Détection de communautés dans les grands réseaux : Application aux réseaux d'interactions de gènes

Marwa Ben M'Barek

PSL
UNIVERSITÉ PARIS

جامعة تونس المنار
UNIVERSITÉ DE TUNIS EL MANAR

ED-EST
MISTM
École Doctorale de Mathématiques, Informatique, Science et Technologie de la Matière

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Paris-Dauphine
Dans le cadre d'une cotutelle avec l'Université de Tunis El Manar

Détection de communautés dans les grands réseaux :

Application aux réseaux d'interactions de gènes

Community detection in complex networks : Application to gene interaction network

Soutenue par
Marwa BEN M'BAREK
Le 14/10/2022

Ecole doctorale n° ED 543
Ecole doctorale SDOSE

Spécialité
Informatique

Composition du jury :

Jean-François, PRADAT-PEYRE Professeur, Université Paris Nanterre	*Président*
Pierre, COLLET Professeur, Université de Strasbourg	*Rapporteur*
Lotfi, BEN ROMDHANE Professeur, Université de Sousse	*Rapporteur*
Hajer, BAAZAOUI Professeure, Université de la Manouba	*Examinatrice*
Marta, RUKOZ Professeure, Université Paris Nanterre	*Directrice de thèse*
Amel, BORGI Professeure, Université Tunis El Manar	*Directrice de thèse*
Sana, BEN HMIDA Maîtresse de conférences, Université Paris Nanterre	*Co-encadrante*

Dauphine | PSL
UNIVERSITÉ PARIS

Dédicaces

Je dédie ce travail en témoignage de ma profonde reconnaissance à toute personne qui m'a aidée et m'a encouragée.

Je dédie ce travail plus particulièrement à ma famille pour leurs encouragements tout au long de ce projet. Que ce modeste travail, soit une petite compensation et une reconnaissance envers ce qu'ils ont fait d'incroyable pour moi.

Je dédie ce modeste travail à mes chères amies à qui me souhaitent le succès, pour l'amitié qui nous a toujours unis.

Ă tous ceux qui m'aiment et ceux que j'aime.

Remerciements

Selon la loi de Murphy, tout prend plus de temps que prévu. Et selon John Lennon, tout va être bien à la fin. Par conséquent, les souhaits peuvent être un peu retardés, mais ils finiront par venir. C'est la fin alors, et c'est le moment de dire merci, à chaque personne étant ici, autour de moi ou avec moi. C'est avec plaisir et reconnaissance que je leur réserve ces lignes.

Je souhaite remercier mes directrices de thèse qui m'ont guidé tout au long de ces années. Je remercie Mme Amel BORGI pour ces remarques pertinentes, sa rigueur scientifique et son suivi exemplaire.
Je remercie vivement Mme Marta RUKOZ, qui a accepté de codiriger cette thèse, et m'a poussé toujours vers la perfection à travers nos longues discussions.

J'adresse un remerciement particulier à Mme Sana BEN HAMIDA pour son co-encadrement, sa disponibilité toutes ces années et ses nombreuses idées et propositions qui ont imprégné ce travail.

Mes sincères remerciements s'adressent aussi à notre expert biologique, Mr. Walid BEDHIAFI, docteur en bio-informatique au laboratoire de Génétique, Immunologie et Pathologies Humaines à FST et membre au laboratoire Immunologie - Immunopathologie – Immunothérapeutique équipe Immunologie Intégrative Université Paris 6, de m'avoir fait bénéficier de son expérience et de son aide.

Je remercie Mr. Pierre COLLET et Mr. Lotfi BEN ROMDHANE, qui ont accepté d'être rapporteurs de ma thèse.

Je remercie également Mme Hajer BAAZAOUI et Mr. Jean-francois Pradat-peyre d'être examinateurs de ce travail. Vous m'avez fait honneur en faisant partie de mon jury.

Je remercie aussi les membres des laboratoires LAMSADE et LIPAH dans lesquels ce travail a vu le jour. Malgré le peu de temps que j'ai passé dans leurs locaux respectifs, j'ai pu profiter pendant mes courts séjours de la bonne ambiance qui y règne ainsi que de l'assistance et la disponibilité du personnel.

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance à ma mère, à qui je dois la réussite, pour l'éducation qu'elle m'a prodigué ; avec tous les moyens et au prix de tous les sacrifices qu'elle a consentis à mon égard. Leurs prières et leurs Bénédictions m'ont été d'un grand secours tout au long de ma vie.

Je remercie également mes sœurs Kaouther, Amel et Leila et mon frère Saber. Je ne saurais exprimer ma profonde reconnaissance pour vos soutiens continus dont vous avez toujours fait preuve. Vous m'avez toujours encouragée et incitée à faire de mon mieux.

Je remercie chaleureusement mon mari, Ahmed, pour avoir su trouver les mots justes dans les moments les plus difficiles, pour l'encouragement, et pour le soutien moral et matériel. Cher mari, j'aimerais bien que tu trouves dans ce travail l'expression de mes sentiments de reconnaissance les plus sincères.

Résumé

L'une des caractéristiques les plus pertinentes des réseaux est l'existence de zones plus densément connectées que d'autres. Ces zones sont appelées communautés. Une communauté est alors décrite par un ensemble de nœuds intensément liés entre eux, mais faiblement liés au reste du réseau. La détection de communautés est l'un des thèmes de recherche les plus prolifiques en analyse de réseaux. Dans le cadre de cette thèse de doctorat, nous nous intéressons à la détection des communautés de gènes dans les réseaux d'interactions de protéines. Ces communautés correspondent à des ensembles de gènes qui collaborent à une même fonction cellulaire. Notre objectif consiste à déterminer un groupe ou une communauté de gènes à partir des sources d'annotation en se basant sur l'apprentissage automatique. Pour réaliser ce travail, nous combinons 3 niveaux d'informations : i) niveau sémantique : information contenue dans gene Ontology, ii) niveau fonctionnel : information contenue dans des bases de données publiques qui décrivent les interactions des gènes et iii) niveau réseau : informations contenues dans les bases de voies biologiques. Ce travail est multidisciplinaire, à l'intersection de domaine de l'informatique et de la biologie et il comporte 4 volets.
Le premier volet se concentre sur l'extraction des données biologiques utiles pour notre projet et sur l'étude de la similarité sémantique entre des groupes de gènes. Cette dernière sera l'une de caractéristique d'une communauté de gènes. Nous avons proposé, dans le deuxième volet, une approche pour la détection des communautés de gènes basée sur les algorithmes génétiques. Cette approche nommée *GA-PPI-Net* permet de construire et de détecter des communautés de gènes de tailles variables. *GA-PPI-Net* permet de maximiser une mesure communautaire qui combine à la fois des informations topologiques entre les gènes et des informations sémantiques. Par ailleurs, nous avons introduit une solution spécifique pour représenter une communauté (=solution) de taille variable et un opérateur de mutation optimisée. Dans le troisième volet, nous nous focalisons sur l'extension et l'amélioration de *GA-PPI-Net*. La première extension sert à proposer un nouvel opérateur de mutation adaptatif. Cette amélioration a pour objectif d'optimiser l'exploration de l'espace de recherche afin d'améliorer les qualités des communautés obtenues. La deuxième amélioration vise à rendre notre approche générique, *Generic GA-PPI-Net*, pour construire des communautés de gènes qui sont sémantiquement similaires et/ou sont en interaction. Dans le dernier volet, nous avons étendu ce travail afin de mettre à l'échelle *Generic GA-PPI-Net* en utilisant le paradigme d'apprentissage actif. Cette extension nous a permis d'utiliser des sources de données volumineuses (la totalité de réseau d'interaction de protéine chez l'être humain) pour construire des communautés évolutives. Elle est basée sur une approche d'échantillonnage adaptative permettant de définir la partie du graphe à explorer par les opérateurs génétiques pendant l'évolution.

Mots clé : détection de communautés, Gene Ontology, réseau PPI, interaction de gènes, similarité sémantique, algorithmes génétiques, apprentissage actif.

Abstract

The modern science of networks has brought significant advances to our understanding of complex networks. One of the most relevant features of networks system's representation is the existence of areas more densely connected than other areas. These areas are usually called as communities. In our work, we are interested in the communities' detection in protein-protein interaction networks (PPI). These communities give us an idea about the perception of the network's structure. One of the goals in biology is to determine how genes or proteins encode function in the cell. This work is multidisciplinary, as it brings the field of biology and computer science in the broad sense. Thus, our objectif is to find communities of genes having a biological sense (that participate in the same biological processes or that perform together specific biological functions) from gene annotation sources. To make this task, we have combined three levels of information : i) Semantic level : information contained in biological ontologies such as Gene Ontology (GO) and information obtained by the use of a similarity measure such as GO-based similarity of gene sets (GS2). It assesses the semantic similarity between genes, ii) Functional level : information contained in public databases describing the interactions of genes iii) Networks level : information contained in pathway databases. Our work has four parts.
The first part focuses on the extraction of biological data used in our project. Thus, we study the semantic similarity between groups of genes that are annotated by terms of biological ontology. It is one of the characteristics of a gene community. The second part present the proposed approach *GA-PPI-Net* for the detection of gene communities. It is a Genetic Algorithm based approach to detect communities having different sizes from PPI networks. For this purpose, we use a fitness function based on a similarity measure and the interaction value between proteins or genes. Moreover, a specific solution for representing a community and a specific mutation operator are introduced. The third part presents two extensions of *GA-PPI-Net*. The first one proposes a specific adaptive mutation operator. The second aims to make GA-PPI-Net generic by allowing finding different sizes of communities based on the interaction and/or similarity criterion. This approach called *Generic GA-PPI-Net*. Finally, we propose to scale Generic GA-PPI-Net using the active learning paradigm. This approach allowed us to use a large data sets (the whole humain PPI) to build evolutionary communities. It is based on an adaptive sampling approach to define the part of the graph to be explored by the genetic operators during the GA evolution.

Keywords : Community detection, Genetic Algorithm, Biological ontology, PPI networks, semantic similarity, gene interaction, actif learning.

Table des matières

Liste des abréviations

A

AG : Algorithme Génétique

AE : Algorithme Evolutionnaire

ADN : Acide DésoxyriboNucléique

ARN : Acide RiboNucléique

API : Application Programming Interface / Interface de programmation

B

BP : Biological Process

C

CC : Composants Cellulaires

D

DAVID : Database for Annotation, Visualization and Integrated Discovery

G

GO : Gene Ontology

GOA : Gene Ontology Annotation

GS2 : GO-based similarity of gene sets

I

IC : Information content

K

KEGG : Kyoto Encyclopedia of Genes and Genomes

L

LGIPH : Laboratoire de Génétique, Immunologie et Pathologies Humaines

M

MF : Molecular Function

MICA : Ancêtre commun le plus informatif

MIPS : Munich Information center for Protein Sequences

N

NCBI : Centre National de l'information biotechnologique

P

PPI : Protein-Protein Interaction

Table des figures

Liste des tableaux

Introduction générale

Sommaire

Contexte et problématique

Depuis une vingtaine d'années, nous assistons à une explosion des activités autour de la représentation, de la modélisation et de l'analyse de grands volumes de données d'interactions. Ces données ne touchent pas seulement un domaine particulier, mais s'étendent à tout système pouvant être décrit par un réseau au sens mathématique (graphe). Par ailleurs, l'utilisation généralisée des technologies de l'information et de la communication ainsi que la progression des moyens de recueil et de stockage des données rendent la taille de ces réseaux de plus en plus croissante. Parallèlement à cela, les performances des ordinateurs actuels permettent la manipulation de volumes d'information de taille de plus en plus importante. La perception actuelle de la notion de réseau a permis de réaliser des progrès significatifs pour la compréhension des systèmes complexes. L'une des caractéristiques les plus pertinentes de ces réseaux est l'existence de groupes de nœuds plus fortement connectés que d'autres. Ces groupes sont habituellement des sous-réseaux nommés des communautés. La détection automatique de telles communautés (ou groupes) offre un éclairage intéressant sur la structure du réseau. Il devient possible de voir de grandes tendances et d'obtenir une vision macroscopique des systèmes complexes. Ces communautés permettent de ce fait de donner un point de vue sur la structure des réseaux. Elles peuvent avoir des interprétations différentes suivant le type de réseau considéré. Dans les réseaux d'interactions de protéines en biologie, les communautés correspondent à des ensembles de gènes ou de protéines qui collaborent à une même fonction cellulaire.

Les principaux problèmes liés à la détection de communautés sont la définition de ces groupes, leur détection en pratique qui est un problème algorithmique et comment vérifier que les résultats sont effectivement pertinents.

Les propriétés topologiques des réseaux d'interaction protéine-protéine sont les clés de la compréhension des maladies et devraient permettre aux biologistes de trouver des objectifs thérapeutiques précis. Les travaux de cette thèse s'inscrivent dans ce contexte interdisciplinaire et se concentrent sur la question algorithmique de détection de communautés de gènes.

Pour notre contexte, il n'existe pas un graphe qui définisse et structure le réseau biologique utilisé dans notre travail, *le réseau d'interaction protéine-protéine ou gène-gène*. C'est donc à nous de déterminer cette structure en utilisant certaines bases de connaissances. Il existe différentes bases

de connaissances et ontologies qui répertorient et structurent les informations relatives au domaine biologique. Elles constituent une ressource essentielle pour l'annotation des connaissances. Les gènes sont annotés par des ensembles de termes organisés au sein d'une structure sémantique particulière, une ontologie (Bard and Rhee, 2004). Deux types de sources de données sont utilisées dans ce travail : les ontologies qui annotent les gènes et les bases de données qui décrivent les interactions gène-gène et les voies biologiques. Pour structurer et mettre en évidence le réseau biologique utilisé dans notre travail, nous avons combiné les trois niveaux d'informations suivants :

— Niveau sémantique : informations contenues dans les ontologies biologiques (par exemple Gene Ontology) (Ashburner et al., 2000) et la similarité sémantique entre des paires de gènes (Ruths et al., 2009),

— Niveau fonctionnel : informations contenues dans des bases de données publiques qui décrivent les interactions entre les gènes (Snel et al., 2000),

— Niveau réseau : informations contenues dans les bases de voies biologiques (Kanehisa and Goto, 2000).

Les analyses réalisées par les auteurs dans (Guo et al., 2006) ont montré que les gènes qui appartiennent à une même communauté de la base de données de voies biologiques KEGG Pathway sont des gènes qui sont sémantiquement similaires et qui sont en interaction. Partant de cette hypothèse et pour assurer les besoins des biologistes, nous considérons que les gènes qui forment une communauté sont sémantiquement similaires et sont en interaction.

Détecter des partitions en communauté d'un réseau complexe est un problème qui se rencontre dans différents domaines, par exemple dans les réseaux sociaux. Il existe plusieurs approches parmi lesquelles celles basées sur les algorithmes évolutionnaires (AEs) (Pizzuti, 2008). Même si la problématique de détection de communautés dans un réseau est la même, les solutions à apporter dépendent fortement du type de réseau, des liens de différentes natures entre les nœuds et de la sémantique du domaine considéré.

Jusqu'à présent, beaucoup de travaux ont été proposés pour le problème de détection de communautés. Dans le cas des réseaux statiques sans chevauchement entre communautés, des algorithmes de clustering, clustering hiérarchique et d'optimisation, etc., ont été proposés. L'inconvénient des algorithmes de clustering réside dans le besoin de fixer le nombre de clusters et l'utilisation d'un

seul critère à optimiser comme la densité ou la modularité. Quant aux algorithmes de clustering hiérarchique, ils ont une grande complexité quant à la détermination du niveau de la coupure du dendrogramme. Le problème de complexité apparaît aussi dans les algorithmes d'optimisation comme l'algorithme de Girvan Newman (Newman, 2006). Pour cela, les algorithmes basés sur les métaheuristiques sont proposés une solution intéressante, comme les algorithmes génétiques (AG), colonies de fourmis, etc. Dans notre cas, nous avons choisi les algorithmes génétiques. Ces algorithmes commencent par une population de solutions (individus) générée de manière aléatoire sur laquelle sera appliquée la boucle évolutionnaire. Cette dernière consiste à évaluer les différentes solutions selon une fonction de mesure de qualité (fitness) puis à appliquer les opérateurs génétiques sur les individus sélectionnés jusqu'à la satisfaction d'une condition d'arrêt pouvant être une solution satisfaisante ou un nombre de générations cible. Ces algorithmes ont les avantages suivants :

— La possibilité de combiner plusieurs critères (topologiques et sémantiques) à optimiser.

— Pas de nécessité de connaître à l'avance le nombre de communautés à détecter.

— La possibilité d'être adapté aux données massives.

Dans cette thèse, nous étudions donc les algorithmes génétiques pour profiter de leur potentiel dans le contexte de détection de communautés. Nous proposons une approche basée sur les algorithmes génétiques afin de détecter les communautés de gènes dans un réseau d'interaction protéine-protéine humaine. Ces algorithmes ont été très largement utilisés et leur succès est dû en partie à leur grande capacité exploratoire et la facilité d'implémentation. Malgré la séduisante facilité du processus évolutionnaire, produire un algorithme évolutionnaire efficace est une tâche difficile. Les processus évolutionnaires sont très sensibles aux choix algorithmiques et paramétriques et notamment aux choix des représentations. L'expérience a montré que les succès sont fondés essentiellement sur une compréhension fine des mécanismes évolutionnaires et surtout sur une très bonne connaissance du problème à traiter. Une forte spécialisation de l'algorithme évolutionnaire est donc souhaitée pour aboutir à des résultats probants, cependant cette spécialisation va à l'encontre de la facilité d'implantation.

Objectifs et contributions

Les travaux de cette thèse s'inscrivent dans un contexte interdisciplinaire. Leur objectif est de proposer des réponses algorithmiques pour la détection de communautés dans les grands graphes. Dans ce travail, le domaine d'étude est celui des réseaux biologiques, spécifiquement les réseaux d'interaction protéine-protéine (PPI). Les grandes parties de notre travail sont articulés en quatre volets.
Dans un premier volet, nous nous intéressons à l'extraction des données biologiques utiles pour notre projet. Ainsi, nous nous intéressons à l'étude de la similarité sémantique entre des groupes de gènes qui sont annotés par des termes de l'ontologie Gene Ontology. Cette dernière est considérée comme l'une des caractéristiques d'une communauté de gène.
Le second volet concerne la détection des communautés de gènes de taille variable chez l'être humain en se basant sur les algorithmes génétiques (AGs). L'idée est d'adapter les algorithmes génétiques à notre problème pour construire et détecter des communautés de gènes qui sont sémantiquement similaires et sont en interaction de manière évolutive. Nous commençons alors par étudier les différentes approches de détection de communautés. Un tour d'horizon des publications scientifiques dans le domaine de la détection de communautés révèle l'existence de deux familles d'approches : les approches analytiques et les approches évolutionnaires :

— les approches analytiques sont basées sur le partitionnement de graphes. Leur principe consiste à regrouper les nœuds d'un graphe en un nombre généralement prédéterminé de sous-groupes homogènes en minimisant le nombre de liens entre les différents groupes, de telle sorte que les nœuds d'une même communauté se ressemblent le plus possible alors que les nœuds de communautés différentes sont les plus différents possible. Ces approches exigent que l'utilisateur spécifie à l'avance le nombre de communautés à identifier. Or, en pratique, il est rarement possible pour l'utilisateur de fournir le nombre exact de communautés. C'est pour remédier à ce problème que les approches évolutionnaires ont été proposées.

— les approches évolutionnaires sont les approches basées sur les AGs pour détecter les communautés dans les réseaux complexes. Les différentes approches existantes de détection de communautés basées sur les AGs sont basées sur l'optimisation d'une mesure topologique comme la modularité.

En s'inspirant des approches évolutionnaires existantes, une approche **constructive et évolutive**

originale nommée *GA-PPI-Net* est proposée. Le principal intérêt de cette approche est de combiner trois niveaux d'information : i) le niveau sémantique, ii) le niveau fonctionnel et iii) le niveau réseau. Cette approche permet d'optimiser une mesure topologique (= l'interaction entre deux gènes) et une mesure sémantique (= la similarité sémantique entre deux gènes). Elle est basée sur une représentation de solution spécifique et une fonction de fitness basée sur une mesure de similarité et une valeur d'interaction entre deux gènes. Cette mesure permet de construire des communautés de protéines ayant une forte similarité sémantique et une forte interaction. Nous avons introduit une solution spécifique pour représenter une communauté (= solution) de taille variable. Ainsi, nous avons proposé un opérateur de mutation optimisé (OCM1) pour une meilleure exploration du graphe. Le rôle de cet opérateur est de renforcer les scores d'interaction et de similarité sémantique au sein de la solution. Pour cela, nous avons proposé une fonction score appliquée à un gène, pour estimer sa qualité au sein de la communauté à laquelle il appartient. Ce score nous aidera à détecter le gène ayant le meilleur score dans une communauté (BestGene). L'idée est d'introduire dans la communauté un nouveau gène en interaction avec BestGene, ce nouveau gène remplacera le gène ayant le plus mauvais score.

Dans le troisième volet de notre travail, nous nous intéressons à l'extension et l'amélioration de *GA-PPI-Net*. Dans la première extension (M'barek et al., 2019), nous proposons un nouvel opérateur de mutation adaptatif. Cette amélioration a pour objectif d'optimiser l'exploration de l'espace de recherche afin d'améliorer les qualités des communautés pendant l'évolution en renforçant l'interaction et la similarité entre les gènes. La deuxième amélioration vise à rendre notre approche ***générique*** pour construire des communautés de gènes qui sont sémantiquement similaires et/ou sont en interaction. Le dernier volet sert à mettre à l'échelle *GA-PPI-Net* pour la détection de communautés dans les grands réseaux (réseau d'interaction de protéine chez l'être humain). L'objectif est de réduire la complexité due essentiellement aux opérateurs génétiques qui sont amenés à explorer l'ensemble du graphe pour définir l'évolution à appliquer aux communautés dans la population. L'idée est de limiter l'espace de recherche pour ces opérateurs en sélectionnant la partie du réseau à explorer selon l'évolution des meilleures communautés. Cet objectif est mis en place grâce au paradigme d'apprentissage actif. Ce dernier est introduit dans *GA-PPI-Net* par une méthode échantillonnage dynamique et adaptative du graphe en sélectionnant les nœuds respectant des contraintes de voisinage et des seuils de similarité et d'interaction.

Les travaux menés dans le cadre de cette thèse ont abouti aux contributions suivantes :

1. **Proposition d'une approche *constructive et évolutive* pour la détection de communautés de tailles variables** : En s'inspirant de notre approche proposée dans (Ben M'barek et al., 2018) qui permet de détecter des communautés de gènes de taille fixe, nous avons proposé une solution pour détecter des communautés de gènes de tailles variables. L'originalité de l'approche réside essentiellement dans l'introduction d'une mesure communautaire qui combine à la fois des informations topologiques entre les gènes et des informations sémantiques. La méthode proposée est basée sur la maximisation de cette mesure communautaire en explorant sélectivement l'espace de recherche. Par ailleurs, nous avons introduit une solution spécifique pour représenter une communauté (= solution) de taille variable et un opérateur de mutation optimisée ("OCM1") pour une meilleure exploration du graphe. Les communautés denses existant dans le réseau sont obtenues à la fin de l'évolution en explorant sélectivement l'espace de recherche, sans avoir besoin de connaître à l'avance la taille de la communauté.
   Une étude expérimentale a été menée sur des données extraites à partir de bases de données biologiques. L'approche proposée *GA-PPI-Net* a permis de détecter des communautés qui existent partiellement ou entièrement dans des bases de voies biologiques. Ces résultats très satisfaisants ont été validés par un expert en biologie (Ben M'barek et al., 2019). Pour positionner notre travail par rapport à la littérature, nous avons évalué la performance de *GA-PPI-Net* par rapport à des méthodes analytiques de détection de communautés de gènes disponibles dans la littérature (Ben M'barek et al., 2021). Nous avons proposé pour cela de mettre en place un protocole de validation et un prototype expérimental. Les méthodes analytiques sélectionnées sont le clustering MCL (Markov Clustering), RNSC (Restricted Neighborhood Search Clustering) et ClusterOne. Toutes les approches testées ont pu reconstruire efficacement des communautés existantes dans de véritables bases de données de voies biologiques. Toutefois, le taux de recouvrement minimum est meilleur avec GA-PPI-Net, qu'avec les trois approches de clustering, et les valeurs médianes de la similarité moyenne et l'interaction moyenne sont également meilleures avec GA-PPI-Net. De plus, nous avons évalué et comparé notre approche par rapport à la méthode de clustering évolutionnaire EGCPI (He and Chan, 2016). Les deux approches conduisent à des résultats similaires, cela pourrait s'expliquer d'une part par leur recours commun aux AGs et d'autre part par la prise en

compte par chacune de ces approches de l'aspect topologique.

2. **Proposition d'une approche générique et adaptative : *Generic GA-PPI-Net*** : nous avons proposé deux améliorations de *GA-PPI-Net*. La première amélioration (Ben M'barek et al., 2019) a été proposée afin de la rendre générique et adaptative. Nous avons introduit pour cela trois nouvelles composantes spécifiques à l'AG : i) une nouvelle solution adéquate pour représenter une communauté de taille dynamique, ii) une fonction de fitness générique basée sur une mesure de similarité et une valeur d'interaction entre les gènes en fonction des valeurs de seuils, et iii) un opérateur de croisement heuristique pour renforcer les liens dans les communautés. L'AG proposé a été paramétré en fonction de l'importance affectée à chaque critère de mesure (mesure sémantique et mesure d'interaction) (Ben M'Barek. et al., 2020). Cette généralisation nous a permis de détecter trois types de communautés : i) groupes de gènes basés sur l'interaction, ii) groupes de gènes basés sur la similarité et iii) communautés de gènes similaires et qui sont en interaction. De plus, nous avons amélioré *GA-PPI-Net* en proposant un nouvel opérateur de mutation adaptatif ("OCM2") (M'barek et al., 2019). Cet opérateur favorise l'exploration de l'espace de recherche et la détection des communautés de meilleures qualités. Des études expérimentales comparatives ont été effectuées en respectant le même prototype expérimental que dans la première proposition.

3. **Mise à l'échelle de *Generic GA-PPI-Net* en utilisant le paradigme d'apprentissage actif** *Active GA-PPI-Net* : nous proposons la mise à l'échelle de GA-PPI-Net pour la détection de communautés dans des grands réseaux (la totalité du réseau d'interaction de protéines chez l'être humain "Homo Sapiens"). L'objectif est d'explorer la totalité de réseau d'interaction de protéine chez l'être humain pour construire des communautés évolutives en se basant sur une approche d'échantillonnage adaptative. Cette dernière permet de définir la partie du graphe à explorer par les opérateurs génétiques pendant l'évolution selon le voisinage des gènes de la meilleure solution en cours et les seuils d'interaction et de similarité.
   L'efficacité des solutions apportées a été démontrée à travers une étude expérimentale avancée.

Organisation de la thèse

La suite de ce document est structuré en cinq chapitres et une conclusion.
Le chapitre 1 intitulé *Problème biologique et prétraitement des données* introduit quelques notions élémentaires en biologie utilisées dans notre travail. Nous décrivons aussi les ressources de données biologiques utiles pour notre projet. Nous avons proposé des solutions techniques pour leur acquisition, leur représentation et leur intégration. Ce chapitre présente également un aperçu général sur les méthodes de calcul de similarité sémantiques, en particulier la méthode de similarité retenue pour le reste du travail.

Le chapitre 2 intitulé *Détection de communautés : un état de lieux* donne un aperçu général sur le domaine de la détection de communautés dans les réseaux. Il rapporte en premier lieu des généralités sur la présentation des réseaux complexes et leur modélisation par des graphes, ainsi que certains concepts et définitions utiles de la théorie des graphes, notamment une définition formelle de la notion d'une communauté de gènes. La deuxième partie de ce chapitre est un état de l'art des approches analytiques et évolutionnaires pour la détection des communautés, avec un aperçu particulier sur les algorithmes génétiques.

Le chapitre 3 intitulé *Approche proposée pour la détection de communautés de gènes : GA-PPI-Net* est consacré à la description de l'approche *GA-PPI-Net* que nous proposons pour la détection de communautés de gènes de longueur variable dans le réseau PPI. La performance de *GA-PPI-Net* est validée par deux études expérimentales : une étude expérimentale pour la détection de communautés existantes et une étude comparative avec des approches analytiques et une approche de clustering évolutionnaire.

Le chapitre 4 *GA-PPI-Net : Extensions et Améliorations* liste les différentes extensions et améliorations apportées à *GA-PPI-Net*. Une étude expérimentale associée à chaque extension permet de valider expérimentalement les communautés obtenues pour chacune d'elles.

Le chapitre 5 *La mise à l'échelle de GA-PPI-Net : Active GA-PPI-N* est dédié à la mise à l'échelle de GA-PPI-Net pour la détection de communautés dans des grands réseaux. Il introduit la méthode "Adaptive Graph Sampling" utilisée par *Active GA-PPI-Net* pour implémenter l'apprentissage actif

à travers un échantillonnage dynamique et adaptatif du graphe PPI. Il détaille ensuite une étude expérimentale en plusieurs phases pour tester et comparer la validité de cette approche.

Enfin, la conclusion générale donne une synthèse des travaux et contributions effectués durant cette thèse ainsi que l'ouverture de nouvelles voies de recherche à investiguer.

Liste des publications

[1] *Ben M'barek, M., Borgi, A., Bedhiafi, W., and Ben Hmida, S. (2018). Genetic Algorithm for Community Detection in Biological Networks. Proceedings Computer Science, 126 :195–204. Knowledge-Based and Intelligent Information Engineering Systems : Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.*

[2] *Ben M'barek, M., Borgi, A., Ben Hmida, S., and Rukoz, M. (2019). GA-PPI-Net : A genetic algorithm for community detection in protein-protein interaction networks. In International Conference on Software Technologies, pages 133–155. Springer.*

[3] *Ben M'barek, M., Borgi, A., Ben Hmida, S., and Rukoz, M. (2019). Genetic algorithm to detect different sizes' communities from protein-protein interaction networks. In Proceedings of the 14th International Conference on Software Technologies - Volume 1 : ICSOFT,, pages 359– 370. INSTICC, SciTePress.*

[4] *Ben M'Barek., M., Borgi., A., Ben Hmida., S., and Rukoz., M. (2020). Generic GA-PPI-Net : Generic evolutionary algorithm to detect semantic and topological biological communities. In Proceedings of the 15th International Conference on Software Technologies - ICSOFT,, pages 295–306. INSTICC, SciTePress.*

[5] *Ben M'barek, M. B., Hmida, S. B., Borgi, A., and Rukoz, M. (2021). GA-PPI-Net approach vs analytical approaches for community detection in PPI networks. Procedia Computer Science, pages 903–912.*

Chapitre 1

Problème biologique et prétraitement des données

Sommaire

1.1 Introduction

L'objectif de notre travail est de déterminer des communautés de gènes à partir des sources d'annotation en se basant sur les algorithmes évolutionnaires. Pour réaliser ce travail, nous avons combiné trois niveaux d'informations :

— Niveau sémantique : informations contenues dans les ontologies biologiques et la similarité sémantique entre des paires de gènes ;

— Niveau fonctionnel : informations contenues dans des bases de données d'interaction publiques ;

— Niveau réseau : informations contenues dans les bases de voies biologiques.

Dans ce chapitre, nous commençons tout d'abord par définir des notions liées au domaine de notre étude, celui de la biologie. Les gènes qui y interviennent sont annotés par des ensembles de termes organisés au sein d'une structure sémantique particulière appelée *ontologie*. Des méthodes propres aux ontologies ont été développées par de nombreuses équipes afin de comparer ces ensembles d'annotations. Nous présentons aussi différentes approches de mesures de similarité sémantique et plus particulièrement celles qui sont basées sur Gene Ontology.
Nous répertorions donc ici les bases de données utilisées pour extraire des données utiles pour notre travail, nous décrivons les propriétés des ontologies en général et de Gene Ontology en particulier. Enfin, nous présentons les données extraites et utilisées dans notre travail.

1.2 Notions élémentaires de biologie

Au niveau de cette section, nous définissons des notions liées au domaine de la biologie.

1.2.1 ADN (Acide DésoxyriboNucléique)

L'ADN est le support de l'information constituant le programme de la cellule. Il est composé d'une succession de nucléotides. Il existe quatre nucléotides différents : l'adénosine (A), la cytidine (C), la guanosine (G) et la thymidine (T), dont l'ordre d'enchaînement est très précis et correspond à l'information génétique. Ces derniers sont complémentaires, c'est-à-dire qu'ils peuvent se lier

les uns aux autres grâce à leur information spatiale (A s'associe avec T et G avec C) (Briche, 2009).

1.2.2 Gène

Le vivant est entièrement défini par son génome (exemple ADN), en tout cas en ce qui concerne le fonctionnement de ses cellules et les processus biologiques tels que la respiration ou la circulation sanguine. Ce génome correspond à une suite de gènes qui vont définir le comportement macroscopique et microscopique de l'organisme. Les gènes sont une suite de nucléotides présents sur l'ADN qui vont définir le plan de fonctionnement de l'organisme. Pour pouvoir comprendre ce plan de fonctionnement, l'organisme utilise deux mécanismes : la transcription qui va créer un négatif de l'ADN afin de permettre ensuite la traduction des nucléotides en acides aminés qui, liés entre eux, vont finalement former les gènes. Ces gènes sont des molécules utilisables par l'organisme (Briche, 2009).
En résumé, un gène est un ensemble de nucléotides qui vont, grâce à la transcription et à la traduction, être transcrits en protéines fonctionnelles pour l'organisme.

1.2.3 Produit génique ou produit de gènes

Un produit génique est une biomolécule résultant de l'expression d'un gène, par exemple une protéine. La mesure de la quantité de produits géniques est parfois utilisée pour évaluer l'activité d'un gène (McGraw-Hill Concise Dictionary of Modern Medicine, 2002).

1.2.4 Interaction protéine–protéine

Une interaction protéine–protéine apparaît lorsque deux ou plusieurs protéines se lient entre elles, le plus souvent pour mener à bien leur fonction biologique. Beaucoup des molécules les plus importantes qui agissent dans la cellule, comme la réplication de l'ADN, sont mises en œuvre par de grosses machines moléculaires qui sont constituées d'un grand nombre de protéines organisées grâce aux interactions protéine-protéine. Les interactions entre protéines ont été étudiées du point de vue de la biochimie, de la chimie quantique, de la dynamique moléculaire, de la chimie biologique, de la transduction de signal et autre théorie des graphes métabolique et génétique/épi-

génétique. Les interactions protéine-protéine sont au cœur de tout le système interactomique de la cellule vivante.

Les interactions entre protéines sont importantes pour la plupart des fonctions génétiques. Par exemple, un signal provenant de l'extérieur de la cellule est transmis à l'intérieur par des interactions protéine-protéine des molécules constituant le signal.

1.2.5 Réseaux d'interaction protéine-protéine (PPI)

En biologie moléculaire, un interactome est l'ensemble des interactions moléculaires dans une cellule particulière. Dans le cas des protéines, il est connu sous le nom d'interactions protéines-protéines IPP. Les interactions basées sur les IPP doivent être associées au protéome de l'espèce correspondante afin de fournir une vue globale ("omique") de toutes les interactions moléculaires possibles qu'une protéine peut présenter (Alonso-López et al., 2016).

Un réseau d'interaction protéine-protéine stocke les informations sur l'interactome protéine-protéine d'un organisme donné, c'est-à-dire l'ensemble de ses interactions protéine-protéine. Il a été suggéré que la taille des interactomes protéine-protéine augmente proportionnellement à la complexité biologique des organismes, aucun interactome protéine-protéine n'a été complètement identifié et, par conséquent, cette corrélation ne reste qu'une conjecture. En outre, les réseaux d'interaction protéine-protéine disponibles sont sujets à des erreurs, car les méthodes expérimentales utilisées pour découvrir les interactions peuvent inclure des faux positifs ou ne pas révéler certaines interactions existantes. Malgré cela, les réseaux d'interaction protéine-protéine sont particulièrement importants dans de nombreux contextes (Brouard, 2013) :

— l'analyse de ces réseaux facilite la compréhension des mécanismes qui déclenchent le début et la progression des maladies,

— les réseaux d'interaction protéine-protéine ont été utilisés pour découvrir de nouvelles fonctions des protéines et pour identifier des modules fonctionnels et des modèles d'interaction conservés,

— Certains travaux sur la structure des réseaux d'interaction protéine-protéine de plusieurs espèces ont permis de découvrir que, indépendamment de l'espèce, les réseaux d'interaction protéine-protéine sont sans échelle. Cela signifie que certaines protéines "hub" ont un rôle

central participant à la majorité des interactions alors que la plupart des protéines, qui ne sont pas des "hubs", ne participent qu'à une petite fraction des interactions.

1.2.6 Les voies biologiques

D'après le NIH (National Institute of Health) (National Human Genome Research Institute (NHGRI), 2015), une voie biologique est décrite comme une série d'actions entre les molécules dans une cellule qui conduit à un certain produit ou un changement dans la cellule. Une voie biologique peut déclencher l'apparition de nouvelles molécules, par exemple une protéine. Elles peuvent également activer ou désactiver les gènes ou stimuler une cellule à se déplacer.
Il existe de nombreux types de voies biologiques. Parmi les plus connues, nous pouvons citer les voies impliquées dans le métabolisme, dans la régulation des gènes et dans la transmission des signaux :

— Les voies métaboliques décrivent les réactions chimiques qui se produisent dans notre corps. Citons comme exemple d'une voie métabolique le processus de décomposition des aliments en molécules d'énergie qui peuvent être stockées pour une utilisation ultérieure,

— Les voies de régulation des gènes : permettent d'activer ou/et désactiver les gènes. Une telle action est essentielle parce que les gènes fournissent la recette pour que les cellules produisent des protéines, qui sont les composants nécessaires pour accomplir une tâche dans nos corps,

— Les voies de signalisation : permettent de déplacer un signal de l'extérieur d'une cellule vers son intérieur. Différentes cellules sont capables de recevoir des signaux spécifiques à travers des structures sur leur surface appelée récepteurs. Après une interaction avec ces récepteurs, le signal se déplace vers la cellule, où son message est transmis par des protéines spécialisées qui déclenchent une réaction spécifique dans la cellule. Par exemple, un signal chimique provenant de l'extérieur de la cellule peut orienter la cellule à produire une protéine particulière à l'intérieur de la cellule. À son tour, cette protéine peut être un signal qui conduit la cellule à se déplacer.

Nous retenons la définition de voie biologique comme un ensemble de gènes ou de produits de gènes qui interagissent pour assurer une fonction biologique. De ce fait, nous émettons l'hypothèse que des gènes ou des produits de gènes avec des annotations en « biological process » proches

seraient impliqués dans les mêmes voies biologiques.

La reconstruction d'une voie biologique est réalisée par le biais de techniques de bio-informatique. La bio-informatique se définit comme l'activité scientifique liée à la biologie fondée sur l'utilisation de programmes informatiques pour organiser, stocker et analyser l'information et les données biologiques. Son objectif est d'améliorer la compréhension des systèmes biologiques. C'est un domaine scientifique où la biologie, l'informatique et la technologie de l'information convergent en une discipline unique.

1.3 Sources de données

Notre travail consiste à détecter des communautés à partir des réseaux d'interaction de protéines chez l'être humain. Il existe deux types de sources de données utilisées dans notre travail :

— les ontologies qui annotent les protéines,

— les bases de données qui décrivent les interactions protéine-protéine et les voies biologiques.

Pour utiliser ces données, nous avons implémenté des programmes écrits en python afin d'extraire les données utiles pour notre travail. Dans cette section, nous présentons les sources de données utilisées, à savoir Gene Ontology (Bard and Rhee, 2004), String-dB (Snel et al., 2000) et les bases de voies biologiques (National Human Genome Research Institute (NHGRI), 2015). Nous présentons également les mesures de similarité sémantiques basés sur les Gene Ontology.

1.3.1 Bases de connaissances et ontologies

Il existe différentes bases de connaissances et ontologies qui répertorient et structurent les informations relatives au domaine biologique. Elles constituent une ressource essentielle pour l'annotation des connaissances.

1.3.1.1 Définition et propriétés d'une ontologie

Dans (Bard and Rhee, 2004), Bard et Rhee décrivent les ontologies comme des outils qui permettent de représenter précisément une collection de connaissances sous une forme utilisable par une machine. Ainsi, les concepts de cette ontologie sont décrits à la fois par leur signification et par

leurs relations.

Les concepts d'une ontologie sont organisés dans un graphe dont les relations peuvent être des relations sémantiques et/ou des relations de composition et d'héritage (au sens objet). Le graphe d'une ontologie est un graphe orienté, c'est-à-dire que les relations entre les nœuds ont un sens. Cela permet la description de la connaissance formalisée en allant des concepts les plus généraux aux plus précis. Dans une ontologie, une « classe » (ou « concept » ou « terme ») est un nœud du graphe. Les termes situés en amont d'un nœud sont ses « ancêtres » et ceux situés en aval sont ses « descendants ». Parmi les ancêtres d'un terme, ceux qui ne sont séparés de ce terme que par une relation sont ses « parents ». De même, parmi les descendants d'un terme, ceux qui ne sont séparés de ce terme que par une relation sont ses « enfants ». Le concept le plus général d'une ontologie n'a pas de parent ; il s'agit de la « racine ».

Dans une ontologie, il existe certaines relations telles que la relation « Is-a » qui sont transitives, elles permettent l'héritage des ancêtres : si un terme *C* est relié à un terme *B* par une relation «Is-a » et que *B* est également relié à *A* par un « Is-a » alors nous pouvons conclure que *C* « Is-a » *A*. Cette règle est vraie quel que soit le nombre de termes « intermédiaires ». En plus, la relation « Is-a » définit une hiérarchie de classes. Une ontologie peut comporter des propriétés affectées à certaines classes. Par exemple, on peut appliquer les propriétés des classes à toutes ses instances, qu'elles soient directes ou indirectes.

Les termes qui constituent une ontologie peuvent être utilisés pour décrire des données par un processus d'annotation sémantique. En effet, l'annotation sémantique consiste à formaliser l'interprétation qui peut être faite par des textes sous la forme de métadonnées attachées aux textes ou à certains de leurs segments. Cette interprétation s'exprime couramment en termes ontologiques quand il s'agit d'associer un type sémantique aux noms des entités mentionnées dans le texte (personnes, gènes, organisations, etc.) ou de les associer à un concept. L'intérêt d'une ontologie réside dans les trois propriétés suivantes :

— Une ontologie est générique, c'est- à-dire que la connaissance qui y est formalisée est toujours vraie, par opposition aux données annotées, qui sont anecdotiques. Par exemple, « Wallace est un chien » est une annotation anecdotique, alors que « les chiens sont des mammifères » est une connaissance universelle.

— Une ontologie permet le partage et la réutilisation des connaissances. En effet, une même

ontologie peut servir à annoter différents jeux de données

— Il est possible de procéder à du raisonnement sur une ontologie (Everitt et al., 2011). Plusieurs types de raisonnements peuvent être appliqués comme la généralisation, la classification, la mesure de distance ou de similarité entre concepts ou ensembles de concepts [(Kulik et al., 2005),(Zhang et al., 2002), (Zhao et al., 2009)].

En biologie, un besoin d'unification du vocabulaire s'est fait sentir lors de l'apparition des données de séquences sur des génomes entiers, et avec la découverte qu'un grand nombre de gènes et protéines qui sont à conserver. Les informations sur les gènes conservés permettent une meilleure compréhension des organismes et la connaissance de la fonction de l'un d'entre eux chez un organisme permet la déduction de sa fonction dans les autres organismes. L'importance de cette conservation souligne l'intérêt de l'utilisation d'une ontologie pour l'annotation des génomes. De plus, une annotation de génome utilisant un vocabulaire unifié devient transférable d'un génome à l'autre. Dans ce contexte, le recours à une ontologie est importante puisqu'elle permet une description, parmi les possibles, d'un domaine particulier. Elle contient des concepts et des relations admis par une communauté, et vise au partage et à la réutilisation des connaissances.
Nous détaillons dans la section suivante une première piste pour la modélisation des réseaux biologiques basée sur la proximité sémantique des annotations Biological Process de Gene Ontology. Nous avons décidé de détailler la présentation de Gene Ontology car c'est là notre source centrale de données.

1.3.1.2 Consortium Gene Ontology (GO)

Dans (Ashburner et al., 2000), Ashburner décrit GO comme un projet visant à standardiser la représentation des connaissances concernant les gènes et produits de gènes. Son but est de définir un vocabulaire structuré, contrôlé et précisément défini pour décrire la fonction des gènes et des produits des gènes dans tous les organismes. Cette ontologie se compose d'environ 30 000 termes d'annotation organisés sous forme de vocabulaire contrôlé. Elle comporte trois branches (sous ontologies) (voir figure 1.1) :

— Processus Biologiques (BP) : regroupe les annotations relatives à des processus biologiques dans lesquels interviennent des gènes ou des produits de gènes.

— Composants Cellulaires (CC) : décrit les structures cellulaires des organismes dans lesquels on peut trouver le produit des gènes.

— Fonctions Moléculaires (MF) : rassemble les termes qui décrivent les activités biochimiques des produits des gènes (les protéines au niveau moléculaire).

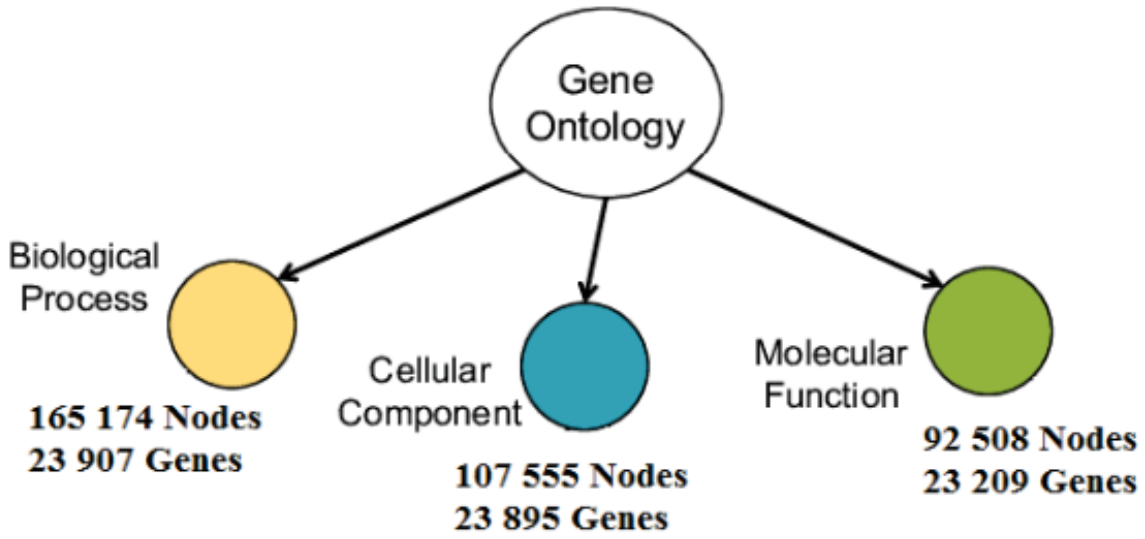


FIGURE 1.1 – Les trois sous-ontologies définies par GO (processus biologiques, composants cellulaires et fonctions moléculaires), et le nombre des nœuds et gènes correspondant à chaque sous-ontologie.

Cette ontologie est structurée sous forme de Graphe Acyclique Dirigé avec une racine unique (rDAG : rooted Directed Acyclic Graph), c'est-à-dire qu'il n'y a pas un chemin qui commence et se termine par le même nœud. Les nœuds du graphe de l'ontologie représentent les termes d'annotation et les arcs représentent les relations sémantiques qui existent entre les annotations. Comme dans n'importe quel vocabulaire structuré, la profondeur d'un terme d'annotation dans le graphe de l'ontologie est en lien avec la spécificité de ce terme. Plus un terme est profond, plus sa distance au nœud racine est grande, plus il apporte de détails sémantiques et fonctionnels sur les produits de gènes. Par conséquence, un terme descendant est plus spécifique que ses parents. Les trois aspects de l'ontologie (BP, CC, MF) constituent 3 sous-ontologies quasiment indépendantes de GO. Ils sont représentés sous forme de 3 sous-graphes (branches) orthogonaux. Généralement, une même protéine est annotée avec des termes de ces 3 ontologies et un terme peut avoir des relations avec des termes appartenant à la même sous-ontologie comme aux autres sous-ontologies.

La Figure 1.2 illustre un extrait de relations entre quelques termes de l'ontologie GO de l'aspect

Biological Process.

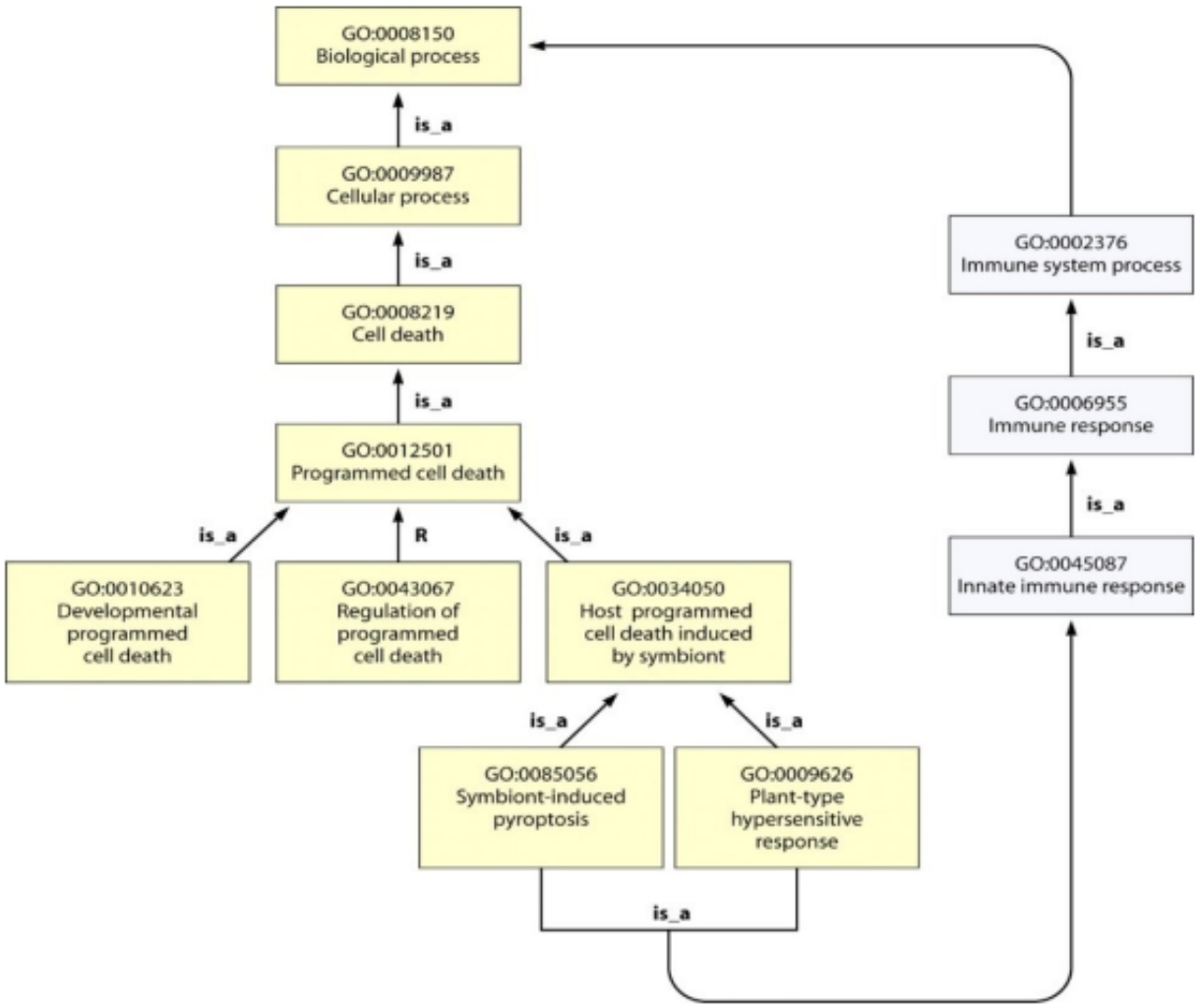


FIGURE 1.2 – Exemple simplifié illustrant quelques termes GO.

Il existe trois principales relations sémantiques dans cette ontologie :

— « Is-a » : une relation simple de type classe/sous-classe. A Is a B signifie que A est une sous-classe de B, c'est- à-dire que toutes les instances de A sont des instances de B. Citons comme exemple de la Figure 1.2 le terme GO :0009626 IS a GO :0034050.

— « Part of » : une relation de composition partielle. C part of D signifie que chaque instance de C est toujours une partie d'au moins une instance de D. Cela n'implique pas que toutes les instances de D aient au moins une partie qui soit une instance de C. Si A part of B part of C, alors A part of C.

— «Regulates » et ses 2 sous-relations «Positively Regulates» et «Negatively Regulates» décrivent une interaction entre un processus biologique et un autre. A Regulates B signifie que chaque instance de A régule B, mais que toutes les instances de B ne sont pas forcément régulées par A. Si A regulates B is a C, ou bien si A is a B regulates C, alors A regulates C , de même pour les relations Positively et Negatively Regulates.

En général, un terme GO possède une structure comme suit :

- un numéro d'accession unique (préfixe « GO ») : « 0006096 »
- un nom de terme : par exemple « cell killing »
- une « ontologie » : qui peut être BP ou MF ou CC
- une définition dont les sources sont mentionnées
- des commentaires sur la signification ou l'utilisation du terme
- le cas échéant : des synonymes qui ont un lien avec le nom du terme (« related ») ou qui sont exactement équivalents (« exact ») ou qui ont une acception plus large ou plus restrictive (« broad »)
- le cas échéant : des références croisées avec d'autres bases de données (« xref »)
- Relation

Ci-dessous, nous donnons un exemple de représentation d'un terme de GO :

- id : GO :0016049
- name : cell growth
- namespace : biological_process
- def : The process in which a cell irreversibly increases in size over time by accretion and biosynthetic production of matter similar to that already present.
- subset : goslim generic
- subset : goslim plant
- subset : gosubset prok
- synonym : « cell expansion » RELATED
- synonym : « cellular growth » EXACT
- synonym : «growth of cell » EXACT
- is_a : GO :0009987 ! cellular process
- is_a : GO :0040007 ! growth
- relationship : part_of GO :0008361 ! regulation of cell size

Plus précisément, GO comporte 40700 entrées dont la relation est «Is_a » réparties en 3639 pour la branche CC, 10713 pour MF et 26348 pour la branche BP. La relation « part_of » est présente

dans la branche BP avec 3239 entrées, 933 entrées pour la branche CC et pour la branche MF il y a seulement 3 relations de type « part_of ». Ce qui fait un total de 4175 relations de type « part_of » dans toute l'ontologie. La relation « regulates » est peu utilisée dans GO avec 1330 entrées seulement. Le consortium GO met régulièrement à jour l'ontologie en ajoutant de nouvelles annotations aux produits de gènes (Briche, 2009).

1.3.1.3 Gene Ontology Annotation (GOA)

GOA est un projet d'European Bioinformatics Institute (EBI). Ce projet a pour but d'annoter les produits de gènes de différentes espèces par des termes de GO. Il se base sur plusieurs bases de données d'annotation comme Uni-Prot [1] ou Ensembl [2] (Camon et al., 2004). GOA est donc un trait d'union entre les bases de données (d'interaction et de voies biologiques) et Gene Ontology. La façon dont un terme GO a été associé à un produit de gène au cours du processus d'annotation est précisée par un "Evidence Code" (EC). Il existe actuellement 21 codes différents. Ces EC sont séparés en 5 catégories principales de niveau de preuve : expérimental ("Experimental EC "), computationnel ("Computational Analysis EC"), déclaration d'auteur ("Author Statement EC"), déclaration de correcteur ("Curator Statement EC") et annotation automatique ("Automatically-assigned EC"). Tous ces niveaux sont subdivisés en EC plus précis, sauf le dernier qui ne contient que le code Inferred from Electronic Annotation (IEA), qui est le seul code qui qualifie une annotation non vérifiée par un correcteur (Rogers and Ben-Hur, 2009). Un gène est annoté par un ou plusieurs termes GO. Par exemple, le gène MEIKIN est annoté par l'ensemble de termes GO : [«GO :0007060 », « GO :0010789 », « GO :0016321 », « GO :0045143 », « GO :0051754 »].

1.3.2 Bases de données

Il existe plusieurs bases de données de voies métaboliques et d'interaction de gènes. Ci-dessous nous décrivons quelques bases de données publiques existantes et nous citons celles qui seront adoptées pour notre travail.

1. https ://www.uniprot.org/
2. https ://www.ensembl.org/info/genome/genebuild/index.html

1.3.2.1 Bases de données de voies biologiques

Les bases de voies biologiques diffèrent par deux aspects principaux. Premièrement, elles peuvent être dédiées à une ou plusieurs espèces. Deuxièmement, chacune d'entre elles définit différemment le découpage des suites de réactions qui constituent une voie biologique. Parmi les bases de données de voies biologiques, nous citons :

— Reactome [3] est une base de données de voies biologiques multiespèces. Les données sont toutes revues manuellement par des experts biologistes. L'unité de base employée pour décrire une voie biologique est la réaction. Les différentes entités biologiques participant aux réactions biochimiques forment un réseau d'interactions biologiques et sont groupés au sein de grandes voies biologiques (Croft et al., 2011).

— Biocarta [4] : cette base concerne plusieurs espèces. Elle permet de visualiser, construire ou identifier les réseaux cartographiant les relations génomiques et protéomiques connues. Elle offre une synthèse de ces voies et les représente par des graphiques. Chaque voie est illustrée avec une description assez détaillée au format texte (Nishimura, 2001).

— KEGG (Kyoto Encyclopedia of Genes and Genomes) est une base de données revue manuellement. Cette base concerne plusieurs espèces et a été développée pour analyser des fonctionnalités des cellules, des organismes et des écosystèmes (Kanehisa and Goto, 2000). Elle se base sur l'information moléculaire issue de technologies expérimentales à haut-débit telles que le séquençage de génomes. KEGG répertorie 2793 espèces, dont 192 eukaryotes. Parmi ceux-ci, on compte 26 vertébrés dont l'Humain, la Souris et la Poule.

Les voies biologiques correspondent à des communautés de gènes réelles. La base de voie biologique utilisée dans notre travail est KEGG car c'est qui a été proposée par notre expert en biologie du laboratoire de recherche en Génétique, Immunologie et Pathologies Humaines (LGIPH) de la Faculté des Sciences de Tunis. Les autres bases de voies biologiques sont utilisées pour valider les communautés obtenues dans la partie résultats expérimentaux.

3. http ://www.reactome.org/
4. https ://cgap.nci.nih.gov/Pathways/BioCarta_Pathways

1.3.2.2 Search Tool for Recurring Instances of Neighbouring Gene (STRING)

Cette base de données recense les interactions protéines-protéines. C'est l'une des ressources les plus complètes. Elle possède une très bonne interface de navigation permettant l'exploration et la visualisation des associations protéines-protéines connues et prédites selon différents critères : voisinage physique de gène sur le chromosome, existence d'un évènement de fusion entre deux gènes, co-occurrence de deux gènes dans différentes espèces, co-expression de gènes, interaction protéique connue et obtenue expérimentalement, co-citation des gènes ou des protéines dans une référence bibliographique (Snel et al., 2000). Un exemple de visualisation de différents types d'interactions et d'association du gène A1CF d'organisme Homo sapiens avec les autres gènes CDYL, APOBEC1, CELF2, SYNCRIP, MPHOSPH8, CBX3, APOBEC3G, APOBEC3D, APOBEC3E et KHSRP est illustré dans la figure 1.3 selon différents critères. Au niveau de cette figure, les différentes couleurs des liens représentent l'origine de voisinage ou de l'association des gènes ou protéines : Vert (gènes voisins sur le chromosome), Bleu (Cooccurrence des gènes des différentes espèces), Violet (interaction protéique expérimentale), Turquoise (association/interaction décrites par des autres bases) et Jaune (interaction/association décrite dans la littérature).

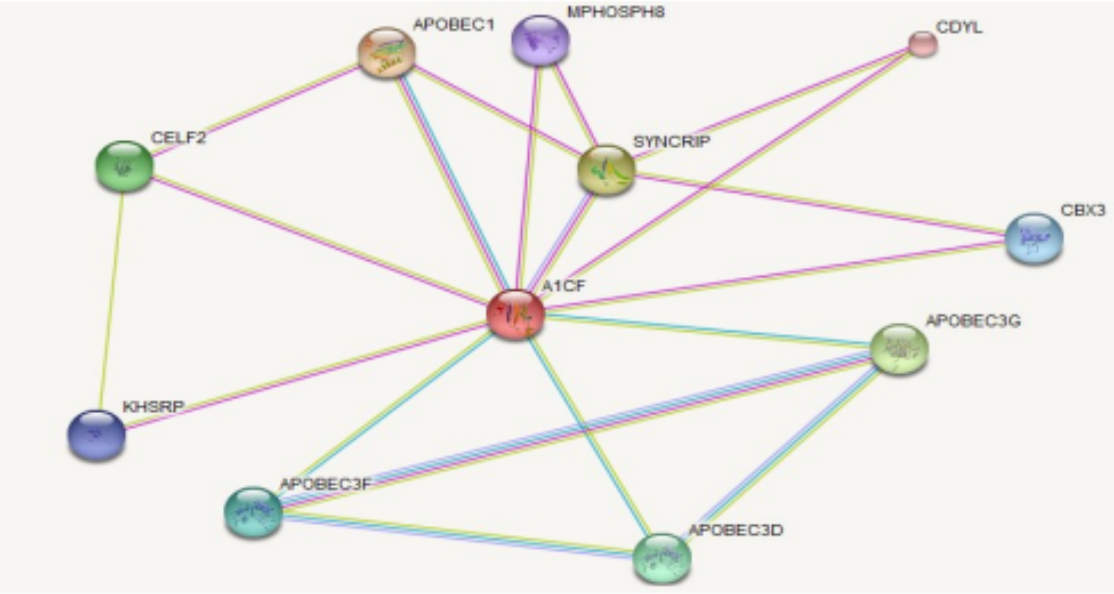


FIGURE 1.3 – Exemple de visualisation de différents types d'interactions et d'association connus selon différents critères pour le gène A1CF, obtenu de la base STRING organisme Homo Sapiens.

1.4 Similarité sémantique sur Gene Ontology

L'identification de la similarité dans les ontologies est un concept fondamental qui est adopté par plusieurs techniques telles que le regroupement, la fouille de données (data mining) ou le Web sémantique (Thabet Slimani and Ben Yaghlane, 2007). La notion de similarité est mise en avant dans plusieurs domaines d'activités liés à l'ingénierie des ontologies tels que l'apprentissage, l'alignement ou encore le peuplement d'ontologies (Thabet Slimani and Ben Yaghlane, 2007).

En général, la similarité sémantique est une notion utilisée pour mesurer la proximité entre deux concepts appartenant à la même ontologie ou à deux ontologies différentes. Le développement de Gene Ontology a permis l'émergence d'une nouvelle notion, la similarité fonctionnelle [(Wang et al., 2007),(Pesquita et al., 2009),(Ruths et al., 2009). Cette notion est utilisée pour mesurer la proximité fonctionnelle entre deux gènes en se basant sur la similarité sémantique entre leurs annotations fonctionnelles.

Nous nous intéressons à déterminer la similarité entre des gènes, car nous avons émis l'hypothèse qu'une communauté de gènes se caractérise par des gènes qui sont similaires sémantiquement. Notre objectif consiste à déterminer des groupes de gènes similaires. Pour cela, nous calculons la similarité entre un ensemble de gènes.

Un gène peut être annoté par plusieurs termes de GO. Pour calculer la similarité entre deux gènes, il nous faut utiliser une approche permettant de comparer des ensembles de termes d'une ontologie afin de quantifier la similarité entre ces ensembles. Dans la littérature, plusieurs travaux sur la mesure de similarité sémantique entre les objets d'une ontologie ont été développés dans différents contextes.

Soit Sim une fonction de similarité et $G1$ et $G2$ deux gènes, $Sim(G1;G2)$ est grande si les deux gènes $G1$ et $G2$ sont similaires, sinon cette valeur est faible. Cette mesure nécessite les propriétés mathématiques suivantes :

$$\begin{cases} Sim(G1;G2) \in [0,1] \\ Sim(G1;G1) = 1 \\ Sim(G1;G2) = Sim(G2;G1) \end{cases} \tag{1.1}$$

Pour interpréter les résultats, il n'est pas possible d'établir formellement que deux gènes sont similaires ou que l'un d'eux a des particularités significatives sans disposer d'un seuil de similarité.

Jusqu'à présent, ces interprétations se faisaient sur la base d'un seuil implicite ou arbitraire. Il existe trois grandes familles d'approches pour l'identification de la similarité sémantique entre les termes d'une ontologie en général (Thabet Slimani and Ben Yaghlane, 2007) : des approches basées sur les nœuds, des approches qui reposent sur les arrêtes et des approches hybrides. Dans les sections suivantes, nous présentons des mesures relatives à chaque famille.

1.4.1 Approches basées sur les nœuds

Les mesures de similarité de cette approche sont fondées sur la notion de Contenu Informationnel (CI) pour déterminer la similarité conceptuelle. Ces approches reposent sur la recherche du plus proche parent commun et du nombre de liens jusqu'à ce plus proche parent. En plus, la similarité entre les concepts est déterminée par le degré de partage de l'information. Une variation de mesure de similarité basée sur le contenu informationnel est adoptée pour trouver une meilleure façon d'organiser et d'interroger les données d'une ontologie de gène comme les méthodes de Resnik (Resnik, 2011), celle de Lin (Lin, 1998) et celle de Rel et Jiang (Jiang and Conrath, 1997).

— **Mesure de Resnik**

Dans (Resnik, 2011), Resnik a développé une mesure pour calculer la similarité entre des termes de Word-Net. Il attribue à chaque terme une valeur qui représente la quantité d'information qu'il contient, on parle de « Contenu d'information » **(Information content IC)**. Ce concept est utilisé par toutes les autres mesures basées sur les nœuds. Plus un terme est rare, plus il est considéré comme informatif, plus son IC est élevé. L'IC du terme c est calculé de la façon suivante :

$$IC(c) = -log(P(c)) \tag{1.2}$$

Avec $P(c)$: la probabilité de trouver une instance du concept c. La probabilité d'un concept c est calculée en divisant le nombre des instances de c par le nombre total des instances. La similarité sémantique entre deux termes est définie par la quantité d'information qu'ils partagent : elle est égale à **l'IC de leur ancêtre commun le plus informatif (Most Informative Common Ancestor, MICA)**. Elle est définie comme suit :

$$IC(MICA) = max_{c \in A(T1,T2)}[-log(P(c)] = SIM_{RES}(T_1, T_2) \tag{1.3}$$

Avec $A(T_1, T_2)$: l'ensemble des ancêtres communs de deux termes T_1 et T_2.

$SIM_{RES}(T_1, T_2)$: similarité de Resnik entre T_1 et T_2.

— **Mesure de Jiang et Conrath**

Dans (Jiang and Conrath, 1997), Jiang and Conrath partent du constat qu'utiliser seulement l'ancêtre commun n'offre pas une granularité assez fine et proposent de prendre en compte la quantité d'information portée par les deux concepts comparés et leur MICA (Most Informative Common Ancestor). La similarité de Jiang and Conrath, notée SIM_{JIANG} s'exprime comme suit :

$$SIM_{JIANG}(T_1, T_2) = max_{c \in A(T1,T2)}(1 - IC(T_1) + IC(T_2) - 2 * IC(MICA)) \tag{1.4}$$

— **Mesure de Lin**

Dans (Lin, 1998), Lin définit la similarité entre deux termes comme le rapport de la quantité d'information nécessaire pour décrire le point commun de deux termes et la quantité d'information spécifique de chaque terme. Le point commun des termes est capturé par leurs ancêtres communs. Lin propose également une mesure de similarité très proche, qui revient essentiellement à une reformulation sous forme de rapport de la formule de Jiang and Conrath. La mesure de similarité telle que décrit dans (Lin, 1998) est :

$$SIM_{LIN}(T_1, T_2) = max_{c \in A(T1,T2)}(\frac{2 * log(P(c))}{log(P(T_1) + log(P(T_2)}) \tag{1.5}$$

— **Mesure de Relevance**

Cette mesure, notée $SIM_{REL}(T_1, T_2)$, est une méthode qui combine les mesures de similarité de Lin (Lin, 1998) et Resnik (Resnik, 2011). Elle est dans (Thabet Slimani and Ben Yaghlane, 2007) comme suit :

$$SIM_{REL}(T_1, T_2) = max_{c \in A(T1,T2)}(\frac{2 * log(P(c))}{log(P(T_1) + log(P(T_2)} * (1 - P(c))) \tag{1.6}$$

Ce type de mesure, basée sur les nœuds, présente l'inconvénient de ne pas tenir compte de la position dans l'ensemble de l'ontologie de ces concepts (T_1, T_2) et $MICA(T_1, T_2)$). Par conséquent, des mesures entre des termes généraux peuvent être similaires à des concepts précis.

1.4.2 Approches basées sur les arrêtes

Le principe de calcul de similarité de cette approche est basé sur l'idée suivante : plus le chemin entre deux nœuds est court plus ils sont semblables. La mesure de similarité la plus intuitive des objets dans une ontologie est la distance qui sépare ces objets dans l'ontologie. Le calcul des distances dans l'ontologie est basé sur un graphe de spécialisation des objets. Dans chaque graphe, la distance de l'ontologie doit être caractérisée par le plus court chemin qui fait intervenir un ancêtre commun ou le plus petit généralisant, connectant potentiellement deux objets à travers des descendants communs. Parmi les travaux classifiés sous cette bannière on peut citer la méthode de Rada (Rada et al., 1989), Wu & Palmer (Wu and Palmer, 1994).

— **Mesure de Wu & Palmer**

Dans (Wu and Palmer, 1994), Wu et Palmer décrivent le principe de cette similarité comme suit : étant donné une ontologie O formée par un ensemble de nœuds et un nœud racine R. Soit T_1 et T_2 deux éléments de l'ontologie dont nous désirons calculer la similarité en utilisant trois distances :

— N_1 : la distance entre le nœud T_1 et la racine R ;

— N_2 : la distance entre le nœud T_2 et la racine R ;

— N : la distance entre le plus proche ancêtre commun de T_1 et de T_2 et la racine R.

La mesure de Wu et Palmer est définie par la formule suivante :

$$SIM_{WU(T1,T2)} = \frac{2 * N}{N_1 + N_2} \tag{1.7}$$

— **Mesure de Rada**

Dans (Rada et al., 1989), Rada suggère que pour mesurer la distance entre deux concepts ontologiques, notée $dist(T_1, T_2)$, on se base sur le nombre d'arcs minimum à parcourir pour aller du concept T_1 au concept T_2. Cette mesure est comparable à l'algorithme de Dijkstra qui calcule le plus court chemin entre deux points d'un graphe. La mesure de similarité est ainsi de la forme :

$$SIM_{RADA(T1,T2)} = \frac{1}{1 + dist(T_1, T_2)} \tag{1.8}$$

$$dist(T_1, T_2) = cheminmin(T_1, T_2) \tag{1.9}$$

Ces approches reposent sur deux hypothèses :

1. Les nœuds et les arêtes sont uniformément distribués sur la totalité de l'ontologie
2. Les arêtes se trouvant sur un même niveau ont des distances équivalentes à celles entre termes du même niveau.

L'utilisation de ces méthodes de similarité posent plusieurs problèmes, notamment celui des chemins à considérer : est-ce le chemin le plus long ? Le chemin le plus court ? Ou bien un chemin intermédiaire ? Cette question ne se pose pas si on considère que Wu et Palmer appliquent leur distance sur des ontologies simples avec un seul chemin possible entre les différents nœuds. Ce qui n'est pas le cas d'une ontologie aussi complexe que GO qui comporte plus de 45 000 termes et 8 types de relations. Pour atténuer ces hypothèses, on peut à titre d'exemple ajouter un poids au nœud en fonction de la profondeur hiérarchique ou utiliser la densité des liens et considérer le type de lien (Pesquita et al., 2009).

1.4.3 Approches hybrides

Ces approches combinent les approches basées sur les nœuds avec les approches basés sur les arêtes. Elles essaient de combiner la quantification de l'information du terme et aussi sa position dans l'ontologie (son niveau de précision) pour pallier les limites des deux approches citées ci-dessus (basées sur les nœuds ou sur les arêtes). Parmi les approches de cette méthode : les mesures de Wang (Wang et al., 2007) et GS2 (GO-based similarity of gene sets) (Ruths et al., 2009) qui sont définies spécialement pour GO.

— **Mesure de Wang**

Wang attribue à chaque terme une valeur sémantique qui est la somme des contributions sémantiques de ses ancêtres (Wang et al., 2007). Cette démarche relève de la catégorie des mesures basées sur les nœuds. Cependant, cette valeur sémantique est elle-même calculée en parcourant le graphe, ce qui relève de la catégorie des mesures basées sur les arrêtes. Wang commence par calculer les contributions sémantiques des ancêtres de chacun des termes à comparer. Cette contribution sémantique $S_A(t)$ d'un terme t à un terme A dépend du nombre

et de la nature des relations qui les séparent (coefficient W_e).

$$\begin{cases} S_A(A) = A \\ S_A(t) = max\{w_e * S_A(t') \mid \text{ t' } \in \text{ childrenof(t) if t } \neq \text{A}\} \end{cases} \tag{1.10}$$

Avec : A désigne un terme de GO.

Wang propose deux valeurs de coefficient w_e : 0.8 pour une relation is-a et 0.6 pour une relation part of pour toute l'ontologie GO.

Ensuite, il faut calculer la valeur sémantique (SV) de chaque terme à comparer, elle est définie d'après l'équation suivante :

$$SV(A) = \sum_{t \in T_A} S_A(t) \tag{1.11}$$

Avec T_A : l'ensemble des ancêtres du terme A.

À partir des valeurs sémantiques calculées, on peut mesurer la similarité sémantique entre deux termes A et B respectivement dotés des ensembles d'ancêtres T_A et T_B comme suit :

$$S_{GO}(A,B) = \frac{\sum_{t \in T_A \cap T_B} (S_A(t) + S_B(t))}{SV(A) + SV(B)} \tag{1.12}$$

La similarité entre un terme t et un ensemble go de k termes de GO notés $\{go_1, go_2, \ldots, go_k\}$ se calcule comme suit :

$$Sim(t, go) = \max_{1 \leq i \leq k} (S_{GO}(t, go_i)) \tag{1.13}$$

Enfin, la similarité entre deux gènes G_1 et G_2 est la similarité entre les ensembles des termes qui les annotent. Cette similarité se calcule comme suit :

$$Sim(G_1, G_2) = \frac{\sum_{1 \leq i \leq m} (Sim_{GO}(go_{1i}, GO_2) + \sum_{1 \leq j \leq n} (Sim_{GO}(go_{2j}, GO_1))}{m+n} \tag{1.14}$$

Où : m et n sont les tailles de l'ensemble des termes qui annotent respectivement les gènes G_1 et G_2.

Les deux gènes G_1 et G_2 sont annotés respectivement par un ensemble de termes GO tel que : $GO_1 = \{go_{11}, go_{12}, \ldots, go_{1m}\}$ et $GO_2 = \{go_{21}, go_{22}, \ldots, go_{1n}\}$.

— **Mesure GS2 (GO-based similarity of gene sets)**

Cette méthode offre des résultats expérimentaux similaires à ceux de Wang mais elle est beaucoup plus rapide. Cela la rend très appropriée pour des études à grande échelle des ensembles de gènes et de leurs annotations GO.

Pour calculer la similarité d'un ensemble de gènes, chaque gène est comparé avec l'ensemble des gènes restants en calculant à quel point il fait suite à la distribution de la fonctionnalité des autres. La distribution de la fonctionnalité est représentée par la distribution des termes ancêtre de GO pour chaque gène. Ensuite, il faut appliquer la fonction **Rank** pour quantifier la répartition des termes ancêtres (Ruths et al., 2009).

$$Rank_y(i) = |\{g_j \in y : i \in A_{gj}\}| \tag{1.15}$$

Avec :

— i : un terme de GO.

— $y = \{g_1, g_2, ..., g_k\}$: ensemble de k gènes annoté par des termes GO. Ces dernières doivent être les ancêtres de i.

— A_{gj} : l'ensemble des ancêtres du terme g_j.

La fonction de similarité de GS2 est définie pour un ensemble de gènes noté $y = \{g_1, g_2, ..., g_k\}$ comme suit :

$$GS2(y) = \frac{1}{|y|} \sum_{g_i \in y} Comp(g_i, y - \{g_i\}) \tag{1.16}$$

$$Comp(g_i, h) = \frac{1}{G_i} \sum_{j \in G_i} \frac{1}{|A_{\{j\}}|} \sum_{k \in A_{\{j\}}} \frac{Rank_h(k)}{[h|} \tag{1.17}$$

Avec :

— g_i : gène i.

— $h = \{y - g_i\}$.

— G_i : ensemble de terme GO qui annote le gène i.

— $A_{\{j\}}$: ancêtre du terme j.

1.4.4 Choix de la mesure de similarité

En général, la similarité sémantique est une notion utilisée pour mesurer la proximité entre deux concepts appartenant à la même ontologie ou à deux ontologies différentes. Le développement de Gene Ontology a permis l'émergence d'une nouvelle notion, la similarité fonctionnelle [(Wang et al., 2007),(Pesquita et al., 2009), (Ruths et al., 2009)]. Cette notion est utilisée pour mesurer la proximité fonctionnelle entre deux produits de gènes en se basant sur la similarité sémantique entre leurs annotations fonctionnelles. Dans ce cadre, plusieurs mesures ont été développées comme présentées précédemment. Dans cette partie, nous nous sommes intéressés aux mesures entre termes et notre objectif est de déterminer la mesure la plus pertinente dans notre contexte.
Pour déterminer une valeur de similarité entre les gènes, nous avons utilisé la bibliothèque GOSemSim [5] pour estimer la similarité sémantique des gènes basés sur les méthodes suivantes : Resnik, Lin, Jiang et Conrath, Rel and Wang. De plus, nous avons créé un code en python pour tester la mesure GS2.
Généralement, un gène est annoté par un ensemble de termes GO. En utilisant une méthode de calcul de similarité, nous déterminons la similarité sémantique de deux gènes en comparant l'ensemble de termes d'annotations qui les définissent. Pour interpréter les résultats obtenus, il faut définir un seuil. Nous avons utilisé un seuil égal à 0.5 proposé par notre expert en biologie :

— Si la valeur de similarité entre deux gènes $>= 0.5$ alors ces deux gènes sont similaires

— Sinon ils ne sont pas similaires.

Afin de choisir une mesure de similarité à adopter pour notre travail, nous avons réalisé des tests sur onze communautés de réseaux de gènes extraites de la base KEGG. Nous savons donc a priori que ces groupes de gènes devraient avoir des gènes qui sont similaires. Chaque méthode de calcul de similarité prendra en entrée un ensemble de gène et en sortie donnera une matrice de similarité symétrique. À partir de chaque matrice de similarité, nous déterminons le nombre de paires de gènes similaires par rapport à toutes les combinaisons possibles. Les résultats des tests sont décrits dans le tableau 1.1. Chaque communauté est entre parenthèse sa taille (nombre de gènes).

Dans le tableau 1.1, le taux moyen désigne le pourcentage moyen de paires de gènes similaires obtenus pour chaque méthode. Par exemple, pour la communauté R1, la mesure GS2 trouve que

5. http ://bioconductor.org/packages/release/bioc/html/GOSemSim.html

TABLE 1.1 – Nombre de paires de gènes similaires obtenues pour chaque mesure de similarité.

Communauté connue	Resnik	Jiang	Lin	Rel	Wang	GS2
R1 (61 gènes)	61/1891	78/1891	68/1891	68/1891	855/1891	1657 /1891
R2(10 gènes)	13/55	33/55	37/55	31/55	31/55	49/55
R3 (11 gènes)	15/66	34/66	26/66	27/66	26/66	42/66
R4 (41 gènes)	125/861	111/861	105/861	141/861	322/861	536/861
R5 (13 gènes)	22/91	73/91	71/91	68/91	48/91	81/91
R6 (66 gènes)	15/66	54/66	51/66	51/66	28/66	60/66
R7 (14 gènes)	15/105	57/105	64/105	65/105	67/105	100/105
R8(15 gènes)	23/120	58/150	95/120	95/120	93/120	105/120
R9 (16 gènes)	15/136	57/136	54/136	43/136	41/136	95/136
R10 (17 gènes)	18/153	70/153	65/153	55/153	50/153	108/153
R11 (7 gènes)	0/28	8/28	8/28	6/28	4/28	11/28
Taux moyen	9.09%	16.68%	17.95%	18.17%	44.05%	80.45%

1657 paires de gènes sur 1891 sont similaires, soit 87.63% alors que ce pourcentage ne dépasse pas 45,2% avec les autres mesures.

À la lecture de ce tableau, nous constatons que la mesure de similarité GS2 est celle qui permet de détecter le plus grand nombre de gènes similaires sur les 11 réseaux KEGG. En effet, le taux moyen de détection de similarité par GS2 atteint 80.45%. Nous constatons donc que la mesure ayant conduit aux meilleurs résultats est la mesure GS2 dans ces tests expérimentaux. C'est celle que nous avons donc retenue dans nos travaux de thèse.

1.5 Données extraites

Au niveau de cette section, nous présentons un récapitulatif sur les données extraites et utiles pour notre thèse. La figure 1.4 résume toutes les données extraites utiles pour notre travail.

— un gène est décrit par un ID, un nom et un ensemble de termes qui l'annotent qui sont récupérés de Gene Ontology (GO) et Gene Ontology Annotation (GOA). Nous avons obtenu 17404 gènes. Citons par exemple la description du gène MEKIN : ID : 728637 || NOM : MEIKIN || Termes qui l'annotent : ["GO : 0007060", "GO :0010789", "GO :0016321", "GO :0045143", "GO :0051754"].

— Les données relatives à une interaction entre paires de gènes sont le résultat d'un programme

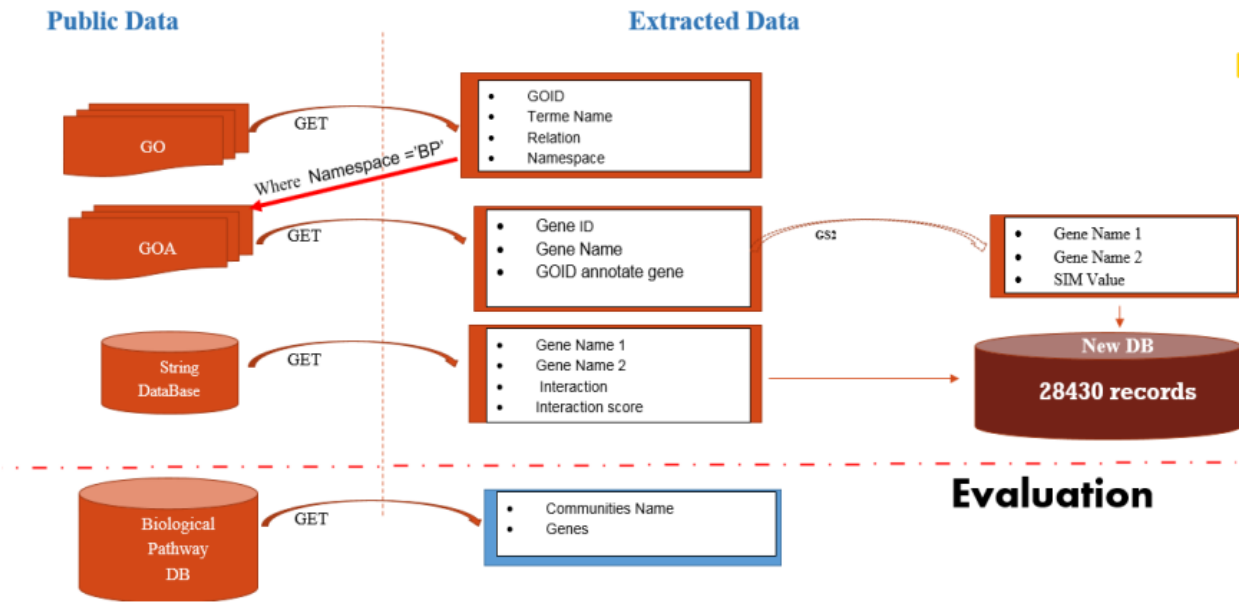


FIGURE 1.4 – Récapulatif des données utilisées.

écrit en python. Ce programme permet d'interroger l'interface graphique de STRING-db pour obtenir les données d'interaction de chaque paire. Par exemple l'interaction entre HSPA1A et le GRPEL1 est décrite comme suit : Nom1 : «HSPA1A»|| Nom2 : «GRPEL1»|| Action : «catalysis» || valeur d'interaction : 335

— Une voie biologique est décrite par un nom et un ensemble de gènes. Ces données sont extraites de la base KEGG.

1.6 Conclusion

Dans ce chapitre, nous avons situé ce travail dans son cadre général en définissant brièvement quelques notions biologiques. Ensuite, nous avons mené une présentation des sources de données utilisées dans notre travail. Nous avons également présenté différentes mesures de similarité sémantiques utilisées dans GO et expliqué le choix de la mesure retenue. Dans le chapitre suivant, nous présentons les réseaux complexes et des méthodes de détection de communautés dans ces derniers. Un focus spécial est porté sur les algorithmes génétiques qui constituent la base de notre travail.

Chapitre 2

Détection de communautés : un état des lieux

Sommaire

2.1 Introduction

Beaucoup de systèmes complexes dans divers domaines comme la biologie peuvent être représentée de manière abstraite par des réseaux comme le réseau d'interaction de gènes. Une des caractéristiques communes que l'on retrouve dans de nombreux réseaux concerne l'existence de zones fortement connectées entre elles. Ces zones sont habituellement appelées communautés. La détection de ces zones est un outil important pour la compréhension des structures et des fonctionnements des grands réseaux. Notre projet s'inscrit dans ce contexte interdisciplinaire et se concentre sur la question algorithmique de détection de communautés de gènes à partir d'un réseau d'interaction protéinique. De tels réseaux peuvent être modélisés en termes de graphes, où un nœud représente un membre individuel du système et une arête représente un lien entre les nœuds selon une relation bien déterminée du système.

Ce chapitre tente de rapporter l'essentiel concernant la notion de communauté et présentera le cadre formel de la détection de communautés. Mais avant cela, il convient d'introduire au préalable une présentation des réseaux complexes et leurs modélisations par des graphes, ainsi que certaines notions et définitions utiles de la théorie des graphes. Nous décrivons ainsi les principales approches proposées dans la littérature pour la détection de communautés. De ce fait, la liste des travaux cités, reste non exhaustive. Nous nous sommes particulièrement intéressés dans le cadre de cette recherche aux approches basées sur les algorithmes génétiques. Nous présentons ainsi les différentes notions des algorithmes évolutionnaires relatives à la notion de détection de communautés.

2.2 Modélisation des réseaux complexes par les graphes

En théorie des graphes, un réseau complexe est un réseau possédant une architecture et une topologie complexe et irrégulière. Comme tous les réseaux, les réseaux complexes sont composés de nœuds (ou sommets ou points) représentant des objets, interconnectés par des liens (ou arêtes ou lignes). Ces réseaux sont des représentations abstraites des relations principalement présentes dans la vie réelle comme les systèmes biologiques (réseau d'interaction de protéines) et les systèmes technologiques (réseau de télécommunication, réseau aérien). L'étude de ces réseaux a fait l'objet d'une grande attention de la part de la communauté scientifique depuis le début des années 2000

(Hafez et al., 2012).

Les graphes permettent de modéliser de nombreux phénomènes qui proviennent d'horizons très variés, citons comme exemple les réseaux d'interaction de protéines où les sommets sont des gènes, ces sommets sont liés par leurs interactions chimiques. En fait, la théorie des graphes fournit un support de modélisation des réseaux complexes tel que :

— un élément constitutif du réseau (individu, ordinateur, gène ou protéine, etc.) est représenté par un sommet ou un nœud du graphe ;

— une relation ou un lien entre deux éléments est représenté par une arête ou un arc du graphe.

Cette modélisation permet d'exprimer les propriétés distinctives des réseaux complexes, et d'y appliquer des algorithmes pour résoudre les problèmes que ceux-ci soulèvent. Il convient maintenant d'introduire certaines notions et définitions utiles de la théorie des graphes en se basant sur (Hafez et al., 2012).

Graphe : Un graphe non orienté est composé d'un ensemble V de sommets (ou nœuds) et d'un ensemble E de paires (non ordonnées) de sommets nommées arêtes (ou liens).

Nous adoptons les notations suivantes : n représente le nombre de sommets ($n = |V|$) et N le nombre d'arêtes ($N = |E|$), le graphe est dit d'ordre n et de taille N.

Les arêtes du graphe peuvent être pondérées grâce à une fonction de poids $w : E \longrightarrow \mathbf{R}^+$ permettant de modéliser plus finement les interactions entre sommets, nous obtenons ainsi un graphe pondéré $G = (V, E, w)$.

— Deux sommets liés dans un graphe sont dits voisins l'un de l'autre.

— On qualifie de chemin entre deux sommets, une séquence de liens consécutifs dont ils sont les extrémités ; la longueur de ce chemin sera le nombre de liens qu'il comporte ; la distance entre deux sommets sera le minimum des longueurs de chemins allant de l'un à l'autre.

— Si un graphe ne contient pas de liens associant un sommet à lui-même (boucle), on parle de graphe simple.

Graphe connexe : Un graphe $G = (V, E)$ est dit connexe si, quels que soient les sommets u et v de V, il existe un chemin de u vers v.

Graphe complet : Un graphe non orienté $G = (V, E)$ est dit complet si quel que soit la paire $(u, v) \in V$, il existe un arc $\in E$ reliant les deux sommets u et v.

Sous-graphe : C'est un graphe constitué de certains sommets de G et de toutes les arêtes qui les relient.

Taille (Size) : La taille d'un réseau est le nombre total de sommets dans le réseau. Elle est notée $(N = |E|)$.

Clique : Une clique de G est un sous-graphe complet de G.

Densité : La densité d'un graphe est définie comme le rapport entre le nombre d'arêtes et le nombre maximum d'arêtes possibles compte tenu du nombre de nœuds du graphe.

Voisinage : Le voisinage d'un nœud correspond à l'ensemble de tous ses nœuds adjacents. Autrement dit, l'ensemble des voisins d'un sommet $v \in V$, notée $N(v)$, est défini comme $N(v) = \{u \in V | (v,u) \in E\}$.

Chemin : Le chemin entre les nœuds u et v est une séquence $(v0,...,vk)$ de nœuds tel que $v_0 = u, \quad v_k = v \quad et \quad \forall \ i : \quad 0 \leq i < k, (v_i, v_{i+1}) \in E$. C'est une succession d'arêtes qui permettent de naviguer d'un nœud à un autre.

Distance : La distance entre deux sommets est la longueur de la plus courte chaîne entre ces sommets ; elle est aussi appelée distance géodésique.

Degré d'un nœud ou valence : C'est le nombre de liens (arêtes ou arcs) reliant ce sommet. Le degré d'un sommet s est noté deg $deg(s)$. **Diamètre :** Le diamètre d'un graphe est la plus grande distance qui puisse exister entre deux sommets de ce graphe.

Longueur moyenne de chemin : C'est la moyenne des longueurs du plus court chemin entre toutes les paires possibles de nœuds.

Modularité : Elle est définie comme la différence entre le nombre de liens présents dans un module (ou groupe, ou cluster, ou communauté), et le nombre de liens attendus dans un graphe. La valeur de la modularité est dans l'intervalle [-1 ;1]. Elle permet de mesurer et de justifier la qualité et la pertinence d'un partitionnement de graphe. Si on considère P la partition en p clusters du graphe $G = (V,E)$, on note : $P = c_1,...,c_n,...c_p$.

La modularité (Q) est définie comme suit :

$$Q(P) = \sum_i (e_{ci} - a_{c_i}^2) \tag{2.1}$$

avec e_{ci} la part des liens d'un cluster c_i sur le total, a_{ci} la probabilité qu'un sommet se trouve dans le cluster c_i et donc $a_{c_i}^2$ la probabilité que les deux sommets d'un lien se trouvent dans le même

cluster c_i. Cette expression générale 2.1 est transformée dans la forme usuelle de présentation de la modularité (Fortunato, 2010). La modularité peut s'écrire sous la forme :

$$Q(P) = \frac{1}{2m} \sum_{i,j \in V} (A_{ij} - \frac{d_i}{d_j}) \beta(c_i, c_j) \tag{2.2}$$

Avec

- m le nombre d'arêtes du graphe ;
- A la matrice d'adjacence du graphe ;
- A_{ij} le poids des liens entre les sommets i et j ;
- d_i la somme des degrés de i avec $d_i = \sum_i A_{ij}$;
- $a^2 c_i = \frac{\sum_j d_i * d_j}{4m^2}$
- $\beta(c_i, c_j)$ une fonction de Kronecker qui vaut 1 si les deux sommets appartiennent à la même communauté et 0 sinon.

Nous rapportons dans les sections suivantes, les diverses définitions relatives à la notion de communautés. Nous proposons ensuite la définition formelle et l'intérêt de la détection de communautés dans les réseaux d'interaction de protéines.

2.3 Graphes et réseaux de gènes

Les réseaux peuvent représenter toutes sortes de systèmes du monde réel. Par exemple, on pourrait décrire l'Internet comme un réseau où les sommets sont des ordinateurs ou d'autres périphériques et les arêtes sont des connexions physiques entre les dispositifs. Le World Wide Web est un énorme réseau où les pages sont des sommets et les liens sont les arêtes. D'autres exemples incluent les réseaux sociaux, les réseaux de publications liés par des citations, les réseaux de transport, les réseaux métaboliques, etc [(Jeong et al., 2000),(Ravasz et al., 2002)].

Les réseaux biologiques font partie du monde des réseaux. Plusieurs exemples de tels réseaux existent comme le réseau transcriptionnel, le réseau d'interaction protéine-protéine qui tient compte des relations entre protéines, ou le réseau métabolique qui cherche à modéliser les réactions métaboliques d'un organisme ou encore les réseaux de neurones et les réseaux alimentaires. Un réseau

biologique peut être dirigé ou non dirigé en fonction de la relation biologique utilisée pour joindre les paires de sommets dans le réseau (Ravasz et al., 2002). Le réseau biologique auquel nous intéressons dans notre travail est le réseau d'interaction protéine-protéine (PPI). Ce réseau est souvent représenté comme un réseau non dirigé dans lequel les nœuds correspondent à des protéines, où un arc entre deux nœuds signifie l'existence d'une interaction entre les deux protéines correspondantes et le poids d'un arc désigne le nombre de présence de cette interaction expérimentalement. Cependant, cette représentation ne prend pas en compte le type d'interaction entre les deux protéines correspondantes (exemple : Building, Reacting, ...).

PPI est caractérisé par différentes propriétés topologiques. La caractéristique topologique la plus élémentaire d'un nœud est son degré, c'est-à-dire le nombre d'interactions existantes entre ce nœud et les autres nœuds dans le réseau. La figure 2.1 illustre un exemple de réseau d'interaction protéine-protéine chez la levure du boulanger (Jeong et al., 2001).

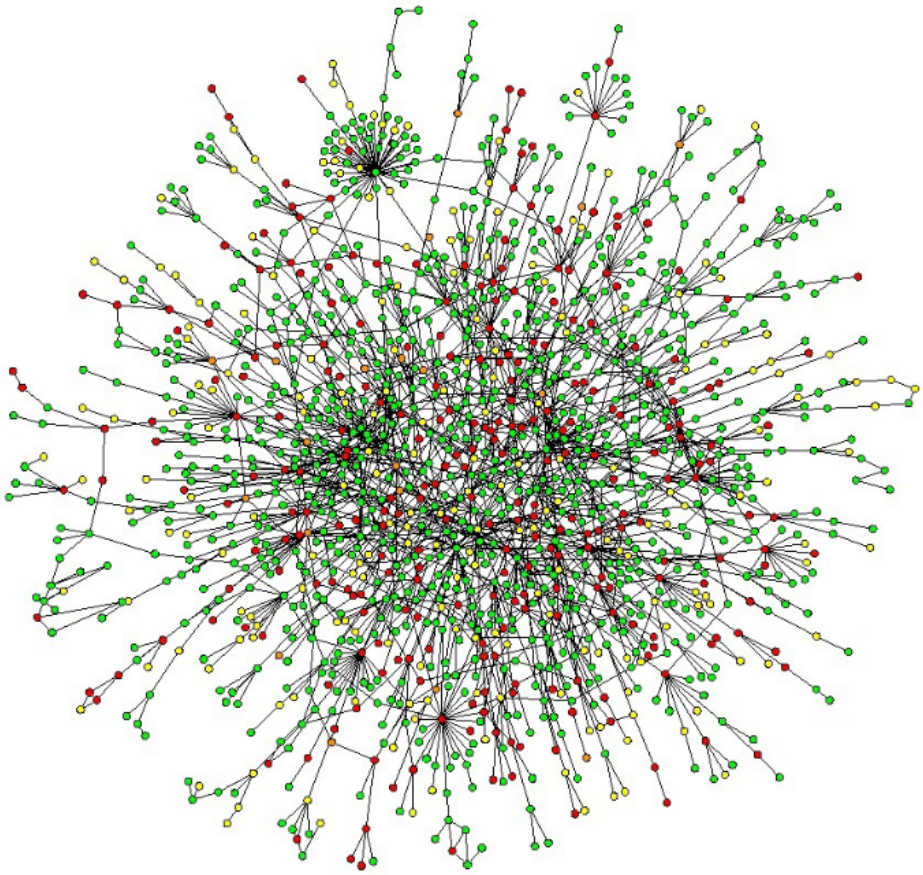


FIGURE 2.1 – Visualisation d'un réseau d'interaction protéine-protéine chez la levure du boulanger (Jeong et al., 2001)

Dans ce travail, nous avons constaté à partir de la base String-DB (voir section 1.3.2.2 du chapitre 1) que les interactions entre deux protéines sont directionnelles et peuvent être de plusieurs types. Ainsi, chaque paire de protéines peut avoir plusieurs valeurs d'interaction différentes.

2.4 Notion de détection de communautés

Les communautés dans un réseau complexe ne sont pas définies de façon unanime, mais l'idée de connexions internes fortes et de connexions externes faibles est partagée par tous. Ainsi, bien qu'aucune définition formelle de ce qu'est une communauté ne soit actuellement reconnue, ce terme désigne intuitivement un groupe de nœuds du réseau plus fortement connectés entre eux qu'avec les autres nœuds du réseau. Les applications de ce concept sont variées selon les disciplines d'origine des graphes. Là encore, le cadre général des réseaux complexes, fourni par la théorie des graphes, permet d'identifier des communautés et de développer des méthodes de détection de communautés.

2.4.1 Définition d'une communauté

Une communauté, que nous appelons aussi groupe ou module, est définie de façon la plus universelle et minimaliste comme un sous-ensemble de nœuds possédant plus d'arêtes internes (c'est-à-dire entre nœuds du même groupe) que d'arêtes externes (c'est-à-dire avec une extrémité dans le groupe et l'autre en dehors). Ce terme désigne un ensemble d'individus ayant une ou plusieurs caractéristiques en commun. Cette notion de communauté a fait l'objet de nombreuses études depuis plusieurs décennies. Nous retrouvons cette extension d'usage dans plusieurs disciplines telles que la biologie, plus précisément les réseaux d'interaction protéine-protéine et les réseaux sociaux (Bornholdt and Schuster, 2003).

Il n'y a pas de consensus sur la définition d'une communauté. Il est cependant communément admis qu'il s'agit d'une partie dense du graphe (Fortunato, 2010). Girvan et Newman (Girvan and Newman, 2002) définissent une communauté comme un ensemble d'entités au sein duquel il y a plus de relations internes qu'externes. Radicchi et al. en (Radicchi et al., 2004) ont proposé deux définitions d'une communauté selon une condition forte et une condition faible. Le degré interne d'un nœud v par rapport à sa communauté C, noté, $d_{in}(v)$ est défini comme le nombre d'arêtes

incidentes à v ayant leurs deux extrémités dans C. Par opposition, le degré externe $d_{out}(v)$ est le nombre d'arêtes incidentes à v ayant une extrémité en dehors de C. Pour un graphe pondéré, ces degrés s'entendent comme la somme des poids des arêtes concernées. Par extension, le degré interne $d_{in}(C)$ d'une communauté C est la somme des degrés internes de ses nœuds, ce qui revient au double du nombre d'arêtes internes, chacune étant comptée deux fois pour chacun des deux nœuds extrémités. Le degré externe $d_{out}(C)$ d'une communauté C est la somme des degrés externes de ses nœuds. À partir de ces définitions, une communauté C est :

$$\begin{cases} forte \quad si \quad \forall \ v \in C, \ d_{in}(v) > d_{out}(v); \\ faible \quad si \quad \sum_{v \in C} \ d_{in}(v) > \quad \sum_{v \in C} \ d_{out}(v), \quad si \quad d_{in}(C) > d_{out}(C). \end{cases} \tag{2.3}$$

2.4.2 Intérêts des communautés

L'existence des communautés dans les réseaux constitue une caractéristique commune indépendante du domaine d'origine du réseau. La détection de communautés peut donc être vue comme un procédé commun qui peut s'appliquer à tous les réseaux rencontrés. La structure communautaire existe dans de nombreux réseaux réels et, la plupart du temps, elle a une signification concrète en termes d'organisation. En réalité, les communautés peuvent avoir des interprétations différentes suivant le type de réseau considéré. Ainsi, dans les réseaux d'interaction Protéine-Protéine en biologie, les communautés correspondent généralement à des ensembles de gènes qui collaborent à une même fonction cellulaire. L'identification des communautés permet de mieux comprendre la structure de ces réseaux, de regrouper et d'identifier les nœuds qui jouent potentiellement des rôles semblables. Il existe plusieurs approches pour la détection des communautés, nous nous s'intéressons en particulier, dans ce travail, à celles basées sur les algorithmes génétiques (AGs).
Dans la section suivante, nous allons décrire les différentes notions liées aux algorithmes génétiques.

2.5 Introduction aux algorithmes génétiques

Les AGs font partie de la classe des algorithmes dits stochastiques. En effet, une grande partie de leur fonctionnement est basée sur le hasard. Cependant, ce hasard est dirigé grâce à la fonction

d'évaluation (fitness) qui permet d'introduire dans les opérateurs génétiques une quantité de déterminisme utile afin d'obtenir une solution satisfaisante (Holland, 1975). Ce sont des algorithmes évolutionnaires pour l'optimisation heuristique s'appuyant sur l'évolution au fil des générations d'un ensemble d'individus où chaque individu représente une solution potentielle au problème. Chaque génération d'un AG implique une sélection compétitive qui exclut progressivement les mauvaises solutions à travers l'évaluation d'une valeur de fitness qui indique la qualité d'un individu en tant que solution du problème. Par ailleurs, un ensemble d'opérateurs génétiques comme la mutation et le croisement sont appliqués à chaque génération pour créer de nouveaux individus qui viendront remplacer toute ou une partie de la population (Petrowski and Ben-Hamida, 2017).
Les AGs peuvent être particulièrement utiles dans les domaines suivants :

— Optimisation : optimisation de fonctions, planification, etc.

— Apprentissage : classification, prédiction, robotique, etc.

— Programmation automatique : programmes LISP, automates cellulaires, etc.

— Étude du vivant, du monde réel : marchés économiques, comportements sociaux, systèmes immunitaires, etc.

Les AGs puisent une grande partie de leur terminologie de la biologie, et pour une bonne compréhension de bon nombre de ces analogies, nous donnons ici un certain nombre de définitions. Nous expliquons ensuite le fonctionnement d'un AG standard.

2.5.1 Terminologie

Les AGs n'ont pas seulement imité la sélection naturelle, mais ils ont aussi emprunté son vocabulaire. Ce paragraphe énumère les différents termes utilisés en donnant leurs définitions succinctement.

Gène ou symbole : Un gène (ou symbole) est une unité d'information génétique. Dans les AGs, on appelle gène la suite de symboles qui codent la valeur d'une variable. Dans le cas général, un gène correspond à un seul symbole (0 ou 1 dans le cas binaire). Dans ce présent travail, pour éviter toute confusion avec les termes biologiques, nous utiliserons le terme "Symbole".

Génotype : Correspond à l'ensemble des valeurs des gènes.

Individu : Représente le codage d'une solution potentielle (un élément de l'espace de recherche).

Chromosome : Un chromosome est constitué de symboles qui regroupe l'intégralité de son patrimoine génétique. Généralement, un AG utilise un chromosome par individu. De même, pour éviter toute confusion avec notre domaine d'étude, nous utiliserons les termes individu et solution.

Performance : Représente la mesure de la qualité (fitness) des individus basés sur l'objectif de l'optimisation et permettant de comparer les individus entre eux afin d'en déterminer les plus et moins aptes.

Évaluation d'un individu : Elle représente le calcul de la performance de l'individu.

Population : Un ensemble fini (de taille N) d'individus.

Évolution : Représente un processus d'optimisation itératif de recherche d'une (ou plusieurs) solution(s).

Génération : Correspond à l'itération, mais ce terme signifie parfois la population en une certaine itération.

Croisement : Il s'agit d'un opérateur de reproduction (Crossover) appliqué avec la probabilité P_c et qui correspond à un brassage d'information entre les individus de la population. Il consiste à échanger des parties composantes (symboles) entre deux ou plusieurs individus.

Mutation : C'est un opérateur de modification d'un ou plusieurs symboles appliqués avec la probabilité P_m dans le but d'introduire une nouvelle variabilité dans la population.

Sélection parentale : Processus du choix des individus pour la reproduction basée sur leur performance. Cette opération est exécutée deux fois dans la boucle évolutionnaire. Elle détermine les individus qui vont se reproduire par application des opérateurs génétiques, le plus souvent avec remise.

Remplacement : Appelé aussi sélection environnementale, il représente le processus de formation d'une nouvelle population à partir des ensembles de parents et d'enfants, effectué le plus souvent sur la base de leurs performances. Il est indépendant de la représentation des individus. Il confère à l'AG un caractère élitiste ou générationnel.

2.5.2 Principe et fonctionnement

Le principe récurrent dans l'utilisation des algorithmes génétiques est l'adaptation des principes de l'évolution naturelle à des entités représentant des solutions potentielles d'un problème à résoudre ou à optimiser. Une entité est représenté par un individu constitué de symboles qui contiennent

les caractères héréditaires de l'individu. Les principes de sélection, de croisement et de mutation s'inspirent des processus naturels de même nom. Pour un problème d'optimisation donné, un individu représente un point de l'espace d'états, donc une solution potentielle. On lui associe la valeur du critère à optimiser, traduisant dans le contexte darwinien son adaptation à l'environnement. On génère ensuite de façon itérative (voir Figure 2.2) des populations d'individus sur lesquelles on applique les processus de sélection, de croisement et de mutation. La sélection a pour but de favoriser les meilleurs éléments de la population pour le critère considéré (les mieux adaptés), le croisement et la mutation assurent l'exploration de l'espace d'états.

Avant d'expliquer le fonctionnement d'un AG standard, nous expliquons la notion de codage.

Codage : Il existe essentiellement deux types de codage : le codage binaire ou représentation sous forme de chaîne binaire et le codage réel qui est une représentation directe des valeurs réelles de la variable. Le codage initialement utilisé par John Holland dans l'AG standard est le codage binaire (Holland, 1975). Nous allons par conséquent d'abord présenter ce cas.

— *Codage binaire :* On appelle "séquence ou chaîne" de longueur $L(A)$ une séquence avec $A = \{a_1, a_2, ..., a_l\}$ *avec* $\forall i \in \{1, ...l\}$ $a_i \in V = \{0, 1\}$. Chaque individu est constitué d'une chaîne de l bits. Un individu est donc une suite de bits en codage binaire, appelé aussi chaîne binaire. Présentant beaucoup d'inconvénients pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire a été remplacé par le codage réel.

— *Codage réel :* Dans le cas d'un codage réel, les solutions sont codées en utilisant des vecteurs réels. La séquence A ne contient qu'un point dans dans l'espace de recherche, avec $A = \{a\}$ avec $a \in \mathbb{R}^N$ (avec N dimension de l'espace de recherche) . Ce type de codage est généralement plus précis et offre des solutions plus efficaces.

Nous expliquons le fonctionnement d'un AG standard qui est représenté par la figure 2.2. Ce fonctionnement général est générique et ne dépend pas du codage. Seuls les opérateurs de reproduction dépendent de la représentation utilisée.

On commence par générer une population aléatoire d'individus POP_0. Pour passer d'une génération t à la génération $t+1$, les opérations suivantes sont effectuées :

— Dans un premier temps, une population de parents est sélectionnée de POP_t pour la reproduction (POP_{sel}).

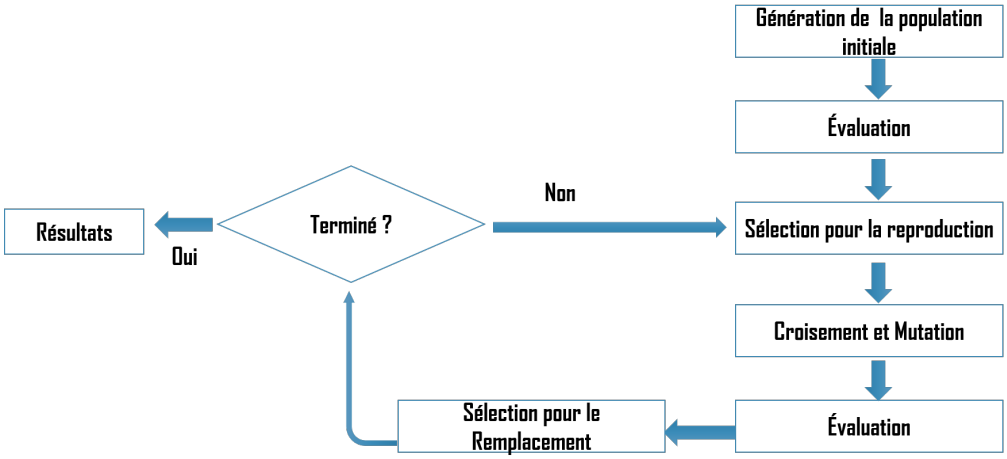


FIGURE 2.2 – Principe général des algorithmes génétiques.

— Pour chaque couple de parents dans POP_{sel} (P1,P2), un croisement est appliqué avec une certaine probabilité P_c pour générer deux enfants (C1,C2).

— Un opérateur de mutation est également appliqué aux nouveaux enfants (C1,C2) avec une certaine probabilité P_m (généralement très inférieure à P_c).

— Enfin, les nouveaux individus (Ci) sont évalués et éventuellement intégrés à la population de la génération suivante POP_{t+1} en appliquant une sélection environnementale.

— Ce processus itératif, appelé évolution, s'arrête si une condition d'arrêt est satisfaite comme le nombre maximum de générations atteint.

L'algorithme génétique standard est donné ci-après (Algorithme 1), avec les notations suivantes :

— POP_t : population à la génération t ;

— POP_sel : population sélectionnée pour la reproduction.

— POP_child : population enfant.

Nous expliquons avec plus en détails dans ce qui suit chaque étape de l'algorithme génétique standard présenté précédemment dans l'algorithme 1.

1- ***Initialisation de la population***

Généralement, la population initiale est choisie aléatoirement, chacun des symboles de chaque individu étant choisi au hasard dans l'espace de recherche. Cependant, rien n'interdit d'utiliser des initialisations gloutonnes, de fournir des solutions déjà connues que l'on désire améliorer. Ainsi, on peut utiliser d'autres algorithmes de recherche qui fourniront à l'AG des

Algorithme 1 Algorithme Génétique Standard (SGA : Standard Genetic Algorithm) (Holland, 1975).

Entrée: Fonction de fitness, P_c, P_m
Sortie: meilleure solution
Début
1: t = 0 ;
2: Initialisation de la population POP_t ;
3: Évaluation des individus de la population POP_t ;
4: **tant que** Condition de terminaison non satisfaite **faire**
5: t ← t + 1 ;
6: *POP_sel* ← Sélection et copies des parents de la population POP_{t-1} dans *POP_sel* ;
7: *POP_child* ← Application des opérateurs génétiques (croisement et mutation) sur *POP_sel* (avec les probabilités P_c et P_m) ;
8: Évaluation *POP_child* avec la fonction de performance ;
9: POP_t ← *POP_child* (pour le remplacement générationnel)
10: **fin tant que**
11: Retourner la meilleure solution ;
End

solutions ayant subi une première phase d'optimisation. Le choix de l'initialisation se fera en fonction des connaissances que l'utilisateur a du problème. S'il n'a pas d'informations particulières, l'initialisation aléatoire est préférée, la plus uniforme possible, afin de favoriser une exploration maximale de l'espace de recherche.

2- ***Évaluation :***

L'opérateur d'évaluation ou la fitness est un opérateur très dépendant du problème à traiter. Il est généralement défini par type de problème, voire par problème. Cet opérateur est loin d'être banal. Il sert à l'opérateur de sélection pour faire son choix des individus à conserver. Il agira donc sur la convergence de l'algorithme.

3- ***Condition de terminaison :***

La condition de terminaison d'un AG diffère suivant le type du problème traité. Si on traite un problème de décision, c'est-à-dire que l'on connaît la valeur de l'optimum que l'on cherche à atteindre, la fonction fitness serait la convergence à cette valeur. En revanche, pour des problèmes d'optimisation, où l'optimum est inconnu et on ne sait jamais si celui-ci est atteint, généralement on se restreindra au nombre de générations maximum à réaliser depuis le début ou depuis la dernière amélioration trouvée. On peut aussi, décider d'arrêter lorsqu'il y a une convergence prématurée. On parle de convergence prématurée quand la diversité entre les

individus est insignifiante, pour espérer sortir d'un bassin d'attraction.

4- ***Sélection pour la reproduction :***

L'opérateur de sélection parentale sert à choisir, dans la population courante, les individus qui auront le droit de survivre et de se reproduire. Plus clairement, cet opérateur va générer, à partir de la population courante, POP_{t-1} une population *POP_sel* par copie des individus choisis dans POP_{t-1} . Les opérateurs génétiques seront ensuite appliqués sur la population *POP_sel*. Le nombre d'occurrences d'un individu X_i de la population POP_{t-1} dans la population *POP_sel* tend généralement à être proportionnel au rapport entre sa fitness et la fitness moyenne. De fait, les individus les mieux adaptés ont plus de chance de figurer (parfois plusieurs fois) dans la population *POP_sel* et les individus les moins adaptés ont peu de chance d'y figurer. Nous relatons ici les procédures de sélection les plus utilisées :

— **La roulette :** la sélection des individus par le système de la roulette s'inspire des roues de loterie. Cette méthode consiste à affecter une probabilité de sélection à chaque individu proportionnelle à sa fitness. Se voyant affectés ces probabilités, tous les individus de la population peuvent participer à la reproduction. En effet, une roulette est divisée en secteurs, un secteur par individu. L'inconvénient de cette méthode réside dans la présence probable d'un super individu. Le super individu est un individu dont la probabilité de sélection est très supérieure à celle des autres individus. Il risque alors d'être toujours choisi, ce qui peut limiter le champ d'exploration et la diversité de la population. De plus, sa variance et son coût sont élevés.

— **Sélection par tournoi :** cette technique utilise la sélection proportionnelle sur un petit groupe d'individus, puis choisit dans ce groupe l'individu ayant le meilleur score d'adaptation. L'avantage de cette méthode est d'éviter d'avoir le super individu qu'on peut rencontrer dans la méthode de sélection précédente. Par contre, le meilleur individu peut ne pas être sélectionné par cette méthode, et ainsi le champ d'exploration est réduit.

— **Sélection par rang :** elle est similaire à la roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation. La sélection par rang trie d'abord la population par fitness. Ensuite, chaque individu se voit associé un rang en fonction de sa position. Ainsi le pire individu aura le rang 1, le suivant 2, et ainsi de suite jusqu'au

meilleur individu qui aura le rang N (pour une population de N individus)

5- ***Sélection pour le remplacement :***

Cette étape du processus itératif consiste en l'incorporation des nouvelles solutions dans la population courante. Les nouvelles solutions sont ajoutées à la population courante par remplacement *(total ou partiel)* des anciennes solutions (parents). Les procédures de remplacement les plus utilisées sont :

— **Remplacement déterministe :** Il est utilisé essentiellement dans les stratégies d'évolution (Petrowski and Ben-Hamida, 2017). Son caractère purement déterministe lui donne un rôle clef dans l'évolution vu qu'il guide la recherche vers les zones des meilleurs individus. On note que μ (= les parents) et λ (= les enfants). Il opère en sélectionnant les μ $(1 < \mu \leq \lambda)$ meilleures solutions parmi :

(a) l'union des μ parents et λ enfants : schéma appelé $(\mu + \lambda)$. Dans ce schéma, les meilleurs des $(\mu + \lambda)$ parents plus enfants sont les parents de la génération suivante. Ce remplacement est élitiste et garantit une amélioration monotone de la performance de la population, mais il s'adapte mal à un éventuel changement d'environnement.

(b) l'ensemble des λ enfants : schéma appelé (μ, λ). Dans ce schéma, les meilleures μ solutions dans les λ enfants deviennent les parents de la génération suivante.

— **Remplacement générationnel :** Avec cette approche, la population des enfants remplace entièrement la population parente. La durée de vie d'un individu est alors d'une seule génération. Il peut être associé au concept d'élitisme qui consiste à copier un ou plusieurs des meilleurs individus dans la nouvelle population. Son objectif est d'éviter la perte des meilleures solutions rencontrées.

— **Remplacement « steady-state » :** L'idée principale est qu'une grande partie de la population puisse survivre à la prochaine génération. À chaque génération sont sélectionnés quelques individus (parmi ceux qui ont le meilleur coût) pour créer des individus fils. Ensuite, les individus les plus mauvais sont retirés et remplacés par les nouveaux. Le reste de la population survit à la nouvelle génération.

6- ***Les opérateurs génétiques :***

La reproduction consiste à créer de nouvelles solutions (les enfants) à partir des solutions

présentes au début de la génération et sélectionnées à l'étape précédente (les parents). Les AGs se sont inspirés des méthodes de reproduction du vivant pour créer des opérateurs de diversification.

— **Opérateur de croisement :** est généralement considéré comme un opérateur d'exploitation. Il va permettre de découvrir de meilleures solutions en combinant les avantages de solutions déjà découvertes. Dans la version classique d'un AG, nous choisissons deux individus de la population *POP_sel* et leur appliquons l'opérateur de croisement en fonction de la probabilité de croisement. Il existe plusieurs types de croisement. Les plus connus sont :

(a) Croisement à un point (Figure 2.3 a)) : créer deux enfants à partir des deux parents de telle sorte que chacun des enfants ait une partie de chaque parent.

(b) Croisement 2-points (Figure 2.3 b)) : même principe que le croisement simple, mais sélectionne deux points et applique le croisement sur la section entre les deux points.

(c) Croisement uniforme : sélectionner plusieurs points de coupure au hasard et appliquer le croisement sur ces points.

FIGURE 2.3 – Illustration des principes de croisement 1-point (a) et 2-points (b).

— **Opérateur de mutation** : permet l'exploration de l'espace de recherche. En effet, les autres opérateurs ne permettent que des déplacements dans certaines zones de l'espace de recherche. En utilisant la mutation, on peut théoriquement atteindre n'importe quelle zone de l'espace de recherche. Dans le cas général, cela consiste à effectuer une altération aléatoire d'un ou de plusieurs symboles d' individu (Figure 2.4) en fonction du taux de mutation, généralement assez faible.

FIGURE 2.4 – Illustration du principe de mutation.

Il est à noter que les opérateurs génétiques dépendent du codage choisis et sont souvent modifiés selon le contexte. Nous nous sommes limités à la présentation du principe de fonctionnement des opérateurs standards qui ont inspiré notre travail.

2.6 Approches de détection de communautés

Cette section présente une brève revue de littérature sur la détection de communautés. Comme il existe de nombreuses approches proposées, nous allons retenir celles ayant reçu le plus d'intérêt de la part de la communauté scientifique. Ces approches illustrent aussi la diversité de méthodologies et donnent une vue d'ensemble des techniques proposées selon leurs principes méthodologiques. Nous catégorisons ces différentes approches en deux groupes : les approches analytiques et les approches évolutionnaires.

2.6.1 Approches analytiques

Les approches analytiques sont basées sur le partitionnement de graphes. Cela consiste à regrouper les nœuds d'un graphe en un nombre généralement prédéterminé de sous-groupes homogènes ou clusters en minimisant le nombre de liens entre les différents groupes, de telle sorte que les nœuds d'un même groupe ou cluster se ressemblent le plus possible alors que les nœuds de communautés différentes sont les plus différents possible. Un cluster ainsi défini est potentiellement une communauté. Dans ce contexte, nous présentons les deux familles d'approches les plus connues, à savoir les approches hiérarchiques et les approches de partitionnement (Newman, 2006).

2.6.1.1 Les approches hiérarchiques

Les méthodes hiérarchiques génèrent une succession de partitions emboîtées les unes dans les autres au lieu d'une seule partition de l'espace des données. Un cluster peut être divisé en sous

clusters, l'ensemble des clusters étant généralement représenté par un arbre. Un objet appartient à une et une seule feuille dans la hiérarchie, mais également à son nœud père, et ainsi de suite jusqu'à la racine. Les méthodes de clustering hiérarchiques se subdivise en deux grandes familles :

1. Les algorithmes agglomératifs : ils consistent à décomposer l'ensemble d'individus en une arborescence de groupes (communautés). Nous avons initialement n clusters (où n est le nombre de nœuds). On commence par calculer les distances entre les clusters et fusionner les deux clusters les plus proches pour former un nouveau cluster. À chaque étape, on recalcule toutes les distances entre les communautés et on fusionne deux communautés. Après la dernière fusion, nous disposons du dendrogramme qui représente les différentes communautés obtenues après chaque itération. Le problème qui se pose alors est comment choisir la bonne structure de communautés au cours des itérations. Newman propose de faire appel à la modularité (= la différence entre la valeur d'adjacence entre deux nœuds de la même communauté et la probabilité pour que ceux-ci soient connectés) pour traiter le dendrogramme et dégager les communautés finales (Newman, 2006). On obtient un dendrogramme et un découpage du graphe en communautés, comme le présente la figure 2.5.
   Le problème de cette approche agglomérative est que le nombre de communautés k doit être connu à l'avance.

2. Les algorithmes divisifs : ils consistent à scinder un réseau en plusieurs communautés en éliminant itérativement les liens entre les nœuds. Ils commencent par une seule communauté (=le réseau entier), en haut du dendrogramme, jusqu'à avoir n communautés à un seul nœud représentant les feuilles du dendrogramme. Dans chaque itération, tout réseau connexe est considéré comme une communauté. Les méthodes existantes se distinguent par le choix des liens à éliminer et par les poids accordés aux liens. Nous retenons l'algorithme le plus connu de Newman And Girvan *Edge Betweenness* (Newman and Girvan, 2004) comme exemple d'algorithme divisif. Les méthodes de détection de communautés s'intéressent généralement aux nœuds du réseau. Or, Newman dans (Newman and Girvan, 2004) s'est penché sur les liens plutôt que sur les nœuds. En effet, l'identification des liens se trouvant entre les communautés, ainsi que leur élimination, permet d'identifier les différentes communautés dans un réseau. Afin de trouver les liens inter-communautés, l'algorithme *Edge Betweenness* accorde à chaque lien une mesure de centralité d'intermédiarité (*Edge Betweenness Centrality*).

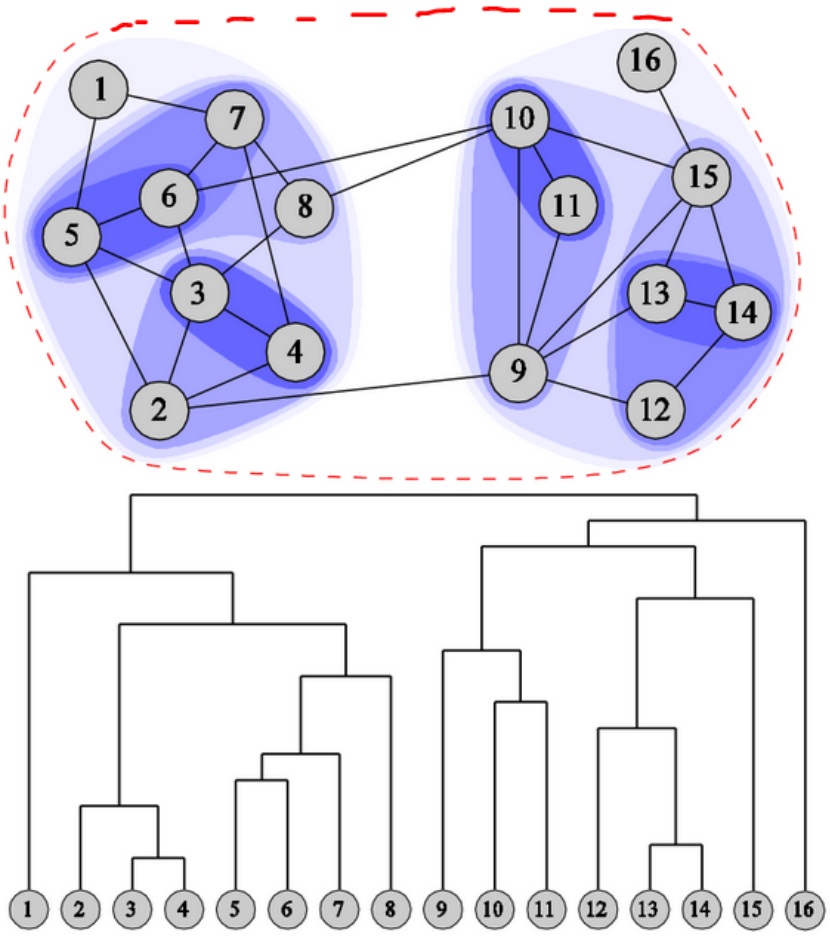


FIGURE 2.5 – Exemple d'un dendrogramme et un découpage du graphe en communautés.

Cette mesure se base sur le calcul du plus court chemin.

2.6.1.2 Approches basées sur le partitionnement de graphe

Le but du partitionnement de graphe est de diviser un graphe G en plusieurs sous-graphes (correspondants à des communautés). En pratique, la plupart des approches de partitionnement de graphes procèdent par une division du graphe en deux sous-graphes, puis par un partitionnement récursif des deux sous-graphes ainsi obtenus. L'arrêt du partitionnement a lieu lorsque le nombre de communautés souhaité est atteint. Mais cette approche ne convient pas totalement à ce que l'on cherche, car elle possède l'inconvénient de requérir des connaissances préalables sur le nombre de communautés recherchées et leurs tailles. Les deux méthodes ayant connu le plus de succès sont la méthode de bissection spectrale (Barnes, 1982) et la méthode de Kernighan et Lin (Kernighan and Lin, 1970). Dans (Barnes, 1982), Barnes a défini le principe de la méthode de bissection spec-

trale en se basant sur le calcul des vecteurs propres. Plus précisément, elle est basée sur le premier vecteur propre non trivial de la matrice de Laplace. Le graphe est ainsi séparé en deux parties en fonction du signe de leur composante selon ce vecteur propre. La méthode de Kernighan et Lin (Kernighan and Lin, 1970) procède par bissection afin de trouver la coupe du graphe minimisant le nombre d'arêtes existantes entre les deux groupes. L'algorithme nécessite en paramètre la taille des communautés à détecter, une coupe de la bonne taille est choisie aléatoirement comme point de départ et utilise une heuristique gloutonne. La bissection est améliorée progressivement en effectuant des permutations de sommets entre les communautés. À chaque étape, on échange les deux sommets fournissant la meilleure réduction du nombre d'arêtes externes, avec la condition imposée de ne jamais changer deux fois un même sommet (Kernighan and Lin, 1970).

Dans le domaine biologique, différentes méthodes de clustering (ou de segmentation) ont été proposées pour l'identification et la détection de communautés.

La méthode la plus utilisée est la méthode de flux ou Markov Clustering (MCL) développée par Stijn Van Dangen dans le cadre du partitionnement de graphes de grandes tailles (Van Dongen, 2000). L'idée principale de cette méthode est de considérer que les sommets avec le même comportement de marche aléatoire sont dans le même cluster. Le but de l'algorithme MCL est de grouper les sommets dont les marches aléatoires associées convergent vers le même état. En fait, cet algorithme simule un flux sur le graphe en calculant les puissances successives de la matrice d'adjacence associée. À chaque itération, une étape de gonflage (inflation) est appliquée pour améliorer le contraste entre les régions de flux fort ou faible dans le graphe. Le processus converge vers une partition du graphe, avec un ensemble de régions à haut débit (les clusters) séparées par des frontières sans flux. La valeur du paramètre d'inflation influence fortement le nombre de clusters (Van Dongen, 2000). Brohee et Van (Brohee and Van Helden, 2006) ont constaté que MCL donne des résultats satisfaisants pour l'extraction de communautés à partir de réseaux d'interactions protéine-protéine.

Un autre algorithme de clustering utilisé dans le réseau biologique est Restricted Neighborhood Search Clustering (RNSC) (King et al., 2004). Il s'agit d'un algorithme de clustering qui utilise des méthodes de recherche locale pour améliorer de manière attrayante la qualité des clusters obtenus. Cet algorithme explore l'espace de solution en minimisant une fonction de coût basée sur le nombre d'arêtes intra-cluster et inter-cluster. Partant d'une solution aléatoire initiale, RNSC dé-

place itérativement un sommet d'un cluster à un autre si ce déplacement réduit le coût général. Lorsqu'un nombre de coups a été atteint sans que la fonction de coût soit optimisée, le programme se termine. Étant donné, que le réseau d'interaction de protéines est représenté sous la forme d'un graphe tel que les sommets sont des gènes et les arêtes les interactions protéiques, cet algorithme peut découvrir des sous-graphes ou des clusters basés sur différentes propriétés topologiques telles que la densité, les k-cores (c'est un sous-graphe où chaque sommet est adjacent à au moins k autres sommets de ce sous-graphe. La valeur de k est fixée par le ou la chercheur-e en fonction de la densité du graphe étudié.), etc.

Nepusz et al, proposent un algorithme basé sur le partitionnement d'un réseau biologique nommé ClusterOne (Nepusz et al., 2012). Cet algorithme permet de partitionner facilement des graphes pondérés et de générer les clusters qui se chevauchent. Son principe est le suivant : fusionner des clusters qui se chevauchent fortement et supprimer tous les autres clusters de petites tailles. À chaque itération, il faut sélectionner un sommet comme noyau et l'étendre à travers les sommets voisins dans le but d'augmenter la densité du cluster.

Les algorithmes de détection de communautés supposent que le réseau lui-même se divise en partitions ou en communautés. Le but est alors de trouver ces communautés. Le partitionnement des réseaux biologiques est mieux servi par des algorithmes de détection de communautés. Comme il existe des cas où les algorithmes de détection de communauté adoptent des techniques à partir d'algorithmes de clustering de graphes, l'étude des algorithmes de clustering de graphes en soi est très fructueuse. La plupart de la recherche actuelle est axée sur la recherche d'une structure de la communauté avec une grande connectivité entre les sommets du même cluster et une faible connectivité entre les sommets des différents clusters. Cette stratégie a été utilisée dans le domaine de la biologie moléculaire pour obtenir des « modules indépendants » dans les réseaux d'interactions métaboliques ou protéine-protéine de la biologie moléculaire. Cependant, cette stratégie a ses propres limites, car dans certains cas, les sommets connectés peuvent être très différents.

D'après notre étude, nous constatons que la plupart des algorithmes de détection de communautés (ou clusters) de gènes permettent d'identifier un certain nombre de clusters à travers l'optimisation de propriétés particulières. De plus, ces algorithmes sont basés seulement sur des propriétés topologiques. En effet, les méthodes analytiques existantes souffrent d'une ou de plusieurs des limitations suivantes (He and Chan, 2018) :

1. Certaines approches exigent que l'utilisateur spécifie à l'avance le nombre de communautés à identifier. Or, en pratique, il est rarement possible pour l'utilisateur de fournir le nombre exact de communautés. Spécifier le nombre de communautés comme paramètre d'entrées d'un algorithme requiert des connaissances a priori sur le graphe sous investigation. En réalité, de telles connaissances ne sont pas toujours disponibles.
2. Les approches existantes (MCL, RNSC, etc) traitent les nœuds du réseau d'une manière similaire sans prendre en compte l'importance et la contribution de chaque nœud à la formation de sa communauté. En effet, une présence significative des nœuds discriminants (principalement les nœuds qui se connectent massivement aux autres nœuds du graphe et les nœuds avec des connexions éparses) affecte les caractéristiques topologiques du réseau sous investigation, ce qui rend la tâche de clustering plus difficile.

2.6.2 Approches évolutionnaires

Les algorithmes génétiques ont montré leur efficacité pour apporter des solutions satisfaisantes à des problèmes d'optimisation combinatoire réputés difficiles. Ils combinent typiquement une méthode évolutionnaire à base de population de solutions et une recherche locale servant à l'exploration de l'espace de recherche. La population initiale, généralement de taille fixe, est constituée d'individus qui sont des solutions du problème. À chaque génération, elle évolue par deux mécanismes :

— la production de nouvelles solutions par un opérateur génétique (le croisement ou la mutation)

— la sélection des nouvelles solutions qui prennent la place d'anciennes solutions dans la population pour améliorer la qualité générale de la population et préserver sa diversité.

Nous présentons dans cette partie les approches les plus connues basées sur les AGs pour détecter les communautés dans les réseaux complexes et spécifiquement les réseaux sociaux puisque la plupart des approches ont été appliqués sur ce type de réseau (Teng et al., 2019).
Rappelons que la détection de communautés consiste en un procédé commun s'appliquant à tout type de réseau, permettant le partitionnement de nœuds en communautés. Les algorithmes génétiques (Holland, 1975) ont été utilisés pour optimiser la modularité d'un réseau. La modularité est

une mesure pour la qualité d'un partitionnement des nœuds d'un graphe en communautés. Elle est principalement utilisée en analyse des réseaux sociaux. Elle a été introduite par M. E. J. Newman (Newman, 2004) et présenté dans la section 2.2. C'est aussi une fonction d'optimisation pour certaines tâches de détection de communautés dans les graphes. Elle est décrite comme la proportion des arêtes incidentes sur une classe donnée moins la valeur qu'aurait été cette même proportion si les arêtes étaient disposées au hasard entre les nœuds du graphe. L'utilisation de la modularité comme fonction objective à optimiser est une idée qui a été explorée par plusieurs chercheurs [(Newman and Girvan, 2004), (Tasgin and Bingol, 2006),(Liu et al., 2007)]. Le but est de trouver, parmi toutes les partitions possibles, celle qui possède la meilleure modularité. Une autre fonction de qualité a été également utilisée (Pizzuti, 2008). Elle consiste en la détermination d'une mesure globale de la qualité d'une division au sein des communautés appelée *score de la communauté*. Nous décrivons ci-après trois algorithmes génétiques pour détecter des communautés.

Dans (Tasgin and Bingol, 2006), Tasgin et Bingol proposent un AG dans lequel les partitions sont des individus (un individu = une partition) et la modularité est la fonction de fitness. Un choix approprié des paramètres de l'algorithme, comme le nombre d'individus et les taux de mutation et de croisement, a été adopté. Un tableau d'entiers est utilisé pour la représentation des données du problème de la détection de communautés. Le tableau stocke les identifiants communautaires des nœuds, c'est-à-dire à l'entrée i du tableau, on trouve l'identifiant de la communauté du nœud i. Le tableau à m éléments est pris comme individu pour l'AG. L'algorithme commence par la création d'une population initiale, en mettant dans la même communauté certains nœuds qui sont connectés dans le graphe. L'algorithme utilise un croisement, une mutation et un nouvel opérateur "clean-up" visant à réduire le nombre de nœuds mal placés dans des communautés. En effet, une communauté doit contenir plus de liens internes entre les nœuds à l'intérieur de la communauté que de liens externes avec d'autres communautés. Pour cette raison, les voisins d'un nœud doivent être la plupart du temps à l'intérieur d'une même communauté. Ainsi, les auteurs ont défini une métrique nommée *variance de la communauté* d'un nœud i qui représente le nombre de communautés différentes entre les voisins et le nœud lui-même. Cette valeur doit être faible pour une bonne structure de la communauté. L'opérateur clean-up analyse la variance de communauté de certains nœuds choisis au hasard. Si la variance de communauté du nœud choisi est supérieure à une valeur seuil constante, alors le nœud et tous ses voisins sont mis dans une même communauté. La nouvelle communauté

sera la communauté la plus fréquente chez les voisins, autrement dit la communauté qui contient le plus grand nombre de nœuds dans le voisinage du nœud sélectionné. Si la valeur seuil n'est pas dépassée, aucune opération n'est effectuée sur les identifiants des communautés.

Liu et al. présentent dans (Liu et al., 2007), un algorithme dans lequel la modularité maximale de la partition est obtenue par l'intermédiaire de bipartitions successives du graphe. Ainsi, chaque bi-partition est calculée en appliquant un algorithme génétique pour chaque sous-graphe (à partir du graphe d'origine lui-même). Chaque sous-graphe est considéré comme isolé du reste du graphe et une bipartition n'est acceptée que si elle augmente la modularité totale du graphe.

Fortunato et Barthemely (Fortunato and Barthélemy, 2007) ont montré que les algorithmes fondés sur l'optimisation de la modularité souffrent d'une limite de résolution. Ces derniers ne peuvent pas distinguer des communautés plus petites d'une certaine taille limite. D'autre part, les auteurs dans (Chen et al., 2014) ont montré que la maximisation de la modularité n'a pas seulement tendance à fusionner les petites communautés, mais aussi à éclater des grandes communautés, et il semble impossible d'éviter simultanément les deux problèmes.

Dans (Pizzuti, 2008), Pizzuti propose un algorithme génétique nommé GA-Net pour découvrir des communautés dans les réseaux sociaux. Certaines communautés présentes dans la structure du réseau sont obtenues à la fin de l'algorithme en procédant par une exploration sélective de l'espace de recherche, sans avoir besoin de connaître à l'avance le nombre exact de communautés. Au niveau de cette approche, deux nouvelles mesures sont introduites qui sont le score de la communauté et la notion d'individu sûr. La première mesure la qualité du partitionnement du réseau en communautés, l'algorithme cherche à optimiser cette mesure. Elle est calculée comme suit :

$$CS = \sum_{i=1}^{k} Q(S_i) \tag{2.4}$$

Avec :

— k : la taille d'une partition (= communauté)

— S : une sous matrice de la matrice d'adjacence de graphe de réseau

— Q(S) : score de la sous matrice S. Ce score est défini comme suit :

$$\begin{cases} QS = M(S) * V_S \\ V_S = \sum_{i \in I, j \in J} a_{ij} \; si \; a_{ij} = 1 \\ M(S) = \frac{\sum_{i \in I} (a_{ij})^r}{|I|} \end{cases} \tag{2.5}$$

Avec :

— I, J : les lignes et les colonnes de la sous matrice S. $S = a_{ij}$

— $|I|$: la taille d'une ligne de I.

L'utilisation des AGs pour la détection de communautés a été appliquée essentiellement dans le cadre des réseaux sociaux. À notre connaissance, au moment de l'écriture de ce manuscrit, une seule tentative de détection de communautés dans les réseaux PPI a été publiée (He and Chan, 2016). Cette approche, appelée graph clustering algorithm (**EGCPI**), applique un clustering de graphe évolutionnaire. La segmentation du graphe évolue en optimisant une mesure topologique nommée Independence of Cluster (IoC). Ainsi, dans l'AG la similarité sémantique n'est pas prise en compte lors de l'évolution. Les étapes de détection de communautés de gènes par *EGCPI* sont les suivantes :

1. Construire un graphe attribué (APPIG) des annotations de gènes obtenues à partir de GO. Le graphe APPIG est représenté par $G = (V, E, \Lambda)$, où V est l'ensemble des sommets, E représente l'ensemble des arcs et Λ est l'ensemble des annotations de chaque gène dans G dans toutes les sous ontologies de GO (BP, MF, CC).

2. Créer un graphe pondéré (wAPPIG). Ce graphe est construit en associant un poids à chaque arc du graphe en fonction du degré de similarité topologique entre la paire de gènes de l'arc.

3. Étant donné un wAPPIG, un individu (= solution) est représenté sous forme d'un vecteur d'arrangement avec pour chaque nœud le numéro de cluster associé.

4. Un algorithme évolutionnaire est appliqué pour identifier les clusters dans lesquels les gènes ou les protéines sont densément connectées en maximisant la densité au sein d'un même cluster. Cette mesure topologique est nommée Independence of Cluster (IoC) et définie comme

suit :

$$IoC_{ci} = \frac{\sum\limits_{v_{j,vk} \in Ci} W_{jk}}{\sum\limits_{v_{j \in Ci}} W_{jk}} \tag{2.6}$$

$$IoC_{wAPPIG} = \frac{\sum\limits_{i=1}^{S} n_{vi}}{n_v} IoC_{ci} \tag{2.7}$$

Avec :

- c_i : le ième cluster,
- w_{jk} : le poids attribué à l'interaction e_{jk} entre le gène j et le gène k.
- n_{vi} : le nombre total de sommets dans le cluster i.
- n_v : le nombre total de sommets du graphe wAPPIG.
- S : nombre de clusters.
- IoC_{ci} : représente le poids total des intra-interactions par rapport à toutes les interactions reliant les gènes du cluster c_i
- IoC_{wAPPIG} : mesure l'indépendance entre cluster c_i par rapport aux autres clusters de la population. Cette dernière permet de diminuer l'interdépendance entre deux clusters.

Après avoir obtenu un ensemble de clusters à partir du Graphe wAPPIG, EGCPI effectue une étape supplémentaire pour identifier les communautés de gènes. Ces communautés sont détectées en calculant le degré d'homogénéité entre chaque paire de gène du cluster c_i. Le degré d'homogénéité θ est défini par :

$$\theta_{ij} = \frac{\sum\limits_{k=p,c,f} \Lambda_k^{v_i} \cap \Lambda_k^{v_j}}{\sum\limits_{k=p,c,f} \Lambda_k^{v_i} \cup \Lambda_k^{v_j}} \tag{2.8}$$

Avec :

- k : représente les sous ontologies de GO : processus biologiques (p), des fonctions moléculaires (f) et des composants cellulaires (c),
- $\Lambda_k^{v_i}$, $\Lambda_k^{v_i}$: correspond aux annotations de gènes v_i et v_j selon k, respectivement.

Le degré d'homogénéité θ détermine la similarité entre chaque paire de gènes en se basant sur leurs annotations. Pour créer des communautés de gènes ayant une forte connexion, EGCPI applique la méthode de recherche en largeur (BFS) dans chaque cluster détecté. Tout d'abord, il commence par sélectionner les deux gènes (i et j) ayant un degré d'homogénéité θ le plus élevé. Ensuite, l'algorithme EGCPI détermine tous les voisins de ces deux gènes (i et j) et choisit ceux qui satisfont un seuil minimum prédéfini de θ.

Les approches existantes basées sur les AGs qui permettent de détecter des communautés sont basés sur des mesures topologiques telles que la modularité, la densité et la mesure proposée par Pizzuti. Les avantages d'utiliser de méthodes évolutionnaires pour la détection des communautés sont (Pizzuti, 2018) :

— Lors du processus de recherche, la taille et le nombre des communautés sont générés automatiquement ;

— des connaissances spécifiques au domaine peuvent être incorporées dans la méthode, à travers l'initialisation biaisée ou des opérateurs de variation spécifiques au lieu d'opérateurs aléatoires permettant une exploration plus efficace de l'espace de recherche des solutions possibles ;

— les implémentations de modèles basés sur la population permet de traiter des réseaux de grande taille cela dépend de la taille de réseau à traiter.

Toutes ces raisons nous ont conduit à proposer une approche originale basée sur les AGs pour la détection de communautés dans les réseaux de gènes.
La figure 2.6 récapitule la catégorisation des différentes approches de détection de communautés présentées dans cette thèse.

2.7 Conclusion

Nous avons consacré le premier volet de ce chapitre à l'introduction de certains concepts de base indispensables issus de la théorie des graphes afin de décrire comment les réseaux complexes sont modélisés par des graphes. Ensuite, nous avons présenté des définitions relatives à la notion de communauté et nous avons décrit le problème de la détection de communautés. Nous nous sommes

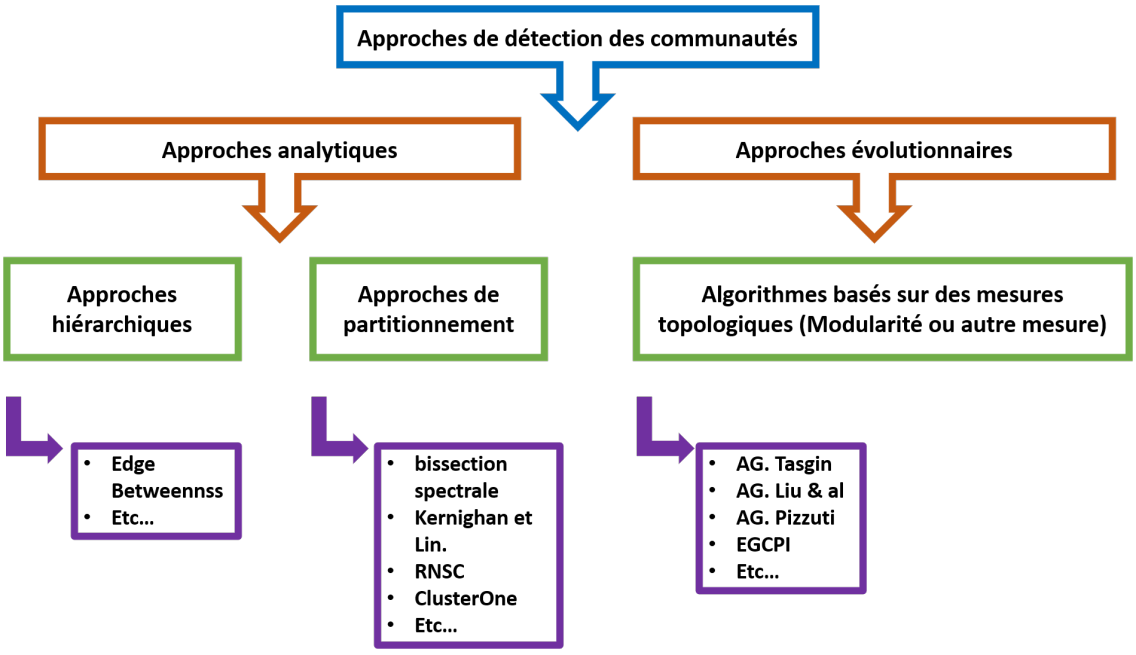


FIGURE 2.6 – Les approches de détection de communautés.

penchés dans le deuxième volet de ce chapitre sur une présentation des notions des AGs. Au niveau du dernier volet de ce chapitre, nous avons présenté un bref aperçu des travaux traitant la détection de communautés dans les réseaux complexes. Dans le chapitre suivant, nous présentons notre approche *GA-PPI-Net* basée sur les AGs pour la détection de communautés de gènes.

Chapitre 3

GA-PPI-Net : Approche proposée pour la détection de communautés de gènes

Sommaire

3.1 Introduction

Les algorithmes génétiques ont été très largement utilisés dans le domaine de l'optimisation et leur succès est dû en partie à la rapidité et la facilité d'implantation qu'ils permettent. Malgré la séduisante facilité du processus évolutionnaire, produire un algorithme évolutionnaire efficace est une tâche difficile. Les processus évolutionnaires sont très sensibles aux choix algorithmiques et paramétriques et notamment aux choix des représentations. L'expérience a montré que les succès sont fondés essentiellement sur une compréhension fine des mécanismes évolutionnaires et surtout sur une très bonne connaissance du problème à traiter. Une forte spécialisation de l'algorithme évolutionnaire est donc souhaitée pour aboutir à des résultats probants, cependant cette spécialisation va à l'encontre de la facilité d'implantation. Un compromis est donc préférable lors de la conception d'un AG. C'est sur ces points importants que nous nous appuyons pour concevoir une approche évolutionnaire dédiée à la détection de communautés dans les réseaux d'interaction de protéines humaine.

Par ailleurs, les méthodes de détection de communautés existantes (cf. chapitre précédent) considèrent essentiellement l'aspect topologique des graphes d'interaction (modularité). Notre objectif à travers notre approche évolutionnaire est de combiner les deux aspects topologique et sémantique de graphe d'interaction. L'approche construit les communautés d'une manière évolutive et ne nécessite pas une connaissance préalable du nombre de communautés dans le réseau.

Le présent chapitre est consacré à la description de l'approche *GA-PPI-Net* que nous avons proposé pour la détection de communautés de gènes de tailles différentes (Ben M'barek et al., 2019). Il est organisé comme suit, nous commençons tout d'abord par une définition formelle de notre problème de détection de communautés biologiques. Ensuite, nous détaillons le principe de notre approche. Les données utilisées dans cette approche sont décrites dans la section 1.5 du Chapitre 1.

3.2 Définition du problème de détection de communautés de gènes

Un réseau d'interaction Protéine-Protéine est représenté par un graphe d'interaction $G = (V, E)$ tel que V est un ensemble de nœuds et E est un ensemble d'arêtes entre paire de gènes.

Pour notre problème, nous avons défini une communauté (ou cluster) comme étant un groupe de noeuds ayant une densité élevée d'arêtes intra-groupe et une densité plus faible d'arêtes extra-groupe. Nous concevons une communauté **C** comme un groupe de gènes qui sont sémantiquement similaires et qui sont en interaction. Un ensemble de gènes $C = \{G_1, G_2, ..., G_n\}$ est une communauté s'il respecte les propriétés suivantes :

$$\forall\, G_i, G_j \in C,\ S(G_i, G_j) \geq \nabla_S \ ou\ I(G_i, G_j) \geq \nabla_I \tag{3.1}$$

Tel que :

— $S(G_i, G_j)$: est la similarité sémantique entre deux gènes G_i et G_j. Pour déterminer la valeur de similarité, nous avons utilisé la méthode GS2 (GO-based similarity of gene sets) (Ruths et al., 2009) (voir section 1.4.3).

— $I(G_i, G_j)$: le score d'interaction entre deux gènes

— ∇_S and ∇_I : sont deux seuils. Ils sont définis pour le critère sémantique et le critère d'interaction, respectivement. Leurs valeurs sont fixées selon les recommandations d'un expert en bioinformatique.

— Un gène est annoté par un ou plusieurs termes GO (Gene Ontology) (Camon et al., 2003). Nous proposons de noter *TP* un terme de GO qui annote un gène G. L'ensemble des termes GO qui annotent G est noté **A(G)**. Plus formellement,

$$A(G) = \{TP\ \ qui\ \ annote\ \ G \mid\ TP\ \ \in\ \ GO\} \tag{3.2}$$

Par exemple, le gène MEIKIN est annoté par :

$A(MEIKIN) = \{GO:0007060, GO:0010789, GO:0016321, GO:0045143, GO:0051754\}$.

3.3 Présentation générale de l'approche *GA-PPI-Net*

Les AGs sont probablement les algorithmes les plus connus et utilisés dans le calcul évolutionnaire. Ils ont été développés dans les années soixante par Holland (Holland, 1975). L'idée de base de son

système était d'étudier, dans le cadre de la psychologie/biologie, le processus d'adaptation des populations, en se basant sur des données sensorielles introduites au système grâce à des détecteurs binaires ((Holland, 1975), (Petrowski and Ben-Hamida, 2017)). Ils ont été appliqués à l'optimisation paramétrique pour la première fois par De Jong en 1975 (De Jong, 1975), qui a posé les fondements de cette technique d'application. Cependant, le manque de puissance des ordinateurs à l'époque ne permettait pas leur application sur des problèmes réels de grande taille. Ce n'est que pendant les années quatre-vingt-dix, précisément avec l'apparition de l'ouvrage de référence écrit par Goldberg (Goldberg, 1989), que les AGs se sont fait connaître dans la communauté scientifique. Nous proposons dans ce travail une nouvelle approche d'AG pour la détection de communautés, que nous appelons GA-PPI-Net. Cette approche est introduite dans (Ben M'barek et al., 2019), respecte le principe global de l'AG, mais elle possède des spécificités traduites par deux idées principales pendant la phase d'initialisation des individus et pendant l'évolution de la population par des opérateurs génétiques (croisement et mutation). En effet, la population est formée d'individus qui sont des solutions ou une partie de la solution du problème ("approche parisienne) (Collet et al., 2000), c'est-à-dire dans notre cas un ensemble de gènes qui forment une communauté de gènes dans le réseau PPI qui sont similaires et sont en interactions. Ainsi, différents éléments de la même communauté peuvent apparaître dans différents individus. Les solutions de la population sont initialisées d'une manière aléatoire (cf 3.3.2), avec une suppression des gènes redondants (aucun gène ne peut apparaître plus qu'une fois dans la même solution. Les ensembles de gènes sont ensuite évolués afin de reconstruire des communautés dans le réseau PPI, ce qui nous amène à la deuxième spécificité de notre algorithme. Dans l'objectif de trouver des communautés de gènes qui sont similaires et sont en interaction, *GA-PPI-Net* dispose de deux opérations génétiques (croisement et mutation). Selon l'hypothèse de l'approche parisienne, que des éléments de la même communauté peuvent apparaître dans différents individus de la population, le croisement tentera alors de grouper ces éléments dans une même solution. L'opérateur appliqué est le croisement 2-points qui permettra une meilleure exploration des solutions courantes et une bonne exploration des meilleures solutions, essentiellement à la fin de l'évolution. Par ailleurs, du fait qu'une petite partie de l'espace de recherche (réseau PPI), c'est l'opérateur de mutation qui permettra d'explorer et d'introduire de nouvelles composantes (nouveaux gènes) dans les communautés en cours de construction. *GA-PPI-Net* dispose d'un opérateur de mutation spécifique permet de restreindre l'in-

troduction de nouveaux gènes dans les solutions aux cas ou une amélioration de la performance est possible (cf. section 3.3.5.2). La fonction objectif qui guide le processus d'évolution de notre algorithme est basée sur le calcul de score d'un groupe de gènes en se basant sur la valeur de similarité et la valeur d'interaction de chaque paire de gène. Ce score est à maximiser. Le fonctionnement de *GA-PPI-Net* peut être résumé comme suit. Après la création de la population initiale de n individus, nous appliquons des opérations génétiques pour un certain nombre d'itérations max_iteration. Au cours de chaque itération, nous sélectionnons deux parents pour réaliser le croisement et obtenir un fils unique. Une méthode spécifique de mutation sera appliquée à ce fils avec une certaine probabilité. Ce nouvel individu fils sera évalué et remplacera le pire individu de la population courante. Afin de préserver les meilleures solutions rencontrées lors du processus évolutif, nous utilisons la stratégie "Steady State" (Goldberg, 1989). Avec ce type de remplacement, seul le(s) mauvaise(s) solution(s) sont remplacée(s) par les nouveaux descendants. Ainsi, à chaque itération après la phase de reproduction, la population est triée selon les valeurs de performances et les descendants remplacent les derniers dans le classement. À la dernière génération, la meilleure solution retournée par *GA-PPI-Net* est récupérée afin d'évaluer sa performance avec des outils biologiques. *GA-PPI-Net* est exécuté autant de fois que le nombre de communautés souhaitées (avec des populations initiales différentes). Un pseudo code de l'algorithme général de la solution proposée est présenté dans l'algorithme 2.

Rappelons que notre objectif consiste à déterminer des communautés de gènes de longueur variable à partir du réseau d'interaction de protéines humaine. Nous expliquons maintenant le principe de chaque étape de l'approche proposée.

3.3.1 Représentation d'un individu

Un individu représente une solution de notre problème, c'est une communauté. Il est représenté par un tableau T de dimension $n+3$ tel que n représente le nombre de gènes d'une communauté. Nous disposons des valeurs de similarité et des valeurs d'interaction de chaque paire de gènes. Le tableau T permet de stocker la taille de la communauté ou size (= le nombre de gènes formant cette communauté), la valeur moyenne de similarité (AVGSIM) et de valeur moyenne d'interaction (AVGInteraction) de chaque paire de gènes et enfin les noms des gènes.

L'avantage de cette représentation est qu'elle nous permet de faire évoluer des communautés de

Algorithme 2 Algorithme général de notre approche *GA-PPI-Net*

Entrée: P_m : probabilité d'appliquer la mutation, P_c : probabilité d'appliquer le croisement, max_iteration : nombre maximum d'itérations possibles ;
Sortie: BestIndividual // Meilleur individu issu du processus d'évolution
Début
1: $POP_0 \leftarrow$ Generate-initial-Population ;
2: Evaluer (POP_0) ; // Calcul de la fonction objectif
3: **pour** $i = 1$ to max_iteration **faire**
4: P1, P2 $\leftarrow$ SelectFrom (POP_i) ; // par tournoi
5: E1, E2 $\leftarrow$ Crossover (P1, P2, Pc) ;
6: $E1'$, $E2' \leftarrow$ Mutate(E1, E2, Pm) ;
7: Évaluer ($E1'$, $E2'$) ;
8: $POP_{i+1} \leftarrow$ Replace(POP_i, $E1'$, $E2'$) ; // Remplacer les mauvais individus par $E1'$ et $E2'$
9: **fin pour**
10: Retourner BestIndividualIn(POP_i) ;
Fin

taille variable. La figure 3.1 illustre la représentation d'un individu adoptée dans notre algorithme. La taille (size) est mise à jour après chaque opération de croisement ou mutation (étape 5 et 6 de l'algorithme 2). AVGSIM et AVGInteraction sont mis à jour par la fonction de performance (étape 7 de l'algorithme 2).

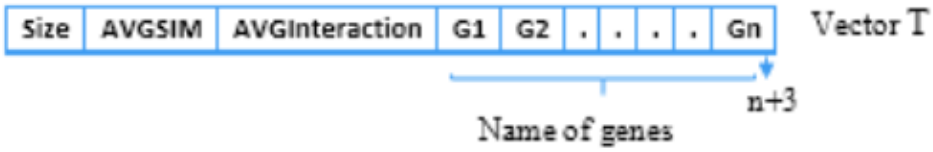


FIGURE 3.1 – Représentation d'un individu.

3.3.2 Initialisation de la population

La population est formée d'un ensemble d'"individus" qui sont différentes solutions du problème. Une population est donc un ensemble de communautés. De ce fait, après avoir récupéré aléatoirement des groupes de gènes de la base KEGG, nous commençons par la suppression des gènes redondants. Puis, nous créons notre population initiale par des gènes choisis aléatoirement de ces groupes. Ensuite, nous calculons pour chaque individu de la population initiale sa taille, la valeur de similarité moyenne et la valeur d'interaction moyenne définies dans le paragraphe suivant. La figure 3.2 schématise un exemple d'une population initiale. Les trois premières colonnes repré-

sentent respectivement la taille, la similarité et l'interaction. Les autres colonnes représentent les noms des gènes des différentes solutions.

5	0.553	0.398	PDHA2	MTHFD2L	RAC2	GRHPR	ANAPC1			
8	0.793	0.543	ANAPC5	SOS1	CDC16	AURKA	IL4	ANAPC2	ccNB2	BUB1B
5	0.340	0.410	RFK	HYI	GPI	UBC	IGF1R			
7	0.578	0.687	HSD3878	PFKP	LDHAL6B	FBXW7	ACSM3	MAX		
6	0.632	0.591	ALPP	BPGM	PLK1	HK3	HK1	KEAP1		

FIGURE 3.2 – Exemple d'une population initiale de cinq individus.

3.3.3 Fonction de fitness

La fonction de fitness ou d'évaluation permet d'évaluer la qualité d'un individu, c'est-à- dire, d'une communauté de gènes. Nous avons proposé une fonction qui est fondée sur le calcul de la valeur de similarité et la valeur d'interaction moyenne entre chaque paire de gènes de la communauté. La fitness d'une solution S est définie comme suit :

$$\begin{cases} F(S) = W_1 * AVGSIM + W_2 * AVGInteraction \\ W_1, W_2 \in [0,1] \end{cases} \tag{3.3}$$

Avec :

$$AVGSIM = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} SIM\_GS2(G_i, G_j)}{\frac{(n-1)*n}{2}} \tag{3.4}$$

$$AVGInteraction = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ValueInteraction(G_i, G_j)}{\frac{(n-1)*n}{2}} \tag{3.5}$$

Avec :

— n : désigne le nombre de gènes d'une communauté.

— G_i : un gène dans la communauté ($i \in 1...n$).

— ValueInteraction : représente la valeur d'interaction entre une paire de gènes d'une communauté dans le réseau PPI.

— SIM_GS2 : désigne la valeur de similarité d'une paire de gènes calculée en utilisant la méthode de calcul de similarité GS2

Des tests ont été réalisés dans (Ben M'barek et al., 2018) avec différentes valeurs des poids W_1 et W_2 et les valeurs choisies sont les valeurs qui donnent les meilleurs résultats en termes de nombre de réseaux connus obtenus, à savoir $W_1, W_2 = 0.5$.

3.3.4 Sélection

Une fois la fonction fitness définie, il faut choisir une méthode de sélection (première étape dans la boucle du notre algorithme *GA-PPI-Net*). Nous avons adopté la méthode de sélection par tournoi. Le principe de cette méthode est le suivant (Goldberg and Deb, 1991) : un groupe d'individu est choisi au hasard, ces individus sont comparés entre eux selon la fitness : le meilleur de ce groupe sera déclaré gagnant du tournoi. Ainsi, une paire d'individus est formée pour la reproduction. L'avantage de cette méthode est qu'elle donne une chance à un grand nombre d'individus d'être sélectionné pour la reproduction, permettant ainsi une meilleure exploitation des solutions existantes.

3.3.5 Les opérateurs génétiques

Les opérateurs génétiques utilisés classiquement dans les AGs (croisement et mutation) sont repris dans notre proposition de détection de communautés de taille variable.

3.3.5.1 Croisement

C'est la deuxième étape de la boucle de notre algorithme *GA-PPI-Net* (ligne 5, algorithme 2). Le croisement permet de simuler des reproductions d'individus dans le but d'en créer de nouveaux. Pour notre proposition, nous avons utilisé le croisement multipoint. Ce type de croisement est une généralisation du croisement à un point avec k points de coupure. Il s'agit alors de déterminer k points, puis d'échanger les blocs entre ces points afin d'obtenir de nouveaux enfants. Pour mettre en œuvre cet opérateur, nous devons tout d'abord sélectionner deux parents par la méthode de section par tournoi. Ensuite, des points de coupure sont choisis de façon aléatoire à partir desquels nous faisons une permutation entre les gènes de deux parents sélectionnés et une mise à jour de tailles

des deux descendants (E1, E2). Pour comprendre le type de croisement choisi, une illustration graphique est présentée dans la figure 3.3. Dans cet exemple, deux points de coupure sont choisis au hasard en position 1 et 4. Ensuite, deux descendants (E1, E2) sont générés en échangeant les valeurs des parents sélectionnés (P1,P2).

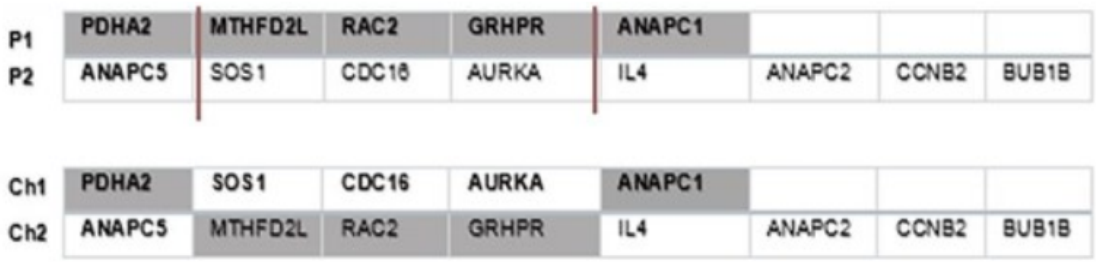


FIGURE 3.3 – Exemple de croisement 2- points.

3.3.5.2 Mutation

C'est la troisième étape de *GA-PPI-Net*. Cet opérateur consiste à altérer un gène dans un individu selon un facteur de mutation. Ce facteur est la probabilité qu'une mutation soit effectuée sur un individu. C'est l'application du principe de variation de la théorie de Darwin et permet d'éviter une convergence prématurée de l'algorithme vers un extremum local. Après avoir effectué le croisement, nous appliquons la mutation aux fils obtenus, avec une certaine probabilité P_m de valeur assez faible (qui ne dépasse pas $0,4$ pour notre algorithme). Nous avons défini un opérateur de mutation nommé Optimized Community Mutation (OCM1) adapté aux réseaux PPI. Le rôle de cet opérateur est de renforcer les scores d'interaction et de similarité sémantique au sein de la solution mutée. Pour cela, nous avons proposé une fonction score, notée *GS* appliquée à un gène, pour estimer sa qualité au sein de la communauté à laquelle il appartient. Ce score nous aidera à détecter le gène ayant le meilleur score dans une communauté, il est noté BestGene. L'idée est d'introduire dans la communauté un nouveau gène en interaction avec BestGene, ce nouveau gène remplacera le gène ayant le plus mauvais score nommé WorstGene. Le score *GS* est égal à la somme de la similarité moyenne et l'interaction moyenne d'un gène dans une communauté S. Il est calculé comme suit :

$$GS(S,G) = \frac{\sum_{i=1}^{n-1} SIM\_GS2(G,G_i)}{m} + \frac{\sum_{i=1}^{n} ValueInteraction(G,G_i)}{m} \tag{3.6}$$

Avec :

— $SIM\_GS2(G, G_i)$: la valeur de la similarité du gène G par rapport à un autre gène G_i de la communauté.

— $ValueInteraction(G, G_i)$: la valeur d'interaction du gène G par rapport à une autre gène G_i de la communauté.

— n : la taille de la solution S (communauté).

— m : le nombre de valeurs de similarité et d'interaction du gène G par rapport aux autres gènes de la même communauté

L'opération de mutation sur un individu S adopté dans notre solution est comme suit :

— Déterminer le gène ayant le plus grand score GS parmi les gènes de l'individu S (= candidat de la mutation), ce gène sera appelé BestGene.

— Chercher aléatoirement un gène qui interagit avec le BestGene dans la base des interactions PPI (qui n'apparait pas dans la communauté), ce gène sera appelé GenInt.

— Déterminer le gène ayant le plus faible score GS parmi les gènes de S, ce gène sera appelé WorstGene.

— Si le score de WorstGene est inférieur à 0.5 alors remplacer WorstGene par le gène GenInt dans la solution S. Sinon insérer le gène GenInt dans la solution S et incrémenter la taille.

Un pseudo code de la fonction mutation proposée est donné dans l'algorithme 3.

3.3.6 Spécificité et originalité de GA-PPI-Net

L'originalité de GA-PPI-Net réside dans l'introduction d'une solution spécifique pour représenter une communauté (=solution) de taille variable. Ainsi, nous nous avons proposé un opérateur de mutation optimisée (OCM1) pour une meilleure exploration du graphe. Le rôle de cet opérateur est de renforcer les scores d'interaction et de similarité sémantique au sein de la solution. Pour cela, nous avons proposé une fonction score, noté GS appliquée à un gène, pour estimer sa qualité au sein de la communauté à laquelle il appartient. Ce score nous aidera à détecter le gène ayant le meilleur score dans une communauté (BestGene). L'idée est d'introduire dans la communauté un nouveau gène en interaction avec BestGene pour remplacer le gène ayant le plus mauvais score. Par ailleurs, la

Algorithme 3 Algorithme de la fonction mutation OCM1

Entrée: P_m : probabilité d'appliquer la mutation, S : Enfant obtenu après la réalisation de croisement ;
Sortie: S' // individu enfant issu du processus de mutation.

Début

si random <Pm **alors**
  Scores ← ScoreGeneIndividu ; // Calcul des scores GS de chaque gène de S : somme de ses valeurs de similarité et ses valeurs d'interaction avec les autres gènes de l'individu.
  BestGene ← GetBestGene(Scores) ; // Récupérer le gène de S qui possède le score maximal.
  GenIn ← SelectGeneInteraction(BestGene) ;//Sélectionner aléatoirement un gène qui interagit avec le bestGene.
  WorstGene ← GetWorstGene(Scores) ; // Récupérer le gène qui possède le score minimal.
  si $scores(WorstGene) < 0.5$ **alors**
    S' ← Replace(S, WorstGene, GenIn) ; //Remplacer WorstGene par GenIn
  sinon
    S' ← insert (S, GenIn) ;
    updatesize(S') ; // Mettre à jour la taille de la solution mutée
  fin si
fin si
Return S' ;

End

spécificité de GA-PPI-Net réside essentiellement dans l'introduction d'une mesure communautaire pour évaluer, et par la suite détecter des communautés de bonne qualité. Cette mesure, exprimée dans la fonction fitness de l'AG, combine à la fois des informations topologiques et des informations sémantiques. Elle est basée sur la maximisation d'une mesure communautaire entre les gènes et tente de construire et de trouver la meilleure communauté en explorant sélectivement l'espace de recherche. Cette mesure est basée sur la similarité sémantique et l'interaction entre les gènes. Finalement, les communautés denses existant dans le réseau sont obtenues à la fin de l'évolution après une exploration sélective de l'espace de recherche, sans avoir besoin de connaître à l'avance la taille de la communauté (Ben M'barek et al., 2019).

Avec les méthodes évolutionnaires comme l'AG de clustering EGCPI et l'AG de Pizzuti chaque nœud est affecté à un vecteur d'arrangement. Ceci suppose que chaque gène appartient à une communauté. Et la représentation d'une solution exige le choix au préalable du nombre de clusters. Par contre, avec *GA-PPI-Net*, il n'est pas nécessaire de définir un nombre de clusters dans la représentation des solutions. De plus, *GA-PPI-Net* est moins couteux du fait qu'il fait évoluer des

communautés et non des vecteurs d'arrangement. Notre approche ne possède pas une étape supplémentaire pour identifier les communautés à partir de clusters détectés par l'AG comme EGCPI en calculant le degré d'homogénéité. *GA-PPI-Net* fait évoluer chaque communauté indépendamment du reste de graphe PPI. Seule la performance au sein d'une communauté est calculée, il n'y a pas besoin de calculer l'inter-connexion entre les clusters, ce qui réduit la complexité. Considérer qu'un individu (une solution) est une partition (clustering) de tout le graphe, comme dans l'AG de clustering EGCPI et l'AG de Pizutti, rend le passage à l'échelle difficile. Il faudrait représenter le graphe entier autant de fois qu'il y a d'individus dans la population, ce qui n'est pas envisageable. Avec notre choix de considérer qu'un individu correspond à une communauté, le passage à l'échelle devient envisageable, comme nous le montrons à travers notre proposition décrit dans le chapitre 5.

3.3.7 Étude Expérimentale

Dans cette section, nous présentons les tests effectués afin d'évaluer la performance de notre approche *GA-PPI-Net*. Ces tests visent à montrer l'efficacité de l'approche proposée à découvrir des communautés de gènes de différentes tailles. Les AGs ont prouvé leur efficacité en optimisation dans différents domaines. Cependant, cette efficacité est fortement liée à un choix judicieux des paramètres de l'algorithme. Afin de sélectionner la meilleure combinaison de ces paramètres à considérer dans la suite des expérimentations, plusieurs tests d'évaluation de ses paramètres ont été réalisés. Et nous avons choisi d'expérimenter notre approche GA-PPI-Net pour la combinaison des paramètres qui a donné les meilleurs résultats. Cette combinaison de paramètres est présentée dans le tableau 3.1.

Nous avons expérimenté notre approche *GA-PPI-Net* avec cinq jeux de données proposés par un expert en biologie. Ces jeux de données représentent des communautés de gènes réelles collectées de la base KEGG. Le tableau 3.2 décrit les jeux de données utilisés. L'évaluation dans le contexte *GA-PPI-Net* consiste à vérifier dans quelle mesure l'approche proposée est susceptible de retrouver des communautés de gènes existantes dans la base de voies biologiques KEGG Pathway. En effet, notre approche permet la détection soit des communautés qui existent dans cette base de référence

TABLE 3.1 – Paramètres de *GA-PPI-Net*.

Paramétres	Valeur
Taille de la population	30
Nombre de générations	100
Probabilité de croisement	0.8
Probabilité de mutation	0.01
Taille tournoi	10
Taille des individus de la population initiale	[0..40]

TABLE 3.2 – Jeux de données utilisés.

Datasets	Nombre de gènes
Apoptosis	88
B cell receptor signalling pathway	75
Purine metabolism	159
Rna degradation	159
Oocyte meiosis	114

(KEGG) soit des nouvelles communautés ayant une forte interaction et des valeurs de similarité assez élevées entre chaque paire de gènes qui les constituent, mais qui n'apparaissent pas dans KEGG.

Nous avons effectué des tests pour déterminer les communautés ayant des tailles différentes. Nous exécutons notre approche 20 fois avec tous les gènes de jeux de données présentés dans le tableau 3.1. La population initiale de chaque exécution est créée aléatoirement par des gènes de ces jeux de données. À la fin de chaque test, nous avons retenu à chaque fois la meilleure communauté. Donc, nous avons vingt meilleures communautés de tailles différentes variant de 5 à 40. Le tableau 3.3 illustre trois exemples de communautés détectées de tailles 8, 13 et 24. Les valeurs de la fitness varient entre 0.54 et 0.65.

Lorsqu'une solution est trouvée, la question qui se pose est celle de son évaluation. L'expert en biologie nous a proposé d'évaluer les solutions obtenues (les communautés) en vérifiant si elles existent dèjà dans KEGG Pathway ou dans d'autres bases de voies biologiques comme KEGG, Biocarta, Reactome, BBID et Ec Number.

Chaque communauté *R_new* détectée par notre approche est présentée à l'outil DAVID [1], qui com-

1. https ://david.ncifcrf.gov/ (Database for Annotation Visualization and Integrated Discovery) (Sherman et al., 2007)

TABLE 3.3 – Exemple de communautés détectées.

Communautés	Fitness	Taille
ADCY4, PPP2R1A, IL12RB2, CHUK, IL4, IKBKG, NFKB1, IL1B	0.69	8
NME4, ENTPD5, GUCY2D, FOS, NRGN, ADCY8, ADCY6, BAX, GSK3B, POLR2E, POLR2D, POLR2L, PKM2	0.54	13
PRKX, CPEB1, BIRC2, PKMYT1, CHP2, MAP2K1, PRKAR2B, IKBKG, AKAP10, CALM2, SKP1, YWHAQ, NRAS, PDE3B, MAPK3, PIK3R3, AURKA, PTTG1, MYD88, PIK3R2, PRKAR1B, RPS6KA2, BTRC, DDX6	0.65	24

pare cette communauté aux bases existantes et donne le pourcentage des gènes de *R_new* qui appartiennent à une même communauté dans les autres bases. DAVID est l'un des programmes d'annotation fonctionnelle les plus populaires auprès des biologistes et s'appuie sur une annotation fonctionnelle des gènes particulièrement riche (Sherman et al., 2007). Cet outil prend en entrée une liste de gènes et exploitent les annotations fonctionnelles disponibles sur ces gènes dans des bases de données publiques comme BIB, KEGG, BioCarta, etc., afin de trouver des fonctions communes et suffisamment spécifiques à ces gènes.

Notre premier objectif pour la validation est de voir si on arrive à reconstruire des communautés connues. Pour cela, nous avons choisi de partir de voies biologiques répertoriées dans la base de données KEGG. Notre AG prend en entrée les gènes de jeux de données présentés dans le tableau 3.2 en éliminant les gènes redondants. Donc, les premiers tests effectués permettent d'évaluer la capacité de notre approche à reconstruire des communautés avec les gènes initiaux de jeux de données (les 5 communautés de la table 3.2). Nous avons testé avec l'outil DAVID toutes les communautés obtenues pour déterminer le taux de recouvrement (pourcentage min et max) de communautés obtenues par rapport aux communautés de jeux de données. Le schéma expérimental pour la validation de *GA-PPI-Net* à reconstruire les communautés KEGG de jeux de données 3.2 est présentée dans la Figure 3.4.

Le tableau 3.4 représente les résultats de ces tests.

Les résultats présentés dans le tableau 3.4 montrent que les communautés obtenues par notre approche *GA-PPI-Net* correspondent à des "parties" de communautés réelles (pourcentage <100%)

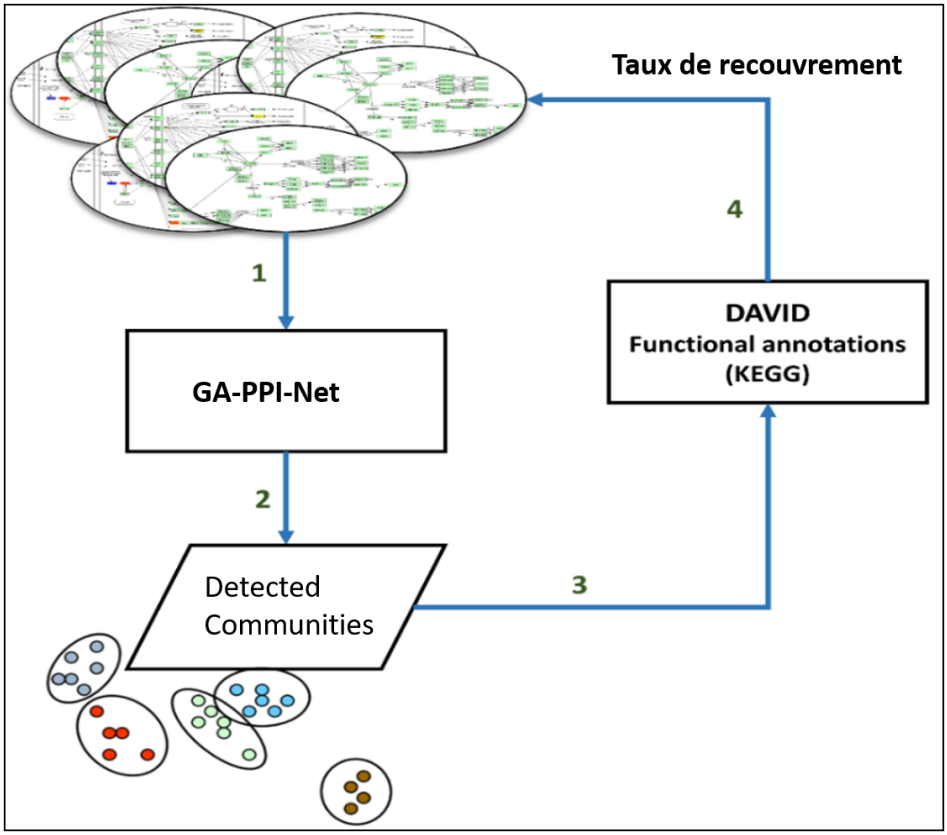


FIGURE 3.4 – Schéma expérimental pour la validation de *GA-PPI-Net* de reconstitution des communautés de départ. (1) Nous partons d'une liste de gènes qui appartiennent à des voies biologiques connues à partir de la base de données KEGG. (2) Nous appliquons notre approche sur cette liste de gènes (indépendamment de leur classement initial). (3) Le résultat obtenu est annoté par l'outil DAVID. Nous avons utilisé les API de DAVID pour automatiser le processus et avons paramétré la requête pour n'avoir que les communautés KEGG utliées dans les jeux de données de la table 3.2. (4) Pour l'évaluation de l'outil, nous comparons les résultats obtenus aux communautés KEGG de départ.

TABLE 3.4 – Évaluation des communautés détectées par *GA-PPI-Net* par rapport à la base KEGG (base de départ).

Jeux de données	% Minimum	% Maximum
Apoptosis	25%	67%
B cell receptor signaling pathway	25%	67%
Purine metabolism	50%	100%
RNA degradation	-	29%
Oocyte meiosis	50%	100%

et dans certains cas à des communautés réelles complètes (pourcentage = 100%). Par conséquent, *GA-PPI-Net* est capable de reconstruire des communautés en utilisant les gènes de jeux de données

d'entrée. Ces résultats sont considérés comme satisfaisants par notre expert, même avec un taux de recouvrement relativement faible (29%) pour RNA degradation. L'expert estime que l'obtention d'un taux de recouvrement supérieur à 50% est considéré comme satisfaisant.

Nous avons évalué également les nouvelles communautés obtenues en vérifiant si elles existent dans d'autres bases de voies biologiques que KEGG. De même, toutes les communautés obtenues sont testées avec l'outil DAVID pour déterminer les pourcentages minimum et maximum qui représentent le taux d'appartenance de communautés obtenues par rapport aux autres communautés de bases de voies biologiques. Les bases que nous avons utilisées pour évaluer nos résultats sont : Biocarta, Reactome , BBID et Ec Number. Les résultats de ces tests sont présentés dans le tableau 3.5.

TABLE 3.5 – Évaluation des communautés détectées par *GA-PPI-Net* par rapport aux bases de voies biologiques.

Bases de voies biologiques	% Minimum	% Maximum
BBid	25%	50%
Biocarta	20%	66.66%
Ec_number	15.38%	100%
Reactome	15.38%	100%
KEGG	15.38%	100%

D'après ce tableau, on peut remarquer que les communautés obtenues par notre approche *GA-PPI-Net* correspondent à des "parties" de réseaux réels existant dans d'autres bases de données de voies biologiques et, dans certains cas, des communautés réelles (pourcentage =100%). Ces résultats sont considérés comme très satisfaisants. Ils constituent une première validation de notre algorithme et montrent la pertinence de la fonction fitness proposée ainsi que les opérateurs génétiques de GA-PPI-Net.

3.4 Étude comparative 1 : *GA-PPI-Net* Vs approches analytiques

Dans le chapitre 2 section 2.6, nous avons introduit différentes approches existantes pour la détection de communautés dans les réseaux complexes. Ces approches se classent en deux catégories : les

approches analytiques et les approches évolutionnaires. Notre approche *GA-PPI-Net*, faisant partie des approches évolutionnaires, il n'est pas possible de comparer son fonctionnement avec celui des approches analytiques. Nous proposons toutefois de comparer sa performance en termes de qualité des communautés détectées avec une sélection de méthodes analytiques appliquées aux réseaux d'interaction de protéines ou gènes. Cette section présente une étude expérimentale comparative de notre approche par rapport aux approches analytiques MCL (Markov Clustering) (Van Dongen, 2000), RNSC (Restricted Neighborhood Search Clustering) (King et al., 2004) et ClusterOne (Nepusz et al., 2012). Nous avons mené cette étude dans un cadre expérimental homogène pour toutes les méthodes. Ainsi, ces expérimentations utilisent le même jeu de données et le même protocole expérimental. Cette étude est une évaluation quantitative et qualitative de la capacité de trois méthodes analytiques de clustering et de *GA-PPI-Net* pour la détection des communautés dans les réseaux d'interaction de protéines.

3.4.1 Matériels et méthodes

Nous avons choisi de comparer les performances *GA-PPI-Net* en termes de qualité des communautés détectées avec les trois méthodes de clustering : MCL (Van Dongen, 2000), RNSC (King et al., 2004) et ClusterOne (Nepusz et al., 2012). Nous avons choisi ces 3 méthodes de clustering selon les critères suivants :

— Obtention des meilleures performances dans le domaine de la biologie.

— Disponibilité de leurs implémentations en ligne via la plateforme (CDAP) `http://www.eslahchilab.ir/sofwares/cdap` (Wu et al., 2020).

3.4.1.1 Protocole expérimental

L'algorithme *GA-PPI-Net* a été évalué avec les trois méthodes de clustering (MCL (Van Dongen, 2000), RNSC (King et al., 2004) et ClusterOne (Nepusz et al., 2012)) en utilisant les cinq jeux de données présentés dans le tableau 3.2 de la section 3.3.7 du chapitre 3. Ces ensembles de données sont collectés à partir de la base de données des voies biologiques KEGG et correspondent à des communautés réelles existantes. Ils représentent un extrait du réseau d'interaction de protéines chez l'être humain.

Le protocole expérimental utilisé est le suivant :

— Afin d'évaluer *GA-PPI-Net*, nous sélectionnons, tout d'abord, des gènes aléatoirement à partir de jeux de données collectés de la base KEGG. Ensuite, nous exécutons *GA-PPI-Net* 20 fois et nous retenons à chaque fois la meilleure communauté détectée. Donc, nous avons 20 meilleures communautés avec des tailles variant de 5 à 40.

— Pour évaluer les méthodes de clustering (MCL, RNSC et ClusterOne), nous avons utilisé la plateforme en ligne CDAP en incorporant les cinq jeux de données collectés de la base KEGG qui contiennent des communautés réelles. CDAP est un package en ligne qui regroupe différentes méthodes de clustering appliqué sur les réseaux d'interaction de protéines. Il est accessible via le lien `http://www.eslahchilab.ir/sofwares/cdap`. Ensuite, nous faisons les tests en se basant sur les paramètres de chaque méthode proposés par Brohée dans (Brohee and Van Helden, 2006). Puis, nous filtrons les communautés identifiées afin d'obtenir des communautés ayant une taille supérieure à cinq. Nous avons obtenu respectivement 23, 21 et 28 communautés pour chaque méthode (MCL, RNSC et ClusterOne). Finalement, nous choisissons aléatoirement 20 clusters de différentes tailles de chaque méthode afin de les évaluer et de les comparer avec *GA-PPI-Net*.

3.4.1.2 Mesures d'évaluation

Nous avons proposé d'évaluer la performance quantitative et qualitative de différentes méthodes en utilisant les mesures suivantes :

1. Taux de recouvrement : désigne le taux recouvrement de communautés obtenues par rapport aux communautés réelles existantes dans les bases de voies biologiques tel que KEGG, BBID, Reactome, etc. Il est déterminé en utilisant l'outil DAVID (Jiao et al., 2012).
2. Similarité sémantique moyenne : présente la similarité sémantique moyenne de chaque paire de gènes appartenant à une même communauté. Sa valeur est déterminée en utilisant la mesure similarité GS2 (GO-based similarity of gene sets) (Ruths et al., 2009).
3. Interaction moyenne : c'est le score d'interaction moyen de chaque paire de gènes appartenant à une même communauté. Cette valeur est déterminée à partir de la base d'interaction de protéines STRING (Snel et al., 2000).

3.4.2 Étude expérimentale

Une évaluation et une comparaison de performances du *GA-PPI-Net* et des trois méthodes de clustering (MCL, RNSC et ClusterOne) en utilisant des jeux de données réelles est présentée dans cette section. Ces jeux de données sont collectés de la base KEGG et ils incluent : (i) Apoptose (soit 88 gènes), (ii) Signalisation des récepteurs des cellules B (soit 75 gènes), (iii) Métabolisme des purines (soit 159 gènes), (iv) Rna dégradation (soit 159 gènes) et (v) Méiose des ovocytes (soit 114 gènes).

Pour tester les performances des trois méthodes de clustering choisies et de *GA-PPI-Net* en ce qui concerne la qualité structurelle, fonctionnelle et la pertinence biologique des communautés résultantes, nous évaluons ces communautés en vérifiant si elles existent dans des bases de voies biologiques comme KEGG, Biocarta, etc en utilisant l'outil DAVID (Jiao et al., 2012). Ce dernier compare les communautés obtenues par chaque méthode avec d'autres existantes et réelles et donne le pourcentage de recouvrement dans ces communautés existantes. Le tableau 3.6 présente le pourcentage minimum et le pourcentage maximum de ce taux de recouvrement.

Comme les autres tests réalisés par DAVID, les résultats du tableau 3.6 montrent que les communautés obtenues correspondent à des « parties » ou à des communautés complètes (pourcentage =100%) de communautés réelles existant dans des bases de données de voies biologiques connues. Nous constatons que les différentes méthodes de clustering (MCL, ClusterOne et RNSC) et GA-PPI-Net ont pu reconstruire efficacement de réelles communautés existantes dans des bases de voies biologiques. L'approche *GA-PPI-Net*, ainsi que les trois approches de clustering, atteignent le pourcentage le plus élevé de 100% dans trois bases de données : Reactome, KEGG et Ec Number. Par exemple, GA-PPI-Net atteint la valeur de pourcentage max 100% dans la base Reactome. Le pire pourcentage est de 15,38%, ce qui correspond à une "partie" de communauté réelle. Néanmoins,*GA-PPI-Net* a atteint la meilleure valeur du pourcentage Min par rapport à MCL, RNSC et ClusterOne. Des tests d'évaluation seront également effectués sur des jeux de données de grande échelle pour valider notre approche.

Pour étudier la qualité fonctionnelle et structurelle des différentes communautés obtenues, nous avons calculé la similarité sémantique et la valeur d'interaction. Pour déterminer la similarité et la valeur d'interaction entre les gènes, nous avons utilisé la méthode GS2 (GObased similarity of gene sets) (Ruths et al., 2009) et la base de données STRING, respectivement. Dans cette étude, la

TABLE 3.6 – Taux de recouvrement Min et Max de communautés obtenues.

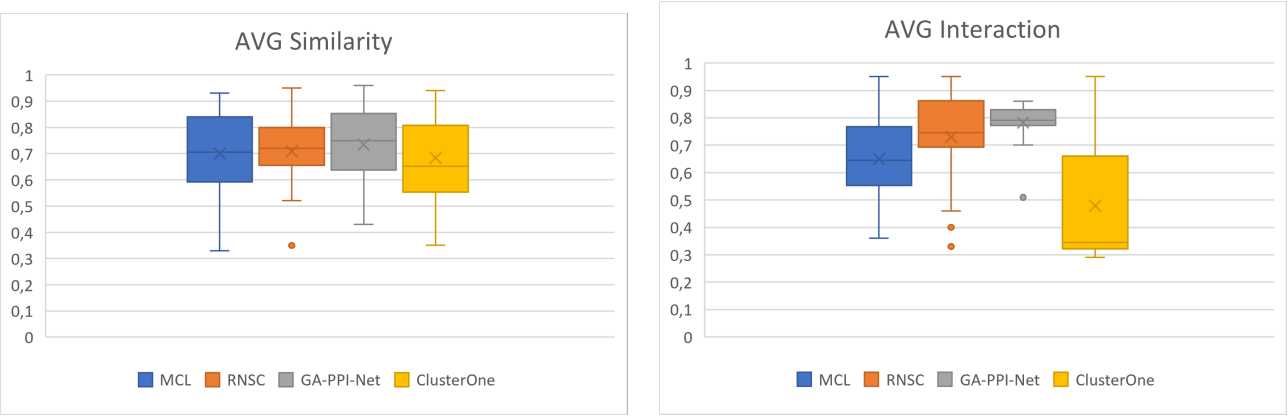
Méthodes	Bases de voies biologiques	Pourcentage Min	Pourcentage Max
MCL	BBid	9.09%	62.5%
	Biocarta	9.09%	87,5%
	Ec Number	9.09%	**100%**
	Reactome	6,81%	**100%**
	KEGG	**10.52%**	**100%**
RNSC	BBid	13,33%	80%
	Biocarta	4,28%	**100%**
	Ec Number	**34,28%**	**100%**
	Reactome	13,33%	**100%**
	KEGG	13,33%	**100%**
ClusterOne	BBid	8,33%	35.71%
	Biocarta	7,14%	80%
	Ec Number	**30%**%	**100%**
	Reactome	8,33%	**100%**
	KEGG	5,55%	**100%**
GA-PPI-Net	BBid	**25%**	50%
	Biocarta	20%	66.66%
	Ec Number	15.38%	**100%**
	Reactome	15.38%	**100%**
	KEGG	15.38%	**100%**

similarité sémantique et l'interaction sont résumés respectivement par la similarité moyenne de la valeur d'interaction moyenne des paires de gènes des communautés obtenues. Les résultats de ces tests sont décrits dans le tableau 3.7. Les résultats sont synthétisés à travers un graphique à boîte à moustaches présenté par la figure 3.5.

D'après le tableau 3.7, nous constatons que *GA-PPI-Net* atteint la meilleure valeur de similarité moyenne par rapport aux trois méthodes de clustering pour une taille de communauté 24 gènes. La figure 3.5 montre que *GA-PPI-Net* permet de détecter des communautés de bonne qualité par rapport aux autres méthodes de clustering selon les valeurs médianes de la similarité moyenne et l'interaction moyenne. La figure 3.5(a) montre clairement que *GA-PPI-Net* est capable de détecter des communautés de bonne qualité en termes de similarité sémantique. Cela est décrit dans la boîte à moustache qui contient 50% avec la valeur médiane qui varie entre 0.75 et 0.86. La borne inférieure du dernier quartile avec les méthodes de clustering est d'environ 0.83 donnée par la mé-

TABLE 3.7 – Similarité, interaction et taille des communautés : méthodes analytiques Vs *GA-PPI-Net*.

Méthode	Taille	Similarité	Interaction	Méthode	Taille	Similarité	Interaction
	5	0.67	0.75		5	0.76	0.33
	5	0.68	0.9		5	0.79	0.34
	5	0.72	0.57		7	0.54	0.41
	5	0.77	0.46		7	0.8	0.32
	5	0.82	0.94		8	0.92	0.9
	5	0.95	0.84		9	0.81	0.95
	6	0.81	0.33		10	0.53	0.7
	6	0.9	0.4		11	0.85	0.66
	7	0.72	0.74		11	0.94	0.4
RNSC	8	0.52	0.89	ClusterOne	12	0.55	0.48
	9	0.57	0.81		13	0.6	0.34
	9	0.63	0.78		13	0.9	0.87
	9	0.65	0.73		13	0.62	0.34
	9	0.74	0.73		14	0.72	0.66
	9	0.81	0.95		22	0.61	0.3
	12	0.69	0.72		22	0.56	0.3
	13	0.76	0.83		23	0.35	0.31
	14	0.35	0.71		25	0.625	0.32
	15	0.68	0.69		30	0.68	0.35
	16	0.73	0.76		38	0.527	0.29
	5	0.77	0.46		6	0.66	0.8
	6	0.74	0.73		6	0.86	0.86
	6	0.74	0.51		6	0.83	0.86
	7	0.54	0.58		6	0.93	0.83
	8	0.63	0.86		7	0.74	0.84
	8	0.92	0.66		7	0.56	0.77
	9	0.93	0.68		7	0.79	0.7
	9	0.81	0.95		7	0.63	0.78
	10	0.53	0.78		7	0.83	0.74
MCL	11	0.55	0.71	*GA-PPI-Net*	7	0.77	0.84
	11	0.66	0.63		8	0.43	0.79
	11	0.85	0.78		8	0.68	0.79
	13	0.6	0.55		8	0.86	0.8
	13	0.9	0.56		9	0.76	0.79
	14	0.78	0.36		10	0.71	0.51
	14	0.85	0.32		11	0.89	0.83
	19	0.67	0.66		12	0.66	0.78
	20	0.59	0.63		13	0.58	0.78
	21	0.33	0.56		14	0.56	0.79
	44	0.61	0.49		24	0.96	0.77



(a) AVG Similarity (b) AVG Interaction

FIGURE 3.5 – Synthèse de valeurs de similarité et d'interaction moyennes pour les méthodes MCL, RNSC, ClusterOne et *GA-PPI-Net*.

thode MLC. D'après la figure 3.5(b), nous constatons que *GA-PPI-Net* n'atteint pas les valeurs les plus élevées concernant la valeur de l'interaction moyenne. Cependant, cet algorithme possède la meilleure performance médiane et globale. La plupart des communautés détectées par *GA-PPI-Net* ont des valeurs d'interaction moyenne qui varient entre 0.7 et 0.86. Alors que, la méthode ClusterOne ne permet pas de détecter des communautés de bonne performance en terme d'interaction.

GA-PPI-Net permet de détecter des communautés en maximisant la fitness proposée, qu'est basée sur une mesure sémantique (la valeur similarité moyenne) et une mesure topologique (la valeur de l'interaction moyenne). Cette fitness n'est pas lié à la densité des communautés. Notre approche *GA-PPI-Net* permet d'explorer l'espace de recherche afin de détecter la meilleure communauté, tandis que les méthodes de clustering (MCL, RNSC et ClusterOne) permettent de partitionner le réseau en utilisant une mesure de distance.

Pour conclure, les résultats des analyses effectuées montrent la capacité et la performance de *GA-PPI-Net* à traiter le problème de détection de communautés dans les réseaux d'interaction de protéines. En termes de taux de recouvrement, les résultats obtenus par *GA-PPI-Net* et par les approches analytiques sont globalement similaires, même si les plus mauvaises communautés obtenues restent meilleurs pour *GA-PPI-Net* (taux de recouvrement min tableau 3.6). En termes de similarité moyenne, *GA-PPI-Net* obtient les meilleurs résultats selon la valeur médiane et la valeur max. Enfin, en termes d'interaction moyenne, *GA-PPI-Net* n'atteint pas les valeurs max, mais sa médiane est meilleure que les autres méthodes analytiques, de même que les valeurs de 50% de

communautés obtenues (répartition très concentrée autour de la médiane). Il s'avère que l'approche évolutive peut être complémentaire à l'approche analytique. En effet, *GA-PPI-Net* analyse le réseau de différentes manières et il est capable de détecter des communautés qui n'ont pas été identifiées par les méthodes analytiques. Il serait intéressant d'appliquer les deux approches en parallèle pour une meilleure analyse du réseau d'interaction de protéines.

3.5 Étude comparative 2 : GA-PPI-Net Vs EGCPI

Dans le chapitre 2, nous avons présenté la méthode de clustering évolutionnaire EGCPI (He and Chan, 2016). Au moment de l'écriture de cette thèse, EGCPI est la seule approche évolutionnaire appliqué pour la détection de communautés dans les réseaux PPI.

Dans cette section, nous proposons une étude comparative de l'algorithme *GA-PPI-Net* avec la méthode EGCPI en utilisant le jeu de données Collins. Ce dernier a été téléchargé de la plateforme de BioGRID database [2]. Le dataset Collins concerne les gènes de l'espèce yeast Saccharomyces cerevisiae (levure). Il est composé de 1620 gènes et 9064 interactions.

Vu que le code source de l'approche EGCPI n'est pas disponible, nous avons choisi d'expérimenter de notre approche avec les résultats disponibles en ligne [3]. Ces résultats sont obtenus en utilisant Collins dataset. Pour pouvoir faire notre étude comparative, nous avons du fait un prétraitement sur les données collins pour déterminer la similarité sémantique en utilisant la méthode GS2 et récupérer les valeurs d'interaction de la base String.

Le protocole expérimental utilisé est le suivant :

— Exécuter *GA-PPI-Net* 20 fois en utilisant le dataset Collins et nous retenons à chaque fois la meilleure communauté détectée. Notre AG est exécuté en utilisant les mêmes paramètres décrits dans (He and Chan, 2016) à savoir taille de population = 100, $PC = 0.6$ et nombre de générations = 30.

— Récupérer à partir des données disponibles en ligne tous les clusters obtenues par EGCPI. Puis, nous filtrons les communautés identifiées afin d'obtenir des communautés ayant une taille supérieure à cinq.

2. https ://thebiogrid.org/
3. https ://github.com/he-tiantian/EGCPI.

— Choisir aléatoirement 20 communautés à partir de données disponibles en ligne de EGCPI ayant la même taille que les communautés détectées par GA-PPI-Net.

Les performances des deux méthodes GA-PPI-Net et EGCPI en ce qui concerne la qualité structurelle, fonctionnelle et la pertinence biologique des communautés obtenues sont évaluées en vérifiant si elles existent dans des bases de voies biologiques multi-espèces comme KEGG, Reactome et Ec Number. Cette évaluation est effectuée en utilisant l'outil DAVID (Jiao et al., 2012). Le tableau 3.8 décrit le pourcentage minimum et le pourcentage maximum de taux de recouvrement de chaque approche.

TABLE 3.8 – Taux de recouvrement Min et Max de communautés obtenues (données Collins).

Méthodes	BDs biologiques	% Min	% Max
GA-PPI-Net	Ec Number	2.4%	25.0%
	KEGG	12.5%	100%
	Reactome	5.3%	100%
EGCPI (phase 1 : clustering)	Ec Number	2.5%	28.6%
	KEGG	5.1%	100%
	Reactome	6.9%	100%
EGCPI (phase 2 : détection de communautés)	Ec Number	-	-
	KEGG	-	100%
	Reactome	30%	100%

D'après le tableau 3.8, nous constatons que *GA-PPI-Net* et la méthode de clustering évolutionnaire EGCPI atteignent le meilleur taux de recouvrement (100%) pour les deux bases Reactome et Kegg. Ce tableau montre que les deux méthodes permettent de détecter de réelles communautés existantes dans Reactome et Kegg et des sous parties de communautés de la Base Ec Number avec un pourcentage 25% et 28.6% respectivement pour la méthode GA-PPI-Net et EGCPI. De plus, nous avons proposé de comparer les deux communautés finales détectées par la phase supplémentaire d'EGCPI avec celles détectées par notre approche. Nous constatons d'apres le tableau 3.8 (ligne EGCPI (phase 2 : détection communauté)) que les deux approches ont des résultats similaires avec un taux de recouvrement 100% pour les deux bases Reactome et Kegg. Les deux approches conduisent donc à des résultats similaires, cela pourrait s'expliquer d'une part par leur recours commun aux AGs et d'autre part par la prise en compte par chacune de ces approches de l'aspect

topologique. Il est à noter que l'approche EGCPI ne prend pas en compte l'aspect sémantique lors de l'évolution de l'AG. Cet aspect est utilisé comme une étape supplémentaire pour déterminer les communautés finales.

3.6 Conclusion

Dans ce chapitre, nous avons présenté notre approche de détection des communautés de différentes tailles à partir d'un réseau d'interaction de gène en se basant sur les AGs. Cette approche nommée *GA-PPI-Net* a permis d'obtenir des résultats très satisfaisants, qui montrent la pertinence de l'approche et en particulier de la fonction fitness proposée. Ensuite, nous avons présenté une étude comparative de l'approche évolutionnaire *GA-PPI-Net* qui permet de détecter des communautés de gènes de différentes tailles, par rapport à quelques approches analytiques MCL, RNSC et ClusterOne ainsi que par rapport à une méthode de clustering evolutionnaire EGCPI. Le chapitre suivant sera consacré à la présentation des extensions et des améliorations de notre algorithme *GA-PPI-Net*.

Chapitre 4

GA-PPI-Net : Extensions et améliorations

Sommaire

4.1 Introduction

Dans le chapitre précédent, nous avons présenté notre algorithme *GA-PPI-Net*. Cet algorithme permet de détecter des communautés de tailles variables, en optimisant la similarité et l'interaction entre les gènes au sein de la même communauté. Nous proposons d'étendre et d'améliorer GA-PPI-Net de la façon suivante :

— une généralisation de l'approche *GA-PPI-Net* permettant de détecter trois types de communautés : i) groupes de gènes basés sur l'interaction, ii) groupes de gènes basés sur la similarité et iii) communautés de gènes similaires et qui sont en interaction.

— une amélioration de l'approche *GA-PPI-Net* qui consiste à proposer un opérateur de mutation adaptatif. Cet opérateur favorise l'exploration de l'espace de recherche et la détection de communautés de bonne qualité.

4.2 Extension et amélioration 1 : *Generic GA-PPI-Net*

Rappelons que notre objectif vise à rendre notre approche *GA-PPI-Net* générique pour construire des communautés de gènes qui sont sémantiquement similaires *et/ou* sont en interaction.
Nous avons introduit trois nouvelles composantes spécifiques à l'AG par rapport à *GA-PPI-Net* :

— une solution adéquate et simplifiée pour représenter une communauté de taille dynamique ;

— une fonction de fitness basée sur une mesure de similarité et une valeur d'interaction entre gènes en fonction des valeurs de seuils. Cette nouvelle fonction que nous proposons réduit la complexité de calcul par rapport à celle proposée dans *GA-PPI-Net* ;

— un opérateur de croisement heuristique pour renforcer les liens dans les communautés.

L'extension de *GA-PPI-Net* proposée a été paramétrée en fonction de l'importance affectée à chaque critère de mesure (mesure sémantique et mesure d'interaction). Les communautés denses existant dans le réseau sont obtenues à la fin de l'évolution en explorant sélectivement l'espace de recherche, toujours comme dans *GA-PPI-Net*, sans avoir besoin de connaître à l'avance la taille de la communauté. Cette généralisation permet de détecter trois types de communautés :

— groupes de gènes basés sur l'interaction,

— groupes de gènes basés sur la similarité,

— communautés de gènes similaires et qui sont en interaction.

La complexité de calcul de la nouvelle fitness est largement réduite par rapport à celle de *GA-PPI-Net*. En effet, il y'a plus besoin de calculer les matrices de similarité sémantique moyen et d'interaction moyen. D'où, le temps de calcul de la fitness est optimisé.

Dans les sections suivantes, nous décrivons en détails les améliorations apportées à l'algorithme 2.

4.2.1 Représentation d'un individu

Comme introduit dans la section 3.3.1 du chapitre 3, une solution de notre problème est représenté par un tableau T. Ce tableau permet de stocker la taille de la communauté (= le nombre de gènes formant cette communauté) et les noms de gènes. Cependant, la similarité et l'interaction moyennes de l'individu ont été supprimées de la représentation d'une solution. En effet, la fonction de performance ayant été modifiée avec une formule générique, les valeurs correspondantes sont enregistrées dans un tableau dynamique dépendant des valeurs de seuils ∇S et ∇I. La figure 4.1 illustre la représentation d'un individu adoptée dans cette extension.

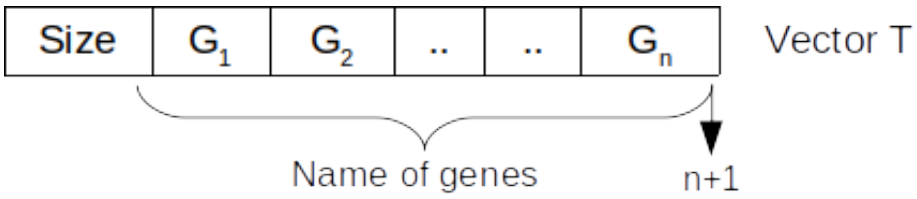


FIGURE 4.1 – Représentation d'un individu propre à l'algorithme *Generic GA-PPI-Net*.

4.2.2 Initialisation de la population

La population est un ensemble d'individus représentant différentes solutions du problème. Le process de l'initialisation suit les étapes suivantes :

1. Récupérer aléatoirement des communautés de gènes stockées dans la base de données KEGG (Kanehisa and Goto, 2000) ;
2. À partir des communautés générées, créer une population initiale composée par des gènes choisis aléatoirement ;

3. Calculer la taille (= le nombre de gènes formant cette communauté) de chaque individu de la population.

La figure 4.2 décrit un exemple d'une population initiale composée par cinq individus de taille variable. Les lignes représentent les différentes solutions. Pour chaque solution, la première colonne décrit sa taille tandis que les autres représentent les noms de ses gènes.

5	PDHA2	MTHFD2L	RAC2	GRHPR	ANAPC1			
8	ANAPC5	SOS1	CDC16	AURKA	IL4	ANAPC2	CCNB2	BUB1B
5	RFK	HYI	GPI	UBC	UGFAR			
7	HSD3B7	PFKP	LDHAL6B	FBXW7	ACSM3	MAX	IL5	
6	ALPP	BPGM	PLK1	HK3	HK1	KEAP1		

FIGURE 4.2 – Exemple d'initialisation de la population relatif à l'algorithme *Generic GA-PPI-Net*.

4.2.3 Fonction de fitness

La fonction fitness évalue la capacité d'un individu à survivre et à se reproduire. Elle prend en entrée une solution candidate et produit en sortie une mesure de sa performance. Dans le contexte du problème de la détection de communautés, la fonction la plus populaire décrivant la propriété topologique est la modularité, introduite par Girvan et Newman (Girvan and Newman, 2002). Nous ne prenons pas directement en compte la modularité. Néanmoins, la propriété topologique d'une communauté est prise en compte à travers la valeur d'interaction entre les gènes. En outre, cette fonction est enrichie par la similarité sémantique. Nous définissons, pour chaque paire de gènes dans une communauté S, une mesure générique M toujours basée sur deux propriétés : i) la similarité et ii) l'interaction. La performance $F1$ d'une solution S est alors définie comme suit :

$$F1(S) = \sum_{i \neq j, i=1, j=1}^{n} M(G_i, G_j) \tag{4.1}$$

$$M_{ij}(G_i, G_j) = \begin{cases} 0 \text{ if } S(G_i, G_j) \leq \nabla S \text{ ou } I(G_i, G_j) \leq \nabla I. \\ S(G_i, G_j) + I(G_i, G_j) \qquad \text{sinon.} \end{cases} \tag{4.2}$$

Avec :

— n : désigne le nombre de gènes de la communauté S.

— G_i : un gène dans la communauté.

— $S(G_i, G_j)$ et $I(G_i, G_j)$: désignent la valeur de la similarité (définit en utilisant la mesure de similarité sémantique GS2) et la valeur d'interaction entre une paire de gènes (G_i, G_j), respectivement.

— ∇S : est un seuil relatif à la propriété sémantique.

— ∇I : est un seuil relatif à la propriété topologique.

— Les valeurs de ∇S et ∇I sont fixées au début de l'évolution de l'AG.

Cette fonction fitness est une généralisation de la fitness de l'algorithme *GA-PPI-Net* décrite dans l'équation 3.3. Cette dernière est basée sur le calcul de la valeur de similarité moyenne et de la valeur d'interaction moyenne d'une paire de gènes existants dans la même communauté. La nouvelle fitness est toujours basée sur une mesure de similarité et une valeur d'interaction entre gènes, mais elle affine la prise en compte de ces valeurs d'une part en les considérant au niveau de chaque paire de gènes, et d'autre part en introduisant des seuils permettant de contrôler les niveaux minimums de similarité et d'interaction. Il est à noter que les valeurs de similarité et d'interaction appartiennent tous les deux à l'intervalle $[0, 1]$.

La densité du graphe de la communauté S est modifié en imposant les seuils. En effet, certains liens d'interaction ou de similarité ne sont plus pris en compte si leurs valeurs ne respectent pas les conditions de seuils. La nouvelle fitness permet à *Generic GA-PPI-Net* de construire des communautés non seulement basées sur la densité des arcs, mais des communautés basées sur les valeurs d'interaction et de similarité sémantique.

4.2.4 Croisement

Le croisement consiste à simuler des reproductions d'individus dans le but d'en créer des nouveaux (= les descendants). Nous avons proposé d'appliquer de façon aléatoire deux types de croisement :

— **Croisement 2-points**

Cet opérateur est déjà utilisé dans *GA-PPI-Net*. Son principe est décrit dans la section 3.3.5.1 du chapitre 3. Pour cette extension, nous avons proposé d'ajouter une étape "Clean-up" qui

permet de supprimer les gènes redondants dans chaque descendant obtenu. Cette étape est appliquée aux descendants obtenus.

— **Croisement heuristique**

Nous proposons un opérateur de croisement spécifique à notre problème. L'objectif de cet opérateur est de créer des descendants qui sont sémantiquement similaires ou sont en interaction. L'algorithme 4 décrit le principe de cet opérateur spécifique.

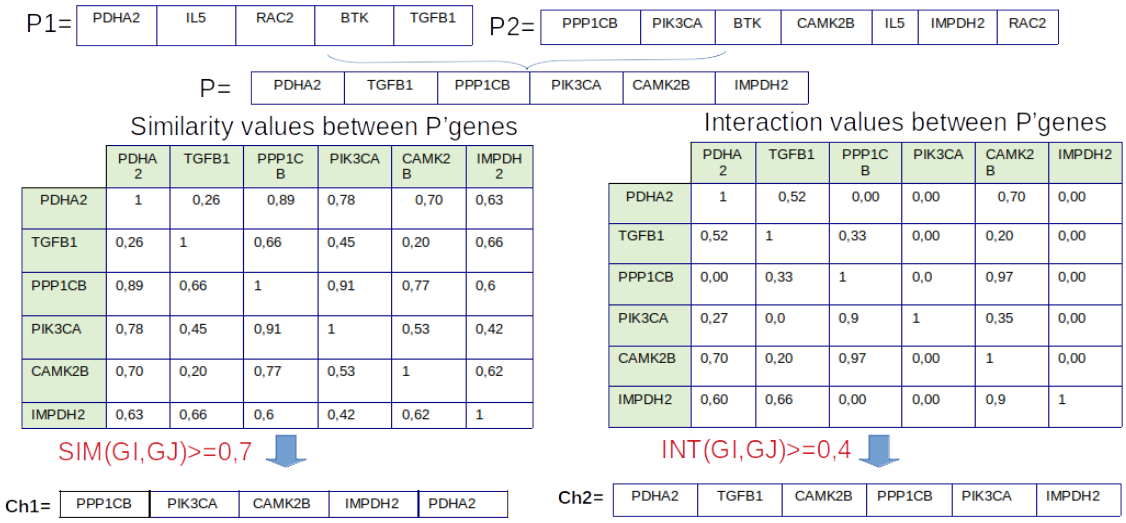
Algorithme 4 Algorithme de l'opérateur de croisement heuristique

Choisir aléatoirement deux parents (P1,P2) de la population parentale ;
Fusionner le contenu de deux parents (P1,P2), en supprimant les gènes redondants, afin d'obtenir un parent P.
pour chaque deux gènes G_i and $G_j \in P$ **faire**
  si $i \neq j$ **alors**
    déterminer la similarité : Sim(G_i,G_j) ;
    déterminer l'interaction : Interaction(G_i,G_j) ;
  fin si
fin pour
pour chaque deux gènes G_i and $G_j \in P$ **faire**
  si Sim(G_i,G_j) $\geq \nabla S$ **alors**
    Insérer les gènes dans le premier descendant Ch1 ;
  fin si
  si Interaction $(G_i, G_j) \geq \nabla I$ **alors**
    Insérer les gènes dans le deuxième descendant Ch2 ;
  fin si
fin pour
Supprimer les gènes redondants de Ch1 et Ch2 ;
retourner (Ch1, Ch2)

L'application de cet opérateur heuristique aide l'AG à détecter des communautés ayant des valeurs assez élevées de similarité et d'interaction entre les gènes selon les valeurs des seuils ∇S et ∇I . La figure 4.3 présente une illustration graphique pour comprendre l'opérateur de croisement proposé. Deux parents (P1,P2) sont choisis aléatoirement parmi la population parentale. Ensuite, les symboles(=contenus) des deux parents (P1,P2) sont fusionnés en supprimant les redondants pour obtenir un individu P. Puis, deux descendants (Ch1,Ch2) sont créés selon les conditions suivantes :

1. $\forall i \neq j : G_i, G_j \in P$, *si* $Sim(G_i, G_j) \geq \nabla S$ alors ajouter le gène G_i au premier descendant Ch1.

2. $\forall i \neq j : G_i, G_j \in P$, *si* $Interaction(G_i, G_j) \geq \nabla I$ alors ajouter le gène G_i au descendant Ch2.



	PDHA2	TGFB1	PPP1CB	PIK3CA	CAMK2B	IMPDH2
PDHA2	1	0,26	0,89	0,78	0,70	0,63
TGFB1	0,26	1	0,66	0,45	0,20	0,66
PPP1CB	0,89	0,66	1	0,91	0,77	0,6
PIK3CA	0,78	0,45	0,91	1	0,53	0,42
CAMK2B	0,70	0,20	0,77	0,53	1	0,62
IMPDH2	0,63	0,66	0,6	0,42	0,62	1

	PDHA2	TGFB1	PPP1CB	PIK3CA	CAMK2B	IMPDH2
PDHA2	1	0,52	0,00	0,00	0,70	0,00
TGFB1	0,52	1	0,33	0,00	0,20	0,00
PPP1CB	0,00	0,33	1	0,0	0,97	0,00
PIK3CA	0,27	0,0	0,9	1	0,35	0,00
CAMK2B	0,70	0,20	0,97	0,00	1	0,00
IMPDH2	0,60	0,66	0,00	0,00	0,9	1

FIGURE 4.3 – Exemple d'application de l'opérateur de croisement heuristique.

Ce nouvel opérateur permet de renforcer les liens dans les communautés par la prise en compte de meilleurs gènes au sens de leur haut niveau de similarité sémantique ou d'interaction.

4.2.5 Étude Expérimentale

Nous avons mené une étude expérimentale pour évaluer l'efficacité de l'extension proposée *Generic GA-PPI-Net* à construire et à détecter différents types de communautés de gènes qui sont sémantiquement similaires et/ou sont en interaction et ayant une taille variable. Nous avons choisi la combinaison de paramètres d'AG offrant les meilleurs résultats. Les paramètres utilisés sont présentés dans le tableau 4.1.

Les valeurs des seuils ∇S et ∇I ont comme rôle d'obtenir des communautés de qualité. ∇S est relatif à la valeur de la similarité sémantique entre une paire de gènes (G_i, G_j). Si $\nabla S \geq 0.5$ alors les gènes (G_i, G_j) sont similaires (Ben M'barek et al., 2018). ∇I fait référence à la valeur d'interaction entre les gènes (G_i, G_j). Si $\nabla I \geq 0.4$ alors les gènes (G_i, G_j) ont une forte interaction. Les valeurs des ∇S et ∇I sont proposées par un expert en biologie et sont modifiées selon nos besoins.

La première évaluation consiste à vérifier si l'extension proposée est susceptible de trouver des communautés de gènes ayant une forte similarité et/ou une forte interaction. Les communautés obtenues sont analysées et comparées selon ∇S et ∇I. Nous avons effectué des tests avec différentes

TABLE 4.1 – Paramètres de *Generic GA-PPI-Net*.

Paramétres	Valeur
Taille de la population	30
Nombre de génération	100
Probabilité de croisement	0.8
Probabilité de mutation	0.01
Taille tournoi	10
Taille des individus de la population initiale	[5..40]
∇S	[0.5..0.9]
∇I	[0.4..0.9]

valeurs de ∇S et ∇I et nous avons constaté que cette extension permet de détecter trois types de communauté :

1. $\nabla S \geq 0.5$ *et* $\nabla I < 0.4$: un groupe de gènes basé sur la similarité (*GS*).
2. $\nabla S \geq 0.5$ *et* $\nabla I \geq 0.4$: une communauté de gènes basée sur la similarité et l'interaction (*CSI*).
3. $\nabla S < 0.5$ *et* $\nabla I \geq 0.4$: un groupe de gènes basé sur l'interaction (*GI*).

Pour déterminer ces trois types de communautés, nous avons réalisé différents tests. Nous avons testé l'extension *Generic GA-PPI-Net* avec des gènes choisis aléatoirement parmi les cinq jeux de données présentés dans le tableau 3.2 du chapitre 3. De plus, nous avons varié les valeurs des seuils ∇S et ∇I et nous retenons à chaque fois la meilleure communauté détectée.
Pour bien illustrer notre proposition, nous définissons un exemple pour chaque type avec variation des deux seuils :

1. Dans le cas de (*GS*) : $\nabla S = 0.6$ *et* $\nabla I = 0.3$. Nous remarquons que *Generic GA-PPI-Net* détecte des groupes de gènes sémantiquement similaires. L'interaction dans ce cas n'est pas pris en compte. Un exemple de ce type de groupe est affiché dans la figure 4.4.
2. Dans le cas de (*CSI*) : $\nabla S = 0.6$ *et* $\nabla I = 0.4$. Nous constatons que *Generic GA-PPI-Net* détecte des communautés de gènes sémantiquement similaires et en interaction. Un exemple de communauté obtenue est exposé dans la figure 4.5.
3. Dans le cas de (*GI*) : $\nabla S = 0.3$ *et* $\nabla I = 0.4$. Nous constatons que *Generic GA-PPI-Net* détecte des groupes de gènes en interaction. La similarité sémantique dans ce cas n'est pas pris en compte. Un exemple de ce type de groupe est illustré dans la figure 4.6.

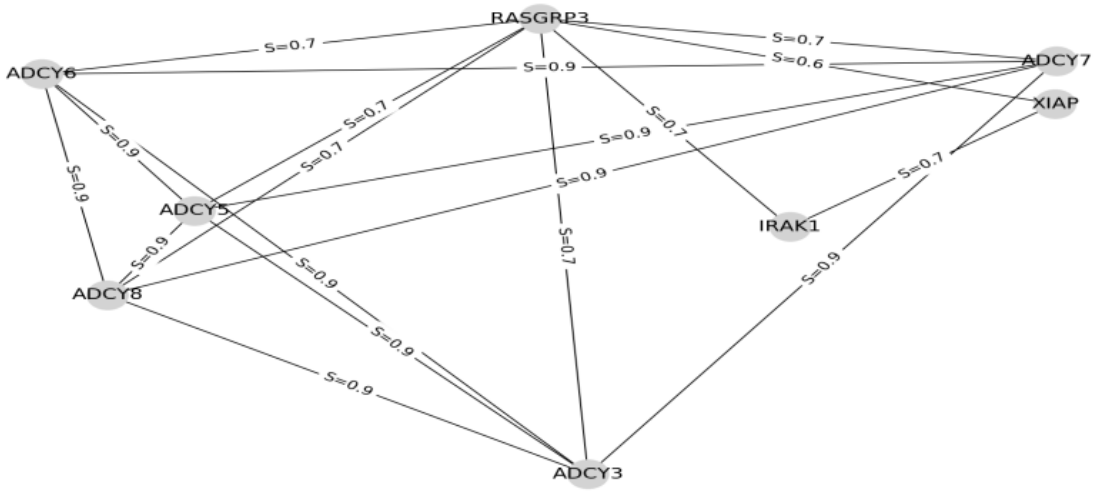


FIGURE 4.4 – Exemple d'une communauté de gènes basée sur la similarité ($\nabla S = 0.6$ *et* $\nabla I = 0.3$).

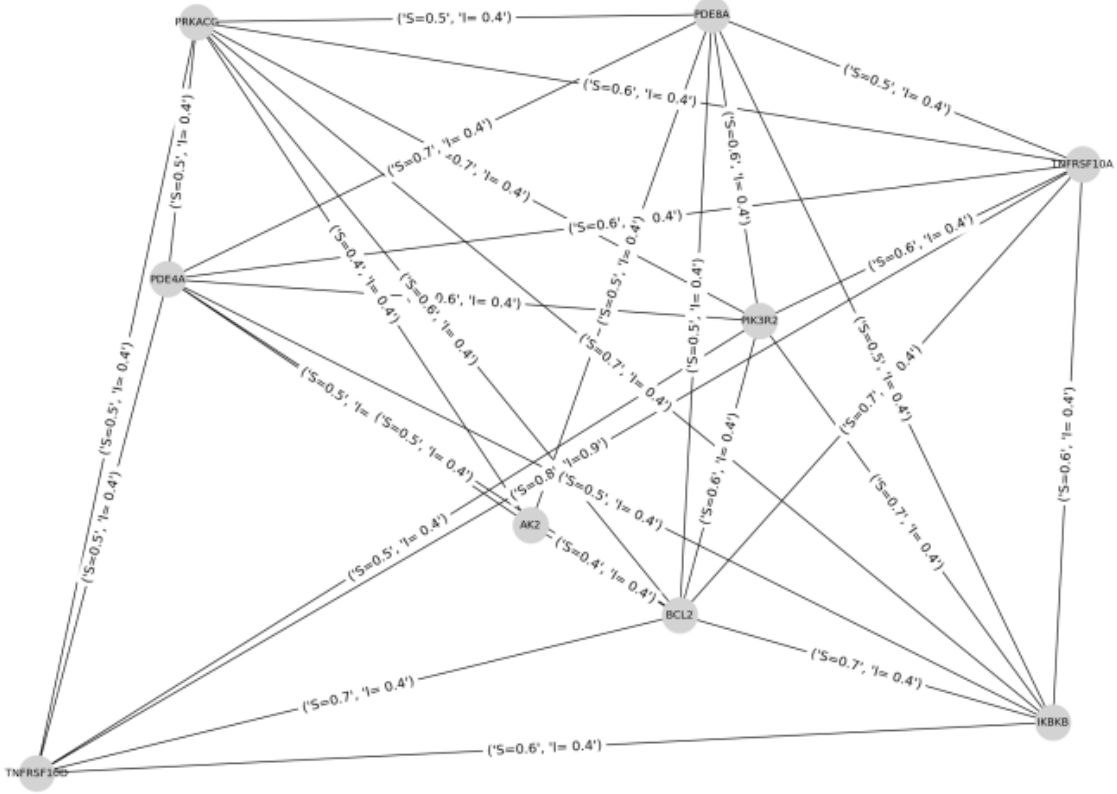


FIGURE 4.5 – Exemple d'une communauté de gènes détecté basée sur la similarité et l'interaction ($\nabla S = 0.6$ *et* $\nabla I = 0.4$.

La qualité des communautés obtenues est déterminée pour chaque type en suivant le protocole expérimental proposé par l'expert en biologie utilisé avec *GA-PPI-Net*. Nous exécutons *Generic GA-PPI-Net* 20 fois, pour chaque type de communautés, avec des gènes choisis aléatoirement

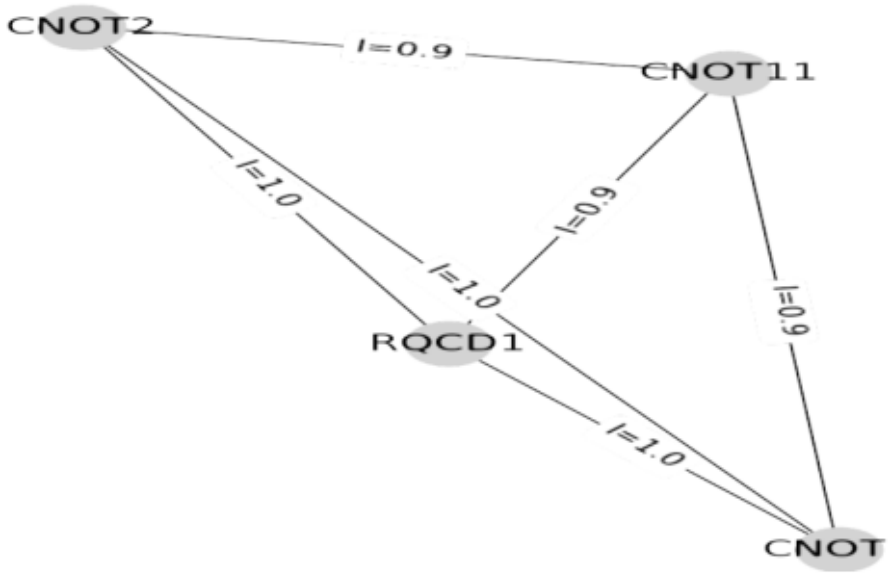


FIGURE 4.6 – Exemple d'une communauté de gènes détéctée basée sur l'interaction $\nabla S = 0.3$ *et* $\nabla I = 0.4$.

parmi les cinq jeux de données présentés dans le tableau 3.2. Nous avons retenu à chaque fois la meilleure communauté. Donc, nous avons 20 meilleures communautés de tailles variables, pour chaque type. Cette évaluation consiste à vérifier si les communautés *R_new* obtenues existent dans des communautés réelles de bases de voies biologiques en utilisant DAVID. Les résultats de ces tests expérimentaux sont présentés dans le tableau 4.2.

TABLE 4.2 – Évaluation de communautés détectéeés. Comparaison avec GA-PPI-Net.

DBs	*Generic GA-PPI-Net*						*GA-PPI-Net*	
	SIM groups		INT groups		SIM & INT communities			
	$\nabla S >= 0.5, \nabla I < 0.4$		$\nabla S > 0.5, \nabla I >= 0.4$		$\nabla S >= 0.5, \nabla I >= 0.4$			
	%Min	%Max	%Min	%Max	%Min	%Max	%Min	%Max
BBIB	20%	45%	15%	50%	20%	90%	25%	50%
Biocarta	10%	70%	20 %	50%	20%	100%	20%	66%
EC Number	10%	90%	10%	60%	30%	100%	10%	100%
KEGG	9%	78%	10%	62%	11%	100%	15%	100%
Reactome	20%	75%	15%	100%	10%	80%	14%	100%

L'approche *Generic GA-PPI-Net* est configurée selon l'importance affectée à chaque seuil (∇S et ∇I). Cette approche permet de détecter des communautés selon l'un des deux seuils ou bien selon la combinaison entre les deux.

Les résultats présentés dans le tableau 4.2 (colonne *Generic GA-PPI-Net*) montrent que les nouvelles communautés obtenues par notre algorithme correspondent à certaines « Parties » (pourcentage < 100%) de communautés réelles et dans certains cas à une communauté complète (pourcentage = 100%) dans les bases de voies biologiques. *Generic GA-PPI-Net* atteint les pourcentages 80%, 90 % et 100% (pour % Max avec les cinq bases) lorsque la fonction de fitness est basée à la fois sur des valeurs de similarité et d'interaction. Et, il atteint le pourcentage 90% et 100% lorsque la fonction fitness est basée sur la similarité sémantique ou sur l'interaction, respectivement, mais cette fois avec EC Number et Reactome uniquement. Ces résultats sont considérés comme très satisfaisants. Ils constituent une première validation de notre algorithme et montrent la pertinence de la fonction fitness.

Afin de valider les performances de cette extension, nous effectuons une comparaison des résultats obtenus avec l'approche *GA-PPI-Net*. Une comparaison approfondie n'est pas facile, car les communautés obtenues par les deux approches sont de tailles et de structures distinctes. De ce fait, nous avons proposé le protocole expérimental suivant :

— Utiliser les mêmes datasets présentés dans le tableau 3.2 et les mêmes paramètres pour les deux approches (la table 4.1).

— Exécuter chaque approche vingt fois.

— Déterminer le taux de recouvrement des gènes de chaque communauté obtenue par rapport aux autres communautés de voies biologiques en utilisant l'outil DAVID.

Dans le tableau 4.2, les données présentées dans l'avant-dernière colonne (*Generic GA-PPI-Net, SIM Int communities*) sont similaires ou meilleurs que celles présentées dans la dernière colonne (*GA-PPI-Net*). *Generic GA-PPI-Net* atteint des pourcentages de recouvrement plus élevés que *GA-PPI-Net* suivant les deux bases Biocarta et BBIB et le même taux de 100% suivant les deux bases KEGG et EC Number lorsque la fonction fitness est basée sur la similarité et l'interaction. Toutefois, *GA-PPI-Net* possède le pourcentage maximal suivant la base Reactome. De plus, *Generic GA-PPI-Net* conserve un pourcentage satisfaisant suivant la base Reactome pour les trois types de communautés détectées (75% pour les groupes basés sur la similarité, 100% pour les groupes basés sur l'interaction et 80% pour les communautés basées sur la similarité et l'interaction).

Certaines communautés dans les bases de voies biologiques considérées dans nos études ont une mesure de similarité élevée comme Ec Number , KEGG, Reactome et Biocarta. Ceci est démontré

par le taux de recouvrement qui a atteint 90% pour EC Number et 78% pour KEGG en se basant essentiellement sur la similarité ($\nabla S >= 0.5$). En gardant ce même seuil tout en maximisant en parallèle l'interaction entre les gènes en fixant $\nabla I = 0.4$) la capacité de recouvrement de *Generic GA-PPI-Net* a été largement améliorée par rapport à *GA-PPI-Net*. En effet, le taux de recouvrement maximal de la voie Biocarta passe de 66% à 100% et celui de BBIB passe de 50% à 90%. Cependant, le taux de recouvrement peut baisser si on augmente le seuil de similarité alors que les communautés de la voie biologique sont constituées essentiellement de gènes en forte interaction, comme c'est le cas pour la voie Reactome. Il est donc préférable d'effectuer des tests de *Generic GA-PPI-Net* avec différents seuils ∇S et ∇I afin de garantir une performance de détection satisfaisante.

4.2.6 Conclusion sur *Generic GA-PPI-Net*

Generic GA-PPI-Net permet grâce à la nouvelle fonction fitness d'avoir une approche plus fine de la "ressemblance" entre les gènes d'une même communauté. D'une part, cette ressemblance se fait au niveau de chaque paire de gène et non pas à travers une mesure moyenne comme dans *GA-PPI-Net*. D'autre part, l'introduction de seuils permet de contrôler le type de ressemblance à privilégier : i) sémantique à travers ∇S ou ii) d'interaction à travers ∇I.

Par ailleurs, la possibilité de choisir la valeur des seuils offre aux biologistes un outil simple pour détecter des communautés de nature différentes. Ainsi, l'opérateur de croisement heuristique proposé renforce la distribution entre ressemblance sémantique ou par interaction. Il permet en effet de garder des paires de gènes proches sémantiquement ou en forte interaction.

Les tests expérimentaux réalisés sont très satisfaisants et sont à compléter de façon plus approfondie avec une évaluation faisant intervenir un expert en biologie.

4.3 Extension et amélioration 2 : *Adaptative GA-PPI-Net*

GA-PPI-Net explore le réseau PPI essentiellement grâce à l'opérateur de mutation proposé OCM1. Ce dernier remplace un gène dans une communauté candidate en introduisant un nouveau gène en interaction avec le centre de la communauté (gène ayant le plus grand score qui est basé sur l'interaction et la similarité sémantique) en se basant sur un seuil à définir. Cependant, avec l'évolution

des communautés au long des générations, cet opérateur peut exclure des gènes qui font réellement partie de la communauté et la fitness peut se dégrader au lieu de s'améliorer. Nous proposons d'améliorer *GA-PPI-Net* en introduisant un opérateur de mutation adaptative nommé Optimized Community Mutation (OCM2). Cet opérateur permet d'améliorer la qualité d'une solution en remplaçant les gènes ayant un faible score ou en élargissant la communauté en y introduisant un gène de bonne qualité. Cette extension a pour objectif de mieux explorer l'espace de recherche afin d'améliorer la qualité de communautés obtenues.

4.3.1 Opérateur de mutation adaptatif

Cet opérateur de mutation OCM2 s'adapte à l'évolution des communautés. Ainsi, le remplacement du gène ayant le plus faible score d'interaction et de similarité avec le centre de communauté(= gène ayant le score le plus fort score) est remplacé par un nouveau gène si et seulement le score de ce dernier est meilleur. Sinon l'opérateur de mutation ne fait pas de remplacement, mais une insertion de nouveau gène. La mutation peut ainsi agir sur la taille de la communauté comme dans *GA-PPI-Net*.

Pour mieux comprendre la procédure de mutation optimisée, une illustration graphique est donnée dans la figure 4.7.

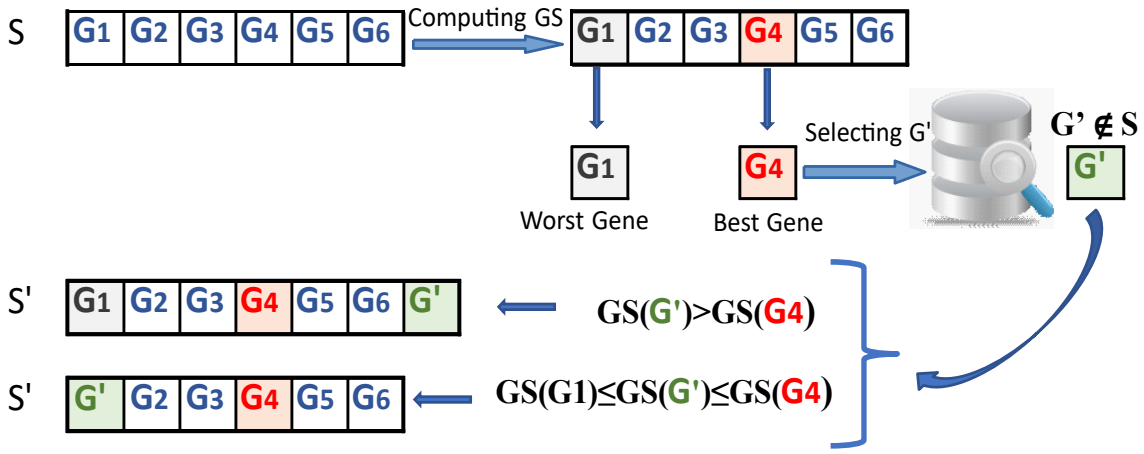


FIGURE 4.7 – Illustration de l'opérateur de mutation adaptatif.

L'algorithme 5 décrit les instructions de l'opérateur OCM2.

L'algorithme 5 de l'opérateur de mutation OCM2 diffère de l'algorithme 3 de l'opérateur de mu-

Algorithme 5 Algorithme de l'opérateur de mutation adaptatif OCM2

Entrée: P_m : probabilité d'appliquer la mutation, S : Enfant obtenu après la réalisation de croisement ;

Sortie: S' // individu enfant issu du processus de mutation.

Début

1: **si** random <Pm **alors**
2: Scores ← ScoreGeneIndividu ; // Calcul des scores GS de chaque gène de S : somme de ses valeurs de similarité et ses valeurs d'interaction avec les autres gènes de l'individu.
3: BestGene ← GetBestGene(Scores) ; // Récupérer le gène de S qui possède le score maximal.
4: GenIn ← SelectGeneInteraction(BestGene) ;//Sélectionner aléatoirement un gène qui interagit avec le bestGene.
5: WorstGene ← GetWorstGene(Scores) ; // Récupérer le gène qui possède le score minimal.
6: **si** $Scores(\text{"}WorstGene\text{"}) \leq Scores(GenIn)$ and $Scores(\text{"}BestGene\text{"}) \geq Scores(GenIn)$" **alors**
7: $S' \leftarrow$ Replace(S, WorstGene, GenIn) ; //Remplacer WorstGene par GenIn
8: **sinon si** $Score(bestGene) < Score(GenIn)$ **alors**
9: $S' \leftarrow$ insert (S, GenIn) ;
10: updatesize(S') ; // Mettre à jour la taille de la solution mutée
11: **fin si**
12: **fin si**
13: Return S' ;

Fin

tation, proposé dans *GA-PPI-Net*, dans les étapes 7 à 10. Dans OCM2, la décision finale de mutation à appliquer dépend du score *GS* du gène G' (gène sélectionné aléatoirement et en interaction avec BestGene) qui est comparé aux scores du WorstGene et BestGene. Les différents scores évoluent durant le cycle évolutionnaire et selon le gène G' sélectionné, d'où le caractère "adaptatif" de OCM2. Au contraire, dans OCM1, la décision de mutation ne dépend que de la comparaison de score de WorstGene à un seuil fixé($= 0.5$).

4.3.2 Étude expérimentale

Afin de valider la qualité des communautés obtenues, nous avons suivi la même démarche d'évaluation proposée dans la section précédente. Les tests de validation de l'extension *Adaptative GA-PPI-Net* effectués sont décrits ci-dessous :

1. Évaluation de la capacité de reconstruction des communautés en utilisant dans la population initiale les gènes tirés aléatoirement des jeux de données présentées dans le tableau 3.2.

Toutes les communautés obtenues sont évaluées par l'outil DAVID. Les résultats d'évaluation permettent de déterminer le taux de recouvrement maximal et minimal des *R_new*(les communautés obtenues par *Adaptative GA-PPI-Net* par rapport aux communautés de jeux de données initiales. Le tableau 4.3 ci-dessous présente les résultats de ces tests.

TABLE 4.3 – Évaluation des communautés retournées par *Adaptative GA-PPI-Net* par rapport aux jeux de données d'entrée.

Datasets	Pourcentage Min	Pourcentage Max
Apoptosis	15%	88%
B cell receptor signalling pathway	20%	75%
Purine metabolism	23%	100%
Rna degradation	-	34%
Oocyte meiosis	19%	78%

Les résultats du tableau 4.3 montrent que les communautés obtenues par *Adaptative GA-PPI-Net* correspondent à des "parties" et dans certains cas à des communautés réelles complètes. Nous constatons ainsi qu'*Adaptative GA-PPI-Net* est capable de reconstruire des communautés complètes ou des sous parties de jeux de données d'entrée.

2. Évaluation de performance en calculant le taux de recouvrement des communautés obtenues par rapport aux communautés existantes dans les bases de voies biologiques.

   Toutes les communautés obtenues *R_new* sont évaluées par l'outil DAVID. Les résultats d'évaluation permettent de déterminer le taux de recouvrement maximal et minimal des *R_new* en comparant aux communautés de voies biologiques. Les résultats de ces tests sont décrits dans le tableau 4.4.

TABLE 4.4 – Évaluation des communautés détectées par *Adaptative GA-PPI-Net* par rapport aux bases de voies biologiques.

BDs	Pourcentage Min	Pourcentage Max Max
BBid	20%	60%
Biocarta	30%	100%
Ec number	30%	100%
Reactome pathway	25%	100%
KEGG pathway	33%	95%

D'après le tableau 4.4, on remarque que la majorité des communautés obtenues corres-

pondent à des communautés réelles (pourcentage =100%) dans les bases Biocarta, Ec number, Reactome. Ces résultats sont considérés comme très satisfaisants. Et ils montrent la performance de l'opérateur de mutation OCM2 à mieux explorer l'espace de recherche afin d'améliorer les qualités de communautés obtenues.

3. La comparaison des performances d'*Adaptative GA-PPI-Net* et *GA-PPI-Net* est présentée dans le tableau 4.5.

   Une comparaison approfondie n'est pas facile, car les communautés obtenues par les deux approches sont de tailles et de structures distinctes. De ce fait, nous avons proposé le protocole expérimental suivant :

   — Utiliser les mêmes datasets présentés dans le tableau 3.2 et les mêmes paramètres GA présenté dans le tableau 4.1.

   — Exécuter chaque approche vingt fois.

   — Déterminer le taux de recouvrement des gènes de chaque communauté obtenue par rapport aux autres communautés de voies biologiques en utilisant l'outil DAVID.

TABLE 4.5 – *GA-PPI-Net* VS *Adaptative GA-PPI-Net*.

BDs	*GA-PPI-Net*		*Adaptative GA-PPI-Net*	
voies biologiques	%Min	%Max	%Min	%Max
BBid	25%	50%	20%	60%
Biocarta	20%	66%	30%	100%
Ec number	10%	100%	30%	100%
Reactome	14%	100%	25%	100%
KEGG	15%	100%	33%	95%

D'après les résultats affichés dans le tableau 4.5, nous constatons que les deux approches permettent de détecter des communautés de bonne qualité. Les données présentées dans la colonne *Adaptative GA-PPI-Net* sont similaires ou meilleurs que celles présentées dans la colonne *GA-PPI-Net*.

— *Adaptative GA-PPI-Net* atteint des pourcentages plus élevés suivant les 2 bases de voies biologiques : Biocarta et BBid.

— *Adaptative GA-PPI-Net* et *GA-PPI-Net* obtiennent le même taux de recouvrement de 100% pour Ec Number et Reactome.

— *GA-PPI-Net* possède le pourcentage maximal (=100%) suivant la base KEGG pathway. Tandis que, *Adaptative GA-PPI-Net* conserve un pourcentage satisfaisant (=95%).

Ces résultats confirment la pertinence de la fonction fitness et l'intérêt de l'opérateur de mutation proposé.

GA-PPI-Net explore le réseau PPI essentiellement grâce à l'opérateur de mutation OCM1. Ce dernier remplace un gène dans une communauté candidate en introduisant un nouveau gène en interaction avec le centre de la communauté (gène ayant le plus grand score qui est basé sur l'interaction et la similarité sémantique) en se basant sur un seuil à définir. Cependant, avec l'évolution des communautés au long des générations, cet opérateur peut exclure des gènes qui font réellement partie de la communauté et la fitness peut se dégrader au lieu de s'améliorer. Nous proposons alors une nouvelle version de cet opérateur OCM2 qui s'adapte à l'évolution des communautés. Ainsi, le remplacement du gène ayant le plus faible score d'interaction et de similarité avec le centre de communauté est remplacé par un nouveau gène si et seulement le score de ce dernier est meilleur (l'étape 7 et 8, l'algorithme 5). Sinon l'opérateur de mutation ne fait pas de remplacement, mais une insertion de nouveau gène (étape 8 et 9, l'algorithme 5). La mutation peut ainsi agir sur la taille de la communauté.

4.4 Conclusion

Dans ce chapitre, nous avons présenté les extensions et les améliorations apportées à *GA-PPI-Net*. L'objectif de la première amélioration est de rendre *GA-PPI-Net* générique permettant de détecter des communautés de gènes toujours de longueur variable comme avec *GA-PPI-Net* et qui sont sémantiquement similaires et/ou sont en interaction. Cette généralisation permet de détecter trois types de communautés : i) groupes de gènes basés sur l'interaction, ii) groupes de gènes basés sur la similarité et iii) communautés de gènes similaires et qui sont en interaction. L'objectif de la deuxième amélioration de l'approche *GA-PPI-Net* consiste à mieux explorer l'espace de recherche. Elle est réalisée en proposant un opérateur de mutation adaptative. Cet opérateur favorise l'exploration de l'espace de recherche et la détection des communautés de meilleure qualité. Le

chapitre suivant sera consacré à la mise à l'échelle de l'approche *GA-PPI-Net* pour les réseaux de très grandes tailles en appliquant les techniques d'échantillonnages

Chapitre 5

La mise à l'échelle de GA-PPI-Net : *Active GA-PPI-Net*

Sommaire

5.1 Introduction

Les études expérimentales ont prouvé la performance de l'approche proposée *Generic GA-PPI-Net* pour la construction et la détection de communautés de gènes qui sont similaires sémantiquement et/ou interagissant entre eux. La performance de cette approche a été testée sur un ensemble de gènes de voies biologiques connus. Nous nous intéressons, dans ce dernier chapitre, à détecter des communautés de gènes à partir des données volumineuses (le réseau entier d'interaction de protéines humaines). Ce travail est basé sur la mise à l'échelle de *Generic GA-PPI-Net*.
La mise à l'échelle des AGs se fait essentiellement par deux techniques :

— La parallélisation (verticale ou horizontale),

— L'introduction de paradigmes à l'intérieur de l'AG comme le paradigme d'apprentissage actif.

Pour ce travail, nous nous inspirons des méthodes d'échantillonnage de graphe pour appliquer le paradigme d'apprentissage actif à l'intérieur de l'AG afin de construire et de détecter des communautés à partir du réseau complet d'interaction des protéines humaines (Homo sapiens). Deux challenges se posent avec la base d'interaction de protéines humaines : le volume et la complexité. Il est à noter que l'application de *Generic GA-PPI-Net* sur la base entière est coûteux, car le réseau d'interaction protéine-protéine est très volumineux. Pour relever les deux challenges, nous appliquons une extension à *Generic GA-PPI-Net* en intégrant le paradigme d'apprentissage actif à l'intérieur de notre AG.
Pour ce faire, nous prenons comme base d'entrée la base d'interaction de tous les gènes humains disponible sur le site de STRING [1]. Le nombre de gène est de plus de 11 millions de gènes. Cette base servira au calcul de la mesure d'interaction au sein d'une communauté. Pour le calcul de la mesure de similarité, la méthode GS2 (Ruths et al., 2009) est utilisée entre chaque paire de gènes.
La recherche de communautés dans des bases de telles tailles est un problème complexe et très couteux, voire impossible. Nous proposons donc *Active GA-PPI-Net* qui est une extension de *Generic GA-PPI-Net* introduisant un échantillonnage actif comme solution pour la réduction du volume de données en sélectionnant les entrées à exploiter. De nouvelles composantes sont alors introduites dans *Generic GA-PPI-Net* lui permettant de sélectionner les données de manière dynamique ou

1. https://string-db.org/

adaptative à l'évolution des communautés dans la population courante. Le but avec l'apprentissage actif est d'offrir la possibilité à *Active GA-PPI-Net* de décider des nouvelles données à exploiter selon l'état courant des meilleures solutions.

Ce chapitre est consacré à la présentation et l'étude du paradigme d'apprentissage actif avec des méthodes d'échantillonnage de graphe dans le cadre de la détection des communautés de gènes avec l'AG. Nous commençons par rappeler les principes du paradigme d'apprentissage actif et de l'échantillonnage de graphe. Nous proposons ensuite une solution pour un échantillonnage dynamique et adaptatif du graphe PPI et nous expliquons les détails d'introduction de la méthode proposée dans *Active GA-PPI-Net* pour la détection de communautés dans de larges réseaux biologiques.

5.2 Pré-traitement des données Homo sapiens

Au niveau de cette section, nous présentons les données extraites et utiles pour ce travail.

Afin de diversifier nos données et montrer l'efficacité de notre algorithme, nous avons utilisé la base d'interaction de tous les gènes humains disponible sur le site de STRING. Cette base décrit toutes les interactions pour chaque paire de gènes. Chaque ligne de cette base décrit l'interaction entre deux gènes comme suit :

« Gene1 Gene2 combined_score »

Exemple : « 9606.ENSP00000000233 9606.ENSP00000272298 490

Les deux premiers éléments désignent les identifiants des deux gènes en interaction. Le dernier nombre désigné le score d'interaction qui est un nombre positif strictement inférieur à 1000. Nous avons appliqué des traitements au réseau complet d'interaction de protéines humaines résumés ci-dessous et illustrés dans la figure 5.1 :

1. Extraire tous les noms de gènes pour chaque identifiant de la base String. Ces noms sont récupérés de la base en ligne NSBI[2]. Par exemple, le gène "MRPL27" est identifié par "9606.ENSP00000225969" dans la base String-BD (figure 5.1, étape 1). Ce travail est nécessaire, car ces noms de gènes sont ceux utilisés dans le calcul de la similarité et dans

2. https://www.ncbi.nlm.nih.gov/

l'évaluation ultérieur des communautés par l'outil DAVID.

2. Déterminer les annotations de chaque paire de gènes du réseau d'interaction. Un gène est annoté par un ou plusieurs termes de Gene Ontology. Ces annotations sont déterminées de la source Gene Ontology Annotation. Le gène "MRPL27" est annoté par l'ensemble des termes GO ["GO :0032543", "GO :0006412", "GO :0006412"] de la sous ontologie Biological Process (figure 5.1, étape 2).
3. Calculer la similarité sémantique en utilisant la méthode GS2 (figure 5.1, étape 3).
4. Fusionner les données d'interaction et de similarité sémantique de chaque paire de gènes dans une seule base (figure 5.1, étape 4). Cette base sera la source pour *Active GA-PPI-Net*.

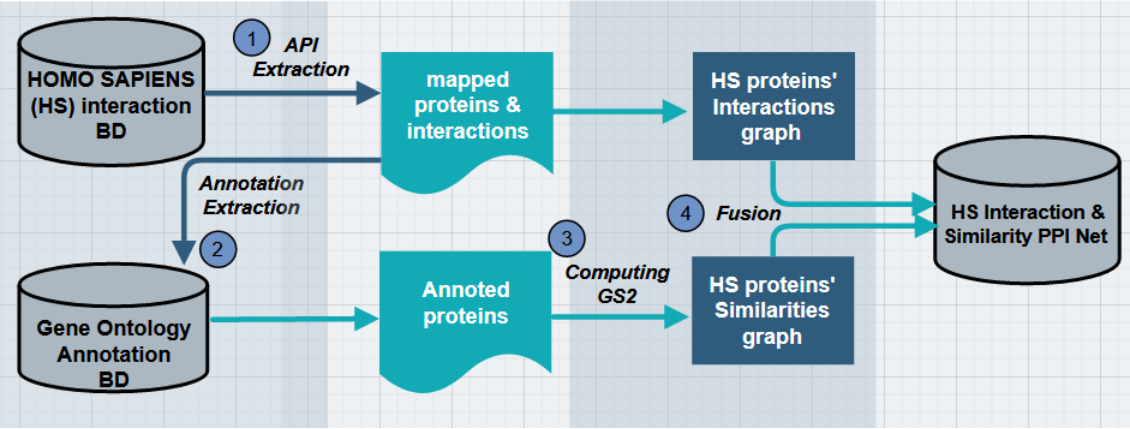


FIGURE 5.1 – Extraction et préparation des données utilisées.

À la fin des étapes de pré-traitement des données décrites ci-dessus, nous obtenons une base unique du réseau d'interaction Homo sapiens présentant les couples de gènes avec leurs noms, leur valeur d'interaction et de similarité. La figure 5.2 représente un extrait de données obtenues. Ces données sont modélisées sous forme de graphe pour la phase de détection de communautés par *GA-PPI-NET* (figure 5.2). Ce graphe comporte 16655 nœuds et 892023 arcs.

5.3 Mise à l'échelle de *Generic GA-PPI-Net* par l'apprentissage actif

Cette section est consacrée à la présentation de l'approche proposée *Active GA-PPI-Net* dans le cadre de la détection des communautés de gènes avec l'AG à partir de données volumineuses. Nous

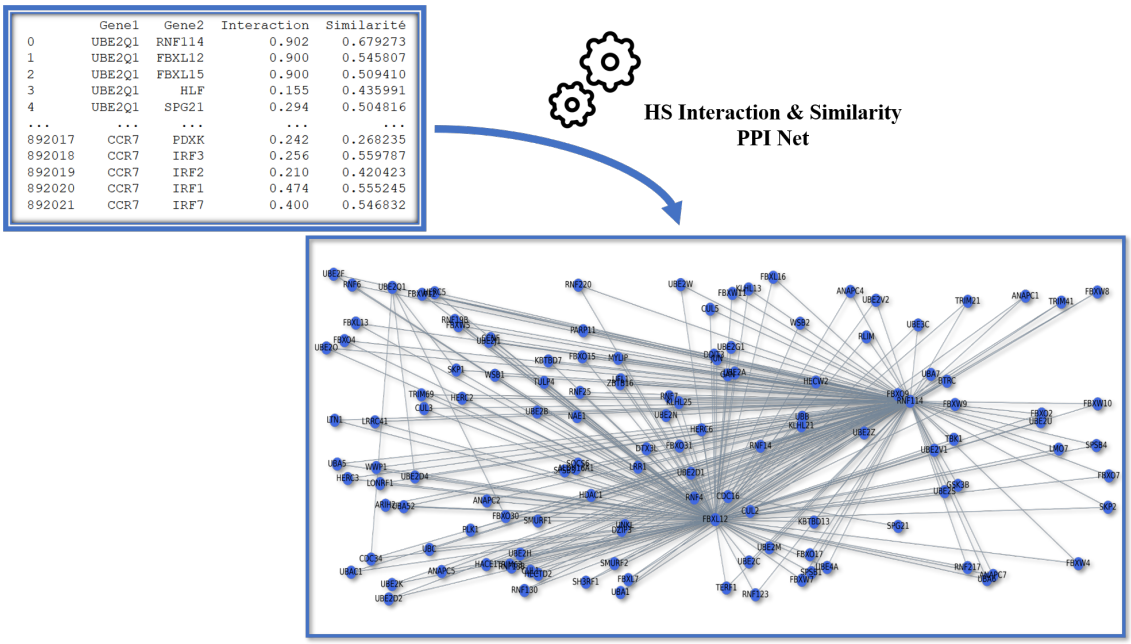


FIGURE 5.2 – Extrait de données obtenues.

commençons par rappeler les principes du paradigme d'apprentissage actif et de l'échantillonnage de graphe. Nous proposons ensuite une solution pour un échantillonnage dynamique et adaptatif du graphe PPI.

5.3.1 Apprentissage Actif Vs Échantillonnage des graphes

Plusieurs solutions ont été proposées pour faire face aux données volumineuses et multidimensionnelles en utilisant les AGs. Ces solutions sont basées sur l'ajout des paradigmes d'apprentissage, tels que l'apprentissage actif, pour améliorer la capacité d'exploration des données. Dans le contexte de la détection de communautés, l'apprentissage actif pourrait être une solution efficace pour détecter des communautés à partir des réseaux complexes très larges comme le réseau PPI.
L'apprentissage actif [(Atay et al., 2017), (Cohn et al., 1994)] peut être défini comme :

« toute forme d'apprentissage dans laquelle le programme d'apprentissage possède un contrôle sur les entrées à partir desquelles il apprend. »

L'apprentissage actif a été introduit essentiellement avec des techniques d'échantillonnage de données dites dynamiques.
Les objectifs des approches d'échantillonnage sont de :

— Réduire la taille de l'ensemble d'apprentissage et le coût de calcul.

— Améliorer la performance de l'apprenant en variant l'espace de recherche.

Une approche d'échantillonnage dépend de la topologie des données. Pour des données organisées en ligne ou *"row-based data"*, plusieurs techniques d'échantillonnage ont été proposées, utilisant une technique de sélection parmi celles données ci-dessous :

1. La sélection aléatoire des données ;
2. La sélection pondérée des données : elle est basée sur l'utilisation des informations complémentaires sur l'état actuel des données d'apprentissage ;
3. La sélection des données basée sur la topologie : elle utilise des informations sur la topologie des données afin d'éviter de sélectionner des données redondantes.

Les techniques de sélection proposées pour les *"row-based data"* ne peuvent pas être appliquées telles quelles pour les données en graphe ou *"graph-based data"*. Des adaptations ou de nouvelles propositions s'imposent pour définir une technique d'échantillonnage de graphes.
L'échantillonnage de graphes est un sujet largement étudié en apprentissage automatique. Il est basé sur l'apprentissage de la représentation graphique pour créer des sous-ensembles ou sous groupes. Ces sous-ensembles seront utilisés comme un vecteur de caractéristiques pour des tâches d'apprentissage automatique de graphes en aval. L'apprentissage de la représentation graphique est une étape fondamentale dans plusieurs domaines tels que l'analyse des réseaux sociaux, les systèmes de recommandations, l'épidémiologie, etc (Jangda et al., 2021). Le principe des algorithmes d'apprentissage de la représentation graphique est le suivant :

1. Échantillonner le graphe d'entrée pour obtenir des groupes de données (=échantillons).
2. Appliquer un algorithme d'apprentissage, par exemple un réseau neuronal profond (DNN) ou un réseau neuronal graphique (GNN) sur la base des échantillons créés.

Les techniques d'échantillonnage de graphes les plus connues sont basées sur l'approche de la marche aléatoire, telles que DeepWalk (Perozzi et al., 2014) ou Node2Vec (Grover and Leskovec, 2016).
L'échantillonnage de graphes étudié dans ce chapitre est différent à la fois de l'échantillonnage de données organisées en lignes pour définir l'apprentissage actif et de l'échantillonnage des graphes

pour l'apprentissage de la représentation graphique. Nous proposons une méthode d'échantillonnage de graphe adaptatif permettant de créer des échantillons à partir du réseau d'interaction de protéines humaines. Ces échantillons aideront *Active GA-PPI-Net* à mieux évoluer pour détecter les meilleures communautés sans explorer l'ensemble du graphe PPI. Cette méthode, que nous appelons Adaptive Graph Sampling, prend en compte : i) l'évolution des meilleures communautés construites tout au long du processus d'apprentissage ; et ii) les valeurs des seuils de similarité sémantiques ∇S et d'interaction ∇I définis comme paramètres de *Generic GA-PPI-Net*.
Pour mieux exploiter les meilleures solutions, *Adaptive Graph Sampling* sélectionne dans les échantillons tous les voisins des meilleurs gènes des meilleures communautés courantes en respectant le seuil d'interaction et de similarité.

5.3.2 Adaptive Graph Sampling

Adaptive Graph Sampling est une solution d'échantillonnage proposée comme une approche pour la réduction du volume de données. Elle sélectionne les entrées à exploiter selon l'état courant des meilleures solutions (individus). Elle permet de créer des échantillons à partir du réseau PPI humain. Les gènes sélectionnés dans un échantillon dépendent non seulement du meilleur individu donné en entrée à l'algorithme, mais aussi des valeurs de similarité et d'interaction reliant les meilleurs gènes avec ses voisins dans le graphe. Seuls les voisins respectant les seuils définis dans l'algorithme sont sélectionnés. Elle est décrite dans l'algorithme 6.
Le principe de la méthode de création des échantillons en se basant sur les valeurs de similarités et d'interactions est le suivant :

A) Récupérer tous les gènes qui existent dans le dictionnaire *DictG* et qui sont reliés aux gènes G_b de la meilleure solution ou communauté (lignes 1 et 2, Algorithme 6). Le dictionnaire *DictG* contient tous les gènes du réseau PPI humain avec leur valeur de similarité et d'interaction.

B) Pour chaque gène récupéré G_c, si la similarité des deux gènes G_b et G_c est supérieure à ∇S et leur interaction, est supérieure à ∇I alors le gène G_c est ajouté à l'échantillon (lignes 3 à 6, Algorithme 6).

Cette méthode permet d'implémenter un échantillonnage dynamique et adaptatif du graphe en sé-

Algorithme 6 Active learning : Fonction Adaptive Graph Sampling

Paramètres :
$\mathcal{B}$: Meilleure communauté
DictG : Dictionnaire contenant les valeurs d'interaction et de similarité pour chaque paire de gènes
$\nabla I, \nabla S$: seuil d'interaction et seuil de similarité, respectivement.
Début

1: **pour chaque** G_b in $\mathcal{B}$ **faire**
2: *Connected* ← Récupérer tous les gènes qui existent dans DictG et relié à G_b
3: **pour tout** G_c in *Connected* **faire**
4: **si** Interaction$(G_b, G_c) \geq \nabla I$ AND Similarité(G_b,G_C) $\geq \nabla S$ **alors**
5: Ajouter G_c à Échantillon
6: **fin si**
7: **fin pour**
8: **fin pour**
9: Retourner Échantillon
Fin

lectionnant les nœuds respectant des contraintes de seuils de similarité et d'interaction selon l'évolution des meilleures communautés. L'adaptabilité dans la méthode *Adaptive Graph Sampling* est assurée par la composante **B** (meilleure communauté). En effet, la sélection des nouveaux gènes à insérer dans l'échantillon dépend de la disposition de la meilleure communauté courante dans le graphe, c'est-à-dire celle ayant la meilleure fitness. Seule le voisinage de cette solution est considéré tout en respectant les seuils ∇I et ∇S données en paramètres. Par ailleurs, cette méthode est dynamique. Elle est appliquée par *Active GA-PPI-Net* d'une manière régulière au long de l'évolution, permettant un changement dynamique des données explorées par les opérateurs génétiques. L'intervalle de temps à respecter pour rappeler la méthode est appelé fréquence d'échantillonnage f_s.

5.3.3 *Active GA-PPI-Net*

Notre objectif vise à détecter des communautés de gènes à partir de données volumineuses (réseau d'interaction de protéines humaines) en minimisant le temps de calcul de notre AG. Ce temps de calcul est dû essentiellement aux opérateurs génétiques qui sont amenés à explorer l'ensemble du graphe pour définir l'évolution à appliquer aux communautés dans la population. L'objectif est de limiter l'espace de recherche pour ces opérateurs en sélectionnant la partie du réseau à explorer selon l'évolution des meilleures communautés. Cet objectif peut être atteint avec le paradigme de

l'apprentissage actif. Pour atteindre cet objectif, nous proposons une extension de *Generic GA-PPI-Net* en appliquant l'apprentissage actif. Comme les données sont remplacées à toutes les f_s générations (f_s est la fréquence d'échantillonnage, nombre à définir), cela laisse le temps à la population de, à la fois, s'adapter à l'échantillon en cours et d'explorer un espace de recherche varié. L'algorithme proposé se base sur deux idées principales qui sont la phase d'initialisation des individus et l'évolution de la population par des opérateurs génétiques de croisement et mutation. L'évolution de l'opérateur de mutation est réalisée en appliquant l'opérateur de mutation OCM1 décrit dans la section 3.3.5.2 et présenté par l'algorithme 3.

Le principe général de *Active GA-PPI-Net* est le même que celui de *Generic GA-PPI-Net*. En effet, la population est formée d'individus qui sont des solutions du problème (= un ensemble de gènes qui forment une communauté de gènes qui sont similaires et/ou sont en interactions). Nous avons utilisé la fonction objective qui est basée la valeur de similarité et/ou la valeur d'interaction de chaque paire de gènes à maximiser décrite dans l'équation 4.1 et 4.2. Après la création de la population initiale de n individus, nous réalisons des opérations génétiques pendant un certain nombre des itérations ($max\_iteration$). Au cours de chaque , deux parents sont sélectionnés par tournoi pour réaliser le croisement et obtenir deux fils. La méthode de mutation présentée dans l'algorithme 3 est appliquée à ces deux fils avec une certaine probabilité en utilisant des données générées par la méthode d'échantillonnage adaptative décrite dans l'algorithme 6. Les deux nouveaux individus fils seront évalués et remplaceront les deux plus mauvais individus de la population courante. Afin de préserver la meilleure solution rencontrée lors du processus évolutif, nous utilisons la stratégie par tournoi. À ce propos, à chaque itération, le meilleur membre est sauvegardé pour être réinjecté dans la population de la génération suivante. De cette manière, nous garantissons la survie du meilleur individu pour les générations suivantes. Un pseudo code de l'algorithme général de la solution proposée est présenté dans l'algorithme 7.

Les objectifs avec les nouvelles contributions introduites dans *Active GA-PPI-Net* sont de :

— Permettre à la population de voir le plus d'exemples d'apprentissage possible durant l'évolution pour éviter qu'un type particulier de connaissances ne soit ignoré à cause de l'échantillonnage.

— Réduire le temps d'exécution de l'AG et lui donner suffisamment de temps pour apprendre à

Algorithme 7 Algorithme Général de *Active GA-PPI-Net*

Entrée: Paramètres de l'AG, $\nabla I, \nabla S$, f_s (Fréquence d'échantillonnage)
Sortie: Meilleure communauté
Début
Initialiser aléatoirement la population P_0
Évaluer la population initiale P_0 en utilisant l'équation 4.1
$Best \leftarrow$ Récupérer le meilleur individu de P_0
$Sample(1) \leftarrow$ AdaptiveGraphSampling($Best$,$Dict$,∇I, ∇S)
pour $i = 1$ to max_iteration **faire**
  Sélectionner deux parents de la population
  pour chaque paire de parents sélectionnés **faire**
    $E1, E2 \leftarrow$ Générer un offspring en utilisant le croisement 2-point ou heuristique avec une probabilité p_c
    $ch1, ch2 \leftarrow$ Muter $E1$ et $E2$ en utilisant l'opérateur de mutation OCM1 et $Sample(i)$ avec une probabilité p_m
    Évaluer la fitness de offsprings $ch1$ et $ch2$ en utilisant l'équation 4.1
    Remplacer les mauvais individus de la population par les deux offsprings $ch1$ et $ch2$
  fin pour
  si $MOD(i, f_s) = 0$ **alors**
    $Best \leftarrow$ Récupérer le meilleur individu de P_i
    $Sample(i) \leftarrow$AdaptiveGraphSampling($Best$,$Dict$,∇I, ∇S)
  fin si
fin pour
Fin

partir des données disponibles.

Étant donné ces deux sous-objectifs, nous avons évalué l'aptitude et la pertinence de *Active GA-PPI-Net*. Cette évaluation est présentée dans les sections suivantes.

5.4 Études expérimentales comparatives

Dans cette section, nous présentons les tests effectués afin d'évaluer la performance de notre approche *Active GA-PPI-Net*. Ces tests visent à montrer l'efficacité et les performances de l'approche proposée à construire des communautés de gènes en utilisant des échantillons du réseau d'interaction de protéines chez l'être humain. Deux études expérimentales sont effectuées dans ce chapitre. La première étude vise à valider la méthode *Adaptive Graph Sampling* et vérifier la capacité de *Active GA-PPI-Net* à atteindre les objectifs posés notamment la réduction du coût de calcul en présence d'un grand réseau, optimiser le travail des opérateurs génétiques avec une réduction ci-

blée de l'espace de recherche et renforcer le respect des seuils de similarité et d'interaction. Des séries d'expériences ont été réalisées pour chaque objectif et discutées dans les deux sous-sections ci-dessous.

5.4.1 *Generic GA-PPI-Net* Vs *Active GA-PPI-Net*

Dans cette section, nous proposons d'évaluer l'apport de l'ajout d'une méthode d'échantillonnage à *Generic GA-PPI-Net* quand il est appliqué à la totalité de réseau d'interaction de protéines humaines (graphe de 16655 nœuds et 892023 arcs). Nous étudions la robustesse de l'algorithme avec et sans appliquer la méthode *Adaptive Graph Sampling* (algorithme 6). L'évaluation sera faite en comparant la qualité des communautés et la durée d'exécution (temps de calcul) par rapport à *Generic GA-PPI-Net*. Pour cette étude, nous avons utilisé les paramètres de l'AG comme suit : i) taille de population =20, ii) nombre de générations =600, iii) taux de croisement =0.1 , vi) taux de mutation=0.5 et v) $\nabla I = 0,4, \nabla S = 0.5$. Et la population initiale est constituée de communautés de gènes tirés aléatoirement de réseau PPI. Avec l'échantillonnage adaptatif, le taux de mutation a été augmenté afin de favoriser l'exploitation des meilleures solutions. Les tests sont effectués sur une machine de type DELL Intel(R) Core(TM) i7-8665U CPU 1.90GHz 2.11 GHz, RAM 16 GO.
Pour chaque méthode, dix tests (run) ont été réalisés avec les paramètres cités ci-dessus. Pour chaque test, nous enregistrons les valeurs moyennes, médiane et maximale de la performance de la population courante. Et à la fin du run, nous retenons la meilleure solution (communauté).
La figure 5.3 illustre les courbes d'évolution des valeurs moyennes, médianes, minimales (worst) et maximale (best) de la performance pendant 600 générations, enregistrées pour un cas test avec *Generic GA-PPI-Net* (5.3 à gauche) et *Active GA-PPI-Net* (5.3 à droite). Nous attirons l'attention sur le fait que l'échelle des ordonnées (fitness) n'est pas la même : elle varie de 0 à 150 pour *Generic GA-PPI-Net* et de 0 à 600 pour *Active GA-PPI-Net*.

Deux grandes observations peuvent être faites à partir des courbes de la figure 5.3. D'abord, l'évolution de la meilleure performance ainsi que celle médiane et moyenne avec *Active GA-PPI-Net* est continue et accélérée. En effet, à chaque génération, les deux communautés enfants générées avec les opérateurs génétiques ont une meilleure performance que leurs parents grâce au renforcement des interactions et des similarités entre les gènes avec l'échantillonnage adaptatif. La méthode

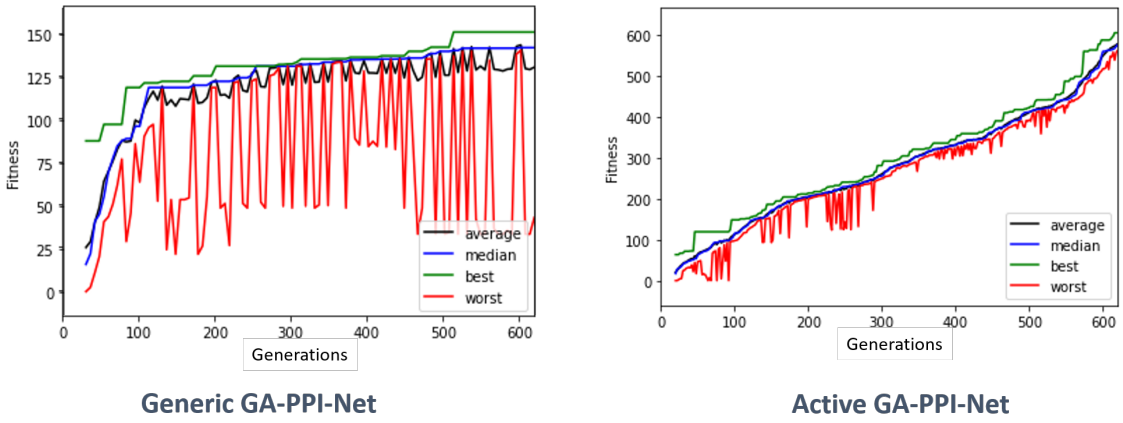


FIGURE 5.3 – Exemple d'évolution de fitness pour les deux approches **Generic GA-PPI-Net** et **Active GA-PPI-Net**

Graph Adaptive Sampling a permis de réduire l'espace d'exploration des opérateurs génétiques, permettant ainsi une meilleure exploitation des meilleures solutions obtenues au cours de l'évolution. Par ailleurs, avec *Generic GA-PPI-Net*, nous avons constaté que le temps de calcul total varie entre 10 heures et 21 heures pour les 600 générations. Par contre, pour tous les tests réalisés avec *Active GA-PPI-Net*, ce temps de calcul varie entre 49mn et 2h49mn indépendamment des seuils d'interaction et de similarité. Il est donc amélioré avec l'approche mise à l'échelle sans augmentation de la complexité de l'algorithme. *Active GA-PPI-Net* offre de loin le meilleur compromis rapidité/optimisation. La figure 5.4 décrit un exemple d'évolution de communautés en utilisant *Active GA-PPI-Net*. Partant d'un ensemble de gènes choisis aléatoirement, l'AG proposé détermine les voisins de ces gènes ayant une forte interaction et une forte similarité sémantique grâce à la méthode *Adaptive Graph Sampling*. Le choix sélectif de l'ensemble des gènes à étudier pendant l'apprentissage a permis à la fois une meilleure convergence et une grande accélération de l'AG.

5.4.2 *Random GA-PPI-Net* Vs *Active GA-PPI-Net*

L'objectif de cette section est d'effectuer une étude expérimentale pour comparer la méthode *Random GA-PPI-Net* et *Active GA-PPI-Net* pour construire et détecter des communautés de gènes à partir de données volumineuses. La méthode*"Random GA-PPI-Net"* consiste à faire évoluer *Generic GA-PPI-Net* en utilisant des échantillons ou groupe de données créés aléatoirement. Nous avons mené cette étude dans un cadre expérimental homogène pour les deux méthodes. Chaque

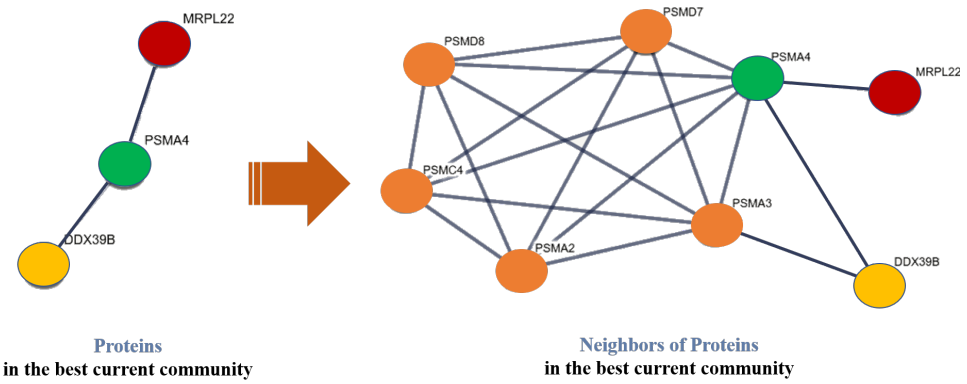


FIGURE 5.4 – Exemple d'évolution d'une communauté de gènes obtenue par *Active GA-PPI-Net*

méthode est testée 11 fois afin d'obtenir une performance moyenne. À la fin de chaque run (test), le meilleur individu selon la fonction de fitness est retenu. Les deux algorithmes ont été paramétrés selon le tableau 5.1. Cette combinaison est utilisée pour faire la comparaison entre les deux techniques d'échantillonnage.

TABLE 5.1 – Paramètres de comparaison.

Paramètres	Valeur
Taille de la population	20
Nombre de générations	100
Probabilité de croisement	0.1
Probabilité de mutation	0.5
Taille tournoi	4
∇S	0.5
∇I	0.4

Le tableau 5.2 décrit les résultats d'exécution en utilisant le paradigme d'apprentissage actif selon deux techniques : aléatoire *(Random AG-PPI-Net)* et *adaptative (Active GA-PPI-Net)*.

Les résultats présentés dans le tableau 5.2 montrent que *Active GA-PPI-Net* obtient la meilleure valeur de la fonction fitness ($= 422,22$) pour construire et détecter des communautés de gènes. Les valeurs de fitness obtenus avec *Active GA-PPI-Net* sont en général plus élevés ou de même ordre qu'avec *Random GA-PPI-Net*. Il est clair que *Active GA-PPI-Net* est performant pour résoudre le problème de détection de communautés à partir des données massives comme celui du réseau d'interaction de protéines humaines. En effet, un run de cette approche dure en moyenne $0.5s$ par génération pour 20 individus et $1.16s$ pour détecter et construire des communautés de taille 182 et

TABLE 5.2 – *Random GA-PPI-Net* Vs *Active GA-PPI-Net*.

Run	*Random AG-PPI-Net*			*Active GA-PPI-Net*		
	fitness	time(mn)	Size	fitness	time(mn)	Size
1	201,1	24	117	**422,2**	30	182
2	165,42	21	96	**281,65**	25	153
3	**246,39**	25	109	114,1	20	90
4	286,69	27	149	**296,50**	27	147
5	**258,21**	24	125	178,74	23	109
6	231,84	26	117	**310,22**	32	163
7	242,5	23	113	274,96	60	148
8	290,91	29	121	**364,82**	70	185
9	**239,73**	24	142	183,15	36	89
10	**334,42**	32	170	304,63	45	150
11	**233,64**	30	132	221,46	32	128

185 respectivement. Par contre, une exécution de *Random GA-PPI-Net* prend 0.55 seconde pour détecter des communautés de taille 170. La première observation qui peut être faite à partir du tableau 5.2 est que les deux approches *Random GA-PPI-Net* et *Active GA-PPI-Net* permettent une grande optimisation du coût de calcul, mais avec un temps d'exécution légèrement inférieur avec l'échantillonnage aléatoire. En effet, ce dernier ne nécessite pas des tests comparatifs pour sélectionner les nœuds à insérer dans le nouvel échantillon comme pour l'échantillonnage adaptatif. Ces tests sont effectués par *Active GA-PPI-Net* afin de respecter les seuils de similarité et d'interaction imposés pour les communautés. Ainsi, le fait d'introduire le paradigme d'apprentissage actif permet d'améliorer les performances de l'AG pour détecter des communautés assez large et avec une valeur de fitness maximale. Comme indiqué plus haut, *Active GA-PPI-Net* est légèrement couteux vu la sélection des gènes par rapport à la meilleure communauté courante. Ce coût est toutefois compensé par la qualité des communautés obtenues à la fin.

5.5 Analyse et évaluation d'*Active GA-PPI-Net*

Dans cette section, nous présentons premièrement les résultats obtenus par *Active GA-PPI-Net* pour détecter des communautés existantes dans des bases de voies biologiques selon la variation de seuils ∇S et ∇I. Ces communautés peuvent être similaires (seuil de similarité élevé), en interaction

(seuil d'interaction élevé), ou les deux. En deuxième lieu, nous introduisons les tests effectués pour étudier la performance et la faisabilité d'*Active GA-PPI-Net* pour détecter des communautés de gènes connues et réelles.

5.5.1 Détection de communautés avec *Active GA-PPI-Net* selon différents seuils d'interaction et de similarité

Dans la section 4.2 du chapitre 4, nous avons introduit l'approche *Generic GA-PPI-Net* pour détecter des communautés qui sont sémantiquement similaires et/ou en interaction basée sur une fonction de performance générique paramétrée par les seuils ∇S et ∇I. L'approche proposée dans ce chapitre, *Active GA-PPI-Net*, s'adapte parfaitement avec l'approche Generic GA-PPI-Net. En effet, *Active GA-PPI-Net* peut optimiser et accélérer la construction des communautés selon la définition des deux seuils ∇S et ∇I grâce à l'échantillonnage adaptatif. Ainsi, les gènes sélectionnés dans le nouvel échantillon dépendent non seulement du meilleur individu donné en entrée à l'algorithme, mais aussi des valeurs de similarité et d'interaction reliant les meilleurs individus avec ses voisins. Seuls les voisins respectant les seuils définis dans l'algorithme sont sélectionnés. Ceci permettrait d'accélérer l'évolution et la convergence du meilleur individu en cours de construction. De plus, ceci permettrait d'améliorer la tâche de l'opérateur de mutation OCM1.

L'objectif principal de *Generic GA-PPI-Net* est la détection de communautés ou groupes de gènes en respectant des seuils de similarité et d'interaction définis préalablement. L'introduction de la méthode d'échantillonnage « Adaptive Graph Sampling » permet, en plus de la réduction du coût, de renforcer cet objectif et d'orienter la convergence de l'AG vers des communautés respectant strictement les seuils imposés. L'étude expérimentale dans cette section vise à démonter cette nouvelle capacité apportée à GA-PPI-Net. Nous faisons varier les seuils d'interaction et de similarité et nous analysons les résultats obtenus. L'étude expérimentale a été réalisée en deux étapes : i) A la première étape, seul le seuil de similarité varie dans l'intervalle $[0.5, 0.9]$ avec un seuil d'interaction fixe ; ii) A la deuxième étape, c'est le seuil d'interaction qui varie dans le même intervalle avec un seuil de similarité fixe.

5.5.1.1 *Active GA-PPI-Net* : Variation du seuil de similarité

Pour étudier l'efficacité de notre approche proposée, nous avons exécuté *Active GA-PPI-Net* 5 fois pour chaque valeur de seuils ∇S variant de 0.5 à 0.9 et $\nabla I = 0.4$. À la fin de chaque run, nous retenons la valeur de fitness maximale, moyenne et médiane. Pour réaliser cette étude, une combinaison de paramètres d'AG retenue est présentée dans le tableau 5.3.

TABLE 5.3 – Paramètre d'*Active GA-PPI-Net* : Variation de seuil de similarité.

Paramètres	Valeur
Taille de la population	20
Nombre de générations	300
Probabilité de croisement	0.1
Probabilité de mutation	0.5
Taille tournoi	4
∇S	[0.5, 0.9]
∇I	0.4

Les valeurs de fitness maximales et moyennes associées à chaque variation de seuil sont présentées dans le tableau 5.4.

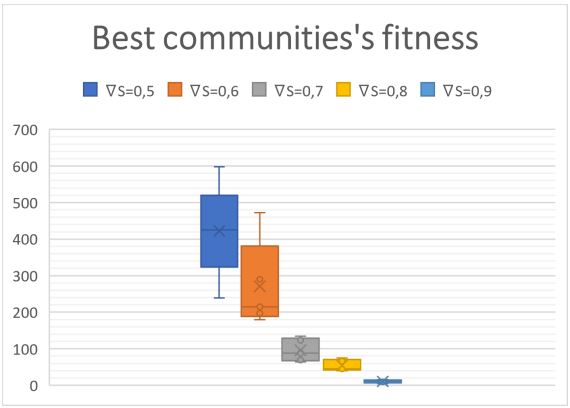
TABLE 5.4 – Évolution de la fitness selon la valeur de similarité ∇S.

∇S	**Fitness Max sur 5 tests**			**Fitness Moyen sur 5 tests**		
	Best	AVG	Median	Best	AVG	Median
0,5	597,37	573,01	567,63	422,17	408,97	402,85
0,6	472,06	449,78	450,45	270,75	258,30	256,95
0,7	134,68	127,20	127,97	95,95	92,47	89,41
0,8	74,87	65,12	65,89	54,10	49,84	50
0,9	15,14	15,14	14,20	10,72	10,72	10,53

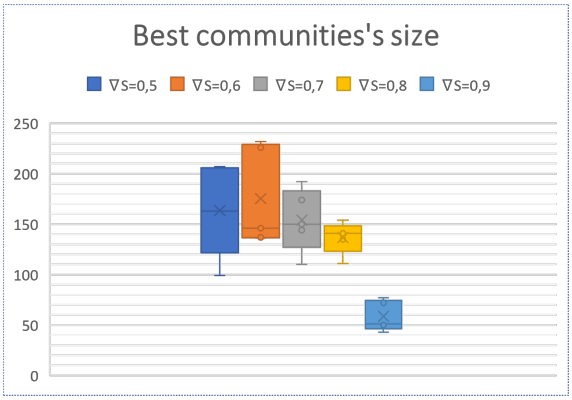
Nous constatons, au vu des résultats présentés dans la table 5.4 qu'*Active GA-PPI-Net* possède une meilleure de fitness quand $\nabla S \in [0.5, 0.6]$. Notre algorithme *Active GA-PPI-Net* détecte des communautés de gènes qui sont sémantiquement similaires, ce qui signifie qu'elles obtiennent une détection significative avec une qualité élevée.

La figure 5.5 (Évolution de la fitness) présente l'évolution de la valeur du fitness en faisant varier ∇S. La valeur de la meilleure fitness est 597.37. On constate que *Active GA-PPI-Net* obtient donc

sur cette tâche un bon résultat si $\nabla S = 0.5$ ou $\nabla S = 0.6$ mais les valeurs de fitness sont moins importantes si $\nabla S = 0.9$. Rappelons que cet algorithme nécessite un paramétrage correspondant à la valeur de seuil. Cela signifie que notre méthode a la capacité de détecter des communautés de gènes qui sont similaires sémantiquement si la valeur de seuil ∇S est comprise entre 0.5 et 0.7.
L'augmentation des seuils de similarité permet une réduction de l'espace de recherche de notre AG. En effet, la taille des échantillons générés par la méthode «Adaptive Graph Sampling» diminue en parallèle que les seuils, ce qui réduit aussi l'effort d'exploration locale effectuée par l'opérateur de mutation et du croisement heuristique. Ceci explique également la réduction progressive de la taille des communautés obtenues par l'AG illustrées dans la figure 5.5 (Évolution de la taille). En effet, la taille des communautés atteint des valeurs inférieures à 50 avec des $\nabla S = 0.9$. En effet, la réduction



Évolution de la fitnes



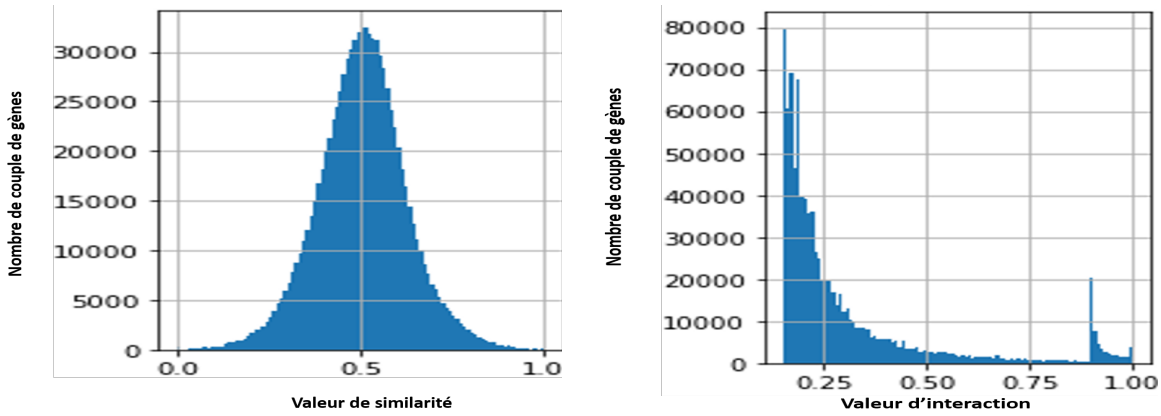
Évolution de la taille

FIGURE 5.5 – Evolution de la fitness en variant ∇S

de l'espace de recherche est beaucoup plus rapide en augmentant ∇S, ceci est dû essentiellement à la distribution générale de ces seuils sur le graphe illustrés dans la figure 5.6 (a). Les valeurs de similarités sont centrées autour de la valeur 0.5, où la majorité des arcs ont des valeurs appartenant à l'intervalle [0.4, 0.6]. Cependant, la fréquence des gènes similaires devient très faible aux bornes avec des seuils faibles (inférieur à $0,2$) ou élevés (supérieur à $0,8$).

5.5.1.2 *Active GA-PPI-Net* : Variation du seuil d'interaction

Dans cette section, nous analysons les résultats en termes de variation du seuil d'interaction ∇I. Nous avons fait varier la valeur de seuil ∇I de 0.5 à 0.9 et nous avons fixé $\nabla S = 0.5$. Nous retenons, comme dans la série d'expérimentations précédentes, les valeurs de la fitness maximale, la fitness



(a) Distribution de la valeur de Similarité. (b) Distribution de la valeur d'interaction .

FIGURE 5.6 – Distribution valeur de Similarité et d'interaction dans le graphe PPI humain.

moyenne,la fitness médiane, le temps d'exécution en minutes et la taille du meilleur individu, à la fin de chaque test. Pour réaliser cette étude, nous avons utilisé la combinaison de paramètres de AG présentée dans le tableau 5.3 juste une modification de seuils. Le tableau 5.5 présente les résultats retenus par cette expérimentation.

TABLE 5.5 – Évolution de la fitness selon la valeur d'interaction ∇I.

∇I	**Fitness Max**			**Fitness Moyen**		
	Best	AVG	Median	Best	AVG	Median
0,5	486,49	467,90	468,78	374,23	358,13	354,81
0,6	570,97	527,38	527,54	410,15	387,79	381,34
0,7	516,33	495,67	498,31	376,38	358,30	356,87
0,8	411,54	371,08	376,89	288,76	274,87	271,77
0,9	338,31	352,71	344,21	248,60	244,23	242,50

Le tableau 5.5 montre qu'Active *GA-PPI-Net* a des valeurs acceptables de fitness sur la plupart des variations de seuil ∇I. Les valeurs de fitness de notre méthode sont comprises entre 338.31 et 570.97, ce qui signifie qu'il existe une détection significative de communauté de gènes qui sont en interaction.

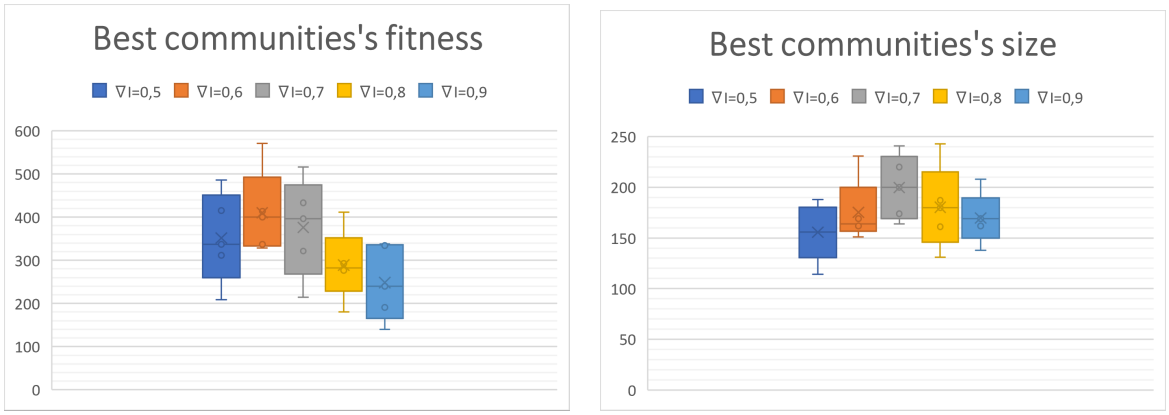
Contrairement aux résultats obtenus dans le tableau 5.4 avec la variation de la similarité, la fitness des meilleures communautés n'évalue pas proportionnellement à l'évolution de ∇I. En effet, la meilleure performance avec $\nabla I = 0,5$ est de $486,49$, elle augmente avec $\nabla I = 0,6$ et baisse à $411,54$

avec $\nabla I = 0,8$ et à $338,31$ avec $\nabla I = 0,9$. Deux grandes constatations peuvent être faites :

— La distribution des interactions n'est pas la même que celle de des similarités (voir figure 5.6. Le graphe PPI présente un grand nombre d'arcs avec des interactions faibles ($< 0,4$) ou moyenne ($< 0,6$), mais on note aussi un grand nombre de gènes en forte interaction ($> 0,9$), Ce qu'explique la taille des communautés détectées par *Active GA-PPI-Net* pour $\nabla I = 0.9$.

— Le champ d'exploration d'*Active GA-PPI-Net* est moins constraint avec l'augmentation de seuil d'interaction qu'avec l'augmentation de seuil de la similarité.

La figure 5.7 présente l'évolution de la valeur du fitness en faisant varier ∇I. On constate qu'*Active GA-PPI-Net* obtient un bon résultat si $\nabla I >= 0.5$ pour déterminer des communautés de gènes qui sont en interaction.



Évolution de la fitness Évolution de la taille

FIGURE 5.7 – Evolution de la fitness et de la taille des meilleures communautés en variant ∇I entre 0.9 et 0.5

Cependant, les réductions de l'espace de recherche en variant ∇S et ∇I ne suivent pas les mêmes courbes. En effet, la réduction de l'espace de recherche est beaucoup plus rapide en augmentant ∇S qu'en augmentant ∇I. Ceci est du essentiellement à la distribution générale de ces seuils sur le graphe illustrés dans la figure 5.6 (b). En effet, la majorité des arcs du graphe PPI ont des seuils d'interaction inférieure à 0.22.

Active GA-PPI-Net détecte et construit des communautés avec des valeurs de fitness (basé sur la similarité et/ou l'interaction) élevées, ce qui signifie que cette approche obtient une détection significative avec une qualité élevée.

5.5.2 Évaluation biologique des communautés de gènes obtenues

Dans cette section, nous proposons de valider la performance d'*Active GA-PPI-Net* et vérifier leur faisabilité, nous avons conduit nos expérimentations sur deux fronts : i) Dans le premier, nos tests consistent d'abord à montrer la performance de notre approche à reconstruire les communautés de gènes d'entrée. ii) Dans le second, nos tests visent à déterminer si les communautés de gènes trouvées par notre approche apparaissent dans la base de KEGG ou dans d'autres bases de voies biologiques. Rappelons que ces communautés peuvent être similaires (seuil de similarité élevé) ou en interaction (seuil d'interaction élevé) ou les deux. Nous avons expérimenté notre approche en suivant le protocole expérimental suivant :

— Utiliser la combinaison de paramètres décrite dans le tableau 5.1

— Exécuter *Active GA-PPI-Net* 20 fois, pour chaque type de communautés, avec des gènes choisis aléatoirement parmi les cinq jeux de données présenté dans le tableau 3.2 pour initialiser la population initiale $P0$.

— Utiliser le réseau PPI Homo sapiens pour créer des échantillons. Ces derniers servent lors de l'évolution de l'AG et à varier les données.

— Effectuer des tests avec différentes valeurs de ∇S et ∇I et retenir à chaque fois la meilleure communauté. Donc, nous avons 20 meilleures communautés pour chaque type.

— Présenter chaque communauté obtenue à l'outil DAVID (Jiao et al., 2012). Ce dernier permet de comparer les communautés obtenues avec d'autres dans différentes bases de données et donne le pourcentage minimal et maximal de recouvrement dans des communautés connues et existantes.

La première évaluation consiste à évaluer la performance et la capacité d'*Active GA-PPI-Net* à reconstruire des communautés avec les gènes initiaux de jeux de données présentés dans le tableau 3.2. Les communautés obtenues sont analysées et comparées selon ∇S et ∇I. Le tableau 5.6 présente les résultats de comparaison donnés par l'outil DAVID.

Le tableau 5.6 fournit le taux de recouvrement minimal et maximal donné, par l'outil DAVID pour chaque type de communautés détecté par rapport aux jeux de données initiaux. On observe que *Active GA-PPI-Net* permet reconstruire des "parties" de communautés réelles et dans certains cas à des communautés réelles entières. Ces communautés obtenues correspondent aux jeux de données

TABLE 5.6 – Évaluation des communautés détectées par *Active GA-PPI-Net* par rapport aux jeux de données de départ).

Datasets	*Active GA-PPI-Net*					
	SIM groups		INT groups		SIM & INT communities	
	%Min	%Max	%Min	%Max	%Min	%Max
Apoptosis	20%	80%	10%	100%	40%	100%
B cell receptor signaling pathway	23%	100%	12%	100%	30%	100%
Purine metabolism	30%	60%	11%	70%	30%	75%
RNA degradation	30%	100%	25%	50%	35%	100%
Oocyte meiosis	25%	40%	20%	45%	30%	100%

initiaux. La troisième colonne du tableau représente le type des communautés basées sur la similarité et l'interaction, le pourcentage de 100% de recouvrement est atteint pour quatre communautés connues et existantes parmi les cinq jeux de données initiaux. Par contre, si la fonction de fitness est basée sur des valeurs de similarité uniquement ou d'interaction uniquement, notre approche reconstruit deux communautés connues parmi les cinq jeux de données.

La deuxième évaluation sert à vérifier dans quelle mesure *Active GA-PPI-Net* est susceptible de retrouver dans le graphe PPI des communautés de gènes connues et réelles existantes dans des bases de voies biologiques comme KEGG Pathway, Biocarta, etc. Le tableau 5.7 présente les résultats de comparaison de l'outil DAVID pour calculer le pourcentage minimal et le maximal de taux de recouvrement.

TABLE 5.7 – Évaluation de communautés détectées par *Active GA-PPI-Net*.

DBs	*Active GA-PPI-Net*					
	SIM groups		INT groups		SIM & INT communities	
	$\nabla S >= 0.5, \nabla I < 0.4$		$\nabla S < 0.5, \nabla I >= 0.4$		$\nabla S >= 0.5, \nabla I >= 0.4$	
	%Min	%Max	%Min	%Max	%Min	%Max
BBIB	10%	70%	10%	70%	14%	100%
Biocarta	10%	46%	20%	45%	25%	100%
EC Number	30%	76%	11%	16%	20%	77%
KEGG	25%	100%	25%	100%	25%	100%
Reactome	30%	100%	11%	100%	30%	100%

Le tableau 5.7 décrit le taux de recouvrement pour toutes les communautés obtenues par ce cas de test. Nous constatons que *Active GA-PPI-Net* atteint le pourcentage 100% lorsque la fonction de

fitness est basée à la fois sur des valeurs de similarité et d'interaction pour les bases BBID, Biocarta, EC Number, Reactome et KEGG. Et, il atteint le pourcentage 100% lorsque la fonction fitness est basée sur la similarité sémantique ou l'interaction, respectivement, mais cette fois avec KEGG et Reactome. Ces résultats constituent une validation en terme biologique. Avec la base EC Number, on constate que le taux de recouvrement est de 76% pour les communautés basées sur la similarité et il baisse jusqu'à 16% pour les communautés basées sur l'interaction. cela pouvait s'expliquer par le fait que certaines communautés dans les voies biologiques de notre étude ont une mesure de similarité ou d'interaction élevée. Pour conclure, nous constatons que *Active GA-PPI-Net* a pu reconstruire efficacement des communautés existantes dans de véritables bases de données de voies biologiques, ce qui montre la pertinence de la fonction fitness et de la méthode d'échantillonnage adaptative proposée.

5.6 Conclusion

Dans ce chapitre, nous avons présenté l'extension *Active GA-PPI-Net* qui permet de construire et de détecter des communautés de gènes de différentes longueurs à partir de données volumineuses. L'algorithme *Active GA-PPI-Net* permet : i) La mise à l'échelle de l'approche *Generic GA-PPI-Net*. ii) L'amélioration de l'espace d'exploration à travers l'intégration de nouvelles données (à travers l'utilisation des échantillons après un nombre de générations défini). iii) L'optimisation du travail de l'opérateur de la mutation OCM en réduisant l'espace d'exploration. iv) Le renforcement de la généricité de *Generic GA-PPI-Net* pour détecter de communautés avec des seuils variables. v) La validation de communautés obtenues en utilisant l'outil DAVID. Ceci a permis de montrer la performance d'*Active GA-PPI-Net* à trouver des communautés réels connus et existants dans les bases de données publiques de référence KEGG, Biocarta, Reactome, etc. Cependant, une étude complémentaire sur les paramètres des AGs s'impose afin d'étudier les irrégularités sur le temps de calcul et d'optimiser la convergence de l'algorithme.

Conclusion générale

Sommaire

Bilan

Le travail réalisé dans le cadre de cette thèse s'inscrit dans le domaine de la détection de communautés dans les réseaux complexes. Nous nous sommes particulièrement intéressés à la détection de communautés d'interaction de gènes chez l'être humain à partir des sources d'annotation en se basant sur l'apprentissage automatique. Pour réaliser ce travail, nous avons combiné trois niveaux d'informations : i) Niveau sémantique : informations contenues dans les ontologies biologiques (par exemple Gene Ontology) (Ashburner et al., 2000) et la similarité sémantique entre des paires de gènes (Ruths et al., 2009) ii) Niveau fonctionnel : informations contenues dans des bases de données publiques qui décrivent les interactions entre les gènes (Snel et al., 2000) et iii) Niveau réseau : informations contenues dans les bases de voies biologiques (Kanehisa and Goto, 2000)
Notre travail comporte quatre phases. La première phase se centralise sur l'extraction des données existant dans l'ontologie Gene Ontology (GO), la base de données publique d'interactions de gènes string-dB et la base de voies biologique KEGG utiles pour notre projet. Ainsi, nous nous intéressons à l'étude de la similarité sémantique entre des groupes de gènes qui sont annotés par des termes de l'ontologie Gene Ontology. Cette similarité est considérée comme l'une de caractéristique d'une communauté de gène. Il existe plusieurs approches qui permettent de comparer des ensembles de termes d'une ontologie afin de quantifier la similarité entre ces ensembles. La mesure de similarité GS2 a été adoptée pour notre travail. La deuxième phase consiste à proposer une approche pour détecter des communautés de gènes de longueur variable, chez l'être humain, qui sont en interaction et sont similaires sémantiquement. Il existe plusieurs approches pour détecter des communautés à partir d'un réseau, parmi lesquelles les méthodes basées sur les algorithmes génétiques. Nous avons proposé une approche nommée *GA-PPI-Net*. Nous avons défini pour cela une représentation de solution spécifique et une fonction fitness basée sur le calcul d'une mesure communautaire. Cette mesure est basée sur une mesure topologique (= score d'interaction entre les gènes) et une mesure sémantique (= similarité sémantique entre deux gènes). Nous avons proposé également un opérateur de mutation spécifique nommé OCM. Cet opérateur a le rôle de renforcer les scores d'interaction et de similarité sémantique au sein d'une solution. Pour cela, une fonction score appliquée à un gène a été proposée pour estimer sa qualité au sein de la communauté à laquelle il appartient. Ce score nous aidera à détecter le gène ayant le meilleur score dans une communauté. Le principe

de cet opérateur est d'introduire dans la communauté un nouveau gène en interaction avec Best-Gene, ce nouveau gène remplacera le gène ayant le plus mauvais score. Pour valider cette approche, nous avons réalisé une étude comparative avec des méthodes analytiques les plus connues comme MCL, RNSC et ClusterOne ainsi avec une méthode de clustering évolutionnaire EGCPI. Toutes ces méthodes ont été appliquées sur des réseaux biologiques. Dans la troisième phase, nous avons proposé des améliorations et des extensions de *GA-PPI-Net*. La première vise à proposer un nouvel opérateur de mutation adaptatif OCM2. Cet opérateur a pour objectif d'optimiser l'exploration de l'espace de recherche afin d'améliorer les qualités des communautés obtenues. De plus, nous avons rendu notre approche générique pour construire des communautés de gènes de différents types : i) communauté de gènes sémantiquement similaires, ii) communauté de gènes en interaction et iii) ou les deux à la fois. Trois nouvelles composantes spécifiques à l'AG ont été introduites : i) une nouvelle solution adéquate pour représenter une communauté de taille dynamique, ii) une fonction de fitness générique basée sur une mesure de similarité et une valeur d'interaction entre les gènes en fonction des valeurs de seuils et iii) un opérateur de croisement heuristique pour renforcer les liens dans les communautés. L'AG proposé a été paramétré en fonction de l'importance affectée à chaque critère de mesure (mesure sémantique et mesure d'interaction). Finalement, dans notre dernière contribution, nous avons mis à l'échelle notre approche *Generic GA-PPI-Net* en utilisant le paradigme d'apprentissage actif. Cette approche nous a permis d'utiliser des sources de données volumineuses (la totalité de réseau d'interaction de protéine chez l'être humain) pour construire des communautés évolutives. Elle est basée sur une approche d'échantillonnage adaptative permettant de définir la partie du graphe à explorer par les opérateurs génétiques pendant l'évolution.

Les conclusions que l'on peut tirer de nos expériences sont de trois ordres. D'abord, l'algorithme génétique présente un paradigme intéressant pour la détection de communautés dans les graphes. Ensuite, le choix des valeurs des différents paramètres s'avère très important pour la performance. Enfin, les opérateurs génétiques proposés (croisement heuristique et mutation) apporte une amélioration significative comparée à d'autres méthodes analytiques.

Perspectives

Les travaux et résultats présentés dans ce manuscrit ouvrent plusieurs directions de recherches, soit pour approfondir certains travaux entamés ici, soit pour explorer de nouvelles perspectives. Nous les décrivons dans les points qui suivent.

À court terme :

- Combiner les méthodes de clustering et les algorithmes génétiques pour détecter des communautés de gènes.
- Prendre en compte la modularité, comme le font certains AGs pour la détection de communauté. Plus généralement, il serait intéressant d'étudier d'autres critères et mesures de la qualité d'une communauté de gènes.

À plus long terme :

- Adapter les différentes approches proposées sur d'autres réseaux comme les réseaux sociaux, par exemple Facebook.
- Étudier d'autres approches d'échantillonnage, comme l'échantillonnage hiérarchique (multi-niveaux) ou incrémental dans le cadre du passage à l'échelle.
- Intégrer dans la mesure de similarité les deux autres sous-ontologies de Gene Ontology, à savoir Cellular Components (CC) et Molecular Functions (MF).
- Intégrer nos approches dans une plateforme d'aide à la décision pour les biologistes. Cela permettrait d'exploiter nos résultats, c'est-à-dire les communautés de gènes d'intérêt détectées, de façon concrète.

Bibliographie

Alonso-López, D., Gutiérrez, M. A., Lopes, K. P., Prieto, C., Santamaría, R., and De Las Rivas, J. (2016). APID interactomes : providing proteome-based interactomes with controlled quality for multiple species and derived networks. *Nucleic Acids Research*, 44(W1) :W529–W535.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology : tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25(1) :25–29.

Atay, Y., Koc, I., Babaoglu, I., and Kodaz, H. (2017). Community detection from biological and social networks : A comparative analysis of metaheuristic algorithms. *Applied Soft Computing*, 50 :194–211.

Bard, J. B. L. and Rhee, S. Y. (2004). Ontologies in biology : design, applications and future challenges. *Nat Rev Genet*, 5(3) :213–222.

Barnes, E. (1982). An Algorithm for Partitioning the Nodes of a Graph. *SIAM. J. on Algebraic and Discrete Methods*, 3(4) :541–550.

Ben M'barek, M., Borgi, A., Bedhiafi, W., and Ben Hmida, S. (2018). Genetic Algorithm for Community Detection in Biological Networks. *Proceedings Computer Science*, 126 :195–204. Knowledge-Based and Intelligent Information Engineering Systems : Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

Ben M'barek, M., Borgi, A., Ben Hmida, S., and Rukoz, M. (2019). Genetic algorithm to detect different sizes' communities from protein-protein interaction networks. In *Proceedings of the 14th International Conference on Software Technologies - Volume 1 : ICSOFT,*, pages 359–370. INSTICC, SciTePress.

Ben M'Barek., M., Borgi., A., Ben Hmida., S., and Rukoz., M. (2020). Generic GA-PPI-Net : Generic evolutionary algorithm to detect semantic and topological biological communities. In *Proceedings of the 15th International Conference on Software Technologies - ICSOFT,*, pages 295–306. INSTICC, SciTePress.

Ben M'barek, M. B., Hmida, S. B., Borgi, A., and Rukoz, M. (2021). GA-PPI-Net approach vs analytical approaches for community detection in PPI networks. *Procedia Computer Science*, pages 903–912.

Bornholdt, S. and Schuster, H. G. (2003). *Handbook of Graphs and Networks : From the Genome to the Internet*. John Wiley & Sons, Inc., New York, NY, USA.

Briche, J. (2009). *Adaptation d'un algorithme génétique pour la reconstruction de réseaux de régulation génétique : COGARE.* phdthesis, Université du Sud Toulon Var.

Brohee, S. and Van Helden, J. (2006). Evaluation of clustering algorithms for protein-protein interaction networks. *BMC bioinfo*, 7(1) :488.

Brouard, C. (2013). *Inférence de réseaux d'interaction protéine-protéine par apprentissage statistique*. PhD thesis.

Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A., and Apweiler, R. (2003). The Gene Ontology Annotation (GOA) Project : Implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Res*, 13(4) :662–672.

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R., and Apweiler, R. (2004). The Gene Ontology Annotation (GOA) Database : sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res*, 32(Database issue) :D262–D266.

Chen, M., Kuzmin, K., and Szymanski, B. K. (2014). Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, 1(1) :46–65.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, 15(2) :201–221.

Collet, P., Lutton, E., Raynal, F., and Schoenauer, M. (2000). Polar ifs+ parisian genetic programming= efficient ifs inverse problem solving. *Genetic Programming and Evolvable Machines*, 1(4) :339–361.

Croft, D., O'Kelly, G., Wu, G., Haw, R., Gillespie, M., Matthews, L., Caudy, M., Garapati, P., Gopinath, G., Jassal, B., Jupe, S., Kalatskaya, I., Mahajan, S., May, B., Ndegwa, N., Schmidt, E., Shamovsky, V., Yung, C., Birney, E., Hermjakob, H., D'Eustachio, P., and Stein, L. (2011). Reactome : a database of reactions, pathways and biological processes. *Nucleic Acids Res.*, 39(Database issue) :D691–697.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* University of Michigan.

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley, Chichester, West Sussex, U.K, 5 edition edition.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5) :75–174. arXiv : 0906.0612.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *PNAS*, 104(1) :36–41.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.*, 99(12) :7821–7826.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann.

Grover, A. and Leskovec, J. (2016). node2vec : Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Guo, X., Liu, R., Shriver, C. D., Hu, H., and Liebman, M. N. (2006). Assessing semantic similarity

measures for the characterization of human regulatory pathways. *Bioinformatics*, 22(8) :967–973.

Hafez, A. I., Ghali, N. I., Hassanien, A. E., and Fahmy, A. A. (2012). Genetic Algorithms for community detection in social networks. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 460–465.

He, T. and Chan, K. C. (2016). Evolutionary graph clustering for protein complex identification. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(3) :892–904.

He, T. and Chan, K. C. C. (2018). Evolutionary graph clustering for protein complex identification. *IEEE/ACM Trans on Comp Biology and Bioinfo*, 15(3) :892–904.

Holland, J. H. (1975). *Adaptation in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence*, volume viii. U Michigan Press, Oxford, England.

Jangda, A., Polisetty, S., Guha, A., and Serafini, M. (2021). Accelerating graph sampling for graph machine learning using gpus. In *Proceedings of the Sixteenth European Conference on Computer Systems*, pages 311–326.

Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833) :41–42.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A.-L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804) :651–654.

Jiang, J. J. and Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *arXiv :cmp-lg/9709008*. arXiv : cmp-lg/9709008.

Jiao, X., Sherman, B. T., Huang, D. W., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2012). DAVID-WS : a stateful web service to facilitate gene/protein list analysis. *Bioinformatics*, 28(13) :1805–1806.

Kanehisa, M. and Goto, S. (2000). KEGG : kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28(1) :27–30.

Kernighan, B. W. and Lin, S. (1970). An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, 49(2) :291–307.

King, A. D., Pržulj, N., and Jurisica, I. (2004). Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17) :3013–3020.

Kulik, L., Duckham, M., and Egenhofer, M. (2005). Ontology-driven map generalization. *Journal of Visual Languages & Computing*, 16(3) :245–267.

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann.

Liu, X., Li, D., Wang, S., and Tao, Z. (2007). Effective algorithm for detecting community structure in complex networks based on ga and clustering. In *International Conference on Computational Science*, pages 657–664. Springer.

McGraw-Hill Concise Dictionary of Modern Medicine (2002). gene product. Page Version ID : 117170239.

M'barek, B., Borgi, A., Ben Hmida, S., and Rukoz, M. (2019). GA-PPI-Net : A genetic algorithm for community detection in protein-protein interaction networks. In *International Conference on Software Technologies*, pages 133–155. Springer.

National Human Genome Research Institute (NHGRI) (2015). Biological Pathways Fact Sheet.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature methods*, 9(5) :471.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6). arXiv : cond-mat/0309508.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U.S.A.*, 103(23) :8577–8582.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2). arXiv : cond-mat/0308217.

Nishimura, D. (2001). BioCarta. *Biotech Software & Internet Report*, 2(3) :117–120.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk : Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Pesquita, C., Faria, D., Falcão, A. O., Lord, P., and Couto, F. M. (2009). Semantic Similarity in Biomedical Ontologies. *PLoS Comput Biol*, 5(7).

Petrowski, A. and Ben-Hamida, S. (2017). *Evolutionary Algorithms*. John Wiley & Sons. Google-Books-ID : fvRRCgAAQBAJ.

Pizzuti, C. (2008). Ga-net : A genetic algorithm for community detection in social networks. In *International conference on parallel problem solving from nature*, pages 1081–1090. Springer.

Pizzuti, C. (2018). Evolutionary Computation for Community Detection in Networks : A Review. *IEEE Transactions on Evolutionary Computation*, 22(3) :464–483.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1) :17–30.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and identifying communities in networks. *PNAS*, 101(9) :2658–2663.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *science*, 297(5586) :1551–1555.

Resnik, P. (2011). Semantic Similarity in a Taxonomy : An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *arXiv :1105.5444 [cs]*. arXiv : 1105.5444.

Rogers, M. F. and Ben-Hur, A. (2009). The use of gene ontology evidence codes in preventing classifier assessment bias. *Bioinformatics*, 25(9) :1173–1177.

Ruths, T., Ruths, D., and Nakhleh, L. (2009). GS2 : an efficiently computable measure of GO-based similarity of gene sets. *Bioinformatics*, 25(9) :1178–1184.

Sherman, B. T., Huang, D. W., Tan, Q., Guo, Y., Bour, S., Liu, D., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2007). DAVID Knowledgebase : a gene-centered database integrating heterogeneous gene annotation resources to facilitate high-throughput gene functional analysis. *BMC Bioinformatics*, 8 :426.

Snel, B., Lehmann, G., Bork, P., and Huynen, M. A. (2000). STRING : a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucl. Acids Res.*, 28(18) :3442–3444.

Tasgin, M. and Bingol, H. (2006). Community Detection in Complex Networks using Genetic Algorithm. *arXiv :cond-mat/0604419*. arXiv : cond-mat/0604419.

Teng, X., Liu, J., and Li, M. (2019). Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm. *IEEE transactions on cybernetics*, 51(1) :138–150.

Thabet Slimani and Ben Yaghlane, B. (2007). Une extension de mesure de similarité entre les concepts d'une ontologie. *Sciences of Electronic, Technologies of Information and Telecommunications*.

Van Dongen, S. M. (2000). *Graph clustering by flow simulation*. PhD thesis, Utrecht University Repository.

Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. (2007). A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23(10) :1274–1281.

Wu, Z., Liao, Q., and Liu, B. (2020). A comprehensive review and evaluation of computational methods for identifying protein complexes from protein–protein interaction networks. *Briefings in bioinfo*, 21(5) :1531–1548.

Wu, Z. and Palmer, M. (1994). Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhang, J., Silvescu, A., and Honavar, V. (2002). Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. In Koenig, S. and Holte, R. C., editors, *Abstraction, Reformulation, and Approximation*, number 2371 in Lecture Notes in Computer Science, pages 316–323. Springer Berlin Heidelberg.

Zhao, Y., Dong, J., and Peng, T. (2009). Ontology Classification for Semantic-Web-Based Software Engineering. *IEEE Transactions on Services Computing*, 2(4) :303–317.

RÉSUMÉ

Dans le cadre de cette thèse de doctorat, nous nous intéressons à la détection des communautés de gènes dans les réseaux d'interactions de protéine-protéine. Ces communautés correspondent à des ensembles de gènes qui collaborent à une même fonction cellulaire. Notre objectif consiste à déterminer un groupe ou une communauté de gènes à partir des sources d'annotation en se basant sur l'apprentissage automatique. Pour réaliser ce travail, nous combinons trois niveaux d'informations : i) le niveau sémantique : information contenue dans les ontologies biologiques (gene Ontology), ii) le niveau fonctionnel : information contenue dans des bases de données publiques qui décrivent les interactions des gènes et iii) le niveau réseau : informations contenues dans les bases de voies biologiques. Ce travail est multidisciplinaire, à l'intersection de domaine de l'informatique et de la biologie et il comporte quatre volets.

Le premier volet se concentre sur l'extraction des données biologiques utiles pour notre projet et sur l'étude de la similarité sémantique entre des groupes de gènes. Cette dernière sera l'une de caractéristique d'une communauté de gènes. Nous avons proposé, dans le deuxième volet, une approche pour la détection des communautés de gènes basée sur les algorithmes génétiques. Cette approche nommée *GA-PPI-Net* permet de construire et de détecter des communautés de gènes de tailles variables. *GA-PPI-Net* permet de maximiser une mesure communautaire qui combine à la fois des informations topologiques entre les gènes et des informations sémantiques. Par ailleurs, nous avons introduit une solution spécifique pour représenter une communauté (=solution) de taille variable et un opérateur de mutation optimisée. Dans le troisième volet, nous nous focalisons sur l'extension et l'amélioration de *GA-PPI-Net*. La première extension sert à proposer un nouvel opérateur de mutation adaptatif. Cette amélioration a pour objectif d'optimiser l'exploration de l'espace de recherche afin d'améliorer les qualités des communautés obtenues. La deuxième amélioration vise à rendre notre approche générique, *Generic GA-PPI-Net,* pour construire des communautés de gènes qui sont sémantiquement similaires et/ou sont en interaction. Dans le dernier volet, nous avons étendu ce travail afin de mettre à l'échelle *Generic GA-PPI-Net* en utilisant le paradigme d'apprentissage actif. Cette extension nous a permis d'utiliser des sources de données volumineuses (la totalité de réseau d'interaction de protéine chez l'être humain) pour construire des communautés évolutives. Elle est basée sur une approche d'échantillonnage adaptative permettant de définir la partie du graphe à explorer par les opérateurs génétiques pendant l'évolution.

MOTS CLÉS

Détection de communautés, Gene Ontology, réseau PPI, interaction de gènes, similarité sémantique, algorithmes génétiques, mise à l'échelle, apprentissage actif.

ABSTRACT

In our work, we are interested in the communities' detection in protein-protein interaction networks (PPI). These communities give us an idea about the perception of the network's structure. One of the goals in biology is to determine how genes or proteins encode function in the cell. This work is multidisciplinary, as it brings the field of biology and computer science in the broad sense. Thus, our objectif is to find communities of genes having a biological sense (that participate in the same biological processes or that perform together specific biological functions) from gene annotation sources. To make this task, we have combined three levels of information : i) Semantic level: information contained in biological ontologies such as Gene Ontology (GO) and information obtained by the use of a similarity measure such as GO-based similarity of gene sets (GS2). It assesses the semantic similarity between genes, ii) Functional level: information contained in public databases describing the interactions of genes iii) Networks level: information contained in pathway databases. Our work has four parts.

The first part focuses on the extraction of biological data used in our project. Thus, we study the semantic similarity between groups of genes that are annotated by terms of biological ontology. It is one of the characteristics of a gene community. The second part present the proposed approach GA-PPI-Net for the detection of gene communities. It is a Genetic Algorithm based approach to detect communities having different sizes from PPI networks. For this purpose, we use a fitness function based on a similarity measure and the interaction value between proteins or genes. Moreover, a specific solution for representing a community and a specific mutation operator are introduced. The third part presents two extensions of GA-PPI-Net. The first one proposes a specific adaptive mutation operator. The second aims to make GA-PPI-Net generic by allowing finding different sizes of communities based on the interaction and/or similarity criterion. This approach called Generic GA-PPI-Net. Finally, we propose to scale Generic GA-PPI-Net using the active learning paradigm. This approach allowed us to use a large data sets (the whole humain PPI) to build evolutionary communities. It is based on an adaptive sampling approach to define the part of the graph to be explored by the genetic operators during the GA evolution.

KEYWORDS

Community detection, Gene Ontology, PPI networks, semantic similarity, gene interaction, Genetic Algorithm, active learning