCL result, because the principle of the algorithm is to iteratively enforce the weight of the most 'important' edges in the network. The 'importance' of an edge is determined by both its initial weight and its place in the network.

7| Choose an inflation value (between 1.2 and 5). For the study case, select 1.8.
▲ CRITICAL STEP This parameter acts on the granularity of the clustering procedure, that is, the number of clusters (and consequently the number of elements per cluster). The number of clusters increases with the inflation value. This parameter must thus be fine-tuned according to the structure of the network. In a recent evaluation, we found that an inflation value of 1.8 was optimal for protein–protein interaction networks[17].

8| Click on the ***GO*** button. The processing should take a bit less than one 1 min.
? TROUBLESHOOTING

9| The result page displays a figure showing the cluster size distribution, that is, the number of clusters (ordinate) of each size (abscissa). The page also contains a link to the clustering result. This file can be saved by right-clicking on the URL link and selecting ***Save link as...***, and save it on your computer under the name *yeast_string_MCL_clusters.tab*.

10| To inspect the result file, you can either use a text editor to open the file *yeast_string_MCL_clusters.tab* stored on your computer, or click on the URL on the result page. The MCL result is a simple two-column table, where the first column indicates the node names (gene names in our case), and the second column the cluster names. A quick inspection of this table from top to bottom shows that the first clusters contain more nodes than the last ones. MCL sorts the clusters by decreasing order of size.

11| *Extracting the subnetwork defined by the clusters (Steps 11–17).* We will now map the clusters resulting from MCL onto their original network. For this, there are two alternative ways to proceed: directly load the files stored on the server (option A) or transfer the network and cluster files from your computer (option B).

(A) Directly load the files stored on the server

(i) The MCL result page displays a '***Next step***' box, allowing you to send the MCL output to several alternative tools. Click the button ***Map those cluster on the network***. This will call a form *graph-get-clusters*, with prefilled values for the parameters ***Graph*** and ***Clusters***.

(B) Transfer the network and cluster files from your computer

(i) An alternative way to enter data in the tool *graph-get-clusters* is to click on the link '***Map clusters onto network***' in the left menu. This will open an empty form, in which you will have to enter the data (for our study case, the graph is in the file *yeast_string_database_graph_names_undirected.tab* and the clusters in the file *yeast_string_MCL_clusters.tab*). However, this would require to transfer those two files to the server, albeit it already contains a copy of both in the temporary directory. Whenever possible, you should thus use the '***Next step***' buttons rather than transferring the files back and forth between your computer and the server.

12| The main choice for the tool *graph-get-clusters* is the ***output type***. The program supports two types of operations between the network and the clusters. (i) The option ***annotated graph*** labels each edge according to its intracluster or intercluster nature. (ii) ***intracluster edges*** selects a subnetwork restricted to the intracluster edges (intercluster edges are simply deleted from the network). You can experiment the three options. In this section of the protocol, we will extract the subnetwork defined by the MCL clusters. For this, select the option ***intracluster edges***.

13| Several output formats are proposed, but for the visualization purpose, select the intracluster edges output in the ***GML format***.

14| Click on the button ***GO***.

15| The result page should appear after <1 min, displaying a set of buttons for postprocessing the graph-get-cluster result, and a link toward the result file. You can store the resulting GML graph on your computer for later use by right-clicking on the URL in the *graph-get-clusters* result page. Save the result in a file named *yeast_string_MCL_intra_cluster.gml*.

16| A quick way to visualize the result is to fetch it to the NeAT visualization tool. However, beware that this tool offers limited functionalities: it returns a static image, with a simplistic layout. The main function of this tool is to provide a quick view of the result, before visualizing it with specialized tools. To visualize the result network with NeAT, click on the button ***Display the graph***. A new form will then be displayed. Select the desired output format. If the network is weighted (e.g., our study case), you can activate the option ***Edge width proportional to the weight***. To obtain the figure, click on the button ***GO***. This process may be slow (>1 min) depending on the size of the graph.

17| For a better visualization of the network, open the GML formatted file obtained in Step 15 with Cytoscape, yEd or any other visualization program of your choice. After having opened the GML file, you need to apply some layout to display nodes and edges harmoniously. For yEd and Cytoscape we recommend the option ***Organic layout***. After this, you should see a set of well-separated components, each corresponding to a MCL cluster. Each cluster is displayed with a specific color for the edges (**Fig. 2a**).

? TROUBLESHOOTING

18| *Mapping the clusters onto the network (Steps 18 and 19).* In the previous section, we used ***graph-get-clusters*** to separate the MCL clusters by deleting intercluster edges from the original network. Alternatively, the same tool can be used to label all the edges according to the cluster composition. Come back to Step 12, but this time, select '***annotated graph (all edges)***' as ***output type***.

19| Repeat Steps 13–17, and compare visually the result with that obtained in the previous section (**Fig. 2b**).

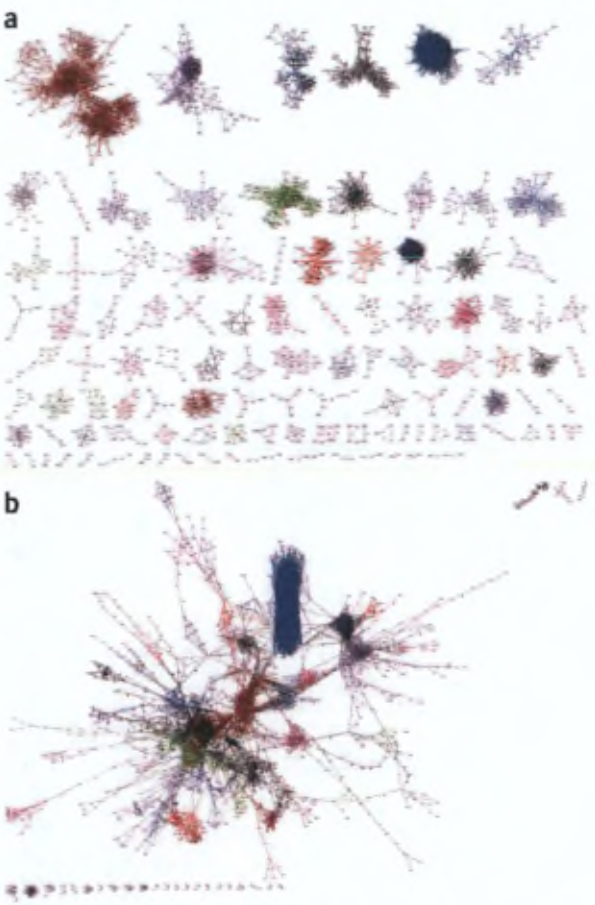


Figure 2 | Mapping of the clusters obtained with the MCL algorithm on the STRING database data set. (**a**) Only the intracluster edges were conserved and each cluster is highlighted with a different color. (**b**) Intercluster (black) and intracluster (colored) edges are both displayed. The layout and display were obtained with yEd.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

Fuzzy clustering

20| We will now use the tool ***graph-cluster-membership*** to compute the degree of membership of each node to each of the cluster obtained from MCL. This can be done in either of two ways. (i) Come back to the page with the MCL output and click on '***Cluster membership***' to open ***graph-cluster-membership***. (ii) Alternatively, you can click on the link ***Cluster membership*** in the left panel, and specify the graph parameters as in Steps 5 and 6. To upload the MCL output, click on the button besides '***Upload clusters from file***': and specify the location of the file *yeast_string_MCL_clusters.tab* on your disk.

21| Search for the membership matrix and select weight as stat.

▲ **CRITICAL STEP** For weighted graphs, weight or relative weight may be chosen. Otherwise, the strength of the links is not considered for calculating the membership of a node to a cluster. When relative weight or relative edge is selected, the weights or number of edges of a node to a cluster are divided by the number of nodes of that cluster.

22| Click on the ***GO*** button. After a minute, a page appears with links to three files: a tab-delimited text file, and two image files providing, respectively, low- and high-resolution heatmaps. In all cases, the output displays the membership matrix, where entries correspond to the membership degree of the node given by the row to the cluster given by the column. The text-formatted table contains the numeric values of the memberships coefficient associating each node (row) to each cluster (column). This is a tab-delimited file that can be loaded in various programs (e.g., Excel, R) for further processing. The heatmap is a graphical representation of the same data, where the gray level is proportional to the degree of membership (**Fig. 3**).

? TROUBLESHOOTING

23| *Comparing the clusters with reference classes (Steps 23–27).* To evaluate the biological relevance of the clusters discovered with MCL, we can compare them with some reference classification, for example the Gene Ontology[36] or the collection of protein complexes from the MIPS database. To illustrate this, we will compare the MCL clusters obtained above with the complexes stored in the MIPS database. Each MCL cluster will be compared to each complex of the database. Cluster/complex comparisons will then be scored with different statistics described in the manual page of the tool and in **Box 2**. Come back to the page with the MCL result (Step 9). On in the ***Next step*** box, click on the button '***Compare these clusters with other clusters***'. Alternatively, in case you saved the MCL result in a file, you can directly click on the link ***Compare clusters/classes*** in the left panel and upload the MCL result file with the ***Browse*** button under ***query class***.

24| As reference classes, we will use the collection of MIPS complexes. For this, first download the file *mips_complexes_names.tab* from the data repository (see EQUIPMENT).

25| We will now specify the ***reference classes*** in the *compare-classes* form. To indicate that MIPS complexes will serve as reference classes, click on the ***Browse...*** button below ***Reference classes***, and select the file ***mips_complexes_names.tab*** that you downloaded on your computer at Step 24. In NeAT, classes are described with the same tab-delimited format as clusters: each row describes the membership of one element (first column) to a class (second column). Optionally, an additional column can be specified with the option '***Score column***', to indicate a score that will be used to compute some similarity metrics (e.g., dot product). For this study case, the MIPS complexes are described in a three-column file indicating the protein name (first column), the complex (second column) and the gene ID (third column). There is no score associated to the protein-complex membership, and we will thus leave empty the option ***score column***.

	cl_1	cl_2	cl_3	cl_4	cl_5	cl_6	cl_7	cl_8	cl_9	cl_10	cl_11	cl_12	cl_13	cl_14	cl_15	cl_16	cl_17	cl_18
EFB1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RPL19B	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
...																		
PDA1	0.00	0.83	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.08
LPD1	0.00	0.74	0.05	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.08
LEU1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PYC1	0.00	0.77	0.10	0.10	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08
RPB9	0.00	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
RNR4	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ENO1	0.00	0.57	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FOL2	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
RPC10	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ENO2	0.00	0.57	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RPB3	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RNR3	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CYR1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RNR2	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RPB4	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RPB34	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HYS2	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
POL32	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 3 | Fuzzy-clusters obtained by combining MCL and the graph-cluster-membership tools. The section of the heatmap shows that several proteins have cluster membership percentages larger than zero for several clusters; for example, ENO1 belongs to both clusters cl_2 (57%) and cl_5 (43%).

26| The options ***Thresholds on return fields*** (see **Box 2**) allow one to combine various constraints to select the most significant intersections between query and reference clusters/classes. The default threshold on significance (sig $\geq$ 0) usually gives satisfactory results. For our study case, the other thresholds do not need to be changed.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

BOX 2 | METRICS FOR COMPARISONS BETWEEN CLASSES OR BETWEEN GRAPHS

In several sections of this protocol, we try to detect significant intersections between two classifications (e.g., MCL clusters, MIPS complexes, etc.) or between two graphs (e.g., interactome). The Network Analysis Tools suite includes specialized programs to compare classes/clusters (*compare-classes*), or graphs (*compare-graphs*), using various comparison statistics.

In both cases (classes or graphs), we can consider that we have a finite set of *N* elements. For *compare-classes*, *N* is the total number of elements that can be a member of any reference or query class (e.g., all the yeast genes). For *compare-graphs*, the *N* elements are all the edges that could possibly be traced between any pair of nodes of the input graph (e.g., all possible intersections between any pair of proteins).

Let us then define

N	the total number of elements in the universe (cluster/class members for *compare-classes*, graph edges for *compare-graphs*);
R	a reference set (one class/cluster, or one graph), containing *Nr* elements;
Q	a query set (one class/cluster, or one graph), containing *Nq* elements;
C	the intersection between a query and the reference set;
Nc	the number of elements in this intersection.

Maximal number of edges in a graph

The maximal number of arcs between a set of *X* nodes depends on whether this graph is directed or not and on whether it does or does not admit self-loops. We can easily compute the value in the four possible cases.

Directed	Self-loops	Number of edges
Yes	Yes	$N = X^2$
Yes	No	$N = X^2 - X = X(X-1)$
No	No	$N = \frac{X(X-1)}{2}$
No	Yes	$N = \frac{X + X(X-1)}{2} = \frac{X(X+1)}{2}$

The column "Number of edges" corresponds to the *N* used for the statistics on graph comparisons.

Jaccard coefficient

The Jaccard coefficient is defined as the ratio between the intersection and the union between two sets.

$$J = \frac{R \cap Q}{R \cup Q} = \frac{Nc}{Nr + Nq - Nc}.$$

The advantage of the Jaccard coefficient is that it gives us an intuitive perception about the mutual coverage of the query and the reference. However, it presents the weakness to be independent of the absolute sizes of the union and intersection. For example, an intersection of 1 element between a set of 3 and a set of 2 elements will give the same Jaccard coefficient as an intersection of 100 between a set of 300 and a set of 100 elements, whereas the random expectation for these two events is very different. A more reliable statistics is the hypergeometric coefficient, as discussed below.

The hypergeometric probability

The hypergeometric distribution is often used to estimate the significance of the intersection between two random selections in a set. The classical example of application of the hypergeometric distribution is the random selection without replacement in an urn containing a set of white and black balls.

The reference set (classes or graph) can be assimilated to the black balls of the urn example. The query set corresponds to the selection without replacement (indeed, a member cannot appear several times in the same class and an edge cannot appear several times in the same graph). The hypergeometric *P* value indicates the probability to observe by chance at least *x* elements at the intersection between the query set and the reference set.

$$P_{val} = P(X \geq Nc) = \sum_{i=Nc}^{Nq} \frac{C_{Nr}^{i} C_{N-Nr}^{Nq-i}}{C_{N}^{Nq}}.$$

The *P* value can be interpreted as the probability for one comparison to return a false positive.

In the case of *compare-classes*, an important correction has to be done for multitesting. Indeed, each class of the query set (e.g., MCL clusters) will be compared to each class of the reference set (e.g., MIPS complexes). The number of comparisons is thus the product between the number of classes in the query set, and in the reference set, respectively. Thus, the nominal *P* value can be misleading because even a low *P* value is expected to emerge by chance when the number of comparisons is very high. A classical correction for this multitesting is to compute the E value.

$E_{val} = T \cdot P_{val}$

where *T* is the number of tests. The E value represents the number of false positives to be expected in a battery of *t*-tests.

To give a realistic order of magnitude, in our study case, we compared 243 clusters obtained from MCL with 107 complexes annotated in the MIPS. The number of comparisons is thus $T = 243 \times 107 = 26{,}001$. Consequently, with an upper threshold of 1% on the *P* value, we would expect at least 260 false positives!

It is also usual to perform a minus log conversion of the E value, which gives the 'significance score'.

$sig = -\log_{10}(E_{val})$.

The *sig* score gives an intuitive perception of the exceptionality of the intersection between sets: a negative significance indicates that an intersection of at least that size is likely to occur by chance, a positive value means that it is significant.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

27| Click on the ***GO*** button. After a few seconds, a result page appears with links pointing toward two alternative output formats: tab-delimited text file or hypertext markup language (HTML) page (**Fig. 4**). The tab-delimited text file can be downloaded to your computer and then imported to various applications for further analysis. The HTML format is useful for inspecting and handling the result on the Web browser. The NeAT HTML tables support dynamic reordering of the rows according to the values of any column, by clicking on the header of this column. The first line of the result file indicates the parameters used for the analysis and documents the content of the columns. This is followed by a result table, where each line reports the comparison between one MCL cluster and one complex.

Network comparison

28| In this section, we will use the tool ***compare-graphs*** to compare the interactions annotated for the yeast *Saccharomyces cerevisiae* in the STRING database with the synthetic lethality relationships obtained from the BioGRID database. Download on your computer the file *Saccharomyces_cerevisiae_biogrid_synthetic_lethality_names.tab* from the data repository (see EQUIPMENT).

29| In the NeAT menu of the left panel, click on the link ***Network comparison***. For our study case, select as ***Query graph*** the previously downloaded file *yeast_string_database_graph_names_undirected.tab*, and as ***Reference graph*** the file *Saccharomyces_cerevisiae_biogrid_synthetic_lethality_names.tab*, by clicking on the corresponding ***Browse...*** buttons. For each file, you need to specify the columns containing source and target nodes, respectively. In both files, the first column contains the source and second column the target. We thus just have to fill the weight column for the query graph (weight = 3) as done previously. The reference network (synthetic lethality) does not contain edge weights.

30| Since in our example, only the edges of the STRING network are weighted, select ***weight/label of the query*** for the option ***Weight/label on the edges of the output graph***.

```
; compare-classes -v 1 -r /home/rsat/rsa-tools/public_html/tmp/compare-ref-classes.4BlvXAMVXb -q /home/rsat/rsa-tools/public_html/tmp/compare-query-classes.rw4INKwcRT -return rank,
  ncc,proba -lth Q 1 -lth R 1 -lth QR 1 -lth sig 0 -sort sig
; Input files
;                                query_classes          /home/rsat/rsa-tools/public_html/tmp/compare-query-classes.rw4INKwcRT
;                                ref_classes            /home/rsat/rsa-tools/public_html/tmp/compare-ref-classes.4BlvXAMVXb
; Query classes (nq)                                 136
; Reference classes (nr)                             241
; Elements  in ref classes (nr)                     1121
; Elements in query classes (nq)                    1240
; Elements in query+ref classes                     1861
; Population size                                   1861
; Comparisons (nr*nq)                              25736
; Multi-testing correction (nc)                    25758
; Sort Key                       sig
; Thresholds                     lower                 upper
;                                Q                             1 NA
;                                QR                            1 NA
;                                R                             1 NA
;                                sig                           0 NA
; Colums contents
;                                1                     rank    Rank of the comparison
;                                2                     ref     Name of the first class (called class Q hereafter)
;                                3                     query   Name of the second class (called class R hereafter)
;                                4                     R       Number of elements in class R
;                                5                     Q       Number of elements in class Q
;                                6                     QR      Number of elements found in the intersection between classes R and Q
;                                7                     QvR     Number of elements found in the union of classes R and Q. This is R or Q.
;                                8                     RiQ     Number of elements found in class R but not class Q
;                                9                     QiR     Number of elements found in the class Q but not in class R
;                                10                    niQiR   Number of elements of the population (P) found neither in class Q nor in the class R
;                                11                    P_val   P-value of the intersection, calculated witht he hypergeometric function. Pval = P(X >= QR).
;                                12                    E_val   E-value of the intersection. E_val = P_val * nb_tests
;                                13                    sig     Significance of the intersection. sig = -log10(E_val)
```

#rank	ref	query	R	Q	QR	QvR	RiQ	QiR	niQiR	P_val	E_val	sig
1	Cytoplasmic-ribosomes	cl_1	158	151	123	166	15	28	1695	4.00E-144	1.09E-141	140.99
2	cytoplasmic-ribosomal-large-s	cl_1	81	151	74	158	7	77	1703	7.30E-81	1.98E-78	75.73
3	26S-proteasome	cl_8	36	36	32	40	4	4	1821	2.80E-60	7.18E-58	55.15
4	cytoplasmic-ribosomal-small-s	cl_1	57	151	49	159	8	102	1702	1.10E-48	2.88E-46	42.56
5	19-22S-regulator	cl_8	16	36	16	36	0	16	1825	8.80E-34	2.30E-29	28.64
6	Pre-replication-complex	cl_17	14	16	14	16	0	2	1845	1.80E-33	4.70E-29	28.33
7	Replication-complexes	cl_17	49	16	16	49	33	0	1813	3.60E-27	9.30E-23	22.03
8	Replication-complex	cl_17	19	16	13	22	6	3	1839	3.00E-26	7.80E-22	21.11
9	H+-transporting-ATPase-vacuol	cl_9	15	34	14	35	1	20	1826	3.20E-25	8.20E-21	20.09
10	20S-proteasome	cl_8	15	36	14	37	1	22	1824	9.60E-25	2.20E-20	19.65
11	F0-F1-ATP-synthase	cl_9	15	34	13	36	2	21	1825	1.90E-22	5.00E-18	17.3
12	Cytochrome-bc1-complex	cl_18	9	7	7	9	2	0	1852	2.40E-18	6.10E-14	13.21
13	Oligosaccharyltransferase	cl_21	9	13	8	14	1	5	1847	3.30E-19	8.50E-14	13.07
14	Replication-initiation-comple	cl_17	8	16	8	16	0	8	1845	3.70E-19	9.40E-14	13.03
15	Replication-fork-complexes	cl_10	30	27	11	44	17	14	1817	4.30E-19	1.10E-13	12.95
16	Anaphase-promoting-complex	cl_4	11	58	11	58	0	47	1803	1.00E-17	2.60E-13	12.58
17	Cytochrome-c-oxidase	cl_27	6	10	7	11	1	3	1850	4.30E-17	1.60E-12	11.79
18	RNA-polymerase-I	cl_2	14	149	14	149	0	135	1712	2.50E-16	6.40E-12	11.19
19	Ddc2Mp-complexes	cl_4	10	58	10	58	0	48	1803	3.90E-16	1.00E-11	11

Figure 4 | Most significant associations between MCL clusters versus MIPS complexes. This figure shows only the top of the table returned by the program *compare-classes*. Each row represents the comparison between one complex (reference) and one MCL cluster (query).

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

31| Several alternatives are possible for the option ***Output type***, corresponding to various combinations of the query and reference graphs (union and difference). The arcs of the resulting graph will be labeled and colored differently depending on whether they belong to the query graph only, the reference graph only or to their intersection. For the study case, to only return the arcs that are in common to both graphs, select ***Intersection*** as ***Output type***.

32| If you want to visualize the resulting network with yEd or Cytoscape, select ***GML format*** as ***Output format***.

33| In case your graph is directed, check the option ***Graphs must be considered as directed***, so that an edge from node A to node B is considered as distinct from an edge from B to A. In our study case, protein interactions are undirected, so this option must remain unchecked.

34| Finally, you can indicate whether or not your graph may admit ***self-loops*** (edges having the same node as source and target). In our study case (synthetic lethality versus STRING interactions), the graph is undirected and has no self-loop.
▲ **CRITICAL STEP** The intersection statistics will be strongly affected by the nature of the graph (directed or not, with or without self-loops), as described in **Box 2**.

35| When this form is filled, click on the ***GO*** button. The computation of the comparison may take some time (between 10 s and a few minutes) depending on the size of the input networks.

36| The result page (**Fig. 5a**) shows statistics about the sizes of the input graphs, their union, intersection and differences (see **Box 2**) and a link pointing to a separate file corresponding to the comparison network. To save this network on your computer, right click on its URL and select ***Save link as***.... The resulting network can be visualized as described above (**Fig. 5b**).

37| The '***Next step***' box permits to use the network resulting from *compare-graphs* as input for some other NeAT programs (clustering, display, randomization, etc.).

Negative controls

38| To check that the results described previously were not obtained by chance only, we can run random negative controls by applying the processes described previously to random graphs. The program ***random-graph*** can be used to generate random graphs according to various random models. Click on the link ***Network randomization*** in the left menu. Upload a graph (e.g., *yeast_string_database_graph_names_undirected.tab*) and select the output format of your choice.
▲ **CRITICAL STEP** The most important parameter is the choice of the type of randomization. In general, we would recommend to select the option ***node degree conservation*** that consists in shuffling the edges, each node keeping the same number of neighbors as in the original graph. This procedure is specially designed to avoid duplicating edges, unless you check the option ***'Allow duplicated edges'*** (this should usually not be done). Another randomization type is the ***node degree distribution conservation*** where the global distribution of the node degree is conserved but each node presents a different degree than in the original graph. Finally, the program also supports ***Erdös-Renyi randomization***, where edges are distributed between pairs of nodes with equal probability.

39| To obtain the randomized network, click on the ***GO*** button.

40| You can now apply to this randomized network all the steps described in the previous paragraphs (clustering, subnetwork extraction, comparison with reference graph, etc.). In principle, the results obtained with the randomized graph should be clearly less convincing than those obtained with the real STRING interaction network.

Path finding

41| Given an interaction network (e.g., the STRING database network) and two query proteins, we can ask which intermediate proteins connect them. This question can be answered using *Pathfinder*, a tool that retrieves the *k*-shortest paths in a network for given source and target nodes (see **Box 3** for more information on *k*-shortest paths finding). The STRING network with converted weights is available in the data repository (see EQUIPMENT), in the file *string_database_graph_converted_weights.tab*. Download this file to your computer.

42| In the NeAT main menu, click on the menu ***Path finding***, then on ***k shortest path finding*** to open the *Pathfinder* query form. Upload your network by clicking on the ***Browse...*** button in the section ***Network***. Alternatively, you can copy–paste the network into the text field. For the case you would like to store this network on the server for later use, click the

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

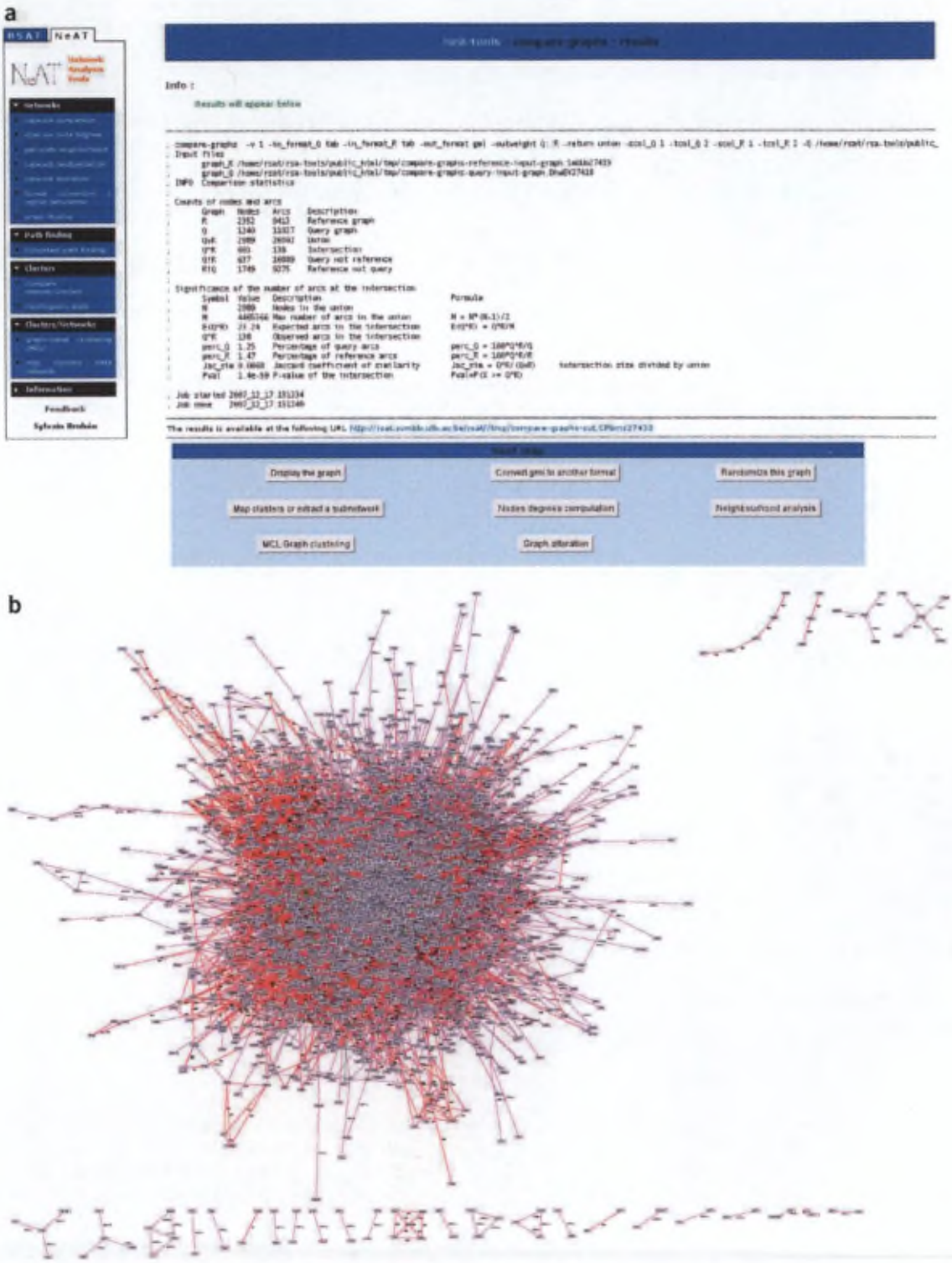


Figure 5 | Result of the fusion between the BioGRID synthetic lethality data set (reference graph) and the yeast–protein interaction data set annotated in the STRING database (query graph). (**a**) Comparison statistics (see **Box 2**). (**b**) Drawing of the union graph (with yEd). Color code: red edges, false positives (edges found in the query graph but not in the reference graph); blue edges, false negatives (edges found in the reference graph but not in the query); green edges, true positives (edges present in both networks, in this case, only 138 among the ~20,000).

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

BOX 3 | *k*-SHORTEST PATHS FINDING IN WEIGHTED NETWORKS

Path finding attempts to find the shortest path between a given start node and a given end node in a network (graph). If several such paths exist, they should be all returned as equally valid solutions. Sometimes, we are not only interested in the shortest path, but also in the second shortest, third shortest or, in general, the *k*-shortest paths. The task of a *k*-shortest paths algorithm is to enumerate all paths up to the requested rank (*k*) in the order of their length. This is accomplished for example by the recursive enumeration algorithm[37] or by Eppstein's algorithm[38]. Often, the edges in biological networks are not equally relevant. For example, experimentally validated protein–protein interactions are more trustable than those observed in only one high-throughput experiment. To express such differential reliabilities, a higher *cost* (*weight*) can be placed on edges representing less trustable protein–protein interactions. When costs, or weights, have been set on the nodes or edges of a network, we no longer search for the *shortest* but for the *lightest* (that is less costly) path. Consequently, the *k*-shortest paths algorithm returns paths ranked according to their *weight* with the lightest path on top.

The weights have to be selected in a relevant way for the biological network of interest. The choice of a relevant weighting criterion clearly depends on your experience about this network and about the quality of the data available.

In a previous study[21,22], we evaluated the accuracy of *k*-shortest path finding for inferring metabolic pathways from compound/reaction networks, and showed that a graph where each node is weighted according to its degree (number of incoming + outgoing edges) achieves an accuracy of 83%.

In our study case with the yeast interaction network, we will use the scores provided by STRING as weights. In this case, the score assigned to an edge is a measurement for the reliability of the protein–protein interaction represented by this edge. In contrast, for Pathfinder, an edge weight is the cost of this edge. Therefore, we converted the scores into costs using the following formula:

$$W_e = \frac{1{,}000}{S_e},$$

where S_e is the score of an edge as defined in STRING (from 0 to 1,000), and W_e is the weight assigned to that edge for path finding.

Store network on server check box. This will allow you to perform further analyses on the same network, without having to transfer it repeatedly from your computer to the server.

43| Enter the IDs of the source and target nodes. For this study case, type RAS2 in the ***Source nodes*** field, and *TEC1* into the ***Target nodes*** field.

44| For the option ***Weighting scheme***, select *'as given in input graph'*. This will specify that weights should be taken from the third column of the input file and not calculated according to a predefined weighting scheme.

45| The result can be exported in various formats, depending on what you want to do with the resulting paths. (i) If you want to display the path in a tabular format, select ***Table*** as ***Output format***. (ii) The result can also be exported to GML format, to visualize the resulting paths as a subset of the original network (this can be done with visualization Cytoscape or yEd). For this, select ***Graph*** as ***Output format***, set the ***Graph format*** to ***GML format*** and set the ***Graph output type*** to ***paths unified into one graph***.

46| Click ***GO*** to start the computation.

? TROUBLESHOOTING

47| The result will be displayed according to the option chosen at Step 45. (i) Output format 'Table'. After a few seconds (or minutes, depending on the size of your graph), the results should appear in the form of two links. The first link points to the table of paths in simple text format, the second to the same table in HTML format (**Fig. 6**). If the checkbox ***Store network on server*** has been clicked, Pathfinder returns the identifier of the submitted network in addition. Submitting this identifier instead of the network itself speeds up the next path-finding job performed on it, because the previously transferred network is used, thereby avoiding to upload it again. (ii) The result form will contain a link to the resulting network in a GML file, which can be downloaded on your computer and displayed with Cytoscape or yEd. This link is followed by a ***'Next steps'*** box, which will permit you to fetch the result network into another NeAT tool.

● **TIMING**

The timings listed below depend on the server load (the number of jobs currently running on the server). However, for the study case we expect the tools to finish within 5 min.

Compare graph: <30 s; MCL: <20 s; graph-get-clusters: <40 s; display-graph: <1 min; compare-classes: <20 s; Pathfinder: <1 min; Fuzzy clustering: <2 min

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 1**.

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

```
; Experiment exp_0
; Pathfinding results
; Date=Thu Jun 26 16:03:53 CEST 2008
; ================================
; INPUT
; Source=RAS2
; Target=TEC1
; Graph=Pathfinder_tmpGraph_d597cd86-3095-475f-99e7-d70a542d072a.tab
; Undirected=true
; Metabolic standard format=false
; REA format=false
; Temporary directory=Temp
; CONFIGURATION
; Algorithm=rea
; Weight Policy=
; Weights given on arcs=true
; Maximal weight=1000000
; Maximal length=1000000
; Minimal length=0
; Exclusion attribute=ReferencedObject.PublicId
; Rank=5
; REA timeout=10
; EXPLANATION OF COLUMNS
; Start node=given start node identifier
; End node=given end node identifier
; K=path index
; Rank=rank of path (paths having same distance have the same rank, though their step number might differ)
; Distance=weight of path (sum of edge weights)
; Steps=number of nodes in path
; Path=sequence of nodes from start to end node that forms the path
; ================================
```

#start node	end node	path index	rank	distance	steps	path
RAS2	TEC1	1	1	6.25	6	RAS2->CDC42->STE20->FUS3->STE12->TEC1
RAS2	TEC1	2	2	7.5	7	RAS2->CDC42->STE20->STE11->FUS3->STE12->TEC1
RAS2	TEC1	3	2	7.5	7	RAS2->CDC42->SHO1->STE20->FUS3->STE12->TEC1
RAS2	TEC1	4	2	7.5	7	RAS2->CDC42->STE20->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	5	2	7.5	7	RAS2->CDC42->BEM1->STE20->FUS3->STE12->TEC1
RAS2	TEC1	6	2	7.5	7	RAS2->CDC42->STE20->STE5->FUS3->STE12->TEC1
RAS2	TEC1	7	2	7.5	7	RAS2->CDC42->STE20->FUS3->DIG2->STE12->TEC1
RAS2	TEC1	8	2	7.5	7	RAS2->CDC42->STE20->STE7->FUS3->STE12->TEC1
RAS2	TEC1	9	9	8.75	8	RAS2->CDC42->STE20->STE5->STE7->FUS3->STE12->TEC1
RAS2	TEC1	10	9	8.75	8	RAS2->CDC42->BEM1->STE20->STE11->FUS3->STE12->TEC1
RAS2	TEC1	11	9	8.75	8	RAS2->CDC42->STE20->STE7->STE11->FUS3->STE12->TEC1
RAS2	TEC1	12	9	8.75	8	RAS2->CDC42->STE20->STE5->STE11->FUS3->STE12->TEC1
RAS2	TEC1	13	9	8.75	8	RAS2->CDC42->SHO1->STE20->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	14	9	8.75	8	RAS2->CDC42->BEM1->STE20->STE7->FUS3->STE12->TEC1
RAS2	TEC1	15	9	8.75	8	RAS2->CDC42->STE20->STE5->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	16	9	8.75	8	RAS2->CDC42->STE20->STE7->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	17	9	8.75	8	RAS2->CDC42->STE20->STE11->STE7->FUS3->STE12->TEC1
RAS2	TEC1	18	9	8.75	8	RAS2->CDC42->STE20->STE11->STE5->FUS3->STE12->TEC1
RAS2	TEC1	19	9	8.75	8	RAS2->CDC42->STE20->STE7->KSS1->DIG1->STE12->TEC1
RAS2	TEC1	20	9	8.75	8	RAS2->CDC42->SHO1->STE20->STE5->FUS3->STE12->TEC1
RAS2	TEC1	21	9	8.75	8	RAS2->CDC42->BEM1->STE20->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	22	9	8.75	8	RAS2->CDC42->STE20->STE11->FUS3->DIG2->STE12->TEC1
RAS2	TEC1	23	9	8.75	8	RAS2->CDC42->STE20->STE7->FUS3->DIG2->STE12->TEC1
RAS2	TEC1	24	9	8.75	8	RAS2->CDC42->BEM1->STE20->FUS3->DIG2->STE12->TEC1
RAS2	TEC1	25	9	8.75	8	RAS2->CDC42->STE20->FUS3->DIG1->DIG2->STE12->TEC1
RAS2	TEC1	26	9	8.75	8	RAS2->CDC42->STE20->FUS3->DIG2->DIG1->STE12->TEC1
RAS2	TEC1	27	9	8.75	8	RAS2->CDC42->SHO1->STE20->STE11->FUS3->STE12->TEC1
RAS2	TEC1	28	9	8.75	8	RAS2->CDC42->STE20->STE7->KSS1->DIG2->STE12->TEC1
RAS2	TEC1	29	9	8.75	8	RAS2->CDC42->STE20->STE7->STE5->FUS3->STE12->TEC1
RAS2	TEC1	30	9	8.75	8	RAS2->CDC42->STE20->STE5->FUS3->DIG2->STE12->TEC1
RAS2	TEC1	31	9	8.75	8	RAS2->CDC42->SHO1->STE20->STE7->FUS3->STE12->TEC1
RAS2	TEC1	32	9	8.75	8	RAS2->CDC42->STE20->STE11->FUS3->DIG1->STE12->TEC1
RAS2	TEC1	33	9	8.75	8	RAS2->CDC42->BEM1->STE20->STE5->FUS3->STE12->TEC1
RAS2	TEC1	34	9	8.75	8	RAS2->CDC42->SHO1->STE20->FUS3->DIG2->STE12->TEC1

Figure 6 | Result obtained with *Pathfinder* upon execution of protocol with the study case. The table lists the paths found between RAS2 (source node) and TEC1 (target node), ranked by increasing value of weight (distance). RAS2 and TEC1 are the start and end node of the filamentous growth pathway in yeast.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution
8	After a few minutes, I still do not have any answer and the browser displays "Server is not responding"	If you submitted a heavy task, the processing may exceed 5 min. After that delay, Internet browser programs stop waiting for the server and display the error message Another possibility (if the task you submitted is not heavy) is that there is a problem with your Internet connection	For heavy tasks, it is preferable either to install the stand-alone version of the command-line tools on your machine or to write a client script for the RSAT Web services
17	No graph layout after having loaded a GML file into yEd or Cytoscape	When a graph is loaded in yEd or CytoScape, it is initially displayed with a trivial layout (all nodes on a diagonal)	In yEd: select **Layout** from the menu, then select the submenu **Organic** and choose the option **Classic**, then click **OK** In Cytoscape: select **Layout** from the menu, then select the submenu **yFiles** and choose **Organic** Both editors offer other layouts that you may try

(continued)

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
22	You obtain the message: "Error Incongruence between graph and cluster files"	Cluster and graph files do not correspond to the same network, or the specified format of the graph is not correct	Check that the clusters correspond to the graph. If so, check the format of your graph file is the one you entered in the relevant field
	The low-resolution heatmap does not display properly	The image is scaled so it fits on the window	Click on the image to zoom. You may inspect the whole map by using the scrolling bar
46	You obtain the message: "PATHFINDER ERROR: One of your seed nodes is not part of the input graph"	You provided seed node identifiers that do not match any of the node identifiers of the input graph	Check the spelling of your seed node identifiers In general, all tools require an exact match between input node identifiers and those of nodes in the network

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

ANTICIPATED RESULTS

Clustering

Figure 2 shows the results that should be obtained by applying the MCL graph clustering algorithm on the STRING database interaction network. Each cluster is highlighted with a specific color. **Figure 2a** only displays intracluster edges, so that each cluster appears as a separate component. This representation highlights the intracluster structure and edge density, and could give indication about possible improvement of the clusters by further subdivision. For example, the top-left cluster seems to be composed of several various connected regions, which could be explored in more detail, taking into account some biological knowledge. On **Figure 2b**, both intra- and interclusters are displayed. Intracluster edges are highlighted by cluster-specific colors, whereas intercluster edges are displayed in black, thereby revealing the interactions that were discarded during the clustering procedure. These two representations thus provide complementary indication for the interpretation of the clustering result.

The heatmap in **Figure 3** represents a section of the node–cluster membership matrix, with cells colored according to the membership degree (the darker the cell, the higher the membership value). Within each cell, the membership degree values are displayed, indicating how strongly each node (row) is connected to each cluster (column). This strength (node–cluster membership) is defined as the sum of weights of the edges connecting the considered node to the considered cluster, divided by the sum of weights of all edges starting from this node. The figure shows only a fragment of the table, but it already appears that some genes have similar membership profiles, thereby suggesting their involvement in common functions. This is the case of the genes *LPD1*, *PDA1* and *PYC1*, which are involved in pyruvate metabolism.

Figure 4 shows the results of the comparison between the dense clusters of the STRING graph and the complexes annotated in the MIPS database. The header gives a short explanation for the content of each column of the result table. In this case, results are sorted by decreasing values of the hypergeometric significance (last column) calculated as described in **Box 2**. Each row describes the comparison between one MCL cluster and one MIPS complex. For example, the first row compares the MCL cluster *'cl_2'*, which contains 151 proteins, with a set of 138 proteins involved in cytoplasmic ribosomes. The intersection contains 123 proteins, which represents a very high fraction of both the MCL cluster, and the annotated complex. The probability to observe such an intersection by chance is 4E–146. The E value, obtained with the correction for multitesting, indicates that the number of false positives expected with such a *P* value would be 1E–141. In other terms, the correspondence between this MCL cluster and the cytoplasmic ribosome is too high to be explained by chance.

Network comparison

Figure 5a shows the statistics of comparison between the STRING 'database' network and the BioGRID synthetic lethality data set. The 'synthetic lethality' network used as reference contains 2,352 proteins linked by 9,413 edges, and the query graph contained 1,240 nodes and 11,027 edges. The intersection between those graphs is apparently weak: 138 edges only. The Jaccard coefficient indicates that this intersection represents no $>0.68\%$ of the union. However, the number of edges expected by chance at the intersection is even smaller: E(Q^R) = 23.24. The hypergeometric *P* value (**Box 2**) indicates the probability to observe at least 138 edges at the intersection when 23.24 are expected by chance. In this study case, we observe that even with no $>0.68\%$ of edges at the intersection, the *P* value is very low (1.4E−59). In other terms, the number of edges at the intersection is too much high to be explained by chance, and is more likely to result from the biological relevance of both datasets.

Pathfinder

The known signal transduction path connecting RAS2 and TEC1 consists of the following steps[18]:
RAS2 - CDC42 - STE20 - STE11 - STE7 - KSS1 - DIG1/2 - TEC1

Pathfinder reports the following path of first rank (the matching parts are underlined, and the nonseed matching part are highlighted in bold):

RAS2 - **CDC42 - STE20** - FUS3 - STE12 - TEC1

This path connects STE20 to TEC1 via FUS3 and STE12, bypassing STE11, STE7, KSS1 and DIG1/2.

Among the paths of length 8 (third rank paths), we find paths closer to the annotated pathway, such as

RAS2 - **CDC42 - STE20 - STE11** - FUS3 - **DIG1** - STE12 - TEC1

Scott and colleagues applied their path-finding algorithm to another yeast protein–protein interaction network of similar size (4,500 nodes and 14,500 edges) taken from MIPS. For RAS2 and TEC1, they obtain the following as best path of length 8:

RAS2 - CDC25 - HSP82 - **STE11** - STE5 - **STE7** - **KSS1** - TEC1

Although Pathfinder has not been designed in particular for protein interaction networks, it can be used to predict signal transduction pathways if appropriate weights have been set on the network under investigation. The accuracy of the prediction depends also on the data quality. For example, the STRING interaction network does not contain any edge between DIG2/DIG1 and TEC1, making it impossible to reach a prediction accuracy of 100%. When predicting pathways from real-world interaction networks, one must always keep in mind that these data might be incomplete or contain false positive interactions.

ACKNOWLEDGMENTS S.B. is the recipient of a PhD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA). K.F. is supported by the Actions de Recherches Concertées de la Communauté Française de Belgique (ARC grant number 04/09-307). G.L.-M. was funded by a PhD grant from the Fonds Xenophilia [Université Libre de Bruxelles (ULB)] and by a postdoctoral fellowship from the Région Wallonne de Belgique (TransMaze project 415925). The BiGRe laboratory is supported by the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265) and by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet). We acknowledge the students of the Master in Bioinformatics and Modeling (ULB, Belgium) for their useful corrections and suggestions. S.B. and K.E. equally contributed to this article.

Published online at http://www.natureprotocols.com/
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/

1. Thomas-Chollier, M. *et al.* RSAT: regulatory sequence analysis tools. *Nucleic Acids Res.* **36**, W119–W127 (2008).
2. Brohée, S. *et al.* NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Res.* **36**, W444–W451 (2008).
3. Turatsinze, J.-V., Thomas-Chollier, M., Defrance, M. & van Helden, J. Using RSAT to scan genome sequences for transcription factor binding sites and *cis*-regulatory modules. *Nat. Protoc.* doi:10.1038/nprot.2008.97 (2008).
4. Defrance, M., Janky, R., Sand, O. & van Helden, J. Using RSAT oligo-analysis and dyad-analysis tools to discover regulatory signals in nucleic sequences. *Nat. Protoc.* doi:10.1038/nprot.2008.98 (2008).
5. Sand, O., Thomas-Chollier, M., Vervisch, E. & van Helden, J. Analyzing multiple data sets by interconnecting RSAT programs via SOAP Web services–an example with ChIP-chip data. *Nat. Protoc.* doi:10.1038/nprot.2008.99 (2008).
6. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N. & Barabási, A.L The large-scale organization of metabolic networks. *Nature* **407**, 651–654 (2000).
7. Jeong, H., Mason, S.P., Barabási, A.L. & Oltvai, Z.N. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
8. Fell, D.A. & Wagner, A. The small world of metabolism. *Nat. Biotechnol.* **18**, 1121–1122 (2000).
9. Blatt, M., Wiseman, S. & Domany, E. Superparamagnetic clustering of data. *Phys. Rev. Lett.* **76**, 3251–3254 (1996).
10. Bader, G.D. & Hogue, C.W.V. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* **4**, 2 (2003).
11. Gagneur, J., Jackson, D.B. & Casari, G. Hierarchical analysis of dependency in metabolic networks. *Bioinformatics* **19**, 1027–1034 (2003).
12. Spirin, V. & Mirny, L.A. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA* **100**, 12123–12128 (2003).
13. King, A.D., Przulj, N. & Jurisica, I. Protein complex prediction via cost-based clustering. *Bioinformatics* **20**, 3013–3020 (2004).
14. Van Dongen, S. *Graph Clustering by Flow Simulation*. PhD Thesis (Centers for Mathematics and Computer Science (CWI), University of Utrecht, 2000).
15. Pereira-Leal, J.B., Enright, A.J. & Ouzounis, C.A. Detection of functional modules from protein interaction networks. *Proteins* **54**, 49–57 (2004).
16. Enright, A.J., Van Dongen, S. & Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**, 1575–1584 (2002).
17. Brohée, S. & van Helden, J. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* **7**, 488 (2006).
18. Scott, J., Ideker, T., Karp, R.M. & Sharan, R. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.* **13**, 133–144 (2005).
19. Bebek, G. & Yang, J. PathFinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC Bioinformatics* **8**, 335 (2007).
20. Rahman, S.A., Advani, P., Schunk, R., Schrader, R. & Schomburg, D Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics* **21**, 1189–1193 (2004).
21. Croes, D., Couche, F., Wodak, S. & van Helden, J. Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.* **33**, W326–W330 (2005).
22. Croes, D., Couche, F., Wodak, S. & van Helden, J. Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.* **356**, 222–236 (2006).
23. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
24. de Nooy, W., Mrvar, A. & Batagelj, V. *Exploratory Social Network Analysis with Pajek* Series: Structural Analysis in the Social Sciences (No. 27) (Cambridge University Press, Cambridge, 2005).
25. Baitaluk, M., Sedova, M., Ray, A. & Gupta, A. BiologicalNetworks: visualization and analysis tool for systems biology. *Nucleic Acids Res.* **34**, W466–W471 (2006).
26. Hu, Z. *et al.* VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res.* **35**, W625–W632 (2007).
27. Hull, D. *et al.* Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.* **34**, W729–W732 (2006).
28. Lima-Mendez, G., van Helden, J., Toussaint, A. & Leplae, R. Reticulate representation of evolutionary and functional relationships between phage genomes. *Mol. Biol. Evol.* **25**, 762–777 (2008).
29. Croes, D., Couche, F., Wodak, S.J. & van Helden, J. Inferring meaningful pathways in weighted metabolic networks. *J. Mol. Biol.* **356**, 222–236 (2006).
30. Croes, D., Couche, F., Wodak, S.J. & van Helden, J. Metabolic PathFinding: inferring relevant pathways in biochemical networks. *Nucleic Acids Res.* **33**, W326–W330 (2005).
31. von Mering, C. *et al.* STRING 7-recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.* **35**, D358–D362 (2007).
32. Breitkreutz, B.J. *et al.* The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.* **36**, D637–D640 (2008).
33. Keseler, I.M. *et al.* EcoCyc: a comprehensive database resource for *Escherichia coli*. *Nucleic Acids Res.* **33**, D334–D337 (2005).
34. Kanehisa, M., Goto, S., Kawashima, S. & Nakaya, A. The KEGG databases at GenomeNet. *Nucleic Acids Res.* **30**, 42–46 (2002).
35. Mewes, H.W. *et al.* MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* **32**, D41–D44 (2004).
36. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
37. Jimenez, V.M. & Marzal, A. Computing the K shortest paths: a new algorithm and an experimental comparison. In *Proceeding of the 3rd International Workshop on Algorithm Engineering (WAE 1999)* Vol. **1668**, 15–29 (Springer-Verlag, London, 1999).
38. Eppstein, D. Finding the k shortest paths. *SIAM J. Comput.* **28**, 652–673 (1998).

© 2008 Nature Publishing Group http://www.nature.com/natureprotocols

Chapitre 6

Ré-implémentation de *YTPdb*, une base de données dédiée aux perméases de la levure

Pour permettre une analyse approfondie des interactions entre les transporteurs membranaires, nous avons ré-implémenté la base de données *YTPdb* (**Y**east **T**ransport **P**rotein database). *YTPdb* abrite les données collectées depuis 1997 par Bruno André reprenant une caractérisation et une annotation précise et régulièrement mise à jour de 284 protéines de levures classifiées comme transporteurs membranaires caractérisés ou potentiels [17].

6.1 Bases de données de transporteurs

Un bon nombre de bases de données dédiées aux transporteurs étaient déjà accessibles en ligne. Par exemple, TCDB (Transport Commission database) qui établit une classification de transporteurs dans plus de 500 classes et TransportDB qui classe les transporteurs de tous les organismes entièrement séquencés selon ce système. D'autres bases de données plus spécifiques ne s'intéressent qu'à un type de transporteurs ou aux transporteurs d'un seul organisme. C'est le cas notamment de YETI (Yeast Transport Information) [146], de YMPL et AMPL (Yeast (ou Arabidpsis) Membrane Protein Library, *http ://wardlab.cbs.umn.edu/yeast/*) [147], de Aramemnon (dédiée au transporteurs membranaires de plusieurs plantes) [148] ou de P-type ATPase database [149]. Enfin, des informations sur les transporteurs peuvent aussi être trouvées sur des bases de données générales abordant tous les gènes d'un organisme comme SGD (*Saccharomyces* genome database) [150].

6.2 Specificités de YTPdb

Les bases de données que nous avons évoquées sont très utiles puisqu'elles fournissent des informations permettant d'appréhender la diverité et l'évolution des transporteurs membranaires. Cependant, les biologistes moléculaires travaillant de le domaine des perméases ont aussi besoin de resources fournissant une caractérisation fonctionnelle de ces protéines (localisation, substrat, classification fonctionnelle, etc). Par ailleurs, les informations contenues dans ces resources doivent être basées sur des données de la littérature et pas seulement sur des extrapolations provenant uniquement de comparaisons de séquences.

C'est dans cette optique que Bruno André et Daniel Van Belle ont implémenté la première version de YTPdb (qui consistait en une série de tables HTML) [17], une base de données accessible à tous en ligne et dédiée à l'annotation précise et à la classification des perméases de *S.cerevisiae*. Au cours de cette thèse, nous avons élaboré un schéma relationnel de base de données et l'avons implémenté dans le système de gestion PostgreSQL.

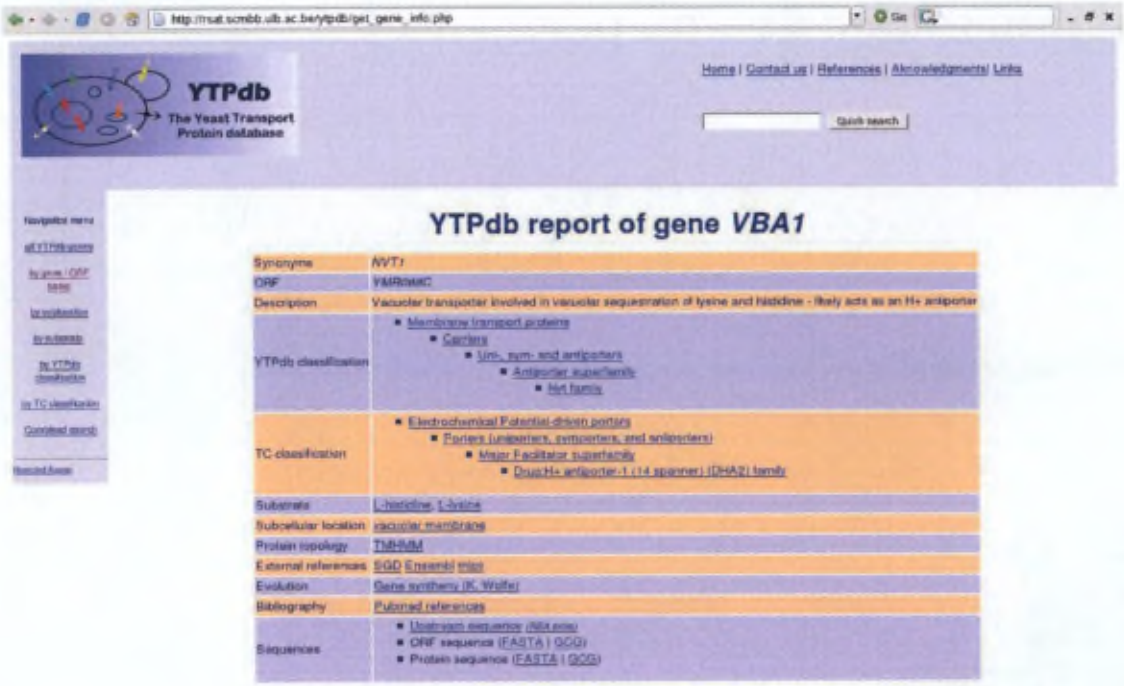


FIG. 6.1 – Données annotées dans YTPdb pour la perméase vacuolaire Vba1p.

6.2.1 Contenu

YTPdb contient des annotations manuelles issues de la littérature et concernant 284 perméases établies ou putatives présentant une forte similarité de séquence avec une autre perméase annotée. Pour chaque protéine, YTPdb propose une courte description, sa séquence, sa localisation cellulaire, la liste des substrats qu'elle transporte, un lien vers les références à son sujet dans PubMed et la prédiction de ses segments transmembranaires (Figure 6.1).

Les différents critères de classification comme la localisation ou le substrat sont organisés de façon hiérarchique. Ainsi, un transporteur de *glucose* est aussi classé comme un transporteur d'*hexose*, de *monosaccharides* et d'*hydrates de carbone*. De même une protéine localisée dans la *membrane de l'appareil de Golgi* est aussi répertoriée comme une protéine de la *membrane de l'appareil de Golgi et de l'endosome*, de la *voie de sécrétion* et des *membranes internes*. Ceci permet d'aussi classer les protéines même lorsqu'elles n'ont pas été annotées avec précision.

Les transporteurs de YTPdb sont classés selon la Transport Commision qui fait référence dans le domaine. Bruno André a par ailleurs développé une autre classification basée sur les similarités de séquences et comprenant plus de subdivisions mieux adaptées aux transporteurs membranaires de la levure. Un alignement des séquences protéiques (au moyen de l'algorithme MUSCLE) est accessible pour les dernières subdivisions.

Enfin, on peut effectuer des recherches dans YTPdb selon un seul critère (localisation, substrat, classe) ou une combinaison de plusieurs critères.

6.2.2 Accès et interface

YTPdb est donc une base de données relationnelles accessible à l'adresse *http ://rsat.ulb.ac.be/ytpdb/*. L'accès par le web aux 13 tables et aux fichiers la composant est permis au moyen du langage de programmation PHP, utilisé pour interroger la base de données et renvoyer les données sous format HTML.

6.2.3 Conclusions et perspectives

Avec YTPdb, nous avons fourni aux chercheurs qui travaillent dans le domaine des transporteurs membranaires une base de données manuellement annotée concernant 284 perméases de *S.cerevisiae*. Ses caractéristiques principales sont l'annotation basée sur la littérature et les différentes classifications fonctionnelles organisées hiérarchiquement.

Plusieurs améliorations pourraient être envisagées. Parmi celles-ci, l'ajout des modifications post-traductionnelles (phosphorylation, ubiquitination, etc) étudiées dans certains articles récents ou les différentes interactions physiques détectées entre les transporteurs.

Enfin, le schéma de notre base de données pourrait intéresser d'autres groupes impliqués dans l'étude d'autres classes de protéines (Figure 6.2).

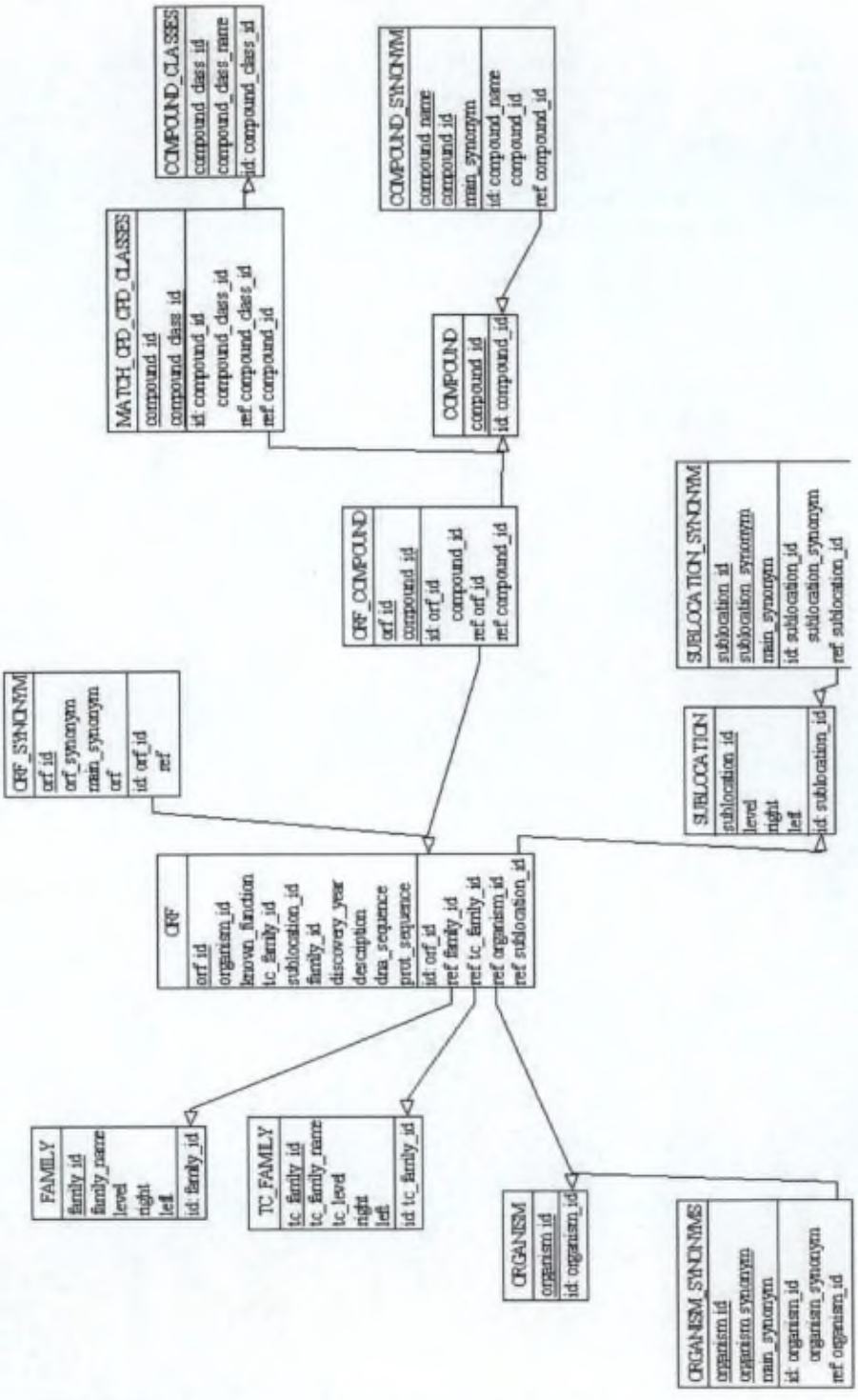


FIG. 6.2 – Schéma relationnel de YTPdb. Chaque rectangle représente une table relationnelle et les différents champs qu'elle contient. Les champs soulignés représentent les clefs primaires (identificateurs) de chaque table. Les champs précédés de *id :* présentent une contrainte d'unicité. Les champs précédés de *ref* sont des clefs secondaires, ils se réfèrent à un identifiant qui est la clef primaire d'une autre table. Ce diagramme a été réalisé au moyen du programme DBMAIN (*http ://www.db-main.be/*).

Chapitre 7

Evaluation de méthodes de clustering de réseaux d'interactions entre protéines

Dans la suite, nous disposerons d'un réseau d'interactions entre protéines associées aux membranes chez la levure. Afin de découvrir au sein de ce réseau des groupes de protéines densément interconnectées et correspondant éventuellement à des complexes protéiques, nous utiliserons des méthodes de clustering. En effet, une des applications du clustering dans la théorie des graphes consiste à regrouper dans un cluster les noeuds présentant plus de connexions entre eux qu'ils n'en présentent avec les autres noeuds du graphe (section 3.3 p. 42).

7.1 Intérêt de l'évaluation

Comme nous l'avons déjà mentionné, on ne compte plus le nombre de publications décrivant l'application d'une nouvelle méthode de clustering à un graphe d'interactions entre protéines. Chacune de ces méthodes présente des paramètres qu'il convient de choisir judicieusement selon le réseau qui va être analysé. Généralement, selon leurs auteurs, ces méthodes surpassent ou présentent des résultats au moins comparables aux méthodes pré-existantes. Cependant, ces affirmations reposent souvent sur des évaluations quasiment non existantes (quelques exemples de clusters biologiquement valides) ou biaisées (le développeur de l'algorithme compare sa méthode aux autres, il a donc une meilleure connaissance des paramètres optimaux de son approche dans son cas d'étude particulier).

Dans le domaine des statistiques, des études proposant des métriques d'évaluation (index de Rand [151] ou autres [152]) ou des comparaisons de différentes approches de clustering [153] avaient déjà été menées précédemment. Cependant, à notre connaissance, aucune comparaison objective n'avait encore été menée pour le cas précis de la découverte de complexes protéiques dans les réseaux d'interactions. Nous avons donc décidé de comparer les résultats obtenus suite à l'application de différentes méthodes de clustering utilisées en biologie à un réseau d'interactions construit à partir de complexes annotés. Pour cela, nous avons utilisé nos propres métriques d'évaluations définies à partir de métriques classiquement utilisées pour l'évaluation, notamment en bioinformatique (la sensibilité et la *PPV*).

7.2 Description de l'article présenté

Sylvain Brohée et Jacques van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics. 2006 Nov 6 ;7 :488 [111].

Dans cet article nous évaluations la capacité de 4 algorithmes de clustering à retrouver des clusters correspondants à des complexes protéiques biologiquement valides au sein de graphe d'interaction entre protéines.

7.2.1 Description des algorithmes évalués

Le choix des quatre algorithmes évalués s'est fait en fonction de deux critères : premièrement, l'algorithme devait avoir été appliqué à un réseau d'interaction pour la découverte de complexes et ensuite et ensuite il devait fonctionner en ligne de commande sur nos systèmes (Unix). Les algorithmes étudiés sont *MCODE* [130], *MCL* [129, 113], *RNSC* [126] et *SPC* [127, 128]. *MCODE* et *MCL* ont déjà été décrits dans la partie *Introduction* de ce travail (section 3.3.3.5 p. 46).

RNSC et *SPC* sont tous les deux des algorithmes d'optimisation. *RNSC* vise à minimiser le nombre d'arêtes inter-cluster et à maximiser le nombre d'arêtes intra-cluster.

Le principe de *SPC* est basé sur une analogie avec les propriétés physiques *spins* d'un aimant ferromagnétique à des températures supérieures à 0. Dans une première étape, chaque noeud du graphe se voit attribuer une valeur de spin aléatoire. Suite à l'application de l'algorithme, les *spins* des groupes de noeuds densément interconnetés varient de façon corrélée et en fin de processus, les noeuds présentant des spins corélés sont placés dans les mêmes clusters. La température est un paramètre important puisque lorsque la température augmente, le système est moins stable et les clusters seront plus petits.

7.2.2 Principe de l'évaluation

Pour évaluer les résultats retournés par les différents algorithmes de clustering, nous avons défini un graphe pour lequel les complexes sont connus. A partir de 220 complexes protéiques décrits dans la banque de données MIPS [20], nous avons généré un graphe en considérant une interaction entre chacune des paires de protéines d'un complexe. Le graphe ainsi construit contient 1120 protéines et 12300 arêtes. Les graphes d'interaction qui sont analysés par les méthodes de clustering sont généralement obtenus à partir des données à haut débit et contiennent un certain nombre de faux négatifs et de faux positifs. Nous avons donc choisi d'étudier la robustesse des algorithmes à l'addition de bruit. Les faux positifs et faux négatifs ont été générés en combinant l'ajout et la délétion aléatoires de différentes proportions d'arêtes aux graphes que nous avions construits.

Chacun des graphes obtenus suite à cette altération du graphe original a ensuite été soumis aux quatre algorithmes en envisageant différentes combinaisons de leurs paramètres. Pour chacun des résultats de clustering obtenu, nous avons chaque fois deux contrôles aléatoires par ré-échantillonage. Le premier est construit en ré-assignant aléatoirement les protéines à des clusters de même taille que les clusters renvoyés par l'algorithme et le second en appliquant l'algorithme sur un graphe aléatoires obtenu en ré-échantillonnant aléatoirement les arêtes entre les noeuds.

Pour comparer la composition en protéines entre clusters prédits et complexes annotés, nous avons construit une *table de contingence* pour chaque résultats de clustering. Il s'agit d'un tableau T où chaque ligne i correspond à un complexe annoté et chaque colonne j à un des clusters retournés par l'algorithme. Dans chaque cellule T_{ij}, on indique le nombre de protéines communes au complexe i et au cluster j.

Cette table nous a permis de calculer la sensibilité et la PPV (valeur prédictive positive) des deux approches pour chaque complexe et cluster, respectivement. Ces deux métriques courament utilisées dans les études d'évaluation, ont été adaptées pour notre cas : la sensibilité indiquant à quel point les protéines d'un même complexe sont regroupées dans le même cluster (fraction maximale des protéines du complexe trouvées dans un cluster commun) et la PPV quantifiant à quel point un cluster est fiable pour la prédiction d'un complexe (fraction maximale des protéines du cluster trouvées dans un complex commun).

La *sensibilité* et la PPV pouvant présenter certains inconvénients (décrits dans l'article), nous avons aussi créé une métrique appelée *séparation* qui reflète la correspondance bi-directionnelle entre complexes et clusters. Ainsi, la *séparation au niveau d'un complexe* indique si un complexe est bien isolé des autres. Sa valeur est maximale lorsque toutes les protéines du complexe sont contenues dans un seul cluster et que ce cluster ne contient pas d'autres protéines. De même, la *séparation au niveau d'un cluster* indique si un cluster isole un ou plusieurs complexes des autres clusters, c'est-à-dire s'il comprend l'entièreté des protéines de un plusieurs complexes.

Pour chaque résultat de clustering, les moyennes des valeurs de sensibilité ou de séparation au niveau d'un complexe et de PPV ou de séparation au niveau d'un cluster ont été calculées sur l'ensemble des complexes et des clusters, respectivement.

7.2.3 Résultats

7.2.3.1 Paramètres optimaux et étude de la robustesse

Suite à cette évaluation, nous avons systématiquement évalué l'impact des paramètres des alogorithmes sur les performances des programmes. Nous avons observé que le choix des paramètres n'altérait que peu les résultats obtenus avec *RNSC*, contrairement aux autres et avons pu dériver les valeurs optimales des paramètres de chaque algorithme.

Par ailleurs, il est apparu que les algorithmes *RNSC* et *MCL* donnent généralement des résultats supérieurs et sont généralement plus robustes à l'addition et à la suppression aléatoire d'arêtes que *MCODE* et *SPC*.

7.2.3.2 Evaluation des performances des algorithmes sur des réseaux d'interactions à haut débit

Nous avons ensuite appliqué les algorithmes à 6 jeux de données à haut débit issus de la littérature en utilisant les paramètres optimaux définis dans la première partie et estimé la qualité de ces résultats en les comparant aux complexes de référence. Nous avons pu observer que

(*a*) *SPC* donne les moins bons résultats. *MCL* retourne les meilleurs résultats sur tous les jeux de données, suivi de très près par `MCODE` et ensuite par *RNSC*.

(*b*) les résultats obtenus sur les jeux de données de double hybride et de co-immunoprécipitation de Ho *et al* [34] sont très inférieurs au résultats obtenus sur les autres jeux de données de co-immunoprécipitation (Gavin *et al* [36, 35][36, 35] et Krogan *et al* [37]). Il avait déjà été signalé par ailleurs que les données de cette étude étaient de moindre qualité [38].

(*c*) *MCODE* montre les meilleurs résultat pour le contrôle négatif. Comme il ne s'agit pas d'un algorithme de partitionnement, les clusters non significatifs ne sont pas inclus dans le résultat.

7.2.4 Conclusion

Cette analyse nous a permis de déterminer quel algorithme utiliser et avec quels paramètres dans la suite de notre travail. En effet, dans notre étude l'algorithme *MCL* combine plusieurs avantages. Il donne les meilleurs résultats à la fois lorsqu'il est appliqué au réseau construit à partir des complexes et lorsqu'il est appliqué à des vrais jeux de données à grande échelle. Ensuite, il est relativement facile à utiliser et son implémentation est souvent mise à jour sur le site Internet de son développeur. Son principal inconvénient réside dans le choix de ses paramètres qui peuvent grandement affecter la qualité des résutalts.

Par ailleurs, grâce à cette évaluation, nous avons pu établir un standard pour l'évaluation de méthodes de clustering qui a déjà été repris plusieurs fois par d'autres auteurs dans les mois qui ont suivi sa publication [37, 131, 154, 155, 156].

BMC Bioinformatics

BioMed Central

Research article

Open Access

Evaluation of clustering algorithms for protein-protein interaction networks

Sylvain Brohée* and Jacques van Helden

Address: Service de Conformation des Macromolécules Biologiques et de Bioinformatique. Université Libre de Bruxelles, CP 263, Campus Plaine, Bd. du Triomphe, B-1050 Bruxelles, Belgium

Email: Sylvain Brohée* - sylvain@scmbb.ulb.ac.be; Jacques van Helden - jvanheld@scmbb.ulb.ac.be

* Corresponding author

Published: 06 November 2006

BMC Bioinformatics 2006, **7**:488 doi:10.1186/1471-2105-7-488

Received: 15 March 2006
Accepted: 06 November 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/488

© 2006 Brohée and van Helden; licensee BioMed Central Ltd.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: Protein interactions are crucial components of all cellular processes. Recently, high-throughput methods have been developed to obtain a global description of the interactome (the whole network of protein interactions for a given organism). In 2002, the yeast interactome was estimated to contain up to 80,000 potential interactions. This estimate is based on the integration of data sets obtained by various methods (mass spectrometry, two-hybrid methods, genetic studies). High-throughput methods are known, however, to yield a non-negligible rate of false positives, and to miss a fraction of existing interactions.

The interactome can be represented as a graph where nodes correspond with proteins and edges with pairwise interactions. In recent years clustering methods have been developed and applied in order to extract relevant modules from such graphs. These algorithms require the specification of parameters that may drastically affect the results. In this paper we present a comparative assessment of four algorithms: Markov Clustering (MCL), Restricted Neighborhood Search Clustering (RNSC), Super Paramagnetic Clustering (SPC), and Molecular Complex Detection (MCODE).

Results: A test graph was built on the basis of 220 complexes annotated in the MIPS database. To evaluate the robustness to false positives and false negatives, we derived 41 altered graphs by randomly removing edges from or adding edges to the test graph in various proportions.

Each clustering algorithm was applied to these graphs with various parameter settings, and the clusters were compared with the annotated complexes.

We analyzed the sensitivity of the algorithms to the parameters and determined their optimal parameter values.

We also evaluated their robustness to alterations of the test graph.

We then applied the four algorithms to six graphs obtained from high-throughput experiments and compared the resulting clusters with the annotated complexes.

Conclusion: This analysis shows that MCL is remarkably robust to graph alterations. In the tests of robustness, RNSC is more sensitive to edge deletion but less sensitive to the use of suboptimal parameter values. The other two algorithms are clearly weaker under most conditions.

The analysis of high-throughput data supports the superiority of MCL for the extraction of complexes from interaction networks.

Background

Protein-protein interactions (PPI) play major roles in the cell: transient protein interactions are often involved in post-translational control of protein activity; enzymatic complexes ensure substrate channeling which drastically increases fluxes through metabolic pathways; large protein complexes play essential roles in basal cellular mechanisms such as DNA packaging (histones), transcription (RNA polymerase), replication (DNA polymerase), translation (ribosome), protein degradation (proteasome) ...

Various methods have been used to detect PPI. Co-immunoprecipitation, co-sedimentation, and two-hybrid systems have traditionally been used to characterize interactions at the level of a single protein complex. More recently, high-throughput methods have been developed for large-scale detection of pairwise interactions (two-hybrid systems, the split-ubiquitin method) [1-3] or multi-protein complexes (TAP-TAG, HMS-PCI) [4-7].

In 2002, von Mering *et al.* estimated that data resulting from combined experimental and computational approaches provide clues in favor of approximately 80,000 PPI in the yeast *Saccharomyces cerevisiae* [8]. Clearly, however, this information should be considered with caution, since all methods are known to yield a non-negligible amount of noise (false positives) and to miss a fraction of existing interactions (false negatives). The error rate depends strongly on the method, high-throughput and computational methods being less reliable than traditional methods [9].

The network of interactions between proteins is generally represented as an interaction graph, where nodes represent proteins and edges represent pairwise interactions. Graph theory approaches have been applied to describe the topological properties of the network: distribution of node degree (number of incoming and outgoing edges per node), network diameter (average of the shortest distance between pairs of nodes), clustering coefficient (proportion of the potential edges between the neighbors of a node that are effectively observed in the graph). These analyses have led to the observation of some apparently recurrent properties of biological networks: power-law degree distribution, small world, high clustering coefficients, and modularity [10-15].

Beyond these descriptive statistics, an important challenge for modern biology is to understand the relationship between the organization of a network and its function. In particular, it is essential to extract functional modules such as protein complexes [16] or regulatory pathways [17] from global interaction networks.

To achieve this goal, several clustering methods have been applied to the protein interactome graph in order to detect highly connected subgraphs (e.g. [18-34]). These algorithms rely on very different approaches. Each of them requires specifying several parameters, some of which may drastically affect the results. To our knowledge, no systematic study has yet been performed to evaluate and compare these programs. It is thus very difficult for a biologist to estimate the reliability of hypotheses emerging from computer-based analyses of interaction networks.

In this paper we present a systematic quantitative evaluation of the capability of four clustering methods for inferring protein complexes from a network of pairwise protein interactions. The four methods tested here are Markov Clustering (MCL [35,36]), Restricted Neighborhood Search Clustering (RNSC [21]), Molecular Complex Detection (MCODE [19]), and Super Paramagnetic Clustering (SPC [37]). For each program, we sample the parameter space and select optimal parameters. We evaluate the robustness of the programs to false positives and false negatives. The algorithms are then applied to six data sets from high-throughput experiments.

Results and discussion

Algorithms

The four algorithms tested here rely on distinct approaches for extracting clusters from the graph (Table 1). We give hereafter a short conceptual description. More information can be found in the supplementary material [see Additional file 1] and original publications.

The Markov Cluster algorithm (MCL) [35,36] simulates a flow on the graph by calculating successive powers of the associated adjacency matrix. At each iteration, an *inflation step* is applied to enhance the contrast between regions of strong or weak flow in the graph. The process converges towards a partition of the graph, with a set of high-flow regions (the clusters) separated by boundaries with no flow. The value of the *inflation parameter* strongly influences the number of clusters.

The second algorithm, Restricted Neighborhood Search Clustering (RNSC) [21]), is a cost-based local search algorithm that explores the solution space to minimize a cost function, calculated according to the numbers of intra-cluster and inter-cluster edges. Starting from an initial random solution, RNSC iteratively moves a vertex from one cluster to another if this move reduces the general cost. When a (user-specified) number of moves has been reached without decreasing the cost function, the program ends up.

The third algorithm, Super Paramagnetic Clustering (SPC) [37] is a hierarchical clustering algorithm inspired from

Table 1: Main features of the graph clustering approaches presented in this study.

	Restricted Neighborhood Search Clustering (RNSC)	Markov Clustering (MCL)	Molecular Complex Detection (MCODE)	Super-paramagnetic clustering (SPC)
Type	Local search cost based	Flow simulation	Local neighbourhood density search	Hierarchical
Allow multiple assignations	No	No	Yes	No
Allow unassigned nodes	No	No	Yes	No
Edge-weighted graphs supported	No	Yes	No	Yes
First application	Protein complex prediction	Protein family detection	Protein complex detection	
Other applications	/	Identification of ortholog groups, protein complexes, peer-to-peer node clustering, image retrieval, Word Sense Discrimination, molecular pathway discovery, structural domains, ...	/	Image clustering, microarray data clustering, protein complexes detection, protein structure classification, identification of ortholog groups, ...
Availability	Upon request	http://micans.org/mcl/	ftp://ftp.blueprint.org/pub/BIND/README	Upon request
Developper	King AD	Van Dongen S	Bader GD and Hogue CWV	Blatt M, Wiseman S, Domany E
References	[21]	[35]	[19]	[18]

an analogy with the physical properties of a ferromagnetic model subject to fluctuation at nonzero temperature. At first, SPC associates a *spin* with each node of the graph. Spins belonging to a highly connected region fluctuate in a correlated fashion and nodes with correlated spins are placed in the same cluster. When the temperature increases, the system becomes less stable and the clusters become smaller.

The fourth method, Molecular Complex Detection (MCODE) [19], detects densely connected regions. First it assigns a weight to each vertex, corresponding to its local neighborhood density. Then, starting from the top-weighted vertex (seed vertex), it recursively moves outward, including in the cluster vertices whose weight is above a given threshold. This threshold corresponds to a user-defined percentage of the weight of the seed vertex.

Interaction graphs

From the collection of protein complexes annotated in the MIPS database [38], we constructed an interaction graph by instantiating a node for each protein, and linking by an edge any two proteins that belong to the same complex. This graph is hereafter referred to as the *test graph*. As depicted in Figure 1A, the structure of the original test graph is almost trivial: most complexes correspond to isolated components. In this test graph each complex is represented as a clique (each protein is connected to each other one). This generally does not reflect the actual complex structure, where each protein is linked to specific partners. Consequently, this original graph is of poor value for evaluating the performances of clustering algorithms on real data sets. This applies particularly to high-throughput data sets, which are generally fragmentary (missing interactions), and noisy (false interactions).

In order to evaluate the robustness of the algorithms to missing and false interactions, we generated 41 *altered graphs* from the original test graph, by combining addition and removal of edges in various proportions. We refer to altered graphs as $A_{add,del}$, where *add* and *del* indicate respectively the percentage of added and deleted edges (percentages with respect to the number of edges in the original test graph).

Figure 1B shows an example of an altered graph $A_{100,40}$, with 100% edge addition and 40% edge removal. Another problem of evaluation is that a certain proportion of interacting proteins can be assigned to the same cluster by chance. In order to estimate the random expectation of correct grouping, we built a *random graph* by shuffling the edges between nodes of the test graph. With this type of randomization, each node preserves the same number of links as in the original graph.

We also built 41 *altered random graphs* from the random graph, by randomly adding and removing random edges in the same proportions as for the original test graph.

To each of these 84 graphs (test, altered test, random, altered random), we applied the four algorithms described above, with varying parameter values. As a second way to estimate the random expectation, each clustering result was also randomized so as to obtain a set of *permuted clusters* of the same sizes as those obtained from the test graph or altered graphs.

Parameter optimization

The quality of a clustering result was evaluated by comparing each cluster with each annotated complex. The *complex-wise sensitivity* (*Sn*) represents the coverage of a

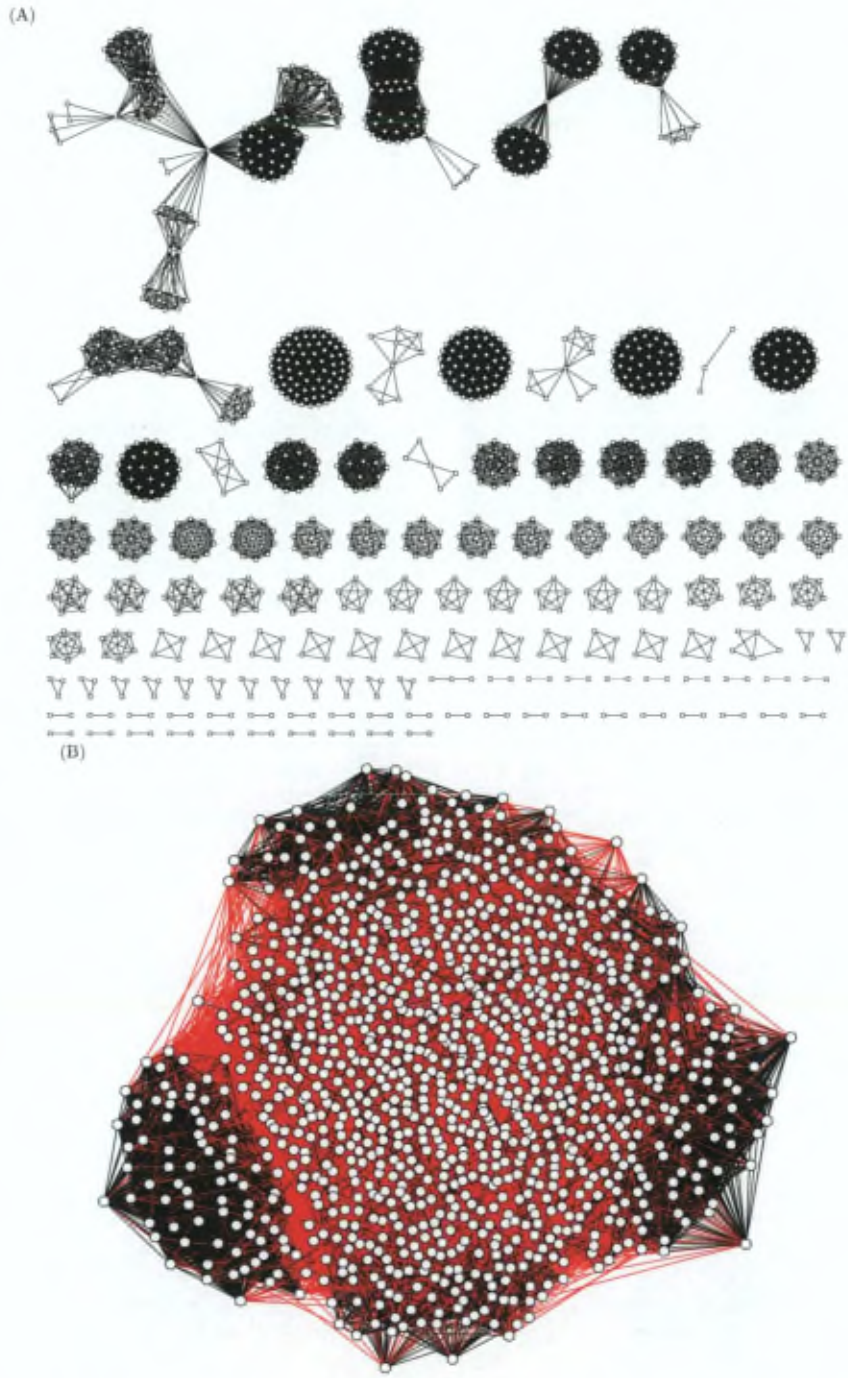


Figure 1
Graphical representation of interaction networks. (A) Test graph built from the complexes annotated in the MIPS database (high-throughput data were excluded). **(B)** Altered graph $A_{100,40}$ with 100% of random edge addition (red) and 40% of random edge removal.

complex by its best-matching cluster (the maximal fraction of proteins in the complex found in a common cluster). Reciprocally, the *cluster-wise Positive Predictive Value* (*PPV*) measures how well a given cluster predicts its best-matching complex (see the chapter *Methods* for a detailed description of the matching statistics).

To estimate the overall correspondence between a clustering result (a set of clusters) and the collection of annotated complexes, we computed the weighted means of all *PPV* values (averaged over all clusters) and *Sn* values (averaged over all complexes). The resulting statistics, *clustering-wise PPV* and *clustering-wise Sn*, provide complementary and somewhat contradictory information: when the number of clusters decreases, the *Sn* increases and, in the trivial case where all proteins are grouped in a single cluster, the calculated *Sn* reaches 1. Reciprocally, the *PPV* increases with the number of clusters, reaching 1 in the trivial case where each protein is assigned to one separate cluster. In order to integrate the two statistics, we computed a *geometrical accuracy* (*Acc*), defined as the geometrical mean of the averaged *Sn* and *PPV* values.

Each algorithm has one or more parameters that influence properties such as number of clusters, cluster size, and cluster density (number of intra-cluster edges). For each algorithm we measured the impact of the main parameters on *Sn*, *PPV* and *Acc* and selected the combination of parameters giving maximal accuracy. This analysis revealed that some parameters have a drastic impact on accuracy, whereas others have a limited effect.

Let us illustrate in more detail the procedure of parameter selection with the inflation parameter of the MCL algorithm. With the original test graph, interestingly, the effect of this parameter is barely detectable (Figure 2A). Yet this apparent robustness is an artifact due to the trivial structure of the graph. In the MIPS data set used as a reference, most proteins (73%) are members of a single complex, so that most complexes correspond to isolated components in the test graph (Figure 1A) on which the clustering is performed. Consequently, the clustering algorithm tends to define one cluster per connected component, irrespectively of the inflation parameter. Consistently with this interpretation, the number of clusters is almost constant whatever the inflation parameter value (Figure 2B, blue curve). In contrast, when the same algorithm is applied to a randomized graph, the number of clusters increases with the inflation parameter (Figure 2B, gray curve).

The crucial impact of the inflation parameter becomes obvious when MCL is applied to highly altered graphs. For example, for the altered graph $A_{100,40}$ (Figure 2C), the increase in inflation causes a decrease in *Sn* (red curve) and an increase in *PPV* (blue curve). These effects are explained by the fact that the number of clusters increases with the inflation parameter (Figure 2D). The optimal tradeoff between *Sn* and *PPV* is obtained for an inflation value of 1.7, and yields an *Acc* of 66% (green curve).

We performed the same analysis and selected the optimal parameter values for each one of the 42 graphs (test and altered), as summarized in Table 2 for the MCL algorithm. Since the optimal parameter values depend on the level of alteration, we cannot view one value as systematically optimal. We chose as a general optimum the most frequent value in this table. This criterion ensures a good robustness to graph alteration (it covers the widest range of graph alterations).

Note that in the case of the inflation parameter, the most frequent value (1.8) is especially well suited for graphs with a high level of alteration, such as those resulting from high-throughput data. In addition, for the less altered graphs, the accuracy is generally more robust to fluctuations of the inflation (the extreme case of the unaltered test graph shown in Figure 2A,B is discussed above).

For the RNSC algorithm, we tested the impact of 7 parameters on the quality of the clustering. This represents a total of 2,916 combinations of parameter values. Figure 3 displays the *Sn* (abscissa) and *PPV* (ordinate) obtained with the same altered graph as in Figure 1B ($A_{100,40}$). Each dot corresponds to one particular combination of parameter values. This figure shows that the RNSC algorithm is remarkably robust to the choice of parameter values: all the results are grouped in a cloud, with an almost constant *PPV* (58%) and a restricted range of *Sn* (between 61% and 87%).

The same analysis was carried out for each parameter of each algorithm. The complete tables of optimal values for the 42 graphs using both Accuracy and Separation (see next section) are available as supplementary material [see Additional file 2 and 3]. Table 3 synthesizes the optimal values obtained for the four tested algorithms. These optimal values were systematically used for the robustness analysis in the next section.

Robustness analysis

In this analysis, we chose fixed parameter values for each algorithm (Table 3) and analyzed the robustness of the different algorithms to various levels of graph alteration (edge removal and addition).

Figure 4A displays the impact of edge addition on the geometric accuracy. Increasing proportions of edges (0%, 5%, 10%, 20%, 40%, 80% and 100%) were randomly added to the test graph. MCL and RNSC are barely affected by addition of up to 100% edges (blue and red curves,

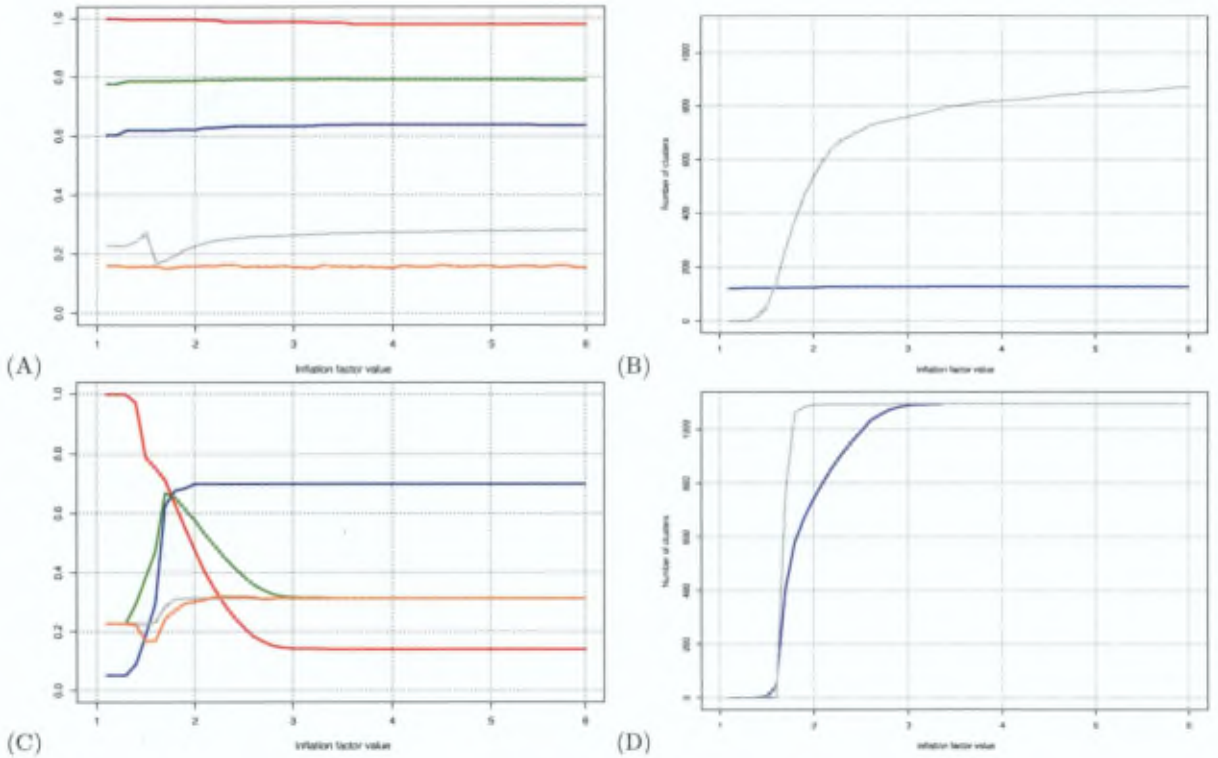


Figure 2

Impact of the inflation parameter on MCL clustering results. **(A)** Impact of the inflation parameter on the clustering-wise Sensitivity (*Sn*), Positive Predictive Value (*PPV*) and geometric accuracy (*Acc*). Each curve represents the value of one evaluation statistics (ordinate) as a function of the inflation parameter (abscissa). Color code: *blue* : *Sn*; *red* : *PPV*; *green* : *Acc*; *grey* : geometrical accuracy for the first random control (randomized graph); orange : geometrical accuracy for the second random control (permuted clusters). **(B)** Number of complexes predicted as a function of the inflation factor for the original test graph. Color code: *blue* : test graph; *red* : random graph. **(C)** *Sn*, *PPV* and *Acc* scores obtained with a highly altered graph ($A_{100,40}$). **(D)** Number of complexes predicted as a function of the inflation factor for $A_{100,40}$.

respectively). The performances of MCODE and SPC are reasonably good for low values of noise, but drop to 40% when the percentage of added edges increases to 100% (orange and green curves, respectively).

To estimate the random expectation, we performed for each clustering result a permutation test, by shuffling the proteins between clusters. The number of clusters and their respective sizes thus remained unchanged. The geometric accuracy of the permuted clusters is displayed with dotted lines in Figure 4A. For MCL, RNSC and MCODE, the accuracy is relatively stable (between 15% and 22%). For SPC, surprisingly, the accuracy of the permuted clusters progressively increases with the addition of edges, reaching 38% when more than 80% egdes are added. This value almost equals that obtained with the non-random altered graph $A_{100,0}$. This illustrates the importance of the permutation test: the test makes it possible to estimate the performance of an algorithm in terms of gains relative to the random expectation. We inspected the clustering result in more detail in order to understand why the program can yield high accuracy values even when clusters are permuted. This effect comes from the fact that, under the chosen conditions, SPC yields a huge cluster of 567 proteins, plus a multitude of very small clusters of 1 or 2 pro-

Table 2: Optimal values for MCL inflation parameter for the test and altered graphs

% removal\% addition	0	5	10	20	40	80	100
0	3.4	3.1	2.7	2.4	2	1.8	1.8
5	5.7	4	2.6	2	1.9	1.8	1.8
10	2.35	2.2	2.2	2.3	1.8	1.8	1.8
20	1.7	2.2	2.1	2	1.8	1.7	1.8
40	1.8	1.8	1.8	1.9	1.7	1.7	1.7
80	1.3	1.4	1.5	1.5	1.5	1.6	1.6

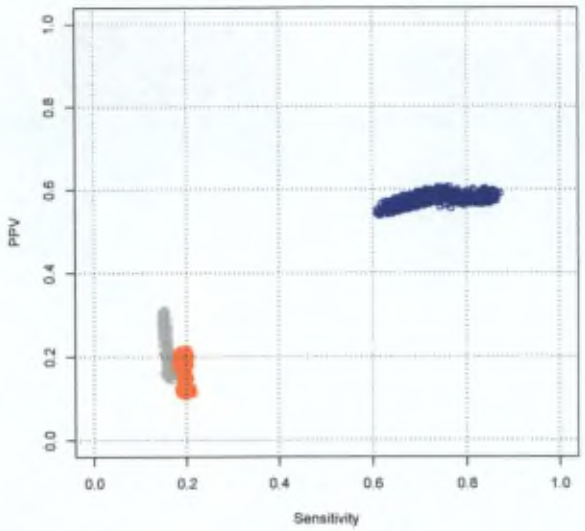


Figure 3
Impact of the RNSC parameters on the clustering of an altered graph $A_{100,40}$. Each dot represents the clustering-wise *PPV* and *Sn* value for one combination of the seven tested parameters. Color code: *blue* : altered graph $A_{100,20}$ (100% random edge addition and 20% of random edge removal); *orange* : randomized graph $R_{100,40}$; *grey* : permuted clusters.

teins. The effect of the huge cluster is to artificially increase the *Sn*, since a good fraction of each complex is covered by this cluster. Each of the very small clusters yields a high *PPV* : single-element clusters have by definition a *PPV* of 1, and 2-member clusters have a minimal *PPV* of 0.5. This particular distribution of cluster sizes thus creates an artefactual situation by reaching, for two separate reasons, reasonably high scores for both criteria (*Sn* and *PPV*).

In order to circumvent this problem, we defined an additional statistic, which we call *separation*, as the product of the proportion of complex elements found in the cluster by the proportion of cluster elements found in the complex (see Methods for the formula). High separation values indicate a bidirectional correspondence between a cluster and a complex: a maximal value of 1 is reached when a cluster corresponds perfectly with a complex, i.e. when it comprises all of its proteins and nothing more.

The *complex-wise separation* indicates how well a given complex is isolated from the other complexes. The maximal value for complex-wise separation is 1. The simplest way to obtain $Sep_{co_i} = 1$ is the perfect match, i.e. when all the proteins in the complex are contained in a single cluster, and this cluster does not contain any other protein (Table 4, cluster 1/complex 1). Yet the value of 1 can also be reached if the complex is split into two or more clusters, if each of these clusters contains only members of the complex (Table 4, complex 2 split into clusters 2 and 3). In other words, $Sep_{co_i} = 1$ indicates that the clustering algorithm separates this complex perfectly from all other complexes (although this complex may be split into several clusters).

Similarly, we defined a *cluster-wise separation*, which indicates how well a given cluster isolates one or several complexes from the other clusters. The maximal value, Sep_{cl_j}

Table 3: Optimal parameters

Algorithm	Parameter	Optimized for accuracy	Optimized for separation
MCL	Inflation	1.8	1.8
MCODE	Depth	100	5
	Node score percentage	0	0
	Haircut	TRUE	TRUE
	Fluff	FALSE	FALSE
	Percentage for complex fluffing	0.2	0.9
RNSC	Diversification frequency	50	50
	Shuffling diversification length	9	3
	Tabu length	50	50
	Tabu list tolerance	1	1
	Number of experiments	3	3
	Naive stopping tolerance	1	15
	Scaled stopping tolerance	15	15
SPC	Number of nearest neighbours	15	10
	Temperature	0.132	0.116

= 1, indicates that a cluster fully and exclusively comprises all the elements of one or several complexes, i.e. it contains all the proteins of the considered complex(es), and no other cluster contains any of these proteins.

The *clustering-wise separation* statistic integrates separation values over all complexes and clusters, and indicates the general correspondence between a clustering result and the set of annotated complexes. Separation is particularly relevant to assessing clustering algorithms like MCODE, which permit assigning a protein to multiple clusters. Under some particular parameter combinations, this program tends to yield highly redundant clusters. Table 5 shows a fragment of the contingency table indicating the number mutual intersections between the 607 clusters obtained from the unaltered test graph. For example, the 50 first rows/columns show a series of imbricated clusters, each resulting from the addition of one node to the preceding cluster. Such strongly overlapping clusters artificially increase the performance, since a set of clusters representing the same complex will be taken into account multiple times in the average *PPV*.

Cluster-wise separation penalizes this effect by using the marginal sums rather than the cluster size. Thus, if a method generates many redundant clusters, each one intersecting with a given complex, the marginal sum will increase drastically, and Sep_{cl} will be reduced accordingly. Note that the result of Table 5 is not representative of all MCODE conditions: when appropriate parameters are chosen, the level of mutual overlap between clusters is reasonable.

Figure 4B displays the impact of edge addition on clustering-wise separation. The general trends are similar to those revealed by the accuracy curves (Figure 4A), but the random expectation curves are now roughly horizontal for SPC as well as for the other algorithms. We defined a second set of parameters optimized for separation, in the same way as described above for accuracy. These separation-optimized parameters are displayed in Table 3 and were used for all separation curves in this robustness analysis (right panels in Figure 4).

In Figure 4C and 4D, increasing proportions (0%, 5%, 10%, 20%, 40%, and 80%) of edges are randomly removed from the test graph. The general trend is for RNSC and MCL to outperform the other two algorithms under most conditions. RNSC, however, shows a higher sensitivity to edge removal, and its performance strongly decreases when more than 40% of the edges are removed. SPC is quite robust to edge removal, but its performance remains lower than that of MCL under all conditions. Note that this removal experiment is not very indicative of algorithm capability under realistic conditions, because the partitioning of the test graph corresponds almost with complex composition (Figure 1A). Thus, when edges are simply removed, this partitioning is mostly maintained: given the high level of intra-complex connectivity, most complexes remain linked, and no new inter-complex link is created.

In order to obtain a realistic estimate of algorithm robustness, we thus need to combine edge addition and removal. Figure 4E and 4F shows the robustness to edge removal, starting from a graph with 100% edge addition. The performances of all programs are of course reduced as compared to Figures 4C and 4D. In terms of accuracy (Figure 4E), RNSC and MCL show grossly similar behaviours: the accuracy shows a good robustness in the low range of removal percentages (0–40%) but strongly decreases at higher percentages (80%). Yet in terms of separation (Figure 4F), RNSC shows a better performance than MCL at low rates of removal. The separation values of all algorithms drop to their respective levels of the random expectation when 80% of the edges are removed. MCODE and SPC show generally low performance, and are drastically affected by the combination of addition and removal. The performance of SPC is similar to that obtained by selecting random clusters, in terms of both accuracy (Figure 4E) and separation (Figure 4F).

Figures 4G and 4H show the effect of edge addition on graphs from which 40% of the edges had previously been removed. These curves confirm the trends observed in Figures 4A and 4B: MCL and RNSC are weakly affected by edge addition, but as little as 20% edge addition suffices to prevent SPC from identifying the complexes (Figure 4H). MCODE is relatively robust to edge addition, but shows a weaker performance than MCL and RNSC over the whole range of conditions.

Analysis of data sets obtained in high-throughput experiments

In the previous chapters our evaluations were based on artificial graphs obtained by adding and removing various proportions of edges to a reference network (the MIPS complexes). The next step was to evaluate the capability of these algorithms to extract relevant information from high-throughput data sets. To this end, we downloaded from the GRID database [39] six data sets representing the network of protein interactions in the yeast *Saccharomyces cerevisiae*. Two of these data sets consist of pairs of interacting proteins detected by the two-hybrid technique published respectively by Uetz *et al.* [1] and Ito *et al.* [2]. The four other data sets contain protein complexes characterized by mass spectrometry, published respectively by Gavin *et al.* [4,6], Ho *et al.* [5], and Krogan *et al.* [7] (Table 6). For each of these data sets we built a graph with one node per protein, and one edge per interaction.

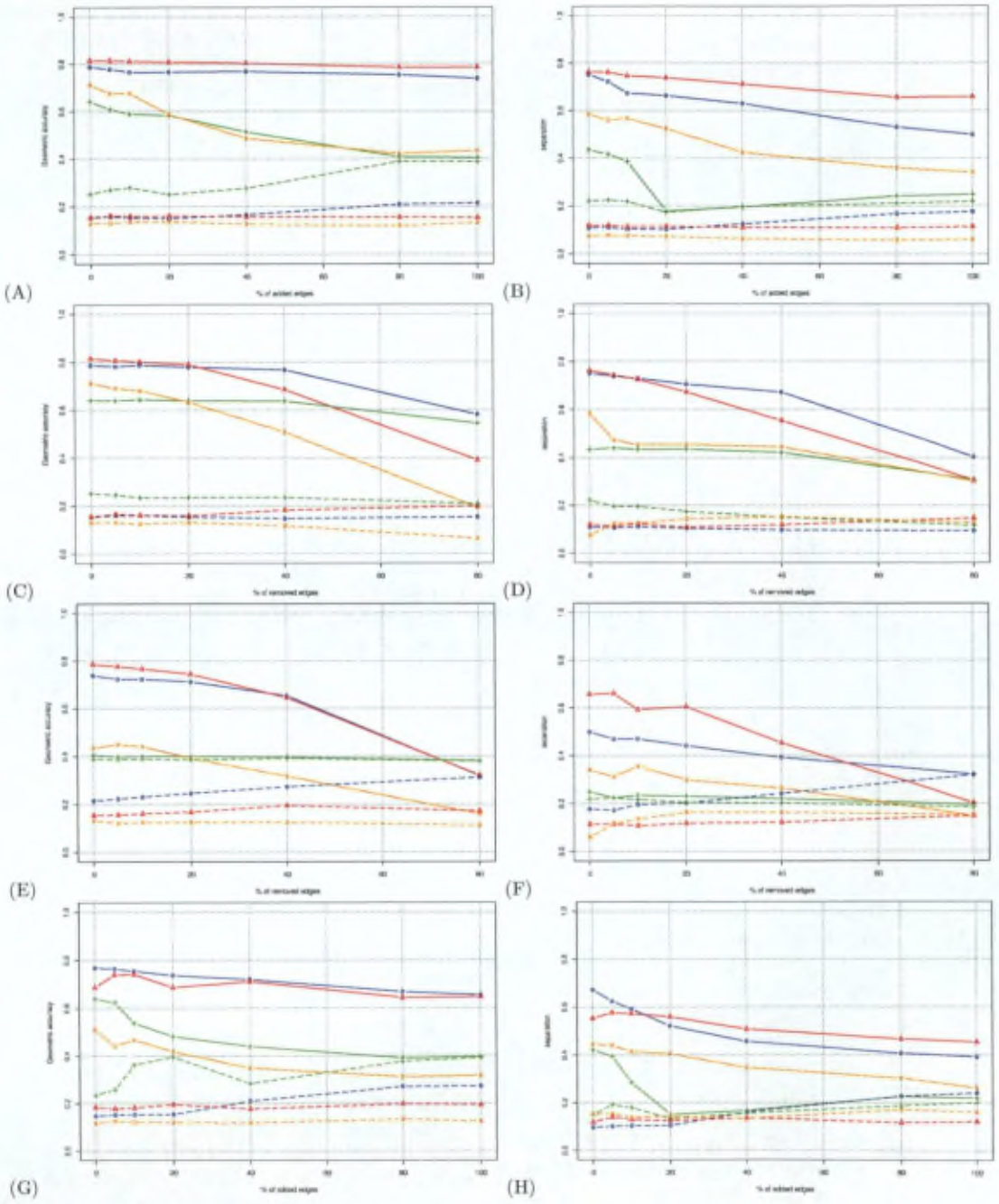


Figure 4

Robustness of the algorithms to random edge addition and removal. Each curve represents the value of accuracy (left panels) or separation (right panels). **(A-B)** edge addition to the test graph. **(C-D)** edges removal from the test graph. **(E-F)** Edge removal from an altered graph with 100% of randomly added edges. **(G-H)** Edge addition to an altered test graph with 40% of randomly removed edges. Color code: *blue* : MCL, *red* : RNSC, *orange* : MCODE, *green* : SPC. Dotted lines show the results obtained by permuting the clusters (negative control).

We then ran the four clustering algorithms on these graphs, with the optimal parameters determined in the first part of this study. The clusters obtained from these high-throughput networks were compared with the complexes annotated in the MIPS database by computing the same statistics as described above (Table 6, Figure 5). In each case, a negative control was done by calculating the same statistics on permuted clusters (shaded boxes in Figure 5).

Some precautions should be taken before interpreting these results. In particular, it is not trivial to interpret the "positive predictive value", as our reference set is the MIPS collection, filtered to discard any high-throughput result. This collection should by no means be considered exhaustive, since the complexes detected by previous studies represent only a fraction of all existing complexes. High-throughput methods are thus expected to yield many complexes that have not previously been characterized by other methods. Thus, interactions detected by high-throughput methods that are not annotated in MIPS cannot be considered "false positives". The same holds true for cluster-wise separation. Thus, the *PPV* and cluster-wise separation values should be interpreted as an indication of the fraction of high-throughput results which are also detected by other methods and have been annotated in the MIPS so far. In contrast, the sensitivity is likely to yield more directly relevant information, by indicating the fraction of annotated complexes recovered in the clusters obtained from high-throughput data. Bearing in mind these limitations, we may now analyse the data presented in Table 6 and Figure 5.

An important criterion for this analysis is the contrast between the scores reached with the real clustering results and the random expectation estimated with permuted clusters. A look at this contrast already reveals some general characteristics of the data sets. Whatever the clustering method used, the *Sep* values obtained were similar for the real and pemuted clusters in the case of the data sets resulting from two-hybrid experiments [1,2]. This confirms the conclusions of Von Mering *et al.*, who compared the positive predictive value (called "accuracy" in their paper) and sensitivity ("coverage") of the interaction graphs obtained by various high-throughput methods [8], and who also observed very poor values for the two-hybrid data sets. The HMS-PCI data set [5] shows a better contrast between real and permuted clusters, but the best results are clearly obtained with the three other mass-spectrometry data sets [4,6,7]. We will thus focus on these three data sets in our comparison of the algorithms.

Compared to the other algorithms, MCODE yields a lower number of clusters, with a higher number of proteins per cluster. It generally yields a moderate sensitivity, a low *PPV*. It is characterized by very weak cluster-wise separation (Figure 5A), contrasting with a very high complex-wise separation (Figure 5B). The resulting clustering-wise separation values (Figure 5C) are lower than for the other algorithms. Despite its relatively weak general performance, MCODE interestingly shows the best performances for the negative control (i.e. the lowest values). This reflects the fact that this algorithm has the capability to discard nodes from the clustering result (unassigned nodes). Apparently, this property enables the program to discard most nodes when a random graph is submitted, but this seems to be at the expense of sensitivity with real interaction graphs.

SPC is characterized by a high sensitivity and a low *PPV*. Yet the high sensitivity is an artifact due to the presence of one mega-cluster, generally accompanied by a multitude of mini-clusters. The asymmetry between cluster sizes is revealed by the differences between the mean and median numbers of proteins per cluster. For Gavin (2002), for example, SPC yields 87 clusters with a mean size of 15.5 but with a median of only 2 proteins per cluster. The mega-cluster includes no less than 1074 proteins and thus comprises most complexes, raising the sensitivity to Sn = 91.8%. The artificial aspect is indicated by the fact that the permuted clusters also reach a very high score Sn = 81.4%. As discussed above, this bias is avoided by the separation statistics, which yield relatively low values for SPC. Similar figures are obtained with the other data sets.

For all the data sets, RNSC yields a large number of mini-clusters (the average number of proteins per cluster is typically 2, the median is 1 or 2), plus a few clusters of reasonable size (up to 35 proteins per cluster). It shows a relatively high cluster-wise separation value (Figure 5A) but a lower complex-wise separation value (Figure 5B), resulting in a reasonable tradeoff in terms of clustering-wise separation (Figure 5C). The contrast between real and permuted clusters is low, however, even for the mass-spectrometry data with which other algorithms reach a good contrast.

Finally, MCL clearly outperforms the other algorithms in terms of general performance (Table 6, *Acc*, *Sep*) and also as regards the contrast between real and permuted clusters (Table 6, Figure 5). This general performance results from a good balance between cluster-wise (Figure 5A) and clustering-wise (Figure 5B) separation.

Conclusion

We have evaluated the capability of four graph-based clustering algorithms to extract protein complexes from networks of protein-protein interactions. This evaluation has led us to elaborate a testing procedure for the selection of optimal parameters and the analysis of robustness to

Table 4: Schematic illustration of a contingency table, and the derived statistics

Counts

T	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	sum	complex size
complex 1	7	0	0	0	0	7	7
complex 2	0	6	8	0	0	14	14
complex 3	0	0	0	14	3	17	20
complex 4	0	0	0	4	5	9	8
sum	7	6	8	18	8	47	
cluster size	7	6	8	16	8		

Positive Predictive Value (PPV)

PPV	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
complex 1	1	0	0	0	0
complex 2	0	1	1	0	0
complex 3	0	0	0	0.78	0.38
complex 4	0	0	0	0.22	0.62
cluster-wise PPV	1	1	1	0.78	0.62

Sensitivity

Sn	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	complex-wise Sn
complex 1	1	0	0	0	0	1
complex 2	0	0.43	0.57	0	0	0.57
complex 3	0	0	0	0.70	0.15	0.70
complex 4	0	0	0	0.50	0.62	0.62

Frequency per row

F_{row}	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
complex 1	1	0	0	0	0
complex 2	0	0.43	0.57	0	0
complex 3	0	0	0	0.82	0.18
complex 4	0	0	0	0.44	0.56

Separation

C	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	complex-wise separation
complex 1	1	0	0	0	0	1
complex 2	0	0.43	0.57	0	0	1
complex 3	0	0	0	0.64	0.07	0.71
complex 4	0	0	0	0.10	0.35	0.45
cluster-wise separation	1	0.43	0.57	0.74	0.41	

Clustering-wise sensitivity 0.69
Clustering-wise PPV 0.85
Accuracy 0.77
Average cluster-wise separation 0.63
Average complex-wise separation 0.79
Clustering-wise separation 0.70

Table 5: Mutually overlapping clusters obtained under some parameter conditions with MCODE

cluster\cluster	1	2	3	4	5	...	49	50	51	52	...	102	103	...	607
1	81	80	79	78	77	...	47	46	0	0	...	0	32	...	0
2	80	80	79	78	77	...	47	46	0	0	...	0	32	...	0
3	79	79	79	78	77	...	47	46	0	0	...	0	32	...	0
4	78	78	78	78	77	...	47	46	0	0	...	0	32	...	0
5	77	77	77	77	77	...	47	46	0	0	...	0	32	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
49	47	47	47	47	47	...	47	46	0	0	...	0	32	...	0
50	46	46	46	46	46	...	46	46	0	0	...	0	32	...	0
51	0	0	0	0	0	...	0	0	46	0	...	0	0	...	0
52	0	0	0	0	0	...	0	0	0	46	...	32	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
102	0	0	0	0	0	...	0	0	0	32	...	32	0	...	0
103	32	32	32	32	32	...	32	32	0	0	...	0	32	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
607	0	0	0	0	0	...	0	0	0	0	...	0	0	...	3

noise. We have defined new matching statistics called *separation* to circumvent some pitfalls of classical estimators (Sensitivity, PPV, Accuracy). The methodology proposed here could be used as well to assess the capabilities of other clustering algorithms with other data sets.

To study the ability of the tested algorithms to extract protein complexes from an interaction network, we built a test graph from the complexes annotated in the MIPS database.

In a first step we assessed the impact of the parameters of each algorithm, and determined the optimal values for extracting complexes from an interaction network. This analysis shows that under most conditions, RNSC and MCL outperform MCODE and SPC. RNSC is remarkably robust to variations in the choice of parameters, whereas the other algorithms require appropriate tuning in order to yield relevant results. Secondly we assessed the robustness of these programs to noise and to missing information in the data, by randomly adding and removing edges from the test graph. This analysis clearly revealed differences between the algorithms, highlighting the robustness of MCL, and to a lesser extent RNSC, to graph alterations.

We then applied the same four algorithms to interaction networks obtained from six high-throughput studies. This analysis revealed that whatever the algorithm used, some data sets provide insufficient information for extracting the correct protein complexes. An analysis of the more informative data sets confirmed the general superiority of MCL over the three other algorithms tested here.

An important limitation of the present evaluation is that it was performed by naive users. Any algorithm is likely to work better in the hands of its own developer than in those of external users. As we did not participate in the development of any of the tested algorithms, our evaluation may underestimate the capabilities of some of the algorithms tested here. An advantage of such an external evaluation, however, is that the evaluators are not biased by better knowledge of one particular algorithm. Consequently, our evaluation might be biased in favour of algorithms which are more user-friendly, or easier to configure. It thus reflects a compromise between algorithm user-friendliness and efficiency.

Another limitation is that all of our analyses were performed on unweighted graphs, because our reference graph (the MIPS complexes) does not contain any information that would enable us to assign reliability values (weights) to the edges. It should be mentioned that MCL and SPC can deal with weighted graphs and are likely to give better performances if the weights reflect the reliability of the links between proteins [20].

Methods

Test graphs

Annotated protein complexes

In order to test the ability of each algorithm to extract complexes from a network of binary interactions, we built a graph representing a large collection of experimentally characterized complexes. We collected from the MIPS database the collection of protein complexes annotated for the yeast *Saccharomyces cerevisiae* [40], from which we discarded those resulting from high-throughput experiments [41]. The filtered collection contains some cases of hierarchically related complexes. For example, the complex annotated as "ribosome" includes the small and large ribosomal subunits. In such cases, we discarded the parent complex (ribosome) and retained the sub-complexes only (small and large subunits). The final set comprises 220

Table 6: Main features of the four large scale data sets and clustering performances of the algorithms when applied to them

Dataset	Nb nodes	Nb edges	Mean degree	Mean clust coeff		MCL		MCODE		RNSC		SPC	
						real	permuted	real	permuted	real	permuted	real	permuted
Uetz et al. [1]	926	865	1.175	0.018	Number of clusters		288		10		48		234
					Mean nb prot/cluster		3.22		11.2		1.91		3.96
					Median nb prot/cluster		3		4.5		2		2
					Largest cluster size		16		53		6		276
					Sn	57.3%	38.6%	84.3%	74.5%	49.4%	36.5%	65.5%	43.3%
					PPV	53.8%	45.9%	25.5%	21.6%	59.6%	54.4%	38.0%	38.9%
					Acc_g	55.6%	42.3%	54.9%	48.0%	54.5%	45.5%	51.8%	41.1%
					Sep_{in}	23.0%	20.6%	48.9%	62.5%	15.5%	14.8%	19.1%	21.2%
					Sep_{ut}	30.1%	26.9%	2.2%	2.8%	34.3%	32.7%	20.3%	22.6%
					Sep	26.3%	23.5%	10.4%	13.3%	23.1%	22.0%	19.7%	21.9%
Ito et al. [2]	2937	4038	2.682	0.019	Number of clusters		630		9		1746		410
					Mean nb prot/cluster		4.66		97.8		1.68		7.16
					Median nb prot/cluster		3		11		2		2
					Largest cluster size		157		485		4		1928
					Sn	34.9%	26.0%	66.9%	68.0%	31.4%	24.0%	73.2%	64.6%
					PPV	42.7%	38.5%	8.2%	5.8%	63.6%	61.8%	24.3%	23.8%
					Acc_g	38.8%	32.2%	37.5%	36.9%	47.5%	42.9%	48.8%	44.2%
					Sep_{in}	12.7%	11.8%	41.6%	33.0%	7.1%	7.0%	11.3%	11.0%
					Sep_{ut}	36.2%	33.9%	1.7%	1.3%	56.7%	55.9%	20.1%	20.4%
					Sep	21.4%	20%	8.4%	6.7%	20.1%	19.8%	15.4%	15.0%
Ho et al. [5]	1564	3600	4.6	0.029	Number of clusters		314		13		957		63
					Mean nb prot/cluster		4.98		49.5		1.63		24.8
					Median nb prot/cluster		3		13		1		3
					Largest cluster size		34		432		8		1383
					Sn	50.6%	28.2%	81.2%	76.5%	37.0%	27.4%	90.1%	92.1%
					PPV	47.1%	35.6%	12.9%	8.5%	61.5%	57.1%	10.4%	8.2%
					Acc_g	48.9%	31.9%	47.1%	42.5%	49.3%	42.2%	50.2%	50.2%
					Sep_{in}	22.6%	19%	44.7%	37.2%	11%	10.5%	19.3%	13.8%
					Sep_{ut}	32.3%	27.1%	2.6%	2.2%	48%	45.6%	5.5%	4.0%
					Sep	27.0%	22.7%	10.9%	9.0%	23%	21.9%	10.3%	7.4%

Table 6: Main features of the four large scale data sets and clustering performances of the algorithms when applied to them *(Continued)*

Gavin et al. [4]	1352	3210	4.7	0.148	Number of clusters	212		27		709		87		
					Mean nb prot/cluster	6.38		32.5		1.91		15.5		
					Median nb prot/cluster	4		7		1		2		
					Largest cluster size	54		414		16		1074		
					Sn	74.1%	24.2%	67.0%	51.1%	52.1%	20.8%	91.8%	81.4%	
					PPV	57.0%	23.9%	20.4%	9.4%	62.0%	46.0%	18.1%	10.7%	
					Acc_s	65.6%	24.0%	43.7%	30.3%	57.1%	33.4%	54.9%	46.0%	
					Sep_{cc}	39.4%	17.6%	44.5%	16.1%	14.5%	11.3%	34.4%	15.7%	
					Sep_{cl}	38.0%	17.0%	5.5%	2.0%	46.9%	36.5%	13.6%	6.2%	
					Sep	38.7%	17.3%	15.6%	5.6%	26.1%	20.3%	21.6%	9.8%	
Gavin et al. [6]	1430	6531	9.1	0.348	Number of clusters	189		39		487		136		
					Mean nb prot/cluster	7.57		40.3		2.94		10.5		
					Median nb prot/cluster	4		9		2		3		
					Largest cluster size	90		697		35		620		
					Sn	75.7%	23.7%	58.3%	43.2%	60.8%	20.9%	79.8%	48.4%	
					PPV	54.3%	21.0%	20.6%	8.0%	63.3%	37.3%	37.0%	16.5%	
					Acc_s	65.0%	22.4%	39.5%	25.6%	62.1%	29.1%	58.4%	32.4%	
					Sep_{cc}	38.1%	15.5%	44.7%	15.3%	20.1%	12.9%	34.9%	14.9%	
					Sep_{cl}	32.7%	13.3%	7.9%	2.7%	44.5%	28.6%	21.6%	9.2%	
					Sep	35.3%	14.4%	18.8%	6.4%	29.9%	19.2%	27.4%	11.7%	
Krogan et al. [7]	~2675	7088	5.296	0.146	Number of clusters	813		70		1405		114		
					Mean nb prot/cluster	4.93		28.3		2.1		10.3		
					Median nb prot/cluster	3		5.5		2		3		
					Largest cluster size	50		387		21		1724		
					Sn	62.8%	19.8%	56.3%	30.9%	53.1%	19.1%	82.6%	64.0%	
					PPV	56.2%	33.5%	21.9%	9.7%	63.3%	51.1%	25.4%	17.2%	
					Acc_s	59.5%	26.7%	39.1%	20.3%	58.2%	35.1%	54.0%	40.6%	
					Sep_{cc}	20.0%	12.1%	33.2%	13.6%	10.3%	8.7%	20.3%	11.9%	
					Sep_{cl}	49.5%	29.9%	8.8%	3.6%	59.6%	50.3%	24.0%	14.1%	
					Sep	31.5%	19.0%	17.0%	7.0%	24.7%	21.6%	20.9%	12.9%	

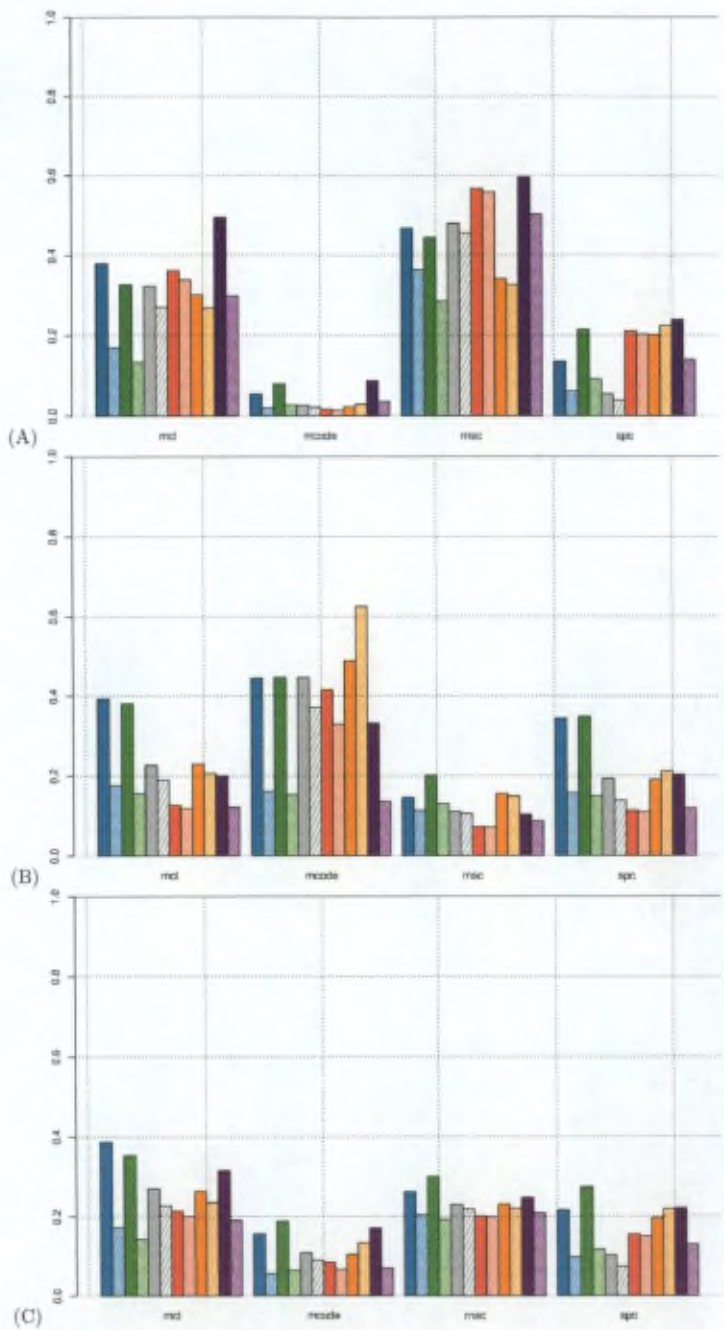


Figure 5
Application of clustering on high-throughput data sets. (A) Cluster-wise separation. **(B)** Complex-wise separation. **(C)** Clustering-wise separation. **Color code:** *blue* : mass-spectrometry data set from Gavin et al. (2002); *green* : mass-spectrometry data set from Gavin et al. (2006); *grey* : mass-spectrometry data set from Ho et al. (2002); *orange* : two-hybrid data set from Ito et al. (2001); *yellow* : two-hybrid data set from Uetz et al. (2000); *purple* : mas-spectrometry data set from Krogan et al. (2006). Shaded boxes show the results obtained by permuting the clusters (negative control).

complexes. It was converted to a graph where each node represents one polypeptide. A link (edge) was created between each pair of polypeptides involved in a common complex. The resulting graph (referred to as *test graph*) contains 1,095 polypeptides and 12,261 interactions (Figure 1A).

Altered graphs

A series of 41 altered graphs was derived from the test graph described above by combining various proportions of random edge deletions (0%, 5%, 10%, 20%, 40%, 80%) and additions (0%, 5%, 10%, 20%, 40%, 80%, 100%). We refer to altered graphs as $A_{add,del}$ where *add* and *del* indicate, respectively, the percentage of added and deleted edges.

Random expectation

The random expectation of clustering results was estimated in two ways: with randomized graphs and permuted clusters.

Randomized graphs

A *randomized graph* was obtained by shuffling all the edges of the test graph. This procedure preserves the connectivity of each node while reallocating edges at random. We also generated 41 *altered randomized graphs* by randomly adding edges to and deleting edges from the random graph in the same proportions as for the test graph. We refer to altered randomized graphs as $R_{add,del}$ where *add* and *del* indicate respectively the percentage of added and deleted edges.

Permuted clusters

A set of permuted clusters can be obtained from a clustering result by shuffling the associations between proteins and clusters. This randomization procedure preserves cluster sizes. We applied it to each clustering result obtained with the test graph and the altered graphs.

Matching statistics

Each clustering result was compared with the annotated complexes by building a contingency table, as schematically exemplified in Table 4. Having n complexes and m clusters, the contingency table T is a $n \cdot m$ matrix where row i corresponds to the i^{th} annotated complex, and column j to the j^{th} cluster. The value of a cell $T_{i,j}$ indicates the number of proteins found in common between complex i and cluster j. Note that some proteins belong to several complexes, and that with one one algorithm (MCODE), some proteins may be assigned to multiple clusters or not assigned to any cluster. The marginal sums (per row or per column) of the contingency table thus do not always correspond with complex or cluster sizes. For example, cluster 4 in Table 4 contains 16 proteins, but the sum of intersections between this cluster and all complexes is 18, because complexes 3 and 4 have 2 proteins in common. Complex 3 contains 20 proteins, but the sum of its intersections with clusters is 17, because 3 of its proteins are not assigned to any cluster. On the contrary, the fourth complex of Table 4 contains 8 proteins, but there are 9 assignations to clusters in all, because one protein is assigned to two separate clusters.

Sensitivity, positive predictive value (*PPV*), and accuracy are classically used to measure the correspondence between the result of a classification and a reference. We describe hereafter how these concepts can be adapted to measuring the match between a set of protein complexes and a clustering result. As discussed in the text, these statistics can in some particular cases lead to erroneous interpretations. We thus define an additional statistic, which we call *separation*.

Sensitivity

Considering the annotated complexes as our reference classification, we define *sensitivity* as the fraction of proteins of complex i which are found in cluster j.

$$Sn_{i,j} = T_{i,j}/N_i$$

In this formula, N_i is the number of proteins belonging to complex i. We also calculate a *complex-wise sensitivity* Sn_{co_i} as the maximal fraction of proteins of complex i assigned to the same cluster. Sn_{co_i} reflects the coverage of complex i by its best-matching cluster.

$$Sn_{co_i} = max_{j=1}^{m} Sn_{i,j}$$

To characterize the general sensitivity of a clustering result, we compute a *clustering-wise sensitivity* as the weighted average of Sn_{co_i} over all complexes.

$$Sn = \frac{\sum_{i=1}^{n} N_i Sn_{co_i}}{\sum_{i=1}^{n} N_i}$$

Positive predictive value

The positive predictive value is the proportion of members of cluster j which belong to complex i, relative to the total number of members of this cluster assigned to all complexes.

$$PPV_{i,j} = T_{i,j} / \sum_{i=1}^{n} T_{i,j} = T_{i,j} / T_{\cdot j}$$

$T_{\cdot j}$ is the marginal sum of a colum j. As exemplified by the fourth cluster in Table 4, this marginal sum can in some

cases differ from the cluster size, because some proteins can belong to several complexes. We also calculate a *cluster-wise positive predictive value* PPV_{cl_j}, which represents the maximal fraction of proteins of cluster j found in the same annotated complex. PPV_{cl_j} reflects the reliability with which cluster j predicts that a protein belongs to its best-matching complex.

$$PPV_{cl_j} = max_{i=1}^{n} PPV_{i,j}$$

To characterize the general PPV of a clustering result as a whole, we compute a *clustering-wise PPV* as the weighted average of PPV_{cl_j} over all clusters.

$$PPV = \frac{\sum_{j=1}^{m} T_{\cdot j} PPV_{cl_j}}{\sum_{j=1}^{m} T_{\cdot j}}$$

Accuracy

The *geometric accuracy* (*Acc*) indicates the tradeoff between sensitivity and predictive value. It is obtained by computing the geometrical geometrical mean of the *Sn* and the *PPV*.

$$Acc = \sqrt{Sn \cdot PPV}$$

The advantage of taking the geometric rather than arithmetic mean is that it yields a low score when either the *Sn* or the *PPV* metric is low. High accuracy values thus require a high performance for both criteria. It is of particular importance to use the geometric accuracy, as the arithmetic mean would give a false idea of quality in the trivial cases where all proteins are assigned to a single cluster ($Sn = 1 \Rightarrow Acc_{arithm} > 0.5$) or where, on the contrary, each protein is assigned to a single-element cluster ($PPV = 1 \Rightarrow Acc_{arithm} > 0.5$).

Separation

The contingency table indicates the absolute frequency of intersections between complexes and clusters. From these values, we derive relative frequencies with respect to the marginal sums, either per row ($F_{row_{i,j}}$) or per column ($F_{col_{i,j}}$).

$$F_{row_{i,j}} = \frac{T_{i,j}}{\sum_{j=1}^{m} T_{i,j}}$$

$$F_{col_{i,j}} = \frac{T_{i,j}}{\sum_{i=1}^{n} T_{i,j}} = PPV_{i,j}$$

Note that the frequency per column is identical to the PPV defined above. The frequency per row, on the contrary, can differ from the sensitivity for some algorithms, if the algorithm permits assigning a protein to multiple clusters (Table 4, complex 4), or leaving some proteins unassigned (Table 4, complex 3). In such cases, the frequency per row provides a more drastic criterion than the sensitivity defined above.

We define the *separation* as the product of column-wise and row-wise frequencies.

$$Sep_{i,j} = F_{col_{i,j}} \cdot F_{row_{i,j}}$$

The separation is comprised between 0 and 1. The maximal value $Sep_{i,j} = 1$ indicates a perfect and exclusive correspondence between complex j and cluster i: it indicates that the cluster contains all the members of the complex and only them (Table 4, complex 1 and cluster 1). In addition, the separation statistic deals efficiently with multiple assignations. It penalizes cases where proteins of a given complex are assigned to multiple clusters, by using row sums rather than complex sizes.

The *complex-wise separation* Sep_{co_i} is calculated as the sum of separation values for a given complex i.

$$Sep_{co_i} = \sum_{j=1}^{m} Sep_{i,j}$$

Reciprocally, we calculate a *cluster-wise separation*, which reflects the concentration of one or several complexes within a given cluster.

$$Sep_{cl_j} = \sum_{i=1}^{n} Sep_{i,j}$$

To estimate a clustering result as a whole, clustering-wise Sep_{co} and Sep_{cl} values are computed as the averages of Sep_{co_i} over all complexes, and of Sep_{cl_j} over all clusters, respectively.

$$Sep_{co} = \frac{\sum_{i=1}^{n} Sep_{co_i}}{n}$$

$$Sep_{cl} = \frac{\sum_{j=1}^{m} Sep_{cl_j}}{m}$$

We then compute the *geometrical separation* (*Sep*) as the geometrical mean of Sep_{co} and Sep_{cl}.

$$Sep = \sqrt{Sep_{co} \cdot Sep_{cl}}$$

Computation

Clustering was performed on a PC cluster of 40 nodes. Statistical treatments were done and figures made with the freeware statistical package *R* [42]. Graphical representations of the interaction networks were done with CytoScape, an open-source, platform-independent environment for the visualization and analysis of biological networks [43,44].

Graphs of protein interactions were manipulated using the Java classes developed by the aMAZE group [45,46].

Authors' contributions

SB collected the data, built the graphs, tested the algorithms, and ran the evaluation procedures. JvH conceived of the study and participated in its design, coordination, and in defining the matching statistics. Both authors were equally involved in writing the manuscript.

Additional material

Additional File 1

Supplementary information about the algorithms
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-7-488-S1.pdf]

Additional File 2

Optimal accuracy parameter values
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-7-488-S2.pdf]

Additional File 3

Optimal separation parameter values. These files and supplementary figures are also available on http://rsat.scmbb.ulb.ac.be/~sylvain/clustering_evaluation.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-7-488-S3.pdf]

Acknowledgements

Sylvain Brohée is the recipient of a PhD grant from the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA). The PC cluster used for the intensive computations was funded by the Belgian Fonds de la Recherche Fondamentale Collective and the Fonds d'Encouragement à la Recherche. We thank Raphaël Leplae and Marc Lensink for their valuable work to install and maintain this cluster. We thank Professor Bruno André for co-supervising this project.

References

1. Uetz P, Giot L, Cagney G, Mansfield TA, Judson RS, Knight JR, Lockshon D, Narayan V, Srinivasan M, Pochart P, Qureshi-Emili A, Li Y, Godwin B, Conover D, Kalbfleisch T, Vijayadamodar G, Yang M, Johnston M, Fields S, Rothberg JM: **A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.** *Nature* 2000, **403(6770):**623-7.
2. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y: **A comprehensive two-hybrid analysis to explore the yeast protein interactome.** *Proc Natl Acad Sci USA* 2001, **98(8):**4569-74.
3. Miller JP, Lo RS, Ben-Hur A, Desmarais C, Stagljar I, Noble WS, Fields S: **Large-scale identification of yeast integral membrane protein interactions.** *Proc Natl Acad Sci USA* 2005, **102(34):**12123-8.
4. Gavin AC, Bosche M, Krause R, Grandi P, Marzioch M, Bauer A, Schultz J, Rick JM, Michon AM, Cruciat CM, Remor M, fert 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 MA, Copley RR, 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* 2002, **415(6868):**141-7.
5. Ho Y, Gruhler A, Heilbut A, Bader GD, Moore L, Adams SL, Millar A, Taylor P, Bennett K, Boutilier K, Yang L, Wolting C, Donaldson I, Schandorff S, Shewnarane J, Vo M, Taggart J, Goudreault M, Muskat B, Alfarano C, Dewar D, Lin Z, Michalickova K, Willems AR, Sassi H, Nielsen PA, Rasmussen KJ, Andersen JR, Johansen LE, Hansen LH, Jespersen H, Podtelejnikov A, Nielsen E, Crawford J, Poulsen V, Sorensen BD, Matthiesen J, Hendrickson RC, Gleeson F, Pawson T, Moran MF, Durocher D, Mann M, Hogue CWV, Figeys D, Tyers M: **Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry.** *Nature* 2002, **415(6868):**180-3.
6. Gavin AC, Aloy P, Grandi P, Krause R, Boesche M, Marzioch M, Rau C, Jensen LJ, Bastuck S, Dmpelfeld B, Edelmann A, Heurtier MA, Hoffman V, Hoefert C, Klein K, Hudak M, Michon AM, Schelder M, Schirle M, Remor M, Rudi T, Hooper S, Bauer A, Bouwmeester T, Casari G, Drewes G, Neubauer G, Rick JM, Kuster B, Bork P, Russell RB, Superti-Furga G: **Proteome survey reveals modularity of the yeast cell machinery.** *Nature* 2006, **440(7084):**631-636.
7. Krogan NJ, Cagney G, Yu H, Zhong G, Guo X, Ignatchenko A, Li J, Pu S, Datta N, Tikuisis AP, Punna T, Peregrn-Alvarez JM, Shales M, Zhang X, Davey M, Robinson MD, Paccanaro A, Bray JE, Sheung A, Beattie B, Richards DP, Canadien V, Lalev A, Mena F, Wong P, Starostine A, Canete MM, Vlasblom J, Wu S, Orsi C, Collins SR, Chandran S, Haw R, Rilstone JJ, Gandi K, Thompson NJ, Musso G, Onge PS, Ghanny S, Lam MHY, Butland G, Altaf-Ul AM, Kanaya S, Shilatifard A, O'Shea E, Weissman JS, Ingles CJ, Hughes TR, Parkinson J, Gerstein M, Wodak SJ, Emili A, Greenblatt JF: **Global landscape of protein complexes in the yeast Saccharomyces cerevisiae.** *Nature* 2006, **440(7084):**637-643.
8. von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P: **Comparative assessment of large-scale data sets of protein-protein interactions.** *Nature* 2002, **417(6887):**399-403.
9. Sprinzak E, Sattath S, Margalit H: **How reliable are experimental protein-protein interaction data?** *J Mol Biol* 2003, **327(5):**919-23.
10. Jeong H, Mason SP, Barábási AL, Oltvai ZN: **Lethality and centrality in protein networks.** *Nature* 2001, **411(6833):**41-2.
11. Han JDJ, Dupuy D, Bertin N, Cusick ME, Vidal M: **Effect of sampling on topology predictions of protein-protein interaction networks.** *Nat Biotechnol* 2005, **23(7):**839-44.
12. Goldberg DS, Roth FP: **Assessing experimentally derived interactions in a small world.** *Proc Natl Acad Sci USA* 2003, **100(8):**4372-6.
13. Yook SH, Oltvai ZN, Barabási AL: **Functional and topological characterization of protein interaction networks.** *Proteomics* 2004, **4(4):**928-942.

14. Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabàsi AL: **Hierarchical organization of modularity in metabolic networks.** *Science* 2002, **297(5586)**:1551-1555.
15. Ravasz E, Barabàsi AL: **Hierarchical organization in complex networks.** *Phys Rev E Stat Nonlin Soft Matter Phys* 2003, **67(2 Pt 2)**:026112.
16. Poyatos JF, Hurst LD: **How biologically relevant are interaction-based modules in protein networks?** *Genome Biol* 2004, **5(11)**:R93.
17. Rives AW, Galitski T: **Modular organization of cellular networks.** *Proc Natl Acad Sci USA* 2003, **100(3)**:1128-33.
18. Spirin V, Mirny LA: **Protein complexes and functional modules in molecular networks.** *Proc Natl Acad Sci USA* 2003, **100(21)**:12123-8.
19. Bader GD, Hogue CWV: **An automated method for finding molecular complexes in large protein interaction networks.** *BMC Bioinformatics* 2003, **4**:2.
20. Pereira-Leal JB, Enright AJ, Ouzounis CA: **Detection of functional modules from protein interaction networks.** *Proteins* 2004, **54**:49-57.
21. King AD, Przulj N, Jurisica I: **Protein complex prediction via cost-based clustering.** *Bioinformatics* 2004, **20(17)**:3013-20.
22. Arnau V, Mars S, Marin I: **Iterative cluster analysis of protein interaction data.** *Bioinformatics* 2005, **21(3)**:364-78.
23. Lu H, Zhu X, Liu H, Skogerb G, Zhang J, Zhang Y, Cai L, Zhao Y, Sun S, Xu J, Bu D, Chen R: **The interactome as a tree – an attempt to visualize the protein-protein interaction network in yeast.** *Nucleic Acids Res* 2004, **32(16)**:4804-4811.
24. Brun C, Herrmann C, Gunoche A: **Clustering proteins from interaction networks for the prediction of cellular functions.** *BMC Bioinformatics* 2004, **5**:95.
25. Altaf-Ul-Amin M, Shinbo Y, Mihara K, Kurokawa K, Kanaya S: **Development and implementation of an algorithm for detection of protein complexes in large interaction networks.** *BMC Bioinformatics* 2006, **7**:207.
26. Said MR, Begley TJ, Oppenheim AV, Lauffenburger DA, Samson LD: **Global network analysis of phenotypic effects: protein networks and toxicity modulation in Saccharomyces cerevisiae.** *Proc Natl Acad Sci USA* 2004, **101(52)**:18006-18011.
27. Dunn R, Dudbridge F, Sanderson CM: **The use of edge-betweenness clustering to investigate biological function in protein interaction networks.** *BMC Bioinformatics* 2005, **6**:39.
28. Bandyopadhyay S, Sharan R, Ideker T: **Systematic identification of functional orthologs based on protein network comparison.** *Genome Res* 2006, **16(3)**:428-435.
29. Middendorf M, Ziv E, Wiggins CH: **Inferring network mechanisms: the Drosophila melanogaster protein interaction network.** *Proc Natl Acad Sci USA* 2005, **102(9)**:3192-3197.
30. Friedrich C, Schreiber F: **Visualisation and navigation methods for typed protein-protein interaction networks.** *Appl Bioinformatics* 2003, **2(3 Suppl)**:S19-S24.
31. Ding C, He X, Meraz RF, Holbrook SR: **A unified representation of multiprotein complex data for modeling interaction networks.** *Proteins* 2004, **57**:99-108.
32. Brun C, Chevenet F, Martin D, Wojcik J, Gunoche A, Jacq B: **Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network.** *Genome Biol* 2003, **5**:R6.
33. Vazquez A, Flammini A, Maritan A, Vespignani A: **Global protein function prediction from protein-protein interaction networks.** *Nat Biotechnol* 2003, **21(6)**:697-700.
34. Gagneur J, Krause R, Bouwmeester T, Casari G: **Modular decomposition of protein-protein interaction networks.** *Genome Biol* 2004, **5(8)**:R57.
35. Van Dongen S: **Graph clustering by flow simulation.** In *PhD thesis* Centers for mathematics and computer science (CWI), University of Utrecht; 2000.
36. Enright AJ, Dongen SV, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families.** *Nucleic Acids Res* 2002, **30(7)**:1575-84.
37. Blatt M, Wiseman S, Domany E: **Superparamagnetic clustering of data.** *Phys Rev Lett* 1996, **76(18)**:3251-3254.
38. Mewes HW, Amid C, Arnold R, Frishman D, Guldener U, Mannhaupt G, Munsterkotter M, Pagel P, Strack N, Stumpflen V, Warfsmann J, Ruepp A: **MIPS: analysis and annotation of proteins from whole genomes.** *Nucleic Acids Res* 2004:D41-4.
39. Breitkreutz BJ, Stark C, Tyers M: **The GRID: the General Repository for Interaction Datasets.** *Genome Biol* 2003, **4(3)**:R23.
40. Mewes HW, Amid C, Arnold R, Frishman D, Guldener U, Mannhaupt G, Munsterkotter M, Pagel P, Strack N, Stumpflen V, Warfsmann J, Ruepp A: **MIPS: analysis and annotation of proteins from whole genomes.** *Nucleic Acids Res* 2004:D41-4.
41. Simonis N, van Helden J, Cohen GN, Wodak SJ: **Transcriptional regulation of protein complexes in yeast.** *Genome Biol* 2004, **5(5)**:R33.
42. R Development Core Team: **R: A language and environment for statistical computing.** 2005 [http://www.R-project.org]. R Foundation for Statistical Computing, Vienna, Austria [ISBN 3-900051-07-0]
43. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13(11)**:2498-504.
44. **Cytoscape : Analyzing and Visualizing Biological Network Data** [http://www.cytoscape.org]
45. Lemer C, Antezana E, Couche F, Fays F, Santolaria X, Janky R, Deville Y, Richelle J, Wodak SJ: **The aMAZE LightBench: a web interface to a relational database of cellular processes.** *Nucleic Acids Res* 2004:D443-8.
46. **aMAZE Project** [http://www.scmbb.ulb.ac.be/amaze/]

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

BioMedcentral

7.3 Erratum

Suite à une erreur lors de l'analyse des données dans la deuxième partie de notre article, les résultats de *MCODE* appliqués au données à haut débit sont erronés. Cette erreur modifie quelque peu les conclusions de l'article : *MCODE* semble présenter de meilleurs performances que *RNSC* lorsqu'il est appliqué aux jeux de données à large spectre, les résultats de *MCL* et *SPC* restant cependant globalement supérieurs et inférieurs, respectivement (Figure 7.1, Tableau 7.1). *MCL* restant l'algorithme le plus performant dans notre approche et le travail subséquent n'en a donc pas été affecté.

TAB. 7.1 – Version corrigée de la Table 6 de l'article.

Dataset	Nb nodes	Nb edges	Mean degree	Mean clust coeff		MCL		MCODE		RNSC		SPC	
						real	permuted	real	permuted	real	permuted	real	permuted
Uetz *et al.* [31]	926	803	1.175	0.018	Number of clusters	288		10		48		234	
					Mean nb prot/cluster	3.22		3.4		1.91		3.96	
					Median nb prot/cluster	3		3		2		2	
					Largest cluster size	16		4		6		276	
					Sn	57.3%	38.6%	95.5%	54.5%	49.4%	36.5%	65.5%	43.3%
					PPV	53.8%	45.9%	40.9%	31.8%	59.6%	54.4%	38.0%	38.9%
					Acc_g	55.6%	42.1%	62.3%	41.7%	54.2%	44.6%	49.9%	41.0%
					Sep_{co}	23.0%	20.6%	43.3%	37.2%	15.5%	14.8%	19.1%	21.2%
					Sep_{cl}	80.1%	26.9%	2%	1.7%	34.3%	32.7%	20.3%	22.6%
					Sep	26.3%	23.5%	9.2%	7.9%	23.1%	22.0%	19.7%	21.9%
Ito *et al.* [32]	2937	4038	2.682	0.019	Number of clusters	630		20		1746		410	
					Mean nb prot/cluster	4.66		3.8		1.68		7.16	
					Median nb prot/cluster	3		3		2		2	
					Largest cluster size	157		9		4		1928	
					Sn	34.9%	26.0%	100%	70%	31.4%	24.0%	73.2%	64.6%
					PPV	42.7%	38.5%	75%	75%	63.6%	61.8%	24.3%	23.8%
					Acc_g	38.6%	32.6%	86.6%	72.4%	44.7%	38.5%	42.2%	39.2%
					Sep_{co}	12.7%	11.8%	40%	35.2%	7.1%	7.0%	11.3%	11.0%
					Sep_{cl}	36.2%	33.0%	3.6%	3.2%	56.7%	55.9%	20.1%	20.4%
					Sep	21.4%	20%	8.4%	10.6%	20.1%	19.8%	15.4%	15.0%
Ito *et al.* [34]	1564	3681	4.6	0.029	Number of clusters	314		18		957		63	
					Mean nb prot/cluster	5.94		49.5		1.63		24.8	
					Median nb prot/cluster	3.5		13		1		3	
					Largest cluster size	36		432		8		1383	
					Sn	50.6%	28.2%	91.8%	70.7%	37.0%	27.4%	90.1%	92.1%
					PPV	47.1%	35.6%	41.8%	29.3%	61.5%	57.1%	10.4%	8.2%
					Acc_g	48.8%	31.7%	61.6%	45.5%	47.7%	39.6%	30.6%	27.5%
					Sep_{co}	22.6%	19%	48.0%	40%	11%	10.5%	19.3%	13.8%
					Sep_{cl}	32.3%	27.1%	4%	3.8%	48%	45.6%	5.5%	4.0%
					Sep	27.0%	22.7%	13.9%	13.2%	23%	21.9%	10.3%	7.4%
Gavin *et al.* [35]	1352	3210	4.7	0.148	Number of clusters	232		42		709		87	
					Mean nb prot/cluster	6.38		3.9		1.91		15.5	
					Median nb prot/cluster	4		3		1		2	
					Largest cluster size	54		7		16		1074	
					Sn	74.1%	24.2%	92.6%	42.6%	52.1%	20.8%	91.8%	81.4%
					PPV	57.0%	23.8%	60.1%	29.1%	62.0%	40.0%	18.1%	10.7%
					Acc_g	65%	24.0%	74.6%	35.2%	56.8%	30.93%	40.8%	29.51%
					Sep_{co}	39.4%	17.6%	69.1%	37%	14.5%	11.3%	34.4%	15.7%
					Sep_{cl}	38.0%	17.0%	13.2%	7.1%	46.9%	36.5%	13.6%	6.2%
					Sep	38.7%	17.3%	30.2%	16.2%	26.1%	20.3%	21.6%	9.8%
Gavin *et al.* [36]	1430	6531	9.1	0.348	Number of clusters	189		85		487		130	
					Mean nb prot/cluster	7.57		4.6		2.94		10.5	
					Median nb prot/cluster	4		4		2		3	
					Largest cluster size	90		14		35		620	
					Sn	75.7%	23.7%	82.0%	29.9%	60.8%	20.9%	79.8%	48.4%
					PPV	54.3%	21.0%	61.0%	25.9%	63.3%	37.3%	57.0%	36.5%
					Acc_g	64.1%	22.3%	70.7%	27.8%	62.0%	27.9%	54.3%	28.2%
					Sep_{co}	38.1%	15.5%	57.1%	24.0%	20.1%	12.9%	34.9%	14.9%
					Sep_{cl}	32.7%	13.3%	22.1%	9.5%	44.5%	28.6%	21.6%	9.2%
					Sep	35.3%	14.4%	35.5%	15.3%	29.9%	19.2%	27.4%	11.7%
Krogan *et al.* [37]	2675	7088	5.296	0.146	Number of clusters	813		81		1405		114	
					Mean nb prot/cluster	4.93		5		2.1		10.3	
					Median nb prot/cluster	3		4		2		3	
					Largest cluster size	50		28		21		1724	
					Sn	62.8%	19.8%	79.5%	28.9%	53.1%	19.1%	82.6%	64.0%
					PPV	56.2%	33.5%	62.8%	26.2%	63.3%	51.1%	25.4%	17.2%
					Acc_g	59.4%	25.7%	70.7%	27.5%	58%	31.2%	45.8%	33.17%
					Sep_{co}	20.0%	12.1%	42.4%	22.5%	10.3%	8.7%	20.3%	11.9%
					Sep_{cl}	49.5%	29.9%	15.5%	8.3%	59.6%	50.3%	24.0%	14.1%
					Sep	31.5%	19.0%	25.7%	13.6%	24.7%	21.6%	20.9%	12.9%

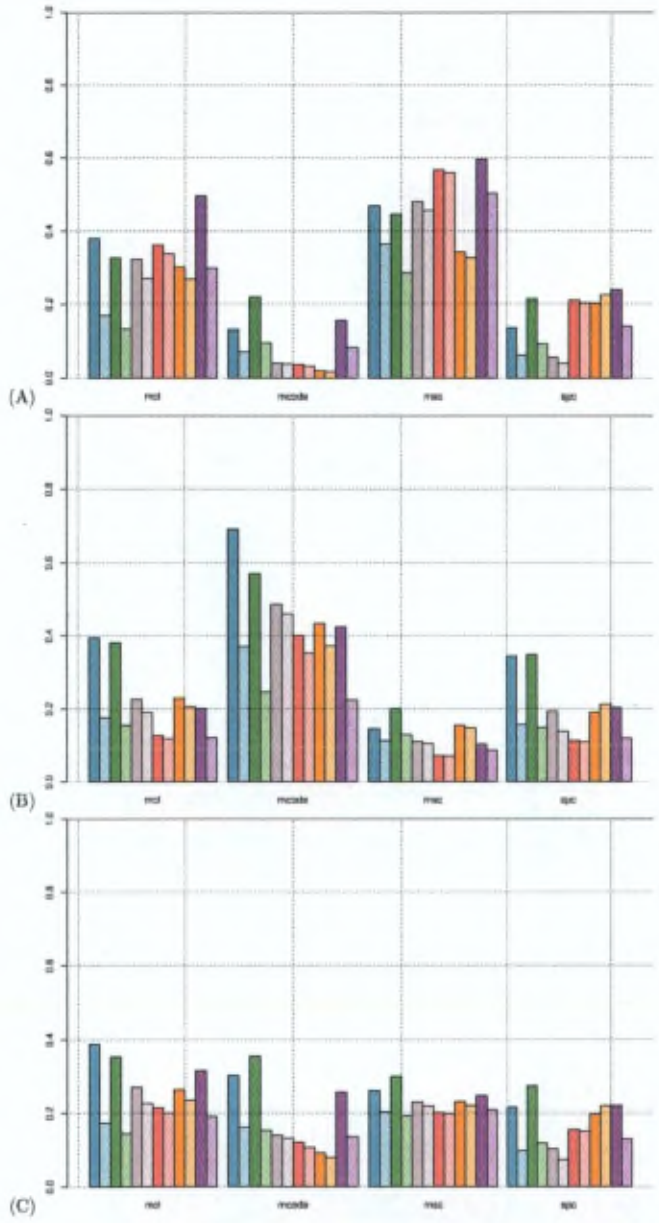


FIG. 7.1 – Version corrigée de la Figure 5 de notre article en fonction des résultats de la Table 7.1. Application des méthodes de clustering sur les jeux de données à grande échelle. (A) Séparation au niveau des clusters. (B) Séparation au niveau des complexes (C) Séparation globale. Code couleur bleu : jeu de données de spectrométrie de masse de Gavin *et al* (2002) ; vert :jeu de données de spectrométrie de masse de Gavin *et al* (2006) ; grey : jeu de données de spectrométrie de masse de Ho *et al* (2002) ; orange : jeu de données de double hybride de Ito *et al* (2001) ; yellow : jeu de données de double hybride de Uetz *et al* (2000) ; purple : jeu de données de spectrométrie de masse de Krogan *et al* (2006). Les histogrames hachurés montrent les résultats obtenus pour le contrôle négatif (ré-échantillonage aléatoire des clusters).

Chapitre 8

Etudes de données expérimentales d'interactions entre transporteurs membranaires

8.1 Introduction

Comme nous l'avons évoqué dans l'Introduction, en raison d'une plus grande difficulté de manipulation, les données concernant les perméases (ainsi que les autres protéines membranaires) sont assez lacunaires en comparaison avec les informations concernant les protéines solubles.

Le projet Associoport, financé par l'U.E. et décrit dans notre Introduction, a été donc monté pour tenter de combler ces lacunes. Ce projet consiste en une étude systématique des interactions entre transporteurs membranaires au moyen de la technique du double hybride split-ubiquitin.

Les professeurs Bruno André et Eckhard Boles nous ont fourni des fichiers contenant les valeurs de densités optiques de 200 souches exprimant une protéine fusionnée à Cub et croisées avec les souches fusionnées à Nub. En outre, afin de terminer la matrice d'interaction, nous nous sommes rendus dans le laboratoire du professeur Bruno André et avons réalisé les expériences de split-ubiquitin pour caractériser les interactions de quarante protéines supplémentaires. La première étape à consisté à intégrer l'ensemble de ces données dans une banque de données de façon à avoir à notre disposition les valeurs de densité optique de 240 souches exprimant une protéine fusionnée à Cub croisées systématiquement avec 279 souches fusionnées à Nub sur trois milieux (sélectif (YNB seul), permissif (YNB, adénine et histidine) et répressif (YNB et méthionine)) et suivies pendant trois jours (Figure 8.1).

La Figure 8.1 représente le suivi sur les trois milieux de la souche exprimant la protéine Yea4p fusionnée à Cub et croisée avec 93 autres protéines sur une plaque de 96 puits. Chaque plaque présente trois témoins. Les deux premiers sont des témoins négatifs : un puits contenant du milieu non ensemencé (Empty Well control) qui permet de vérifier une éventuelle contamination de la plaque et un puits contenant une souche n'exprimant que Nub non fusionnée à une autre protéine (Empty vector control) qui permet de vérifier que la protéine de fusion n'est pas capable de lier Nub à elle seule et d'ainsi activer la transcription des gènes de sélection. Le troisime contrôle est un contrôle positif où est testée l'interaction entre les protéines Ssy5p et Ptr3p déjà décrite dans la littérature [157].

8.2 Procédure bioinformatique

8.2.1 Estimation de la qualité des données

Dans une première étape, nous avons voulu évaluer la qualité globale des données de croissance mises à notre disposition. Pour cela, nous avons étudié les distributions de croissances des témoins positif et négatifs ainsi que des différentes souches diploïdes testées.

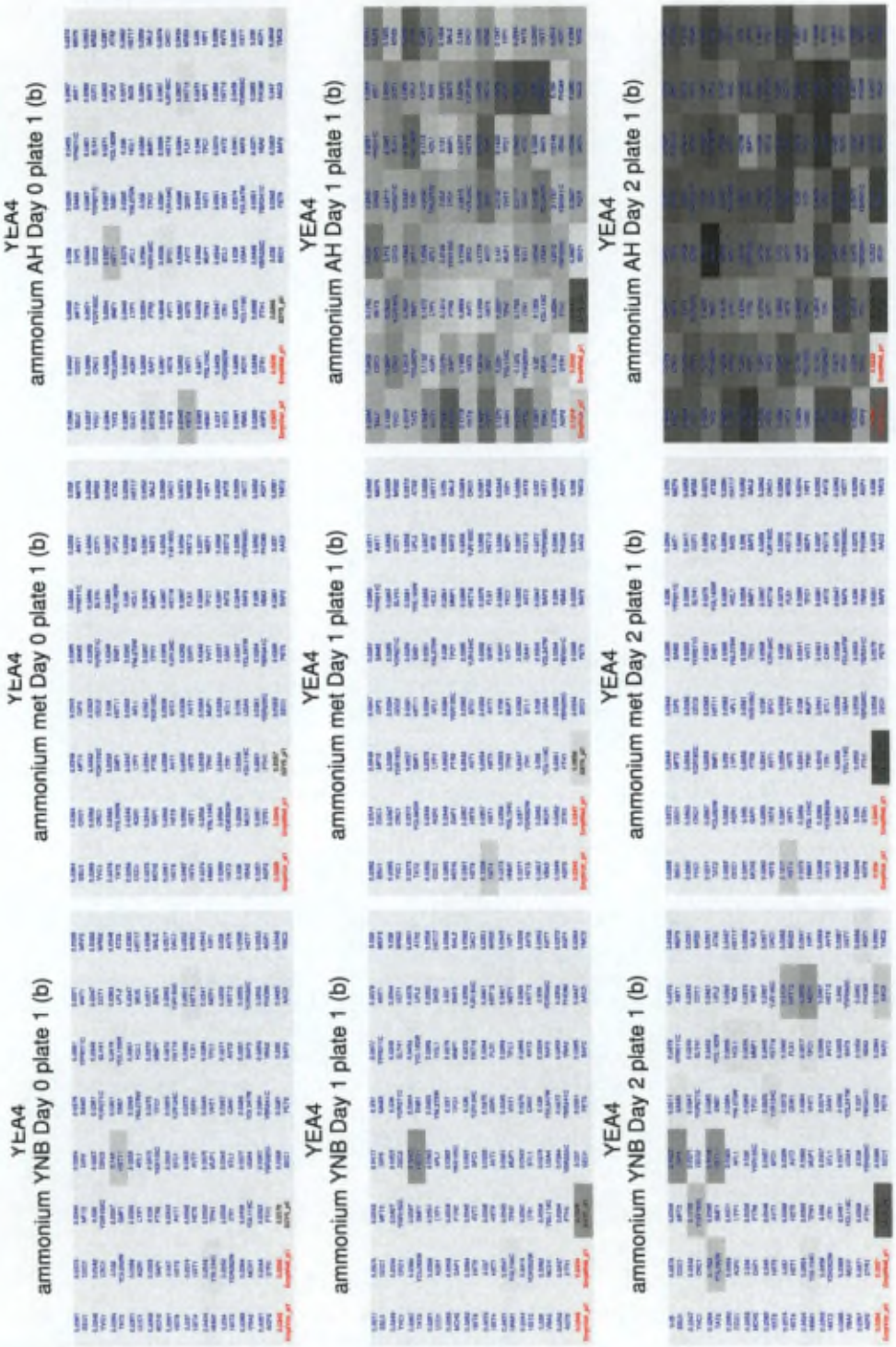


FIG. 8.1 – Représentation du suivi de culture sur trois milieux (colonnes respectivement : (1) sélectif : YNB seulement, (2) répressif : met (methionine) et (3) permissif : AH (adénine histidine)) de la souche exprimant la protéine Yea4p fusionnée à la partie carboxy-terminale de l'ubiquitine et croisée avec 93 souches exprimant différentes protéines fusionnées à la partie amino-terminale de l'ubiquitine. Les cultures sont suivies sur deux jours (lignes : respectivement jour 0, jour 1 et jour 2). Chaque plaque présente trois témoins (trois premiers puits de la dernière ligne de chaque plaque). Un puits contenant contenant une souche n'exprimant que Nub non fusionné (1er puits), un puits non ensemencé (2ème puits) et et un puits où est testée l'interaction annotée entre Ssy5p et Ptr3p (3ème puits).

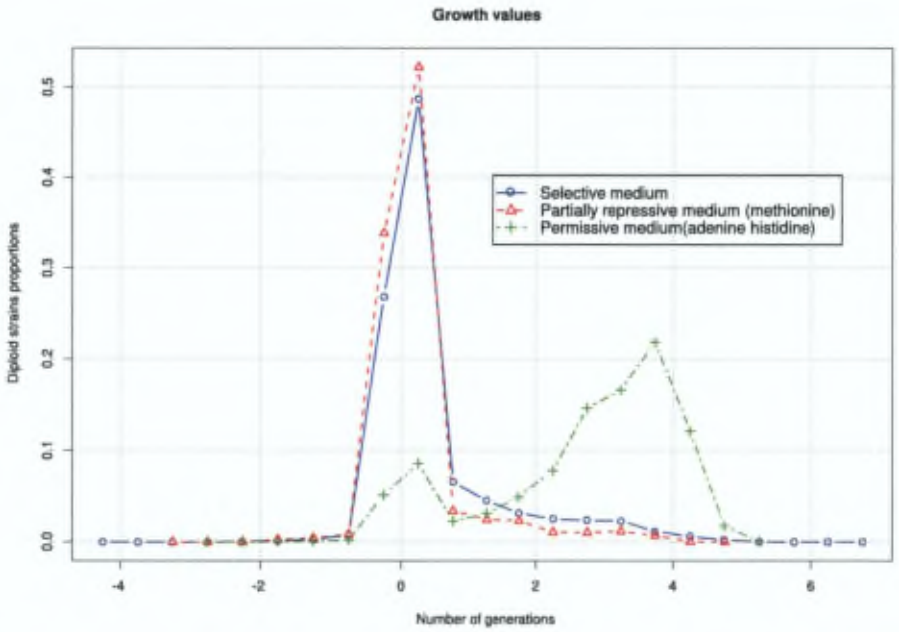


FIG. 8.2 – Distribution des fréquences du fombre de générations entre le jour 0 et le jour 1 pour chacune des souches diploïdes cultivées et sur chacun des milieux.

Afin d'estimer la croissance d'une souche donnée, nous avons calculé le nombre de générations G entre le jour 0 et les jours 1 et 2, pour chacune des souches diploïdes issues du croisement entre les souches exprimant Cub et Nub et fusionnées à deux autres protéines. Pour cela, nous calculons le logarithme du quotient de la densité optique mesurée au jour i (DO_i) par la densité optique mesurée au jour 0 DO_0.

$$G_i = log_2 \frac{DO_i}{DO_0}$$

La densité optique doublant à chaque génération, le logarithme en base 2 indique le nombre de fois que la population a doublé. Notons que les valeurs obtenues par ce calcul ne sont pas entières car elles reflètent le nombre moyen de générations pour une population donnée dans un puits de culture.

8.2.1.1 Distribution de croissances des souches testées

La Figure 8.2 reprend la distribution du nombre de générations entre le jour 0 et le jour 2 pour l'ensemble des souches diploïdes cultivées et sur chacun des milieux testés. Sur le milieu permissif (adénine - histidine), on observe comme attendu une croissance pour la quasi totalité des expériences même si aucune croissance n'a été détectée dans une minorité. Cette absence de croissance sur le milieu permissif pourrait être due un raté lors de l'ensemencement du puits ou lors des phases préparatoires de l'expérience (conjugaison des souches haploïdes par exemple).

Sur les milieux sélectif (YNB seulement) et partiellement répressif (méthionine), aucune croissance n'a été détectée dans la grande majorité des cas. En effet, sur toutes les interactions potentielles testées, on s'attendait à ce qu'il n'en existe réellement qu'un petit nombre. Par ailleurs, le nombre de diploïdes présentant une croissance est légèrement plus élevé sur le milieu sélectif que sur le milieu partiellement répressif.

Enfin, nous pouvons observer un petit nombre de valeurs négatives indiquant des densités optiques moindres au deuxième jour qu'au jour où les puits ont été ensemencés. Ces valeurs sont très vraisemblablement artéfactuelles (par exemple, erreur de lecture du spectrophotomètre) car il est difficilement concevable que des levures aient disparu du

Nombre interactions par protéine liée à Cub	Milieux considérés
inférieur au premier quartile de la fréquence d'interaction par protéine Cub	YNB jours 1 et 2, YNB + méthionine, jours 1 et 2
inférieur à la médiane de la fréquence d'interaction par protéine Cub	YNB jour 1, YNB + méthionine jours 1 et 2
inférieur au troisième quartile de la fréquence d'interaction par protéine Cub	méthionine jours 1 et 2
supérieur au troisième quartile de la fréquence d'interaction par protéine Cub	méthionine jour 1

TAB. 8.1 – Jeu de données considérés en fonction du nombre d'interaction par protéine Cub

puits en cours d'expérience. Dans la suite, comme nous ne sommes intéressés qu'aux souches diploïdes présentant une croissance importante, ces cas n'ont pas été retenus.

8.2.1.2 Distribution de croissances des témoins

Suite à cette analyse globale de la croissance sur différentes milieux, nous avons contrôlé le comportement des témoins présents sur l'ensemble des plaques de 96 puits (deux témoins négatifs et un témoin positif).

La Figure 8.3a montre que les souches servant de témoins positifs, qui expriment les protéines Ssy5-Nub et Ptr3-Cub poussent dans la quasi totalité des expériences. D'autre part, lorsque Nub est exprimé non fusionné à une autre protéine, on n'observe une croissance que sur le milieu non sélectif (Figure 8.3b). A la Figure 8.3c, on peut observer que lorsque le milieu n'a pas été inoculé, presque aucune croissance n'est détectable, mettant ainsi en évidence la bonne qualité des résultats..

8.2.2 Traitement statistiques des données

La Figure 8.4 présente de manière synthétique l'intensité de la croissance dans l'ensemble des puits entre les jours 0 et 1 en fonction des gènes fusionnés à Cub (lignes) et à Nub (colonnes) sur le milieu sélectif (YNB). Le niveau de gris est proportionnel aux nombre de générations : les points blancs indiquent donc une croissance nulle tandis que les points noirs dénotent une forte croissance. Les trois flèches sur le bord gauche de la figure indiquent la position des contrôles positif (une ligne noire) et négatifs (deux lignes blanches) contenus sur chaque plaque.

La présence de lignes noires verticales et horizontales suggère que certaines protéines interagiraient avec presque toutes les autres, ce qui semble fortement improbable (on parle parfois de *protéines collantes* ou *sticky proteins*). Ces cas sont vraisemblablement artéfactuels : plaques contaminées, erreurs dans la mesure des densités optiques pour certaines plaques à puits (colonnes noires). En outre, en raison de leur fusion avec Cub-PLV ou Nub, les protéines pourraient ne pas être sécrétées correctement et rester bloquées dans l'appareil de Golgi ou le réticulum endoplasmique ce qui faciliterait éventuellement des interactions transitoires non spécifiques.

Afin de tenir compte de ces différents effets et de ne sélectionner que les interactions significatives, nous avons recouru aux processus de standardisation et de filtrage. Le nombre de valeurs étant très important, il est essentiel de déterminer un seuil à partir duquel on considère qu'il y a réellement eu croissance dans un puits.

Pour comparer différents échantillons de données présentant des échelles variables, on peut leur appliquer un processus de standardisation. Celui-ci est effectué en soustrayant à chaque valeur de l'échantillon la valeur d'un estimateur de la tendance centrale et en le divisant par un estimateur de la dispersion. Dans notre cas, nous avons standardisé indépendamment en fonction de trois niveaux alternatifs de regroupement : (1) par plaque de 96 puits (tiers de colonne de la Figure 8.4), (2) par laboratoire et (3) par gène fusionné à Nub (ligne de la Figure 8.4))

Bien qu'il s'agisse des estimateurs les plus couramment utilisés, nous n'avons pas utilisé la moyenne et l'écart-type comme estimateurs mais la médiane et l'espace interquartile (IQR) qui sont plus robustes car moins sensibles aux valeurs aberrantes. Un facteur de correction doit cependant être utilisé car dans une distribution standard, la valeur de l'espace interquartile est d'environ 1,34, alors que celle de l'écart-type est de 1.

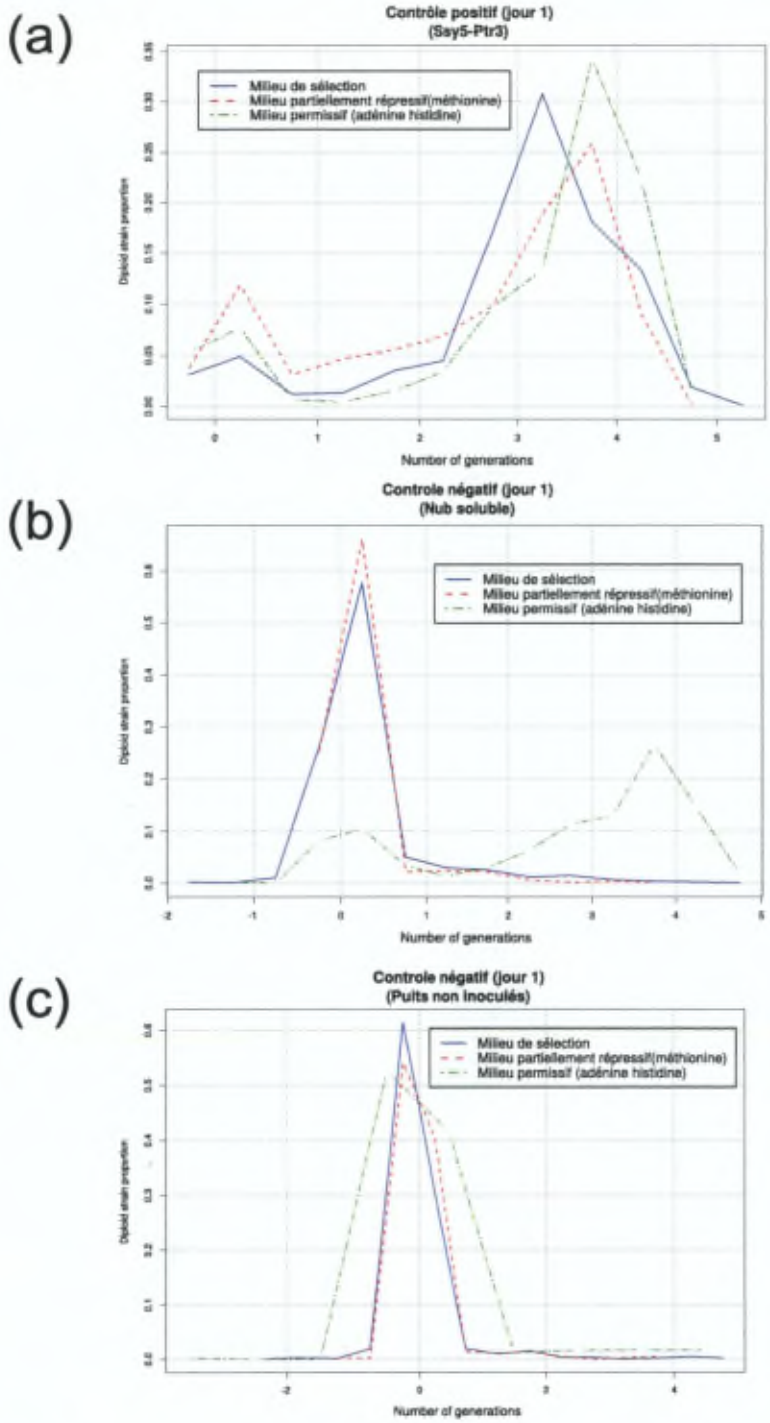


FIG. 8.3 – Distribution des valeur de croissances des témoins présents sur chaque plaque (a) Témoins positifs (b) Témoins négatif (Vecteur vide) (c) Témoins négatifs (Puits non inoculé).

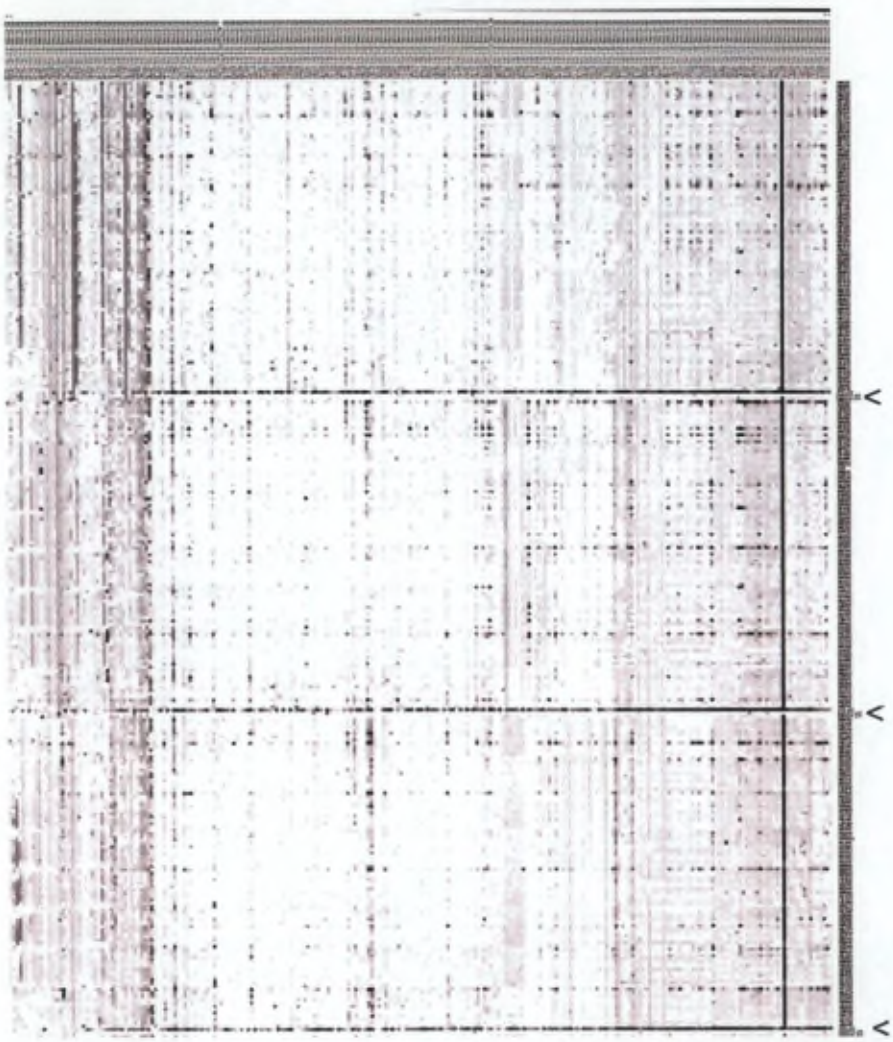


FIG. 8.4 – Représentation du nombre de générations entre le jour 0 et le jour 1 pour chacun des croisements réalisés entre deux souches haploïdes exprimant une protéine fusionnée à Cub (colonnes) et une souche haploïde exprimant une protéine fusionnée à Nub (lignes). Chaque colonne représente trois plaque de 96 puits. L'intensité du gris est proportionnelle à l'intensité de la croissance de la souche diploïde. Les flèches indiquent le témoin positif Ssy5p-Ptr3p suivi des deux témoins négatifs.

$$Z = \frac{x - \hat{m}}{\hat{\sigma}} = \frac{G_i - mediane(distribution)}{\frac{IQR(distribution)}{1,34}}$$

Le processus de standardisation retourne un Z-score correspondant à la valeur de croissance standardisée. Les valeurs proches ou inférieures à 0 indiquent une absence de croissance ou une croissance proche de la valeur médiane de croissance dans le groupe de normalisation (pas de croissance ou croissance observée dans tous les cas), alors que les valeurs élevées indiquent une croissance plus rapide. A chaque Z-score, peut être associée une P-valeur correspondant à la probabilité qu'il existe par hasard une valeur plus grande ou égale à ce Z-score. La P-valeur calcule donc la probabilité qu'une expérience au sein d'un puits donné soit un faux positif et plus elle est faible, moins on a de chances de se tromper en affirmant qu'il y a effectivement eu une croissance dans celui-ci. Cependant, même si l'on choisissait un seuil présentant une P-valeur de 0,01, on s'attendrait à retrouver 900 faux positifs par le jeu du hasard puisque l'on considère environ 90 000 expériences. Nous avons donc corrigé cet effet du test multiple en calculant la E-valeur.

$$E - valeur = P - valeur * N$$

La E-valeur représente le nombre de faux positifs attendus dans une batterie de N tests. Nous avons sélectionné tous les puits présentant une E-valeur inférieure ou égale à 1. Avec un tel seuil, on s'attend donc à avoir en moyenne un faux positif pour l'ensemble des 90 000 mesures. En appliquant ce procédé au quatre jeux de données (croissance sur milieu YNB avec ou sans méthionine jour 1 et jour 2), nous obtenons finalement environ 6000 interactions uniques statistiquement significatives.

Pour augmenter encore la fiabilité des interactions détectées, nous n'avons considéré que les interactions de certains jeux de données selon le nombre d'interactions détectées par protéine liée à Cub. Par exemple, dans le cas où une protéine ne présente que peu d'interactions, nous avons considéré pour cette protéine les interactions provenant des quatre jeux de données (YNB avec ou sans méthionine au jours 1 et 2). A l'inverse, lorsque qu'une protéine présente un grand nombre de partenaires d'interactions, nous n'avons considéré que les interactions significatives détectées sur méthionine au jour 1 (Tableau 8.1).

Cette seconde étape de filtrage nous a permis de conserver 3052 interactions significatives (Figure 8.5a) entre l'ensemble des 279 perméases étudiées.

8.2.3 Analyse du réseau d'interactions

8.2.3.1 Topologie du réseau et premiers résultats

Topologie du réseau Dans notre jeu de données, chaque perméase interagit en moyenne avec 21 protéines (minimum 2, maximum 52). Ce nombre semble excessif car il signifie qu'en moyenne une protéine interagit avec un peu moins de 10% des autres perméases. 31 protéines interagissent avec elles-mêmes et 75 interactions sont réciproques (l'interaction est détectée que les protéines soient fusionnées à Cub et Nub ou invérsement à Nub et à Cub) (Table 8.2). La présence des ces seules 75 interactions réciproques est elle aussi assez surprenante. En effet, les interactions étant testées pour la grande majorité dans les deux sens, nous nous serions attendu à ce que le nombre d'interactions significatives réciproques soit bien supérieur. De plus, parmi ces interactions, seules quelques unes présentent semblent impliqué dans le transport du même type de substrat et pourraient correspondre à une réalité biologique (ligne grisées, Table 8.2).

La figure 8.6 représente la distribution des degrés dans notre réseau, nous pouvons observer qu'il n'est pas de nature *power law* comme c'est habituellement le cas pour les réseaux d'interactions.

Premiers résultats intéressants La figure 8.5(a) représente le réseau d'interactions complet entre les perméases et les couleurs des arêtes indiquent si celles-ci relient deux arêtes annotées dans la même famille de la base de données *YTPdb* (arêtes intra-classes colorées) ou non (arêtes inter-classes grises), en ne considérant que les familles terminales de la hiérarchie. En raison du nombre important d'arêtes, ce réseau semble assez difficile à aborder, c'est pourquoi à la Figure 8.5(b), nous avons représenté le même réseau en supprimant toutes les arêtes grises inter-classes. Avec ce nouveau graphe, nous pouvons observer à l'oeil quelques groupes intéressants qui confirment des données connues ou proposent des pistes de recherches.

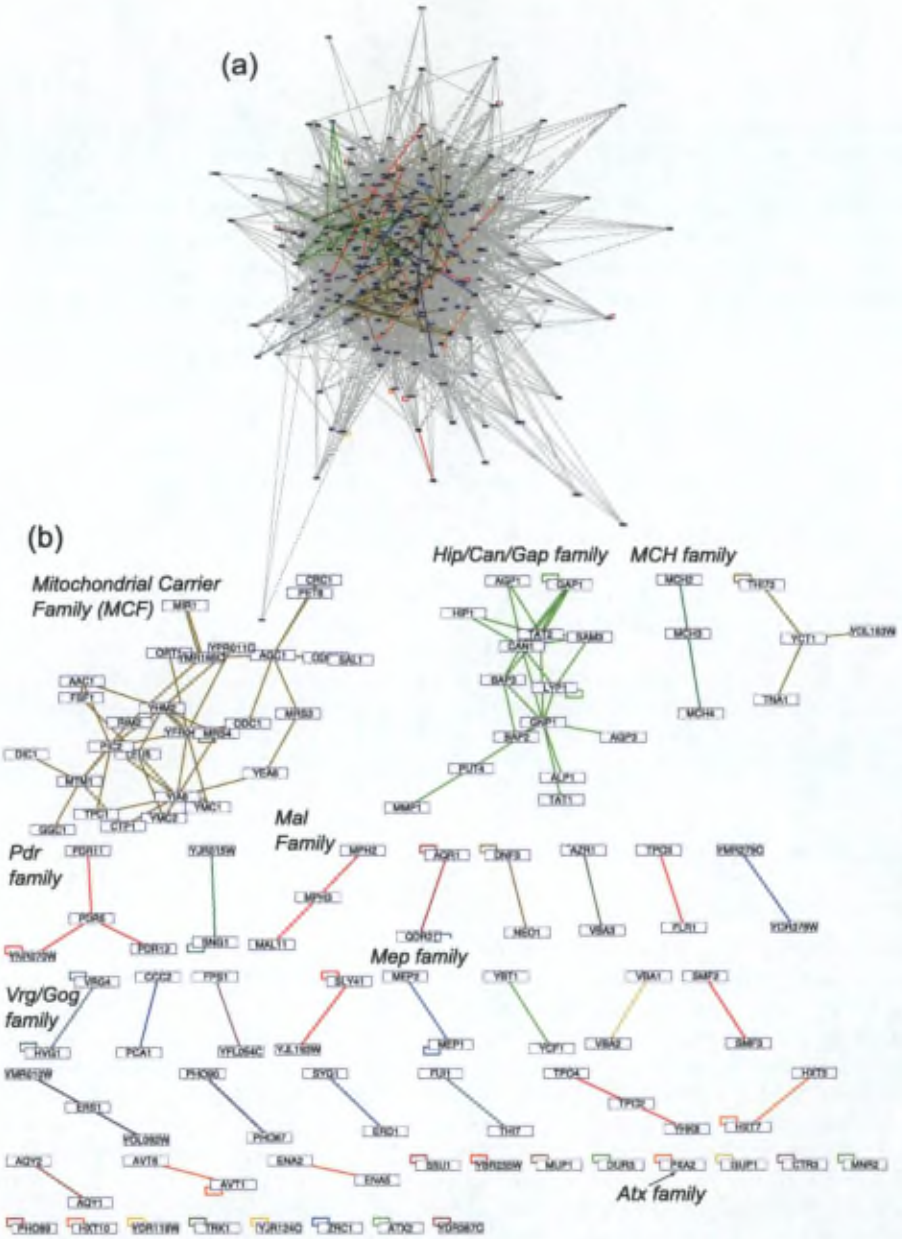


FIG. 8.5 – Représentation du réseau d'interaction constitué des 3052 interactions significatives obtenues après traitement des données expérimentales. (a) Réseau complet. Les couleurs correspondent aux arêtes entre protéines appartenant aux mêmes classes de la base de données *YTPdb* et le gris correspond aux arêtes entre protéines n'appartenant pas à la même classe. Parmi celles-ci on observe deux groupes principaux, les protéines de la famille des transporteurs mitochondriaux (brun) et les transporteurs d'acides aminés de la membrane plasmique (vert clair). (b) Arêtes intra-classes uniquement.

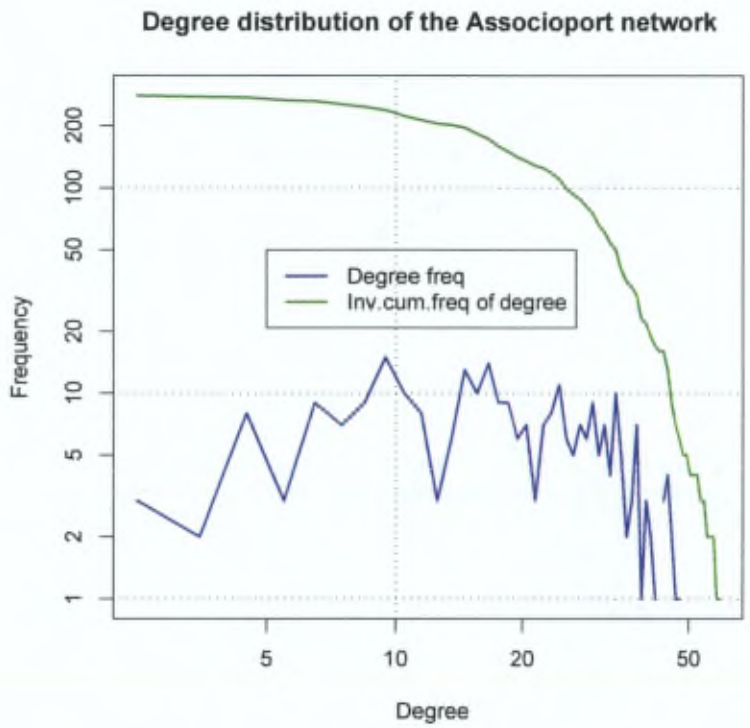


FIG. 8.6 – Distribution des fréquences des degrés du réseau Associoport.

TAB. 8.2 – Liste des 75 interactions réciproques (parmi les 3052 interactions significatives)

PDR15	Plasma membrane ATP binding cassette (ABC) multidrug transporter	AVT5	Putative transporter related to vesicular GABA-glycine transporters
SPF1	P-type ATPase, ion transporter of the ER membrane involved in ER function and Ca2+ homeostasis	VBA2	Permease of basic amino acids in the vacuolar membrane
SPF1	P-type ATPase, ion transporter of the ER membrane involved in ER function and Ca2+ homeostasis	YDR387C	Putative transporter, member of the sugar porter family
ALR2	Probable Mg(2+) transporter	SSY1	Component of the SPS plasma membrane amino acid sensor system (Ssy1p-Ptr3p-Ssy5p
GUP1	Plasma membrane protein with a possible role in proton symport of glycerol	SNQ2	Plasma membrane ATP-binding cassette (ABC) multidrug transporter
GUP1	Plasma membrane protein with a possible role in proton symport of glycerol	HVG1	Protein of unknown function, has homology to Vrg4p
VRG4	Golgi GDP-mannose transporter	SPF1	P-type ATPase, ion transporter of the ER membrane involved in ER function and Ca2+ homeostasis
VRG4	Golgi GDP-mannose transporter	TAT1	Amino acid transporter protein
ZRT3	High-affinity zinc transporter of the plasma membrane	GUP1	Plasma membrane protein with a possible role in proton symport of glycerol
AZR1	Plasma membrane transporter of the major facilitator superfamily, involved in resistance to azole drugs	HVG1	Protein of unknown function, has homology to Vrg4p
YHK8	Presumed antiporter of the DHA1 family of multidrug resistance transporters	LEU5	Mitochondrial carrier protein involved in the accumulation of CoA in the mitochondrial matrix ;
POR2	Putative mitochondrial porin (voltage-dependent anion channel)	SNF3	Plasma membrane glucose sensor that regulates glucose transport
MNR2	Putative magnesium transporter	AVT5	Putative transporter related to vesicular GABA-glycine transporters
AVT3	Putative transporter related to vesicular GABA-glycine transporters	YBR235W	Putative ion transporter, similar to mammalian electroneutral Na(+)-(K+)-Cl- cotransporter family ;
TPO5	Protein involved in excretion of putrescine and spermidine ; putative polyamine transporter in the Golgi or post-Golgi vesicles	YCF1	Vacuolar glutathione S-conjugate transporter of the ATP-binding cassette family
ZRT3	Vacuolar membrane zinc transporter, transports zinc from storage in the vacuole to the cytoplasm	PMR1	High affinity Ca2+/Mn2+ P-type ATPase required for Ca2+ and Mn2+ transport into Golgi ;
ZRT3	Vacuolar membrane zinc transporter, transports zinc from storage in the vacuole to the cytoplasm	FUR4	Uracil permease, localized to the plasma membrane
STE6	Plasma membrane ATP-binding cassette (ABC) transporter required for the export of a-factor	ZRT1	High-affinity zinc transporter of the plasma membrane
GAP1	General amino acid permease ; localization to the plasma membrane	CAN1	Plasma membrane arginine permease
AQY2	Water channel that mediates the transport of water across cell membranes	GUP1	Plasma membrane protein with a possible role in proton symport of glycerol
AQY2	Water channel that mediates the transport of water across cell membranes	HVG1	Protein of unknown function, has homology to Vrg4p
THI7	Plasma membrane transporter responsible for the uptake of thiamine	GAP1	General amino acid permease ; localization to the plasma membrane
YMD8	Putative nucleotide sugar transporter, has similarity to Vrg4p	AVT6	Vacuolar amino acid transporter, exports aspartate and glutamate from the vacuole ; member of a family of s S. cerevisiae genes (AVT1-7) related to vesicular GABA-glycine transporters
PHO84	High-affinity inorganic phosphate (Pi) transporter and low-affinity manganese transporter	PHO90	Low-affinity phosphate transporter
YMR011W	Putative protein of unknown function	AVT3	Vacuolar transporter, exports large neutral amino acids from the vacuole
HXT2	High-affinity glucose transporter	YCT1	High-affinity cysteine-specific transporter of the ER
DNF3	Aminophospholipid translocase (flippase) that maintains membrane lipid asymmetry in post-Golgi secretory vesicles ;	TPO2	Polyamine transport protein specific for spermine
YMR056C	Predicted transporter of the mitochondrial inner membrane	MIR1	Mitochondrial phosphate carrier, imports inorganic phosphate into mitochondria
YMR166C	Predicted transporter of the mitochondrial inner membrane	MEP1	Ammonium permease
MMT1	Putative metal transporter involved in mitochondrial iron accumulation ; closely related to Mrs4p	YCT1	High-affinity cysteine-specific transporter with similarity to the Dal5p family of transporters
MMT1	Putative metal transporter involved in mitochondrial iron accumulation ; closely related to Mrs4p	DRS2	Aminophospholipid translocase (flippase) that maintains membrane lipid asymmetry in post-Golgi secretory vesicles
MMT1	Putative metal transporter involved in mitochondrial iron accumulation ; closely related to Mrs4p	TAT1	Amino acid transporter protein
YHM2	Mitochondrial DNA-binding protein	YPR045W	Putative mitochondrial transport protein
YHM2	Mitochondrial DNA-binding protein	KHA1	Putative Golgi K+/H+ antiporter with a probable role in intracellular cation homeostasis
ZRC1	Vacuolar membrane zinc transporter	VBA1	Permease of basic amino acids in the vacuolar membrane
ZRC1	Vacuolar membrane zinc transporter	KHA1	Putative Golgi K+/H+ antiporter with a probable role in intracellular cation homeostasis
ZRC1	Vacuolar membrane zinc transporter	ZRT3	Vacuolar membrane zinc transporter
ATM1	Mitochondrial inner membrane ATP-binding cassette (ABC) transporter, exports mitochondrially synthesized precursors of iron-sulfur (Fe/S) clusters to the cytosol	SUL2	High affinity sulfate plasma membrane permease
FET4	Low-affinity Fe(II) transporter of the plasma membrane	GAP1	General amino acid permease ; localization to the plasma membrane
FET4	Low-affinity Fe(II) transporter of the plasma membrane	PHO87	Low-affinity inorganic phosphate (Pi) transporter,
FET4	Low-affinity Fe(II) transporter of the plasma membrane	NHX1	Endosomal Na+/H+ exchanger
POR1	Mitochondrial porin	AAC1	Mitochondrial inner membrane ADP/ATP translocator
POR1	Mitochondrial porin	ATM1	Mitochondrial inner membrane ADP/ATP translocator
AVT4	Vacuolar transporter related to vesicular GABA-glycine transporters	GUP1	Plasma membrane protein with a possible role in proton symport of glycerol
ESBP6	Protein with similarity to monocarboxylate permeases, appears not to be involved in transport of monocarboxylates	PHO84	High-affinity inorganic phosphate (Pi) transporter
YNR070W	Putative mitochondrial transporter of the ATP-binding cassette (ABC) family, implicated in pleiotropic drug resistance	YFL056C	Putative channel-like protein ; similar to Fps1p ; mediates passive diffusion of glycerol in the presence of ethanol
TAT2	High affinity tryptophan and tyrosine permease, overexpression	GAP1	General amino acid permease ; localization to the plasma membrane
TAT2	High affinity tryptophan and tyrosine permease, overexpression	DRS2	Aminophospholipid translocase (flippase) that maintains membrane lipid asymmetry in post-Golgi secretory vesicles ;
YOL092W	Putative protein of unknown function ;	GNP1	High-affinity glutamine permease
MCH4	Protein with similarity to mammalian monocarboxylate permeases,	AZR1	Plasma membrane transporter of the major facilitator superfamily, involved in resistance to azole drugs
ALR1	Plasma membrane Mg(2+) transporter	POR2	Putative mitochondrial porin (voltage-dependent anion channel)
ALR1	Plasma membrane Mg(2+) transporter	YNR070W	Putative transporter of the ATP-binding cassette (ABC) family, implicated in pleiotropic drug resistance ;
AUS1	Transporter of the ATP-binding cassette family, involved in uptake of sterols and anaerobic growth	AQY2	Water channel that mediates the transport of water across cell membranes
NRT1	High-affinity nicotinamide riboside transporter ; also transports thiamine with low affinity ; shares sequence similarity with Thi7p and Thi72p ; proposed to be involved in 5-fluorocytosine sensitivity	YEL069W-A	Putative purine-cytosine permease, very similar to Fcy2p but cannot substitute for its function
ATX2	Golgi membrane protein involved in manganese homeostasis ; overproduction suppresses the sod1 (copper, zinc superoxide dismutase) null mutation	DUR3	Plasma membrane transporter for both urea and polyamines
YVC1	Vacuolar cation channel, mediates release of Ca(2+) from the vacuole	FAT1	Fatty acid transporter and very long-chain fatty acyl-CoA synthetase
PDR5	Plasma membrane ATP-binding cassette (ABC) transporter, short-lived multidrug transporter actively regulated by Pdr1p ; also involved in sterol transport, cation resistance, and cellular detoxification during exponential growth	YKR100W	Protein of unconfirmed function ; displays a topology characteristic of the Major Facilitators Superfamily
TPO4	Polyamine transport protein, recognizes spermine, putrescine, and spermidine ; localizes to the plasma membrane ; member of the major facilitator superfamily	MEP2	Ammonium permease
TPO4	Polyamine transport protein, recognizes spermine, putrescine, and spermidine ; localizes to the plasma membrane ; member of the major facilitator superfamily	ENA1	P-type ATPase sodium pump, involved in Na+ and Li+ efflux to allow salt tolerance
PDR10	ATP-binding cassette (ABC) transporter, multidrug transporter involved in the pleiotropic drug resistance network ; regulated by Pdr1p and Pdr3p	YPR045W	Putative mitochondrial transport protein ; null mutant is viable, exhibits decreased levels of chitin and normal resistance to calcofluor white
YOR378W	Putative protein of unknown function ; belongs to a family of drug :H+ antiporters	THI73	Putative plasma membrane permease proposed to be involved in carboxylic acid uptake and repressed by thiamine ; substrate of Dbf2p/Mob1p kinase ; transcription is altered if mitochondrial dysfunction occurs
YOR378W	Putative protein of unknown function ; belongs to a family of drug :H+ antiporters	YMR031W	Putative protein of unknown function ; green fluorescent protein (GFP)-fusion protein localizes to the cytoplasm ; YMR031W is not an essential gene ; YMR031W mRNA is transcribed with ADE1
YOR378W	Putative protein of unknown function ; belongs to a family of drug :H+ antiporters	MMT1	Putative metal transporter involved in mitochondrial iron accumulation ; closely related to Mrs4p
YOR378W	Putative protein of unknown function ; belongs to a family of drug :H+ antiporters	HXT2	High-affinity glucose transporter of the major facilitator superfamily, expression is induced by low levels of glucose and repressed by high levels of glucose
YOR378W	Putative protein of unknown function ; belongs to a family of drug :H+ antiporters	VRG4	Golgi GDP-mannose transporter
NCR1	Vacuolar membrane protein that transits through the biosynthetic vacuolar protein sorting pathway, involved in sphingolipid metabolism	FCY2	Purine-cytosine permease, mediates purine (adenine, guanine, and hypoxanthine) and cytosine accumulation
NCR1	Vacuolar membrane protein that transits through the biosynthetic vacuolar protein sorting pathway, involved in sphingolipid metabolism	DAL4	Allantoin permease ; expression sensitive to nitrogen catabolite repression and induced by allophanate, an intermediate in allantoin degradation
SSU1	Plasma membrane sulfite pump involved in sulfite metabolism and required for efficient sulfite efflux ; major facilitator superfamily protein	YBT1	Transporter of the ATP-binding cassette (ABC) family involved in bile acid transport ; similar to mammalian bile transporters
HUT1	Protein with a role in UDP-galactose transport to the Golgi lumen, has similarity to human UDP-galactose transporter UGTrel1, exhibits a genetic interaction with S. cerevisiae ERO1	AQY2	Water channel that mediates the transport of water across cell membranes, only expressed in proliferating cells, controlled by osmotic signals, may be involved in freeze tolerance ; disrupted by a stop codon in many S. cerevisiae strains
MDL2	Mitochondrial inner membrane half-type ATP-binding cassette (ABC) transporter	ATM1	Mitochondrial inner membrane ATP-binding cassette (ABC) transporter, exports mitochondrially synthesized precursors of iron-sulfur (Fe/S) clusters to the cytosol
YPR011C	Putative transporter, member of the mitochondrial carrier family	CCH1	Voltage-gated high-affinity calcium channel involved in calcium influx
ANT1	Peroxisomal adenine nucleotide transporter	YEA6	Putative mitochondrial NAD+ transporter
TPO3	Polyamine transport protein specific for spermine	THI71	Transporter of thiamine or related compound ; shares sequence similarity to Thi7p ; proposed to be involved in 5-fluorocytosine sensitivity
TPO3	Polyamine transport protein specific for spermine	THI7	Plasma membrane transporter responsible for the uptake of thiamine
OPT2	Oligopeptide transporter ; member of the OPT family	YCT1	High-affinity cysteine-specific transporter of ER

Transporteurs mitochondriaux Comme nous l'avions déjà évoqué dans l'introduction, les transporteurs mitochondriaux sont connus pour former des dimères. Ceci n'est pas observable à première vue puisque seule Mrs4p semble dimériser. Cependant, ces protéines semblent néanmoins interagir fortement les unes avec les autres. Les interactions entre ces protéines pourraient toutefois être des transitoires lors de leur prise en charge par les complexes TIM et / ou TOM qui permettent aux protéines d'être intégrées dans les membranes mitochondriales.

Transporteurs d'acides aminés La figure Figure 8.5(b) révèle un autre groupe de protéines inter-connectées au sein de leur classe. Il s'agit des transporteurs d'acides aminés de la membrane plasmique (famille Can/Gap/Hip). Parmi ces protéines plusieurs semblent former des dimères (comme Gap1p et Lyp1p). En outre, bon nombre de ces protéines semblent interagir les unes avec les autres et deux interactions font parties des interactios réciproques détaillées au Tableau 8.2. Néanmoins, ces interactions pourraient être dûes au fait que ces perméases transitent toutes au cours de leur sécrétion par la chaperonine Shr3p au niveau de l'appareil de Golgi [158].

Famille Vrg / Gog Vrg4p, qui transporte du GDP-mannose dans l'appareil de Golgi, est connue pour former des oligomères [26]. Nos données semblent confirmer cette oligomérisation. Par ailleurs, Vrg4p semble aussi interagir avec Hvg1, une protéine homologue. Cette protéine est aussi localisée dans les organelles de la voie de sécrétion et semble former des dimères. Bien qu'il ait été démontré que les fonctions de Hvg1p et Vrg4p ne se recouvraient pas [159], cela pourrait néanmoins indiquer que ces protéines interagissent et oligomérisent dans la membrane de l'appareil de Golgi.

Transporteurs d'ammonium Une interaction significative est détectée entre Mep1p et Mep2p, deux transporteurs d'ammonium localisés à la membrane plasmique. En outre, Mep1p semble interagir avec lui-même. Plusieurs indications de la littérature vont dans le sens d'une oligomerisation de ces transporteurs. Tout d'abord, comme nous l'avons mentionné dans l'introduction, les protéines Mep1p et Mep3p semblent interagir au moins fonctionnellement, une mutation de la proteine Mep1 inhibant l'activité de Mep3p [28], Ensuite, dans d'autres espèces cette association a par ailleurs été identifiée [29]. Enfin, plus récemment, la structure 3D d'un homologue bactérien de ces transporteurs a confirmé qu'ils étaient actifs sous forme de trimères stables [160].

Si une interaction physique entre les transporteurs d'ammonium est tout à fait probable, la détection de cette interaction par notre approche pose néanmoins question. En effet, dans notre expérience, Cub est fusionné à la partie carboxy-terminale de Mep2 et NubG à la partie amino-terminale de Mep1. On sait cependant que la partie amino-terminale des protéines Mep est extra-cellulaire tandis que leur partie extra-cellulaire est intra-cellulaire [160]. Nub est donc en dehors de la cellule, tandis que Cub est à l'intérieur, ce qui rend toute reconstitution de l'ubiquitine théoriquement impossible y compris au cours de leur sécrétion. La détection de l'interaction pourrait être due à un mauvais reploiement de Mep2 et / ou Mep1 dû à l'ajout de Cub-PLV ou de Nub à leur extrémité, respectivement. L'interaction détectée entre ces deux transporteurs doit donc être considérée avec précaution et relève donc nettement plus de l'indice que de la preuve expérimentale.

Enfin, notons aussi les cas encourageants de la perméase Ctr3p (transporteur de cuivre dans la membrane plasmique), connue pour former des trimères [25], de Pxa2p (transporteur d'acides gras dans le peroxysome), connue pour dimériser [161] et de Vba1p et Vba2p (transporteurs d'acides aminés de la vacuole).

8.2.3.2 Comparaison du réseau d'interactions Associoport avec d'autres jeux de données à grande échelle

La table 8.3 reprend les résultats d'une série de comparaisons deux à deux entre différents réseaux d'interactions. Pour la comparaison, nous utilisons des jeux de données issus d'études expérimentales (double hybride, co-immunoprécipitation, etc). Nous avons avons notamment utilisé un jeu de données d'interactions établies entre plus de 500 protéines membranaires par split ubiquitin qui devrait être assez proche de nos résultats [33] et un jeu de données de létalité synthétiques (létalité résultant de l'inactivation d'un gène en présence uniquement d'une autre mutation). Nos jeux de données comprennent aussi des bases de données ne contenant que des données expérimentales (BioGrid) ou intégrant différents types d'interactions (STRING).

Globalement, nous pouvons observer que les différents jeux de données ne se recouvrent généralement que très peu en terme d'arêtes (sauf en ce qui concerne les jeux de données globaux comme STRING et BioGrid). Par exemple,

	Associoport (279 nodes, 2977 edges)	Miller (585 nodes, 1877 edges) [33]	Krogan (2675 nodes, 7088 edges) [97]	Ito (1564 nodes, 3889 edges) [64]	Gavin (1352 nodes, 3210 edges) [30]	Gavin 2006 (1430 nodes, 6531 edges) [36]	Ito core (737 nodes, 785 edges) [32]	Ito complete (2537 nodes, 4038 edges) [32]	Uetz (926 nodes, 865 edges) [31]	STRING interseuil (5270 nodes, 71648 edges) [144]	STRING complet (5611 nodes, 200974 edges) [144]	BioGRID synthetic lethality (2510 nodes, 10053 edges) [162]
Miller	E=197 ; E=3 ; Pval=1.0e+00) ;											
Krogan	E=48 ; E=0 ; E=109 ; E=5 ; Pval=1.0e+00) ; Pval=1.7e-01 ;											
Ito	E=27 ; E=0 ; E=58 ; E=0 ; Pval=1.0e+00) ; Pval=1.0e+00) ;		E=1063 ; E=322 ; Pval<1e-300) ;									
Gavin (2002)	E=11 ; E=0 ; E=48 ; E=6 ; E=659 ; E=198 ; Pval=1.0e+00) ; Pval=1.0e-01 ; E=1091 ; E=809 ; Pval<1e-300) ; Pval=4.5e-248 ;											
Gavin (2006)	E=8 ; E=0 ; E=52 ; E=4 ; E=1140 ; E=1542 ; Pval<1e-300) ; E=684 ; E=294 ; Pval<1e-300) ; E=1074 ; E=2143 ; Pval<1e-300) ;											
Ito core	E=3 ; E=0 ; E=33 ; E=3 ; E=487 ; E=134 ; Pval=1.0e+00) ; Pval=1.0e+00) ; Pval=2.9e-98 ; E=276 ; E=41 ; Pval=2.9e-181 ; E=262 ; E=58 ; Pval=7.4e-41 ; E=75 ;											
Ito complete	E=121 ; E=0 ; E=288 ; E=6 ; E=483 ; E=44 ; E=710 ; E=31 ; E=787 ; E=105 ; Pval=1.0e+00) ; Pval=5.3e-03 ; E=1415 ; E=164 ; Pval=4.3e-70 ; Pval=1.3e-107 ; Pval<1e-300) ;											
Uetz	E=16 ; E=0 ; E=48 ; E=2 ; E=328 ; E=103 ; E=385 ; E=51 ; E=207 ; E=45 ; E=365 ; E=62 ; E=345 ; E=122 ; E=448 ; E=133 ; Pval=1.0e+00) ; Pval=5.1e-01 ; Pval=4.4e-156 ; Pval=4.4e-51 ; Pval=2.3e-62 ; Pval=2.5e-234 ; Pval<1e-300) ;											
STRING interseuil	E=259 ; E=22 ; E=531 ; E=94 ; Pval=6.0e-02 ; Pval<1e-300) ; E=2674 ; E=6980) ; Pval<1e-300) ; E=1581 ; E=1171 ; Pval<1e-300) ; E=1392 ; E=1176 ; Pval<1e-300) ; E=1430 ; E=6072 ; Pval<1e-300) ; E=798 ; E=471 ; Pval<1e-300) ; E=2831 ; E=1117 ; Pval<1e-300) ; E=424 ; E=58 ; Pval<1e-300) ;											
STRING complet	E=291 ; E=65 ; E=533 ; E=94 ; Pval=5.5e-05 ; Pval<1e-300) ; E=2674 ; E=6980) ; Pval<1e-300) ; E=1564 ; E=2175 ; Pval<1e-300) ; E=1392 ; E=1177 ; Pval<1e-300) ; E=1430 ; E=6083 ; Pval<1e-300) ; E=794 ; E=471 ; Pval<1e-300) ; E=2804 ; E=2117 ; Pval<1e-300) ; E=435 ; E=268 ; Pval<1e-300) ; E=3270 ; E=7164S ; Pval<1e-300) ;											
BioGRID synthetic	E=92 ; E=3 ; E=253 ; E=6 ; Pval=9.9e-01 ; Pval=4.0e-01 ; E=1446 ; E=174 ; Pval=7.2e-147 ; E=480 ; E=49 ; Pval=1.5e-43 ; E=841 ; E=82 ; Pval=2.8e-57 ; E=1392 ; E=1176 ; Pval=1.6e-126 ; E=402 ; E=22 ; Pval=1.5e-16 ; E=1030 ; E=31 ; Pval=1.2e-15 ; E=479 ; E=14 ; Pval=2.5e-08 ; E=2462 ; E=5798 ; Pval<1e-300) ; E=2498 ; E=9910 ; Pval<1e-300) ;											
BioGRID complete	E=272 ; E=30 ; E=535 ; E=1977 ; Pval<1e-300) ; E=2672 ; E=7075 ; Pval<1e-300) ; E=1563 ; E=3597 ; Pval<1e-300) ; E=1392 ; E=4210 ; Pval<1e-300) ; E=1430 ; E=6531 ; Pval<1e-300) ; E=737 ; E=785 ; Pval<1e-300) ; E=2798 ; E=977 ; Pval<1e-300) ; E=926 ; E=865 ; Pval<1e-300) ; E=5597 ; E=39582 ; Pval<1e-300) ; E=5290 ; E=42103 ; Pval<1e-300) ; E=2510 ; E=10053 ; Pval<1e-300) ;											

TAB. 8.3 – Comparaison du réseau d'interactions Associoport à d'autres jeux de données d'interactions à large spectre et à différents autres jeux de données. Pour chaque comparaison de réseau, nous reprenons le nombre de noeuds et d'arêtes en commun ainsi que la p-valeur de l'intersection. Ito complet et Ito core correspondent à l'ensemble des données et au données les plus sûres d'une même expérience de double hybride à large spectre [32]. STRING interactions correspond aux interactions physiques annotées dans la base de données STRING complet [144]. BioGRID synthetic correspond aux interactions de létalité synthétique annotées dans la base de données BioGRID [162].

les jeux de données de double hybride de Uetz *et al* et Ito *et al* (core) n'ont que 122 arêtes en commun, cependant la P-valeur de cette intersection, c'est-à-dire, la probabilité que le nombre d'arêtes en commun soit supérieur ou égal à 122, étant donné ces deux réseaux soit due au hasard est très significative (2,5e-228).

Des P-valeurs aussi basses ne sont observées pour aucun des jeux de données auxquels nous avons comparé notre réseau expérimental, le réseau présentant la plus grande intersection avec celui-ci est le réseau d'interaction fonctionnelles complet de la base de données STRING (ligne 11, col 1) qui compte plus de 200000 interactions fonctionnelles (régulation, co-occurence). Par contre, curieusement notre réseau ne compte que trois interactions en commun avec le réseau tiré de l'analyse de Miller *et al* [33] et la P-valeur de cette comparaison est maximale. Notons que ce jeu de données présente une intersection significative avec les jeux de données de STRING et de BioGrid, néanmoins, nous pouvons observer que ces jeux de données présentent des intersections significatives avec toute les autres jeux de données vraisemblablement parce que BioGrid inclut tous les jeux de données à haut débit publiés (dont Miller *et al*) et que STRING intègre au moins partiellement la base de données BioGrid.

Il est assez logique que ni notre étude, ni celle de Miller *et al* ne présente de forte intersection avec les autres jeux de données. En effet, comme nous l'avons déjà souligné par ailleurs, les méthodes utilisées pour constituer ces jeux de données à grande échelle ne sont pas prévues pour la détection d'interactions entre protéines membranaires et les données les concernant sont donc très lacunaires. Par contre que le fait que l'intersection soit presque inexistante entre les données Associoport et celles de Miller *et al* pose déjà plus question étant donnée la similitude entre ces deux études pour les protéines testées et la méthodologie utilisée.

Parmi tous les jeux de données auxquels nous comparons nos données, nous pouvons observer que les intersections entre nos données et celles de Miller *et al* avec les expériences de létalité synthétique sont très faibles elles ($P - val \geq 10e - 1$)(ligne 11, colonne 1 et 2) alors qu'elles sont bien supérieures lorsque l'on compare les expériences létalité synthétiques aux autres jeux de données. Ces faibles intersections ne peuvent cependant plus s'expliquer par le fait que la méthode est peu adaptée aux protéines membranaires et reflètent vraisemblablement leur manque de fiabilité.

Cependant comme nous l'avons déjà observé précédemment en comparant les classes de *YTPdb* au réseau, certaines informations intéressantes peuvent être dégagées de l'étude de ces données, fussent elles bruitées. Pour cela, nous aurons recours à des analyses étudiant des groupes de protéines établis selon différentes méthodes présentées dans la suite de ce chapitre (voisinage, clustering).

8.2.3.3 Analyse du voisinage des protéines

Nous avons étudié si les interactions détectées expérimentalement se faisaient entre des protéines de fonctions similaires ou appartenant aux mêmes classes fonctionnelles. Pour cela, nous avons étudié le voisinage de chacune des protéines au sein du réseau d'interactions. Pour chaque protéine, nous constituons un groupe de voisinage comprenant l'ensemble des protéines qui présentent une interaction significative avec elle ainsi que la protéine elle même. Nous obtenons donc 279 groupes de voisinages.

Chacun des groupes de voisinage est ensuite comparé (au moyen du programme *compare-classes* de *NeAT*) à chacune des familles de notre base de données de façons à détecter les intersections significatives (selon la distribution hypergéométrique [145]) entre les classes de *YTPdb* et les groupes de voisins (Table 8.4).

Seules 3 familles montrent une intersection faiblement significative avec les groupes de voisinage. Il s'agit des familles des transporteurs mitochondriaux, des gènes *AVT* (des transporteurs d'acides aminés vacuolaires putatifs) et les perméases d'acides aminés de la membrane plasmique.

8.2.3.4 Clustering du réseau d'interaction

Dans la section précédente, nous avons extrait les groupes d'interactions correspondant à des groupes fonctionnels pré-annotés dans *YTPdb*. Nous nous sommes aussi demandé s'il était possible de retrouver au sein du réseau de groupes de fortement connectées entre elles et de fonction similaires. Ces groupes de protéines peuvent éventuellement correspondre à des complexes protéiques mais peuvent aussi refléter l'association transitoire de ces protéines, lors de leur sécrétion par exemple et donc l'appartenance à un processus biologique commun.

Au chapitre précédent, nous avons évalué différentes approches de clustering afin d'obtenir la méthode la plus adaptée à nos besoins ainsi que ses paramètres optimaux (section 7 p. 85). Cette analyse avait été faite afin de retrouver

TAB. 8.4 – Résultat de la comparaison des groupes de voisinages avec les classes de la base de données YTPdb. Chaque ligne reprend le calcul différentes statistiques pour la comparaison d'une classe de *YTPdb* (colonne 1) avec un groupe de voisin d'un gène (colonne 2).

Classe de référence	**Groupes de voisins**	**Nbre elem classe de référence**	**Nbre elem classe de voisins**	**Taille de l'intersection**	**Taille de l'union**	**P-valeur**	**E-valeur**	**Sig**
Avt family	*MID1*	7	19	6	20	2.40E-007	0.01	2.19
Hip/Can/Gap family	*CAN1*	18	22	9	31	4.80E-007	0.01	1.89
Hip/Can/Gap family	*GNP1*	18	40	10	48	1.70E-005	0.45	0.34
MCF family	TPO2	35	45	15	65	2.80E-005	0.75	0.13
Hip/Can/Gap family	*TPO3*	18	40	9	49	0	4.03	-0.61
Hip/Can/Gap family	*BAP3*	18	13	5	26	0	13.36	-1.13
Sul/Sfp family	*OAC1*	4	16	3	17	0	14.61	-1.16
Hip/Can/Gap family	*SAM3*	18	8	4	22	0	17.23	-1.24
Hip/Can/Gap family	*TAT2*	18	38	8	48	0	19.23	-1.28
Smf family	*SMF3*	3	6	2	7	0	28.8	-1.46
...								

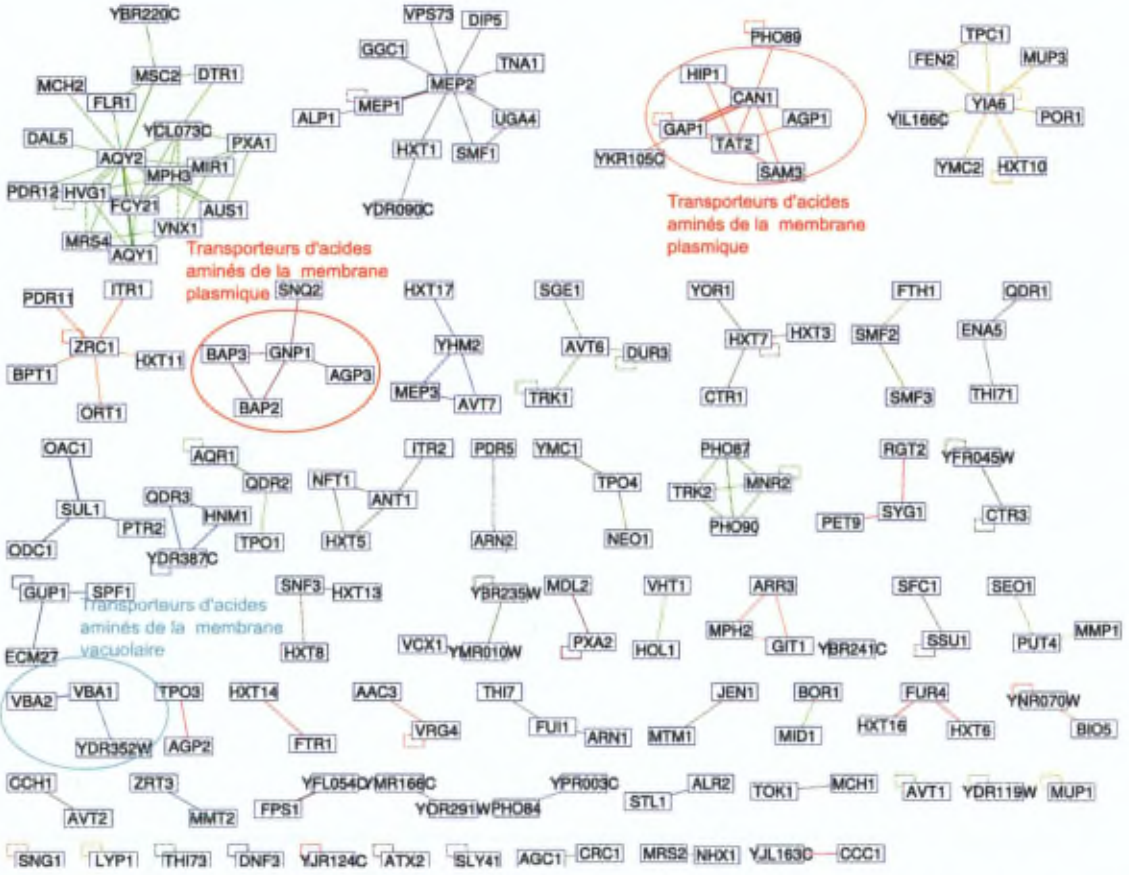


FIG. 8.7 – Clusters retournés par l'algorithme *MCL* appliqué au réseau. La pondération a été effectuée sur base des annotations dans *YTPdb*. Les trois clusters entourés correspondent aux groupes décrits dans le texte (transporteurs d'acides aminés de la membrane plasmique (orange) et vacuolaire (bleu))

au sein du réseau expérimental des groupes de protéines densément interconnectées et de fonction similaires pouvant éventuellement correspondre à des complexes protéiques.

Clustering non-pondéré En appliquant l'algorithme *MCL* sans pondération au graphe issu des données Associoport, nous n'obtenons que 18 clusters (le graphe est extrêment connecté). Parmi ceux-ci, aucun cluster ne semble présenter un nombre significativement élevé de protéines appartenant à la même classe fonctionnelle de *YTPdb*.

Clustering pondéré Les résultats décevants de l'approche de clustering sur le graphe non pondéré nous on conduit à ajouter des poids reflétant la pertinence biologique aux interactions détectées expérimentalement. Nous avons intégré au graphe expérimental les annotations disponibles dans notre base de données. Le poids de chaque interaction dont les membres présentent la même localisation, la même famille ou le même substrat a été augmenté en fonction de la taille du groupe. Ainsi le poids d'une interaction entre deux protéines appartenant à un groupe plus large (protéine de la membrane plasmique, transporteurs ABC) augmentant moins qu'une interaction entre deux protéines d'une famille plus réduite (transporteurs d'ammonium). Par ailleurs, si l'interaction a déjà été détectée dans une autre étude, le poids de cette interaction est doublé.

L'algorithme MCL prend en compte les graphes pondérés et accorde plus d'importances aux arêtes de poids élevé. Ainsi un ensemble de noeuds densément interconnecté et dont les arêtes sont de poids élevés sera plus significatif pour l'algorithme.

Avec l'application de *MCL* au graphe ainsi pondéré nous permet de découvrir plusieurs clusters composés contenant un nombre significatif de protéines apparentées (Figure 8.7). Parmi ceux-ci, notons :

- Deux clusters formés quasi exclusivement de transporteurs d'acides aminés localisés à la membrane plasmique.
- Un cluster formé par trois transporteurs localisés à la membrane vacuolaire dont deux sont des transporteurs d'acides aminés basiques fortement apparentés et le troisième est homologue des transporteurs de cystine localisé dans le lysosome humain.

8.3 Conclusion

Malgré le fait que les expériences et les différents contrôles affichaient un comportement globalement satisfaisant, les résultats expérimentaux du projet Associoport s'avèrent assez décevants. En effet, les taux de recouvrement avec d'autres jeux de données d'interactions physiques ou fonctionnelles s'avèrent très faibles et les donnes filtrées semblent être des faux positifs pour la plupart (et ce, même pour les interactions réciproques découvertes plusieurs fois).

Cependant, plusieurs résultats expérimentaux intéressants ont pu être observés. Notamment, l'application de techniques de clustering et l'analyse des groupes de voisins nous a permis d'établir que les transporteurs d'acides aminés de la membrane plasmique et de la membrane vacuolaire semblent interagir entre eux. Certaines de ces interactions ont été confirmées dans le laboratoire du Professeur Bruno André et feront vraisemblablement l'objet d'une publication dans les mois à venir (Lauwers et André, communication personnelle). Par ailleurs, nous avons tenté de confirmer la dimérisation de la perméase Gap1 par co-immunoprécipitation mais sans grand succès, notamment en raison de la difficulté à précipiter les protéines transmembranaires insolubles. Enfin, plusieurs interactions détectées par le projet Associoport et déjà annotées par ailleurs ont pu être retrouvées.

Dans la suite, nous allons inférer différents réseaux biologiques (co-régulation, co-évolution) pour préciser ce réseau d'interactions expérimentales.

Chapitre 9

Etudes de la comparaison de profils phylogénétiques entre les perméases

Comme nous l'avons déjà mentionné dans l'introduction (section 2.3.1.1 p. 27), la méthode des profils phylogénétiques se base sur la corrélation de la présence ou de l'absence des gènes impliqués dans une même fonction cellulaire au sein des génomes. Ainsi, les paires de gènes qui sont presque toujours présentes ou absents ensemble dans une série de génomes ou à l'inverse qui ne sont presque jamais présents ou absents ensemble, c'est-à-dire qui présentent des profils de présence / absence (anti-)corrélés sont de bons candidats à une interaction fonctionnelle [58].

Nous avons appliqué cette méthode afin d'en dériver un réseau d'interactions fonctionnelles potentielles dont nous espérons qu'il nous permettra de préciser le réseau d'interactions entre perméases décrit dans la section précédente.

9.1 Méthodes

9.1.1 Installation de génomes fongiques

Au moment de commencer cette étude, la suite d'outils bioinformatiques RSA-Tools (RSAT) [142], développée et utilisée au laboratoire pour l'analyse des séquences régulatrices contenait la quinzaine de génomes de *Fungi* hébergés alors par la banque de données du NCBI (*http ://www.ncbi.nlm.nih.gov/*). A partir de différentes banques de données (notamment l'institut BROAD (*http ://www.broad.mit.edu/*) et celle maintenue par Jason Stajich sur le site *http ://fungal.genome.duke.edu/*), nous avons ajouté les séquences et les annotations de près de 25 nouveaux génomes fongiques et avons adapté leur format pour permettre leur installation dans *RSAT*. A ce jour, 44 génomes fongiques plus ou moins distants phylogénétiquement de *S.cerevisiae* sont disponibles sur nos serveurs et accessibles au public (*http ://rsat.ulb.ac.be/rsat/*).

9.1.2 Recherche des protéines orthologues

Pour appliquer la méthode des profils phylogénétiques, il faut retrouver les *orthologues* de tous les gènes de *S.cerevisiae* dans une série d'autres espèces. Deux gènes de deux espèces différentes sont orthologues s'ils dérivent d'un même gène dans le dernier ancêtre commun des espèces.

La méthode la plus classiquement utilisée en génomique comparative pour les prédire consiste à rechercher les meilleures similarités réciproques : la relation entre un gène X du génome A et le gène Y du génome B est appelée un meilleur hit bidirectionnel (ou BBH pour bidirectionnal best hit) quand X est le gène le plus similaire à Y parmi l'ensemble des gènes du génome B et vice versa.

Afin de déterminer les orthologues de chaque protéine de *S.cerevisiae* dans les autres organismes disponibles de *RSAT*, nous avons effectué différentes comparaisons entre génomes. Tout d'abord, pour chaque protéine, nous avons recherché les protéines similaires dans chacun des organismes appartenant aux règne des *Fungi*, des *Archaebacteria* (archae) ou des *Bacteria* (bactéries) à notre disposition dans RSAT et *vice versa*, pour chaque protéine de chaque

Fungus, archae et bactérie nous avons recherché les protéines similaires dans *S.cerevisiae*. Ainsi, pour chaque protéine de *S.cerevisiae*, nous avons pu définir ses meilleurs hits bidirectionnels, dans les différents organismes comparés s'ils existaient et que nous avons considérés dans la suite comme ses orthologues. Ces traitements ont été réalisés au moyen des programmes BLAST [163] (e-valeur <= 1e-5), et *get-orthologs* (développé au laboratoire par Jacques van Helden et Rekin's Janky). Enfin, nous pouvons créer à l'aide du programme *compare-scores* une table semblable à celle de la Figure 2.8(a)B qui indique pour chacun des organismes étudiés s'il présente ou non un orthologue pour chacune des gènes de *S.cerevisae*.

9.1.3 Etude des profils phylogénétiques

Nous avons ensuite utilisé le programme *compare-profiles* de *RSAT* afin d'étudier la table des profils phylogénétiques. Ce programme permet d'évaluer la similarité de deux profils phylogénétiques en utilisant différentes métriques de comparaison, en d'autres termes de quantifier la ressemblance de deux lignes de la matrice booléenne de présence / absence de gènes.

Nous avons choisi de calculer les profils phylogénétiques en fonction de la loi hypergéométrique, c'est-à-dire, que pour toutes les paires possibles de lignes de la matrice, nous calculons la probabilité que la taille du recouvrement entre ces deux profils soit due au hasard étant donné le nombre d'organismes considérés. Pour chaque paire de profils comparée, nous obtenons deux scores de P-valeur qui reflètent le risque de se tromper en considérant que ce recouvrement est significativement élevé (profils similaires) ou faible (profils antinomiques) selon que l'on considère la queue droite ou gauche de la distribution, respectivement.

$$P - val = P(X \geq x) = \sum_{i=x}^{min(m,k)} \frac{C_m^i C_n^{k-x}}{C_{m+n}^k}$$

$$P - val = P(X \leq x) = \sum_{i=0}^{x} \frac{C_m^i C_n^{k-x}}{C_{m+n}^k}$$

- m le nombre d'espèces ou l'orthologue est présent dans le premier profil
- k le nombre d'espèces ou l'orthologue est présent dans le second profil
- x le nombre d'espèces commun aux deux profils
- n le nombre de colonnes de la matrice

La E-valeur est plus fiable que la P-valeur car elle inclut une correction pour les tests multiples est donc à l'échelle de ce que l'on a testé. En effet, même si l'on choisissait un seuil de P-valeur de 0,001, on s'attendrait à retrouver 18000 faux positifs par le seul jeu du hasard puisque l'on considère environ 18 millions de comparaisons de profils (la levure comptant en effet environ 6000 gènes). Nous avons donc corrigé cet effet du test multiple en calculant la E-valeur, obtenue en multipliant la P-valeur par le nombre de tests. Ainsi, si l'on choisit l'ensemble des profils présentant une E-valeur de 1, nous ne devrions obtenir qu'un seul faux positif.

$$E - val = P - val * (N * (N - 1)/2)$$

- N le nombre de gènes dans l'organisme de référence

Enfin, nous utilisons la significativité qui correspond à l'inverse du logarithme de la E-valeur, une valeur plus facile à appréhender : les valeurs positives indiquant que les profils sont significativement semblables ou dissemblables.

$$sig = -log_{10} E - val$$

Nous disposons ainsi d'un réseau d'interactions fonctionnelles obtenu par profils phylogénétiques, les noeuds représentent les gènes comme noeud et les arêtes pondérée représentant le score de significativité de la similarité des profils phylogénétiques entre les deux gènes.

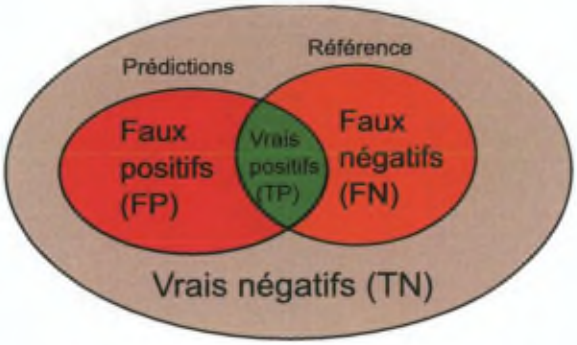


FIG. 9.1 – Représentation schématique des vrais positifs, faux négatifs, faux positifs et vrais positifs tels que calculé dans notre évaluation.

9.1.4 Evaluation

Afin d'évaluer la qualité de nos prédictions, nous l'avons comparé à un réseau considéré comme référence et constitué à partir de la base de données de voies métaboliques KEGG [164]. Au sein de ce réseau de référence, sont reliés par une arête tous les paires de gènes codant pour des protéines appartenant à une même voie métabolique. Notre graphe inféré contient des prédiction pour tous les gènes de la levure alors que notre jeu de données de référence ne comprend que les gènes annotés comme impliqué dans au moins une voie métabolique annotée. Un bon nombre des relations prédites pourraient donc être pertinentes, mais n'ont aucune chance d'être considérées comme correctes car le processus métabolique auquel appartiennent ces gènes n'est pas indiqué dans la référence. Afin d'éliminer ce biais, nous avons utilisé le programme *graph-get-clusters* pour ne sélectionner dans le graphe inféré que les gènes pour lesquels il existe au moins une annotation dans les données (c'est-à-dire les gènes impliqués dans au moins une voie métabolique de KEGG), et les arêtes qui unissent ces gènes. Nous constituons ainsi un graphe *induit* par les noeuds du graphe de référence.

Pour chaque seuil de significativité, nous calculons la sensibilité et la PPV du graphe inféré par rapport au graphe de référence. La vrais positifs (TP) sont les interactions fonctionnelles que nous prédisons et qui sont annotées dans le graphe de référence, les faux positifs (FP) sont les interactions prédites mais non annotées, les faux négatifs (FN) correspondent aux interactions non prédites mais annotées et les vrais négatifs (TN) sont les interactions non prédites et non annotées (Figure 9.1).

$$Sn = \frac{TP}{TP + FN} \quad PPV = \frac{TP}{TP + FP}$$

La sensibilité correspond à la proportion de vrais positifs retrouvés parmi l'ensemble des positifs et la valeur prédictive positive correspond à la proportion des vrais positifs parmi toutes les prédictions. A des valeurs de significativité faibles, la Sn est maximale puisque l'on considère toutes les interactions et à l'inverse la PPV est très faible puisque nous prédisons un grande proportion de faux positifs.

Comme contrôle négatif, nous avons ré-échantillonné aléatoirement dix fois les arêtes entre les noeuds du graphe induit, chaque noeud conservant ainsi le même nombre de voisin que dans le graphe réel et calculé la moyenne des scores obtenus sur les dix ré-échantillonages.

9.2 Evaluation de notre approche

9.2.1 *Fungi*

Nous avons tout d'abord inféré un réseau d'interactions fonctionnelles en utilisant tous les *Fungi* installés sur nos serveurs.

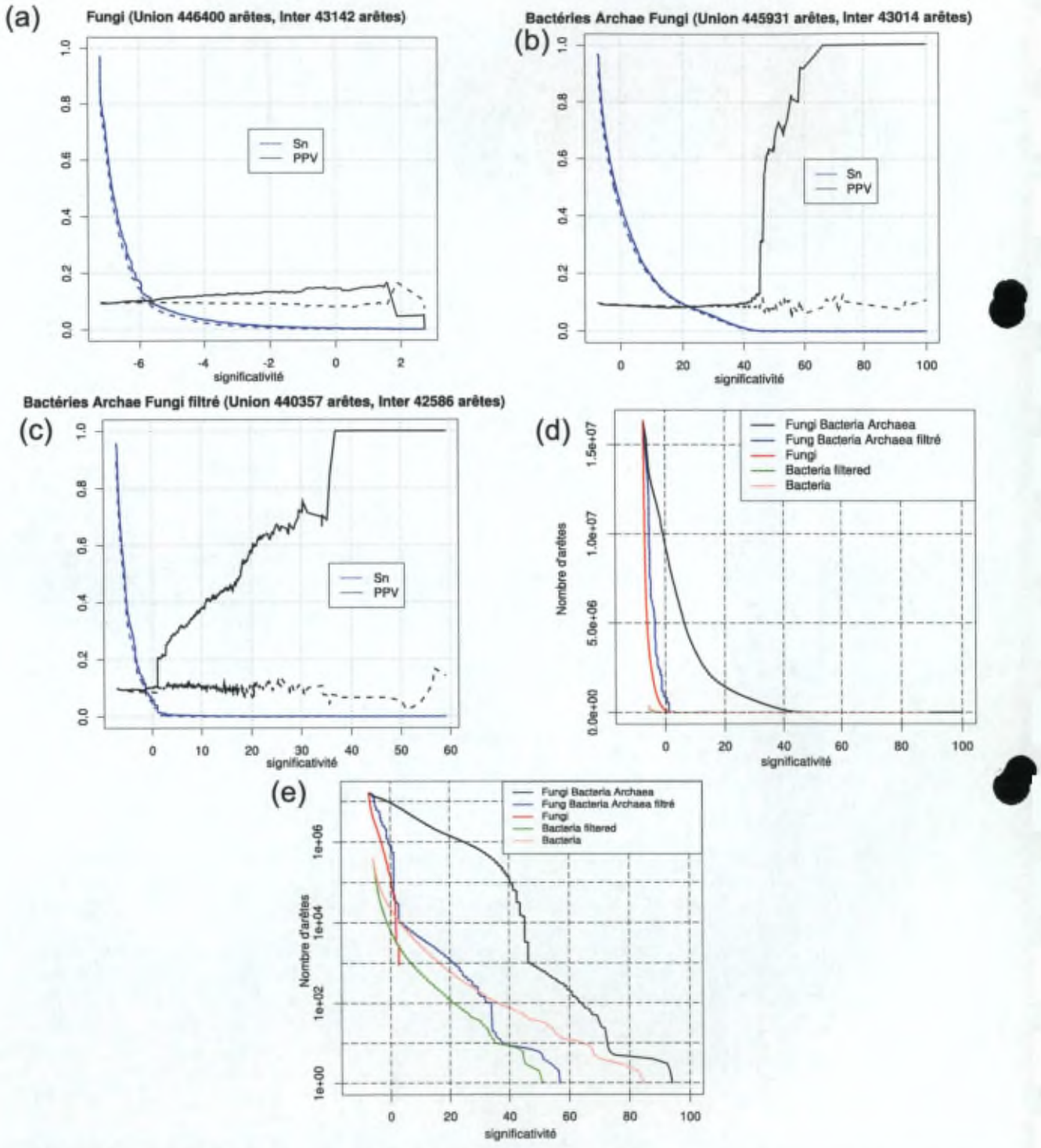


FIG. 9.2 – Evaluation du graphe obtenu par profils phylogénétiques. (a-c) *Sn* et *PPV* en fonction de la significativité hypergéométrique des interactions prédites. Différents groupes d'organismes ont été considéré. (a) Fungi, Bactéries, Archaebactéries et Fungi (b) sans ou (c) avec filgrage des organismes redondants. Les courbes pointillées montrent les résultats pour un graphe ré-échantillonné. Légende : *Union* : nombre d'arêtes dans l'union des graphes de régéférence (KEGG) et induit par les profils phylogénétique ; Inter : nombre d'arêtes dans l'intersection des graphes de régéférence et induit. (d-e) Nombre d'arêtes en fonction du seuil de significativité (d) Axes linéaires. (e) Axe Y logarithmique.

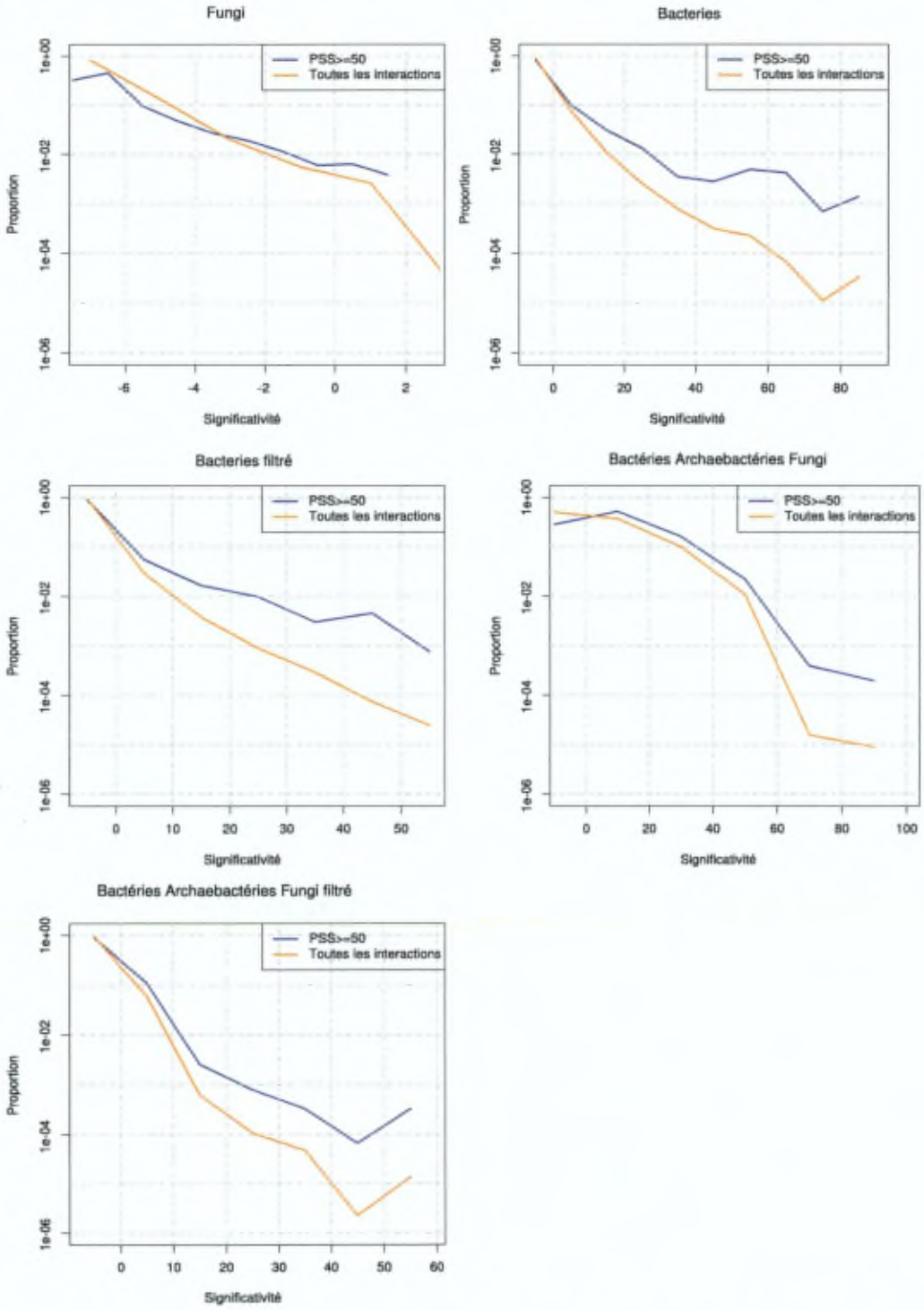


FIG. 9.3 – Distribution des scores de significativité obtenus pour les paires de gènes des graphes induit par les gènes de la base de données KEGG.

TAB. 9.1 – Caractéristiques des réseaux obtenus par l'approche des profils phylogénétiques en considérant différents groupes d'organismes.

	# org	# noeuds	# arêtes total	P-valeur (PSS >= 50)	# noeuds sig >= 0	# arêtes sig >= 0	# noeuds sig >= 10	# arêtes sig >= 10	# noeuds sig >= 30	# arêtes sig >= 30
Fungi	36	5716	16333012	2.20E-016	1455	147663	0	0	0	0
Bactéries Archaea Fungi	455	5716	16275765	2.20E-016	5276	9354508	4176	3400277	1719	651411
Bactéries Archaea Fungi filtré	262	5716	16276044	2.20E-016	2174	818546	284	4271	57	170
Bactéries	391	1123	393443	1.11E-015	713	22294	337	2780	81	190
Bactéries filtré	230	1068	346535	3.75E-013	481	6138	135	572	33	35

Nous inspirant d'une étude menée par Jothi *et al* [59], nous avons tout d'abord vérifié que les paires de gènes fonctionnellement similaires présentent des significativités hypergéométriques significativement plus élevées que l'ensemble des paires de gènes du réseau. Pour cela, nous avons comparé la moyenne des significativités des paires de gènes présentant un *pathway similarity score* (*PSS*) (tel que défini dans [165]) supérieur à 50 aux moyennes de l'ensemble des gènes en utilisant le test non paramétrique de Mann-Whitney unidirectionnel (Table 9.1, ligne 1, col 4, Figure 9.3 en haut à gauche). Deux protéines A et B ont un *PSS* de s lorsque A est présent dans au moins $s\%$ des voies métaboliques de B et vice-versa (coefficient de Jaccard de leur appartenance à des voies métaboliques) [59]. La P-valeur très faible de cette comparaison (2,2e-16) indique que la moyenne des scores de significativité est significativement plus élevée pour les gènes codant pour des protéines de fonctions semblables même si ce n'est pas très évident sur la figure 9.3 qui représente les distribution des scores, les deux courbes de distribution étant très rapprochées.

La Figure 9.2(a) montre les résultats de la comparaison entre ce graphe d'interactions fonctionnelles et le graphe de référence issu des voies métaboliques annotées dans KEGG. Pour chaque seuil de significativité, nous reprenons les valeurs de sensibilité et de valeur prédictive positive. Aux faibles de valeur de significativité (gauche du graphique), on s'attend à avoir une forte sensibilité puisque l'on considère la quasi entièreté des protéines du graphe, par contre la PPV sera réduite (taux élevé de faux positifs). A l'inverse, aux hautes valeurs de sensibilité (à droite), on s'attend à avoir une faible de sensibilité et une PPV élevée puisque l'on augmente l'on diminue le nombre d'interactions que l'on considère mais que l'on ne regarde que celles de score élevé. Nous pouvons observer que les résultats sont peu convainquants. En effet, d'une part les valeurs de significativité restent très faibles et d'autre part les performances réelles (courbes pleines) ne se différencient qu'assez peu de l'espérance aléatoire (courbes pointillées). Au plus hautes valeurs de significativité, on observe cependant une légère différence entre les courbes aléatoires et réelles pour la PPV.

Ces faibles performances sont vraisemblablement dûes au fait que les différentes espèces de *Fungi* considérées dans notre cas sont trop proches les unes des autres : ces génomes contiennent apparement les mêmes gènes que *S.cerevisiae* et cela introduit un biais dans la mesure de la significativité hypergéométrique puisqu'il n'y a plus indépendance entre les génomes. D'autres études avaient déjà constaté que si le nombre de génomes influence favorablement les performances des profils phylogénétiques, c'est surtout le choix de ces génomes qui est essentiel [59].

9.2.2 Fungi, Bactéries et Archae

Nous avons ensuite inféré un réseau en utilisant tous les archae, *Fungi* et bactéries installé sur nos serveurs.

Dans une première étape, nous avons testé avec succès que les protéines de fonction similaires présentaient des significativités plus élevées que l'ensemble des paires de protéines (Table 9.1, ligne 2, Figure 9.3 ligne 2, colonne de droite).

Cependant, la comparaison entre nos prédictions et le graphe annoté s'est une fois encore révélée décevante (Figure 9.2(b)). En effet, la significativité doit être très élevée (>40) pour que les courbes pleines et pointillées correspondant respectivement aux performances réelles et à l'espérance aléatoire soient bien distinctes, et ce uniquement pour la

PPV. Par ailleurs, une variation de la significativité de -7 à 40 n'améliore pas la valeur prédictive, ce qui signifie que avant cette valeur, le score n'est pas informatif. En outre, à ce seuil significativités, la sensibilité est très faible (1,1% et 0.8% aléatoire) et continue logiquement de diminuer dès lors que l'on considère de moins en moins d'interactions.

Le fait que la significativité doive être si élevée pour rendre compte d'une différence marquée entre courbes aléatoire et réelle tient vraisemblablement encore au fait qu'un bon nombre de génomes sont très proches et codent pour un ensemble quasi semblables de gènes. En effet, parmi les 500 génomes utilisés dans l'approche, plusieurs groupes taxonomiques contiennent les mêmes ensembles d'orthologues qui sont spécifiques d'un niveau taxonomique. Ainsi, les génomes fongiques utilisé dans cette études contiennent globalement les mêmes gènes. En outre, parmi les génomes installés sur *RSAT*, nous utilisons aussi différentes souches d'un même espèce bactérienne (10 souches pour *E.coli* et 7 pour *Yersinia pestis*, par exemple).

9.2.3 Filtrage des organismes

Nous avons donc décidé de ne conserver qu'une sélection de 262 organismes dont nous estimions qu'ils *différent entre eux du point de vue de leur ressemblance avec S.cerevisiae*. Pratiquement, nous avons regroupés les génomes présentant le même ensemble de gènes en commun avec *S.cerevisiae* et n'avons conservé qu'un seul organisme par groupe. Par ailleurs, nous avons aussi considéré d'autres groupes d'organismes (bactéries uniquement et les bactéries filtrées) et nous avons aussi vérifié que les paires des protéines ayant un *PSS* supérieur à 50 présentaient des significativités plus élevées que l'ensemble des paires de protéines (Table 9.1, ligne 3-5, Figure 9.3 ligne 3).

Avec les 262 organismes filtrés, nous pouvons observer qu'à une valeur de significativité de 2, une différence marquée est observable entre les performances réelles et aléatoires (*PPV réelle* = 20.9% et *PPV aléatoire* = 10%) (Figure 9.2(c)). De plus, aux valeurs plus élevée de significativité, la PPV continue d'augmenter devenant même maximale pour des valeurs très élevées (sig >= 36.9). Cependant, même à un seuil de significativité de 2, relativement bas, la sensibilité est déjà très faible (1.3%). Les prédictions significatives effectuées par notre approche sont donc globalement biologiquement pertinentes mais ces bonnes performances se font au détriment de la sensibilité. Ainsi, avec une significativité de 30 (ce qui correspond à une PPV d'environ 70% dans notre évaluation), notre réseau qui comptait plus de 16 millions d'arêtes au départ n'en contient plus que 170 (Figure 9.4, Table 9.1, Figure **??**(d-e)) mais qui correspondent presque toutes à une réalité biologique.

Les prédictions délivrées par l'approche des profils phylogénétiques sont donc assez fiables mais très parcellaires. Notons toutefois que la très faible sensibilité affichée tient aussi à un effet de cliques. Ainsi, avec notre méthode d'évaluation, le graphe de référence constitué des voies métoliques de KEGG est très dense car chaque voie métabolique constitue une clique dans le réseau, parfois de très grande taille. En effet, de grosses voies métaboliques annotées (*purin metabolism*) comptent plus de 100 protéines et il est peu probable que nos méthodes permettent de retrouver toutes les relations existant entre les protéines de ces voies (près de 5000, soit environ 10% des arêtes du réseau). La sensibilité calculée avec notre approche est donc nettement sous-évaluée.

D'autre part, il peut également arriver que des prédictions soient qualifiées à tort de faux positifs. Ainsi, il peut arriver que deux gènes liés fonctionnellement et prédits comme tels ne soient pas annotés dans le jeu de données de référence. Par exemple, les protéines *ARG4* et *HIS3* toutes deux impliquées dans le métabolisme des acides aminés basiques sont considérées comme des faux positifs bien qu'ils aient une fonction assez semblable.

9.3 Interactions fonctionnelles impliquant des transporteurs

Pour notre analyse, nous avons décidé d'utiliser le réseau d'interactions constitué des arêtes présentant un score de significativité supérieur ou égal à 10, (ce qui correspond environ à une PPV de 38% et une sensibilité de 0,3%). Ce réseau est représenté à la Figure 9.5. Nous avons agrandi quelques groupes de gènes fortement connectés et de fonctions similaires (dégradation des protéines, métabolisme des acides aminés, etc). Ce réseau contient 284 noeuds et 4271 arêtes (Table 9.1. Malheureusement, parmi ces 284 noeuds, seuls deux (*MEP2* et *CCC2*) sont des perméases annotées dans YTP et aucune interaction ne semble particulièrement pertinente. Il est d'ailleurs surprenant que la perméase Cc2p, un transporteur de cuivre, survienne 4 fois parmi ces interactions (Table 9.2).

Par ailleurs Ccc2 et Mep2 sont les perméases que l'on retrouve majoritairement dans les interactions de score moindre (entre 3 et 10), mais celles-ci ne semblent pas présenter plus de réalité biologique. Il semblerait en fait que

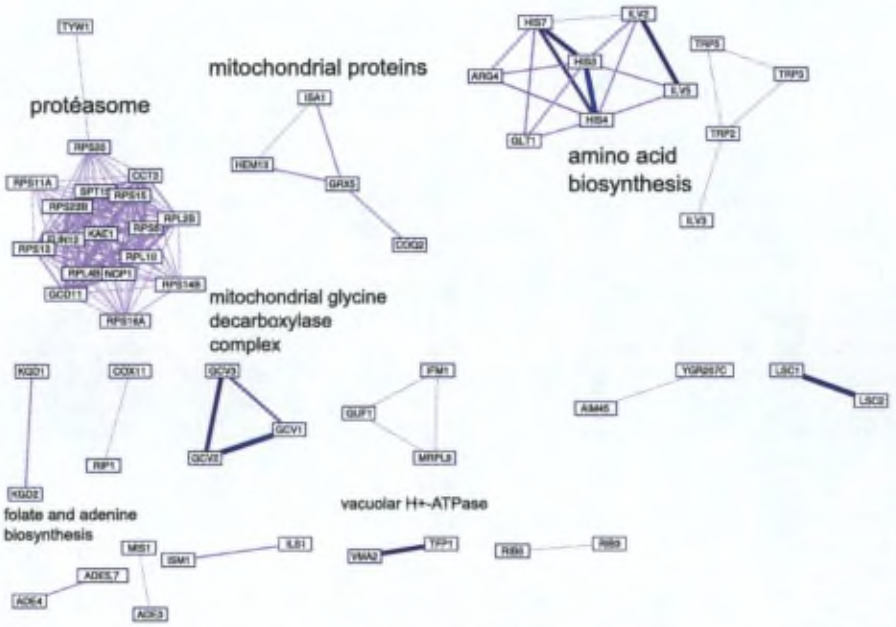


FIG. 9.4 – Réseau d'interactions fonctionnelles de très haute significativité (sig > 30) obtenu par la méthode des profils phylogénétiques (PPV : 71%, Sn : 0,02%). L'épaisseur et l'intensité de la couleur des arêtes sont proportionnelles à la significativité des interactions. Notons l'interaction fonctionnelle détectée entre de *AIM45* et *YGR207C*, deux protéines de fonctions inconnues mais présentant toutes les deux une forte similarité de séquence avec deux sous-unités différentes de la flavoprotéine et pour lesquelles une interaction avec la même protéine Yfh1p a été détectée [166].

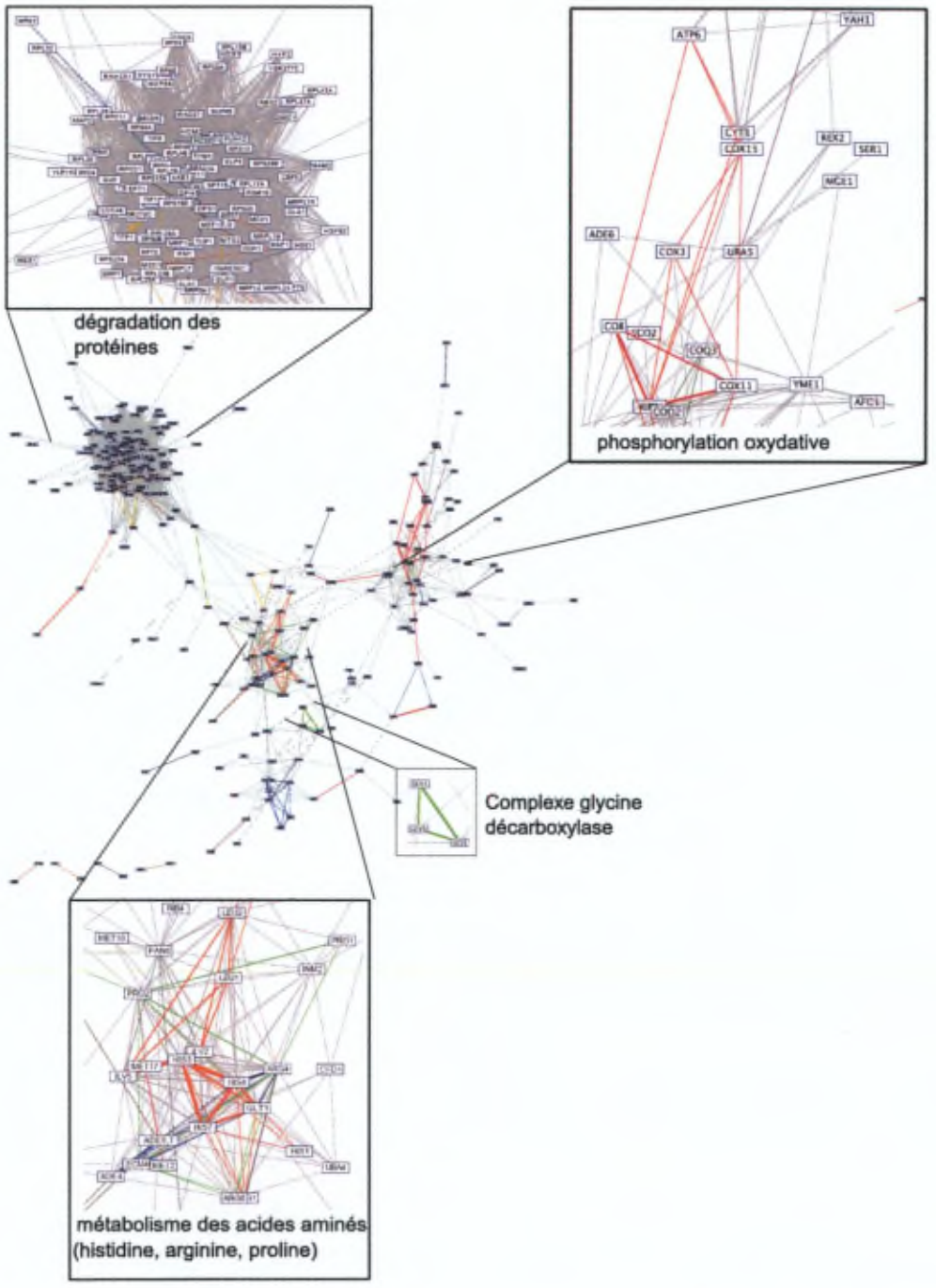


FIG. 9.5 – Réseau d'interactions fonctionnelles de haute significativité (sig > 10) obtenu par la méthode des profils phylogénétiques (PPV : 38%, Sn : 0,3%). L'épaisseur est proportionnelle à la significativité des interactions. Les arêtes colorées correspondent à des interactions fonctionnelles détectées au sein de classes de la base de données de voies métaboliques KEGG.

TAB. 9.2 – Interactions fonctionnelles de significativité supérieure à 0, prédites par la méthode des profils phylogénétiques et impliquant des perméases.

Gène 1	Gène 2	Sig	Description Gène 1	Description Gène 2
HIS7	CCC2	12.08	Protéine impliquée dans la biosynthèse de l'histidine	Perméase ATP-asique intra-cellulaire de cuivre
CCC2	ADE4	11.02	Perméase ATP-asique intra-cellulaire de cuivre	Protéine impliquée dans la biosynthèse de la purine
GLT1	MEP2	10.82	Protéine impliquée dans la biosynthèse du glutamate	Transporteur d'ammonium de la membrane plasmique
CCC2	ECM40	10.16	Perméase ATP-asique intra-cellulaire de cuivre	Protéine impliquée dans la biosynthèse de l'arginine
CCC2	PRO2	10.14	Perméase ATP-asique intra-cellulaire de cuivre	Protéine impliquée dans la biosynthèse de la proline

le grand nombre d'interactions fonctionnelles retournées par notre approche pour ces deux transporteurs soient dûs au grand nombre d'orthologues trouvés pour ces deux gènes dans l'ensemble des organismes que nous avons utilisé (175 pour *CCC2* et 100 pour *MEP2*) (ces deux gènes figurent parmi 3 gènes pour lesquels le nombre d'orthologues retrouvés est le plus important). Il semble en fait que les perméases sont toutes très conservées au sein du groupe taxonomique des *Fungi* mais beaucoup moins dès que l'on sort de ce groupe taxonomique (à part les gènes *CCH2* et *MEP2*) et notre approche de détection des meilleurs hits bidirectionnels ne nous permet pas de retrouver un nombre suffisant d'orthologues pour celles-ci.

9.4 Conclusion

La méthode des profils phylogénétiques est une technique fréquemment utilisée en bioinformatique pour la prédictions d'interactions fonctionnelles. Nous espérions que l'application de cette méthode se révelerait efficace dans notre cas d'étude. Nous avons donc téléchargé et installé sur nos serveurs un grand nombre de génomes fongiques qui se sont révélés trop proche de *S.cerevisiae* notre espèce de référence. L'installation de ces génomes n'a cependant pas été réalisée en vain puisque ceux-ci nous serviront dans la dernière partie de la thèse.

L'utilisation d'un ensemble de génomes procaryotes plus éloigné, le filtrage des génomes redondants et une évaluation rigoureuse nous ont permis de construire un réseau d'interactions fonctionnelles et d'établir le seuil de significativité à partir duquel les prédictions semblaient fiables. Cependant, à ce seuil et même à des valeurs inférieures, aucune prédiction relevante n'était effectuée pour les perméases dont on ne retrouve presque aucun orthologue dans nos données.

Plusieurs possibilités pourraient être envisagées afin de remédier à ce problème.

Notamment, notre méthode permettant de retrouver les gènes orthologues par la méthode des meilleurs hits bidirectionnels est très stringente, nous pourrions nous inspirer de l'approche décrite par Jothi *et al* [59] où chaque protéine est décrite par un profil de score de E-valeur retournées par BLAST et les paires de profils sont comparés en utilisant l'information mutuelle (en n'utilisant pas des valeurs binaires de type présence / absence comme dans notre approche). Ceci devrait pouvoir détecter un plus grand nombre d'orthologues.

Nous pourrions aussi revoir notre ensemble de génomes, d'une part en incluant d'autres organismes ou en affinant encore le filtrage.

Chapitre 10

Etude bioinformatique de la régulation des transporteurs

Dans ce chapitre, nous décrivons tout d'abord une méthode permettant de prédire des relations de co-régulation entre deux gènes de *S.cerevisiae* en n'utilisant que ses séquences génomiques et celle de génomes apparentés. Cette méthode, développée par Jacques van Helden a été évaluée en collaboration avec Rekin's Janky et Fadi Abdel-Sater qui travaille dans le laboratoire du Professeur Bruno André.

Nous appliquons ensuite cette méthode à notre cas d'étude afin de détecter les co-régulations entre perméases ainsi que les gènes co-régulés avec celles-ci.

10.1 L'approche des empreintes phylogénétiques

10.1.1 Découverte de motifs

Nous avons rappelé dans l'introduction comment l'expression des gènes était activée ou réprimée suite à la fixation de facteurs de transcription activateurs ou répresseurs sur les éléments cis-régulateurs des gènes, généralement localisés en amont du gène, dans leur promoteur. Les éléments cis-régulateurs sont des séquences d'ADN généralement courtes (entre 6 et 25 nucléotides), parfois espacées et bien souvent dégénérées. Leur découverte est un domaine de recherche actif en bioinformatique, en effet, un grand nombre de méthodes de découverte de ces éléments de régulation a été développé. A l'origine, ces méthodes s'appliquaient à des groupes de gènes constituant des régulons (groupes de gènes co-regulés) ou extraits d'expériences qui indiquaient leur éventuelle co-régulation (biopuces, par exemple) [167]. Parmi les méthodes utilisées pour la découverte de motifs, nous ne nous attarderons ici que sur *dyad-analysis*, un programme basé sur le comptage de mots et développé par Jacques van Helden [168].

10.1.2 Dyad-analysis

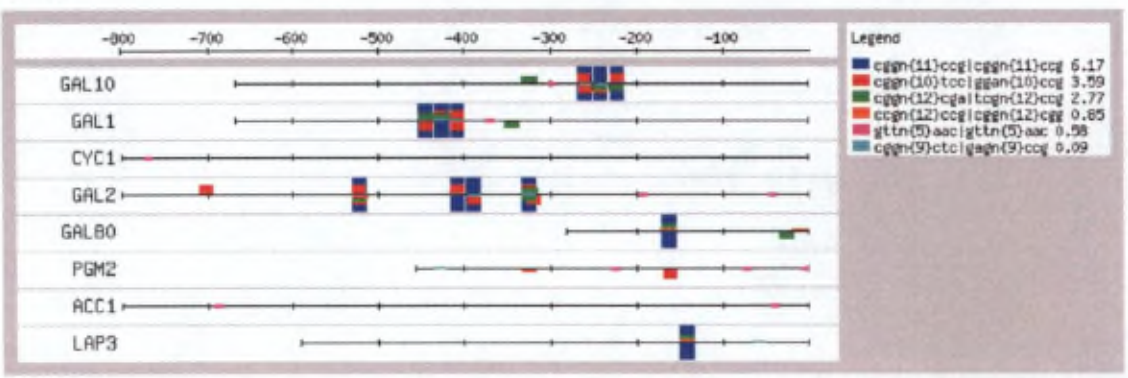
Un *mot* est une séquence nucléotidique sur l'ADN qui est utilisée pour décrire un site de régulation. Par ailleurs, on connait un bon nombre de facteurs de transcription qui forment des dimères et qui établissent deux points de contacts séparés lors de leur liaison à l'ADN. Dès lors, les motifs liés par ces facteurs transcriptionnels sont représentés par des *dyades*, c'est-à-dire des mots courts séparés par une distance fixée.

Les méthodes de comptage de mots comme *dyad-analysis* se basent sur le dénombrement dans une ou plusieurs séquences des occurences de tous les mots décrits selon un modèle de motif donné (taille du mot et de l'espacement). Ils utilisent un test statistique afin de détecter les mots statistiquement sur-représentés par rapport à ce qui est attendu au hasard. Le nombre d'occurences attendu constitue le *modèle de fond* (*background model*), il est estimé sur base des séquences à analyser ou bien sur base d'un jeu de données de séquences de référence. Le choix d'un modèle de fond représentatif de l'espérance aléatoire est essentiel puisqu'il permet d'estimer la significativité statistique de la

TAB. 10.1 – Résultats du programme *dyad-analysis* appliqué aux séquences des promoteurs de gènes régulés par le facteur transcriptionnel Gal4 (*GAL17*, *GAL10*, *GAL1*, *CYC1*, *GAL2*, *GAL80*, *PGM2*, *ACC1* et *LAP3*).

identifier	occ	exp_occ	occ_P	occ_E	occ_sig
cggn{11}ccglcggn{11}ccg	9	0.27	2.00E-011	6.80E-007	6.17
cggn{10}tcclggan{10}ccg	9	0.53	7.50E-009	2.60E-004	3.59
cggn{12}cgaltcgn{12}ccg	8	0.47	4.90E-008	1.70E-003	2.77
ccgn{12}ccglcggn{12}cgg	6	0.39	4.20E-006	1.40E-001	0.85
gttn{5}aaclgttn{5}aac	9	1.23	7.70E-006	2.60E-001	0.58
cggn{9}ctclgagn{9}ccg	6	0.53	2.40E-005	8.00E-001	0.09

(a)



(b)

```
ccgnnnnnnnnnnnnccg. .cggnnnnnnnnnnnncgg 0.85
.cggnnnnnnnnnnnccg. .cggnnnnnnnnnnnccg. 6.17
.cggnnnnnnnnnntcc.. ..ggannnnnnnnnnccg. 3.59
.cggnnnnnnnnnnnncga tcgnnnnnnnnnnnnccg. 2.77
.cggnnnnnnnnnctc... ...gagnnnnnnnnnccg. 0.09
..ggannnnnnnnnccg. .cggnnnnnnnnnntcc.. 3.59
...gagnnnnnnnnnccg. .cggnnnnnnnnnctc... 0.09
ccggagnnnnnnnctccga tcggagnnnnnnnctccgg 6.17 best consensus
```

FIG. 10.1 – Résultat des programmes (a) *feature-map* et (b) *pattern-assembly* de *RSAT* appliqués aux résultats de *dyad-analysis* du Tableau 10.1. La feature-map représente les motifs identifiés par *dyad-analysis* sur les promoteurs des gènes testés et *pattern-assembly* aligne les différents motifs découverts et renvoie un motif consensus. Dans ce cas, ce motif correspond au motif annoté pour le facteur de transcription Gal4p.

sur-représentation des motifs découverts. Notons que *dyad-analysis* peut aussi retrouver aussi les motifs non espacés puisqu'il s'agit des dyades dont l'espacement est de taille 0.

Pour chaque motif découvert, *dyad-analysis* renvoie une P-valeur (occ_P), une E-valeur (occ_E), ainsi qu'une valeur de significativité (occ_sig). A titre d'illustration, le Tableau 10.1 montre les dyades sur-représentées détectées dans les promoteurs de gènes nécessaires à l'utilisation du galactose par la levure et exprimés en présence de galactose dans le milieu de culture (gènes *GAL*).

La P-valeur du motif CGGn{11}CCG (Table 10.1, ligne 1), calculée en utilisant la distribution binomiale est de 2,00E-011. Elle représente la probabilité que le motif soit trouvé 10 fois ou plus (occ) dans une séquence aléatoire alors que le nombre attendu d'occurences de ce mot est 0,31 (exp_occ) selon le modèle de fond. Le choix d'un seuil de P-valeur à partir duquel on considère un motif comme significatif est très important. Dans notre exemple, on teste l'éventuelle sur-représentation de 34 000 mots distincts découverts dans les promoteurs des gènes. Si l'on décidait de considérer l'ensemble des motifs sur-représentés avec une P-valeur de 0.001 ou moins, on s'attendrait au hasard à retrouver 34 motifs. Dès lors, nous calculons la E-valeur, plus fiable que la P-valeur car elle inclut une correction pour les tests multiples. La E-valeur est obtenue en multipliant la P-valeur différents observables dans l'ensemble des séquences soumises au programme (correction de Bonferroni). Elle correspond au nombre de faux positifs attendu pour un seuil donné de P-valeur.

Enfin, la significativité est une transformation logarithmique de la E-valeur $occ_sig = -log_{10}(E - valeur)$, elle est plus facile à appréhender car les valeurs les plus hautes sont associées aux motifs les plus significatifs.

Le plus gros désavantage de ces méthodes est qu'elles recherchent généralement des mots exacts alors que les sites de fixation des facteurs de transcription sont souvent dégénérés. Cependant, la dégénérescence est partiellement détectée sous la forme d'une collection de mots apparentés qui peuvent être alignés. La Figure 10.1(a) reprend une *feature-map* qui représente les motifs retournés par *dyad-analysis* sur les promoteurs des gènes du galactose, on peut observer que les 4 premiers mots découverts par *dyad-analysis* dans le promoteur de certains des gènes révèlent en fait différents fragments du même motif. Ces fragments sont alignés à la figure 10.1(b) et le consensus dégagé par ces alignements constitue le motif annoté du facteur de transcription Gal4p [169].

10.1.3 Les empreintes phylogénétiques

Classiquement, la découverte de motifs est appliquée aux séquences de promoteurs de groupe de gènes (potentiellement) régulés par un même facteur de transcription. Cependant, avec l'augmentation du nombre de génomes séquencés, il est maintenant devenu possible d'appliquer ces méthodes à un ensemble de séquence promotrices de gènes orthologues pour détecter les motifs de régulation conservés. L'hypothèse qui fonde cette approche est que les séquences fonctionnelles (codantes et non codantes) dans un génome ont tendance à rester conservées du fait de la pression de sélection tandis que les séquences non fonctionnelles continuent à diverger [167]. Les séquences conservées sont appelées empreintes phylogénétiques [170].

L'utilisation de *dyad-analysis* pour la détection d'empreintes phylogénétiques de bactéries a récemment fait l'objet d'une évaluation approfondie dans notre laboratoire. Outre l'estimation des perfomances de la méthode, cette étude a permis d'en déterminer les paramètres optimaux à utiliser. En effet, le choix du modèle de fond, du niveau taxonomique considéré et l'application d'un éventuel filtrage des motifs renvoyés par l'algorithme peuvent grandement en affecter les performances [171].

La Table 10.2 représente les résultats de dyad-analysis lorsqu'il est appliqué aux séquences des promoteurs des orthologues du gène *GAL1* dans le groupe taxonomique des *Saccharomycetes* (19 espèces dans *RSAT*). La figure 10.2 montre l'assemblage des motifs découverts les plus significatifs ainsi que le logo constitué à partir de ces séquence. Nous constatons que le motif de fixation de Gal4p est détecté avec une haute significativité. Ceci nous laisse entrevoir la possibilité d'appliquer la même approche à chaque gène d'un génome séparément de façon à prédire les motifs de régulation de tous les gènes.

Puisque nous sommes capables de prédire les élements de régulation en n'utilisant que les séquences génétiques comme information, un nouveau défi serait d'inférer la co-régulation entre les gènes sur base de la prédiction de présence de motifs similaires dans leur promoteur. Ainsi, nous serions capables de reconstruire un réseau de co-régulation entre les gènes *ab initio*.

TAB. 10.2 – Résultat de l'algorithme *dyad-analysis* appliqué aux séquences des promoteurs des orthologues identifiés pour le gène *GAL1* dans le taxon des *Saccharomycetes*

identifier	occ	exp_occ	occ_P	occ_E	occ_sig
cggn{11}ccglcggn{11}ccg	18	0.18	2.70E-023	2.90E-019	18.54
cggn{10}tcclggan{10}ccg	19	0.37	6.00E-019	6.40E-015	14.2
acgn{12}ccglcggn{12}cgt	14	0.34	5.00E-013	5.40E-009	8.27
cggn{12}cgaltcgn{12}ccg	14	0.37	1.30E-012	1.40E-008	7.86
ggan{9}tcclggan{9}tcc	10	0.3	4.80E-009	5.10E-005	4.29
ggan{11}cgaltcgn{11}tcc	13	0.61	6.40E-009	6.80E-005	4.17
cggn{3}acaltgtn{3}ccg	12	0.58	2.60E-008	2.80E-004	3.56
cggn{9}atclgatn{9}ccg	11	0.46	2.60E-008	2.80E-004	3.55
cggn{9}ctclgagn{9}ccg	10	0.41	9.70E-008	1.00E-003	2.99
cggn{2}gaclgtcn{2}ccg	9	0.35	2.20E-007	2.30E-003	2.63
cggn{13}gaglctcn{13}ccg	10	0.45	3.10E-007	3.30E-003	2.48
agan{5}ccglcggn{5}tct	11	0.62	4.50E-007	4.80E-003	2.32
cggn{7}tcclggan{7}ccg	9	0.39	6.90E-007	7.30E-003	2.14
acgn{0}gatlatcn{0}cgt	11	0.69	1.20E-006	1.30E-002	1.9
ccgn{19}gcglcgcn{19}cgg	8	0.29	1.20E-006	1.30E-002	1.89
acan{9}cgglccgn{9}tgt	10	0.55	1.30E-006	1.40E-002	1.86
acgn{10}gaglctcn{10}cgt	10	0.56	1.90E-006	2.10E-002	1.69
ccgn{20}cgglccgn{20}cgg	6	0.14	2.20E-006	2.40E-002	1.63
cgcn{18}ccglcggn{18}gcg	8	0.33	2.80E-006	3.00E-002	1.52
acgn{17}cgglccgn{17}cgt	8	0.34	2.90E-006	3.10E-002	1.51
agan{10}cgglccgn{10}tct	10	0.61	3.60E-006	3.90E-002	1.41
aggn{8}ccglcggn{8}cct	8	0.37	4.10E-006	4.40E-002	1.36
cggn{3}agaltctn{3}ccg	10	0.64	4.40E-006	4.70E-002	1.33
acgn{13}cgaltcgn{13}cgt	9	0.49	4.90E-006	5.20E-002	1.29
acgn{11}tcclggan{11}cgt	9	0.53	7.90E-006	8.40E-002	1.08
agcn{7}cgglccgn{7}gct	8	0.41	8.80E-006	9.40E-002	1.03
acan{5}ccglcggn{5}tgt	9	0.57	1.30E-005	1.40E-001	0.87
agan{4}gcclggcn{4}tct	10	0.73	1.40E-005	1.40E-001	0.84
cggn{1}ggaltccn{1}ccg	8	0.47	1.60E-005	1.70E-001	0.77
cggn{13}gtaltacn{13}ccg	8	0.44	1.90E-005	2.00E-001	0.7
cggn{6}gtclgacn{6}ccg	7	0.33	2.10E-005	2.30E-001	0.64
cgan{15}ccglcggn{15}tcg	7	0.33	2.50E-005	2.70E-001	0.57
ccgn{10}gtclgacn{10}cgg	7	0.34	2.60E-005	2.80E-001	0.55
cagn{4}gcglcgcn{4}ctg	8	0.48	2.70E-005	2.80E-001	0.55
cggn{12}cgclgcgn{12}ccg	7	0.34	2.70E-005	2.90E-001	0.54
cgtn{9}gtclgacn{9}acg	8	0.48	2.80E-005	3.00E-001	0.53
acgn{20}gcaltgcn{20}cgt	9	0.59	2.80E-005	3.00E-001	0.52
tacn{0}ggaltccn{0}gta	9	0.66	3.40E-005	3.60E-001	0.44
acgn{16}cgglccgn{16}cgt	7	0.36	3.40E-005	3.60E-001	0.44
cgcn{15}ccglcggn{15}gcg	7	0.35	3.90E-005	4.10E-001	0.39
ccgn{2}cgglccgn{2}cgg	5	0.16	4.70E-005	5.00E-001	0.3
gttn{15}cgaltcgn{15}aac	10	0.81	4.80E-005	5.10E-001	0.29
atan{1}ggaltccn{1}tat	13	1.47	4.80E-005	5.10E-001	0.29
cgcn{20}ggaltccn{20}gcg	7	0.38	6.70E-005	7.10E-001	0.15
cgcn{5}ggaltccn{5}gcg	7	0.41	6.70E-005	7.10E-001	0.15
ctcn{14}cgaltcgn{14}gag	8	0.55	7.80E-005	8.30E-001	0.08
acan{19}tcclggan{19}tgt	11	1.01	8.60E-005	9.20E-001	0.04
agan{17}caglctgn{17}tct	12	1.25	9.10E-005	9.60E-001	0.02
acgn{11}gcclggcn{11}cgt	7	0.42	9.10E-005	9.70E-001	0.02

(a)

```
tacnnnnnnnnnnnnnccg..   ..cggnnnnnnnnnnnnngta   0.70
tacgga.................   .................tccgta   0.44
.acgnnnnnnnnnnnnccg..   ..cggnnnnnnnnnnnnccgt.   8.27
.acggat................   ................atccgt.   1.90
.acgnnnnnnnnnnntcc...   ...ggannnnnnnnnnnngt.   1.08
..cggnnnnnnnnnnnccg..   ..cggnnnnnnnnnnnccg..   18.54
..cggnnnnnnnnnntcc...   ...ggannnnnnnnnnccg..   14.20
..cggnnnnnnnnnnnccgt.   .acgnnnnnnnnnnnccg..   8.27
..cggnnnnnnnnnatc....   ....gatnnnnnnnnnccg..   3.55
..cggnnnnnnnnnnnnngta   tacnnnnnnnnnnnnccg..   0.70
...ggannnnnnnnnnccg..   ..cggnnnnnnnnnntcc...   14.20
...ggannnnnnnnnntcc...   ...ggannnnnnnnntcc...   4.29
...ggannnnnnnnnnccgt.   .acgnnnnnnnnnntcc...   1.08
....gatnnnnnnnnnccg..   ..cggnnnnnnnnnatc....   3.55
tacggatnnnnnnnatccgta   tacggatnnnnnnnatccgta   18.54 best consensus
```

(b)

FIG. 10.2 – (a) Assemblage des motifs les plus significatifs découverts au Tableau 10.2 dans le promoteur de *GAL1* et dans les promoteurs de ses orthologues. (b) Logo obtenu à partir de cet assemblage. La hauteur de chaque lettre représente la fréquence avec laquelle le nucléotide correspondant est retrouvé dans les séquences. On peut observer qu'il s'agit bien du motif annoté pour Gal4 (cggn{11}ccg) qui est découvert.

10.2 Description de l'article présenté

Sylvain Brohée*, Rekin's Janky*, Fadi Abdel-Sater*, Gilles Vanderstocken, Bruno André et Jacques van Helden. Unraveling networks of co-regulated genes on the sole basis of genome sequences. Soumis.

Dans cet article, nous proposons une nouvelle méthode basée sur la découverte de motifs par la technique des empreintes phylogénétiques pour inférer un réseau de co-régulation entre gènes chez *S.cerevisiae* en reliant au sein du réseau les gènes présentant des motifs similaires. Le réseau de co-régulation prédit est ensuite évalué et plusieurs prédictions sont validées expérimentalement.

10.2.1 Méthode utilisée

En utilisant les séquences de 19 espèces de *Fungi* (dans le taxon des *Saccharomycetes*), nous avons découverts les empreintes phylogénétiques de chacun des gènes de *S.cerevisiae* correspondant à d'éventuels motifs de régulation. Au total, nous avons pu prédire au moins un motif significatif (sig >= 0) pour 95% des gènes de *S.cerevisiae*.

Notre hypothèse est que des gènes présentant les mêmes motifs dans leur promoteur présentent de bonnes chances d'être régulés par le même facteur de transcription et donc d'être co-régulés.

Avec les prédictions, nous avons construit une matrice gène / motif M où chaque ligne i représente un gène et chaque colonne j un motif découvert. Chaque cellule M_{ij} de la matrice contient la significativité de la sur-représentation calculée par *dyad-analysis* pour le motif j dans l'ensemble des promoteurs orthologues du gène i ou 0 si la significativité était inférieure à 0.

Nous avons ensuite mesuré les similarités entre les paires de profils (lignes de la matrice) et inféré un réseau contenant 3171 gènes reliés par 56919 relations de co-régulations. Quatre métriques de similarité ont été testées (coefficient de Jaccard, information mutuelle, significativité hypergéométrique, produit scalaire) et nous avons retenu le logarithme du produit scalaire (que nous avons appelé DPbits) pour les analyses ultérieures car celui-ci présentait le

principal avantage de prendre en compte la significativité des motifs découverts contrairement aux autres métriques qui reposent sur des profils binaires.

10.2.2 Evaluation du réseau de co-régulation

10.2.2.1 Evaluation bioinformatique

Nous avons évalué la qualité de nos prédictions en comparant le réseau prédit à d'autres réseaux de référence : (1) un réseau de co-régulation construit à partir d'un ensemble de régulons annotés [172], (2) un réseau de co-expression construit à partir de données de biopuces extrait de la base de données STRING [144] et (3) un réseau de co-régulation construit à partir d'une étude à haut débit ayant caractérisé les régions intergéniques où se fixent un ensemble de facteurs de transcription [173] (nous relions les paires de gènes dont les promoteurs sont reconnus par les mêmes facteurs). Le principe de ces comparaisons a déjà été décrit au chapitre précédent (section 9.1.4 p. 129). Dans les trois cas, nos prédictions en terme d'arêtes, ont une valeur prédictive positive élevée qui augmente quasiment linéairement dès lors que l'on considère des valeurs de DPbits plus élevées. Cependant, la sensibilité en terme d'arêtes reste très faible.

Comme à eux seuls les trois plus gros régulons testés dans notre approche représentent 40% du reseau de référence, il est raisonnable de penser que cette faible sensibilité est en partie due à un effet de cliques. C'est ce que nous avons vérifié en n'étudiant plus la sensibilité en fonction des arêtes mais en fonction des noeuds. Pour chaque régulon, nous avons considéré la proportion des gènes du régulon reliés à au moins un autre gène du régulon. Bien que notre approche semble moins apte à la détection des gros régulons, la sensibilité observée dans ce cas est bien meilleure et certains régulons sont même entièrement retrouvés.

Enfin, notre réseau inféré et le réseau de coexpression de STRING montrent les mêmes performances globales quand on les compare au réseau de référence construit à partir des régulons annotés. Cela semble indiquer que notre réseau, pourtant inféré uniquement à partir des séquences, retourne des résultats globalement aussi bon qu'un réseau construit à partir d'un ensemble de plusieurs centaines d'expériences de biopuces.

10.2.2.2 Résultats expérimentaux

Une analyse méthodique des prédictions de co-régulation de gènes dont la régulation était encore inconnue nous a permis d'identifier plusieurs candidats à une validation expérimentale. Nous avons entre autres porté notre choix sur 14 gènes co-régulés avec des gènes codant pour des protéines impliquées dans le métabolisme des stérols (régulés par le facteur de transcription Upc2p), dans la dégradation des protéines (Rpn4p), dans le métabolisme azoté (Gzf3p) et dans le cycle cellulaire (Upc2p). Des expériences de RT-qPCR et de chip-PCR ont confirmé nos prédictions.

10.2.3 Conclusion

Alors que le séquençage des génomes d'organismes souvent peu caractérisés augmente de façon exponentielle, le paradigme actuel pour l'annotation des gènes repose sur la transposition de la fonction des gènes hommologues déjà décrits dans quelques organismes bien étudiés. Intrinsèquement, cette méthode ne permet pas de rendre compte des divergences acquises lors des processus évolutifs. Des méthodes non basées sur des connaissances préalables permettant d'effectuer des prédictions sont donc plus que nécessaires. Dans cette optique, nous avons développé et évalué une méthode capable d'inférer avec précision un réseau de co-régulation entre gènes.

Les performances de l'approche devraient encore augmenter avec le nombre de génomes à notre disposition dans les prochaines années.

Unraveling networks of co-regulated genes on the sole basis of genome sequences

Sylvain Brohée[1*], Rekin's Janky[1*], Fadi Abdel-Sater[1,2*], Gilles Vanderstocken[1], Bruno André[2] and Jacques van Helden[1§]

[1]Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe). Université Libre de Bruxelles (ULB), CP 263, Campus La Plaine. Boulevard du Triomphe, 1050 Bruxelles, Belgium.

[2]Laboratoire de Physiologie Moléculaire de la Cellule. Université Libre de Bruxelles (ULB), CP 300, Campus de Gosselies. Rue des Professeurs Jeener et Brachet, 6041 Gosselies, Belgium.

[*] These authors contributed equally to this work

[§] Corresponding author

Email addresses:

SB: sylvain@bigre.ulb.ac.be

RJ: rekins@bigre.ulb.ac.be

FA : fabdelsa@ulb.ac.be

GV : gvdstock@ulb.ac.be

BA: bran@ulb.ac.be

JvH: Jacques.van.Helden@ulb.ac.be

Abstract

Transcriptional regulation is mediated by the binding of transcription factors to specific cis-regulatory elements With the availability of hundreds of sequenced genomes, regulatory signals can now be revealed by detecting conserved motifs in promoters of orthologous genes (phylogenetic footprints). A next challenge is to unravel the complete regulatory network. Here we propose a method that takes as sole input the genome sequences of a few taxonomically related organisms, predicts regulatory elements for each gene, and builds a co-regulation network by linking pairs of genes having similar footprints. This methods permits to detect co-regulated genes, supposedly involved in the same function. In the yeast, we predict a network of co-regulation involving 56,919 relationships between 3,171 genes. We validate the reliability of the method by comparing predictions with three reference datasets, showing that the predicted network is comparable to a co-expression network built from hundreds of microarray experiments. We validate by RT-qPCR 14 out of 16 selected predictions concerning orphan genes. This approach can be applied to depict regulatory networks of microbial genomes for which no other information is available than the genome sequence. Long-term benefits can easily be perceived when considering the zillions of bases resulting from genome and metagenome projects.

Introduction

The availability of several hundreds of genomes opens the exciting perspective to decipher the wiring of transcriptional regulatory networks in many organisms for which regulation has been barely studied so far, and to analyze the evolution of these networks. The prediction of cis-acting elements from non-coding sequences was first applied at the level of single genomes, by detecting over-represented motifs in clusters of co-regulated genes [1-9]. More recently, multi-genome approaches have been developed to detect phylogenetic footprints, i.e. conserved elements in promoters of orthologous genes [10-19]. After having detected cis-acting elements for each gene of a given organism, the next challenge is to infer regulons (groups of genes regulated by a same transcription factor (TF)), in order to decipher the wiring of the transcriptional control of an organism. One approach has been to apply a gene-wise footprint discovery [20], followed by a partitioning of the discovered motifs into clusters, using specific algorithms such as PROCSE [21] or BMC [22]. Such regulon inference approaches have initially been tested with a few proteobacterial genomes, and the results have been shown to match known regulons. A systematic assessment of the predictive power of such network inference methods is however still missing. In addition, previous approaches were essentially producing groups of genes based on motif clustering, and thereby failed to capture the reticulate properties of the actual regulatory network, where genes can be regulated by multiple factors.

In this paper, we propose a new method to directly infer co-regulation networks from genome sequences without any prior knowledge about regulation: we first apply a gene-wise footprint discovery algorithm [19] to predict putative transcription factor binding motifs, and we then infer a co-regulation network by linking genes with similar footprints. We performed a systematic *in silico* evaluation of the results by comparing inferred groups of co-regulations discovered in the genome of *Saccharomyces cerevisiae* to three reference data sets: annotated regulons [23,24], the STRING co-expression network derived from microarray data [25], and TF-gene associations detected by genome-wide location analysis [26]. We further proceed to an experimental validation of a selected set of predictions, confirming the validity of 12 predicted regulations. The high predictive value of the method opens exciting perspectives for the discovery of regulatory interactions in poorly characterized organisms.

RESULTS

Inferring a co-regulation network by linking genes with similar phylogenetic footprints

For each gene of *Saccharomyces cerevisiae*, we identified putative orthologs in 19 species of the class *Saccharomycetes*, collected their upstream non-coding sequences, and discovered phylogenetic footprints as in [19]. The method relies on the detection of over-represented spaced motifs (dyads). Figure 1a,b shows examples of footprints discovered in the promoters of two genes *LYS1* and *LYS12*, involved in lysine biosynthesis. The significant dyads (Figure 1a) reveal various fragments of the motifs bound by the transcription factors Lys14p (lysine regulator) and Gcn4p (general control of amino acid metabolism), respectively. We successively applied the same procedure to each one of the 5,876 protein-coding genes of *S.cerevisiae*. In total, the footprint discovery algorithm returned at least one significant dyad for 5,585 genes (95%), among which 1,638 (27.8 %) gave a high-confidence prediction (sig_B >= 3, see Methods for the description of the metrics).

Gene-wise discovered footprints can be summarized in a matrix of *dyad significance profiles*, where each cell gives the binomial significance (sig_B) of a given dyad (column) for a given gene (row). A section of this gene/dyad matrix presented in Figure 1c illustrates that genes involved in a similar function (e.g. ergosterol, lysine or histidine biosynthesis) are generally characterized by similar dyads, albeit with different significance values. Note that in the complete matrix (5,585 genes x 33,276 dyads), a dyad that appears specific for a functional class will also be occasionally found in other genes. We thus need to define a criterion to infer co-regulation between genes based on the similarities between their dyad significance profiles. Four alternative metrics were evaluated to score the similarity between two dyad significance profiles: Jaccard similarity (*JS*), Hypergeometric significance (sig_H), mutual information (*MI*), and dot product bits

(*DPbits*, see Figure 1d). The comparison between the different metrics is presented in Supplementary Figure S6, showing an almost perfect correspondence between *MI* and sig_H, and strong differences between all the other pairs of metrics. For the remaining of the discussion, we only retained the DPbits score (Figure 1d), on the basis of its good predictive power (Supplementary Figure S7), and because it is the only metric that takes into account the actual values of sig_B (all the other considered metrics rely on binary profiles).

An inferred gene-gene co-regulation network is then obtained by tracing an edge between each pair of genes, weighted according to the similarity score (*DPbits*). Figure 1e displays the co-regulation network built from the significance matrix of Figure 1c, showing that groups of functionally related genes are densely interconnected by highly weighted edges. Interestingly, the inferred network also contains inter-module connections revealing the general control of three amino acid pathways (leucine, histidine and lysine) by the transcription factor Gcn4p. This sample network thus reflects the capability of our approach to detect relationships between genes regulated by multiple transcriptional factors. We also notice a separate module regrouping 3 *ERG* genes with the orphan gene *YHL026C* indicating that these genes may be regulated by the same regulator. This hypothesis will be tested in the section Experimental validation.

Figure 1 only shows an arbitrary subset of genes and dyads selected for illustrative purposes. By applying the same approach to all the yeast genes, we inferred a network linking 3,171 genes by 56,919 predicted co-regulations (Table 1, Supplementary Figure S8). Figure 2a shows the high-scoring slice (DPbits >= 5) of the predicted co-regulation network.

Reference co-regulation networks: co-regulation, co-expression and co-binding

In order to evaluate the reliability of the inferred network, we compared it to several reference networks: (1) annotated regulons (TRANSFAC database complemented by annotations from Simonis [24]); (2) co-expression network obtained from the STRING database; (3) genome-wide location analysis characterized with the ChIP-chip method for 155 transcription factors [26]. The sizes of these networks are summarized in Table 1.

Regulons are usually represented as a "factor-gene" network, where the arcs link transcription factors to their target genes (e.g. see [27]). For the sake of comparison with the inferred co-regulation network, we converted the annotated factor-gene network (Figure 3a) into a gene-gene network, hereafter denoted as "annotated co-regulation network" (Figure 3c), by linking any two genes regulated by a common transcription factor. Similarly, we converted Harbison's ChIP-chip TF-gene data set [26] into a gene-gene network linking each pair of genes whose promoters are bound by a common factor ("co-binding network").

The conversion of the TF-gene network into a gene-gene (co-regulation) network has an important effect: in the co-regulation network, a regulon of G genes is represented by a clique (i.e.

a fully connected subgraph) with $E=G(G-1)/2$ edges. The number of edges per regulon thus increases in a quadratic way with the number of target genes, so that highly connected TFs (global transcription factors) generate very large cliques. For example, the 54 target genes of the factor Zap1p are represented in the gene-gene network by a clique containing (54x53)/2=1,431 edges. In total, the three most general transcription factors (Msn2p, Msn4p and Zap1p) cover 14% of the nodes (genes) and 34% of the edges (co-regulations) in the annotated co-regulation network. Consistently, whereas the degree distribution of the factor-gene network grossly follows a power law (Figure 3b), the gene-gene network shows a high frequency of highly connected nodes, due to the cliques formed by the numerous target genes of global factors (Figure 3d, Supplementary Figure S9).

Top-scoring predictions correspond to annotated co-regulated genes

We compared the highest scoring predictions with the annotated network. For this evaluation, we selected all the co-regulations predicted between any pair of genes found in the database of annotated regulons. The 60 gene pairs having the highest DPbits score are shown in Table 2. A large majority of those gene pairs (44/60=73%) are already annotated as member of a common regulon (highlighted in bold with green background). In addition, 13 other gene pairs (22%) are involved in some common function (italics with yellow background), suggesting that they may correspond to actual co-regulations, despite not being annotated in the regulon dataset. Only 3 (5%) of the 60 top-scoring predictions have no prior evidence of co-regulation. These may either correspond to false predictions, or to actual regulations that are not yet documented. Noticeably, two out of these three gene pairs *(LEU1-TPO1 and LEU1-ACO1)* involve the gene *LEU1*, and the Leu3p binding motif is detected with a high significance score in their footprints, suggesting them as potential targets for leucine regulation.

Inferred co-regulations have a good positive predictive value

In the previous paragraph, we focused on the 60 top-scoring predictions. We will now apply a systematic evaluation of the predicted co-regulations by estimating two parameters: (1) sensitivity (*Sn*), i.e. the fraction of known co-regulations that are detected by the predictive method; (2) positive predictive value (*PPV*) i.e. the fraction of predictions corresponding to actual co-regulations. Figure 2b displays the 2,227 interactions (DPbits >= 1) predicted between the 612 genes of the regulon reference set. This sub-network was used for further validation. Figure 2c shows that the number of predicted co-regulations decreases rapidly when the threshold on DPbits increases (blue curve). Interestingly, the number of true predictions (green) decreases slower than the false predictions (red curve). This trends is confirmed by the Sn/PPV curves (Figure 2d for the DPbits; Supplementary Figure S7 for the other metrics), showing that the reliability of the predicted co-regulations increases with the score, raising 50% with DPbits>= 5.

This increase in predictive value is obviously at the cost of sensitivity: when the threshold on DPbits increases from 1 to 5, the genome-wide inferred network drops from 56,919 to 11,494 predicted co-regulations. Among these, the sub-network connecting nodes from the reference subset (genes with annotated regulation) drops from 2,227 (DPbits >= 1) to 648 (DPbits >= 5) predicted co-regulations. It should be noted that transcription factor databases only cover a subset of the published literature, which itself reflects a fraction of the existing regulatory interactions. It is thus likely that some fraction of our predictions are unduly counted as false positives in our evaluation (Figure 2b,c,d). In conclusion, a large fraction of the 11,494 high-scoring predictions (DPbits >= 5) displayed in Figure 2a are likely to correspond to effective co-regulations.

The sensitivity of the predictions strongly depends on regulon size

The sensitivity computed on the whole annotated network seems rather weak: even at low DPbits score (DPbits >= 0), the inferred network covers no more than 723 among the 10,599 arcs derived from annotated regulons (6.8%). This apparent lack of sensitivity is largely due to a few highly connected TF, which generate large cliques in the reference network (e.g. Zap1p, Gcn4p in Figure 4a; Supplementary Table S3). It should be noted that regulon inference does not require a full coverage of the co-regulation edges. For example, we can see on Figure 4a that the footprint detection covers 69% of the edges between Met4p target genes, but that these edges are sufficient to establish a dense network of links between 100% of the target genes. Similarly, the 13% of edges detected in the Gcn4p regulon cover 67% of its target genes. In complement to the edge-wise sensitivity (Figure 4b), we can define a node-wise sensitivity (Figure 4c), indicating the fraction of target genes that are linked together for a given transcription factor. Not surprisingly, node-wise sensitivity gives higher estimates than edge-wise sensitivity. Both the edge-wise and node-wise estimations plots show that the sensitivity strongly decreases when regulon size increases. All these observations confirm that our method predicts better the specific regulons than the global regulons.

Co-regulation, co-expression and co-binding

We extended the evaluation of the inferred co-regulation network by comparing it with the network of co-expression downloaded from the STRING database. This network was built by linking genes having similar expression profiles in a compendium of microarray experiments, and can thus directly be compared to our inferred network.

The Sn/PPV curves (Figure 2e; Supplementary Figure S7) show the same general trends as discussed above for the annotated co-regulation network: the overall sensitivity is rather weak (10.7% for DPbits >= 0 versus 5.7% in average in random networks), but the PPV is rather good (36.3% at DPbits >= 0 versus 19.5% at random), and increases with the DPbits score (51% for

DPbits >= 5, versus 21.8% in average in random networks), showing that similarities between predicted phylogenetic footprints reliably predict co-expression.

It is well known that expression data is quite noisy, and there are many issues about the choices of normalization, similarity measurements and clustering procedures. We thus wondered to which extend the STRING co-expression data set would correspond to the annotated co-regulation network. To answer this question, we computed Sn/PPV curves (Figure 2f) as described above. Interestingly, the STRING co-expression network shows similar figures of merit as our inferred co-regulation network: the *PPV* is reasonably good (~40% of the STRING co-regulation fits some annotated regulon), and increases with the score, whereas the sensitivity is rather poor (at most 7.5%), and rapidly drops when the STRING score increases above 170. It thus seems that the network inferred by *in silico* analysis of phylogenetic footprints gives more or less as good results as a compendium of several hundreds of microarray experiments.

We also compared our inferred co-regulation network to the network of "co-binding" derived from the genome-wide location results obtained by applying the ChIP-chip method for 155 yeast transcription factors under various culture conditions [26]. When compared to this high-throughput reference set, our inferred co-regulation network shows similar PPV curves as for annotated regulons, but a lower sensitivity (Figure 2g). This reduced sensitivity, also observed when comparing the STRING co-expression network with the co-binding network (Figure 2h), is not surprising since the high-throughput ChIP-chip data produce a huge co-binding network of 178,202 edges, i.e. ~17 times larger than the annotated co-regulation network (Table 1).

In summary, the inference of co-regulation and the analysis of co-expression seem to reveal complementary parts of the annotated co-regulation network.

Experimental validation of selected predictions

After having verified that high-scoring predictions generally reveal pairs of co-regulated genes, we can use the complete inferred network (Figure 2a) to identify a reasonable set of testable hypotheses about regulations. For this, we use a "guilt by association" approach as we will assign a hypothetical function and/or regulation to a gene of interest on the basis of its links to other genes in the predicted co-regulation network. We applied this principle to each gene of unknown function (orphan), by collecting a cluster with their direct neighbours in the inferred co-regulation network, and retaining the clusters showing a significant enrichment for some functional class of Gene Ontology (GO) [28]. On this basis, we selected a reasonable number of clusters with one or more orphan genes and carried out experiments to validate their co-regulation experimentally (Figure 4; Supplementary Table S4).

The first neighborhood cluster regroups 48 genes, 16 of which are involved in uptake and biosynthesis of sterols (*ERG* genes) and 3 in sphingolipid biosynthesis (Figure 4a). In cells

deprived for sterols, these genes are up-regulated by Upc2p, a binuclear zinc cluster transcription factor binding to the sterol response element (SRE) TCGTATA [29]. The inferred cluster included 12 ERG genes, linked to several orphan genes among which *YDR531W*, *YHL026C* and the divergently transcribed *YML082W-YML083C* gene pair . The most significant dyads detected in the footprints of those genes form the motif TCGTTTag (Supplementary Table S4), which corresponds to the annotated SRE. Davies *et al.* [30] have previously shown that sterol depletion can be induced by Lovastatin, a competitive inhibitor of 3-hydroxy-3-methylglutaryl-coenzymeA reductase. We grew wild-type and upc2Δ mutant cells with or without Lovastatin and used qRT-PCR to monitor the relative transcript levels of these genes. We used *ERG2* as a positive control and the actin gene (*ACT1*) as a reference. As expected, the results clearly show Upc2-dependent induction of *ERG2* in response to Lovastatin. The four tested orphan genes displayed a similar expression profile (Figure 5a), thereby confirming that they belong to the Upc2 regulon.

Rpn4p is a transcriptional activator binding to the GGTGGCAAA sequence present in the upstream region of most genes encoding proteasome subunits. In an rpn4Δ mutant, the expression level of these genes is typically reduced about two-fold [31]. The inferred co-regulation network contains a cluster of densely connected genes coding for proteasome subunit genes, together with the two orphan genes *YNL155W* and *YOR052C* (Figure 2a; Figure 5b), and whose footprints include the Rnp4p motif (Supplementary Table S4). The results of qRT-PCR experiments clearly showed that both genes are expressed in a manner very similar to the proteasomal subunit *RPN2* gene, with a reduced expression in the rpn4Δ mutant (Figure 5b). Interestingly, both *YNL155W* and *YOR052C* products contain a conserved AN1-type zinc-finger motif also present in AIRAP, an arsenite-inducible protein recently found to be associated with and to stimulate the 19S regulatory particle of the proteasome, thereby counteracting the toxic effect of arsenic on protein structure [32]. Furthermore, *YNL155W* is also up-regulated by arsenic [33]. These data strongly suggest that Ynl155wp and Yor052cp represent possible functional orthologs of the human protein AIRAP.

The expression of many genes involved in nitrogen catabolism decreases to low levels when readily used nitrogen sources (e.g. glutamine) are provided in the growth medium. This regulation termed nitrogen catabolite repression (NCR) is mediated by a combination of three negative factors (Ure2p, Gzf3p, Dal80p) preventing the trans-acting factors Gln3p and Nil1p from activating transcription through cis-acting sequences containing GATAA core [34]. The inferred network corresponding to NCR-target genes includes two divergently transcribed gene pairs each containing a known NCR-target (*GUD1* and *CHA1*). We were interested in determining whether each associated gene (*YDL237W* and *VAC17*, respectively) is also under NCR control as they had not been fished out by any previous genome-wide study aimed at inventorying NCR-target genes [35-37]. We thus used qRT-PCR to monitor the relative RNA levels of the four genes (Figure 5c). As expected, *GUD1* and *CHA1* were derepressed in ure2Δ gzf3Δ dal80Δ triple mutant cells

grown on glutamine, an expression profile typical of NCR-target genes [36]. The *VAC17* and *YDL237W* genes were also significantly derepressed in the triple mutant strain, though to a lower extent than the *GUD1* and *CHA1* genes. As Vac17p is a vacuole-specific receptor for myosin (Myo2p), it remains uncertain whether its apparent weak NCR control is physiologically relevant.

Finally, we examined four other genes of unknown function (*YDL010W*, *YDL012C*, *YJL181W* and *YDL156W*) from an inferred cluster regrouping a set of Mbp1 target genes (Figure 5d). Mbp1p is a DNA-binding protein associating with Swi6p to form the MBF complex (Mlu1 cell cycle box [MCB] Binding Factor) involved in transcriptional control of several genes during the G1/S transition [38]. Although these four genes have never been shown to be regulated by Mbp1p, their expression profiles show similarities to other genes under MBF control [39]. To determine whether these four genes represent actual MBF target genes, we performed a ChIP-PCR analysis to monitor their possible association with an HA-tagged Mbp1 protein. The *RNR1* gene was used as positive control [26] and *GAL1* as a negative control. The results (Figure 5d) shows that Mbp1p associates with the upstream regions of *RNR1*, *YDL156W* and *YJL181W* and possibly also of *YDL010W* and *YDL012C*. These genes thus represent probable new Mbf1p target genes, and their products are likely to be involved in G1/S transition.

We also tested the response to stress in amino acid for a pair of divergently transcribed genes YHR020W and DED81, which were grouped with many Gcn4p targets, but the result was negative for each of these.

In summary, by combining predicted regulatory elements and gene ontology annotation, we could suggest a hypothetical regulation and function for 16 orphan genes, among which 14 were confirmed by PCR experiments.

Discussion

This study presents a new method to predict cis-regulatory elements for individual genes and to infer co-regulation between pairs of genes by using, as only input, a set of completely sequenced genomes of related organisms. Importantly, the method relies on *ab initio* pattern discovery in promoter sequences of orthologous genes (phylogenetic footprint discovery), and does not require any prior knowledge about regulation, such as transcription factor binding motifs or gene expression data. By comparing predicted groups of co-regulated genes to functional classes (e.g. GO classes) a putative function and regulation can further be assigned to completely uncharacterized genes.

The co-regulation network inferred in the yeast *Saccharomyces cerevisiae* was extensively evaluated in various manners. We first showed that almost all of the top-ranking predictions correspond to pairs of genes previously known to be co-regulated or involved in the same function. We then performed a systematic *in silico* validation of the reliability of various predictive

scores (DPbits, Jaccard similarity, hypergeometric significance and mutual information), and showed that the DPbits score gives a good indication of the reliability of the predictions. The comparison between inferred groups of co-regulations and annotated regulons showed a good capability to infer links between the target genes of specific transcription factors, but a weakness to detect large regulons controlled by global transcription factors. It has been shown recently that global regulators consistently display low levels of binding specificity [40] and consequently the binding sites of a given TF can differ from each other, so that even if they were perfectly predicted, it might still be very difficult to link them. Noticeably, although our method only relies on genome sequences, its figures of merit are comparable to those of the STRING co-expression network, which was derived from several hundreds of microarray experiments. Finally, we selected 12 interesting cases of orphan genes appearing to be linked to some regulons, and all of these predictions were confirmed by experimental testing.

An important strength of our approach is that the binomial significance returned by the footprint discovery algorithm provides a reliable estimate of the risks of false positive [19], thereby enabling to apply more stringent criteria to select the most reliable motifs. Similarly, we show that the DPbits score ranks the infer co-regulations in a relevant way, with a steady enrichment in correct predictions with increasing scores.

As shown by our experimental validations, in silico predicted co-regulations provide testable hypotheses about gene regulation, thereby offering a concrete way to assign a putative function to orphan genes. This method can readily be applied to the hundreds of completely sequenced microbial genomes for which no other information is available than the raw genome sequence, and its long-term benefits can easily be perceived to catch the pace of the zillions of bases resulting from genome and metagenome projects.

METHODS

Genomes

Fungal genomes were downloaded from NCBI (http://www.ncbi.nlm.nih.gov/ [41]), from the web site maintained by Jason Stajic (http://fungal.genome.duke.edu/) and from the BROAD institute website (http://www.broad.mit.edu/) : Yarrowia lipolytica CLIB99, Candida lusitaniae, Debaryomyces hansenii CBS767, *Eremothecium gossypii*, *Kluyveromyces lactis NRRL Y-1140*, *Kluyveromyces waltii*, *Lodderomyces elongisporus*, *Candida guilliermondii*, *Saccharomyces bayanus*, *Saccharomyces castellii*, *Saccharomyces cerevisiae*, *Saccharomyces kluyveri*, *Saccharomyces kudriavzevii*, *Saccharomyces mikatae*, *Saccharomyces paradoxus*, *Candida albicans*, *Candida glabrata CBS138*, *Candida tropicalis*, *Candida dubliniensis*.

Reference networks

The co-regulation reference network was constructed from the annotated regulons collected from the TRANSFAC database and complemented by Simonis [24]). It was built by linking each pair of genes belonging to the same regulon. The co-expression reference network was obtained from the STRING database (http://string.embl.de) [25], and was built by linking pairs of genes detected as co-expressed in a compendium of microarray experiments. The co-binding reference network uses a genome-wide location analysis characterized with the ChIP-chip method [26]. For this purpose we use the list of genes/ORFs bound by each TF and a p-value < 0.001 (release 24.3). This network was built by linking any pair of genes whose promoters is bound by the same transcription factor. The sizes of these networks are summarized in Table 1.

Ortholog detection

We consider as putative orthologs the bi-directional best hits (BBH) with an E-value smaller than 10^{-5}. Pairwise similarities between gene products were detected by running BLAST with all the translated sequences of *S.cerevisiae* against those of each genome of the reference taxonomical group (Saccharomycetes).

Phylogenetic footprint detection

Starting from a set of input genes and their orthologous clusters, the upstream sequences were collected from the start codon up to the upstream neighbor gene, with a maximal length of 800bp. These distances were chosen based on our previous analysis [4] of the distributions of annotated sites in the TRANSFAC [23].

For each gene of *S.cerevisiae*, we collected the promoters of all orthologs in Saccharomycetes. Promoter sequences were purged to mask redundant fragments larger than 30bp. These purged sequences were used to detect over-represented motifs, using the pattern-discovery program *dyad-analysis* [7]. We counted the occurrences of all pairs of trinucleotides separated by a spacing comprised between 0 and 20 base pairs (a dyad with a spacing of 0 corresponds to an hexanucleotide). The program *dyad-analysis* assesses the significance of each dyad by comparing the observed occurrences with those expected by chance, according to some background model. The prior probability of each dyad was estimated by computing the frequency observed for this dyad in the promoters of all genes of all organisms of the reference taxon ("taxfreq" model[19]).

The program *dyad-analysis* estimates the P-value of each dyad, according to the binomial probability ($Pval_b$). A multi-testing correction is then applied by computing an E-value ($Eval_b=Pval_bD$), where D is the number of distinct dyads analyzed in one set of promoters, and the E-value is converted to a significance score $sig=-log_{10}(Eval_b)$. For each gene, we selected all the dyads with $sig >= 0$, which corresponds to an upper threshold of 1 on the E-value.

An organism-specific filtering was applied by selecting only the dyads found in the promoter of *S.cerevisiae*. This option has a double effect of lowering the rate of false positives (by filtering out dyads that are irrelevant for the promoter of the query organism) and increasing the sensitivity (by strongly lowering the correction term *D* in the computation of the E-value, since the analysis is restricted to a few tens or hundreds of dyads among the 43,680 possible dyads).

Dyad significance matrix

The dyad significance matrix was built by collecting the significant dyads detected in promoters of orthologs for each yeast gene.

$$\mathbf{x} = x_{gj}$$

where $x_{g,j}$ indicates the binomial significance of the dyad *j* for the gene *g*.

Similarity between dyad significance profiles

Let *a* and *b* be two genes. We denote hereafter as *A* and *B* the sets of significant dyads (i.e. dyads with a positive binomial significance) in their respective profiles, with N_a=|*A*| and N_B =|*B*| the number of significant dyads. Let *C* be the intersection between the sets *A* and *B*, with N_c =|*C*| be the number of common dyads between profiles of genes *a* and *b*. We further define N_D as the total number of dyads in the significance profiles of all the genes (N_D=*22,565* for the yeast data set).

Jaccard similarity (*JS*)

The Jaccard similarity (*JS*) is a classical metrics for comparing two sets *A* and *B*. It is computed as the ratio between the intersection and the union of the two sets.

$$JS_{AB} = \frac{A \cap B}{A \cup B}$$

Hypergeometric significance (sig_H)

The hypergeometric P-value ($Pval_H$) indicates the probability to observe at least N_c dyads in common between two random selections of sizes N_a and N_b, respectively.

$$\mathrm{Pval} = P(X >= N_c) = \sum_{i=N_c}^{\min(N_a,N_b)} \frac{C_{N_a}^{i} C_{N_D-N_a}^{N_b-i}}{C_{N_D}^{N_b}}$$

$Pval_H$ can be interpreted as the risk of false positive for one given pair of genes, i.e. the risk to erroneously consider as significant the intersection between the sets (*A* and *B*) of dyads selected in their respective phylogenetic footprints. Since the same significance test is performed

successively for each pair of genes, we perform a multi-testing corrections by multiplying this nominal P-value by the number of tests.

$$sig_H = -\log_{10}(Eval_H) = -\log_{10}(T \cdot Pval_H)$$

Where T is the number of pairwise comparison between gene profiles ($T=N_G(N_G-1)/2=15,593,320$), $N_G=5,585$ being the number of genes in the dyad profiles matrix.

Mutual information (*MI*)

The mutual information $MI=I(A,B)$ is defined according to Shannon's information theory.

$$I(A,B) = H(A) + H(B) - H(A,B)$$

In this formula, $H(A)$ and $H(B)$ are the entropies of gene profiles A and B, respectively.

$$H(A) = -P(A)\log(P(A)) - (1-P(A))\log(1-P(A))$$
$$H(B) = -P(B)\log(P(B)) - (1-P(B))\log(1-P(B))$$

whereas H(A,B) is the join entropy of A and B.

$$H(A,B) = -P(AB)\log(P(AB)) - P(A!B)\log(P(A!B))$$
$$- P(B!A)\log(P(B!A)) - P(!A!B)\log(P(!A!B))$$

The probability $P(A)$ of profile A is computed as the fraction of significant dyads in this profile ($P(A) = N_a / N_D$). The other probabilities are estimated in a similar way.

Dot product bits (*DPbits*)

In order to take into account the actual values of the binomial significance (sig_B) returned by *dyad-analysis*, we computed the dot product between each pair of significance profiles.

$$dp_{AB} = \sum_{j=1}^{D} (x_{Aj} \cdot x_{Bj})$$
$$DPbits_{AB} = \log_2(dp_{AB})$$

A dyad j will contribute to the dot product only if it is significant in both genes A ($sig_{Aj}>0$) and B ($sig_{Bj}>0$). The DPbits score thus requires not only for two genes to have the same selection of dyads, but also to have a coincidence between their highest-scoring binomial significance values.

Inference of a co-regulation network

The program *compare-classes* [42] was used to compare sets of significant dyads between each pair of genes, and to compute the four metrics described above, in order to derive a co-regulation network. Each node of this network represents a gene, and an edge is added for each pair of

genes sharing similar motifs. The possible number of edges of this network increases quadratically with the number of genes (N), with a maximum of $E_{max}=N(N-1)/2$ edges. Since the yeast genome contains $N_G=5,876$ protein-coding genes, the network can contain up to $T=17,260,750$ edges. In order to reduce the size of the inferred network, we selected only the genes having at least two significant dyads, and we applied various thresholds on the number of shared dyads, sig_H, *JS*, *MI* or *DPbits*, in order to analyze the impact of these parameters on the accuracy of the predicted network. The program *compare-classes* was also used to compare the clusters of direct neighbours of unknown genes to Gene Ontology functional classes [43].

Graph analysis

For the graph-analysis tasks in this work, we used a series of tools of the Network Analysis Tools (NeAT) suite [42].

Reference networks

The co-regulation reference network is constructed from the annotated regulons collected from the TRANSFAC database and complemented by annotations from Simonis [24]). The co-expression reference network is obtained from the STRING database (http://string.embl.de) [25]. The co-binding reference network uses a genome-wide location analysis characterized with the ChIP-chip method [26]. For this purpose we use the list of genes/orfs bound by each TF with no conservation criteria and a p-value < 0.001.The sizes of these networks are summarized in Table 1.

Computation of performance curves

The program *roc-stats* takes as input a set of scored predictions, each one annotated as either positive or negative. For each score value, the program counts as *true positive (TP)* the number of predictions above this score, as false positives (*FP*) the number of negative predictions above the score, as and as false negative (*FN*) the number predictions below the score. The *Sensitivity* is defined as the proportion of positives found among the predictions: $Sn=TP/(TP+FN)$. The *Positive Predictive value* (PPV) is the fraction of predictions found in the annotations: $PPV=TP/(TP+FP)$. All these statistics are exported in tab-delimited files, as well as in a graphical plot. Sn/PPV curves and Sn plots were generated with the statistical package R (http://www.r-project.org/).

Strains and media

The wild-type strain BY4741 and the isogenic mutant *upc2Δ* (*MATa*; *his3Δ1*; *leu2Δ0*; *met15Δ0*; *ura3Δ0*; *upc2Δ*) were obtained from Euroscarf (Frankfurt am Main). These strains were grown in minimal medium YNB with ammonium sulfate (Difco, USA) and the required supplements. A 25

mg/ml stock solution of Lovastatin (Sigma-Aldrich, USA) was made as previously described (Dimster-Denk et al., 1995). Lovastatin was added for 16 hours in the medium at a final concentration of 40µg/ml. The wild-type strain BY4741 and the isogenic mutant *rpn4Δ* (*MATa*; *his3Δ1*; *leu2Δ0*; *met15Δ0*; *ura3Δ0*; *rpn4Δ)* were obtained from Euroscarf (Frankfurt am Main). These strains were grown on rich medium. The reference strain 23344c (*ura3Δ*) and the isogenic mutant 50054a (*ure2Δ*; *gzf3Δ*; *dal80Δ*; *ura3Δ*) were grown on minimal glutamine medium (Godard et al., 2007). The MBP1-3XHA strain (YSC1070-663055) and his parental strain Y800 (Open Biosystem, USA) were grown on rich medium.

RNA preparation and qRT-PCRs

Total RNA was purified as previously described (Kohrer, K., and H. Domdey, 1991). Quantitative reverse transcription-PCR were used to measure the mRNA levels of the following genes: *ACT1*, *ERG2*, *YML082W*, *YML083C*, *YDR531W*, *YJL026C*, *YDL237W*, *VAC17*, *GUD1*, *CHA1*, *RPN2*, *YNL155W* and *YOR052C*. For this we used the RT-RTCK05 and RT-SN10-05 kits (Eurogentec, Belgium) and the primers listed in Supplementary Table S5.

ChIP

ChIP were done essentially as described (Kuras L, Struhl K, 1999 ; Kuras et al., 2003). The HA-tagged protein was immunoprecipitated with 12CA5 antibody bound to IgG magnetic beads (Dynabead; Dynal Biotech ASA, Oslo, Norway). Immunoprecipitated DNA was analyzed by quantitative realtime PCR on an ABI Prism 7000 machine (Applied Biosystems, Foster City, CA). Relative quantification using a standard curve method was performed and the occupancy level for a specific fragment was defined as the ratio of immunoprecipitated over total DNA. For this we used the Platinum SYBR Green PCR Super Mix-UDG with ROX (Invitrogen) with the primers mentioned in the Supplementary Table S5.

Availability

Phylogenetic footprint detection was performed using a series of programs of the Regulatory Sequence Analysis Tools (RSAT) [44]: *get-orthologs*, *retrieve-seq*, *purge-sequence*, *dyad-analysis*, *pattern-assembly*, *dna-pattern*, *feature-map*. Graph analysis relied on the programs *compare-graphs*, *graph-get-clusters*, *graph-neighbours*, *random-graph*, *roc-stats*, which are part of the Network Analysis Tools (NeAT) [42]. All those programs are available for academic users via the Web interface (http://rsat.scmbb.ulb.ac.be/rsat/). The programs can also be accessed as Web services, in order to integrate them in workflows.

Supplementary methods, scripts and results are available on the RSAT web site http://rsat.scmbb.ulb.ac.be/rsat/data/published_data/coregulation_networks/.

List of abbreviations

DP: dot product ; sig_B: binomial significance score ; sig_H: hypergeometric significance score ; *MI*: mutual information; *Sn*: sensitivity; *PPV*: positive predictive value; *TF*: transcription factor.

Acknowledgements

RJ and SB are supported by a doctoral grant from the Fonds pour la Recherche dans l'Industrie et l'Agriculture (F.R.I.A.) allowed by the Belgian Fond National de la Recherche Scientifique (F.N.R.S.). RJ is also supported by a postdoctoral grant from the Wiener-Anspach Foundation. FA is supported by the Actions de Recherche Concertées de la Communauté Française de Belgique (A.R.C. grant number 04/09-307). This project was partly supported by the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265) and by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet). We thank Madan M. Babu for critically reading this paper and for helpful suggestions.

Author contributions

The method for footprint detection has been previously developed by RJ and JvH, and was applied to all Fungal promoters by JvH and SB. The method and metrics for inferring networks from footprints were developed by JvH. The program *compare-graphs* used to compare inferred and reference networks was developed by GV and SB. SB, RJ and JvH equally contributed to the analysis of the predicted motifs and co-regulation networks, as well as to the writing of the manuscript. FA and BA participated to the biological interpretation of the inferred co-regulation network, and in the experimental tests.

References

1. Hertz, G.Z., Hartzell, G.W., 3rd & Stormo, G.D. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput Appl Biosci* **6**, 81-92 (1990).
2. Lawrence, C.E. et al. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**, 208-14 (1993).
3. Bailey, T.L. & Elkan, C. The value of prior knowledge in discovering motifs with MEME. *Proc Int Conf Intell Syst Mol Biol* **3**, 21-9 (1995).
4. van Helden, J., Andre, B. & Collado-Vides, J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol* **281**, 827-42 (1998).
5. Brazma, A., Jonassen, I., Eidhammer, I. & Gilbert, D. Approaches to the automatic discovery of patterns in biosequences. *J Comput Biol* **5**, 279-305 (1998).
6. Brazma, A., Jonassen, I., Vilo, J. & Ukkonen, E. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res* **8**, 1202-15 (1998).
7. van Helden, J., Rios, A.F. & Collado-Vides, J. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res* **28**, 1808-18 (2000).
8. Liu, X., Brutlag, D.L. & Liu, J.S. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac Symp Biocomput*, 127-38 (2001).
9. Thijs, G. et al. A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17**, 1113-22. (2001).
10. Wasserman, W.W. & Fickett, J.W. Identification of regulatory regions which confer muscle-specific gene expression. *J Mol Biol* **278**, 167-81 (1998).
11. Terai, G., Takagi, T. & Nakai, K. Prediction of co-regulated genes in Bacillus subtilis on the basis of upstream elements conserved across three closely related species. *Genome Biol* **2**, RESEARCH0048 (2001).
12. Blanchette, M. & Tompa, M. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res* **12**, 739-48 (2002).
13. Blanchette, M., Schwikowski, B. & Tompa, M. Algorithms for phylogenetic footprinting. *J Comput Biol* **9**, 211-23 (2002).
14. Alkema, W.B., Lenhard, B. & Wasserman, W.W. Regulog analysis: detection of conserved regulatory networks across bacteria: application to Staphylococcus aureus. *Genome Res* **14**, 1362-73 (2004).
15. Janky, R. & van Helden, J. Discovery of conserved motifs in promoters of orthologous genes in prokaryotes. *Methods Mol Biol* **395**, 293-308 (2007).
16. Yellaboina, S., Ranjan, S., Chakhaiyar, P., Hasnain, S.E. & Ranjan, A. Prediction of DtxR regulon: identification of binding sites and operons controlled by Diphtheria toxin repressor in Corynebacterium diphtheriae. *BMC Microbiol* **4**, 38 (2004).
17. Yellaboina, S., Seshadri, J., Kumar, M.S. & Ranjan, A. PredictRegulon: a web server for the prediction of the regulatory protein binding sites and operons in prokaryote genomes. *Nucleic Acids Res* **32**, W318-20 (2004).
18. Wels, M., Francke, C., Kerkhoven, R., Kleerebezem, M. & Siezen, R.J. Predicting cis-acting elements of Lactobacillus plantarum by comparative genomics with different taxonomic subgroups. *Nucleic Acids Res* **34**, 1947-58 (2006).
19. Janky, R. & van Helden, J. Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution. *BMC Bioinformatics* **9**, 37 (2008).
20. McCue, L. et al. Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res* **29**, 774-82. (2001).

21. van Nimwegen, E., Zavolan, M., Rajewsky, N. & Siggia, E.D. Probabilistic clustering of sequences: inferring new bacterial regulons by comparative genomics. *Proc Natl Acad Sci U S A* **99**, 7323-8 (2002).
22. Qin, Z.S. et al. Identification of co-regulated genes through Bayesian clustering of predicted regulatory binding sites. *Nat Biotechnol* **21**, 435-9 (2003).
23. Wingender, E., Dietze, P., Karas, H. & Knuppel, R. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Res* **24**, 238-41 (1996).
24. Simonis, N., Wodak, S.J., Cohen, G.N. & van Helden, J. Combining pattern discovery and discriminant analysis to predict gene co-regulation. *Bioinformatics* **20**, 2370-9 (2004).
25. Snel, B., Lehmann, G., Bork, P. & Huynen, M.A. STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Res* **28**, 3442-4 (2000).
26. Harbison, C.T. et al. Transcriptional regulatory code of a eukaryotic genome. *Nature* **431**, 99-104 (2004).
27. Babu, M.M., Luscombe, N.M., Aravind, L., Gerstein, M. & Teichmann, S.A. Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol* **14**, 283-91 (2004).
28. Ashburner, M. et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **25**, 25-9 (2000).
29. Vik, A. & Rine, J. Upc2p and Ecm22p, dual regulators of sterol biosynthesis in Saccharomyces cerevisiae. *Mol Cell Biol* **21**, 6395-405 (2001).
30. Davies, B.S., Wang, H.S. & Rine, J. Dual activators of the sterol biosynthetic pathway of Saccharomyces cerevisiae: similar activation/regulatory domains but different response mechanisms. *Mol Cell Biol* **25**, 7375-85 (2005).
31. Xie, Y. & Varshavsky, A. RPN4 is a ligand, substrate, and transcriptional regulator of the 26S proteasome: a negative feedback circuit. *Proc Natl Acad Sci U S A* **98**, 3056-61 (2001).
32. Stanhill, A. et al. An arsenite-inducible 19S regulatory particle-associated protein adapts proteasomes to proteotoxicity. *Mol Cell* **23**, 875-85 (2006).
33. Haugen, A.C. et al. Integrating phenotypic and expression profiles to map arsenic-response networks. *Genome Biol* **5**, R95 (2004).
34. Cooper, T.G. Transmitting the signal of excess nitrogen in Saccharomyces cerevisiae from the Tor proteins to the GATA factors: connecting the dots. *FEMS Microbiol Rev* **26**, 223-38 (2002).
35. Bar-Joseph, Z. et al. Computational discovery of gene modules and regulatory networks. *Nat Biotechnol* **21**, 1337-42 (2003).
36. Godard, P. et al. Effect of 21 different nitrogen sources on global gene expression in the yeast Saccharomyces cerevisiae. *Mol Cell Biol* **27**, 3065-86 (2007).
37. Scherens, B., Feller, A., Vierendeels, F., Messenguy, F. & Dubois, E. Identification of direct and indirect targets of the Gln3 and Gat1 activators by transcriptional profiling in response to nitrogen availability in the short and long term. *FEMS Yeast Res* **6**, 777-91 (2006).
38. Bahler, J. Cell-cycle control of gene expression in budding and fission yeast. *Annu Rev Genet* **39**, 69-94 (2005).
39. Spellman, P.T. et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell* **9**, 3273-97 (1998).
40. Lozada-Chavez, I., Angarica, V.E., Collado-Vides, J. & Contreras-Moreira, B. The role of DNA-binding specificity in the evolution of bacterial regulatory networks. *J Mol Biol* **379**, 627-43 (2008).
41. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. & Wheeler, D.L. GenBank. *Nucleic Acids Res* **36**, D25-30 (2008).

42. Brohee, S. et al. NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Res* (2008).
43. Matys, V. et al. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* **34**, D108-10 (2006).
44. Thomas-Chollier, M. et al. RSAT: regulatory sequence analysis tools. *Nucleic Acids Res* (2008).
45. Becker, B., Feller, A., el Alami, M., Dubois, E. & Pierard, A. A nonameric core sequence is required upstream of the LYS genes of Saccharomyces cerevisiae for Lys14p-mediated activation and apparent repression by lysine. *Mol Microbiol* **29**, 151-63 (1998).

LEGENDS TO FIGURES

Figure 1. Description of the method to infer co-regulation networks by phylogenetic footprint discovery. (a) Phylogenetic footprints discovered in promoters of LYS1 (left) and LYS12 (right) orthologs in Saccharomycetes. Among the 43,680 possible dyads, only 30 reach a positive binomial significance score (sig_B>=0) for *LYS1*, and 24 for *LYS12*. In both cases, the canonical motif WWWTCCRnYGGAWWW [45], bound by the lysine activator Lys14p, is matched by the top scoring dyads (3th dyad for *LYS1* and 1st for *LYS12*). Note that the two top-scoring dyads for *LYS1* match the binding motif of Gcn4p, the general amino acid controlling factor. (b) Sequence logos built from the over-represented dyads matching the Lys14p motif. (c) Dyad significance profiles. A small fragment of the dyad significance matrix (the full matrix comprises 3,585 rows (genes) and 33,276 columns (dyads). Cells indicate the binomial significance sig_B of each dyad in the phylogenetic footprints of each gene. Negative significance values are denoted by a dot, and are replaced by a zero value for further computation. Boxes highlight groups of dyads that match the motif annotated for a given regulator (Upc2p, Gcn4p, Leu3p and Lys14p). (d) Detail of computation of the dot product bits (DPbits) score, used to measure pairwise similarities between footprints found in each pair of genes. (e) A co-regulation network is built by tracing an edge between each pair of genes having a strictly positive DPbits score. Edge thickness and color gradient (yellow to red) is proportional to the round of DPbits.

Figure 2. Comparison between inferred and annotated co-regulation networks. (a) High-scoring layer (DPbits >= 5) of the co-regulation network inferred between all yeast genes. Edge color and thickness reflect the prediction score. (b) Subset of the inferred network restricted to the 612 genes of the reference data set (80 regulons, DPbits >= 1). Arcs are labeled according to the correspondence with the annotated regulons (false positives in red and true positives in green). (c) Impact of the score threshold on the size of the network inferred between genes of the reference set (annotated regulons). Green: true positive edges. Red: false positive edges. Blue : all edges. Dotted curves: random control obtained by permuting the edges of the inferred network. (d) Comparison between inferred co-regulation and annotated regulons. The *Sn* (blue) indicates the fraction of annotated co-regulations (edges) detected by the inference method, whereas the *PPV* (black) indicates the fraction of predictions that correspond to actual annotations. Dotted curves: random control. (e) Comparison between inferred co-regulation and STRING co-expression network. (f) Comparison between STRING co-expression and annotated regulons. (g) Comparison between inferred co-regulation and ChIP-chip network. (h) Comparison between STRING co-expression and ChIP-chip network.

Figure 3. Annotated co-regulation network (dataset from Simonis et al., 2005). This Figure shows the result of the conversion of the regulons from a factor-gene (a) to a gene-gene (c) network. The degree distributions of the factor-gene (b) and gene-gene (d) networks.

Figure 4. Impact of regulon size on sensitivity. (a) Gene-gene cliques generated by specific (*GAL4*, *MET4*) and generic (*GCN4*, *ZAP1*) transcription factors. Correctly predicted co-regulations (True Positives) are highlighted in green, with edge thickness proportional to the DPbits score. Orange edges indicate false negatives (FN), i.e. existing co-regulations that are missed in the predicted co-regulation network. Edge-wise sensitivity (E_{sn}) is the fraction of correctly predicted edges. Node-wise sensitivity (N_{sn}) is computed as the fraction of target genes that are inter-linked by at least one prediction. (b) Edge-wise Sensitivity (proportion of correctly predicted co-regulations) as a function of the regulon size (abscissa). (c) Regulon-wise sensitivity (ordinate) as a function of the regulon size (abscissa). All numerical values are provided in Supplementary Table S3.

Figure 5. Experimental validation of selected inferred regulations by RT-PCR. Each panel shows a group of neighbours in the inferred network, together with the experimental measurement of the transcriptional responses. (a) Effect of Lovastatin and *UPC2* deletion on the expression level of four potentially Upc2 regulated genes, taking *ERG2* as control. (b) Effect of the *RPN4* deletion on two potentially Rpn4 regulated genes taking *RPN2* as control. (c) Effect of *ure2gzf3dal80* deletion on four potentially NCR regulated genes. (d). ChIP PCR analysis results of four potentially Mbp1 regulated genes. Color code for the network clusters: genes known to be submitted to a common regulation are highlighted in yellow. Cyan-shaded octogonal boxes indicate genes submitted to experimental validation. Edge thickness and color reflect the DPbits score of the predicted co-regulations.

Supplementary Figure S6. Comparisons between alternative metrics for computing the similarity between dyad significance profiles. Note the almost perfect correspondence between hypergeometric significance and mutual information, and the strong differences between all the other pairs of metrics.

Supplementary Figure S7. Quantitative evaluation of the impact of different scores on the inferred co-regulation network. Each column shows the results when using one score: DPbits (1st column: a,e,i,m), Jaccard Similarity (2nd column: b,f,j,n), Mutual Information (3rd column: c,g,k,o), Hypergeometric Significance (4th column: d,h,l,p). The first row shows the number of inferred co-regulation interactions as a function of the score (a-d). Color code : Blue : total number of edge. Green : true positive. Red : false positives. Dotted lines : random control. The second row shows the results of the evaluation of the inferred versus annotated co-regulation networks (e-h). The third row shows the results of the comparison of the inferred co-regulation versus co-expression (STRING) (i-l). The last row shows the results of the comparison of the inferred co-regulation

network versus co-binding network (Harbison et al, 2004). Color code : Black : PPV. Blue : Sensitivity. Dotted lines : random control.

Supplementary Figure S8. Complete co-regulation network inferred on the basis of phylogenetic footprints. Edge width and color gradient reflect the DPbits score.

Supplementary Figure S9. Relationship between the number of edges in the annotated factor-gene and in the gene-gene networks.

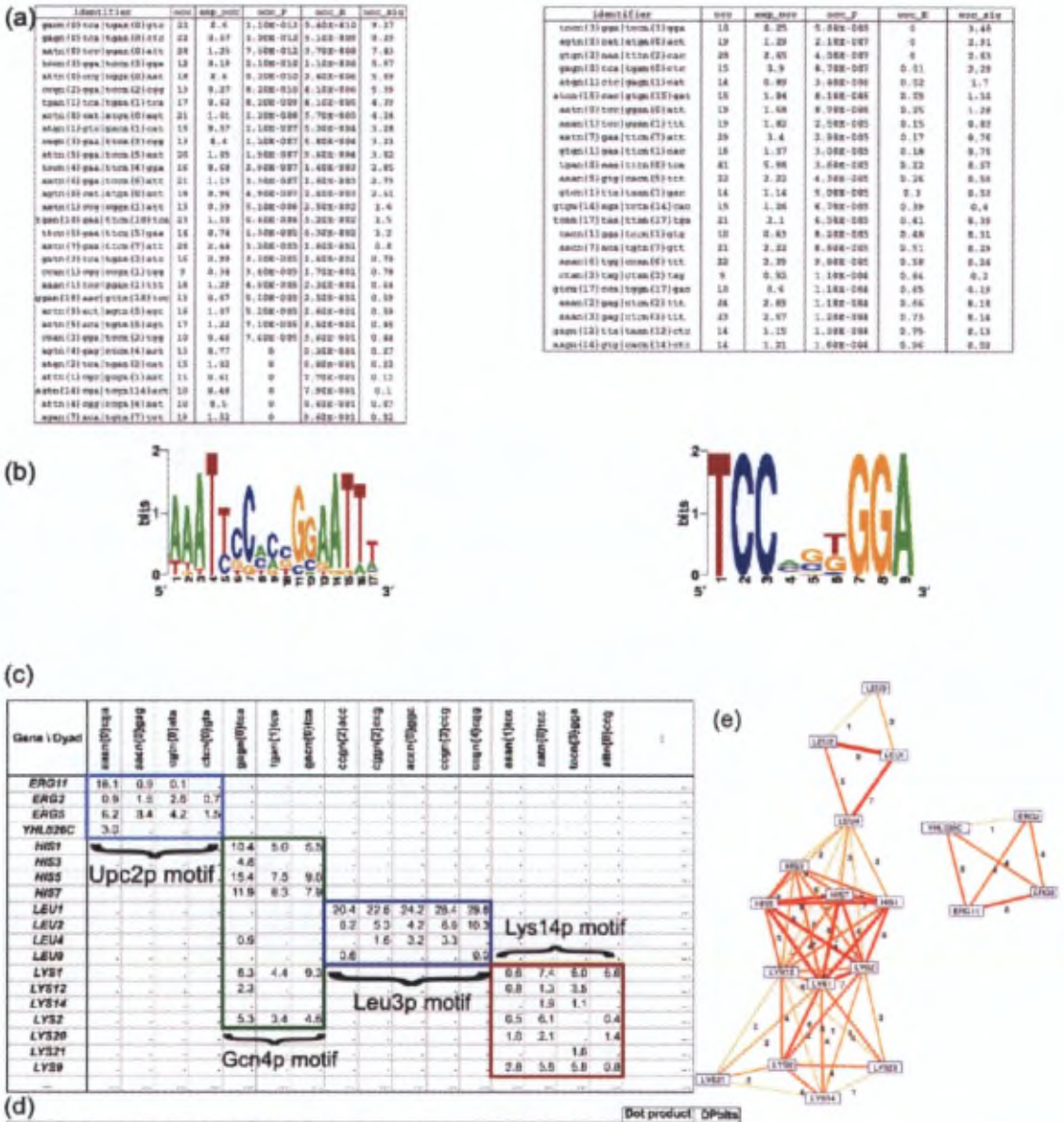


FIG. 1 – Figure 1

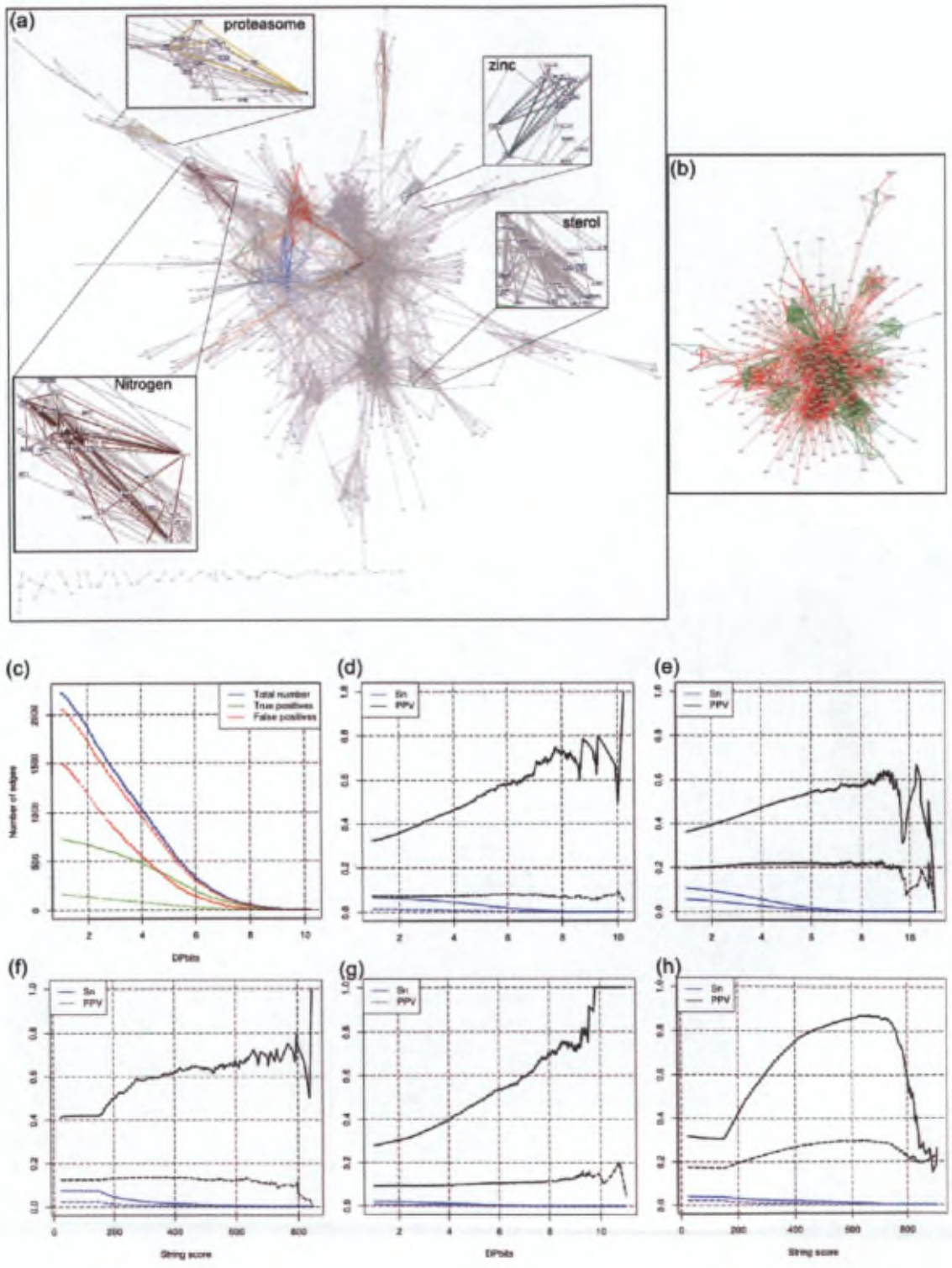


FIG. 2 – Figure 2

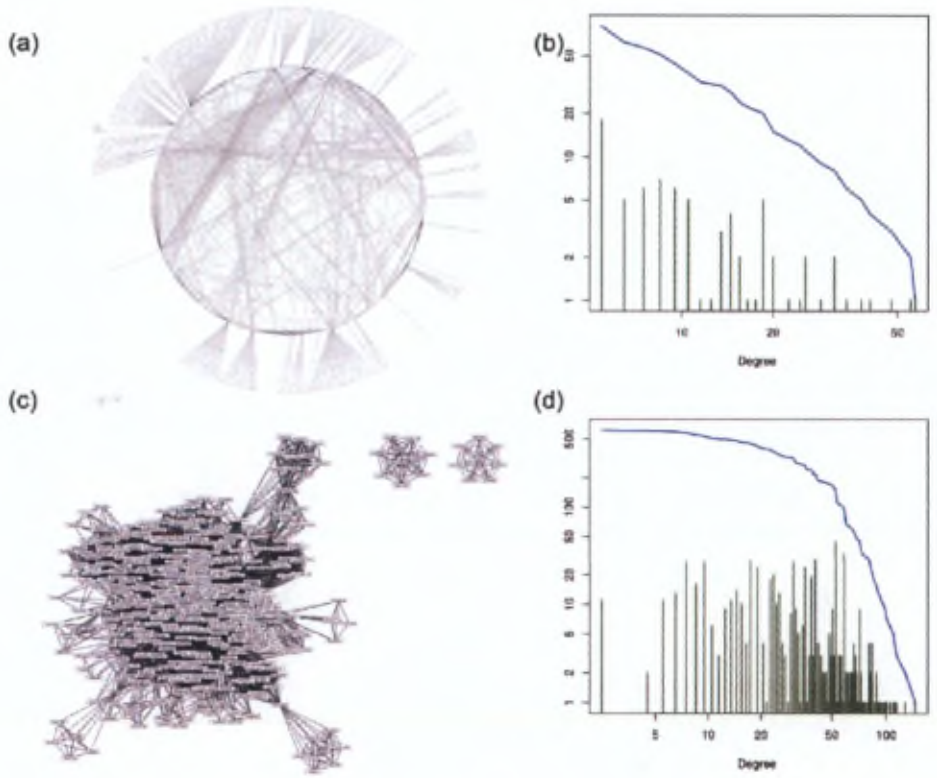


FIG. 3 – Figure 3

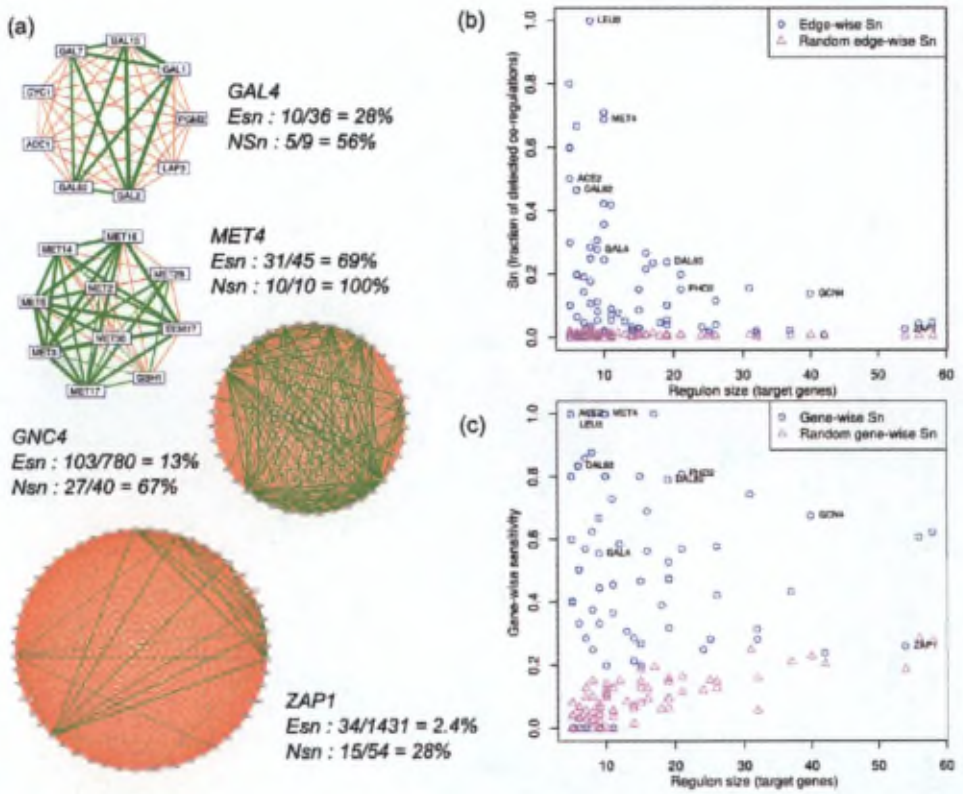


FIG. 4 – Figure 4

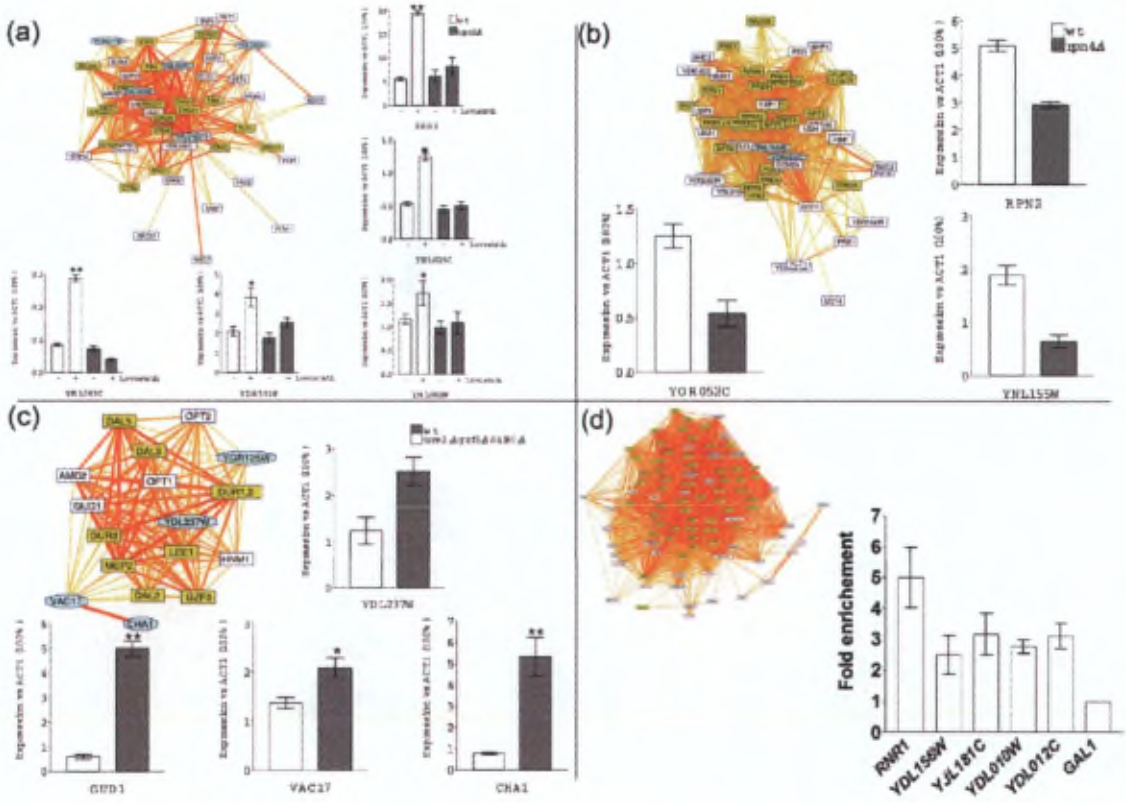


FIG. 5 – Figure 5

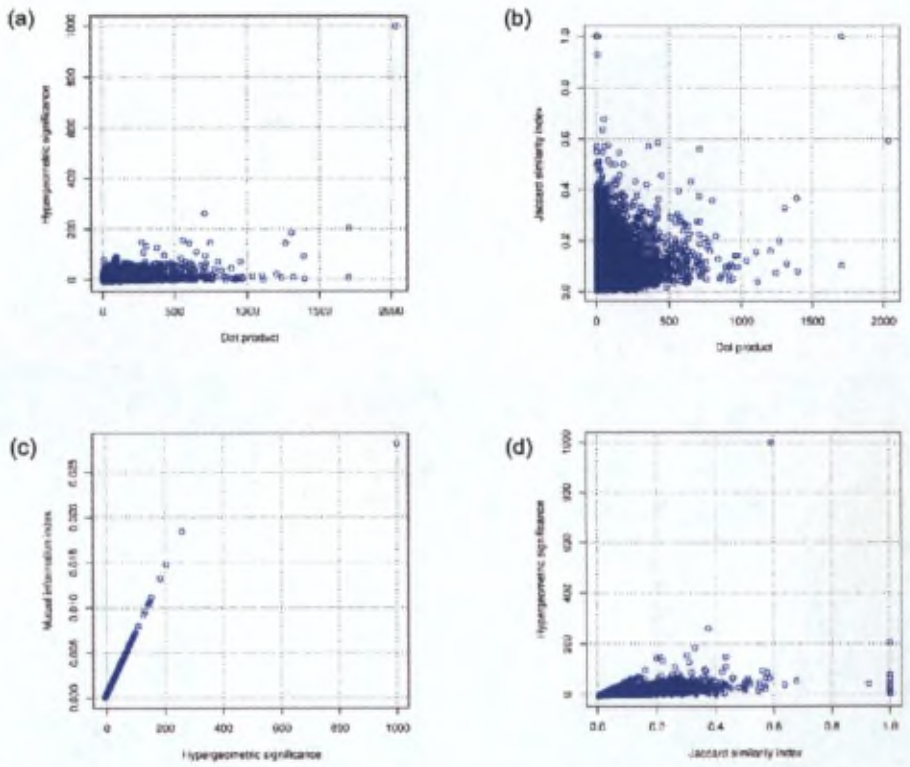


FIG. 6 – Figure S6

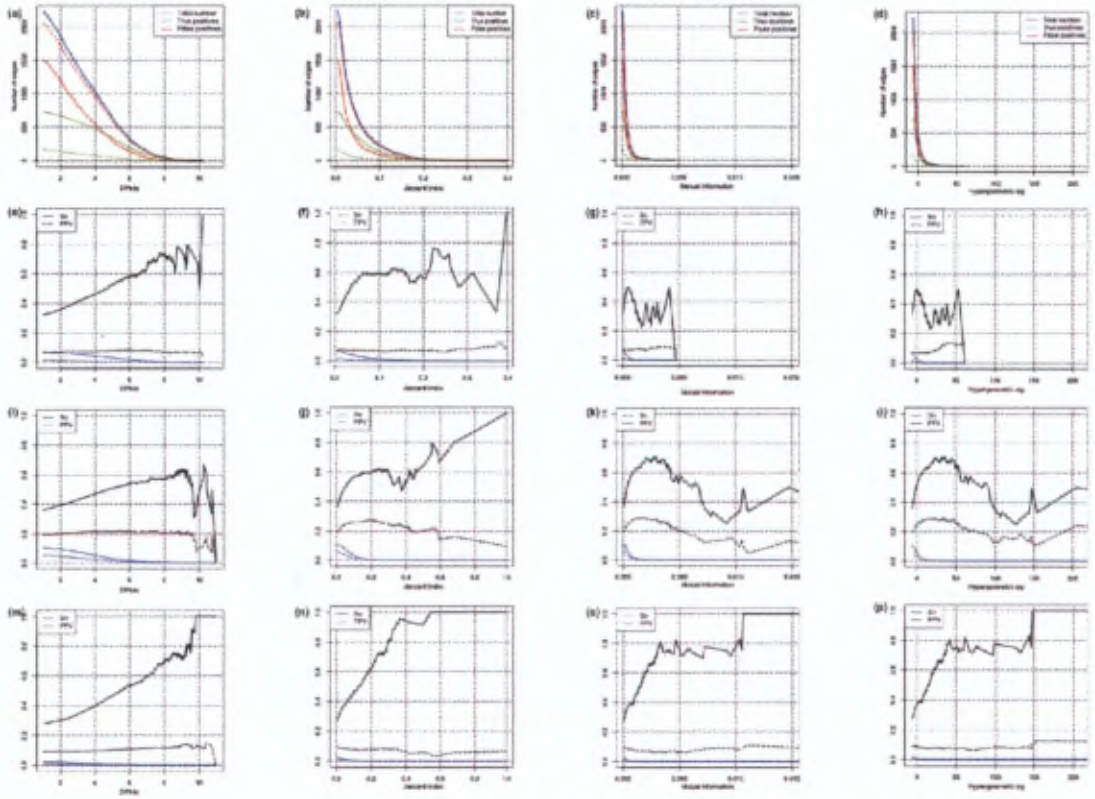


FIG. 7 – Figure S7

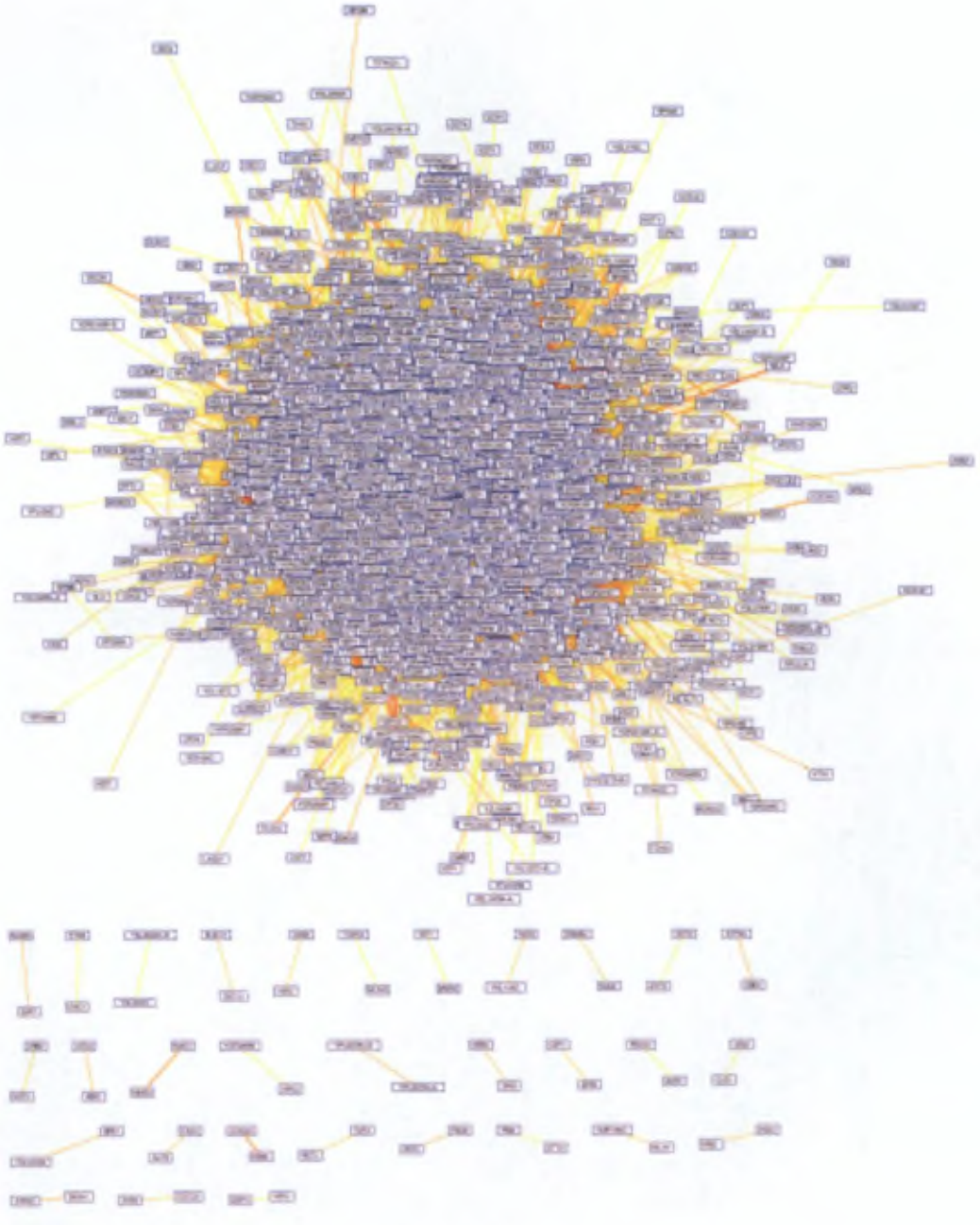


FIG. 8 – Figure S8

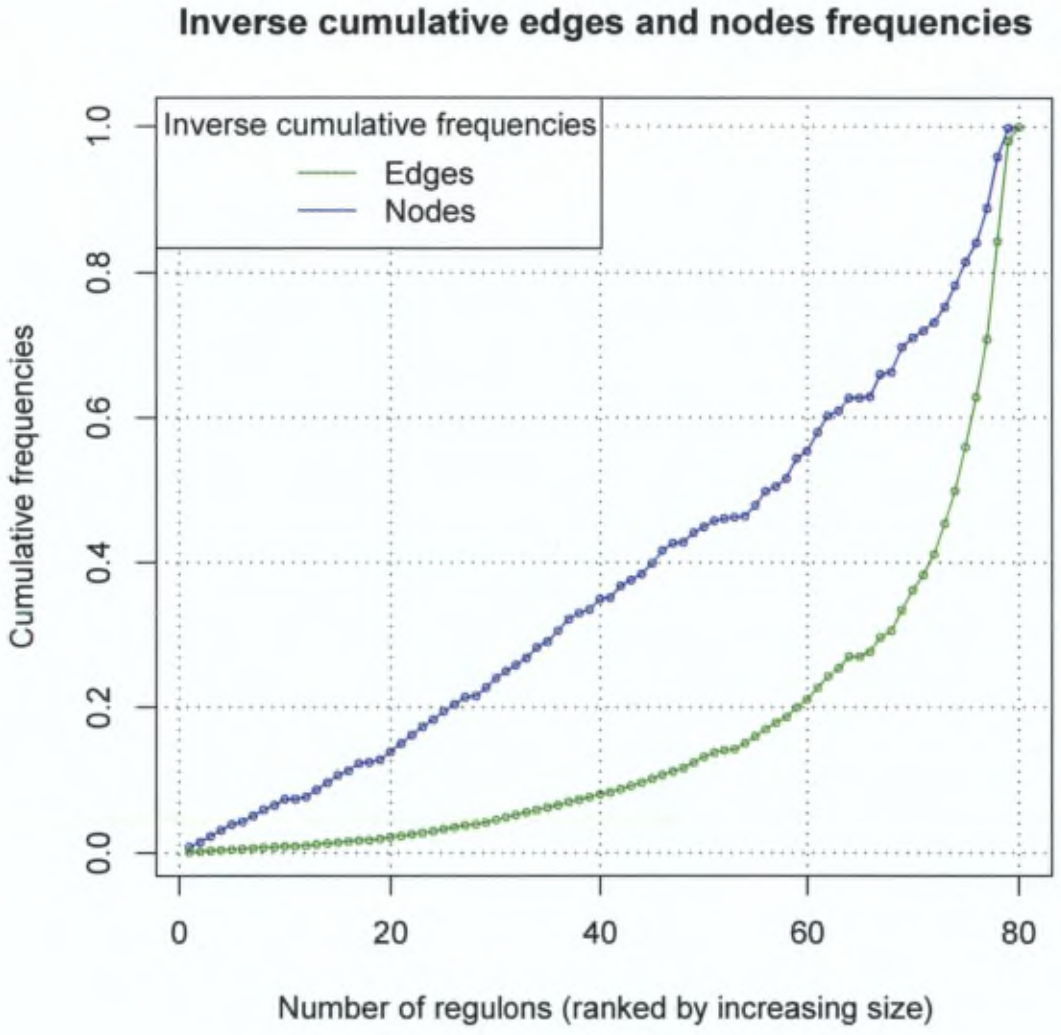


FIG. 9 – Figure S9

TABLES

Table 1. Size of the annotated and inferred co-regulation networks.

A. Discovered footprints	
Coding genes (orfs)	5,876
Genes with at least one significant dyad (sig_B >= 0)	5,585
Genes with at least one high-confidence prediction (sig_B >= 3)	1638
Distinct dyads (footprints)	22,565

B. Factor-target gene networks			
Dataset	**Type**	**Nodes**	**Edges**
Annotated regulons	TF-gene	80 TF 612 genes	1,172
ChIP-chip co-binding (Harbison *et al*, 2004)	TF-gene	155 TF 2,655 genes	5,898
Intersection between co-expression (STRING) and co-binding (Harbison et al., 2004) networks	Gene-gene	833	7,465
Intersection between networks of co-expression (STRING) and co-regulation (annotated regulons).	Gene-gene	297	794
Intersection between networks of co-regulation (annotated regulons) and co-binding (Harbison et al., 2004)	Gene-gene	334	1,095

C. Gene-gene networks								
Subset of the network (in row) restricted to genes in a second network (in column)	Inferred co-regulation network		Annotated co-regulation network		STRING coexpression		ChIP-chip co-binding	
	nodes	edges	nodes	edges	nodes	edges	nodes	edges
	3,171	56,919	612	10,599	2,895	92,770	2,397	178,202
Inferred co-regulation network	3,171	56,919	386	2,226	1,578	27,330	1,440	14,379
Annotated co-regulation network			612	10,599	410	5,320	403	4,638
STRING coexpression					2,895	92,770	1,208	23,456
ChIP-chip co-binding							2,397	178,202

Table 2. Top-ranking predictions of co-regulation in *Saccharomyces cerevisiae*, ordered by DPbit score. Gene pairs belonging to the same regulon are highlighted in bold; italics and yellow background indicate gene pairs for which the annotated regulons contains no evidence of co-regulation, but which are involved in similar functions. Note that in our quantitative evaluation, these pairs are labeled as "false positive" (FP), despite their very likelihood to be co-regulated.

Gene 1	Gene 2	DPbits	Jaccard	hypersig	Mutual Info (MI)	Annotated regulon	Co-expression	Similar function
LEU2	**LEU1**	10.2	16.2%	22.7	0.00203	LEU3		
RNR1	*RAD27*	10.0	12.2%	12.3	0.00129			DNA replication and repair
RNR1	**CDC21**	9.9	9.8%	4.6	0.00077	MBP1		
SWI4	**RNR1**	9.6	4.9%	-1.5	0.00035	SWI6		
RNR1	**CDC6**	9.3	3.9%	-2.3	0.00029	MBP1		
LEU1	*BAT1*	9.3	11.5%	13.1	0.00135		1	Branched amino acid metabolism
RNR1	**RAD53**	9.2	13.7%	9.9	0.00114	SWI6		
YKR075C	*HXT3*	9.2	4.9%	6.0	0.00085			Regulated by glucose
ZRT2	**ZPS1**	9.2	27.1%	25.5	0.00221	ZAP1		
YKR075C	**HXT2**	9.2	11.0%	29.4	0.00246	MIG1		
GAL10	**GAL1**	9.1	39.7%	53.2	0.00413	GAL4	1	
SST2	**FUS1**	9.0	14.7%	5.7	0.00085	MOT3	1	
RNR1	**CLB5**	8.9	1.7%	-4.0	0.00017	SWI6		
PCK1	**FBP1**	8.8	5.8%	8.2	0.00100	CAT8	1	
RAD27	*CDC21*	8.7	7.4%	1.8	0.00058		1	DNA replication and repair
SWI4	*RAD27*	8.7	4.8%	-1.6	0.00035			DNA replication and cell cycle regulation
FTR1	*FIT2*	8.7	4.2%	2.8	0.00063			Iron transport and homeostasis
HIS7	**ARG3**	8.6	5.9%	0.3	0.00047	GNC4		
RNR1	*ABF1*	8.6	6.4%	2.8	0.00064			DNA replication and repair
TPO1	LEU1	8.6	3.2%	-1.4	0.00035			
LEU1	**ILV5**	8.6	5.7%	3.2	0.00066	LEU3		
MEP2	**DUR1,2**	8.6	23.8%	7.2	0.00095	GLN3		
ZRT3	**ZRT2**	8.6	10.9%	5.5	0.00083	ZAP1		
MEP2	**DAL1**	8.6	4.5%	-1.3	0.00037	GLN3		
DAL4	**DAL1**	8.5	23.6%	38.3	0.00309	GLN3		
GAL2	**GAL1**	8.5	4.1%	-1.2	0.00036	GAL4		
SWI4	**CDC21**	8.5	25.0%	10.0	0.00115	SWI6		
RAD27	*CDC6*	8.5	6.5%	1.7	0.00056			DNA replication and repair
LEU1	**GDH1**	8.4	6.2%	9.8	0.00111	LEU3	1	
DUR1,2	**DAL1**	8.4	9.4%	6.3	0.00089	GLN3		
LEU1	**ILV3**	8.4	7.6%	5.6	0.00083	LEU3	1	
MEP2	**DAL4**	8.3	4.8%	-1.2	0.00037	GLN3		
SWI4	**CDC6**	8.3	8.0%	1.3	0.00054	SWI6		
RNR1	*CLB6*	8.3	6.3%	1.5	0.00055			DNA replication and repair
HXT3	**HXT2**	8.3	5.8%	5.2	0.00080	CYC8		
PCK1	**ICL1**	8.2	6.7%	9.2	0.00107	CAT8		
RAD53	*RAD27*	8.2	9.1%	4.1	0.00073			DNA repair
CDC6	**CDC21**	8.2	7.0%	-0.4	0.00043	SWI6		
HIS1	*ARG3*	8.2	7.0%	1.5	0.00056			Amino acid metabolism
PCK1	**MDH2**	8.2	6.7%	10.0	0.00113	CAT8		
RNR3	**RNR1**	8.2	6.2%	10.4	0.00116	TUP1		
LEU1	**ILV2**	8.2	6.1%	2.9	0.00064	LEU3	1	
RAD27	*ABF1*	8.2	37.8%	61.5	0.00470			Dna replication and repair
GAL2	**GAL10**	8.1	4.9%	-0.5	0.00042	GAL4		
HXT4	**HXT2**	8.1	6.9%	10.2	0.00114	MIG1		
ZRT2	**ZRT1**	8.1	12.2%	6.1	0.00087	ZAP1		
DUR1,2	**DAL4**	8.1	10.0%	6.5	0.00090	GLN3		
RAD53	**CDC21**	8.1	26.3%	7.6	0.00099	MBP1		
ZRT2	**VEL1**	8.0	9.1%	3.1	0.00066	ZAP1		
HXT6	*HXT2*	8.0	6.7%	6.5	0.00089			Hexose transporters
SWI4	**RAD53**	8.0	10.3%	0.2	0.00047	SWI6		
LEU1	ACO1	8.0	1.8%	-4.7	0.00013			
LEU4	**LEU1**	8.0	5.1%	2.5	0.00061	LEU3	1	
LEU1	**BAP2**	8.0	2.4%	-2.9	0.00025	LEU3	1	
PRB1	**MEP2**	7.9	0.9%	-4.7	0.00014	GLN3		
YKR075C	**HXT4**	7.9	3.8%	2.6	0.00061	MIG1		
CLB5	**CDC6**	7.9	2.5%	-2.2	0.00030	SWI6		
PDE2	DED1	7.9	21.4%	41.3	0.00330			
TRP3	**ARG3**	7.9	6.9%	1.4	0.00055	GCN4		
CLB5	**CDC21**	7.9	2.2%	-2.1	0.00031	SWI6		

SUPPLEMENTARY TABLES

Supplementary Table S3. Coverage of co-regulation interactions per regulon. Color code. A color gradient has been applied on the edge sensitivity values light yellow : between 0.3 and 0.5; orange : > 0.5 and on the node sensitivity values : light yellow : between 0.4 and 0.7; orange : > 0.7.

regulons	Number of nodes	Max number of edges	Number of edges	Edge Sn	Random Edge Sn	Node Sn	Random Node Sn
ABF1	37	666	14	2,1%	1,4%	43,2%	36,2%
ACE2	5	10	5	50,0%	8,0%	100,0%	28,0%
ADR1	11	55	5	9,1%	1,1%	36,4%	10,9%
BAS1	17	136	32	23,5%	2,5%	100,0%	35,3%
CAT8	10	45	32	71,1%	2,7%	100,0%	20,0%
CBF1	16	120	26	21,7%	2,0%	56,3%	25,0%
CYC8	6	15	7	46,7%	0,0%	83,3%	0,0%
DAL80	19	171	41	24,0%	2,7%	78,9%	36,8%
DAL81	10	45	11	24,4%	1,8%	80,0%	16,0%
DAL82	6	15	7	46,7%	6,7%	83,3%	30,0%
GAL4	9	36	10	27,8%	4,4%	55,6%	28,9%
GAT1	9	36	11	30,6%	1,1%	66,7%	8,9%
GCN4	40	780	107	13,7%	1,9%	67,5%	50,0%
GCR1	18	153	7	4,6%	1,2%	38,9%	15,6%
GCR2	11	55	3	5,5%	1,5%	45,5%	12,7%
GLN3	31	465	73	15,7%	1,9%	74,2%	43,2%
GRR1	5	10	6	60,0%	0,0%	80,0%	0,0%
HAA1	10	45	1	2,2%	0,0%	20,0%	0,0%
HAC1	7	21	1	4,8%	0,0%	28,6%	0,0%
HAP1	7	21	0	0,0%	2,9%	0,0%	17,1%
HAP2	14	91	3	3,3%	1,5%	21,4%	17,1%
HAP3	15	105	3	2,9%	1,5%	20,0%	18,7%
HAP4	14	91	2	2,2%	1,1%	21,4%	14,3%
HAP5	9	36	0	0,0%	0,6%	0,0%	4,4%
HMRA1	5	10	0	0,0%	2,0%	0,0%	8,0%
HOG1	8	28	3	10,7%	0,7%	37,5%	5,0%
HSF1	21	210	42	20,0%	1,5%	57,1%	24,8%
IME1	7	21	3	14,3%	1,0%	57,1%	5,7%
INO2	19	171	17	9,9%	1,2%	47,4%	17,9%
INO4	19	171	17	9,9%	1,2%	47,4%	17,9%
LEU3	8	28	28	100,0%	2,9%	100,0%	20,0%
MAC1	9	36	3	8,3%	1,1%	44,4%	8,9%
MBP1	6	15	10	66,7%	1,3%	83,3%	6,7%
MCM1	14	91	2	2,2%	1,1%	28,6%	12,9%
MET31	5	10	8	80,0%	6,0%	100,0%	24,0%
MET4	10	45	31	68,9%	4,4%	100,0%	36,0%
MIG1	26	325	37	11,4%	1,4%	57,7%	30,8%
MIG2	15	105	9	8,6%	2,5%	46,7%	29,3%

MOT3	15	105	16	15,2%	1,9%	80,0%	25,3%
MSN2	56	1540	67	4,4%	1,4%	60,7%	45,7%
MSN4	58	1653	78	4,7%	1,4%	62,1%	47,2%
NDT80	11	55	4	7,3%	2,9%	45,5%	25,5%
NRG1	5	10	1	10,0%	0,0%	40,0%	0,0%
OAF1	24	276	9	3,3%	1,6%	25,0%	29,2%
PDR1	16	120	32	26,7%	1,5%	68,8%	20,0%
PDR3	10	45	16	35,6%	1,8%	80,0%	16,0%
PHO2	21	210	32	15,2%	1,8%	81,0%	32,4%
PHO4	8	28	1	3,6%	1,4%	25,0%	10,0%
PIP2	19	171	9	5,3%	1,3%	31,6%	21,1%
RAP1	32	496	9	1,8%	1,2%	31,3%	27,5%
RCS1	9	36	4	11,1%	3,3%	44,4%	26,7%
REB1	19	171	7	4,1%	1,1%	52,6%	16,8%
RFA2	5	10	1	10,0%	0,0%	40,0%	0,0%
RFX1	5	10	1	10,0%	0,0%	40,0%	0,0%
RGT1	5	10	6	60,0%	0,0%	80,0%	0,0%
RIM101	7	21	0	0,0%	0,0%	0,0%	0,0%
RLM1	25	300	6	2,0%	1,6%	28,0%	28,0%
ROX1	15	105	3	2,9%	0,8%	26,7%	10,7%
RPN4	11	55	23	41,8%	1,5%	72,7%	14,5%
RTG1	12	66	5	7,6%	0,3%	58,3%	3,3%
RTG3	6	15	3	20,0%	0,0%	50,0%	0,0%
SKN7	6	15	0	0,0%	0,0%	0,0%	0,0%
SKO1	11	55	0	0,0%	0,4%	0,0%	3,6%
SNF2	6	15	3	20,0%	2,7%	50,0%	13,3%
SOK2	5	10	0	0,0%	0,0%	0,0%	0,0%
STE12	13	78	4	5,1%	0,8%	30,8%	9,2%
SWI1	5	10	3	30,0%	2,0%	60,0%	8,0%
SWI4	8	28	5	17,9%	2,9%	62,5%	17,5%
SWI5	8	28	8	28,6%	5,0%	87,5%	30,0%
SWI6	10	45	19	42,2%	2,2%	100,0%	20,0%
TEC1	42	861	7	0,8%	0,5%	23,8%	15,2%
THI2	9	36	0	0,0%	0,0%	0,0%	0,0%
TUP1	7	21	4	19,0%	1,9%	85,7%	11,4%
TYE7	6	15	1	6,7%	2,7%	33,3%	13,3%
UME6	26	325	13	4,0%	0,9%	42,3%	16,9%
UPC2	9	36	2	5,6%	0,6%	33,3%	4,4%
VID30	8	28	7	25,0%	2,9%	87,5%	20,0%
XBP1	5	10	0	0,0%	0,0%	0,0%	0,0%
YAP1	32	496	9	1,8%	0,5%	28,1%	15,0%
ZAP1	54	1431	34	2,4%	0,7%	25,9%	25,2%

Supplementary Table S4. Clusters of predicted co-regulated genes selected for experimental validation.

Shared motifs	Orphan genes	Annotated genes	Annotated Binding site
	YNL155W, YOR052C	RPN11, UBP6, RPT4, RPN9, PUP1, RPN2, RPN13, PRE9, RPN8, RPT6, RPT2, PUP2, RPN7, RPT1, RPT3, RPN6, RPN3, PRE4, PRE6, PRE3, SCL1, PRE1	Rpn4 binding site : GGTGGCAAA
	YPL272C[a], YHL026C, YDR531W, YML082W, YML083C	ERG24, ERG1, ERG4, ERG9, ERG5, ERG26, ERG10, ERG3, ERG7, ERG25, ERG11, ERG2	Upc2 binding site : TCGTATA TCGTTYAG
	YDL237W, YGR125W[a]	DAL2 , DAL5, GUD1, GZF3, DUR3, AMD2, DAL3, MEP2, HNM1, OPT1, OPT2 , LEE1, VAC17[b]	Gln3/Nil1 binding site : AGATAA GATAAG
	YNL313C[a], YPR174C[a], YJR030C, YDL156W, YJL181W, YDL010W, YDL012C	SPT21,RAD27, CLN1, SPO16, OPY2, GIC2, ERP3, CDC9, RNR1, CDC45, ERV25, BIM1,CTF4, EXO1, PMS1, BNI5, RTT107, DBF4, TOF1, CLB6, ASF1, RFC4, CDC7, RFA2, ENT1, SMC1, NSE4, DPB2, MSH6, POL2, UFE1, GIN4, MRC1, RAD54, MGS1, DSS4, RLF2, CDC21, ACM1, ELG1, HCM1, SHO1, POL30, POL1, MCD1, RAD53, ERP5, IRR1, ZDS2	Mbp1 binding site: ACGCGT

(a) Shown in the recent literature to be regulated by the mentioned transcription factors

(b) Not annotated as regulated by the Gln3/Nil1 factors

Supplementary Table S5. Primers used in this study.

Primers	Sequence (5' to 3')
5A RNR1	TGGGGAAAATCTCGTTACCA
3A RNR1	CGTTCCAGGGCTACGAAATA
5B RNR1	AACGCGTCATTTTGATCCTC
3B RNR1	CGAAACTACCTACGCCAACC
5B YDL156W	CCAGATGGCCAGGATAAGAA
3B YDL156W	AAAGGGCACTTTCGTCTTCA
5C YDL156W	GTTTCCTACTTGGTGGCACA
3C YDL156W	AACGCGTAATATGGGGTCCT
5A YJL181C	GCGCGTCAGTTTCAAGTTTT
3A YJL181C	ATAGCAGCTTCACCGTTCGT
5C YJL181C	GCGGGAGCTGTCATAACATT
3C YJL181C	ACTGTCATTTGCTGCCCTTC
5B YDL010W	TGGAAGGAAGCTTCGAGAAC
3B YDL010W	TGAAACGCCACATTTCACAT
5C YDL010W	ATCCCTCCACACAAGATTCG
5C YDL010W	ATGCGCGTTCCAGAAAGTTA
5A YDL012C	TGAAACGCCACATTTCACAT
3A YDL012C	TGGAAGGAAGCTTCGAGAAC
5C YDL012C	ATGCGCGTTCCAGAAAGTTA
3C YDL012C	ATCCCTCCACACAAGATTCG
5 RPN2	GGCAGTCTCGGATTCTAACG
3 RPN2	TGCGGTACCACATCTGACAT
5YNL155W	TTGCTAAAGGTGACCCCAAG
3YNL155W	CTACCCACAGGCCACATTTT
5YOR052C	AGCAGTAGGCGGGTGACTAA
3YOR052C	TGTTTCGAGCAAAAGTGACC
5YDR531W	AGCAGTAGGCGGGTGACTAA
3YDR531W	TGTTTCGAGCAAAAGTGACC
5YML082	TGCTTTTGCTTCCCCTCTTA
3YML082	TTTCCACGTCTTCTGCTCCT
5YML083	CGTGTCAGACCCTACCGATT
3YML083	TCGAGAGTGATGATGCTTGG
5ERG2	GGGTCTTCAACAATGCTGGT
3ERG2	TGTGTGCCCTTCAGTACCAA
5YHL026C	TGCCTTTGTATGCACTCCAG
3YHL026C	TTACCCGTGGAGCAAACTTC

10.3 Etude de la co-régulation des transporteurs

Nous disposons à présent d'un réseau de co-régulation impliquant plus de la moitié des gènes de *S.cerevisiae*. Nous allons utiliser ce réseau afin d'élargir nos connaissances sur la régulation des perméases.

10.3.1 Réseau induit par les transporteurs

La Figure 10.3 représente le réseau de co-régulation induit par les gènes contenus dans notre base de données. Cette représentation nous permet d'effectuer quelques observations.

10.3.1.1 Co-régulations au sein de transporteur de fonction similaire

Perméases soumises à la répression catabolique azotée La répression catabolique azotée ou NCR (pour Nitrogen Catabolite Repression) est un mécanisme de régulation transcriptionnelle par lequel en condition de croissance sur de bonnes sources d'azote (ion ammonium, glutamine, etc), la levure réprime l'expression de gènes permettant l'assimilation d'azote provenant de moins bonnes sources d'azote (urée, proline) permettant une croissance moins rapide. Cette répression est possible par l'intermédiaire de quatre facteurs de transcription (deux positifs et deux négatifs) se liant spécifiquement à une séquence consensus *GATAAG* couramment appelée séquence GATA. Les gènes soumis à la répression catabolique azotée présentent un bon nombre de ces séquences dans leur promoteur [174].

Sur la Figure 10.3, les gènes entourés en vert présentent des séquences GATA conservées et détectées par notre approche. Parmi ceux-ci, un bon nombre (surlignés en vert) ont déjà été identifiés comme étant soumis à la NCR [175]. Notons toutefois parmi ceux-ci, la présence des gènes *FCY21* et *HNM1* pour lesquels aucune régulation par la NCR n'a encore été décrite bien que ces deux gènes présentent des séquences GATA dans leur promoteur (Figure 10.4). *FCY21* code pour une perméase putative de purine / cytosine et *HNM1* pour une perméase de choline. Enfin, même si le site GATA n'est pas retrouvé avec un très haut score de significativité dans les promoteurs de *FCY2* et de ses orthologues (0.16), ce gène qui code pour un transporteur identifié de purine / cytosine contient deux occurences du motif GATA dans son promoteur, ce qui en fait une nouvelle cible potentielle de la NCR (Figure 10.4).

Régulation par le Fer Aft1p et Aft2p sont deux facteurs de transcriptions qui activent les gènes liés au métabolisme du fer en condition de carence en ce nutriment en se liant aux éléments cis-régulateurs de type PyPuCACCCPu situés dans leur promoteur [176]. Sur la Figure 10.3, les gènes entourés en mauve forment un groupe de perméases interconnectées toutes impliquées dans le transport de métaux. Toutes sont régulées par Aft1p et Aft2p [177]. La présence du gène *YMR279C*, codant pour une perméase putative, associé au gène *FTR1* codant pour la principale perméase du fer, semble être fortuite. En effet, d'une part ce gène ne présente qu'une seule séquence du motif Atf1 et d'autre part, un grand nombre d'empreintes très variées et vraisemblablement non spécifiques ont été découvertes dans le promoteur de ce gène.

Régulation du métabolisme soufré Chez la levure, les gènes impliqués dans le métabolisme du soufre sont régulés d'une part par le complexe Met4p-Cbf1p-Met28p qui reconnait le motif cis-régulateur TCACGTG et d'autre part le complexe Met31p-Met32p qui se fixe au motif CTGTGGC. Dans notre réseau, nous trouvons regroupés un certain nombre de gènes connus pour être soumis à ces régulations (surlignés en rouge) [177] et impliqués dans le transport de composés soufrés. Si l'on soumet l'ensemble de ces gènes (entourés en rouge) à une recherche des motifs surreprésentés dans leur promoteur, on ne trouve cependant que le motif reconnu par Met31p-Met32p. Notons parmi ces gènes, la présence du gène de la perméase *VBA2*, un transporteur d'acides aminés localisé à la membrane vacuolaire et de *YIL166C*, un gène apparenté à *YCT1* qui code pour une perméase identifiée récemment comme un transorteur de cystéine [178] (Figure 10.5).

Transporteurs d'acides aminés Les transporteurs d'acides aminés (entourés en noir) semblent former un groupe assez interconnecté. Cependant, aucun motif très convainquant ne ressort de l'analyse de leur promoteur. Ce groupe ne forme d'ailleurs pas un ensemble aussi dense que ce qui a pu être observé pour les groupes précédemment évoqués.

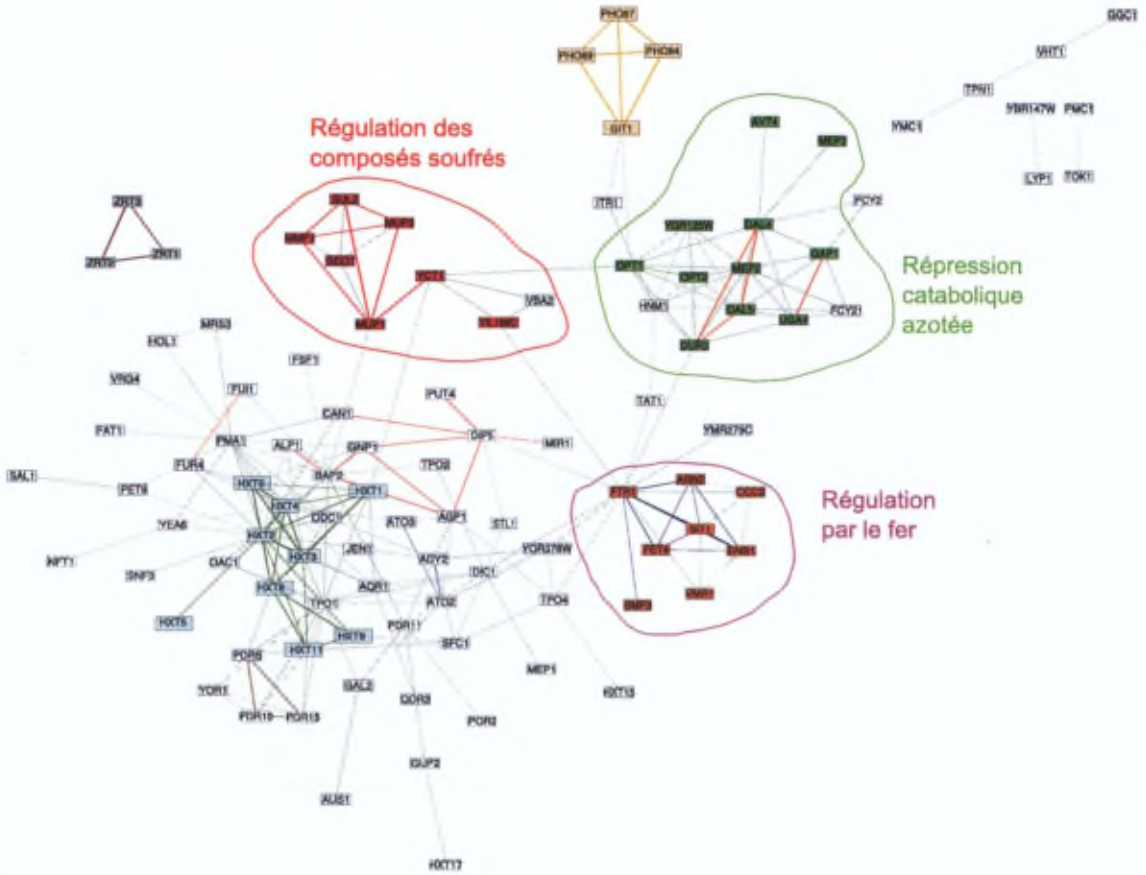


FIG. 10.3 – Réseau de co-régulation inféré induit par les gènes de la base de données YTPdb. Les arêtes colorées représentes co-régulation entre des transporteurs transportant le même type de substrat. L'épaisseur des arêtes est proportionnelle au DpBits.

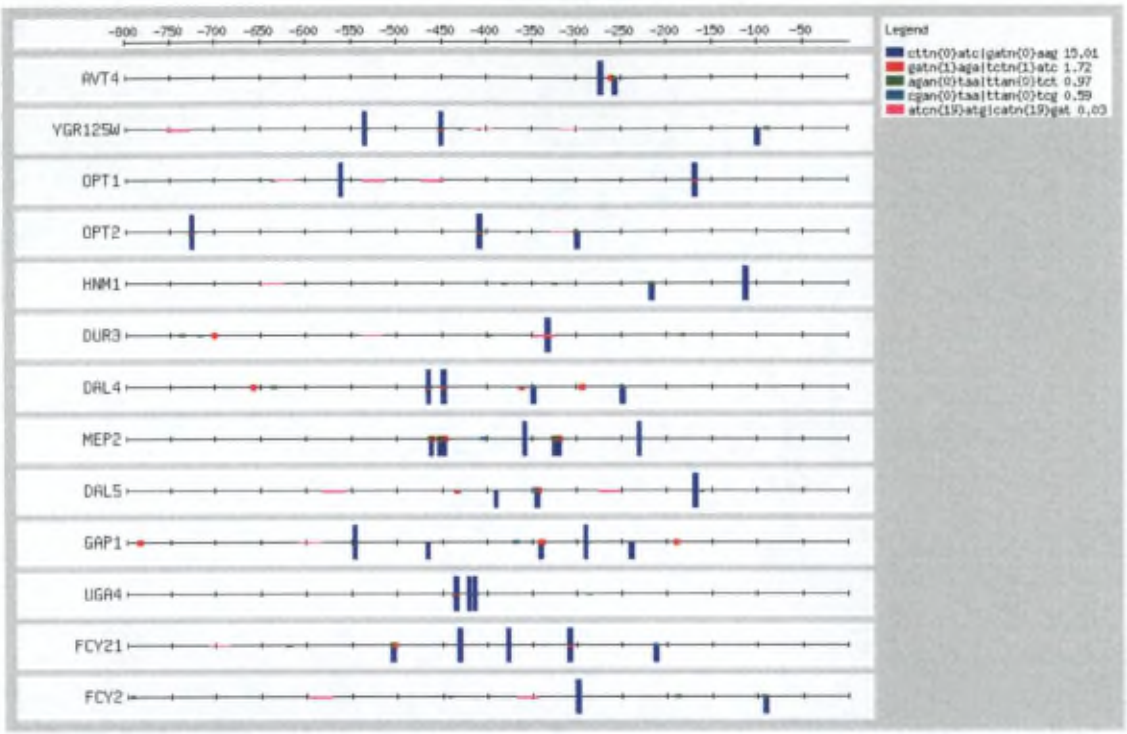


FIG. 10.4 – Feature-map de la découverte des motifs sur-représentés dans les promoteurs d'un groupe de gènes soumis potentiellement ou de manière établie à la répression catabolique azotée.

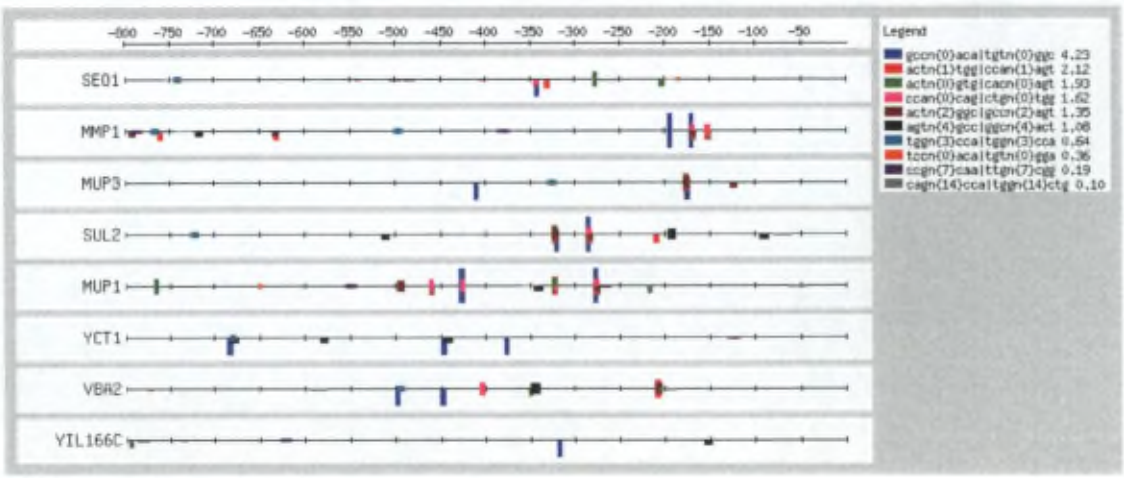


FIG. 10.5 – Feature-map de la découverte des motifs sur-représentés dans les promoteurs d'un groupe de gènes impliqués dans le métabolisme soufré.

Autres cas intéressants Citons encore le cas des gènes des perméases du zinc (*ZRT1*, *ZRT2* et *ZRT3*) qui sont régulées par l'activateur transcriptionnel Zap1p (surlignés en gris) et des gènes codant pour perméases du phosphate (*PHO87*, *PHO89 PHO84* et *GIT1*) qui sont activé par le facteur Pho4p (surlignés en jaune).

Enfin, il semblerait que notre méthode retrouve aussi les co-régulations entre les transporteurs d'hexose (gènes *HXT*) régulés par les facteurs Mig1p et Mig2p (surlignés en bleu clair).

10.3.2 Etude des groupes de voisins des transporteurs orphelins

Dans une deuxième étape, afin d'identifier la régulation et la fonction de transporteurs de fonctions inconnues, nous avons recherché au sein du réseau les voisins des gènes codants pour ces protéines. Sur les 59 protéines de fonctions inconnues de YTPdb, seuls 24 noeuds sont présents dans notre réseau de co-régulations.

Les groupes de voisins de ces noeuds ont été comparés avec les classes de processus biologiques de la Gene Ontology (GO) à l'aide du programme *compare-classes* afin de détecter si parmi les gènes prédits comme co-régulés avec le gène orphelin, un nombre significatif de membres d'une même classe fonctionnelles sont détectés. Notons que dans ce cas, le calcul de la P-valeur n'est pas exact et cette valeur ne doit donc être considérée qu'à titre indicatif. En effet, la distribution hypergéométrique est basée sur l'indépendance entre les classes alors que les différentes classes de la GO sont hiérarchiques. La Table 10.3 présente les résultats ainsi obtenus.

Le gène *YOR271C* semble être un faux positif. En effet, cette protéine est un transporteur mitochondrial et a peu de raison a priori d'être ainsi corégulé avec les gènes codant pour les protéines ribosomiales. Par ailleurs, les motifs découverts pour ce gène sont peu significatifs ce qui expliquerait que ce gène compte 137 voisins impliqués dans des processus très divers.

Par contre, trois gènes ont plus précisément retenu notre attention. Il s'agit de

- *YHL035C* (*VMR1*) : un gène codant pour un transporteur putatif de type ABC, localisé à la membrane vacuolaire. Il s'agit d'un transporteur putatif de type ABC localisé à la membrane vacuolaire. Le promoteur de ce gène présente un domaine de liaison pour les facteurs de transcription Aft1 et Aft2 (gcaccc) et l'activation de ce gène par ces facteurs transcriptionnel a été confirmé par des expériences de microarrays [179] (Figure 10.6) (Figure 10.6).
- *YLL055W* (*YCT1*) : un gène codant pour un transporteur dont l'expression est co-régulée avec les gènes du métabolisme soufré selon nos prédictions. Ce transporteur a tout récemment été idenfié comme étant un transporteur de cystéine de la membrane vacuolaire [178] et il a d'ailleurs été démontré que le gène était régulé par le facteur transcriptionnel Met4p
- *YHL008C* : un gène codant pour un transporteur présentant des similarités avec certains transporteurs d'acide formique bactérien. Selon Hu *et al*[180], il serait vacuolaire et n'a qu'un seul voisin dans le réseau, le gène *FDH1* qui code pour une formate dehydrogenase soumise à la régulation par le formate [181]. Leur motif commun très bien conservé (*K.lactis*, *Candida*, *Debaryomyces*) au cours de l'évolution est ccgn6cc qui n'a pas, à notre connaissance été identifié préalablement. Une étude préliminaire avait déterminé que ce transporteur ne transportait pas certains acides organiques comme le lactate, le pyruvate ou l'acétate à travers la membrane plasmique [182]. Par ailleurs, *FDH1* ne présente qu'un autre voisin dans le réseau, le gène *ACS1*, une acétyl COA synthétase (Figure 10.7).

10.3.3 Etude expérimentale du gène *YHL008C*

Nous avons décidé de nous pencher avec plus d'attention sur le gène *YHL008C* en investigant sa localisation, sa fonction et sa régulation en collaboration avec les docteurs Elsa Lauwers et Stéphan Vissers du laboratoire du Professeur Bruno André. Etant donné son unique lien au sein du réseau avec la formate déshydrogénase *FDH1* d'une part et d'autre part sa similarité de séquence avec des transporteurs de formate bactériens, nous pensons que *YHL008C* pourrait coder pour un transporteur de formate qui comme *FDH1* serait régulé par le formate. La levure n'est pas capable d'utiliser le formate comme seule source d'azote mais le formate peut servir de donneur d'électron afin de réduire le NAD+ en NADH et permet une croissance plus rapide de la levure [181].

TAB. 10.3 – Comparaison des groupes de voisins des perméases de fonctions inconnues dans YTPdb avec les classes de la Gene Ontology (*compare-classes*)

Classe de référence	Groupes de voisins	Nbre elem classe de référence	Nbre elem classe de voisins	Taille de l'intersection	Taille de l'union	P-valeur	E-valeur	Sig	Nom usuel
ribosome biogenesis	YOR271C	111	138	34	215	2.10E-018	1.00E-013	12.99	FSF1
ribonucleoprotein complex biogenesis and assembly	YOR271C	114	138	34	218	5.50E-018	2.60E-013	12.58	FSF1
rRNA processing	YOR271C	87	138	20	205	1.30E-008	0	3.2	FSF1
rRNA metabolic process	YOR271C	92	138	20	210	3.60E-008	0	2.76	FSF1
iron ion transport	YHL035C	14	9	4	19	7.10E-008	0	2.47	VMR1
siderophore transport	YHL035C	5	9	3	11	3.00E-007	0.01	1.83	VMR1
transition metal ion transport	YHL035C	27	9	4	32	1.20E-006	0.06	1.23	VMR1
di-, tri-valent inorganic cation transport	YHL035C	32	9	4	37	2.50E-006	0.12	0.92	VMR1
metal ion transport	YHL035C	40	9	4	45	6.20E-006	0.3	0.52	VMR1
branched chain family amino acid biosynthetic process	YOR271C	4	138	4	138	8.20E-006	0.4	0.4	FSF1
maturation of SSU-rRNA	YOR271C	22	138	8	152	1.00E-005	0.49	0.31	FSF1
sulfur metabolic process	YLL055W	35	45	6	74	2.40E-005	1.15	-0.06	YCT1
ribosomal large subunit biogenesis	YOR271C	13	138	6	145	2.80E-005	1.37	-0.14	FSF1
sulfur amino acid metabolic process	YLL055W	24	45	5	64	4.50E-005	2.18	-0.34	YCT1
cation transport	YHL035C	68	9	4	73	5.30E-005	2.55	-0.41	VMR1
siderophore-iron transport	YHL035C	4	9	2	11	6.60E-005	3.2	-0.51	VMR1
ion transport	YHL035C	80	9	4	85	0	4.85	-0.69	VMR1
sulfur amino acid biosynthetic process	YLL055W	16	45	4	57	0	6.38	-0.8	YCT1
RNA processing	YOR271C	211	138	24	325	0	12.92	-1.11	FSF1
methionine metabolic process	YLL055W	19	45	4	60	0	13.07	-1.12	YCT1
sulfur compound biosynthetic process	YLL055W	20	45	4	61	0	16.13	-1.21	YCT1
branched chain family amino acid metabolic process	YOR271C	8	138	4	142	0	23.44	-1.37	FSF1
DNA synthesis during DNA repair	YMR279C	2	61	2	61	0	27.2	-1.43	YMR279C
DNA synthesis during DNA repair	YDR093W	2	65	2	65	0	30.92	-1.49	DNF2
RNA methylation	YOR271C	9	138	4	143	0	40.42	-1.61	FSF1
formate catabolic process	YHL008C	1	3	1	3	0	56.83	-1.75	YHL008C
positive regulation of gene-specific transcription	YJR124C	1	3	1	3	0	56.83	-1.75	YJR124C
positive regulation of gene-specific transcription involved in unfolded protein response	YJR124C	1	3	1	3	0	56.83	-1.75	YJR124C
one-carbon compound catabolic process	YHL008C	1	3	1	3	0	56.83	-1.75	YHL008C
formate metabolic process	YHL008C	1	3	1	3	0	56.83	-1.75	YHL008C
methionine biosynthetic process	YLL055W	13	45	3	55	0	62.73	-1.8	YCT1
ribonucleoprotein complex assembly	YOR271C	51	138	9	180	0	64.27	-1.81	FSF1
mitotic sister chromatid cohesion	YMR279C	10	61	3	68	0	67.02	-1.83	YMR279C
chromatin assembly	YML038C	49	13	3	59	0	80.49	-1.91	YMD8
mitotic sister chromatid cohesion	YDR093W	10	65	3	72	0	80.66	-1.91	DNF2
positive regulation of gene-specific transcription	YKR103W	1	5	1	5	0	94.72	-1.98	NFT1
positive regulation of gene-specific transcription involved in unfolded protein response	YKR103W	1	5	1	5	0	94.72	-1.98	NFT1

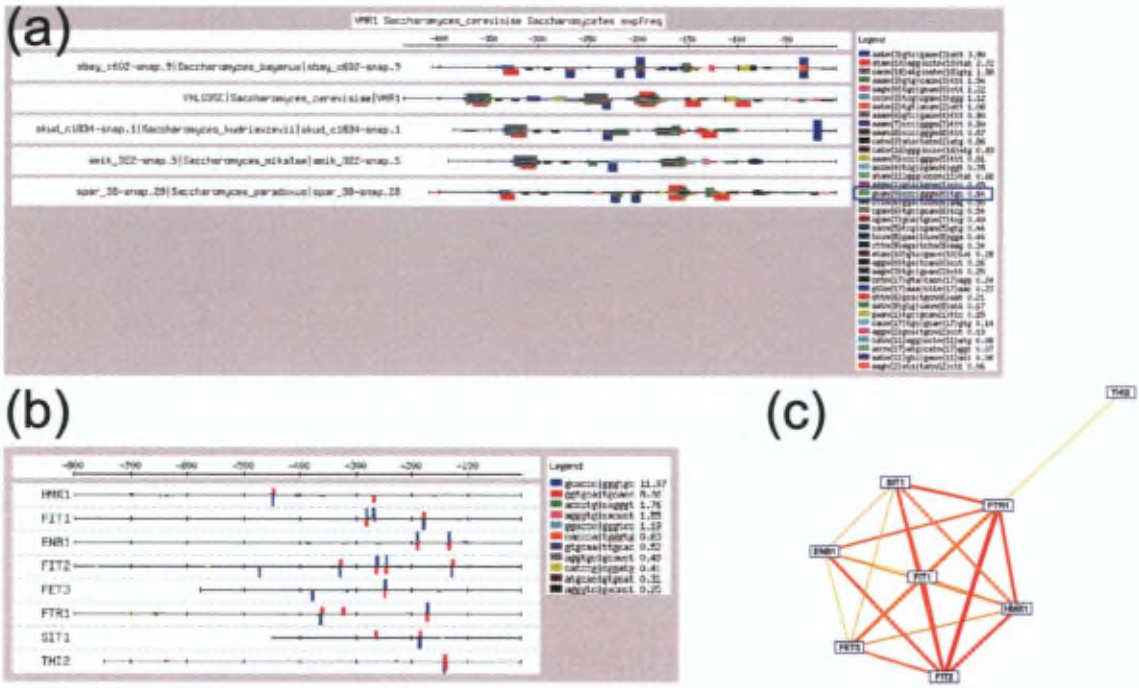


FIG. 10.6 – Analyse de la régulation du gène *VMR1*. (a) *Feature-map* des empreintes phylogénétiques détectées. (b) Prédiction des motifs cis-régulateurs au moyen de *oligo-analysis*. Le motif qui ressort avec le plus haut score est bien *gcaccc*, le motif cis-régulateur reconnus par Atf1p et Atf2p les activateurs transcriptionnels de gènes du métabolisme du fer. (c) Interactions entre VMR1 et ses voisins, tous impliqué dans le métabolisme du fer au sein du réseau de corégulation. L'échelle de couleur (du jaune au rouge) et l'épaisseur des arcs est proportionnelle au DpBits.

(a)

identifier	occ	exp_occ	occ_P	occ_E	occ_sig
cggn(5)tnn|ggan(5)cng	12	0.59	1.10E-007	1.10E-002	2.06
cggn(14)ggs|tncn(14)ccy	11	0.66	2.40E-006	2.30E-002	1.63
aaan(3)cng|cggn(3)ttt	17	1.84	1.10E-005	1.00E-001	0.98
atan(19)gac|gtcn(19)tat	17	1.75	1.30E-005	1.30E-001	0.89
acgn(2)tat|atan(2)cgt	17	1.87	1.70E-005	1.60E-001	0.79
aatn(8)aag|cttn(8)att	27	4.01	1.90E-005	1.90E-001	0.73
aacn(13)taa|ttan(13)gtt	24	3.35	3.80E-005	3.70E-001	0.43

(b)

identifier	occ	exp_occ	occ_P	occ_E	occ_sig
aggn(6)ccg|cggn(6)cct	13	0.6	1.70E-069	2.13E-065	4.68
cggn(18)ttn|gaan(18)cng	14	1.02	1.40E-007	1.70E-003	2.77
ycvn(0)caa|ttgn(0)ggg	16	1.52	3.10E-007	3.70E-003	2.43
cggn(5)tnn|ggan(5)cng	11	0.64	3.50E-007	4.20E-003	2.36
agtn(9)cnt|atgn(9)act	19	2.19	1.40E-006	1.70E-002	1.77
tagn(3)ggs|tcnn(3)cta	13	1.36	1.30E-005	1.60E-001	0.8
yggn(1)taa|tan(1)ccc	13	1.36	1.40E-005	1.60E-001	0.79
cggn(19)tcc|ggan(19)cng	9	0.62	1.50E-005	1.80E-001	0.76
ygan(18)tcc|ggan(18)tcc	8	0.49	3.60E-005	2.40E-001	0.62
cngn(11)ccg|cggn(11)cgg	8	0.55	5.80E-005	4.30E-001	0.37
aaan(5)cgg|ccgn(5)ttt	10	1.09	3.80E-005	4.50E-001	0.35
cncn(1)aac|gtnn(1)ggg	12	1.31	3.80E-005	4.50E-001	0.35
tncn(8)aaa|ttnn(8)gga	21	3.54	4.10E-005	6.80E-001	0.32
cggn(14)ggs|tcnn(14)ccg	9	0.73	4.60E-005	5.40E-001	0.27
aggn(7)cga|tcgn(7)cct	10	0.9	4.80E-005	5.70E-001	0.24
aatn(2)gaa|ttcn(2)agt	16	2.91	6.00E-006	7.20E-001	0.14
aatn(2)ccgg|ccgn(20)agt	9	0.74	6.10E-005	7.30E-001	0.14

(c)

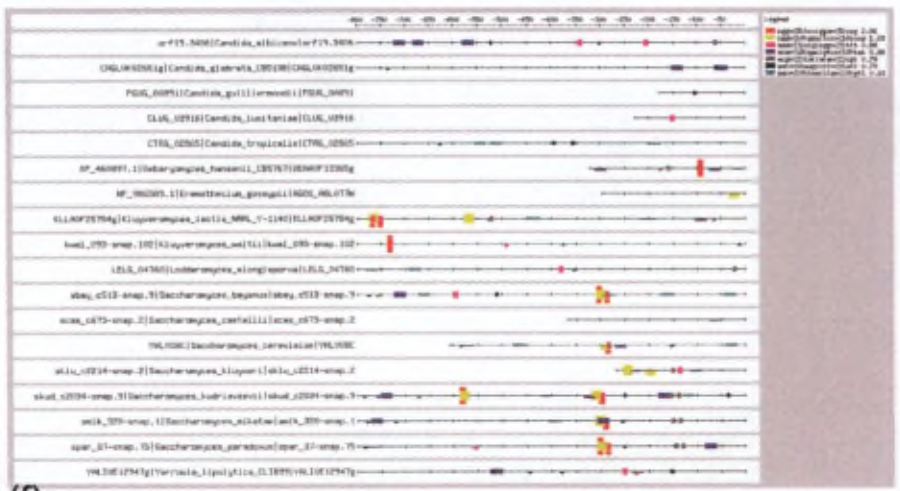
```
aggnnnnnnnng.          .cggnnnnnnnct
aggnnnnnnnnga          tcgnnnnnnnct
.ggannnnnnng.          .ccggnnnntcc.
aggannnnnccga          tcggnnnnntcct

cggnnnnnnnnnnnnnnnnnnttc..  ..gaatcnnnnnnnnnnnnnnnnccg
cggnnnnnnnnnnnnnnnnnntcc.  .ggannnnnnnnnnnnnnnnnccg
.ggannnnnnnnnnnnnnnnnnnng cggnnnnnnnnnnnnnnnnnatcc.
.ggannnnnnnnnnnnnnnnnnatcc. .ggannnnnnnnnnnnnnnnatcc.
..gaannnnnnnnnnnnnnnnnnng cggnnnnnnnnnnnnnnnnttc..
cggaannnnnnnnnnnnnnnnnnttccggaannnnnnnnnnnnnnnnttccg

ccccaa.                .ttgggg
cccnaac                gttnggg
ccccaac                gttgggg
```

(d)

(e)



(f)

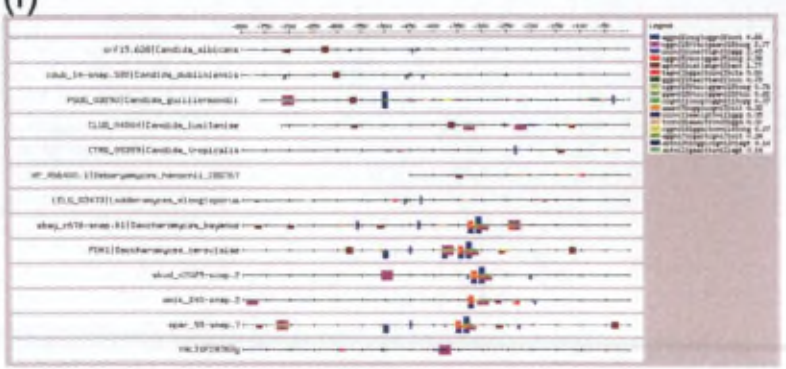


FIG. 10.7 – Analyse de la régulation des gènes *YHL008C* et *FDH1*. Empreintes phylogénétiques découvertes dans les promoteurs des gènes (a) YHL008C et (b) FDH1. (c) Assemblage des dyades découvertes dans le promoteur du gène *FDH1* (d) Voisins du gènes *FDH1* dans le réseau pondéré. *Feature-map* des empreintes phylogénétiques détectées dans les promoteurs de (e) *YHL008C* et (f) *FDH1*.

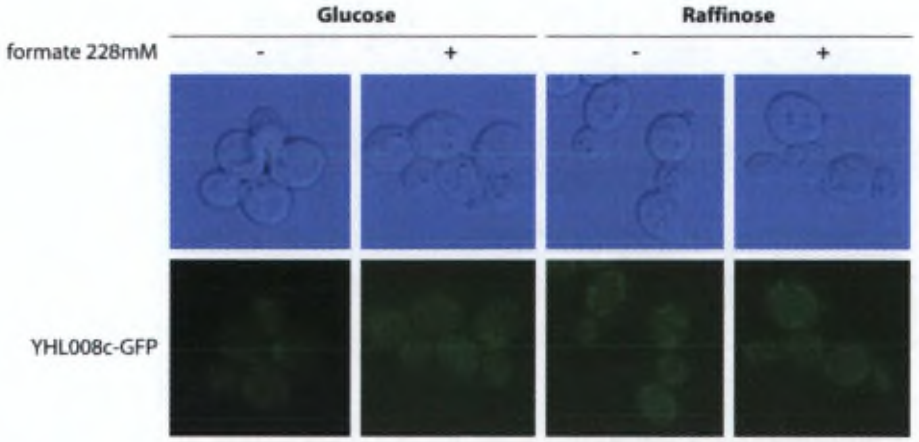


FIG. 10.8 – Etude de la localisation de la protéine Yhl008c-GFP dans la cellule. Expérience réalisée avec l'aide du Dr Elsa Lauwers. La concentration de formate correspond à celle définie par Overkamp *et al* [181] pour l'activation de *FDH1* par le formate.

10.3.3.1 Localisation de Yhl008c

Une méthode couramment utilisée en biologie moléculaire pour détecter dans quel compartiment cellulaire est exprimée une protéine consiste à exprimer cette protéine fusionnée avec la *GFP*, une protéine qui émet une fluoresence verte. La localisation sous-cellulaire des protéines est ensuite directement détectée en visualisant les cellules exprimant cette protéine de fusion avec un microscope à fluorescence.

Comme nous l'avons déjà mentionné précédemment, une étude globale de localisation sous-cellulaire de toutes les protéines de *S.cerevisiae* au moyen de protéines de fusion GFP a permis de localiser Yhl008c dans la vacuole [180]. Cependant, il est envisageable que cette localisation soit artéfactuelle. En effet, la protéine est très peu exprimée et semble être présente dans la lumière vacuolaire alors qu'il s'agit d'une protéine transmembranaire. Cette localisation pourrait dès lors être le fait de la dégradation de Yhl008c dans la vacuole car les conditions nécessaires à son expression ne sont pas rassemblées (la présence de formate, par exemple). C'est le cas par exemple, de la perméase d'acides aminés Gap1 détectée dans cette étude dans la lumière vacuolaire. En réalité, Gap1p est localisé à la membrane plasmique mais étant soumise à la NCR, Gap1 est dégradé dans la vacuole dans les conditions où les expériences ont été menées (milieu riche).

Nous avons donc étudié la localisation de Yhl008c fusionné à la GFP en présence de formate ou en absence de formate et en variant les sources de carbone (glucose ou raffinose) (Figure 10.8). Aucune modification dans la localisation ou dans l'intensité de la luminosité de Yhl008c n'est apparue en fonction de l'ajout de formate dans le milieu. En outre, le compartiment cellulaire dans lequel la protéine est exprimé n'est pas aisément identifiable même s'il s'agit vraisemblablement de la vacuole comme cela avait été indiqué par Huh et al [180].

10.3.3.2 Tests de croissance

Dans l'hypothèse où Yhl008c serait impliqué dans le transport du formate, nous avons voulu vérifier si l'on pouvait détecter un éventuel phénotype lié à la présence du formate dans une souche délétée du gène *YHL008C*. Nous avons comparé la croissance de souches sauvages et délétées respectivement des gène *FDH1* et *YHL008C* en présence de différentes concentration de formate (Figure 10.9). Nous pouvons observer qu'à haute concentration en formate (à partir de 200 mM), les clones *yhl008c°* sont plus petits que les clones sauvages ou *fdh1°*, ce qui indique une sensibilité accrue au formate. Le milieu de culture utilisé pour les tests étant tamponné, cette sensibilité au formate n'est donc pas due à une éventeulle diminution du pH suite à l'ajout du formate.

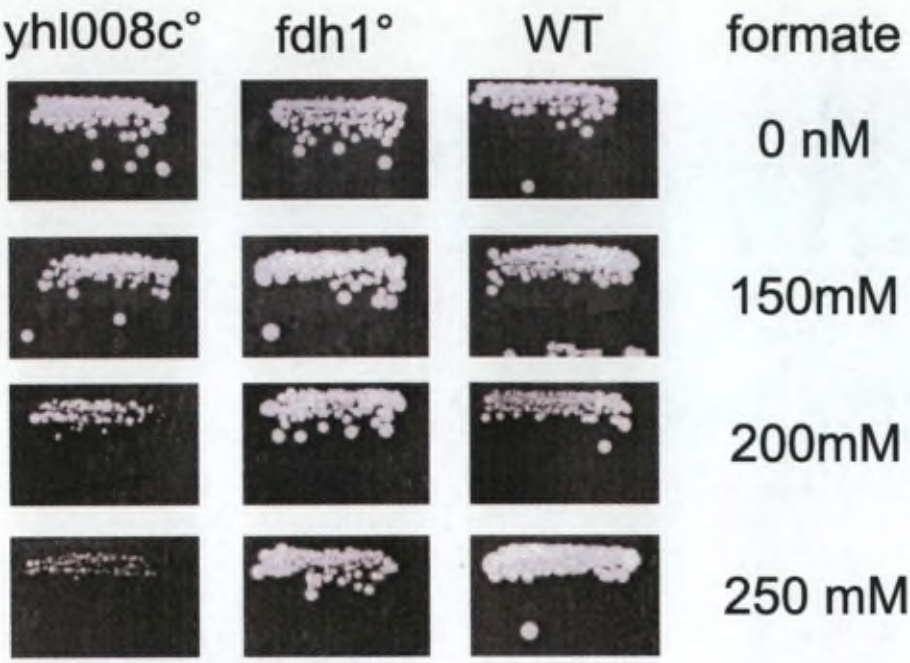


FIG. 10.9 – Croissance de souches *yhl008c°* (colonne 1), *fdh1°* (colonne 2) et sauvage (colonne 3) en présence de différentes concentration de formate.

10.3.3.3 Etude de la régulation par le formate

Dans une troisième étape, nous avons voulu vérifier la régulation du gène *YHL008C* par le formate en comparant l'expression du gène sur des milieux contenant du formate ou non. Cette expérience nous a permis de préciser que *FDH1* était bien régulé au niveau transcriptionnel par le formate, ce qui jusqu'à présent n'avait été détectée qu'au moyen de dosages enzymatiques [181]. Par manque de temps, nous n'avons pas encore eu l'occasion d'observer une telle régulation pour *YHL008C*.

10.3.3.4 Hypothèse quant au rôle de Yhl008c

L'analyse du phénotype et de la localisation de Yhl008c en présence de formate suggèrent que cette protéine pourrait être une perméase de formate qui *détoxifie* la cellule en exportant une partie du formate dans la vacuole lorsque celui-ci est présent à de trop fortes concentrations dans le cytosol.

10.4 Conclusion

Dans ce chapitre, nous présentons une méthode de génomique comparative qui permet de prédire des relations de co-régulations entre gènes avec une valeur prédictive élevée. Par ailleurs, la prédiction de relations de co-régulation entre gènes permet de dériver des interactions fonctionnelles entre les protéines.

L'article compris dans ce chapitre illustre la méthode, son évaluation et la validation expérimentale de plusieurs prédictions. Il fera très prochainement l'objet d'une publication.

Contrairement à l'approche des profils phylogénétiques que nous avons décrite et utilisée dans la section précédente, cette méthode semble bien mieux adaptée aux perméases. En effet, le fait de disposer d'organismes très proches n'est pas un problème car la détection d'empreintes phylogénétiques ne peut se faire sur des organismes trop distants phylogénétiquement. Par ailleurs, dans les cas où les séquences sont trop conservées, notre méthode comprend une

étape de *purge* qui retire de l'analyse les séquences conservées de plus d'une certaine taille de façon à ce que les dyades découvertes reflètent bien la sur-représentation et pas la conservation des séquences.

L'application de la méthode au réseau de perméases nous a permis de retrouver plusieurs régulations déjà connues (azote, hexose, zinc, etc) mais aussi de prédire la régulations et la fonction de certaines perméases putatives pour lesquelles nous ne disposions encore d'aucune information à ce jour (*VMR1*, *YHL008C* etc).

Troisième partie

Conclusion générale et Persepectives

Chapitre 11

Conclusion générale et perspectives

Les deux thèmes principaux abordés dans notre travail sont les réseaux (graphes) biologiques et les perméases de la levure *S.cerevisiae*. En effet, au cours de cette thèse, nous sommes passés par plusieurs étapes bioinformatiques et expérimentales afin d'étudier le réseau des interactions fonctionnelles et physiques entre transporteurs chez la levure.

11.1 Etude des interactions entre transporteurs de la levure

Les perméases sont des protéines hydrophobes enchâssées dans les membranes biologiques. Plus difficiles à étudier d'un point de vue biochimique, ces protéines sont donc moins bien caractérisées que les protéines solubles. Cependant, il est de plus en plus évident que des complexes homo- et hétéromériques existent entre les transporteurs [17], cependant les interactions entre ces protéines restent méconnues pour la plupart. Le but principal de notre thèse était donc d'étudier les perméases et de caractériser et préciser le réseau d'interaction entre perméases chez la levure en utilisant des méthodes bioinformatiques et expérimentales.

Implémentation de *YTPdb* Premièrement, afin de disposer d'un ensemble de données de référence, nous avons réimplémenté la base de données Yeast Transport Protein database (*YTPdb*) (Chapitre 6). Cette base de données, annotée par le Professeur Bruno André depuis 2001 [17] est dévolue à la classification et à l'annotation précise des transporteurs décrits ou putatifs de la levure sur base de données de la littérature. Le petit nombre de protéines (284 transporteurs annotés) contenu dans la base de données permet une annotation précise et facilement mise à jour. Une des caractéristiques majeures de *YTPdb* est la disposition hiérarchique des différentes classications (substrat, famille, localisation). *YTPdb* est une base de données relationnelle accessible en ligne via une interface web.

Etude de données expérimentales d'interaction à grande échelle Nous avons ensuite participé à *Associoport*, un un projet ambitieux visant à étudier l'entièreté des interactions physiques existant entre les perméases chez la levure. Pour cela, une méthode de double hybride à grande échelle adaptée aux protéines membranaires a été employée [57]. Ce projet, financé par l'Union Européenne a notamment impliqué les laboratoires des Professeurs Bruno André (Université Libre de Bruxelles) et Ekhard Boles (Université de Dusseldorf) qui présentent une grande expertise dans le domaine des perméases de levure. Nous avons apporté deux contributions principales à ce projet : d'une part, nous avons produit les données d'interactions pour plus de 40 perméases et d'autre part, nous avons procédé à l'analyse bioinformatique des données à grande échelle (environ 30000 interactions testées) (Chapitre 8).

Dans une première étape, nous avons vérifié la bonne qualité des données en étudiant le comportement des nombreux témoins. Ensuite, les données ont été standardisées et après une étape de filtrage, seules les interactions les plus significatives ont été conservées. Malgré la rigueur avec laquelle cette étude a été menée (nombreux contrôles) et les comportements globalement corrects des témoins, les résultats que nous avons extraits de ces expériences se sont cependant avérés assez décevants. En effet, d'une part l'intersection de nos données avec d'autres jeux de données d'interactions est non significative et pour la plupart, les interactions détectées semblent non valides biologiquement.

Dans une deuxième étape, les interactions filtrées et considérées comme significatives ont été utilisées afin de construire un réseau, les protéines étant représentées par des noeuds et les interactions par des arêtes. Nous avons appliqué des méthodes mathématiques issues de la théorie des graphes. D'une part, nous avons étudié les groupes de voisinage de chaque perméase au sein du réseau et d'autre part, nous avons essayé de détecter des modules de protéines densément interconnectées (clustering). Ces études ont apporté quelques résultats intéressants : plusieurs interactions connues ont été confirmées et certaines pistes de recherche ont été proposées (interactions entre transporteurs d'acides aminés, etc).

Inférence de réseaux d'interactions fonctionnelles En débutant l'analyse, nous savions qu'il était vraisemblable que les résulats obtenus à partir des données expérimentales ne nous satisfassent pas entièrement. En effet, même si les précautions apportées aux études à grande échelle afin d'en augmenter les performances sont de plus en plus nombreuses, les interactions détectées par ces études restent néanmoins moins fiables que celles détectées par les méthodes traditionnelles [52, 38, 139]. Nous avions donc prévu d'étudier les interactions entre les perméases en utilisant d'autres méthodes d'inférence de réseaux d'interactions fonctionnelles. Nous avons principalement eu recours aux méthodes de génomique comparative : l'étude des profils phylogénétiques (Chapitre 9) et l'inférence de réseaux de co-régulation (Chapitre 10). A cette fin, nous avons installé sur nos serveur plus de 30 génomes de *Fungi* provenant de multiples sources (voir Annexes) portant à 44 le nombre de génomes fongiques installés sur *RSAT*.

Profils phylogénétiques La méthode des profils phylogénétiques est basée sur l'étude de l'évolution corélée des gènes au cours de l'évolution, des protéines toujours (ou jamais) présentes ou absentes ensemble dans une série de génomes présentant plus de chances d'interagir fonctionnellement [58]. Premièrement, nous avons procédé à l'évaluation des prédictions en comparant nos prédictions à un réseau de référence constitué de voies métaboliques annotées. Cette étape nous permis d'une part de sélectionner un ensemble approprié d'organismes pour la comparaison des profils. D'autre part, nous avons pu estimer la précision des prédictions d'interactions fonctionnelles en fonction du score de significativité. L'approche semblait prometteuse puisque, dans notre évaluation, la valeur prédictive de notre approche est maximale à des scores élevés (pour des sensibités malgré tout assez faibles). En outre à des scores élevés de significativité statistique, notre réseau affichait clairement des groupes de protéines impliqué dans une même fonction cellulaire (protéasome, métabolisme des acides aminés).

Malgré cela, l'étude des profils phylogénétiques ne nous a pas permis de découvrir de nouvelles interactions fonctionnelles impliquant des transporteurs. En effet, les perfomances de cette approche découlent principalement de la détection des orthologues et nous avons constaté que les perméases de la levure semblaient bien conservées chez les *Fungi* mais très peu à l'extérieur de ce groupe taxonomique, et donc chez les procaryotes que nous avons considéré dans notre analyse. Il est vraisemblables que de meilleurs résultats pourraient être obtenus en envisageant d'autres méthodes de détection d'orthologues, présentant une sensibilité plus élevée [183].

Réseaux de co-régulation Enfin, en étroite collaboration avec Rekin's Janky et Fadi Abdel-Sater, nous avons avons contribué au développement d'une nouvelle méthode de génomique comparative permettant de prédire un réseau de co-régulation entre gènes [167]. Cette méthode ne prend comme données d'entrée que de séquences génétiques et ne nécessite aucune connaissance préalable de la régulation génétique de l'organisme étudié. En résumé, notre approche consiste à détecter les empreintes phylogénétiques de tous les gènes de l'organisme étudié et à construire le réseau en reliant chaque paire de gènes présentant des empreintes phylogénétiques similaires. Comme pour toutes les méthodes bioformatiques utilisées au cours de notre thèse, cette méthode a fait l'objet d'une évaluation rigoureuse. Premièrement, nous avons quantifié la qualité de nos prédictions en les comparant *in silico* avec plusieurs jeux de données de référence (co-régulation [172], co-expression [144] et co-liaison de facteurs transcriptionnels dans les promoteurs [173]). Nous avons pu estimer que les prédictions de notre approches effectuée uniquement àa partir de séquences des promoteurs de gènes orthologues présentait la même valeur prédictive qu'un réseau inféré à partir de centaines d'expériences de biopuce. Deuxièmement, nous avons validé 14 de nos prédictions de co-régulation (sur 16 testées) impliquant des gènes orphelins impliqués potentiellement dans des différents processus cellulaires (ergostérol, cycle cellulaire, répression catabolique azotée, dégradation des protéines).

Les mérites de la méthode ayant été quantifiés, nous avons regardé plus précisément le comportement des perméases au sein du réseau de co-régulation. Ainsi, nous avons pu retrouver et inférer les relations de régulation pour

plus de 100 perméases. Un bon nombre de ces régulations étaient déjà connues (régulation par l'azote, phosphate, fer, etc) mais nous avons proposer et testé quelques hypothèses quant à la régulation de gènes orphelins codant pour des perméases. Gardons à l'esprit que si ces prédictions n'avaient été effectuées sur un organisme modèle comme la levure, nous aurions énormément sur les régulations génétiques ayant lieu au sein de cet organisme en n'utilisant uniquement notre approche.

L'analyse des données de perméases a donc connu des développements plus et moins favorables. En effet, d'une part les résultats expérimentaux n'ont délivré que peu d'informations. Nous nous y attendions en partie et nous avons dès lors adapté et développé plusieurs méthodes de génomiques comparatives, basées sur des approches totalement différentes afin de contourner les problèmes liés à l'approche expérimentale. Comme nous l'avons observé, notre méthode de détection d'orthologues (meilleurs hits bidirectionnels), pourtant la plus courante en bioinformatique ne semble pas convenir aux perméases qui semblent (à l'exception de qulques unes) très bien conservées au sein des *Fungi* mais très peu chez les procaryotes. Nous devrions évaluer plusieurs possibilités parmi lesquelles : l'optimisation des paramètres de BLAST, l'utilisation d'un algorithme plus exact basé sur les alignements globaux et d'une méthode moins stringente que les meilleurs hits bidirectionnels. Dès lors, nous devrions obtenir plus de succès lors des approches de génomique comparative. Par ailleurs, optimiser la détection d'orthologues pour les perméases nous permettrait d'appliquer d'autres approches de génomiques comparatives de détection d'interactions fonctionnelles comme celles décrites dans notre introduction (analyse des fusions fissions, de la synténie).

11.2 Etudes de réseaux biologiques

Au cours de cette thèse, notre contribution au domaine des réseaux biologiques a été multiple.

Développement de nombreux outils de *NeAT* Premièrement, nous avons participé en grande partie au développement de *NeAT*, une suite d'outils dédiés à l'analyse des réseaux biologiques (bien que fonctionnant avec tout type de graphe). Au cours de la thèse, nous avons eu besoin d'outils capables d'une part de traiter rapidement des réseaux de tailles assez importantes et d'autre part de fonctionner en ligne de commande afin d'automatiser et de réitérer facilement les traitements sur les graphes. Au cours du développement de *NeAT* (dont les outils sont tous décrits au Chapitre 5), nous avons priviligié plusieurs aspects : l'efficacité, l'accès et la documentation. En effet, les outils de NeAT ont été conçus pour fonctionner rapidement avec des graphes de tailles courament observées dans de nombreuses études biologiques (plusieurs milliers de noeuds, dizaines de milliers d'arêtes). D'autre part, les différents outils de NeAT sont accessibles en ligne de commande, au travers d'une interface web ou de services web selon les connaissances ou les besoins des utilisateurs. Enfin, chacun des outils possède une aide bien documentée et sur le site web, un manuel d'utilisation présentant la résolution des problèmes biologiques concrets. Bien que de nombreux autres outils pré-existaient (*Cytoscape*, *GraphViz*, *Visant*, etc), l'apport de *NeAT* est non négligeable au vu des qualités énumérées.

Dans l'avenir, il serait néanmoins intéressant que les programmes proposé dans *NeAT* subissent de profondes modifications. En effet, bien que les performances des outils soient satisfaisantes dans la plupart des cas (c'est-à-dire pour des graphes de tailles couramment observées dans la littérature), nous sommes conscients que la taille des jeux de données à analyser va continuer de s'agrandir considérablement. Par exemple, selon les dernières prédictions, le réseau d'interaction entre protéines chez l'humain devrait compter plus de 650000 arêtes entre près de 30000 noeuds [184], ce qui est bien supérieur aux tailles des réseaux qui sont généralement traités par *NeAT* dans ce travail. Par ailleurs, certaines applications de *NeAT* comme le calcul des cliques, de certaines estimateurs topologiques et de la position graphique des noeuds (*layout*) demandent d'effectuer beaucoup de calculs. Dès lors, pour que les applications de NeAT continuent à fonctionner efficacement, il serait intéressant de les réimplémenter en utilisant un langage de programmation plus efficace pour les calculs intensifs comme le C ou le C++.

Nous pourrions aussi améliorer notre outil de visualisation de graphe. En effet, actuellement, la majorité des méthodes de *layout* de graphe sont basées sur la méthode des *champs de force* ou *spring embedding*. Ces méthodes positionnent les noeuds du graphe comme si les arêtes étaient des ressorts et les noeuds des particules chargées qui se repoussent. Ainsi, les noeuds des régions densément connectées auront tendances à être rapprochés car reliés entre eux par beaucoup d'assez de ressorts pour empêcher les noeuds de se repousser. C'est ce type de *layout* qui a principalement été utilisé pour la majorité des graphes représenté dans cette thèse et que nous avons implémenté dans `NeaT`. Bien que comme nous l'avons déjà mentionné, l'affichage et la visualisation de graphes fasse partie des points faibles

de *NeAT*, à notre connaissance, notre outil *display-graph* est le seul programme capable de traiter rapidement de gros graphes en ligne de commande. Nous aimerions améliorer la visualisation de graphes dans *NeAT* de deux façons. Tout d'abord en améliorant l'outil de visualisation de façon à ce qu'il rivalise avec des outils comme *Cytoscape* ou *yED* (qui ne sont malheureusement utilisables au travers d'une interface) et ensuite en incluant ou en développant une méthode de *layout* utilisant non seulement les champs de force mais prenant en compte les poids sur les arêtes (les ressorts des arêtes de poids plus élevés seraient ainsi plus tendus), permettant ainsi une meilleure représentation des prédictions de réseaux biologiques (les graphes inférés étant bien souvent pondérés en fonction de la significativité de la prédiction).

Evaluation quantitative de méthodes de clustering Comme nous l'avons évoqué dans la première partie de ce chapitre, nous avons appliqué des méthodes de clustering afin de trouver des groupes de perméases fortement interconnectées au sein du réseau d'interactions physiques entre les perméases (Associoport). Pour choisir un algorithme parmi les nombreux algorithmes existant, nous avons proposé une méthode permettant d'évaluer au moyen de métriques intuitives (sensibilité, valeur prédictive positive et précision) les performances d'un algorithme de clustering. Cette méthode d'évaluation nous a permis d'établir parmi différentes approches celle qui présentait les meilleurs performances et d'établir ses paramètres optimaux.

Notre méthode d'évaluation a déjà été reprise par plusieurs autres groupes afin d'estimer les performances de leur algorithme [37, 131, 154, 155, 156].

Quatrième partie

Annexes

Chapitre 12

Informations supplémentaires sur les sujet abordés dans la thèse

12.1 Aperçu biochimique de la formation de complexes protéiques

Dans la cellule, les polypeptides se reploient naturellement et adoptent la structure qui minimise l'enthalphie (énergie du système) et augmente l'entropie (désordre), en d'autres termes, les polypeptides auront tendance à adopter la conformation la plus stable et celle permettant le plus de liaisons (covalentes ou autres) avec le moins de contraintes positionnelles. Comme mentionné plus haut, le cytosol est aqueux, donc principalement constitué d'eau. La molécule d'eau présentant une forte polarité, les zones chargées positivement (les atomes d'hydrogène) d'une molécule d'eau sont attirées par les zones de charges négatives des molécules voisines (l'atome d'oxygène) ; ce phénomène est appelé *liaison hydrogène* (Figure 12.1). La présence de liaisons hydrogène diminue le niveau énergétique et augmente le désordre du système.

La présence de corps hydrophobe dans un milieu aqueux diminue l'entropie car il diminue le nombre de liaisons hydrogènes entre molécules d'eau qui ont donc tendance à repousser les composés apolaires de manière à minimiser leur contact avec elles ainsi que le nombre de liaisons hydrogène perdues entre molécules polaires. Lors du reploiement des polypeptides, les parties hydrophobes des protéines solubles sont enfouies dans le coeur de la protéine et à l'inverse, les parties hydrophobes des protéines intramembranaires sont exposées dans la bicouche lipidique hydrophobe.

Dans le cas des interactions entre protéines, la formation d'un complexe entre plusieurs protéines permet de diminuer globalement l'état énergétique du système.

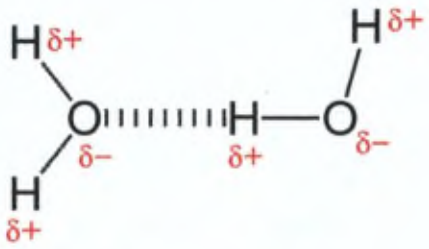


FIG. 12.1 – Représentation schématique de liaisons hydrogène au sein d'une molécule d'eau. Les électrons mis en communs entre l'hydrogène et l'oxygène d'une même molécule d'eau sont attirés par l'oxygène fortement électronégatif induisant la polarité de la molécule d'eau dont la charge reste cependant globalement neutre. Un atome d'hydrogène partiellement chargé positivement subit l'attraction d'un atome d'oxygène partiellement négatif d'une molécule voisine créant ainsi une liaison hydrogène.

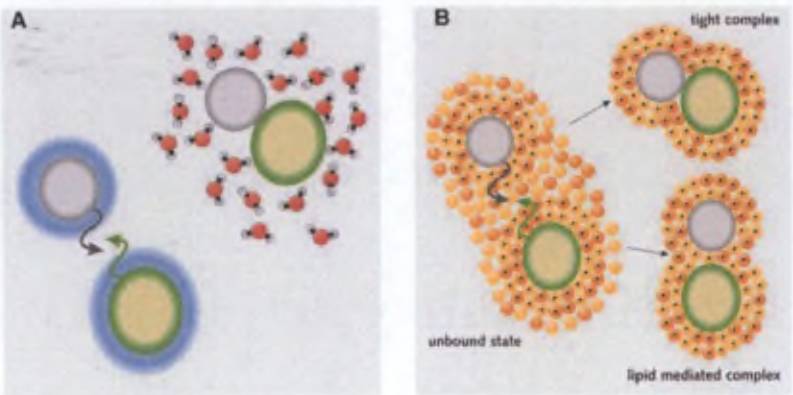


FIG. 12.2 – Effet hydrophobe et mécanisme similaire pour l'assemblage des protéines en complexes au sein des membranes. (A) Liaison de deux protéines (verte et grise) dans l'eau. La liaison est accompagnée d'une diminution de la surface de contact entre les protéines et les molécules d'eau. (B) Assemblage d'un complexe protéique transmembranaire. Les lipides dans la première et la seconde couche autour des protéines sont confinés (marqués par des points noirs) et donc défavorables à l'entropie. Leur nombre diminue lorsque les protéines s'associent. Extrait de [186].

Les principaux facteurs responsables des interactions entre protéines sont l'encombrement stérique (complémentarité des interfaces) ainsi que diverses liaisons non covalentes de différentes natures [185]. Parmi celles-i, on retrouve les liaisons hydrophobes (contacts entre parties hydrophobes de différentes protéines), les forces électrostatiques (reliant deux atomes de charge opposées), les liaisons hydrogène et les forces de Van Der Waals (interaction électrique de faible intensité entre atomes) (Figure 12.2A).

Les protéines transmembranaires sont étroitement entourées par des lipides orientés peu favorables au désordre en comparaison avec les autres phospholipides circulant librement dans la membrane. Dans le cas d'interactions entre ces protéines, l'association permet de réduire la surface protéique exposée et les lipides orientés peuvent donc circuler librement dans la bicouche lipidique ce qui est favorable à l'entropie globale du système [186] (Figure 12.2B).

Chapitre 13

Liste des *Fungi* installés dans RSAT et utilisés dans notre approche de génomique comparative

TAB. 13.1 – Organismes utilisés dans RAT

Nom de l'organisme	Phylogénie	Nombre de séquences codantes	Taille du génome complet
Aspergillus fumigatus	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Eurotiales ; Trichocomaceae ; mitosporic Trichocomaceae ; Aspergillus	9630	29.38
Candida dubliniensis	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; mitosporic Saccharomycetales ; Candida.	6685	15.42
Aspergillus terreus	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Eurotiales ; Trichocomaceae ; mitosporic Trichocomaceae ; Aspergillus	10402	29.33
Candida lusitaniae	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Metschnikowiaceae ; Clavispora	5893	24.23
Saccharomyces bayanus	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	5665	11.93
Saccharomyces kluyveri	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	6082	11.03
Uncinocarpus reesii	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Onygenales ; Onygenaceae ; Uncinocarpus	7790	44.66
Sclerotinia sclerotiorum	Ascomycota ; Pezizomycotina ; Leotiomycetes ; Helotiales ; Sclerotiniaceae ; Sclerotinia	14522	76.66
Saccharomyces castellii	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	5863	11.35
Saccharomyces mikatae	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	5883	11.47
Neurospora crassa	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Sordariomycetidae ; Sordariales ; Sordariaceae ; Neurospora	9817	77.4
Saccharomyces paradoxus	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	5623	11.87
Candida guilliermondii	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Pichia	5898	21.22
Phanerochaete chrysosporium	Basidiomycota ; Hymenomycetes ; Homobasidiomycetes ; Aphyllophorales ; Corticiaceae ; Phanerochaete	11630	29.97
Kluyveromyces waltii	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Kluyveromyces	5349	10.91
Aspergillus oryzae	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Eurotiales ; Trichocomaceae ; mitosporic Trichocomaceae ; Aspergillus.	12074	37.19
Trichoderma reesei	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Hypocreomycetidae ; Hypocreales ; Hypocreaceae ; Hypocrea	13247	34.49
Ustilago maydis	Basidiomycota ; Ustilaginomycetes ; Ustilaginomycetidae ; Ustilaginales ; Ustilaginaceae ; Ustilago	6523	39.37
Aspergillus nidulans	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Eurotiales ; Trichocomaceae ; Emericella	9541	60.14
Candida tropicalis	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; mitosporic Saccharomycetales ; Candida	6216	29.28
Chaetomium globosum	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Sordariomycetidae ; Sordariales ; Chaetomiaceae ; Chaetomium	11124	69.77
Coccidioides immitis	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Onygenales ; mitosporic Onygenales ; Coccidioides	10454	57.67
Magnaporthe grisea	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Sordariomycetes incertae sedis ; Magnaporthaceae ; Magnaporthe	11109	38.76
Rhizopus oryzae	Zygomycota ; Zygomycetes ; Mucorales ; Mucoraceae ; Rhizopus	16467	39.98
Saccharomyces kudriavzevii	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	6370	11.18
Stagonospora nodorum	Ascomycota ; Pezizomycotina ; Dothideomycetes ; Pleosporales ; Phaeosphaeriaceae ; Phaeosphaeria	16598	74.47
Encephalitozoon cuniculi	Microsporidia ; Unikaryonidae ; Encephalitozoon	1997	5
Eremothecium gossypii	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Eremothecium	4714	22.56
Schizosaccharomyces pombe	Ascomycota ; Schizosaccharomycetes ; Schizosaccharomycetales ; Schizosaccharomycetaceae ; Schizosaccharomyces	5045	50.08
Batrachochytrium dendrobatidis	Chytridiomycota ; Chytridiomycetes ; Chytridiales ; Chytridiales incertae sedis ; Batrachochytrium ; Batrachochytrium dendrobatidis	8781	23.72
Saccharomyces cerevisiae	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Saccharomyces	5880	12.16
Lodderomyces elongisporus	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Lodderomyces	5742	15.51
Puccinia graminis	Basidiomycota ; Pucciniomycotina ; Pucciniomycetes ; Pucciniales ; Pucciniaceae ; Puccinia	20534	88.64
Histoplasma capsulatum	Ascomycota ; Pezizomycotina ; Eurotiomycetes ; Eurotiomycetidae ; Onygenales ; Ajellomycetaceae ; Ajellomyces	9261	32.99
Candida albicans	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; mitosporic Saccharomycetales ; Candida	6114	14.32
Fusarium graminearum	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Hypocreomycetidae ; Hypocreales ; Nectriaceae ; Gibberella	13284	36.45
Fusarium oxysporum	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Hypocreomycetidae ; Hypocreales ; mitosporic Hypocreales ; Fusarium	17469	61.36
Fusarium verticillioides	Ascomycota ; Pezizomycotina ; Sordariomycetes ; Hypocreomycetidae ; Hypocreales ; Nectriaceae ; Gibberella	14161	41.78
Candida glabrata CBS138	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; mitosporic Saccharomycetales ; Candida	5182	12.28
Debaryomyces hansenii CBS767	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Debaryomyces	6432	12.22
Schizosaccharomyces japonicus yFS275	Ascomycota ; Taphrinomycotina ; Schizosaccharomycetes ; Schizosaccharomycetales ; Schizosaccharomycetaceae ; Schizosaccharomyce	5146	11.3
Yarrowia lipolytica CLIB99	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Dipodascaceae ; Yarrowia	6521	20.5
Cryptococcus neoformans var JEC21	Basidiomycota ; Hymenomycetes ; Heterobasidiomycetes ; Tremellomycetidae ; Tremellales ; Tremellaceae ; Filobasidiella	6475	19.05
Kluyveromyces lactis NRRL Y-1140	Ascomycota ; Saccharomycotina ; Saccharomycetes ; Saccharomycetales ; Saccharomycetaceae ; Kluyveromyces	5332	10.69

Chapitre 14

Glossaire

Graphe Un graphe consiste en une collection de noeuds (sommets, points) et d'arêtes qui connectent des paires de noeuds. Mathématiquement, un graphe G est un ensemble de sommets V connectés par des arêtes E, $G = (V, E)$.

Empreinte phylogénétique Motif conservé dans les régions non codantes ou promotrices de gènes orthologues. Ces régions correspondent éventuellement à des éléments de régulation transcriptionnelle.

Noeud, sommet Unité fondamentale du graphe.

Arête Lien connectant deux noeuds au sein d'un graphe.

Sous-graphe Un graphe $H = (E', V')$ est un sous-graphe de $G = (E, V)$ si ses arêtes et ses noeuds sont compris dans les arêtes et les noeuds de G, c'est-à-dire si $E' \subset E$ et $V' \subset V$.

Degré Nombre d'arêtes connectées à un noeud. Notons que le degré n'est pas toujours égal au nombre de voisins puisque certains graphes présentent plus d'une connexion entre deux noeuds. Les noeuds des graphes dirigés ont un degré entrant (in-degree) et un degré sortant (out-degree) qui sont les nombre d'arêtes qui atteignent ou qui partent des noeuds.

Coefficient de clustering Le coefficient de clustering d'un noeud est le rapport qu'il y a entre le nombre d'arêtes observés entre les voisins de ce noeud et le nombre maximal d'arêtes qu'il pourrait y avoir.

Circuit Un circuit est un chemin qui se termine qui a le même noeud de départ que d'arrivée.

Chemin Un chemin est une séquence d'arêtes consécutives dans un graphe. La longueur du chemin est la longueur de la séquence. Dans les graphes pondérés, la longueur du chemin peut-être la somme des poids des noeuds ou des arêtes.

Cut Une coupe est la partition des sommets d'un graphe en deux ensembles. Plus formellement, pour un graphe $G = (V, E)$, une coupe est la partition des noeuds V en deux ensembles S et T. Par ailleurs, on dit de toute arête $(u, v) \in E$ dont $u \in S$ et $v \in T$ qu'elle traverse la coupe. La taille d'une coupe est le nombre total d'arêtes qui traverse la coupe. Dans les graphes pondérés, la taille de la coupe est définie comme la somme des poids des arcs qui traversent la coupe.

Libre d'échelle (scale free) Propriété dont la distribution ne varie pas quelle que soit l'échelle à laquelle elle est considérée. Il s'agit d'une distribution de probabilité $p(x)$ pour une quantité x et que l'on peut calculer ainsi pour tout b $p(bx) = g(b)p(x)$. Cela signifie que si on augmente d'un facteur b l'échelle à laquelle on mesure x, la forme de la distribution ne changera pas à part pour une constante générale [72].

Bibliographie

[1] Gorter E. and Grendel F., 1925. *On bimolecular layers of lipoids on the chromocytes of the blood.* J Exp Med, 41(4) :439–443.

[2] van Meer G., Voelker D.R., and Feigenson G.W., Feb 2008. *Membrane lipids : where they are and how they behave.* Nat Rev Mol Cell Biol, 9(2) :112–124.

[3] Singer S.J. and Nicolson G.L., Feb 1972. *The fluid mosaic model of the structure of cell membranes.* Science, 175(23) :720–731.

[4] Engelman D.M., Dec 2005. *Membranes are more mosaic than fluid.* Nature, 438(7068) :578–580.

[5] Guschina I.A. and Harwood J.L., Oct 2006. *Mechanisms of temperature adaptation in poikilotherms.* FEBS Lett, 580(23) :5477–5483.

[6] Devaux P.F., Herrmann A., Ohlwein N., and Kozlov M.M., 2008. *How lipid flippases can modulate membrane structure.* Biochim Biophys Acta, 1778(7-8) :1591–1600.

[7] Maier A., VÃűlker B., Boles E., and Fuhrmann G.F., Dec 2002. *Characterisation of glucose transport in Saccharomyces cerevisiae with plasma membrane vesicles (countertransport) and intact cells (initial uptake) with single Hxt1, Hxt2, Hxt3, Hxt4, Hxt6, Hxt7 or Gal2 transporters.* FEMS Yeast Res, 2(4) :539–550.

[8] Berman H.M., Westbrook J., Feng Z., Gilliland G., Bhat T.N., Weissig H., Shindyalov I.N., and Bourne P.E., Jan 2000. *The Protein Data Bank.* Nucleic Acids Res, 28(1) :235–242.

[9] Savage D.F., Egea P.F., Robles-Colmenares Y., O'Connell J.D., and Stroud R.M., Dec 2003. *Architecture and selectivity in aquaporins : 2.5 a X-ray structure of aquaporin Z.* PLoS Biol, 1(3) :E72.

[10] Nagata T., Iizumi S., Satoh K., and Kikuchi S., Apr 2008. *Comparative molecular biological analysis of membrane transport genes in organisms.* Plant Mol Biol, 66(6) :565–585.

[11] Morsomme P., Slayman C.W., and Goffeau A., Nov 2000. *Mutagenic study of the structure, function and biogenesis of the yeast plasma membrane H(+)-ATPase.* Biochim Biophys Acta, 1469(3) :133–157.

[12] Andre B., Dec 1995. *An overview of membrane transport proteins in Saccharomyces cerevisiae.* Yeast, 11(16) :1575–1611.

[13] Rich D.P., Anderson M.P., Gregory R.J., Cheng S.H., Paul S., Jefferson D.M., McCann J.D., Klinger K.W., Smith A.E., and Welsh M.J., Sep 1990. *Expression of cystic fibrosis transmembrane conductance regulator corrects defective chloride channel regulation in cystic fibrosis airway epithelial cells.* Nature, 347(6291) :358–363.

[14] Mosser J., Douar A.M., Sarde C.O., Kioschis P., Feil R., Moser H., Poustka A.M., Mandel J.L., and Aubourg P., Feb 1993. *Putative X-linked adrenoleukodystrophy gene shares unexpected homology with ABC transporters.* Nature, 361(6414) :726–730.

[15] Saier M.H., Tran C.V., and Barabote R.D., Jan 2006. *TCDB : the Transporter Classification Database for membrane transport protein analyses and information.* Nucleic Acids Res, 34(Database issue) :D181–D186.

[16] Hediger M.A., Romero M.F., Peng J.B., Rolfs A., Takanaga H., and Bruford E.A., Feb 2004. *The ABCs of solute carriers : physiological, pathological and therapeutic implications of human membrane transport proteins.* Pflugers Arch, 447(5) :465–468.

[17] Van Belle D. and André B., Aug 2001. *A genomic view of yeast membrane transporters.* Curr Opin Cell Biol, 13(4) :389–398.

[18] Alberts B., Feb 1998. *The cell as a collection of protein machines : preparing the next generation of molecular biologists.* Cell, 92(3) :291–294.

[19] Srere P.A., Mar 2000. *Macromolecular interactions : tracing the roots.* Trends Biochem Sci, 25(3) :150–153.

[20] Mewes H.W., Amid C., Arnold R., Frishman D., Guldener U., Mannhaupt G., Munsterkotter M., Pagel P., Strack N., Stumpflen V., Warfsmann J., and Ruepp A., 2004. *MIPS : analysis and annotation of proteins from whole genomes.* Nucleic Acids Res, 32(Database issue) :D41–4.

[21] De Las Rivas J. and de Luis A., 2004. *Interactome data and databases : different types of protein interaction.* Comp Funct Genomics, 5(2) :173–178.

[22] Gutkind J.S., Jan 1998. *The pathways connecting G protein-coupled receptors to the nucleus through divergent mitogen-activated protein kinase cascades.* J Biol Chem, 273(4) :1839–1842.

[23] Prantl F., Strasser A., Aebi M., Furter R., Niederberger P., Kirschner K., and Huetter R., Jan 1985. *Purification and characterization of the indole-3-glycerolphosphate synthase/anthranilate synthase complex of Saccharomyces cerevisiae.* Eur J Biochem, 146(1) :95–100.

[24] Humphries M.J., 2000. *Integrin structure.* Biochem Soc Trans, 28(4) :311–339.

[25] Pena M.M., Puig S., and Thiele D.J., Oct 2000. *Characterization of the Saccharomyces cerevisiae high affinity copper transporter Ctr3.* J Biol Chem, 275(43) :33244–33251.

[26] Abe M., Hashimoto H., and Yoda K., Sep 1999. *Molecular characterization of Vig4/Vrg4 GDP-mannose transporter of the yeast Saccharomyces cerevisiae.* FEBS Lett, 458(3) :309–312.

[27] Kunji E.R.S., Apr 2004. *The role and structure of mitochondrial carriers.* FEBS Lett, 564(3) :239–244.

[28] Marini A.M., Springael J.Y., Frommer W.B., and Andr'e B., Jan 2000. *Cross-talk between ammonium transporters in yeast and interference by the soybean SAT1 protein.* Mol Microbiol, 35(2) :378–385.

[29] Ludewig U., Wilken S., Wu B., Jost W., Obrdlik P., Bakkoury M.E., Marini A.M., Andr'e B., Hamacher T., Boles E., von Wir'en N., and Frommer W.B., Nov 2003. *Homo- and hetero-oligomerization of ammonium transporter-1 NH4 uniporters.* J Biol Chem, 278(46) :45603–45610.

[30] Gilstring C.F., Melin-Larsson M., and Ljungdahl P.O., Nov 1999. *Shr3p mediates specific COPII coatomer-cargo interactions required for the packaging of amino acid permeases into ER-derived transport vesicles.* Mol Biol Cell, 10(11) :3549–3565.

[31] Uetz P., Giot L., Cagney G., Mansfield T.A., Judson R.S., Knight J.R., Lockshon D., Narayan V., Srinivasan M., Pochart P., Qureshi-Emili A., Li Y., Godwin B., Conover D., Kalbfleisch T., Vijayadamodar G., Yang M., Johnston M., Fields S., and Rothberg J.M., Feb 2000. *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.* Nature, 403(6770) :623–627.

[32] Ito T., Chiba T., Ozawa R., Yoshida M., Hattori M., and Sakaki Y., Apr 2001. *A comprehensive two-hybrid analysis to explore the yeast protein interactome.* Proc Natl Acad Sci U S A, 98(8) :4569–4574.

[33] Miller J.P., Lo R.S., Ben-Hur A., Desmarais C., Stagljar I., Noble W.S., and Fields S., Aug 2005. *Large-scale identification of yeast integral membrane protein interactions.* Proc Natl Acad Sci U S A, 102(34) :12123–12128.

[34] Ho Y., Gruhler A., Heilbut A., Bader G.D., Moore L., Adams S.L., Millar A., Taylor P., Bennett K., Boutilier K., Yang L., Wolting C., Donaldson I., Schandorff S., Shewnarane J., Vo M., Taggart J., Goudreault M., Muskat B., Alfarano C., Dewar D., Lin Z., Michalickova K., Willems A.R., Sassi H., Nielsen P.A., Rasmussen K.J., Andersen J.R., Johansen L.E., Hansen L.H., Jespersen H., Podtelejnikov A., Nielsen E., Crawford J., Poulsen V., SÃ¿rensen B.D., Matthiesen J., Hendrickson R.C., Gleeson F., Pawson T., Moran M.F., Durocher D., Mann M., Hogue C.W.V., Figeys D., and Tyers M., Jan 2002. *Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry.* Nature, 415(6868) :180–183.

[35] 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., HÃűfert 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., and Superti-Furga G., Jan 2002. *Functional organization of the yeast proteome by systematic analysis of protein complexes.* Nature, 415(6868) :141–147.

[36] Gavin A.C., Aloy P., Grandi P., Krause R., Boesche M., Marzioch M., Rau C., Jensen L.J., Bastuck S., Dumpelfeld B., Edelmann A., Heurtier M.A., Hoffman V., Hoefert C., Klein K., Hudak M., Michon A.M., Schelder M., Schirle M., Remor M., Rudi T., Hooper S., Bauer A., Bouwmeester T., Casari G., Drewes G., Neubauer G., Rick J.M., Kuster B., Bork P., Russell R.B., and Superti-Furga G., Mar 2006. *Proteome survey reveals modularity of the yeast cell machinery.* Nature, 440(7084) :631–636.

[37] Krogan N.J., Cagney G., Yu H., Zhong G., Guo X., Ignatchenko A., Li J., Pu S., Datta N., Tikuisis A.P., Punna T., Peregrin-Alvarez J.M., Shales M., Zhang X., Davey M., Robinson M.D., Paccanaro A., Bray J.E., Sheung A., Beattie B., Richards D.P., Canadien V., Lalev A., Mena F., Wong P., Starostine A., Canete M.M., Vlasblom J., Wu S., Orsi C., Collins S.R., Chandran S., Haw R., Rilstone J.J., Gandi K., Thompson N.J., Musso G., Onge P.S., Ghanny S., Lam M.H.Y., Butland G., Altaf-Ul A.M., Kanaya S., Shilatifard A., O'Shea E., Weissman J.S., Ingles C.J., Hughes T.R., Parkinson J., Gerstein M., Wodak S.J., Emili A., and Greenblatt J.F., Mar 2006. *Global landscape of protein complexes in the yeast Saccharomyces cerevisiae.* Nature, 440(7084) :637–643.

[38] von Mering C., Krause R., Snel B., Cornell M., Oliver S.G., Fields S., and Bork P., May 2002. *Comparative assessment of large-scale data sets of protein-protein interactions.* Nature, 417(6887) :399–403.

[39] Corsi A.K. and Schekman R., Nov 1996. *Mechanism of polypeptide translocation into the endoplasmic reticulum.* J Biol Chem, 271(48) :30299–30302.

[40] Stojanovski D., Johnston A.J., Streimann I., Hoogenraad N.J., and Ryan M.T., Jan 2003. *Import of nuclear-encoded proteins into mitochondria.* Exp Physiol, 88(1) :57–64.

[41] Simonis N., 2005. *Analyse bioinformatique de la regulation transcriptionnelle des complexes proteiques chez la levure Saccharomyces cerevisiae.* Ph.D. thesis, Universite Libre de Bruxelles.

[42] Orphanides G. and Reinberg D., Feb 2002. *A unified theory of gene expression.* Cell, 108(4) :439–451.

[43] Jansen R., Greenbaum D., and Gerstein M., Jan 2002. *Relating whole-genome expression data with protein-protein interactions.* Genome Res, 12(1) :37–46.

[44] Simonis N., Gonze D., Orsi C., van Helden J., and Wodak S.J., Oct 2006. *Modularity of the transcriptional response of protein complexes in yeast.* J Mol Biol, 363(2) :589–610.

[45] Droit A., Poirier G.G., and Hunter J.M., Apr 2005. *Experimental and bioinformatic approaches for interrogating protein-protein interactions to determine protein function.* J Mol Endocrinol, 34(2) :263–280.

[46] Tan K. and Ideker T., 2007. *Biological networks*, chapter Protein interaction networks, pages 133–161. World scientific.

[47] Fields S. and Song O., Jul 1989. *A novel genetic system to detect protein-protein interactions.* Nature, 340(6230) :245–246.

[48] Kaelin W.G., Pallas D.C., DeCaprio J.A., Kaye F.J., and Livingston D.M., Feb 1991. *Identification of cellular proteins that can interact specifically with the T/E1A-binding region of the retinoblastoma gene product.* Cell, 64(3) :521–532.

[49] Aebersold R. and Mann M., Mar 2003. *Mass spectrometry-based proteomics.* Nature, 422(6928) :198–207.

[50] Rual J.F., Venkatesan K., Hao T., Hirozane-Kishikawa T., Dricot A., Li N., Berriz G.F., Gibbons F.D., Dreze M., Ayivi-Guedehoussou N., Klitgord N., Simon C., Boxem M., Milstein S., Rosenberg J., Goldberg D.S., Zhang L.V., Wong S.L., Franklin G., Li S., Albala J.S., Lim J., Fraughton C., Llamosas E., Cevik S., Bex C., Lamesch P., Sikorski R.S., Vandenhaute J., Zoghbi H.Y., Smolyar A., Bosak S., Sequerra R., Doucette-Stamm L., Cusick M.E., Hill D.E., Roth F.P., and Vidal M., Oct 2005. *Towards a proteome-scale map of the human protein-protein interaction network.* Nature, 437(7062) :1173–1178.

[51] Walhout A.J., Sordella R., Lu X., Hartley J.L., Temple G.F., Brasch M.A., Thierry-Mieg N., and Vidal M., Jan 2000. *Protein interaction mapping in C. elegans using proteins involved in vulval development.* Science, 287(5450) :116–122.

[52] Sprinzak E., Sattath S., and Margalit H., 2003. *How reliable are experimental protein-protein interaction data ?* J Mol Biol, 327(5) :919–23.

[53] Droit A., 2007. *De l'identification a la caracterisation des complexes protÃ'iques : developpement d'une plateforme bioinformatique d'analyse.* Ph.D. thesis, DÃ'partement de biologie cellulaire et molÃ'culaire. Faculte de Medecine. UniversitÃ' Laval. Quebec. Canada.

[54] Joung J.K., Ramm E.I., and Pabo C.O., Jun 2000. *A bacterial two-hybrid selection system for studying protein-DNA and protein-protein interactions.* Proc Natl Acad Sci U S A, 97(13) :7382–7387.

[55] Johnsson N. and Varshavsky A., Oct 1994. *Split ubiquitin as a sensor of protein interactions in vivo.* Proc Natl Acad Sci U S A, 91(22) :10340–10344.

[56] Stagljar I., Korostensky C., Johnsson N., and te Heesen S., Apr 1998. *A genetic system based on split-ubiquitin for the analysis of interactions between membrane proteins in vivo.* Proc Natl Acad Sci U S A, 95(9) :5187–5192.

[57] Obrdlik P., El-Bakkoury M., Hamacher T., Cappellaro C., Vilarino C., Fleischer C., Ellerbrok H., Kamuzinzi R., Ledent V., Blaudez D., Sanders D., Revuelta J.L., Boles E., Andr'e B., and Frommer W.B., Aug 2004. *K+ channel interactions detected by a genetic system optimized for systematic studies of membrane protein interactions.* Proc Natl Acad Sci U S A, 101(33) :12242–12247.

[58] Pellegrini M., Marcotte E.M., Thompson M.J., Eisenberg D., and Yeates T.O., Apr 1999. *Assigning protein functions by comparative genome analysis : protein phylogenetic profiles.* Proc Natl Acad Sci U S A, 96(8) :4285–4288.

[59] Jothi R., Przytycka T.M., and Aravind L., 2007. *Discovering functional linkages and uncharacterized cellular pathways using phylogenetic profile comparisons : a comprehensive assessment.* BMC Bioinformatics, 8 :173.

[60] Marcotte E.M., Pellegrini M., Ng H.L., Rice D.W., Yeates T.O., and Eisenberg D., Jul 1999. *Detecting protein function and protein-protein interactions from genome sequences.* Science, 285(5428) :751–753.

[61] Enright A.J., Iliopoulos I., Kyrpides N.C., and Ouzounis C.A., Nov 1999. *Protein interaction maps for complete genomes based on gene fusion events.* Nature, 402(6757) :86–90.

[62] Overbeek R., Fonstein M., D'Souza M., Pusch G.D., and Maltsev N., 1999. *Use of contiguity on the chromosome to predict functional coupling.* In Silico Biol, 1(2) :93–108.

[63] Overbeek R., Fonstein M., D'Souza M., Pusch G.D., and Maltsev N., Mar 1999. *The use of gene clusters to infer functional coupling.* Proc Natl Acad Sci U S A, 96(6) :2896–2901.

[64] Goh C.S., Bogan A.A., Joachimiak M., Walther D., and Cohen F.E., Jun 2000. *Co-evolution of proteins with their interaction partners.* J Mol Biol, 299(2) :283–293.

[65] Goh C.S. and Cohen F.E., Nov 2002. *Co-evolutionary analysis reveals insights into protein-protein interactions.* J Mol Biol, 324(1) :177–192.

[66] Beadle G.W. and Tatum E.L., Nov 1941. *Genetic Control of Biochemical Reactions in Neurospora.* Proc Natl Acad Sci U S A, 27(11) :499–506.

[67] Legrain P. and Vidal M., 2007. *Biologie systemique*, chapter Introduction : Standards et modeles pour la biologie, pages 16–20. Omniscience.

[68] Huber W., Carey V.J., Long L., Falcon S., and Gentleman R., 2007. *Graphs in molecular biology.* BMC Bioinformatics, 8 Suppl 6 :S8.

[69] Barabasi and Albert, Oct 1999. *Emergence of scaling in random networks.* Science, 286(5439) :509–512.

[70] Jeong H., Tombor B., Albert R., Oltvai Z.N., and BarabÃąsi A.L., Oct 2000. *The large-scale organization of metabolic networks.* Nature, 407(6804) :651–654.

[71] Newman M.E.J., 2003. *The structure and function of complex networks.* SIAM Review, 45 :167–256.

[72] Newman M.E.J., 2005. *Power laws, Pareto distributions and Zipf's law.* Contemporary Physics, 46 :323.

[73] Barabasi A.L. and Oltvai Z.N., Feb 2004. *Network biology : understanding the cell's functional organization.* Nat Rev Genet, 5(2) :101–113.

[74] Han J.D.J., Dupuy D., Bertin N., Cusick M.E., and Vidal M., Jul 2005. *Effect of sampling on topology predictions of protein-protein interaction networks.* Nat Biotechnol, 23(7) :839–844.

[75] Khanin R. and Wit E., Apr 2006. *How scale-free are biological networks.* J Comput Biol, 13(3) :810–818.

[76] Ravasz E., Somera A.L., Mongru D.A., Oltvai Z.N., and BarabÃąsi A.L., Aug 2002. *Hierarchical organization of modularity in metabolic networks.* Science, 297(5586) :1551–1555.

[77] Milo R., Shen-Orr S., Itzkovitz S., Kashtan N., Chklovskii D., and Alon U., Oct 2002. *Network motifs : simple building blocks of complex networks.* Science, 298(5594) :824–827.

[78] Maslov S. and Sneppen K., May 2002. *Specificity and stability in topology of protein networks.* Science, 296(5569) :910–913.

[79] Aloy P. and Russell R.B., Oct 2002. *Potential artefacts in protein-interaction networks.* FEBS Lett, 530(1-3) :253–254.

[80] Maslov S. and Sneppen K., Oct 2002. *Protein interaction networks beyond artifacts.* FEBS Lett, 530(1-3) :255–256.

[81] Batada N.N., Reguly T., Breitkreutz A., Boucher L., Breitkreutz B.J., Hurst L.D., and Tyers M., Oct 2006. *Stratus not altocumulus : a new view of the yeast protein interaction network.* PLoS Biol, 4(10) :e317.

[82] Zhu X., Gerstein M., and Snyder M., May 2007. *Getting connected : analysis and principles of biological networks.* Genes Dev, 21(9) :1010–1024.

[83] Jeong H., Mason S.P., BarabÃąsi A.L., and Oltvai Z.N., May 2001. *Lethality and centrality in protein networks.* Nature, 411(6833) :41–42.

[84] Bairoch A., Jan 2000. *The ENZYME database in 2000.* Nucleic Acids Res, 28(1) :304–305.

[85] Croes D., Couche F., Wodak S.J., and van Helden J., Jul 2005. *Metabolic PathFinding : inferring relevant pathways in biochemical networks.* Nucleic Acids Res, 33(Web Server issue) :W326–W330.

[86] Croes D., 2005. *Recherches de chemins dans le r'eseau m'etabolique et mesure de la distance m'etabolique entre enzymes.* Ph.D. thesis, UniversitÃl' Libre de Bruxelles.

[87] Albert, Jeong, and Barabasi, Jul 2000. *Error and attack tolerance of complex networks.* Nature, 406(6794) :378–382.

[88] van Helden J., Wernisch L., Gilbert D., and Wodak S.J., 2002. *Graph-based analysis of metabolic networks.* Ernst Schering Res Found Workshop, (38) :245–274.

[89] Arita M., Feb 2004. *The metabolic world of Escherichia coli is not small.* Proc Natl Acad Sci U S A, 101(6) :1543–1547.

[90] Croes D., Couche F., Wodak S.J., and van Helden J., Feb 2006. *Inferring meaningful pathways in weighted metabolic networks.* J Mol Biol, 356(1) :222–236.

[91] Tarassov K., Messier V., Landry C.R., Radinovic S., Molina M.M.S., Shames I., Malitskaya Y., Vogel J., Bussey H., and Michnick S.W., Jun 2008. *An in vivo map of the yeast protein interactome.* Science, 320(5882) :1465–1470.

[92] Yu H., Greenbaum D., Lu H.X., Zhu X., and Gerstein M., Jun 2004. *Genomic analysis of essentiality within protein networks.* Trends Genet, 20(6) :227–231.

[93] Fraser H.B., Hirsh A.E., Steinmetz L.M., Scharfe C., and Feldman M.W., Apr 2002. *Evolutionary rate in the protein interaction network.* Science, 296(5568) :750–752.

[94] Coulomb S., Bauer M., Bernard D., and Marsolier-Kergoat M.C., Aug 2005. *Gene essentiality and the topology of protein interaction networks.* Proc Biol Sci, 272(1573) :1721–1725.

[95] He X. and Zhang J., Jun 2006. *Why do hubs tend to be essential in protein networks ?* PLoS Genet, 2(6) :e88.

[96] Zotenko E., Mestre J., O'Leary D.P., and Przytycka T.M., Aug 2008. *Why do hubs in the yeast protein interaction network tend to be essential : reexamining the connection between the network topology and essentiality.* PLoS Comput Biol, 4(8) :e1000140.

[97] Han J.D.J., Bertin N., Hao T., Goldberg D.S., Berriz G.F., Zhang L.V., Dupuy D., Walhout A.J.M., Cusick M.E., Roth F.P., and Vidal M., Jul 2004. *Evidence for dynamically organized modularity in the yeast protein-protein interaction network.* Nature, 430(6995) :88–93.

[98] Fraser H.B., Apr 2005. *Modularity and evolutionary constraint on proteins.* Nat Genet, 37(4) :351–352.

[99] Batada N.N., Reguly T., Breitkreutz A., Boucher L., Breitkreutz B.J., Hurst L.D., and Tyers M., Jun 2007. *Still stratus not altocumulus : further evidence against the date/party hub distinction.* PLoS Biol, 5(6) :e154.

[100] Bertin N., Simonis N., Dupuy D., Cusick M.E., Han J.D.J., Fraser H.B., Roth F.P., and Vidal M., Jun 2007. *Confirmation of organized modularity in the yeast interactome.* PLoS Biol, 5(6) :e153.

[101] Babu M.M., Luscombe N.M., Aravind L., Gerstein M., and Teichmann S.A., Jun 2004. *Structure and evolution of transcriptional regulatory networks.* Curr Opin Struct Biol, 14(3) :283–291.

[102] Teichmann S.A. and Babu M.M., Oct 2002. *Conservation of gene co-regulation in prokaryotes and eukaryotes.* Trends Biotechnol, 20(10) :407–10 ; discussion 410.

[103] Guelzim N., Bottani S., Bourgine P., and KÃl'pÃÍs F., May 2002. *Topological and causal structure of the yeast transcriptional regulatory network.* Nat Genet, 31(1) :60–63.

[104] Lee T.I., Rinaldi N.J., Robert F., Odom D.T., Bar-Joseph Z., Gerber G.K., Hannett N.M., Harbison C.T., Thompson C.M., Simon I., Zeitlinger J., Jennings E.G., Murray H.L., Gordon D.B., Ren B., Wyrick J.J., Tagne J.B., Volkert T.L., Fraenkel E., Gifford D.K., and Young R.A., Oct 2002. *Transcriptional regulatory networks in Saccharomyces cerevisiae.* Science, 298(5594) :799–804.

[105] Dobrin R., Beg Q.K., BarabÃąsi A.L., and Oltvai Z.N., Jan 2004. *Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network.* BMC Bioinformatics, 5 :10.

[106] Wuchty S., Sep 2001. *Scale-free behavior in protein domain networks.* Mol Biol Evol, 18(9) :1694–1702.

[107] Diao Y., Li M., Feng Z., Yin J., and Pan Y., Aug 2007. *The community structure of human cellular signaling network.* J Theor Biol, 247(4) :608–615.

[108] Yildirim M.A., Goh K.I., Cusick M.E., BarabÃąsi A.L., and Vidal M., Oct 2007. *Drug-target network.* Nat Biotechnol, 25(10) :1119–1126.

[109] Vendruscolo M., Dokholyan N.V., Paci E., and Karplus M., Jun 2002. *Small-world view of the amino acids that play a key role in protein folding.* Phys Rev E Stat Nonlin Soft Matter Phys, 65(6 Pt 1) :061910.

[110] Sporns O. and Zwi J.D., 2004. *The small world of the cerebral cortex.* Neuroinformatics, 2(2) :145–162.

[111] Brohee S. and van Helden J., 2006. *Evaluation of clustering algorithms for protein-protein interaction networks.* BMC Bioinformatics, 7 :488.

[112] Felsenstein J., 1982. *Numerical Methods for Inferring Evolutionary Trees.* The Quarterly Review of Biology, 57(4) :379.

[113] Enright A.J., Dongen S.V., and Ouzounis C.A., Apr 2002. *An efficient algorithm for large-scale detection of protein families.* Nucleic Acids Res, 30(7) :1575–1584.

[114] Tetko I.V., Facius A., Ruepp A., and Mewes H.W., 2005. *Super paramagnetic clustering of protein sequences.* BMC Bioinformatics, 6 :82.

[115] Eisen M.B., Spellman P.T., Brown P.O., and Botstein D., Dec 1998. *Cluster analysis and display of genome-wide expression patterns.* Proc Natl Acad Sci U S A, 95(25) :14863–14868.

[116] Sharan R., Ulitsky I., and Shamir R., 2007. *Network-based prediction of protein function.* Mol Syst Biol, 3 :88.

[117] Arnau V., Mars S., and Marin I., Feb 2005. *Iterative cluster analysis of protein interaction data.* Bioinformatics, 21(3) :364–378.

[118] Brun C., Chevenet F., Martin D., Wojcik J., GuÃl'noche A., and Jacq B., 2003. *Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network.* Genome Biol, 5(1) :R6.

[119] Samanta M.P. and Liang S., Oct 2003. *Predicting protein functions from redundancies in large-scale protein interaction networks.* Proc Natl Acad Sci U S A, 100(22) :12579–12583.

[120] Gagneur J., Jackson D.B., and Casari G., May 2003. *Hierarchical analysis of dependency in metabolic networks.* Bioinformatics, 19(8) :1027–1034.

[121] Hartuv E. and Shamir R., 1999. *A clustering algorithm based on Graph Connectivity.* Information Processing Letters.

[122] Koyuturk M., Szpankowski W., and Grama A., 2007. *Assessing significance of connectivity and conservation in protein interaction networks.* J Comput Biol, 14(6) :747–764.

[123] Palla G., DerÃl'nyi I., Farkas I., and Vicsek T., Jun 2005. *Uncovering the overlapping community structure of complex networks in nature and society.* Nature, 435(7043) :814–818.

[124] Adamcsek B., Palla G., Farkas I.J., Derenyi I., and Vicsek T., Apr 2006. *CFinder : locating cliques and overlapping modules in biological networks.* Bioinformatics, 22(8) :1021–1023.

[125] Zhang S., Ning X., and Zhang X.S., Dec 2006. *Identification of functional modules in a PPI network by clique percolation clustering.* Comput Biol Chem, 30(6) :445–451.

[126] King A.D., Przulj N., and Jurisica I., Nov 2004. *Protein complex prediction via cost-based clustering.* Bioinformatics, 20(17) :3013–3020.

[127] Spirin V. and Mirny L.A., Oct 2003. *Protein complexes and functional modules in molecular networks.* Proc Natl Acad Sci U S A, 100(21) :12123–12128.

[128] Blatt M., Wiseman S., and Domany E., 1996. *Superparamagnetic clustering of data.* Phys Rev Lett., 76(18) :3251–3254.

[129] Van Dongen S., 2000. *Graph clustering by flow simulation.* Ph.D. thesis, Centers for mathematics and computer science (CWI), University of Utrecht.

[130] Bader G.D. and Hogue C.W.V., Jan 2003. *An automated method for finding molecular complexes in large protein interaction networks.* BMC Bioinformatics, 4 :2.

[131] Lima-Mendez G., Helden J.V., Toussaint A., and Leplae R., Apr 2008. *Reticulate representation of evolutionary and functional relationships between phage genomes.* Mol Biol Evol, 25(4) :762–777.

[132] Suderman M. and Hallett M., Oct 2007. *Tools for visually exploring biological networks.* Bioinformatics, 23(20) :2651–2659.

[133] Shannon P., Markiel A., Ozier O., Baliga N.S., Wang J.T., Ramage D., Amin N., Schwikowski B., and Ideker T., Nov 2003. *Cytoscape : a software environment for integrated models of biomolecular interaction networks.* Genome Res, 13(11) :2498–2504.

[134] Hu Z., Snitkin E.S., and DeLisi C., Jul 2008. *VisANT : an integrative framework for networks in systems biology.* Brief Bioinform, 9(4) :317–325.

[135] Freeman T.C., Goldovsky L., Brosch M., van Dongen S., MaziÃÍre P., Grocock R.J., Freilich S., Thornton J., and Enright A.J., Oct 2007. *Construction, visualisation, and clustering of transcription networks from microarray expression data.* PLoS Comput Biol, 3(10) :2032–2042.

[136] de Nooy W., Mrvar A., and Batagelj V., 2005. *Exploratory Social Network Analysis with Pajek (Structural Analysis in the Social Sciences).* Cambridge University Press.

[137] Brohee S., Faust K., Lima-Mendez G., Sand O., Janky R., Vanderstocken G., Deville Y., and van Helden J., Jul 2008. *NeAT : a toolbox for the analysis of biological networks, clusters, classes and pathways.* Nucleic Acids Res, 36(Web Server issue) :W444–W451.

[138] Yip K.Y., Yu H., Kim P.M., Schultz M., and Gerstein M., Dec 2006. *The tYNA platform for comparative interactomics : a web tool for managing, comparing and mining multiple networks.* Bioinformatics, 22(23) :2968–2970.

[139] Hart G.T., Ramani A.K., and Marcotte E.M., 2006. *How complete are current yeast and human protein-interaction networks ?* Genome Biol, 7(11) :120.

[140] van Helden J., AndrÃl' B., and Collado-Vides J., Jan 2000. *A web site for the computational analysis of yeast regulatory sequences.* Yeast, 16(2) :177–187.

[141] van Helden J., Jul 2003. *Regulatory sequence analysis tools.* Nucleic Acids Res, 31(13) :3593–3596.

[142] Thomas-Chollier M., Sand O., Turatsinze J.V., Janky R., Defrance M., Vervisch E., Brohee S., and van Helden J., Jul 2008. *RSAT : regulatory sequence analysis tools.* Nucleic Acids Res, 36(Web Server issue) :W119–W127.

[143] Jimenez V. and Marzal. A., 1999. *Computing the K Shortest Paths : a New Algorithm and an Experimental Comparison.* Proc. 3rd Int. Worksh. Algorithm Engineering (WAE 1999), 1668 :15–29.

[144] von Mering C., Jensen L.J., Kuhn M., Chaffron S., Doerks T., KrÃijger B., Snel B., and Bork P., Jan 2007. *STRING 7–recent developments in the integration and prediction of protein interactions.* Nucleic Acids Res, 35(Database issue) :D358–D362.

[145] Brohee S., Faust K., Lima-Mendez G., Vanderstocken G., and van Helden J., 2008. *Network Analysis Tools : from biological networks to clusters and pathways.* Nat Protoc, 3(10) :1616–1629.

[146] De Hertogh B., Hancy F., Goffeau A., and Baret P.V., Feb 2006. *Emergence of species-specific transporters during evolution of the hemiascomycete phylum.* Genetics, 172(2) :771–781.

[147] Ward J.M., 2001. *Identification of novel families of membrane proteins from the model plant Arabidopsis thaliana.* Bioinformatics, 17(6) :560–3.

[148] Schwacke R., Schneider A., van der Graaff E., Fischer K., Catoni E., Desimone M., Frommer W.B., Flugge U.I., and Kunze R., 2003. *ARAMEMNON, a novel database for Arabidopsis integral membrane proteins.* Plant Physiol, 131(1) :16–26.

[149] Axelsen K.B. and Palmgren M.G., 1998. *Evolution of substrate specificities in the P-type ATPase superfamily.* J Mol Evol, 46(1) :84–101.

[150] Christie K.R., Weng S., Balakrishnan R., Costanzo M.C., Dolinski K., Dwight S.S., Engel S.R., Feierbach B., Fisk D.G., Hirschman J.E., Hong E.L., Issel-Tarver L., Nash R., Sethuraman A., Starr B., Theesfeld C.L., Andrada R., Binkley G., Dong Q., Lane C., Schroeder M., Botstein D., and Cherry J.M., Jan 2004. *Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms.* Nucleic Acids Res, 32(Database issue) :D311–D314.

[151] Rand W.M., 1971. *Objective Criteria for the Evaluation of Clustering Methods.* Journal of the American Statistical Association, 66(336) :846–850. ISSN 01621459.

[152] Denoeud L. and Guenoche A., 2006. *Data Science and Classification*, chapter Comparison of Distance Indices Between Partitions, pages 21–28. Springer Berlin Heidelberg.

[153] Guenoche A., October 2005. *Comparing recent methods in graph partitioning.* Electronic Notes in Discrete Mathematics, 22 :83–89.

[154] Rungsarityotin W., Krause R., SchÃűdl A., and Schliep A., 2007. *Identifying protein complexes directly from high-throughput TAP data with Markov random fields.* BMC Bioinformatics, 8 :482.

[155] Wang C., Ding C., Yang Q., and Holbrook S.R., 2007. *Consistent dissection of the protein interaction network by combining global and local metrics.* Genome Biol, 8(12) :R271.

[156] Pasupuleti S., 2008. *Detection of Protein Complexes in Protein Interaction Networks Using n-Clubs.* In E. Marchiori and J.H. Moore, editors, *EvoBIO*, volume 4973 of *Lecture Notes in Computer Science*, pages 153–164. Springer. ISBN 978-3-540-78756-3.

[157] Bernard F. and Andre B., Jul 2001. *Genetic analysis of the signalling pathway activated by external amino acids in Saccharomyces cerevisiae.* Mol Microbiol, 41(2) :489–502.

[158] Ljungdahl P.O., Gimeno C.J., Styles C.A., and Fink G.R., Oct 1992. *SHR3 : a novel component of the secretory pathway specifically required for localization of amino acid permeases in yeast.* Cell, 71(3) :463–478.

[159] Dean N., Zhang Y.B., and Poster J.B., Dec 1997. *The VRG4 gene is required for GDP-mannose transport into the lumen of the Golgi in the yeast, Saccharomyces cerevisiae.* J Biol Chem, 272(50) :31908–31914.

[160] Khademi S., O'Connell J., Remis J., Robles-Colmenares Y., Miercke L.J.W., and Stroud R.M., Sep 2004. *Mechanism of ammonia transport by Amt/MEP/Rh : structure of AmtB at 1.35 A.* Science, 305(5690) :1587–1594.

[161] Shani N. and Valle D., Oct 1996. *A Saccharomyces cerevisiae homolog of the human adrenoleukodystrophy transporter is a heterodimer of two half ATP-binding cassette transporters.* Proc Natl Acad Sci U S A, 93(21) :11901–11906.

[162] Breitkreutz B.J., Stark C., Reguly T., Boucher L., Breitkreutz A., Livstone M., Oughtred R., Lackner D.H., BÃd'hler J., Wood V., Dolinski K., and Tyers M., Jan 2008. *The BioGRID Interaction Database : 2008 update.* Nucleic Acids Res, 36(Database issue) :D637–D640.

[163] Altschul S.F., Gish W., Miller W., Myers E.W., and Lipman D.J., Oct 1990. *Basic local alignment search tool.* J Mol Biol, 215(3) :403–410.

[164] Kanehisa M., Araki M., Goto S., Hattori M., Hirakawa M., Itoh M., Katayama T., Kawashima S., Okuda S., Tokimatsu T., and Yamanishi Y., Jan 2008. *KEGG for linking genomes to life and the environment.* Nucleic Acids Res, 36(Database issue) :D480–D484.

[165] Date S.V. and Marcotte E.M., Sep 2003. *Discovery of uncharacterized cellular systems by genome-wide analysis of functional linkages.* Nat Biotechnol, 21(9) :1055–1062.

[166] Gonzalez-Cabo P., Vazquez-Manrique R.P., GarcÃ■a-Gimeno M.A., Sanz P., and Palau F., Aug 2005. *Frataxin interacts functionally with mitochondrial electron transport chain proteins.* Hum Mol Genet, 14(15) :2091–2098.

[167] Janky R., 2007. *Etude Bioinformatique de l'evolution de la regulation transcriptionnelle chez les bacteries.* Ph.D. thesis, Universite Libre de Bruxelles.

[168] van Helden J., Rios A.F., and Collado-Vides J., Apr 2000. *Discovering regulatory elements in non-coding sequences by analysis of spaced dyads.* Nucleic Acids Res, 28(8) :1808–1818.

[169] Schjerling P. and Holmberg S., Dec 1996. *Comparative amino acid sequence analysis of the C6 zinc cluster family of transcriptional regulators.* Nucleic Acids Res, 24(23) :4599–4607.

[170] Janky R. and van Helden J., 2007. *Discovery of conserved motifs in promoters of orthologous genes in prokaryotes.* Methods Mol Biol, 395 :293–308.

[171] Janky R. and van Helden J., 2008. *Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution.* BMC Bioinformatics, 9 :37.

[172] Simonis N., Wodak S.J., Cohen G.N., and van Helden J., Oct 2004. *Combining pattern discovery and discriminant analysis to predict gene co-regulation.* Bioinformatics, 20(15) :2370–2379.

[173] Harbison C.T., Gordon D.B., Lee T.I., Rinaldi N.J., Macisaac K.D., Danford T.W., Hannett N.M., Tagne J.B., Reynolds D.B., Yoo J., Jennings E.G., Zeitlinger J., Pokholok D.K., Kellis M., Rolfe P.A., Takusagawa K.T., Lander E.S., Gifford D.K., Fraenkel E., and Young R.A., Sep 2004. *Transcriptional regulatory code of a eukaryotic genome.* Nature, 431(7004) :99–104.

[174] Cooper T.G., Aug 2002. *Transmitting the signal of excess nitrogen in Saccharomyces cerevisiae from the Tor proteins to the GATA factors : connecting the dots.* FEMS Microbiol Rev, 26(3) :223–238.

[175] Godard P., Urrestarazu A., Vissers S., Kontos K., Bontempi G., van Helden J., and AndrÃΓ B., Apr 2007. *Effect of 21 different nitrogen sources on global gene expression in the yeast Saccharomyces cerevisiae.* Mol Cell Biol, 27(8) :3065–3086.

[176] Yamaguchi-Iwai Y., Stearman R., Dancis A., and Klausner R.D., Jul 1996. *Iron-regulated DNA binding by the AFT1 protein controls the iron regulon in yeast.* EMBO J, 15(13) :3377–3384.

[177] Teixeira M.C., Monteiro P., Jain P., Tenreiro S., Fernandes A.R., Mira N.P., Alenquer M., Freitas A.T., Oliveira A.L., and SÃą-Correia I., Jan 2006. *The YEASTRACT database : a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae.* Nucleic Acids Res, 34(Database issue) :D446–D451.

[178] Kaur J. and Bachhawat A.K., Jun 2007. *Yct1p, a novel, high-affinity, cysteine-specific transporter from the yeast Saccharomyces cerevisiae.* Genetics, 176(2) :877–890.

[179] Rutherford J.C., Jaron S., and Winge D.R., Jul 2003. *Aft1p and Aft2p mediate iron-responsive gene expression in yeast through related promoter elements.* J Biol Chem, 278(30) :27636–27643.

[180] Huh W.K., Falvo J.V., Gerke L.C., Carroll A.S., Howson R.W., Weissman J.S., and O'Shea E.K., Oct 2003. *Global analysis of protein localization in budding yeast.* Nature, 425(6959) :686–691.

[181] Overkamp K.M., KÃűtter P., van der Hoek R., Schoondermark-Stolk S., Luttik M.A.H., van Dijken J.P., and Pronk J.T., Apr 2002. *Functional analysis of structural genes for NAD(+)-dependent formate dehydrogenase in Saccharomyces cerevisiae.* Yeast, 19(6) :509–520.

[182] Makuc J., Paiva S., Schauen M., KrÃď mer R., AndrÃľ B., Casal M., LeÃčo C., and Boles E., Sep 2001. *The putative monocarboxylate permeases of the yeast Saccharomyces cerevisiae do not transport monocarboxylic acids across the plasma membrane.* Yeast, 18(12) :1131–1143.

[183] Wall D.P. and Deluca T., 2007. *Ortholog detection using the reciprocal smallest distance algorithm.* Methods Mol Biol, 396 :95–110.

[184] Stumpf M.P.H., Thorne T., de Silva E., Stewart R., An H.J., Lappe M., and Wiuf C., May 2008. *Estimating the size of the human interactome.* Proc Natl Acad Sci U S A, 105(19) :6959–6964.

[185] Archakov A.I., Govorun V.M., Dubanov A.V., Ivanov Y.D., Veselovsky A.V., Lewi P., and Janssen P., Apr 2003. *Protein-protein interactions as a target for drugs in proteomics.* Proteomics, 3(4) :380–391.

[186] Helms V., Dec 2002. *Attraction within the membrane. Forces behind transmembrane protein folding and supramolecular complex assembly.* EMBO Rep, 3(12) :1133–1138.