NNT/NL : 2022AIXM0391/109ED62

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 13 octobre 2022 par

Anthony Baptista

Modèles multi-couches et méthodes d'exploration de réseaux biologiques

Discipline
Biologie santé

Spécialité
Génomique et Bioinformatique

École doctorale
ED 62 - Sciences de la vie et de la santé

Laboratoire/Partenaires de recherche
UMR 1251 - MMG - Marseille Medical Genetics
UMR 1090 - TAGC - Theories and Approaches of Genomic Complexity
CENTURI - Turing Center For Living Systems

Composition du jury

Sophie DONNET Chargée de recherche - INRAE	Rapportrice
Fabrizio DE VICO FALLANI Chargé de recherche - INRIA	Rapporteur
Jean-Philippe VERT Directeur de recherche - Google Brain / Mines ParisTech	Examinateur
Alain BARRAT Directeur de recherche - CPT	Examinateur
Anaïs BAUDOT Directrice de recherche - MMG	Directrice de thèse
Aitor GONZALEZ Maître de conférences - TAGC	Codirecteur de thèse

Affidavit

Je, soussigné, Anthony Baptista, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique d'Anaïs Baudot et Aitor Gonzalez, dans le respect des principes d'honnêteté, d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect à la fois de la charte nationale de déontologie des métiers de la recherche et de la charte d'Aix-Marseille Université relative à la lutte contre le plagiat.

Ce travail n'a pas été précédemment soumis en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Aix-en-Provence le 26 septembre 2022

Liste de publications et participation aux conférences

Liste des publications réalisées dans le cadre du projet de thèse :

1. Universal Multilayer Network Exploration by Random Walk with Restart, A. Baptista, A. Gonzalez, A. Baudot, *Communications Physics* , Vol. 5, No. 1 (2022).
2. Biological Applications of Universal Multilayer Networks, A. Baptista and A. Baudot, en préparation (2022).
3. Zoo Guide for Network Embedding, A. Baptista and A. Baudot, en préparation (2022).

Participation aux conférences et écoles d'été au cours de la période de thèse :

1. Programme transversal INSERM Variabilité Génomique (GOLD), Brest, November 6-7, 2019
2. CIRM winter school : Networks and molecular Biology, Marseille, March 2-6, 2020 : **Présentation d'un poster**.
3. JOBIM 2020, remote conference, June 30-July 06, 2020.
4. ECCB 2020, 19^{th} European Conference on Computational, remote conference, August 31-September 8, 2020.
5. Satellites of NetSci : Networks in Biology and Medicine, NetBioMed 2020, remote conference, September 17, 2020 : **Présentation d'un poster et d'une présentation orale**.
6. CENTURI Scientific day : PhD / Postdoc day : *Multi-Layer network exploration by Random walk with Restart*, September 24, 2020 : **Présentation d'un poster**.
7. NetSci 2020, remote conference, September 17-25, 2020.
8. JOBIM 2021, remote conference, July 06-09, 2021.
9. Programme transversal Variabilité Génomique (GOLD), Nantes, October 25-26, 2021 : **Présentation orale**.
10. Journées du GDR BIM 2021, Lyon, November 23-24, 2021 : **Présentation orale**.

Table des matières

Table des figures

9.3. La courbe représente la longueur de convergence (ordre de troncature k de la série), en fonction de la valeur de α. La valeur de convergence est obtenue quand la corrélation de Spearman, moyenne entre deux matrices de similarité de Katz issues de deux troncatures successives, est strictement supérieure ou égale à 99 %. Le zoom correspond à une matrice de corrélation de Spearman moyenne obtenue pour une faible valeur de α, valeur associée à une longueur de convergence égale à quatre. Chaque élément de la matrice est calculé à partir de la corrélation de Spearman moyenne entre deux matrices de similarité de Katz, issues de deux troncatures de valeur k. La barre de couleur indique l'intensité de la corrélation, très forte en jaune (près de 1), moins forte en bleu foncé. 160

Liste des tableaux

Résumé

La quantité de données, ainsi que leur variété et leur hétérogénéité, augmentent, et ce, depuis de nombreuses années. Cette disponibilité des données à grande échelle représente une opportunité sans précédent pour mieux comprendre les systèmes complexes. Parmi les modes de représentation de données, les réseaux apparaissent comme particulièrement couronnés de succès. En effet, il existe une grande variété d'outils provenant de la théorie des graphes pour les explorer et en extraire des connaissances pertinentes. Cependant, l'exploration de grands jeux de données multi-dimensionnelles demeure un défi important dans de nombreux domaines. Par exemple, en bioinformatique, l'étude des systèmes biologiques nécessite parfois l'intégration de dizaines de jeux de données différents. Les réseaux multi-couches apparaissent dans ce contexte comme un outil prometteur pour la représentation et l'analyse de ces données biologiques. L'extension récente des méthodes d'exploration de réseaux permet de tirer profit de ces formalismes multi-couches, plus riches et plus complexes. Par exemple, les marches aléatoires ont été étendues aux réseaux multi-couches et sont très utilisées pour explorer la topologie de réseaux à grande échelle. Les marches aléatoires avec *restart* sont un cas particulier de marches aléatoires. Elles permettent de mesurer une similarité entre un nœud donné et les autres nœuds du réseau. Cette stratégie de marches aléatoires avec *restart* offre des performances supérieures aux méthodes classiques basées sur des mesures locales, en particulier dans le cas de la prédiction d'associations entre gènes et maladies. Cependant, les méthodes actuelles sont limitées par le nombre et la variété de combinaisons de réseaux qu'elles peuvent explorer. Par conséquent, de nouvelles méthodes analytiques et numériques doivent être développées, afin de faire face à l'augmentation de la diversité et de la complexité des réseaux multi-couches.

Dans le cadre de ma thèse, je propose un nouveau formalisme mathématique, associé à une librairie Python nommée MultiXrank, pour intégrer et explorer n'importe quelles combinaisons de réseaux. Le formalisme et l'algorithme sont généraux et conviennent aux réseaux hétérogènes et multiplexes, dirigés ou pondérés. J'ai également appliqué cette nouvelle approche à plusieurs questions biologiques, telles que la priorisation de gènes et de médicaments, candidats pour être impliqués dans différentes pathologies, la prédiction d'associations entre gènes et maladies, ainsi que l'intégration de données de conformation 3D de la chromatine avec des réseaux de gènes et de maladies. Cette dernière application offre de nouvelles pistes pour la détermination des relations de comorbidités.

Au cours de ma thèse, je me suis également intéressé à l'extension d'autres méthodes d'analyse aux réseaux multi-couches. Je me suis notamment intéressé à la générali-

sation de la similarité de Katz aux réseaux multi-couches. J'ai aussi développé une nouvelle approche de détection de communautés. Cette méthode est basée sur les marches aléatoires avec *restart* et permet d'identifier des clusters de nœuds à partir de réseaux multi-couches. Enfin, je me suis intéressé à l'*embedding* de réseaux, en particulier au cas des méthodes du type *shallow embedding*. Dans ce cadre, j'ai réalisé une revue de littérature, littérature soumise à des évolutions constantes et rapides. J'ai aussi développé une méthode d'*embedding* basée sur MultiXrank qui ouvre la porte de l'*embedding* à des réseaux multi-couches plus complexes.

Mots clés : marche aléatoire, réseaux multi-couches, intégration de données, données complexes, réseaux biologiques, *embedding* de réseaux.

Abstract

Data amount, variety, and heterogeneity have been increasing drastically for several years, offering a unique opportunity to better understand complex systems. Among the different modes of data representation, networks appear particularly successful. Indeed, a wide and powerful range of tools from graph theory are available for their exploration. However, the integrated exploration of large multidimensional datasets remains a major challenge in many scientific fields. For instance, in bioinformatics, the understanding of biological systems would require the integrated analysis of dozens of different datasets. In this context, multilayer networks emerged as key players in the analysis of such complex data. Moreover, recent years have witnessed the extension of network exploration approaches to capitalize on more complex and richer network frameworks. Random walks, for instance, have been extended to explore multilayer networks. These kinds of methods are currently used for exploring the whole topology of large-scale networks. Random walk with restart, a special case of random walk, allows to measure similarity between a given node and all the other nodes of a network. This strategy is known to outperform methods based on local distance measures for the prioritization of gene-disease associations. However, current random walk approaches are limited in the combination and heterogeneity of networks they can handle. New analytical and numerical random walk methods are needed to cope with the increasing diversity and complexity of multilayer networks.
In the context of my thesis, I developed a new mathematical framework and its associated Python package, named MultiXrank, that allow the integration and exploration of any combinations of networks. The proposed formalism and algorithm are general and can handle heterogeneous and multiplex networks, both directed and weighted. As part of my Ph. D., I also applied this new method to several biological questions such as the prioritization of genes and drugs candidates for being involved in different disorders, gene-disease association predictions, and the integration of 3D DNA conformation information with gene and disease networks. This last application offers new tracks to unveil disease comorbidities relationships.
During my PhD, I was also interested in the extension of several other methods to multilayer networks. In particular, I generalized the Katz similarity measure to multilayer networks. I also developed a new method of community detection. This new community detection is based on random walks with restart and allows the identification of clusters from multilayer network nodes. Finally, I studied network embedding, especially in the case of shallow embedding methods. In this context, I did a literature review, which is quickly evolving. I also developed a network embedding method based on MultiXrank that opens the embedding to more complex multilayer networks.

Keywords: random walk, multilayer networks, data integration, complex data, biological network, network embedding.

Remerciements

Je tiens en premier lieu, à remercier ma directrice de thèse Anaïs Baudot, pour l'opportunité qu'elle m'a offerte de pouvoir mener ma thèse dans une discipline qui n'était a priori pas la mienne. Je tiens aussi à la remercier pour avoir fait son possible pour que la thèse se déroule le plus normalement possible, malgré un contexte sanitaire particulièrement complexe, et d'avoir toujours offert son expertise et son aide dans mes travaux, y compris le présent document par sa relecture.
Je tiens aussi à remercier mes collègues avec qui j'ai pu collaborer, soit directement, soit par les conseils et discussions qu'ils ont pu m'offrir. Je souhaite nommer Elva Novoa Del Toro, Elisabeth Remy et Alberto Valdeolivas à titre individuel, ainsi que les membres du groupe Mabios à titre collectif, que ce soit les membres venant du laboratoire de l'Istitut de Mathématiques de Marseille (I2M), ou ceux du laboratoire Marseille Medical Genetics (MMG). Je remercie aussi les deux unités qui m'ont accueilli : le MMG et le laboratoire Theories and Approches of Genomic Complexity (TAGC).
Dans un second temps, je remercie les membres de mon comité de thèse, Alain Barrat, Laura Cantini, ainsi que Benoit Ballester, qui m'ont permis d'avoir des retours pertinents sur le travail que je menais, et qui ont veillé au bon déroulement du travail de thèse.
Dans un troisième temps, je tiens à remercier les personnes qui m'ont apporté des conseils et du soutien au cours de ma thèse, que ce soit aussi bien à titre professionnel, que personnel. Je remercie ma compagne Alia Aliou, ma mère, Alexandre Schaeffer et Hugo Le Briero, deux amis proches, ainsi que Lucie Khamvongsa, une amie rencontrée au TAGC.
Dernièrement, je tiens à remercier officiellement les instituts et organismes qui ont financé ma thèse et la publication de mes travaux de recherche. Je remercie le programme de financement de thèse CENTURI, le programme d'Investissements d'Avenir de l'Agence National pour la Recherche (ANR-16-CONV-0001), le fonds pour l'initiative d'excellence de l'université Aix-Marseille et d'A*MIDEX, ainsi que le programme transversal GenOmics variability in heaLth & Disease (GOLD) de l'Inserm.

Avant-propos

La physique a été la voie par laquelle je suis entré en science que ce soit d'un point de vue d'intérêt intellectuel, ou d'un point de vue scientifique. Depuis très jeune, comprendre les systèmes physiques était un enjeu un peu trop sérieux. Je me souviens de l'émerveillement que j'ai eu en découvrant le tableau périodique de Mendeleïev ; cette classification de la matière sous forme de composants fondamentaux semblait offrir les clés de compréhension et de classification de tout ce qui était connu. Je pense qu'à ce moment-là s'est joué un moment fondamental dans ma construction intellectuelle. Les choses de la nature peuvent s'expliquer, se classer, s'appréhender par un nombre fini d'éléments. Cette idée que la nature, le monde en général, est accessible à l'intelligence humaine, et de surcroît à l'aide d'un nombre réduit d'objets est le cœur de la pensée physique. À 10 ans, j'avais décidé d'être physicien.
Au cours de mes études en Physique à Paris, à l'Université Pierre et Marie Curie, actuellement Sorbonne Université, j'ai pu parfaire mes compétences mathématiques et mes connaissances sur les systèmes physiques. Mais ce qui retint mon attention le plus fortement, au cours des premières années, fut le modèle d'Ising, un modèle particulièrement simple qui permet d'expliquer le ferromagnétisme, par le biais d'un constituant possédant deux états, ainsi que des règles de transition d'état respectant l'algorithme de Métropolis. J'étais de nouveau émerveillé par l'ingéniosité dont la physique pouvait faire preuve pour comprendre des systèmes complexes sous une forme particulièrement simple et élégante.
En Master 1 de physique fondamentale, d'autres modèles vinrent m'interpeller : le modèle de Vicsek, qui permet de modéliser les mouvements collectifs ; le recuit simulé, algorithme métaheuristique qui, dans une de ses applications, m'a permis de résoudre le problème du voyageur de commerce au cours d'un projet ; la dynamique moléculaire qui m'a été utile au cours de mes stages au LPTMC, où j'étudiais l'apparition de micro-hétérogénéités au sein de mélanges eau-alcool. Ces mélanges semblent d'un point de vue thermodynamique homogènes. Cependant, lorsque l'on regarde de plus près, à l'aide de spectroscopie des rayons X, on voit apparaître des sous-structures préférentiellement agrégées. Tous ces modèles mentionnés appartiennent à une même sous-discipline de la physique, appelée physique statistique. Ainsi, au cours de mon Master, j'ai appris que le modèle qui m'avait tant intéressé en Licence, lors de mon étude des courbes d'hystérisis dans les matériaux ferromagnétiques, appartenait à la même famille que ceux qui me passionnaient désormais en Master.
Ainsi, au cours de mes études, j'avais trouvé l'approche de la physique qui me convenait le mieux. Cette approche était l'extension naturelle de la thermodynamique et de ses principes. Cette approche acceptait l'hypothèse atomistique et le tableau de

Mendeleïev. Cette approche était avant tout la démarche du physicien, dans ce qu'elle a de plus primaire : la découverte de principe universel. Cette approche traitait les systèmes que l'on qualifiait, jusqu'alors, de systèmes complexes. Cette approche était la physique statistique.
Fort de ce choix, j'ai décidé de faire mes stages de Master au LPTMC avec Aurélien Perrera sur la physique statistique des liquides. Au cours de ces stages, mes travaux m'ont permis de publier mon premier article [1] et de choisir ma spécialité de Master : les systèmes complexes et la biophysique. La biophysique apparaissait alors comme le champ le plus naturel d'application de cette approche qui était devenue la mienne. Au cours de cette seconde année de Master, j'ai pu me familiariser avec bon nombre de nouveaux modèles et le champ des sciences bien étrange et bien enthousiasmant qu'était la biologie. Je pris goût à ces systèmes particulièrement abscons, où les variables cachées sont légions et bien trop nombreuses pour espérer une reproductibilité digne de ce nom. Lors d'un projet numérique, j'appliquais les modèles que je connaissais, notamment la dynamique moléculaire, au phénomène du repliement de protéines. Je me familiarisais aussi avec les approches de *machine learning*, notamment la classification supervisée, afin de classer des images de protéines extraites de mon code de dynamique moléculaire en sous-groupe de protéines, repliées et non repliées. Je fis aussi un stage au MSC dans l'équipe de François Graner, où j'étudiais la polarité des cellules épithéliales lors de leurs migrations dans un canal microfluidique avec obstacle. Je développais des outils numériques qui permirent l'analyse des champs de vitesses, de densité et de polarité, ainsi que, par un jeu algébrique, de faire correspondre des images issues de microscopies de différents grossissements. Je découvris aussi lors de cette formation les idées fortes et universelles qui parsèment la biologie. Parallèlement à ma formation, je continuais mon travail avec Aurélien Perrera au LPTMC et publiais mon second article [2]. Puis, à la fin de mon Master, je décidais de changer de champ d'étude et de commencer une thèse à Marseille avec Anaïs Baudot et Aitor Gonzalez, en théorie des graphes appliquée à l'étude des comorbidités entre maladies rares et maladies communes. L'espoir d'en savoir plus sur la biologie, ces idées universelles, ainsi que d'appliquer mes méthodes apprises au cours de mon parcours en physique, me poussèrent à sortir de mon confort parisien.
A posteriori, je pense que je ne me suis jamais senti autant physicien, et peut-être même senti physicien pour la première fois, qu'au cours de ma thèse. Cela peut s'expliquer par deux raisons principalement. La première est que, souvent, mes collègues biologistes me ramenèrent à ma condition de physicien lorsque l'universel, l'unique, était ce que je recherchais, alors que la biologie aurait plutôt préconisé une approche exhaustive. La seconde est que, lorsque l'on est loin de sa terre, on cultive ses origines : la physique dans mon cas. Dans ce monde étranger où la doctrine de Goethe s'applique, je me sentais plus que jamais le disciple de Newton. Je cherchais le commun dans le multiple, ainsi la physique était, non pas une fin en soi, mais un chemin vers l'universel. Et toute ma jeune carrière scientifique a été guidée par cela. Que ce soit mes intérêts pour les idées universelles et transdisciplinaires, ou le champ d'étude de mon sujet de thèse que je vais exposer : la science des réseaux dans le cadre

de la biologie des systèmes complexes.

Introduction

La théorie des graphes est née dans le but de modéliser un problème bien connu au XVIII^{e} siècle : le problème des sept ponts de Königsberg. La question était de savoir s'il était possible, à partir d'un quartier de la ville de départ choisi librement, de pouvoir trouver un chemin traversant une unique fois chaque pont de la ville. La cartographie de la ville est illustrée en Fig. 0.1. En 1741, Leonhard Euler publie une preuve [3] (écrite en 1736) donnant une solution au problème et démontrant qu'il n'est pas possible de trouver un chemin traversant chacun des sept ponts une unique fois. La preuve se base sur la simplification suivante : à chaque quartier de la ville est associé un nœud (A, B, C, D dans la Fig. 0.1) et chacun des sept ponts est modélisé par une arête (a, b, c, d, e, f, g dans la Fig. 0.1). Ainsi était né ce que l'on appelle aujourd'hui un graphe. Une preuve plus moderne a été donnée en 1873 par Carl Hierholzer [4], qui montre qu'un graphe est eulérien (c'est-à-dire qu'il possède un chemin passant par toutes les arêtes une fois uniquement) si, et seulement si, le nombre de nœuds ayant un degré (nombre d'arêtes associées à chaque nœud) impair est égal à zéro ou à deux.

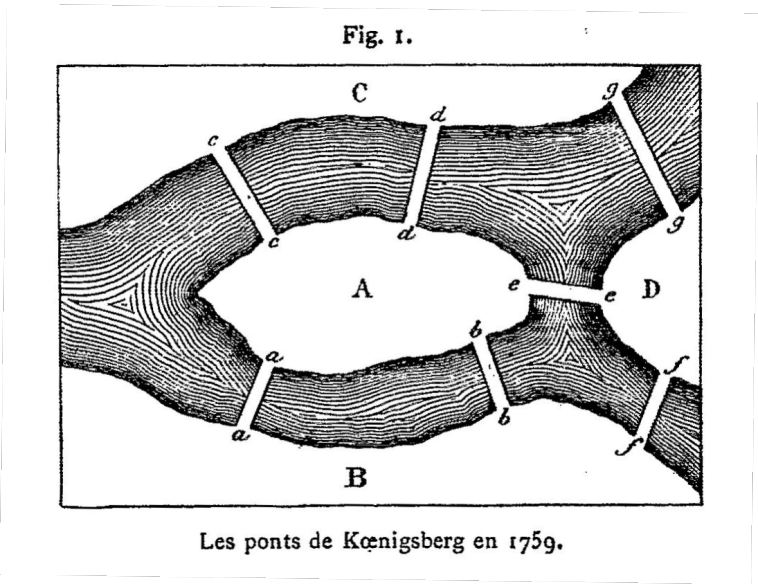


Figure 0.1. – Illustration du problème des sept ponts de Königsberg, extrait de *Récréations mathématiques (2e éd.)* en 1891 [5].

Nous voyons que l'idée fondatrice de la théorie des graphes est de simplifier un système en supprimant ses contingences non essentielles. L'objectif est de garder

seulement l'information pertinente, en vue de rendre les analyses, ainsi que les problématiques, traitables. Le cas des sept ponts de Königsberg est un exemple qui apparaît aujourd'hui comme particulièrement simple. Cependant, la philosophie derrière cette approche résonne comme éminemment moderne au regard de la complexité des données accumulées de nos jours. Ainsi, la théorie des graphes semble adaptée pour analyser des données complexes, que ce soit en physique [6], en chimie [7], en informatique [8], en biologie [9], en écologie [10, 11], en sociologie [12, 13], en économie [14], ou même en linguistique [15]. En biologie, qui est le contexte de mon travail de thèse, représenter certaines données complexes sous forme de graphes est particulièrement efficace. Par exemple, les voies métaboliques sont aisément représentables sous forme de réseaux d'interaction (Fig. 0.2), ce qui peut faciliter leur analyse. L'avantage de représenter les données sous forme de réseau, dans le cadre de la biologie, est de pouvoir déterminer, non seulement les premiers voisins d'un nœud, mais aussi les seconds voisins, ainsi que l'ensemble des interacteurs indirects. L'accès au voisinage des nœuds du réseau est un moyen efficace pour déterminer des proximités entre ces différents nœuds. Ces proximités sont au cœur des stratégies que l'on appelle "coupables par association" (*guilt-by-association*), qui supposent que les nœuds partageant des arêtes en commun partagent également des propriétés communes. Cette stratégie a été utilisée pour prédire les fonctions de certaines protéines [16, 17]. Une autre utilisation bien connue des graphes concerne les réseaux sociaux, où les nœuds représentent des individus et les arêtes, les liens entre ces individus. Ces liens peuvent être de différentes natures : amicale (Facebook), professionnelle (Linkedin) ou scientifique (ResearchGate). À partir de ces réseaux sociaux, il est possible d'étudier la propagation de certains phénomènes comme l'activité physique [18], la consommation de tabac [19], ou l'orientation d'un vote durant des élections [20].

De surcroît, les données auxquelles la recherche contemporaine s'intéresse sont de plus en plus complexes. La complexité de ces données est multiple. Il est possible d'expliciter trois formes majeures de complexité : la taille des données, leur hétérogénéité et la présence de bruit ou d'informations non pertinentes. Pouvoir étudier les systèmes complexes à l'aide de ces données complexes est un défi majeur de la science contemporaine. Face à ce constat, la stratégie déjà éprouvée de représenter les données complexes sous forme de graphes continue d'être une stratégie de premier rang.
Cependant, au regard de l'augmentation croissante de la complexité des données, les graphes pour les représenter se complexifient également. Ainsi, les méthodes jusqu'alors utilisées pour représenter les données sous forme de graphe deviennent, si ce n'est obsolètes, particulièrement lacunaires. Il est ainsi nécessaire de développer de nouvelles manières de représenter les données sous forme de graphes, et de nouvelles méthodes mathématiques et algorithmiques afin de mieux les exploiter.
Au vu de la diversité des contextes et des domaines de recherches basés sur l'exploitation des données complexes et des systèmes complexes qu'elles caractérisent, ainsi que l'interdisciplinarité des approches utilisées, nous parlerons plus volontiers de science

des réseaux que de théorie des graphes. En effet, le champ d'analyse et de développement des approches basées sur les réseaux d'interactions dépasse aujourd'hui très largement le simple cadre de la théorie des graphes. Il est important de noter que, dans la suite du manuscrit, nous utiliserons le terme "graphe" pour se référer à l'objet mathématique et le terme "réseau" pour sa matérialisation dans le cadre de données. Ainsi, nous pouvons remarquer que différents réseaux peuvent correspondre au même graphe.
Mon travail de thèse s'inscrit pleinement dans cette discipline qu'est la science des réseaux, et tente de répondre aux nouvelles problématiques présentées plus haut : la représentation de données diverses et hétérogènes sous forme de réseaux et le développement de nouvelles méthodes mathématiques capables d'explorer et d'analyser ces réseaux.

Ce manuscrit est divisé en trois parties. Une première partie introductive comprend les chapitres de 1 à 4. Une seconde partie concerne les projets menés au cours de ma thèse, du chapitre 5 au chapitre 9, et enfin, une conclusion. Les chapitres introductifs définissent les bases de la théorie des graphes (chapitre 1), ainsi que les bases de la théorie des réseaux multi-couches (chapitre 2). Le chapitre 3 est dédié à l'introduction des marches aléatoires et le chapitre 4 est consacré aux réseaux biologiques. Les cinq chapitres suivants exposent les travaux réalisés dans le cadre de ma thèse. Le chapitre 5 est de type "matériel et méthode" et dédié à la construction des réseaux biologiques. Les chapitres 6, 7, et 8 sont consacrés aux travaux qui ont donné lieu à des articles de recherche. Le chapitre 9, quant à lui, est consacré aux travaux que j'ai commencés dans le cadre de ma thèse, mais qui n'ont pas encore donné lieu à un article de recherche. La conclusion expose les apports de mes travaux de recherche à la communauté scientifique.

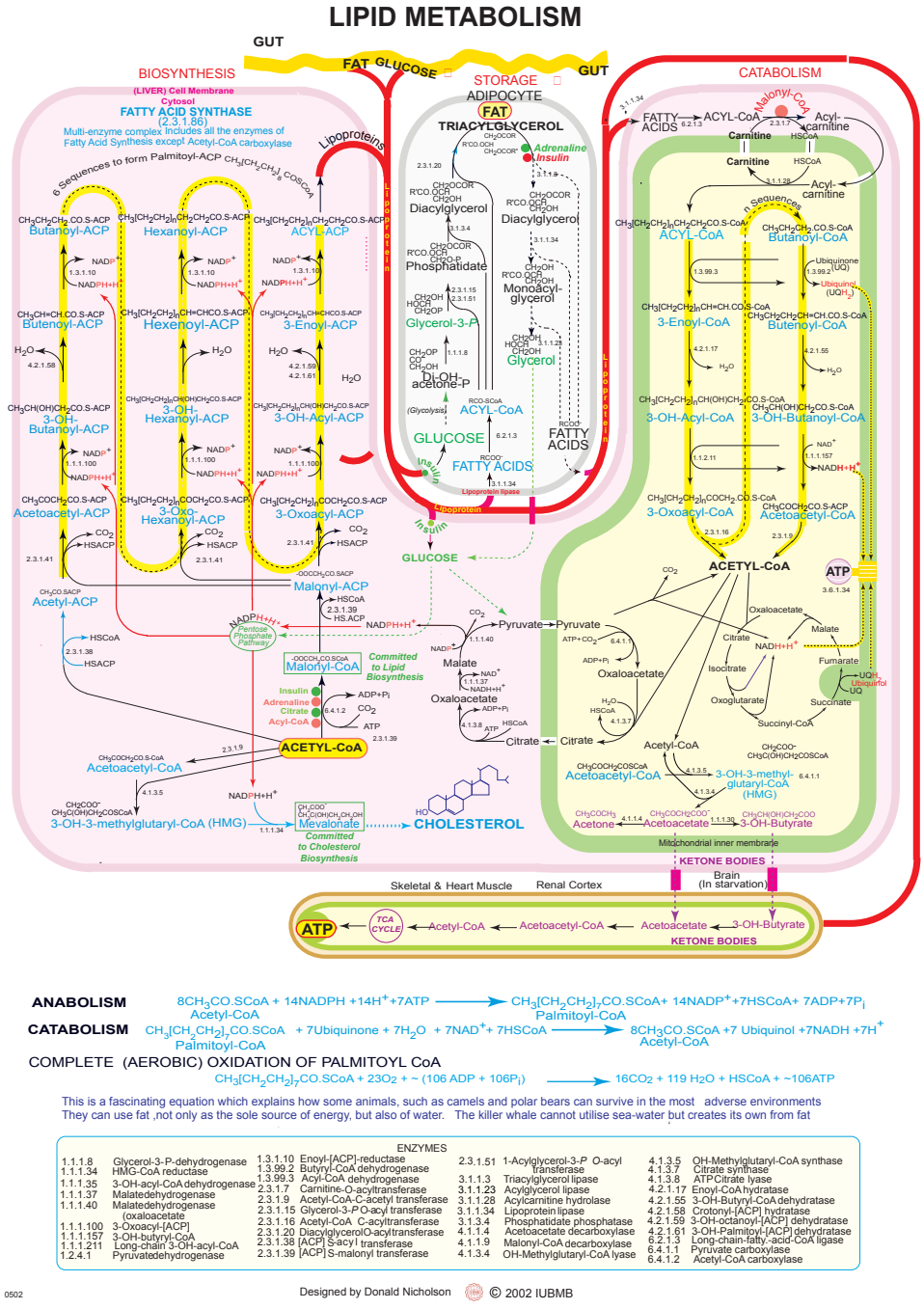


Figure 0.2. – Représentation sous forme de réseau des voies métaboliques des lipides. Image provenant de http://www.iubmb-nicholson.org créée par Donald Nicholson. Reproduite avec la permission de l'Union Internationale de Biochimie et de Biologie Moléculaire.

1. Théorie des graphes

Sommaire

1.1. Définitions et propriétés

Un graphe, noté G, se définit comme une paire de deux ensembles, $G = (V, E)$, où $V = \{v_i,\ i \in [1, n]\}$ est défini comme l'ensemble des nœuds du graphe (Fig. 1.1.a). Nous noterons n le nombre de nœuds dans le graphe. L'ensemble E, étant défini comme l'ensemble des arêtes du graphe, s'écrit $E = \{e_{ij},\ (i, j) \in V \times V\}$. Dans ce contexte, l'arête e_{ij} connecte les nœuds v_i et v_j. Il est bon de noter que dans le cas d'un graphe dirigé, $e_{ij} \neq e_{ji}$. Dans le cas d'un graphe pondéré, chaque arête possède un poids qui n'est pas forcément identique aux poids des autres arêtes. On définit alors une fonction $W : V \times V \rightarrow \mathbb{R}$, qui associe à chaque paire de nœuds v_i et v_j, une valeur w_{ij}. Cette valeur est le poids de l'arête entre les deux nœuds.
De plus, un graphe G' est dit sous-graphe de G si $G' = (V', E')$ tel que $V' \subset V$ et $E' \subset E$. De même, nous pouvons introduire la notion de graphe signé, soit un graphe $G_s = (V_s, E_s)$, auquel nous associerons la fonction de correspondance τ telle que $\tau : V_s \rightarrow \{-1, 1\}$. Cette fonction associe à chaque arête un signe appartenant à l'ensemble $\{-1, 1\}$. Une arête positive est associée à la valeur 1 et une arête négative à la valeur -1.

1.1.1. Matrice d'adjacence

Un graphe G peut s'écrire sous forme ensembliste, comme nous l'avons vu plus haut. Cependant, nous adopterons plus volontiers une représentation matricielle, puisque cette représentation est la représentation algébrique la plus naturelle pour un graphe. Pour ce faire, nous allons introduire la matrice d'adjacence du graphe, notée A. Cette matrice, dans le cadre d'un graphe non dirigé et non pondéré, est définie telle que :

$$A_{ij} = \begin{cases} 1 & \text{si } e_{ij} \neq 0 \\ 0 & \text{sinon} \end{cases} \tag{1.1}$$

Dans le cadre d'un graphe pondéré, la matrice d'adjacence s'écrit de manière similaire. La seule différence avec un graphe non pondéré réside dans la valeur que peuvent prendre les éléments A_{ij}, ces éléments représentant les poids des arêtes. Dans le cas d'un graphe pondéré, ces valeurs peuvent être différentes de 1. Ainsi, la seule contrainte sur la matrice d'adjacence est que $A_{ij} \geq 0$ (i.e tous les éléments de la matrice sont non négatifs).

1.1.2. Degré d'un nœud et distribution des degrés

Une première propriété d'un graphe est donnée par le degré des nœuds qui le constituent et la distribution qui en résulte. Le degré d'un nœud définit le nombre d'arêtes connectées à un nœud, en d'autres termes, le nombre de nœuds adjacent à celui-ci ou le nombre de premiers voisins. Ainsi, le degré du nœud v_i sur un graphe non dirigé se définit comme :

$$k_i = \sum_{j=1}^{n} A_{ij} \tag{1.2}$$

On peut, à partir des degrés des nœuds, définir une distribution, notée P, qui permet de visualiser la répartition des degrés des nœuds au sein d'un graphe. Nous définissons la distribution des degrés des nœuds (Fig. 1.1.b) telle que :

$$P(k) = \sum_{k=1}^{n} p_k = 1 \qquad p_k = \frac{n_k}{n} \tag{1.3}$$

Avec n_k le nombre de nœuds ayant un degré égal à k.

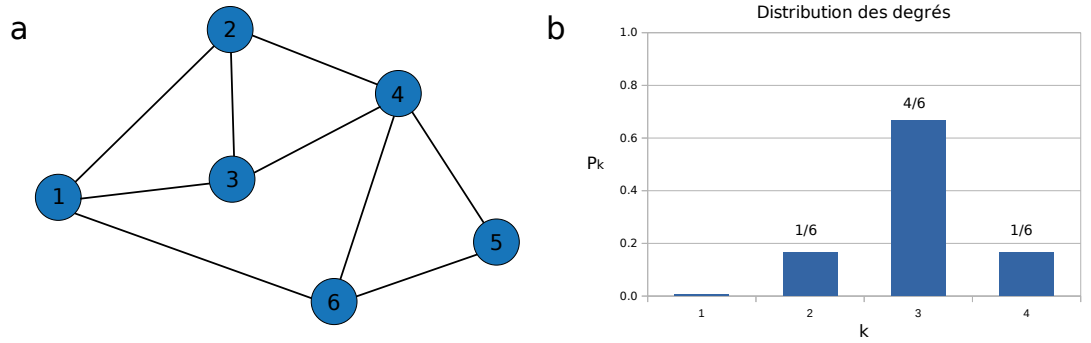


Figure 1.1. – a : Graphe non dirigé composé de six nœuds et de neuf arêtes. b : Distribution des degrés des nœuds du graphe représenté en a

Nous pouvons aussi définir le degré moyen des arêtes dans un graphe. Considérons un graphe qui possède m arêtes. Le nombre de sommets du graphe est donc égal à $2m$. Par conséquent, nous pouvons écrire que :

$$2m = \sum_{i=1}^{n} k_i = \sum_{i,j=1}^{n} A_{ij} \tag{1.4}$$

Ainsi, le nombre moyen d'arêtes par nœud, noté c s'écrit comme :

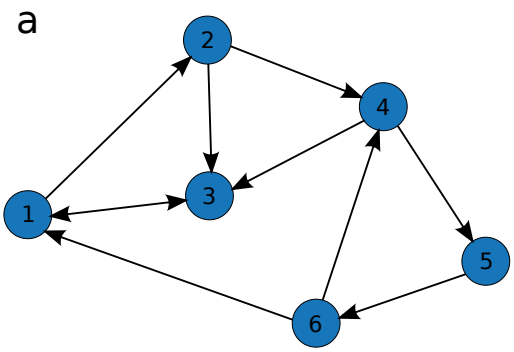
$$c = \frac{1}{n}\sum_{i=1}^{n} k_i = \frac{2m}{n} \tag{1.5}$$

Il est intéressant de noter qu'un nœud ayant un degré élevé, par rapport au degré moyen des nœuds du graphe, sera appelé un *hub*.
Dans la continuité des propriétés introduites, nous pouvons définir la densité d'un graphe, notée ρ, qui correspond au rapport du nombre d'arêtes existantes dans le graphe sur le nombre total d'arêtes possibles.

$$\rho = \frac{m}{\binom{n}{2}} = \frac{m}{\frac{n(n-1)}{2}} = \frac{c}{(n-1)} \approx \frac{c}{n} \tag{1.6}$$

Par définition, un graphe est d'autant plus dense que $\rho \to 1$. A contrario, un graphe ayant une densité proche de zéro sera dit creux (*sparse* en anglais).

Nous avons défini les notions et propriétés précédentes sur des graphes non dirigés, avec le postulat que les arêtes allant du nœud v_i vers le nœud v_j allaient aussi du nœud v_j vers le nœud v_i. Cette bidirectionnalité des arêtes est source de nombreuses implications, comme la symétrie des objets mathématiques, notamment de la matrice d'adjacence. Nous pouvons cependant définir les mêmes notions et propriétés sur des graphes dirigés (Fig. 1.2.a), modulo des adaptations dues à la perte de symétrie des arêtes. Nous définirons ici seulement le cas des degrés entrants et sortants d'un nœud, car il est particulièrement représentatif des pratiques utilisées pour définir les



b

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1.2. – a : Graphe dirigé, composé de six nœuds et de 9 arêtes. b : Matrice d'adjacence correspondante au graphe dirigé représenté en a.

notions et propriétés au sein des graphes dirigés. Nous définissons que la matrice d'adjacence d'un graphe dirigé a pour élément de $A_{ij} = 1$ s'il existe une arête qui va du nœud v_j vers le nœud v_i (Fig. 1.2.b). Dans ce cas, le degré entrant (*in-degree*), noté k_i^{in}, et le degré sortant (*out-degree*), noté k_j^{out}, s'écrivent :

$$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij} \qquad k_j^{\text{out}} = \sum_{i=1}^{n} A_{ij} \tag{1.7}$$

Nous conclurons cette section en précisant que la distribution des degrés des nœuds d'un graphe est au cœur de nombreuses questions. Il est aujourd'hui couramment admis que les réseaux réels obéissent à une distribution des degrés dite en loi de puissance ($P(k) = k^{-\delta}$, avec δ une constante). L'origine de cette particularité structurelle viendrait du fait que ces réseaux seraient plus robustes face aux attaques aléatoires. Ce constat a pu être appuyé par les réseaux biologiques qui semblent obéir à cette loi et dont la structure résulte d'un processus d'évolution.
Cependant, cette idée que les réseaux réels possèdent une distribution des degrés des nœuds en loi de puissance est de plus en plus contestée [21]. Cette vision peut être jugée trop simpliste et soumise à de nombreux biais. Par exemple, il est connu que les réseaux de transport obéissent à cette loi. Mais ils y obéissent, non pas en raison d'un aspect naturel, mais bien parce que ces réseaux sont connus pour être plus robustes aux défaillances aléatoires, et qu'ils permettent un maillage efficace des territoires, tout en étant économiquement viables [22]. Par conséquent, les réseaux construits par l'être humain obéissent souvent à cette loi, puisque ces propriétés sont connues a priori. De plus, l'hypothèse qui semble justifier l'aspect naturel de la loi de puissance, car présente en biologie, est contestable. En effet, les réseaux biologiques sont, eux aussi, soumis à de nombreux biais. Par exemple, la construction des interactomes protéine-protéine peut être biaisée vers une recherche approfondie des interacteurs de certaines protéines. Ces études approfondies de certaines protéines, notamment celles impliquées dans les cancers, comme par exemple *P53* [23], conduisent ces

protéines à devenir des *hubs* au sein des réseaux d'interactions. Ainsi, certains nœuds qui sont des *hubs* ne le sont pas uniquement en raison de leurs fonctions dans les cellules, mais parfois surtout en raison de la quantité de publications qui y sont liées. De manière générale, nous n'avons pas accès aux réseaux complets, mais uniquement à des échantillons biaisés par les interactions facilement identifiables.

1.1.3. Clique, *k-core* et k-composante

Nous venons de définir des propriétés se référant aux nœuds d'un graphe. Il est aussi possible de définir des propriétés se référant à des groupes de nœuds au sein d'un graphe. Nous allons introduire trois notions dont l'objectif est d'identifier les nœuds appartenant à un même sous-graphe : la clique, le *k-core* et la k-composante, tous trois définissant des structures à une échelle intermédiaire entre le nœud et le graphe. La notion de communauté ou de module a aussi pour objectif d'identifier des nœuds appartenant a un même sous-graphe. Une communauté est définie comme étant un sous-graphe composé de nœuds présentant plus d'arêtes entre eux qu'avec des nœuds appartenant au reste du graphe. En biologie, un module fonctionnel est une communauté qui possède une fonction séparée des autres modules [24]. Nous reviendrons à la section 1.3.1 sur les notions de communauté et de module qui sont plus versatiles que les notions de clique, de *k-core* et de k-composante.

Une clique est un ensemble de nœuds au sein d'un graphe non dirigé tel que chaque nœud de l'ensemble est connecté à tous les autres (Fig. 1.3). Il est important de remarquer que des cliques au sein d'un graphe peuvent se recouvrir et partager plusieurs nœuds. La présence d'une clique ou d'une structure s'en rapprochant (quasi-clique) est un indicateur que ce sous-groupe de nœuds est particulièrement cohésif. Cet indicateur est particulièrement important lorsque l'on sait que la plupart des réseaux réels sont peu denses. Dans le cas d'un réseau de protéines, la présence d'une quasi-clique est aussi un indicateur de l'existence d'un module fonctionnel, c'est-à-dire d'un groupe de protéines partageant un grand nombre d'interactions et étant probablement impliqué dans des fonctions similaires au sein des cellules. De la même manière, dans le cas d'un réseau social, la présence d'une clique peut être un indicateur d'une structure familiale, associative ou professionnelle.

Un *k-core* est un sous-groupe au sein d'un graphe dans lequel chaque nœud est connecté à au moins k autres nœuds dans ce même sous-groupe. Ainsi, un *3-core* constitué de 5 nœuds sera un sous-groupe où chacun des 5 nœuds sera connecté à au moins trois autres nœuds parmi les 4 restants. Il est intéressant de remarquer que le graphe de la Fig 1.3, constitué d'une clique à 6 nœuds, peut aussi être vu comme un *5-core*. Cette notion offre une vision moins contraignante que la clique afin de définir la notion d'appartenance à un même sous-groupe.
En dehors d'introduire plus de flexibilité que les cliques dans la définition des sous-

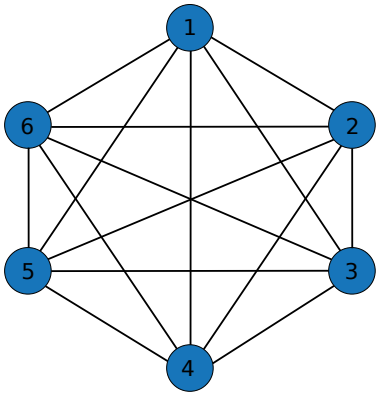


Figure 1.3. – Graphe composé d'une clique de six nœuds.

groupes, la manière de déterminer les *k-core* est numériquement simple. À partir du graphe, il suffit d'enlever tous les nœuds qui ont un degré strictement inférieur à k, ce qui a pour conséquence de réduire le degré de certains nœuds. Puis, on enlève de nouveau tous les nœuds qui ont un degré strictement inférieur à k et on reproduit cette procédure, jusqu'à obtenir un graphe constitué exclusivement des nœuds ayant un degré supérieur ou égal à k. Ce nouveau graphe est, par définition, l'ensemble des nœuds formant le ou les *k-core* du graphe d'origine. Ainsi, la décomposition d'un graphe, pour toutes les valeurs de k, permet d'obtenir une décomposition en oignon ayant des couches correspondantes aux *1-core*, *2-core*, *3-core* et ainsi de suite (composante connexe de gauche de la Fig. 1.4). Cette décomposition peut être vue comme une manière de partitionner un graphe en deux parties : les nœuds du cœur du graphe, appartenant à des couches associées à de hautes valeurs de *k-core*, et inversement ceux appartenant aux couches associées aux faibles valeurs de *k-core* qui définissent la périphérie du graphe. Les nœuds du cœur peuvent être a priori considérés comme plus influents au sein du graphe, puisque mieux connectés aux autres nœuds [25]. Cette propriété est notamment observée, dans les réseaux de régulation de gènes. Il a été montré récemment [26] que la plupart de la variabilité des traits phénotypiques dans une population est causée par un grand nombre de petits effets de trans-régulation (facteur régulant la transcription d'un gène). Ces effets viennent de gènes situés dans la périphérie des réseaux de régulation et qui affectent les gènes situés au cœur du réseau. Ainsi, ces gènes ont des effets directs sur l'apparition des phénotypes associés.

Un graphe $G = (V, E)$ est dit connexe si : $\forall v_i, v_j \in V$, $\exists\, \nu(v_i, v_j)$, avec $\nu(v_i, v_j)$ représentant une chaîne au sein du graphe G connectant v_i à v_j (une chaîne est un ensemble d'arêtes reliant deux nœuds). Ainsi, au sein d'un graphe, une composante connexe est un sous-graphe connexe. Partant de ces définitions, on peut

définir la notion de k-composante : il s'agit d'un sous-graphe constitué d'un ensemble de nœuds dont il faut enlever au moins k nœuds pour rendre ce sous-graphe non connexe (composante connexe de droite de la Fig. 1.4).

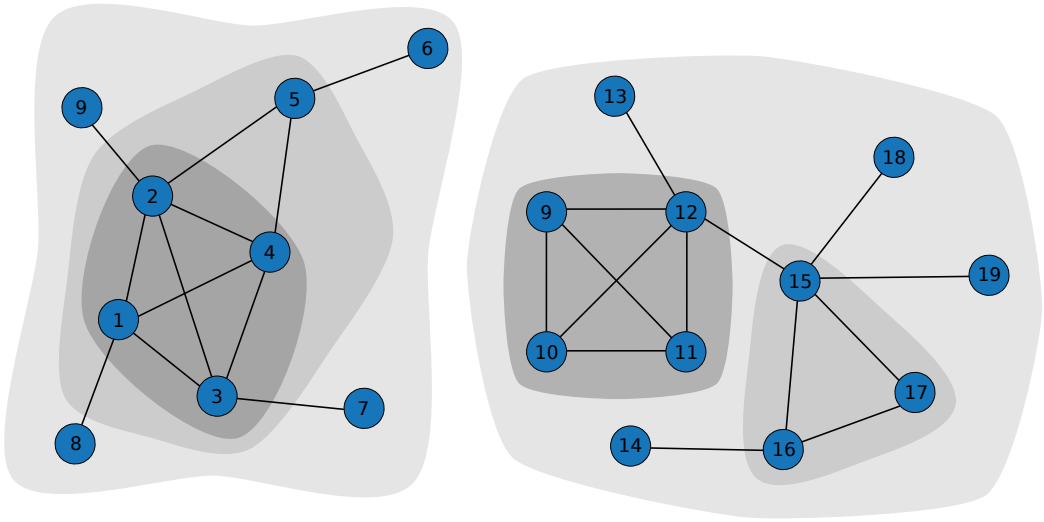


Figure 1.4. – Graphe ayant deux composantes connexes. La composante de gauche est composée de nœuds numérotés de 1 à 8. Le dégradé de gris représente les différents *k-core* : en gris foncé, le *3-core* composé des nœuds au cœur du graphe, en gris, le *2-core* et en gris clair, le *1-core*, autrement dit, les nœuds en périphérie du graphe. La composante de droite est composée de nœuds numérotés de 9 à 19. Le dégradé de gris représente maintenant les différentes k-composantes : en gris foncé, la 3-composante, en gris, la 2-composante et en gris clair, la 1-composante

1.2. Mesures sur les graphes

Les mesures sur les graphes sont fondamentales dans la théorie des graphes. Elles permettent, par exemple, d'identifier les nœuds centraux dans les graphes (mesures de centralité) qui possèdent donc une position stratégique. Dans le cas d'un réseau d'interaction entre protéines, les nœuds centraux sont souvent associés à certaines propriétés biologiques : ils sont souvent essentiels à la survie des cellules [27, 28], ont tendance à être conservés au cours de l'évolution [29], ou ils jouent un rôle central dans l'organisation du réseau en modules fonctionnels [30, 31]. Il existe aussi des mesures permettant de déterminer la similarité entre différents nœuds d'un graphe. Ces mesures de similarité sont souvent utilisées pour la prédiction de liens [32]. Dans le cas d'un réseau d'interaction protéine-protéine, si deux nœuds sont similaires, dans

le sens où ils partagent un voisinage commun, cela peut être un indicateur qu'ils interagissent aussi entre eux [33]. On peut aussi supposer que deux nœuds similaires puissent avoir des fonctions biologiques similaires en suivant le principe du "coupable par association" (*guilt-by-association*) [34].

1.2.1. Mesures de centralité

Dans la section précédente, nous avons introduit une propriété, qui est aussi la mesure la plus directe sur les graphes : le degré des nœuds. Cependant, le degré d'un nœud est une information de premier ordre. Il s'agit d'une mesure uniquement affectée par le voisinage direct d'un nœud. Dans ce cas, un nœud est considéré comme central s'il possède un degré élevé. Néanmoins, le degré n'est pas forcément la mesure la plus pertinente pour définir si un nœud est central au sein d'un graphe. Par exemple, si un nœud possède un degré élevé mais que ses voisins ont tous un degré faible, il sera sans doute moins central qu'un nœud qui a un degré élevé et qui a pour voisins des nœuds également de fort degré. Ainsi, pour éviter cet écueil, d'autres mesures de centralité ont été développées afin de prendre en compte un voisinage plus large, voire la topologie globale du graphe. Nous allons présenter les principales mesures de centralité, tout d'abord celles qui étendent la notion de degré d'un nœud, puis celles qui offrent une vision de la centralité complémentaire à la notion de degré d'un nœud (Table 1.1).

1. **Centralité spectrale (*eigenvector centrality*) :** L'idée de la centralité spectrale est d'obtenir une mesure de centralité du nœud v_i, notée x_i qui dépende de la centralité des voisins de ce nœud. Pour ce faire, on peut définir la centralité spectrale de v_i comme étant :

$$x_i = \alpha^{-1} \sum_{j=1}^{n} A_{ij} x_j \tag{1.8}$$

avec A la matrice d'adjacence du graphe et α^{-1} une constante. On peut réécrire l'équation précédente sous forme vectorielle de la manière suivante :

$$A\mathbf{x} = \alpha\mathbf{x} \tag{1.9}$$

L'équation (1.9) est l'équation aux valeurs propres de la matrice d'adjacence, il existe donc n vecteurs propres. Cependant, le vecteur associé à la mesure de centralité doit être non négatif et de norme égale à 1. Le théorème de Perron-Frobenius nous certifie que, dans le cas d'une matrice à valeurs strictement positives, il existe un unique vecteur à valeurs strictement positives et de norme égale à 1. Il s'agit du vecteur de Perron, associé à la plus grande valeur propre. Ainsi, le vecteur de centralité spectral est le vecteur de Perron associé à la matrice

d'adjacence du graphe. La centralité x_i du nœud v_i est la $i^{\text{ème}}$ composante du vecteur de Perron [35].

2. **Centralité de Katz [36] :** Un constat émerge naturellement suite à la définition de la centralité spectrale : que faire des nœuds n'appartenant pas à une composante connexe ? Dans le cas de la centralité spectrale, leur centralité sera égale à zéro. L'objectif de la centralité de Katz est de dépasser cette difficulté. L'idée de base est de donner à chaque nœud une valeur constante non nulle de centralité. Cela se traduit de la manière suivante :

$$x_i = \alpha \sum_{j=1}^{n} A_{ij} x_j + \beta \tag{1.10}$$

avec A la matrice d'adjacence et α et β deux constantes. On peut réécrire l'équation précédente sous forme vectorielle :

$$\mathbf{x} = \alpha A\mathbf{x} + \beta\mathbf{1} \tag{1.11}$$

Le vecteur $\mathbf{1}$ est défini comme étant $(1,1,1,...)$. Par convention, on pose que $\beta = 1$, et on obtient que la centralité de Katz s'écrit :

$$\mathbf{x} = (I - \alpha A)^{-1}\mathbf{1} \tag{1.12}$$

Il est important de remarquer que la constante α pondère la contribution entre le terme associé à la centralité spectrale et le terme associé à la composante constante donnée a priori à chaque nœud. De plus, il est évident que $\det((I - \alpha A)) \neq = 0$. Nous reviendrons sur cette mesure de centralité et ses applications en section 9.2. Les applications de la mesure de centralité de Katz sont en effet nombreuses en biologie. On peut mentionner par exemple la priorisation de gènes candidats pour certaines maladies [37].

3. ***PageRank* [38] :** Pour la centralité *PageRank*, on se placera dans le cas d'un graphe dirigé, par souci de cohérence avec l'article original de Brin et Page [38]. Le cas des graphes non dirigés est obtenu en considérant le degré du nœud et non le degré sortant. Un défaut de la centralité de Katz est que, si un nœud est associé à un nœud de très forte centralité, il héritera lui aussi d'une centralité importante. Cet artefact peut être gênant dans certaines circonstances, par exemple, lorsqu'un hub est lié à des nœuds de moindre importance. Une idée simple pour remédier à cela est de définir que la centralité d'un nœud est égale à la centralité de ses voisins divisée par leurs degrés sortants (*out-degree*). Cela

se traduit mathématiquement par l'équation suivante :

$$x_i = \alpha \sum_{j=1}^{n} A_{ij} \frac{x_j}{k_j^{\text{out}}} + \beta \tag{1.13}$$

Il faut noter qu'un nœud ayant un degré sortant nul posera problème. Sachant que ce nœud ne contribuera pas à la mesure de centralité, on posera par convention que $k_j^{\text{out}} = 1$. On peut réécrire l'équation précédente sous forme vectorielle :

$$\mathbf{x} = \alpha A D^{-1} \mathbf{x} + \beta \mathbf{1} \tag{1.14}$$

Le vecteur $\mathbf{1}$ est défini comme étant $(1,1,1,...)$, et il est conventionnel de poser que $\beta = 1$. La matrice D est définie telle que $D_{ii} = max(k_i^{\text{out}}, 1)$. Ainsi, nous obtenons que la centralité *PageRank* est égale :

$$\mathbf{x} = (I - \alpha A D^{-1})^{-1} \mathbf{1} \tag{1.15}$$

Le lecteur intéressé pourra se référer aux revues de la littérature suivantes [39, 40] pour plus de détails sur les mesures de centralité et à [41] pour une revue des applications de ces mesures.

	Sans terme constant	Avec terme constant
Sans division par le degré	$\mathbf{x} = \alpha^{-1} A\mathbf{x}$ Centralité spectrale	$\mathbf{x} = (I - \alpha A)^{-1}\mathbf{1}$ Centralité de Katz
Avec division par le degré	$\mathbf{x} = AD^{-1}\mathbf{x}$ Degré	$\mathbf{x} = (I - \alpha AD^{-1})^{-1}\mathbf{1}$ *PageRank*

Table 1.1. – Résumé des principales mesures de centralité, adaptées depuis [42]

4. **Autres mesures de centralité :**

- ***Closeness* [43, 44] :** Il s'agit d'une mesure de centralité définie comme étant égale à l'inverse de la distance moyenne (noté l_i) entre un nœud et tous les autres. Pour la calculer, il faut introduire la notion du plus court chemin entre deux nœuds. Le plus court chemin entre deux nœuds est défini comme étant la distance minimale en nombre d'arêtes séparant deux nœuds. Il est noté d_{ij}, pour définir le plus court chemin entre le nœud v_i et le nœud v_j. Cela se

traduit mathématiquement comme suit :

$$C_i = \frac{1}{l_i} = \frac{n}{\sum_{j=1}^{n} d_{ij}} \tag{1.16}$$

Cette mesure est régulièrement utilisée pour détecter des associations entre nœuds au sein de réseaux biologiques [45].

- ***Betweeness* [46] :** Il s'agit d'une mesure de centralité qui indique à quel point un nœud est important pour relier les autres nœuds entre eux. Introduisons les deux variables suivantes :
  - n_{st}^i qui est le nombre total des plus courts chemins entre les nœuds s et t passant par le nœud i.
  - g_{st} qui est le nombre total des plus courts chemins entre les nœuds s et t.

  La mesure de *betweeness* s'écrit mathématiquement comme suit :

$$B_i = \sum_{s,t=1}^{n} \gamma \frac{n_{st}^i}{g_{st}} \tag{1.17}$$

avec la variable γ qui est un terme de normalisation et qui vaut $\gamma = \frac{2}{(n-1)(n-2)}$ pour un graphe non dirigé. La mesure de *betweeness* est un outil particulièrement efficace pour trouver des nœuds susceptibles de lier des communautés [47]. Les communautés dans les graphes sont des groupes de nœuds qui présentent plus d'arêtes entre nœuds du même groupe, qu'entre nœuds issus de deux groupes différents. Ainsi, la *betweeness* peut être utilisée afin de diviser des graphes en différentes communautés [48].

1.2.2. Mesures de similarité

La notion de similarité est complémentaire à celle de centralité. La centralité permet d'avoir une mesure de l'importance des nœuds par rapport au graphe, tandis que celle de la similarité permet d'avoir une mesure d'équivalence entre les nœuds du graphe. On peut tout d'abord se demander entre quels objets nous voulons définir une similarité. Il est en effet possible de définir des mesures de similarité entre nœuds d'un même graphe, mais aussi entre deux graphes. Dans un deuxième temps, se pose la question de la propriété que l'on cherche à comparer. Il n'existe pas de choix unique. Dans le cas de nœuds d'un graphe, on peut définir que deux nœuds sont similaires s'ils partagent le même voisinage (équivalence structurale), mais on peut aussi définir qu'ils sont similaires s'ils partagent le même rôle au sein du graphe (équivalence régulière). Ces deux notions différentes de similarité pour des nœuds sont illustrées en Fig. 1.5. Dans cette section, nous allons introduire différentes mesures de similarité, tout en reprenant les notions d'équivalence structurale et régulière.

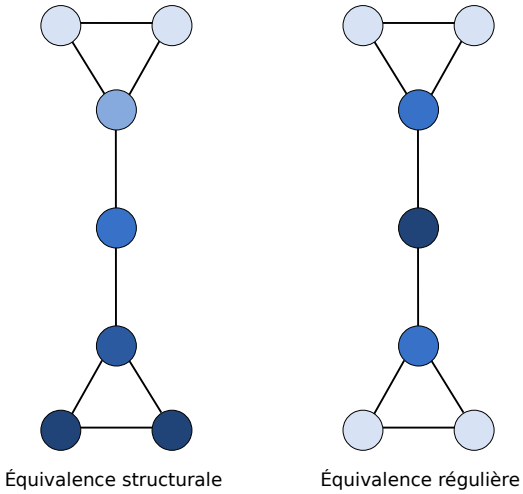


Figure 1.5. – Illustration des équivalences structurales et régulières. La similarité entre les nœuds est représentée par la nuance de bleu. Plus la nuance de bleu est proche, plus les nœuds sont similaires.

1. **Mesures d'équivalence structurale :**

   - **Voisins communs :** La mesure de similarité la plus simple est de calculer le nombre de voisins que partagent deux nœuds, ou en d'autres termes, le recouvrement des voisinages :

   $$n_{ij} = \sum_{k=1}^{n} A_{ik} A_{kj} \tag{1.18}$$

   Cette mesure n'est pas très pertinente, puisqu'elle pénalisera les nœuds à faible degré : ces nœuds auront toujours une faible similarité, qu'importe le nombre de nœuds communs qu'ils auront avec les autres nœuds du graphe. De plus, elle est difficilement interprétable puisque non normalisée.

   - **Indice de Jaccard [49] :** Il s'agit d'une version normalisée de la mesure précédente. L'indice de Jaccard se définit de la manière suivante :

   $$J_{ij} = \frac{n_{ij}}{k_i + k_j - n_{ij}} \tag{1.19}$$

   - **Similarité cosinus [50] :** Il s'agit d'une mesure qui évite le biais de degré des nœuds. Cette mesure provient de la définition géométrique du produit scalaire, $\boldsymbol{x_i}.\boldsymbol{x_j} = \cos(\theta)\|x_i\|\|x_j\|$. La similarité cosinus consiste à calculer le produit scalaire entre les deux vecteurs $\boldsymbol{x_i}$ et $\boldsymbol{x_j}$ représentant les voisins des deux nœuds v_i et v_j (les vecteurs sont extraits de la matrice d'adjacence), puis à normaliser par la norme de ces vecteurs. On obtient la formulation

suivante :

$$\sigma_{ij} = \cos(\theta) = \frac{\sum_{k=1}^{n} A_{ik}A_{kj}}{\sqrt{\sum_{k=1}^{n} A_{ik}^2}\sqrt{\sum_{k=1}^{n} A_{jk}^2}} = \frac{n_{ij}}{\sqrt{k_i k_j}} \tag{1.20}$$

Cette mesure est comprise entre 0 et 1, et si $\sigma_{ij} = 1$ cela implique que les deux nœuds partagent le même voisinage, et qu'ils interagissent entre eux.

- **Corrélation de Pearson [51] :** Cette mesure est une alternative commune à la similarité cosinus. Il est bon de noter que la corrélation de Pearson et la similarité cosinus sont toutes deux invariantes par multiplication. Cependant, seule la corrélation de Pearson est invariante par l'ajout d'un scalaire à tous les éléments. Mathématiquement, la corrélation de Pearson s'écrit :

$$r_{ij} = \frac{\sum_{k=1}^{n} (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_{k=1}^{n} (A_{ik} - \langle A_i \rangle)^2}\sqrt{\sum_{k=1}^{n} (A_{jk} - \langle A_j \rangle)^2}} \tag{1.21}$$

avec $\langle A_i \rangle$ la moyenne de $i^{\text{ème}}$ ligne de la matrice d'adjacence.

En biologie, on désigne les approches utilisant les mesures d'équivalence structurale comme étant des approches de "coupable par association". Dans le cas des protéines, l'hypothèse est que deux protéines ayant des fonctions similaires auront des profils d'interactions similaires [52]. Ainsi, des protéines proches dans le réseau auront plus de chances d'être impliquées dans les mêmes fonctions ou processus biologiques dans la cellule.

2. **Mesures d'équivalence régulière :**

- **Une première mesure de similarité régulière :** Une idée de base [53–55] et intuitive est de définir une mesure telle que les nœuds v_i et v_j ont une forte similarité si leurs voisins v_k et v_l ont eux aussi une forte similarité. Cela se traduit mathématiquement de la manière suivante :

$$\sigma_{ij} = \alpha \sum_{k,l=1}^{n} A_{ik}A_{jl}\sigma_{kl} + \delta_{ij} \tag{1.22}$$

avec α une constante, et la variable δ_{ij} qui est égale à 1, si les nœuds v_i et v_j sont les mêmes, ou égale à 0 autrement. Cette variable permet d'avoir une

auto-similarité égale à 1. Matriciellement, cela se traduit comme étant :

$$\sigma = \alpha A \sigma A + I \tag{1.23}$$

avec I la matrice identité. On peut obtenir une solution de cette équation en définissant le schéma itératif suivant :

$$\begin{cases} \sigma^{(0)} = 0 \\ \sigma^{(t+1)} = \alpha A \sigma^{(t)} A + I \end{cases} \tag{1.24}$$

On obtient les premiers termes de la série en utilisant le schéma itératif précédent (1.24), ce qui nous donne :

$$\begin{cases} \sigma^{(0)} = 0 \\ \sigma^{(1)} = I \\ \sigma^{(2)} = I + \alpha A^2 \\ \sigma^{(3)} = I + \alpha A^2 + \alpha^2 A^4 \end{cases} \tag{1.25}$$

Il se pose un constat : à travers ce processus, nous allons obtenir uniquement dans la série les termes de longueur de chemins pairs. Cependant, il n'y a, a priori, pas de raison de supposer que les termes impairs ne contribuent pas à la mesure de similarité entre les nœuds. Donc, afin d'éviter une mesure partielle, il faut définir une autre mesure qui introduit les termes impairs dans le développement de la série.

- **Similarité de Katz [36, 56] :** La similarité de Katz est une manière de dépasser le problème exposé précédemment en intégrant les termes impairs dans la série, afin d'obtenir une mesure qui intègre toutes les longueurs de chemins. Ainsi, nous supposons ici que les nœuds v_i et v_j ont une forte similarité si les voisins de v_i, notés v_k, sont similaires eux-mêmes à v_j. Mathématiquement, cela se traduit de la manière suivante :

$$\sigma_{ij} = \alpha \sum_{k=1}^{n} A_{ik} \sigma_{kj} + \delta_{ij} \tag{1.26}$$

$$\sigma = \alpha A \sigma + I \tag{1.27}$$

Ainsi, en adoptant de nouveau un schéma itératif, on obtient les premiers termes de la série suivants :

$$\begin{cases} \sigma^{(0)} = 0 \\ \sigma^{(1)} = I \\ \sigma^{(2)} = I + \alpha A \\ \sigma^{(3)} = I + \alpha A + \alpha^2 A^2 \end{cases} \tag{1.28}$$

Nous observons maintenant aussi bien les termes pairs que les termes im-

pairs. Nous voyons que, si nous faisons tendre la somme vers l'infini, nous obtenons une série de Neumann, ce qui permet d'obtenir l'équation suivante :

$$\sigma = \sum_{m=0}^{\infty} (\alpha A)^m = (I - \alpha A)^{-1} \tag{1.29}$$

Cette équation rappelle la centralité de Katz que l'on a vu précédemment : elle est, en quelque sorte, une version matricielle de la centralité de Katz. Il est bon de noter que la similarité de Katz est définie [56] en enlevant le premier terme de la série de Neumann. Ainsi, la similarité de Katz est définie de la manière suivante :

$$\sigma = \sum_{m=1}^{\infty} (\alpha A)^m = (I - \alpha A)^{-1} - I \tag{1.30}$$

Cependant, il est bon de noter que la centralité et la similarité de Katz ne partagent pas de parenté commune.
En suivant la même démarche que précédemment, il est possible d'obtenir l'équation (1.26) tout en la divisant par le degré des nœuds

$$\sigma_{ij} = \frac{1}{k_i}(\alpha \sum_{k=1}^{n} A_{ik}\sigma_{kj} + \delta_{ij}) \tag{1.31}$$

En introduisant la matrice D, qui est la matrice diagonale des degrés des nœuds, définie telle que $D_{ii} = k_i$, il est possible de réécrire matriciellement l'équation précédente de la manière suivante :

$$\sigma = D^{-1}(\alpha A\sigma + I) = (D - \alpha A)^{-1} \tag{1.32}$$

On obtient ainsi une équation qui n'est pas sans rappeler *PageRank*, même si la correspondance n'est pas parfaite. En effet, la somme des similarités σ_{ij} sur tous les nœuds v_j ne donne pas la centralité issue de *PageRank* pour le nœud v_i, mais plutôt la centralité de *PageRank* pour le nœud v_i divisé par k_i.

3. **Autres mesures de similarité :**

Nous allons brièvement mentionner d'autres mesures et méthodes qui permettent de définir des similarités entre nœuds.

- **Similarités basées sur les marches aléatoires :** Les marches aléatoires sur les graphes sont des méthodes particulièrement efficaces (voir chapitre 3) pour définir des mesures intégrant la topologie du graphe. Ainsi, il est possible d'utiliser les marches aléatoires pour définir des mesures de centralité, comme par exemple la mesure *PageRank* décrite précédemment. Mais on

peut aussi utiliser les marches aléatoires pour définir des mesures de similarité, comme nous avons pu le remarquer avec les équations (1.30-1.32). Par ailleurs, il existe d'autres types de marches aléatoires, par exemple les marches aléatoires avec *restart* (RWR), qui permettent de déterminer la similarité entre tous les nœuds d'un graphe et un nœud de référence (appelée "graine") choisie préalablement. Ainsi, on peut déterminer une matrice de similarité basée sur les RWR en itérant sur tous les nœuds du graphe. Nous expliciterons ces méthodes, qui sont au cœur de mon travail de thèse, plus en détail dans les chapitres 3, 6, 7 et 9.

- **Similarités issues d'un *embedding* des nœuds d'un graphe :** Une pratique de plus en plus adoptée dans la communauté des graphes est de passer par un *embedding* des nœuds du graphe avant d'appliquer différentes mesures sur la représentation obtenue. L'*embedding* d'un graphe est la projection de ce graphe dans un espace vectoriel de moindre dimension. Dans le cas de l'*embedding* des nœuds du graphe, chaque nœud du graphe est représenté par un vecteur. La détermination des vecteurs se fait en optimisant une distance entre chacun des vecteurs, afin de préserver des propriétés du graphe initial. L'*embedding* des nœuds d'un graphe a pour intérêt de faire ressortir certaines propriétés jugées pertinentes et d'être plus robuste face aux bruits issus des données et de la construction du réseau. Nous reviendrons plus en détail sur les aspects mathématiques et conceptuels de ces méthodes d'*embedding* dans les chapitres 8 et 9.

1.2.3. Autres propriétés et mesures

Nous allons détailler dans cette section d'autres propriétés et mesures qui ne sont ni des mesures de centralité, ni des mesures de similarité. Cependant, ces propriétés et mesures peuvent donner des informations précieuses sur un graphe et les nœuds d'un graphe.

- **Coefficient de *clustering* [57] :** Ce coefficient mesure la tendance qu'ont trois nœuds v_i, v_j, et v_k à former une clique, sachant que v_i est lié à v_j, et que v_j est lui-même lié à v_k. En d'autres termes, ce coefficient mesure la tendance qu'ont trois nœuds choisis arbitrairement dans le graphe à former un triangle.

$$Cl = \frac{3 * (\text{nombre de triangles})}{(\text{nombre de triplets connectés})} \tag{1.33}$$

  Il est bon de noter que si $Cl \to 1$, cela implique que le graphe tend à être une clique.

De plus, il est possible de définir un coefficient de *clustering* local :

$$Cl_i = \frac{\text{(nombre de paires de voisins de } i \text{ qui sont connectés)}}{\text{(nombre de paires de voisins de } i\text{)}} \tag{1.34}$$

On peut réécrire l'équation précédente (équation 1.34) à l'aide de la notion de redondance d'un nœud i noté R_i. La redondance du nœud i est définie comme étant la moyenne des degrés de ses voisins, à laquelle on retranche 1 (cela permet de déduire le degré dû à l'arête avec le nœud i). Mathématiquement, cela s'écrit :

$$Cl_i = \frac{\frac{1}{2}k_i R_i}{\frac{1}{2}k_i(k_i-1)} = \frac{R_i}{k_i-1} \tag{1.35}$$

- **Diamètre :** Il s'agit du plus grand des plus courts chemins entre tous les nœuds du graphe, pris deux à deux. On parle de réseau "petit monde" si le diamètre d'un graphe est faible. L'expérience de Stanley Milgram sur le passage de lettre (*letter-passing*) a illustré pour la première fois le phénomène "petit monde" (*small-world experiment*). L'expérience consistait à demander à une soixantaine d'habitants du Nebraska d'envoyer une lettre vers un habitant cible à Boston. Une règle leur était donnée : ils pouvaient transmettre la lettre uniquement via une chaîne de connaissances personnelles, ou d'amis étant susceptibles de connaître la cible. L'expérience a montré que les habitants du Nebraska et la cible habitant à Boston étaient liés en moyenne par six personnes [58, 59]. Donc le réseau constitué par l'ensemble des habitants du Nebraska, de l'habitant de Boston et de l'ensemble de leurs connaissances possède un diamètre proche de six. Ce phénomène de "petit monde" (*small-world experiment*) est retrouvé en dehors du cadre de ces expériences de Milgram et il peut se retrouver à l'échelle de réseaux sociaux plus larges, comme Facebook [60], ou de réseaux de protéines [61] et de bien d'autres [62].

Ces différentes mesures forment les outils de base de l'analyse de graphes. Elles peuvent être utilisées comme éléments dans des algorithmes qui permettent une exploitation plus approfondie des informations contenues dans les graphes. Ces mesures ont été aussi utilisées pour classer les graphes et définir des comportements génériques. Les différentes mesures peuvent également être améliorées ou étendues afin d'être utilisées sur des réseaux plus complexes comme les réseaux composés de plusieurs couches d'interactions. Ces réseaux plus complexes permettent d'obtenir une représentation plus complète, et donc plus réaliste, des systèmes complexes. Nous détaillerons ces points dans le chapitre suivant.

1.3. Algorithmes sur les graphes

Nous avons vu dans le chapitre précédent les mesures sur les graphes, qui fournissent un premier niveau d'analyse du graphe et des nœuds qui le constituent. Cependant, ces mesures exploitent seulement une partie de l'information disponible dans le graphe. Par exemple, si l'on veut définir une partition d'un graphe, il est difficile de le faire uniquement à partir des mesures définies dans la section précédente. Dans cette section, nous allons définir deux classes d'algorithmes sur les graphes. Ces deux classes d'algorithmes ont constitué une partie de mon travail de thèse : le partitionnement de graphes et l'*embedding* de graphes. Nous parlerons des algorithmes de diffusion et en particulier des marches aléatoires au chapitre 3, puisque ces algorithmes ont constitué le cœur de mon projet de thèse.

1.3.1. Algorithmes de partitionnement

Les algorithmes de partitionnement de graphes, parfois aussi nommés algorithmes de détection de communautés, ont pour objectif d'identifier des sous-graphes, appelés *clusters* ou communautés. Il est bon de noter que, malgré l'utilisation régulièrement indifférenciée des expressions "partitionnement de graphes" et "détection de communautés", il est possible d'identifier une nuance. En effet, un algorithme de détection de communautés pourra avoir pour objectif d'identifier un unique sous-graphe autour d'un nœud d'intérêt, tandis qu'un algorithme de partitionnement de graphes définira autant de sous-graphes que nécessaires pour partitionner le graphe dans sa totalité. Suivant cette distinction, un algorithme de partitionnement permettra de détecter des communautés, tandis qu'un algorithme de détection de communautés ne permettra pas systématiquement de faire un partitionnement de graphes. On définit un *cluster* ou une communauté comme étant un sous-graphe composé de nœuds présentant plus d'arêtes entre eux qu'avec des nœuds appartenant au reste du graphe. En biologie, ces *clusters* ou communautés sont fréquemment nommés des modules fonctionnels.

Déterminer des *clusters* ou des communautés est d'une grande importance dans de nombreux domaines où les systèmes sont représentés sous forme de graphes, comme en biologie, en sociologie, ou en informatique [63, 64].
Un exemple célèbre de partitionnement de graphes est le partitionnement du club de karaté de Zachary (*Zachary's karate club*) [65]. Dans ce graphe, les nœuds représentent les membres du club et les arêtes, les liens d'affinités entre les membres du club. Ce graphe a régulièrement été utilisé comme graphe-test pour les différents algorithmes de partitionnement de graphes, et ce, depuis qu'il a été popularisé au début du XXI^{e} siècle [66]. Ce graphe illustre bien l'intérêt du partitionnement de graphes. Les algorithmes de partitionnement de graphes définissent deux clusters dans le graphe du club de karaté. Ce partitionnement trouve un sens sociologique, puisque durant la construction du graphe, un conflit a eu lieu entre deux membres du club, ce qui a polarisé le club en deux groupes d'individus. Ainsi, les algorithmes de partitionnement de graphes sont un bon moyen d'obtenir des informations sur l'organisation des systèmes que les graphes représentent. En biologie, le partitionnement de graphes

permet, entre autres, de s'intéresser à la détermination de modules fonctionnels au sein de réseaux de protéines. Les modules fonctionnels sont par définition des groupes qui ont une fonction séparée des autres modules [24]. Ainsi, partitionner un réseau de protéines permet de définir des sous-graphes séparés les uns des autres. Ces sous-graphes peuvent indiquer l'existence de modules fonctionnels. Le partitionnement de graphes est une des approches les plus utilisées de stratégies du "coupable par association" [67]
De nombreuses méthodes de partitionnement de graphes existent. Cependant, dans cette section, je choisis de détailler deux algorithmes particulièrement utilisés et représentatifs de la littérature : l'algorithme de Louvain [68] et l'algorithme *k-means* [69].

- **Algorithme de Louvain :** L'algorithme de Louvain a pour objectif un partitionnement efficace du graphe, aussi bien en termes de performance dans la définition des *clusters*, qu'en termes de complexité temporelle. Il est basé sur la maximisation de la modularité. Nous allons définir la notion de modularité [47] qui est au cœur de cette méthode et qui a été très largement utilisée [70–72]. Dans le cas où l'on considère un graphe contenant m arêtes et n nœuds, et que l'on considère que le graphe est divisé en deux groupes, nommés groupe 1 et groupe 2, la modularité s'écrit mathématiquement de la manière suivante :

$$Q = \frac{1}{4m}\sum_{i,j=1}^{n}(A_{ij} - \frac{k_i k_j}{2m})(s_i s_j + 1) = \frac{1}{4m}\sum_{i,j=1}^{n}(A_{ij} - \frac{k_i k_j}{2m})\delta_{s_i,s_j} \tag{1.36}$$

avec k_i et k_j les degrés des nœuds v_i et v_j, et A la matrice d'adjacence. Les variables s_i et s_j sont les fonctions indicatrices associées aux nœuds v_i et v_j, définies telles que $s_i = 1$, si le nœud v_i appartient au groupe 1 et $s_i = -1$, si le nœud v_i appartient au groupe 2. Il est intéressant de remarquer que l'expression de droite de l'équation (1.36) est obtenue en constatant que $\frac{1}{2}(s_i s_j + 1)$ est égal à 1, si v_i et v_j appartiennent au même groupe, et 0 sinon. Il est courant de voir la modularité exprimée sous une forme matricielle :

$$Q = \frac{1}{4m}\boldsymbol{s}^{\boldsymbol{T}} M \boldsymbol{s} \qquad M_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{1.37}$$

où M est la matrice de modularité. On constate que cette définition de la modularité permet de déterminer deux groupes au sein d'un graphe. Dans le cas où l'on cherche à optimiser cette modularité, il est plus pratique de s'intéresser à la variation de la modularité, notée ΔQ. On suppose que l'on déplace le nœud v_k du premier groupe vers le second groupe. Ainsi, seuls les termes se rapportant au nœud v_k peuvent changer la valeur de la modularité. À partir de l'équation (1.37), on voit que le changement de groupe du nœud v_k, fait intervenir trois termes : $M_{kk}s_k^2$ qui ne contribuera pas, puisque changer de signe s_k ne change rien, $\sum_{i\neq k} M_{ik}s_i s_k$ qui change de signe, et $\sum_{j\neq k} M_{kj}s_k s_j$ qui change également

de signe. De plus, vu que $M_{ij} = M_{ji}$, les deux derniers termes sont égaux. La réaffectation du nœud v_k diminue donc la modularité du premier groupe de $-2\sum_{i\neq k} M_{ik}s_i s_k$, et d'une même quantité le second groupe. Ainsi, la variation de la modularité est égale à :

$$\begin{aligned}\Delta Q &= \frac{1}{4m}(2 * (-2\sum_{i\neq k} M_{ik}s_i s_k)) \\ &= -\frac{1}{m}\sum_{i\neq k} M_{ik}s_i s_k \end{aligned} \tag{1.38}$$

Cependant, il est souvent nécessaire d'avoir un partitionnement en plus de deux groupes. Dans ce cas, la variation de la modularité s'écrit de la manière suivante :

$$\Delta Q = \frac{1}{2m}(\sum_{i,j\in g} M_{ij}\frac{1}{2}(s_i s_j + 1) - \sum_{i,j\in g} M_{ij}) \tag{1.39}$$

où g désigne un des groupes obtenus par partitionnement du graphe. Il est possible de réécrire l'équation (1.39), de la manière suivante :

$$\begin{aligned}\Delta Q &= \frac{1}{4m}(\sum_{i,j\in g} M_{ij}s_i s_j - \sum_{i,j\in g} M_{ij}) \\ &= \frac{1}{4m}(\sum_{i,j\in g} M_{ij} - \delta_{i,j}\sum_{k\in g} M_{ik})s_i s_j = \frac{1}{4m}\sum_{i,j\in g} M^{(g)}_{ij}s_i s_j \end{aligned} \tag{1.40}$$

avec $M^{(g)}_{ij} = M_{ij} - \delta_{i,j}\sum_{k\in g} M_{ik}$.
Une manière d'obtenir un partitionnement optimal du graphe est d'utiliser l'algorithme de Louvain qui cherche à maximiser cette variation de modularité, à l'aide d'une méthode heuristique basée sur un principe d'algorithme agglomératif. Dans le cas de l'algorithme de Louvain, la variation de la modularité s'exprime de la manière suivante :

$$\Delta Q = \left[\frac{\sum_{\text{in}} + 2k^{(\text{g})}_i}{2m} - (\frac{\sum_{\text{tot}} + k_i}{2m})^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - (\frac{\sum_{\text{tot}}}{2m})^2 - (\frac{k_i}{2m})^2\right] \tag{1.41}$$

avec $\sum_{\text{in}}$ la somme des arêtes d'un groupe g, $\sum_{\text{tot}}$ la somme des arêtes associées au groupe g, k_i la somme des arêtes associées au nœud v_i et $k^{(\text{g})}_i$ la somme des arêtes du nœud v_i vers le groupe g. L'algorithme de Louvain est séparé en deux phases. Au cours de la première phase (*Modularity Optimization* Fig. 1.6), chaque nœud du graphe est associé à son propre groupe. Puis, pour chaque nœud v_i, la variation de modularité (équation 1.41) est calculée en plaçant v_i dans chacun des groupes issus de ses nœuds voisins v_j. Finalement, le nœud v_i est placé dans le groupe pour lequel la variation de modularité a été maximale. Si la variation est négative, le nœud v_i reste dans son groupe initial. Ce processus

est répété itérativement jusqu'à ce que l'on n'observe plus d'amélioration de la modularité.
La seconde phase de l'algorithme de Louvain (*Community Aggregation* Fig. 1.6) consiste à construire un graphe où chaque nœud est associé au groupe défini durant la première phase. Pour ce faire, on construit un graphe pondéré où le poids des arêtes entre deux nœuds issus de deux groupes différents est donné par la somme des poids associés aux deux groupes auxquels sont associés les nœuds. Les nœuds associés à un même groupe sont définis comme une boucle dont le poids est donné par la somme des arêtes entre chacun des nœuds du groupe.
Enfin, l'algorithme (Fig. 1.6) est réitéré à partir du nouveau graphe obtenu à la fin de la seconde phase.

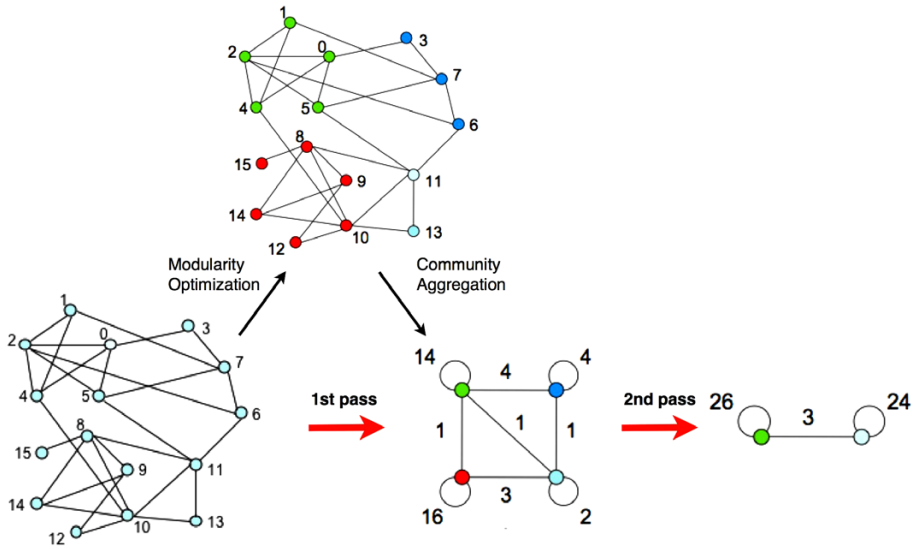


Figure 1.6. – Algorithme de Louvain en deux phases. Chaque *pass* correspond à une itération des deux phases de l'algorithme. La figure est extraite de l'article *Fast unfolding of communities in large networks, Blondel et al. J. Stat. Mech (2008)* [68]. Reproduit avec la permission de l'éditeur *IOP Publishing*, licence numéro 1226872-1.

- **Algorithme *k-means* :** Un autre algorithme classique de partitionnement est l'algorithme *k-means*. Cependant, cet algorithme ne fonctionne pas directement sur les graphes mais sur un ensemble de vecteurs. Une idée pour transformer un graphe en un ensemble de vecteurs est de passer par un algorithme de visualisation (*graph layout*) afin d'obtenir une représentation dans un espace vectoriel qui prend en compte la topologie du graphe. De nombreuses méthodes de visualisation existent [73]. Les méthodes du type *Force-based layout* sont particulièrement utilisées, notamment l'algorithme de Fruchterman–Reingold [74]. Ces méthodes font l'hypothèse que les nœuds du graphe peuvent être associés

à des particules et que les arêtes définissent les forces qui s'appliquent entre ces particules. Différents modèles existent, tels que des modèles électrocinétiques où les nœuds sont assimilés à des particules chargées et où les arêtes sont associées à la force de coulomb. D'autres modèles adoptent plutôt une vision mécanique où les nœuds sont des masses et où les arêtes sont associées à des ressorts dotés d'une force de rappel.
Une autre manière de faire, qui est mathématiquement plus fondée, est de passer par une méthode d'*embedding* (voir détails dans la section suivante). L'*embedding* permet de projeter un graphe dans un espace vectoriel de basse dimension. Il est bon de noter qu'il est possible de projeter le graphe dans un espace d'*embedding* de basse dimension qui est strictement supérieur à deux, afin de préserver le maximum d'informations contenues dans le graphe, puis d'utiliser une méthode de visualisation comme l'Analyse en Composantes Principales (PCA) [75] pour obtenir une représentation en deux dimensions.
Si nous admettons que nous avons une représentation vectorielle d'un graphe sous forme de n données, nous pouvons alors appliquer l'algorithme *k-means*, sachant que nous voulons obtenir k clusters. L'idée centrale de cet algorithme est de définir k centroïdes, tels que les n données soient associées au centroïde le plus proche, de telle manière que chaque centroïde et les points associés forment des clusters. Ainsi, on définit chaque donnée par un vecteur $x_i, \forall\, i \in [\![1, n]\!]$ et chaque cluster par le vecteur $c_j, \forall\, j \in [\![1, k]\!]$. Par conséquent, le problème peut être résolu en minimisant la fonction suivante :

$$\phi = \sum_{i=1}^{n} \sum_{j=1}^{k} \min \left\| x_i - c_j \right\|^2 \tag{1.42}$$

Une amélioration du temps de convergence de l'algorithme *k-means* revient à choisir une condition initiale pour les k centroïdes de manière non aléatoire (nommée *k-means++* [76]).

Comme précisé en introduction, il existe de nombreuses autres méthodes de partitionnement et de partitionnement de graphes, le lecteur intéressé pourra se référer aux revues de la littérature suivantes [77–79].

1.3.2. *Embedding* de graphes

Les algorithmes d'*embedding* permettent de projeter un graphe dans un espace vectoriel de moindres dimensions, en préservant certaines propriétés du graphe initial. L'*embedding* de graphes possède de multiples avantages : faire ressortir les propriétés les plus pertinentes, réduire la sensibilité aux bruits des données ayant permis la construction du graphe, ou exploiter la représentation vectorielle pour utiliser des méthodes supervisées. Les méthodes qui tirent profit de l'*embedding* sont nom-

breuses. On peut citer par exemple : le partitionnement de graphes, la classification de nœuds ou la prédiction d'arêtes.

On considère un graphe G définit tel que $G = (V, E)$, avec $V = \{v_i,\ i \in [1, n]\}$ l'ensemble des nœuds du graphe et E l'ensemble des arêtes du graphe définies telles que $E = \{e_{ij},\ (i, j) \in V \times V\}$. Nous noterons n le nombre de nœuds du graphe. L'*embedding* d'un graphe a pour but de déterminer la projection d'un graphe appartenant à un espace dit direct vers un espace vectoriel de dimension réduite (espace d'*embedding*), de manière à ce qu'à chaque nœud du graphe soit associé un vecteur. Cette correspondance est obtenue en minimisant une distance définie, de telle manière que les nœuds "similaires" dans l'espace direct du graphe le soient aussi dans l'espace d'*embedding*. Mathématiquement, cela revient à déterminer la fonction de correspondance f telle que :

$$f : \begin{cases} V & \to \mathbb{R}^d \\ v_i & \mapsto z_i \end{cases} \tag{1.43}$$

où z_i est la représentation vectorielle du nœud v_i dans l'espace d'*embedding*. Cet espace d'*embedding* possède une dimension d, telle que $d \ll n$.
Ainsi, l'objectif est de définir une représentation vectorielle de moindre dimension de chaque nœud v_i qui préserve les propriétés des nœuds au sein du graphe. Cependant, le choix des propriétés à préserver n'est ni évident, ni universel. Il dépendra de l'application que l'on souhaite faire de l'*embedding*, ainsi que des connaissances que nous avons a priori sur le graphe. Nous allons définir quelques propriétés qui sont régulièrement préservées par les méthodes d'*embedding*. Ces propriétés sont définies en termes de similarité entre les nœuds du graphe.

- **La similarité d'ordre un** est associée à la similarité entre les paires de nœuds du graphe. Les poids entre les nœuds est une mesure de similarité d'ordre un.

- **La similarité d'ordre deux** entre les nœuds du graphe est associée à la similarité entre les voisinages de chaque nœuds. On note $\boldsymbol{s}_{\boldsymbol{v_i}}$ le vecteur de similarité d'ordre un entre le nœud v_i et les autres nœuds du graphe. Ainsi, on définit la similarité d'ordre deux entre le nœud v_i et le nœud v_j comme étant la similarité entre les deux vecteurs de similarité d'ordre un $\boldsymbol{s}_{\boldsymbol{v_i}}$ et $\boldsymbol{s}_{\boldsymbol{v_j}}$, associés aux deux nœuds.
  Les similarités d'ordre supérieur entre les nœuds sont définies de manière similaire. Ces similarités définissent des équivalences structurales.

- **La similarité régulière** des nœuds définit la similarité entre les nœuds qui partagent le même rôle au sein de leurs voisinages (équivalence régulière). À titre d'exemple, on peut imaginer des nœuds qui font le pont entre deux communautés, ou bien des nœuds qui appartiennent à une clique. L'avantage de cette

définition de la similarité est qu'elle permet de dévoiler des similarités entre nœuds distants, ce que ne permettent pas les similarités basées uniquement sur les voisinages.

- **La similarité intra-communautaire** définit la similarité entre les nœuds appartenant à une même communauté. Ainsi, cette similarité a pour but de préserver la structure modulaire d'un graphe.

L'*embedding* de graphes est un domaine récent. Par conséquent, il ne possède pas encore de formalisme universel. Cependant, des tentatives pour regrouper les différentes méthodes sous un formalisme mathématique commun existent. À ce titre, on peut citer le travail de Hamilton et al. [80]. Même si le formalisme de Hamilton et al. n'englobe pas toutes les méthodes d'*embedding*, il reste particulièrement adapté à mon travail qui s'est focalisé surtout sur les méthodes du type *shallow embedding* (chapitres 8 et 9.3). Ce formalisme propose un cadre qui organise les méthodes d'*embedding* en suivant quatre composantes :

1. **Fonction de similarité de paire :** $s_{\mathcal{G}} : V \times V \rightarrow \mathbb{R}^+$.
   Cette fonction définit la mesure de similarité entre les nœuds dans l'espace direct, comme nous l'avons vu précédemment. Il existe de nombreuses mesures de similarité de paire.
2. **Une fonction *encoder* :** $\text{Enc} : V \rightarrow \mathbb{R}^d$.
   Cette fonction permet la projection d'un nœud du graphe dans l'espace d'*embedding*, par exemple le nœud $v_i \in V$ est projeté vers le vecteur $z_i \in \mathbb{R}^d$.
3. **Une fonction *decoder* :** $\text{Dec} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$.
   Cette fonction associe une mesure de similarité à chaque paire de vecteur dans l'espace d'*embedding*.
4. **Une fonction de perte (coût) :** $l : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.
   Cette fonction mesure la qualité de la reconstruction des paires de nœuds de l'espace d'*embedding* vers l'espace direct. En d'autres termes, cette fonction permet de minimiser l'erreur commise par la reconstruction de la manière suivante :

   $$\text{Dec}(\text{Enc}(v_i), \text{Enc}(v_j)) = \text{Dec}(z_i, z_j) \approx s_G(v_i, v_j) \tag{1.44}$$

   Il est bon de noter que la plupart des fonctions de perte effectuent l'optimisation, non pas sur l'ensemble des paires, mais sur un échantillonage des paires de nœuds (noté $\mathcal{D}$) afin de rendre le processus numériquement viable.

   $$\mathcal{L} = \sum_{(v_i, v_j) \in \mathcal{D}} l(\text{Dec}(z_i, z_j), s_{\mathcal{G}}(v_i, v_j)) \tag{1.45}$$

De nombreuses méthodes correspondent bien à ce formalisme. C'est notamment

le cas des méthodes du type *shallow embedding* (Fig. 1.7). Les méthodes de *shallow embedding* sont une classe de méthode où la fonction *encoder* s'écrit sous la forme :

$$\text{Enc}(v_i) = Z\boldsymbol{v_i} \tag{1.46}$$

avec Z, la matrice constituée des vecteurs issus de la projections de chaque nœud du graphe dans l'espace d'*embedding*, et $\boldsymbol{v_i}$, le vecteur indicateur associé à chaque nœud du graphe v_i (vecteur constitué de zéros, sauf en position i associé à la valeur 1). Dans ce cas, l'objectif du processus d'*embedding* est d'optimiser la matrice Z afin d'obtenir une meilleur représentation dans l'espace d'*embedding* des nœuds du graphe, en d'autres termes d'avoir une meilleur correspondance entre les vecteurs de l'espace d'*embedding* et les nœuds du graphe.
Cependant, comme mentionné précédemment, toutes les méthodes ne sont pas englobées dans ce cadre défini par Hamilton et al. [80]. C'est le cas notamment de l'*embedding* sur les hypergraphes où la fonction de similarité ne concerne pas uniquement les paires de nœuds, par définition.

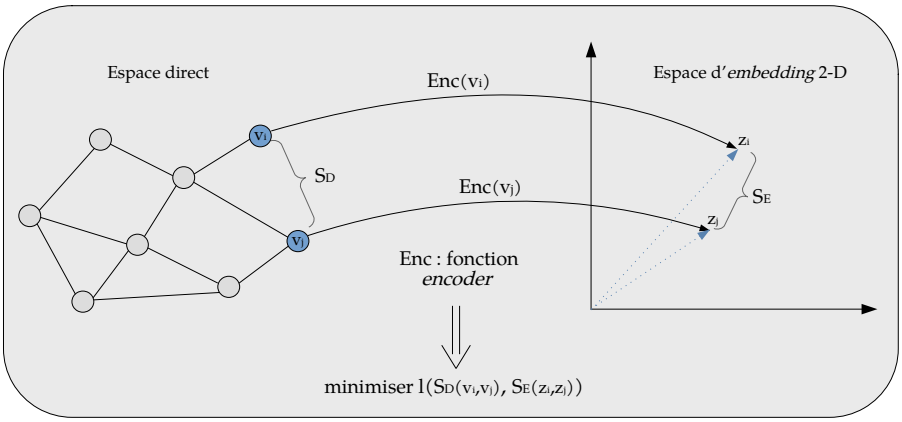


Figure 1.7. – Représentation du *Shallow embedding* : Un graphe est projeté vers un espace vectoriel de basse dimension (ici, 2-D). La fonction *encoder* (Enc) permet de passer de l'espace direct vers l'espace d'*embedding*. Elle est obtenue en minimisant la fonction de perte (l), ce qui minimise l'erreur entre la mesure de similarité entre les nœuds du graphe dans l'espace direct (S_D) et leur projection dans l'espace d'*embedding* (S_E). La fonction S_E est la fonction *decoder*.

Nous allons détailler une méthode, nommée *VERSE* (*VERtex Similarity Embeddings*) [81], qui est au cœur des travaux que l'on présentera en section 9.3. Le principe de la méthode *VERSE* est d'optimiser l'*embedding* d'un graphe en minimisant la divergence de Kullback-Leibler, entre la mesure de similarité entre les nœuds du réseaux dans l'espace direct et la mesure de similarité entre les vecteurs représentant les nœuds dans l'espace d'*embedding*.
On considère un graphe G défini tel que $G = (V, E)$, avec $V = \{v_i,\ i \in [1, n]\}$ l'ensemble

des nœuds du graphe, et E l'ensemble des arêtes du réseau défini tel que $E = \{e_{ij},\ (i,j) \in V \times V\}$. Nous noterons n le nombre de nœuds du graphe. On associe au graphe une fonction de similarité de paire dans l'espace direct, notée $sim_G : V \times V \rightarrow \mathbb{R}^+$. L'*embedding* du graphe G est défini par la représentation matricielle $Z \in \mathbb{R}^{n \times d}$, où les vecteurs z_i sont les représentations vectorielles des nœuds v_i associés au graphe G. Il est bon de noter que l'espace d'*embedding* a une dimension $d \ll n$. On définit dans l'espace d'*embedding* une fonction *decoder* $sim_E : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$, qui détermine une similarité entre les vecteurs de l'*embedding*. L'*embedding* du graphe G est obtenu en optimisant la fonction de perte, notée $\mathscr{L}$, qui correspond à la divergence de Kullback-Leibler entre la matrice de similarité définie entre les nœuds du graphe dans l'espace direct et la matrice de similarité définie entre les vecteurs représentant les nœuds dans l'espace d'*embedding*. Cela se traduit de la manière suivante :

$$\begin{aligned} \mathscr{L} &= \sum_{i=1}^{n} sim_G(v_i,.) \cdot \ln\left(\frac{sim_G(v_i,.)}{sim_E(v_i,.)}\right) \\ &= -\sum_{i=1}^{n} sim_G(v_i,.) \cdot \ln(sim_E(v_i,.)) + C \end{aligned} \tag{1.47}$$

avec $C = \sum_{i=1}^{n} sim_G(v_i,.) \cdot \ln(sim_G(v_i,.))$ une constante qui peut être ignorée dans le cas d'une optimisation. Le vecteur $sim_G(v_i,.)$ correspond au vecteur associé au nœud v_i dans la matrice de similarité définie dans l'espace direct. La matrice de similarité dans l'espace direct (associée à la fonction de similarité de paire), sim_G, peut être définie de plusieurs manières. Les auteurs de l'article original [81] testent trois mesures de similarité différentes : la matrice d'adjacence, la similarité SimRank [54] et la similarité issue des marches aléatoires avec *restart*. Le vecteur $sim_E(v_i,.)$ correspond au vecteur associé au nœud v_i dans la matrice de similarité définie dans l'espace d'*embedding*. Il peut aussi être vu comme le vecteur définissant la similarité entre le vecteur z_i et les autres vecteurs z_j avec $j \neq i, j \in [\![1,n]\!]$, définis dans l'espace d'*embedding*. Les vecteurs constituant la matrice de similarité dans l'espace d'*embedding* (associés à la fonction *decoder*) sont définis par l'équation suivante :

$$sim_E(v_i,.) = \frac{\exp(z_i \cdot Z)}{\sum_{j=1}^{n} \exp\left(z_i \cdot z_j^T\right)}. \tag{1.48}$$

L'encodage des nœuds du graphe vers l'espace d'*embedding* est obtenu en optimisant la fonction de perte à l'aide de la méthode de la descente de gradient. Les vecteurs de la représentation matricielle de l'*embedding* du graphe sont initialement distribués selon une loi normale centrée en zéro. Il est bon de noter que le processus d'optimisation de la fonction de Kullback-Leibler possède une forte complexité numérique et il est préférable d'utiliser une méthode d'échantillonnage négatif (*negative sampling*) comme la méthode *NCE* (*Noise Contrastive Estimation*) [82, 83].

Au cours de ce chapitre, nous avons introduit les définitions et propriétés fondamentales sur les graphes. Nous avons également introduit les mesures et algorithmes sur les graphes qui m'ont été utiles durant mon travail de thèse. Dans les chapitres suivants, nous parlerons plus volontiers de réseaux plutôt que de graphes, aussi bien pour désigner l'objet mathématique que sa matérialisation sous forme de données. Ce choix est motivé par la pratique au sein de la communauté scientifique de la science des réseaux. De manière importante, les graphes sur lesquels les propriétés, mesures et algorithmes ont été définis dans ce premier chapitre sont des graphes classiques ou simples. En effet, bien qu'ils puissent être pondérés ou dirigés, ces graphes sont composés d'un seul type de nœuds et d'arêtes. Ils sont également appelés réseaux monoplexes. Dans le prochain chapitre, nous introduirons des réseaux composés de plusieurs couches d'informations et constitués de plusieurs types de nœuds.

2. Réseaux multi-couches et algorithmes associés

Sommaire

Dans le chapitre précédent, nous avons défini la notion de graphe et les mesures qui leur sont associées. Cependant, les graphes simples (composés d'un seul type de sommets et d'arêtes), introduits il y a plusieurs siècles par Euler, ne sont plus suffisants pour représenter la diversité et l'hétérogénéité des données disponibles actuellement. De nouveaux formalismes ont vu le jour ces dernières années. Parmi les nouveaux formalismes, les réseaux multi-couches sont particulièrement intéressants. L'idée des réseaux multi-couches est de pouvoir intégrer différents réseaux au sein d'un même formalisme, et ainsi, obtenir une représentation plus complète des systèmes que les réseaux représentent. Au sein d'un réseau multi-couche, chaque réseau construit préalablement forme une couche d'informations.

2.1. Les réseaux hétérogènes et multi-couches

Nous allons tout d'abord introduire les réseaux hétérogènes, qui sont des réseaux qui possèdent des nœuds de types différents. Puis, nous introduirons les réseaux multi-couches qui sont constitués de plusieurs réseaux simples, chacun représentant une couche du réseau multi-couche, et qui peuvent être eux-mêmes hétérogènes. Il est bon de noter que les réseaux hétérogènes sont un cas particulier de réseau multi-couche.

2.1.1. Les réseaux hétérogènes

Les réseaux hétérogènes sont des extensions naturelles des réseaux simples. Dans le cas d'un réseau hétérogène, au lieu d'avoir un ensemble unique de nœuds, le réseau possède différents ensembles de nœuds et d'arêtes. Chacun de ces ensembles de nœuds représentant un type d'information particulier et chacun des ensembles d'arêtes un type d'interactions entre nœuds. Nous définissons un réseau hétérogène par le couple $G = (V, E)$ et les fonctions de correspondances $\phi : V \rightarrow A, \psi : E \rightarrow R$ qui associent à chaque nœud et à chaque arête son type. Il est bon de noter que si $|A| = 1$ et $|R| = 1$, le réseau est homogène et on retrouve les notations de la section 1.1. Dans le cadre de ma thèse, je me suis intéressé particulièrement au cas des réseaux bipartites qui connectent deux types de nœuds différents, sans qu'il y ait d'interactions entre nœuds du même type.

- **Les réseaux bipartites :** Il s'agit de réseaux qui connectent des nœuds de deux types différents, comme illustré en Fig. 2.1. Ces réseaux sont couramment utilisés. Par exemple, dans le cadre de réseaux informatiques, on peut imaginer que les nœuds numérotés de 1 à 6 correspondent à des utilisateurs et ceux de a à e à des machines accessibles à ces utilisateurs. Dans ce cas, certains utilisateurs ont accès à différentes machines et certaines machines peuvent être associées à plusieurs utilisateurs. Un second exemple est celui des réseaux biologiques, où, dans ce cas, le réseau bipartite peut être constitué de gènes et de maladies. En suivant la même logique, certains gènes sont responsables de plusieurs maladies et certaines maladies sont causées par plusieurs gènes. Ainsi, un réseau bipartite

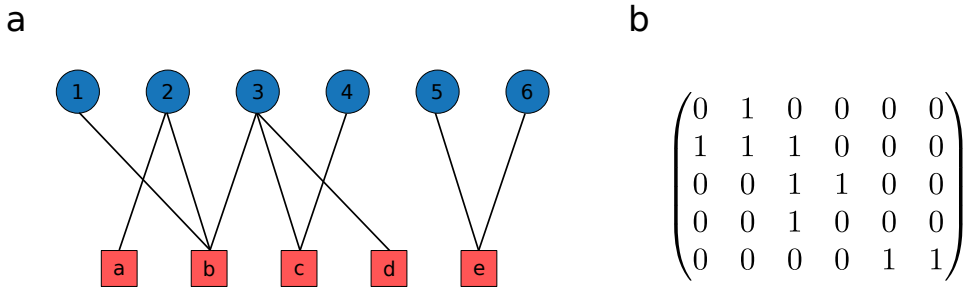


Figure 2.1. – a : Réseau bipartite composé de six nœuds d'un premier type (bleu) et de cinq nœuds d'un second type (rouge). Les deux types de nœuds sont connectés par neuf arêtes bipartites. b : Représentation matricielle (appelée par la suite matrice bipartite) correspondant au réseau bipartite représenté en (a). Les nœuds numérotés de 1 à 6 sont représentés par les colonnes de la matrice et les nœuds numérotés de a à e sont représentés par les lignes de la matrice.

$G_{\mathscr{B}}$, s'écrit comme étant le couple $G_{\mathscr{B}} = (V_{\mathscr{B}}, E_{\mathscr{B}})$, qui se définit comme étant :

$$G_{\mathscr{B}} : \begin{cases} V_{\mathscr{B}} = V_\alpha \cup V_\beta = \{v_i^\alpha, k = 1, ..., n_\alpha\} \cup \{v_j^\beta, l = 1, ..., n_\beta\} \\ E_{\mathscr{B}} = \{e_{i,j}\ i = 1, ..., n_\alpha\ ,\ j = 1, ..., n_\beta\} \end{cases} \tag{2.1}$$

Avec n_α le nombre de nœuds de type v_α, n_β le nombre de nœuds de type v_β. La représentation matricielle de ce réseau est donnée par la matrice bipartite, notée B, et de taille $n_\alpha * n_\beta$ (Fig. 2.1). Matriciellement, cela se traduit de la manière suivante :

$$B_{i,j} = \begin{cases} 1 & \text{si } e_{i,j} \neq 0 \\ 0 & \text{sinon} \end{cases} \tag{2.2}$$

- **Les réseaux multipartites :** De manière analogue aux réseaux bipartites, les réseaux multipartites connectent des nœuds de types différents, en nombre arbitraire. Dans le cas où il y a trois types de nœuds différents, on parle de réseaux tripartites. Le cas des réseaux multipartites dépasse le cadre de ma thèse, mais le lecteur intéressé pourra se référer à l'ouvrage sur la coloration de graphes [84], où le formalisme des réseaux multipartites et k-partites est détaillé.

2.1.2. Les réseaux multi-couches (*multilayer network*)

Les réseaux hétérogènes sont une première tentative d'extension des réseaux simples. Cependant, pour représenter une plus grande diversité des données, il est intéressant de pouvoir intégrer, au sein d'un même système, des réseaux représentant différentes couches d'informations se référant au même ensemble de nœuds. Ainsi, la question de la diversité des données est double. Il faut intégrer des réseaux définissant des informations entre nœuds de différents types (réseaux hétérogènes) et intégrer plusieurs couches d'informations se référant au même ensemble de nœuds (réseaux multiplexes). Par exemple, nous pouvons considérer un réseau multiplex social, dans lequel les nœuds correspondent aux individus et on peut créer plusieurs couches d'interactions à l'aide de l'information représentant les liens professionnels (réseau LinkedIn), ou bien définir des arêtes entre les nœuds représentant des liens amicaux et familiaux (réseau Facebook), ou bien encore des arêtes représentant des liens scientifiques (réseau Researchgate). De plus, les réseaux peuvent avoir une dimension temporelle, et, par conséquent, nous pourrions construire un réseau pour chaque pas de temps disponible. Ainsi, on définit la notion de réseaux multi-couches qui permet l'intégration de différents réseaux au sein d'un même formalisme. Au sein du réseau multi-couche, chaque réseau construit préalablement apporte une couche d'informations. Il est bon de noter que les réseaux multi-couches englobent les réseaux hétérogènes, ainsi que les réseaux multiplexes ou temporels.

- **Les réseaux multi-couches (*multilayer network*) :** Un réseau multi-couche [85–89], noté $\mathscr{M}$, est défini comme un triplet $\mathscr{M} = (Y, G, \mathscr{G})$, avec $Y = \{\alpha, \alpha \in [\![1, M]\!]\}$ et M le nombre de couches du réseau multi-couche. La variable G est définie comme étant la liste ordonnée des couches du réseau $G = (G_1, G_2, ..., G_M)$, telle que $G_\alpha = (E_\alpha, V_\alpha)$, et n_α le nombre de nœuds du réseau (couche) G_α (aussi noté $n_\alpha = |V_\alpha|$). Les réseaux constituant G définissent les intra-liens au sein de chaque couche du réseau multi-couche. La variable $\mathscr{G}$ définit la liste des $M(M-1)$ réseaux bipartites du réseau multi-couche. Chaque réseau bipartite $\mathscr{G}_{\alpha,\beta}$ est donné par $\mathscr{G}_{\alpha,\beta} = (V_\alpha, V_\beta, E_{\alpha,\beta})$. Les réseaux constituant $\mathscr{G}$ définissent les inter-liens entre couches du réseau multi-couche.

- **Les réseaux multiplexes :** Il s'agit d'un cas particulier de réseau multi-couche, qui respecte les propriétés suivantes :
  - Les réseaux multiplexes ont une relation un pour un entre les nœuds des différentes couches ; les nœuds des différentes couches sont nommés nœuds *replica*.
  - Les inter-liens connectent uniquement les nœuds *replica* et tous les nœuds *replica* sont connectés à travers chaque couche.

  Mathématiquement, un réseau multiplex ayant L couches s'écrit comme un réseau multi-couche $\mathscr{M} = (Y, G, \mathscr{G})$, où chacune des couches G_α s'écrit comme $G_\alpha = (V, E_\alpha)$ avec V l'ensemble des nœuds *replica* et $n = |V|$. Chaque couche G_α est caractérisée par sa matrice d'adjacence $A^{[\alpha]}$ où les éléments sont définis comme dans l'équation (1.1). Les réseaux $\mathscr{G}_{\alpha,\beta} = (V_\alpha, V_\beta, E_{\alpha,\beta})$ connectent les nœuds *replica* de la couche α avec ceux de la couche β, dans le cas des réseaux multiplexes $V_\alpha = V_\beta = V$ et $E_{\alpha,\beta}$. Un réseau multiplex est déterminé par sa matrice de supra-adjacence, notée $\mathscr{A}$. Il s'agit d'une généralisation de la matrice d'adjacence. La matrice $\mathscr{A}$ a une taille $(n * L) * (n * L)$ et s'écrit :

$$\mathscr{A}_{i_\alpha, j_\beta} = \begin{cases} A_{ij}^{[\alpha]} & \text{si } \alpha = \beta \\ \delta_{ij} & \text{si } \alpha \neq \beta \end{cases} \tag{2.3}$$

  où δ_{ij} est le symbole delta de Kronecker défini tel que $\delta_{ij} = 1$ si $i = j$ et 0 sinon. Il est possible de visualiser la matrice de la manière suivante :

$$\mathscr{A} = \begin{pmatrix} A^{[1]} & I & \dots & I \\ I & A^{[2]} & \dots & I \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \dots & A^{[L]} \end{pmatrix}, \tag{2.4}$$

  avec I la matrice identité de taille $(n * n)$.

- **Les réseaux temporels :** Il s'agit d'un type particulier de réseaux multi-couches possédant les propriétés suivantes :

– Les réseaux temporels ont une relation un pour un entre les nœuds des différentes couches, qui sont nommés nœuds replica ; chaque couche du réseau représente une étape temporelle du réseau.
– Les inter-liens peuvent connecter uniquement les nœuds replica en suivant l'ordre temporel.

La définition des réseaux temporels est similaire à celle des réseaux multiplexes et possède donc une définition mathématique similaire. La différence entre ces deux types de réseaux réside dans la composante temporelle qui impose une directionalité des inter-liens. Un réseau temporel est caractérisé par sa matrice de supra-adjacence, noté $\mathscr{A}$ et de taille $(n*L)*(n*L)$. La représentation matricielle est la suivante :

$$\mathscr{A} = \begin{pmatrix} A^{[1]} & I & 0 & 0 & 0 & \dots & 0 \\ 0 & A^{[2]} & I & 0 & 0 & \dots & 0 \\ 0 & 0 & A^{[3]} & I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & A^{[L-1]} & I \\ 0 & 0 & 0 & \dots & 0 & 0 & A^{[L]} \end{pmatrix}, \tag{2.5}$$

avec I la matrice identité de taille $(n*n)$. Le lecteur intéressé pourra se référer à la revue suivante [90] pour avoir plus de détails et des utilisations spécifiques des réseaux temporels.

Il est important de préciser que, par la suite, nous nous référerons régulièrement à un type de réseau multi-couche nommé réseau multi-couche universel. Nous avons défini les réseaux multi-couches universels dans notre article *Universal Multilayer Exploration by Random Walk with Restart* (chapitre 6). Un réseau multi-couche universel est un réseau multi-couche qui intègre un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites. Ces réseaux peuvent également être dirigés et pondérés. On peut noter que la définition d'un réseau multi-couche peut convenir à ces réseaux multi-couches universels. Cependant, au vu de la vaste littérature et de la terminologie variée existante [91], nous avons préféré spécifier cette nouvelle définition. Cette définition place en particulier le réseau multiplex (généralisation d'un réseau monoplex) comme élément central d'un réseau multi-couche universel.

Les utilisations des réseaux multi-couches sont nombreuses, que ce soit sous forme de réseaux multiplexes, temporels ou bien de réseaux multi-couches plus généraux. Nous les retrouvons en économie [92, 93], en sciences sociales (réseaux sociaux) [94–97], dans l'étude du comportement animal [98], en écologie [99–101], dans l'étude sur le climat [102], en neurosciences [103–106], dans l'étude des réseaux de transports [107, 108] et en biologie plus généralement [109–112]. Nous reviendrons plus en détail sur les utilisations dans le cas de la biologie dans le chapitre 4.

Le lecteur intéressé pourra se référer aux publications suivantes [89, 91, 113] pour plus de précisions et une classification plus large des réseaux multi-couches.

2.2. Mesures sur les réseaux multi-couches

Comme nous l'avons vu, les représentations des données sous forme de réseaux ont dû évoluer en même temps que les données disponibles afin de prendre en compte leur diversité et leur hétérogénéité. Dans la section 1.2, nous avons défini des mesures sur les graphes. Sachant que les graphes ont évolués vers des réseaux multi-couches, les mesures préalablement définies ne sont plus adaptées. Par conséquent, au même titre que les graphes eux-mêmes, les mesures associées aux graphes ont évolué et se sont généralisées. Nous allons ici détailler différentes mesures qui sont adaptées aux réseaux multi-couches. Il est bon de noter que nous allons définir deux types de mesures sur les nœuds des réseaux multi-couches. Une partie des mesures est définie sur les nœuds à travers les différentes couches. Une autre partie des mesures vise à comparer les propriétés des nœuds entre les différentes couches.

2.2.1. Mesures de centralité sur les réseaux multi-couches

Nous allons définir des mesures de centralité au sein des réseaux multi-couches. Ces mesures sont souvent des extensions des mesures sur les réseaux simples. Nous allons nous restreindre ici au cas des réseaux multiplexes définis dans la section 2.1.2. Cet accent sur les réseaux multiplexes s'explique par le fait qu'ils jouent un rôle prépondérant dans la littérature. Cependant, avant de définir des mesures de centralité, nous allons, tout d'abord, définir certaines propriétés associées aux nœuds d'un réseau multiplex. Dans un premier temps, on définit la notion de degré de recouvrement (*overlapping degree*), noté o_i pour le nœud v_i. Il s'agit du nombre total de voisins que possède chaque nœud à travers chaque couche du réseau multiplex. Elle s'écrit de la manière suivante :

$$o_i = \sum_{\alpha=1}^{M} k_i^{[\alpha]} \tag{2.6}$$

avec M le nombre de couches du réseau multiplex, et $k_i^{[\alpha]}$ le degré du nœud v_i dans la couche α du réseau multiplex. On suppose que le réseau multiplex possède un nombre de nœuds n (nous rappelons que chaque couche du réseau possède les mêmes nœuds, appelés nœuds *replica*). Cette définition nous permet d'introduire deux mesures particulièrement intéressantes dans un réseau multiplex : le coefficient de participation et l'entropie des nœuds. Ces mesures permettent de définir à quel point un nœud a des degrés hétérogènes à travers les différentes couches du réseau

multiplex. Les mesures de centralité que l'on va détailler définissent une mesure sur les nœuds à travers les différentes couches d'un réseau multiplex.

- **Coefficient de participation [114, 115] :** Il s'agit d'une mesure de l'hétérogénéité des degrés des nœuds à travers les différentes couches du réseau multiplex. Elle se définit comme étant :

$$P_i = \frac{M}{M-1}\left[1 - \sum_{\alpha=1}^{M}\left(\frac{k_i^{[\alpha]}}{o_i}\right)^2\right] \tag{2.7}$$

  où $P_i = 1$ quand le degré du nœud v_i est uniformément distribué à travers les différentes couches du réseau multiplex (en d'autres termes quand v_i a le même degré dans chaque couche), et $P_i = 0$ si v_i a tous ses voisins dans une seule couche. Ainsi, un réseau multiplex est d'autant plus homogène à travers ses couches que les coefficients de participation des nœuds sont égaux à 1.

- **Entropie [114, 115] :** Il s'agit d'une mesure permettant de décrire la distribution du degré des différents nœuds à travers les différentes couches du réseau multiplex. Elle est définie de la manière suivante :

$$H_i = -\sum_{i=1}^{M}\frac{k_i^{[\alpha]}}{o_i}\ln\left(\frac{k_i^{[\alpha]}}{o_i}\right) \tag{2.8}$$

  Il est bon de noter qu'il s'agit de l'entropie de Shannon appliquée aux degrés des nœuds.

Nous allons maintenant introduire des définitions qui généralisent certaines mesures déjà définies sur les réseaux classiques.

- **Centralité spectrale (*eigenvector centrality*) [114–116] :** Il existe plusieurs extensions de la centralité spectrale pour les réseaux multiplexes. Nous allons en présenter trois différentes. Dans le premier cas [116], il s'agit de déterminer le vecteur de Perron associé à la matrice. La matrice est définie comme étant la somme des matrices d'adjacences de chacune des couches du réseau multiplex.

$$\tilde{A} = \sum_{\alpha=1}^{M} A^{[\alpha]} \tag{2.9}$$

  La mesure définie dans l'équation représente une centralité spectrale uniforme, c'est-à-dire où chacune des couches du réseau contribue de la même manière sans interférer avec les autres couches. Une autre définition possible [116] revient à prendre en compte les influences qu'ont les couches les unes sur les autres. Pour ce faire, il faut introduire une matrice non négative, nommée matrice d'influence et notée $W \in \mathbb{R}^{M\times M}$. Dans ce cas, on calcule la centralité spec-

trale en prenant en compte la matrice d'influence. La matrice d'influence permet de définir pour chaque couche du réseau une nouvelle matrice d'adjacence de la manière suivante :

$$\check{A}^{[\alpha]} = \sum_{\beta=1}^{M} w_{\alpha,\beta} A^{[\beta]} \tag{2.10}$$

avec α et β deux couches du réseau multiplex. Ainsi, en calculant le vecteur de Perron de chacune des M matrices, on obtient la matrice de centralité spectrale, notée $C^\star$ telle que :

$$C^\star = (c_1^\star, c_2^\star, ..., c_M^\star) \in \mathbb{R}^{n \times M} \tag{2.11}$$

où chacun des $c_\alpha^\star$ correspond au vecteur de Perron associé à la couche α du réseau multiplex.
Dans un troisième cas [114], il est possible de définir un vecteur de centralité spectrale pour le réseau multiplex, et ce, en prenant en compte les contributions de chacune des couches de manière non uniforme, en définissant la matrice suivante :

$$\mathscr{M} = \sum_{\alpha=1}^{M} b_\alpha A^{[\alpha]} \tag{2.12}$$

avec b_α des poids associés à chacune des couches du réseau multiplex, et définis tels que $\sum_\alpha^M b_\alpha = 1$. De nouveau, pour définir la centralité spectrale, on calcule le vecteur de Perron associé à la matrice $\mathscr{M}$.

- **Coefficient de *clustering* [114, 115] :** La généralisation du coefficient de *clustering* aux réseaux multiplexes nous amène à considérer deux cas de figures différents, qui se traduisent par deux coefficients de *clustering*. Cette distinction vient du fait que des triangles peuvent être formés entre des nœuds d'une même couche mais aussi entre nœuds de différentes couches du réseau multiplex. On définit un m-triangle comme étant un triangle constitué par des arêtes appartenant à m couches et la notion de m-triades qui définit une triade (parmi trois nœuds, 2 au moins sont connectés) de nœuds appartenant à m couches différentes. Ainsi, le premier coefficient de *clustering* définit pour chaque nœud v_i le ratio entre le nombre de 2-triangles faisant intervenir le nœud v_i et le nombre

de 1-triades centré sur le nœud v_i. Ce coefficient s'écrit de la manière suivante :

$$C_{i,1} = \frac{\sum_\alpha \sum_{\beta\neq\alpha} \sum_{j\neq i, m\neq i} A^{[\alpha]}_{ij} A^{[\beta]}_{jm} A^{[\alpha]}_{mi}}{(M-1)\sum_{j\neq i, m\neq i} A^{[\alpha]}_{ij} A^{[\alpha]}_{mi}} = \frac{\sum_\alpha \sum_{\beta\neq\alpha} \sum_{j\neq i, m\neq i} A^{[\alpha]}_{ij} A^{[\beta]}_{jm} A^{[\alpha]}_{mi}}{(M-1)\sum_\alpha k^{[\alpha]}_i (k^{[\alpha]}_i - 1)} \tag{2.13}$$

Le deuxième coefficient de *clustering* définit, pour chaque nœud v_i, le ratio entre le nombre de 3-triangles faisant intervenir le nœud v_i et le nombre de 2-triades centré sur le nœud v_i. Ce second coefficient s'écrit de la manière suivante :

$$C_{i,2} = \frac{\sum_\alpha \sum_{\beta\neq\alpha} \sum_{\gamma\neq\alpha,\beta} \sum_{j\neq i, m\neq i} A^{[\alpha]}_{ij} A^{[\gamma]}_{jm} A^{[\beta]}_{mi}}{(M-2)\sum_\alpha \sum_{\beta\neq\alpha} A^{[\alpha]}_{ij} A^{[\beta]}_{mi}} \tag{2.14}$$

Nous avons présenté ces mesures de centralité afin d'illustrer différentes mesures, existant sur les réseaux simples, et qui ont été étendues aux réseaux multiplexes. Le coefficient de participation et l'entropie pour les réseaux multiplexes, présentés en équation (2.7 et 2.8), ont été intégrés dans la librairie Python MultiXrank développée au cours de ma thèse (chapitre 6). Il est bon de noter qu'il existe, bien entendu, d'autres mesures de centralité, aussi bien sur les réseaux multiplexes, que sur toutes autres formes de réseaux multi-couches. Nous pouvons citer notamment la *betweenness* ou la *closeness* [117, 118]. Dans le cadre de ma thèse, je me suis plutôt concentré sur l'extension de mesures de similarité aux réseaux multi-couches. Dans la section suivante, nous allons introduire quelques mesures de similarité sur les réseaux multi-couches préexistantes à mon travail de thèse.

2.2.2. Mesures de similarité sur les réseaux multi-couches

On peut réutiliser les mesures de similarité définies dans la section 1.2.2 et les utiliser directement sur les différentes couches des réseaux multi-couches. Cependant, si les différentes couches du réseau multi-couche ne font pas intervenir les mêmes nœuds, il est plus difficile de pouvoir définir une mesure de similarité. Néanmoins, lorsque l'on considère le cas des réseaux multiplexes, chaque couche du réseau contient les mêmes nœuds. Ainsi, il est simple de considérer, en plus des mesures de similarité entre les nœuds d'une même couche, des mesures de similarité entre les différentes couches du réseau multiplex. Par conséquent, nous allons définir dans un premier temps des mesures de similarité qui visent à comparer des vecteurs de mesure des nœuds entre les différentes couches.
On définit le vecteur $\boldsymbol{v_\alpha}$ représentant un vecteur issu d'une mesure sur les nœuds de la couche α du réseau multi-couche et le vecteur $\boldsymbol{v_\beta}$, représentant un vecteur issu

d'une mesure sur les nœuds de la couche β. Il est possible de définir de nombreuses mesures de similarité entre les deux couches du réseau, à partir de ces deux vecteurs. Par exemple, on peut définir les mesures suivantes :

- La divergence Kullback-Leibler :

$$D_{kL}(\boldsymbol{v_\alpha}, \boldsymbol{v_\beta}) = \sum_{i=1}^{n} (\boldsymbol{v_\alpha})_i \ln\left(\frac{(\boldsymbol{v_\alpha})_i}{(\boldsymbol{v_\beta})i}\right) \tag{2.15}$$

- La similarité cosinus :

$$\sigma(\boldsymbol{v_\alpha}, \boldsymbol{v_\beta}) = \frac{\boldsymbol{v_\alpha}^T \cdot \boldsymbol{v_\beta}}{\|\boldsymbol{v_\alpha}\| \cdot \|\boldsymbol{v_\beta}\|} \tag{2.16}$$

- La corrélation de Pearson :

$$r(\boldsymbol{v_\alpha}, \boldsymbol{v_\beta}) = \frac{(\boldsymbol{v_\alpha} - \langle \boldsymbol{v_\alpha} \rangle)^T \cdot (\boldsymbol{v_\beta} - \langle \boldsymbol{v_\beta} \rangle)}{\|(\boldsymbol{v_\alpha} - \langle \boldsymbol{v_\alpha} \rangle)\| \cdot \|(\boldsymbol{v_\beta} - \langle \boldsymbol{v_\beta} \rangle)\|} \tag{2.17}$$

Il existe bien d'autres mesures de similarité, qui considèrent les réseaux multi-couches [119], basées sur le même principe. Dans le cas présent, nous avons défini une mesure de similarité entre deux couches d'un même réseau. Il est bien entendu possible de définir une matrice de similarité, qui représente les similarités entre toutes les couches prises deux à deux. Nous allons donner le cas de la similarité de Katz que l'on a introduit dans la section 1.2.2, et qui définit une mesure de similarité entre les nœuds à travers les différentes couches d'un réseau multiplex.
La similarité de Katz a été étendue à certains réseaux multi-couches, notamment aux réseaux hétérogènes [120–124] et multiplexes [125]. Nous avons défini dans l'équation 1.30 la similarité de Katz. Dans ce cas, la matrice A était la matrice d'adjacence du réseau. Dans le cas d'un réseau multi-couche, il faut adapter la matrice sur laquelle on itère pour que l'équation prenne en compte les différentes couches du réseau multi-couche. Considérons un réseau multi-couche constitué de deux réseaux simples (notés réseau 1 et réseau 2) non dirigés et non pondérés, définis par deux matrices d'adjacences, notées A_1 et A_2. Les deux réseaux simples sont connectés l'un à l'autre par un réseau bipartite non dirigé défini par la matrice bipartite B. Ainsi, la matrice de transition non normalisée, qui représente l'ensemble des voisins de chaque nœud, est donnée par la matrice suivante :

$$\mathscr{H} = \begin{bmatrix} \mathscr{A}_1 & B \\ B^T & \mathscr{A}_2 \end{bmatrix} \tag{2.18}$$

Sachant que les puissances de la matrice de transition non normalisée donnent les différents voisins de chacun des nœuds, la $k^{\text{ème}}$ puissance de la matrice de transition non normalisée donne accès aux $k^{\text{ème}}$ voisins de chaque nœud. La similarité de Katz

s'écrit comme étant le développement suivant :

$$\sigma = \sum_{m=1}^{\infty} (\alpha \mathscr{H})^m \tag{2.19}$$

Dans le cas où l'on considère un réseau source (s) et un réseau cible (t), on peut définir la similarité de Katz, $\sigma_{s\to t}$ comme étant le développement sur l'ensemble des combinaisons des matrices d'adjacences et bipartites permettant la transition du réseau source au réseau cible. Dans le cas où l'on considère le réseau 1 comme le réseau source, et le réseau 2 comme le réseau cible, alors la similarité de Katz entre les deux réseaux $\sigma_{1\to 2}$ s'écrit comme étant :

$$\begin{aligned}\sigma_{1\to 2} = {} & \alpha B \\ & + \alpha^2 (A_1 B + B A_2) \\ & + \alpha^3 (A_1 B A_2 + A_1^2 B + B A_2^2 + B B^T B) \\ & + \alpha^4 (A_1^3 B + A_1^2 B A_2 + A_1 B A_2^2 + B A_2^3 \\ & + A_1 B B^T B + B B^T B A_2 + B A_2 B^T B + B B^T A_1 B)\end{aligned} \tag{2.20}$$

Dans la précédente équation, on a arrêté le développement au quatrième terme, puisque la série est convergente et qu'il est connu dans la littérature que les termes d'ordre supérieur à quatre ou cinq ne contribuent que très peu à la somme globale [120, 124]. Cette extension présente des lacunes, puisque cette similarité ne peut pas être calculée sur de nombreux réseaux multi-couches, comme par exemple les réseaux multiplexes hétérogènes intégrant un nombre arbitraire de réseaux hétérogènes. Dans la section 9.2, je présenterai une extension réalisée au cours de ma thèse permettant de calculer la similarité de Katz sur les réseaux multi-couches constitués d'un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites.

2.3. Algorithmes sur les réseaux multi-couches

Comme nous l'avons vu dans la section 1.3, les mesures de centralité ou de similarité ne sont pas suffisantes pour exploiter toute l'information disponible dans les réseaux. Ce constat est toujours vrai dans le cadre des réseaux multi-couches. De plus, sachant que la complexité des réseaux augmente de manière croissante, les algorithmes sur les réseaux doivent s'adapter. Dans cette section, nous allons présenter des extensions des algorithmes sur les réseaux simples présentés en section 1.3 aux réseaux multi-couches. Ces extensions sont le point de départ de mon travail, qui a aussi eu pour objectif d'étendre des algorithmes aux réseaux multi-couches. Je vais aussi présenter d'autres méthodes qui m'ont intéressé dans le cadre des réseaux multi-couches, la percolation et la méthode de *cascading failure*, même si ces approches ne sont pas au cœur de mon travail de doctorat.

2.3.1. Algorithmes de partitionnement

De nombreux algorithmes de partitionnement existent pour les réseaux multi-couches. Comme nous l'avons vu dans la section 1.3.1, la modularité permet de mesurer la qualité d'un partitionnement. Elle suppose qu'un bon partitionnement est un partitionnement où le nombre d'arêtes au sein des communautés est important alors que le nombre d'arêtes entre différentes communautés est faible. Nous allons introduire la modularité pour un réseau multiplex, notée Q_m, comme définie dans l'étude de Didier et al. [126] :

$$Q_m = \sum_{\alpha=1}^{M} \frac{1}{2m_\alpha} \left[\sum_{i,j=1}^{n} \left(A_{ij}^{[\alpha]} - \frac{k_i^{[\alpha]} k_j^{[\alpha]}}{2m_\alpha} \right) \delta_{s_i,s_j} \right] \tag{2.21}$$

En d'autres termes, il s'agit de la somme des modularités calculées sur chaque couche du réseau multiplex. À partir de cette modularité-multiplex, les auteurs utilisent l'algorithme de Louvain (voir section 1.3.1) afin de déterminer les communautés. L'algorithme de Louvain est appliqué dans ce cas sur le réseau agrégé obtenu par l'union des couches du réseau. Il existe d'autres méthodes généralisant la modularité, notamment en prenant en compte des termes de couplage entre couches du réseaux. Dans le cas des réseaux temporels, Mucha et al. optimisent leur mesure de modularité avec termes de couplage [127], en utilisant une généralisation de l'algorithme de Louvain nommée GenLouvain.
Bien entendu, des alternatives aux méthodes de partitionnent basées sur la maximisation de la modularité existent. On peut notamment citer la méthode proposée par Z. Kuncheva et G. Montanta [128] qui utilise les marches aléatoires sur les réseaux multiplexes afin de déterminer une mesure de dissimilarité entre nœuds. Le partitionnement du réseau est obtenu en associant les nœuds aux différents clusters basés sur cette mesure de dissimilarité. Une autre méthode basée sur les marches aléatoires consiste à utiliser une version modifiée des marches aléatoires avec *restart* sur les réseaux multiplexes. L'idée de base est de partir d'une graine, puis de faire grossir itérativement cette graine en y intégrant les nœuds les plus proches obtenus par les marches aléatoires, et ce, jusqu'à obtenir une communauté satisfaisant les critères définis par une fonction de qualité. Nous reviendrons sur cette méthode dans la section 9.1, section dans laquelle nous présenterons une méthode de détection de communautés adaptée aux réseaux multi-couches, développée au cours de ma thèse.
Nous conclurons cette partie en mentionnant le concours *Dream Challenge* de 2019, qui a proposé à la communauté bioinformatique la problématique de la détection de communautés pour l'identification de modules dans le cas des maladies complexes [67]. Cette compétition a montré la diversité des méthodes existantes pour le partitionnement des réseaux. De manière intéressante, la compétition s'est intéressée à la fois au partitionnement de réseaux simples et au partitionnement de réseaux multi-couches. Cinq grandes classes de méthodes ont été répertoriées, parmi lesquelles les

méthodes d'optimisation de la modularité et les méthodes basées sur les marches aléatoires que nous avons détaillées précédemment. Mais d'autres approches ont été identifiées, comme les méthodes locales, les méthodes ensemblistes et les méthodes spectrales (*kernel clustering*). La qualité des prédictions offerte par les différentes méthodes est dépendante des propriétés topologiques des réseaux. Par exemple, dans le cas où une méthode met en avant une propriété topologique particulière, cette méthode sera particulièrement efficace sur les réseaux présentant cette particularité topologique, mais elle sera potentiellement médiocre sur des réseaux ayant des propriétés topologiques différentes. Deux autres points à prendre en compte, lorsque l'on considère une méthode de partitionnement de réseaux, y compris sur les réseaux multi-couches, sont la taille des communautés que l'on veut obtenir et si l'on autorise des chevauchements entre communautés (nœuds affectés à plusieurs communautés). Par exemple, dans le cas de la méthode de Louvain (section 1.3.1), plus on itère un grand nombre de fois l'algorithme, plus les communautés deviendront grandes (Fig. 1.6). Il est bon de noter que la méthode de Louvain ne crée pas de chevauchements entre communautés. Cependant, dans la méthode de partitionnement de réseaux multi-couches que j'ai développée dans le cadre de ma thèse, de tels chevauchements entre communautés sont possibles (voir section 9.1).

2.3.2. *Embedding* de réseaux multi-couches

Nous avons vu en section 1.3.2 la notion d'*embedding* de réseaux. Cette notion n'est pas exclusive aux réseaux simples. Il est en effet possible de définir des *embedding* sur des réseaux multi-couches. Par exemple, la méthode d'*embedding VERSE*, vu en section 1.3.2 a été étendue aux réseaux multi-couches constitués de deux multiplexes connectés par un réseau bipartite [129]. Cette extension revient à utiliser comme mesure de similarité dans l'espace direct la matrice de similarité issue des marches aléatoires avec *restart* exécutées sur le réseau multi-couche. Cependant, il est bon de constater que le cas des réseaux multi-couches constitués d'un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites, n'est pas résolu. Nous présenterons en section 9.3, des résultats préliminaires en vue d'une extension de la méthode *VERSE* aux réseaux multi-couches constitués d'un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites (réseaux que nous appelons réseaux multi-couches universels), à l'aide de l'algorithme MultiXrank que nous avons développé au cours de la thèse (voir chapitre 6). Il est possible de définir d'autres algorithmes sur les réseaux simples ou multi-couches, comme par exemple les méthodes de percolation ou de *cascading failure* (voir annexe A.1).

Au cours de cette section, nous avons vu que de nombreuses méthodes dépendent des marches aléatoires. Dans la section suivante, nous allons détailler les bases des marches aléatoires sur les réseaux, après avoir défini les bases des chaînes de Markov.

3. Marche aléatoire sur les réseaux

Sommaire

Les marches aléatoires sont des méthodes couramment utilisées pour explorer la topologie des réseaux. D'une certaine manière, une marche aléatoire peut être décrite comme un marcheur (aléatoire) qui parcourt un réseau, en passant d'un voisin à un autre avec une certaine probabilité [130]. Dans cette section, nous allons détailler les bases théoriques des marches aléatoires et donner des exemples de méthodes de marche aléatoire sur les réseaux. Nous détaillerons notamment la méthode *PageRank*, ainsi que les marches aléatoires avec *restart* (RWR), qui ont été au cœur de mon projet de thèse. L'extension des marches aléatoires avec *restart* aux réseaux multi-couches universels fait l'objet du chapitre 6.

3.1. Des chaînes de Markov à *PageRank*

Dans un premier temps, nous allons donner les bases mathématiques des marches aléatoires. Pour ce faire, il faut introduire le concept de chaînes de Markov. À partir des propriétés des chaînes de Markov, nous verrons un premier exemple de marche aléatoire sur les réseaux avec la méthode *PageRank*.

3.1.1. Marche aléatoire et chaîne de Markov

Les marches aléatoires sur les réseaux sont des processus de Markov, c'est-à-dire des processus stochastiques ayant la propriété d'être sans mémoire (propriété de Markov). En d'autres termes, l'état à l'étape $t+1$, associé à une probabilité p_{t+1}, dépend uniquement de l'état à l'étape t (étape présente), de probabilité p_t, et non des états précé-

dents. Ces états précédents sont associés aux probabilités $\{p_{t-1}, p_{t-2}, ..., p_0\}$. On parle de chaîne de Markov lorsque le processus stochastique concerne une séquence d'évènements, ce qui est le cas d'une marche aléatoire sur un réseau. On définit alors l'espace des états comme l'ensemble $S = \{S_1, S_2, ..., Sn\}$. On définit aussi l'ensemble des variables aléatoires $\{x_t\}_{t=0}^{\infty}$ qui prennent leurs valeurs dans l'espace des états S. Ainsi, pour une chaîne de Markov, on peut définir une probabilité d'un état conditionnellement au passé de la chaîne. Cette probabilité d'un état à l'étape $t+1$ conditionnellement au passé de la chaîne obéit à la propriété de Markov mentionnée précédemment, ce qui se traduit de la manière suivante :

$$\mathbb{P}(x_{t+1} = S_j \mid x_t = S_{i_t}, x_{t-1} = S_{i_{t-1}}, ..., x_0 = S_{i_0}) = \mathbb{P}(x_{t+1} = S_j \mid x_t = S_{i_t}) \tag{3.1}$$

La probabilité $\mathbb{P}(x_{t+1} = S_j \mid x_t = S_{i_t}) = p_{ij}(t)$ définit la probabilité d'être dans l'état S_j à l'étape $t+1$ sachant qu'on était dans l'état S_i à l'étape t. Il s'agit donc de la probabilité de transiter de l'état S_i vers l'état S_j. On l'appellera probabilité de transition. À partir des probabilités de transition, on peut définir la matrice de transition, notée $P(t)$, qui définit la probabilité de passer d'un état à l'autre, pour chaque état et vers n'importe quel état.
Les chaînes de Markov peuvent avoir certaines propriétés particulièrement utiles. Une chaîne de Markov est dite stationnaire si les probabilités ne varient pas au cours du temps, c'est-à-dire si $P(t) = P(t = 0) = P$. De plus, une chaîne de Markov est dite irréductible lorsque tous ses états communiquent, c'est-à-dire que pour toute paire d'états (S_j, S_i), la probabilité d'aller d'un état à l'autre est strictement positive. Une chaîne de Markov est dite apériodique si tout les états ont une période égale à 1 (état apériodique). La période d'un état est définie comme étant le plus grand commun diviseur de l'ensemble $\{t > 0, \mathbb{P}(x_t = S_i \mid x_0 = S_i) > 0\}$. Il est bon de noter qu'une matrice primitive est irréductible et apériodique. Ainsi, pour définir une chaîne de Markov irréductible et apériodique, il suffit de montrer que sa matrice de transition est primitive. Une matrice P est dite primitive s'il existe une puissance k telle que tous les termes de la matrice P^k soient strictement positifs. Dans la suite, nous essayerons de nous placer dans des cas où ces propriétés sont respectées.
Un vecteur de probabilité est défini comme un vecteur non négatif $\boldsymbol{p} = (p_1, p_2, ..., p_n)$, tel que $\sum_k p_k = 1$. Ce vecteur est dit stationnaire s'il obéit à l'équation suivante :

$$\boldsymbol{\pi}^T = P\boldsymbol{\pi}^T \tag{3.2}$$

avec $\boldsymbol{\pi}^T$ qui définit le vecteur transposé du vecteur $\boldsymbol{\pi}$. Enfin, on appelle vecteur de probabilité initiale le vecteur $\boldsymbol{p}(0) = (p_1(0), p_2(0), ..., p_n(0))$, où $p_k(0)$ est la probabilité que la chaîne de Markov débute dans l'état S_k. Nous remarquons ainsi que la matrice de probabilité associée à une chaîne de Markov est une matrice stochastique, c'est-à-dire que chacune des sommes de ses lignes sont égales à 1. On parle dans ce cas de matrice stochastique par ligne. Dans le cas où ce sont les colonnes qui sont normalisées à 1, on parle de matrice stochastique par colonne.
Une propriété remarquable des matrices stochastiques est qu'elles possèdent un

rayon spectral ρ égal à 1. Cette propriété provient directement du théorème de Perron-Frobenius appliqué aux matrices stochastiques. Ce théorème indique que la plus grande valeur propre, notée λ, associée à une matrice stochastique est $\lambda = 1$. De plus, pour les endomorphismes, le rayon spectral est égal à la plus grande valeur propre.

$$\rho(P) = \max_{\lambda \in \sigma(P)} |\lambda| = 1 \tag{3.3}$$

avec $\sigma(P)$ le spectre de la matrice de probabilités P.

En considérant une chaîne de Markov définie comme précédemment, il reste de nombreuses questions sans réponses. Par exemple, quel est le comportement limite d'un tel processus ? Existe-t-il un état stationnaire ? Nous allons répondre à ces deux questions. Dans un premier temps considérons le vecteur de probabilité initial $\boldsymbol{p}(0) = (p_1(0), p_2(0), ..., p_n(0))$. La probabilité d'être dans l'état S_j à l'étape suivante est donnée par l'équation :

$$p_j(1) = \sum_{j=1}^{n} p_i(0)\, p_{ij} \tag{3.4}$$

Ainsi, pour chaque étape, on peut s'inspirer de l'équation précédente pour la réécrire de façon matricielle :

$$\begin{aligned} \boldsymbol{p}^T(1) &= P\boldsymbol{p}^T(0) \\ \boldsymbol{p}^T(2) &= P\boldsymbol{p}^T(1) \\ &\vdots \\ \boldsymbol{p}^T(k+1) &= P\boldsymbol{p}^T(k) \end{aligned} \tag{3.5}$$

Le schéma précédent implique que l'on peut écrire, pour tout k, que :

$$\boldsymbol{p}^T(k) = P^k \boldsymbol{p}^T(0) \tag{3.6}$$

En supposant que l'on a une chaîne de Markov irréductible et apériodique, alors la limite $\lim_{k\to\infty} P^k$ peut être facilement évaluée. Le vecteur de Perron associé à P est égal à $\boldsymbol{e}/n$, avec $\boldsymbol{e}$ le vecteur constitué uniquement de 1 et de taille $(1 * n)$. Donc, si $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)$ est le vecteur de Perron associé à la matrice P^T (i.e $P^T\boldsymbol{\pi}^T = \boldsymbol{\pi}^T$), alors on obtient l'expression suivante :

$$\lim_{k\to\infty} P^k = \frac{(\boldsymbol{e}^T/n)\boldsymbol{\pi}}{\boldsymbol{\pi}(\boldsymbol{e}^T/n)} = \frac{\boldsymbol{e}^T\boldsymbol{\pi}}{\boldsymbol{\pi}\boldsymbol{e}^T} = \boldsymbol{e}^T\boldsymbol{\pi} = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_n \\ \pi_1 & \pi_2 & \dots & \pi_n \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_n \end{pmatrix} > 0 \tag{3.7}$$

Comme P est primitive, alors il existe une distribution limite, que l'on définit comme

le vecteur de probabilité limite et il est donné par :

$$\lim_{k\to\infty} \boldsymbol{p}(k) = \lim_{k\to\infty} \boldsymbol{p}(0)P^k = \boldsymbol{p}(0)\boldsymbol{e}^T\boldsymbol{\pi} = \boldsymbol{\pi} \tag{3.8}$$

Nous considérons à présent le cas particulier des chaînes de Markov sur les réseaux. On considère un réseau non dirigé composé d'une large composante connexe, notée G. Ce réseau est défini par le couple $G = (V, E)$, où V est l'ensemble des nœuds, et $E \subseteq (V^*V)$, l'ensemble des arêtes. Dans ce cas, l'espace des états est défini par l'ensemble des nœuds E, et la variable aléatoire x_t est associée au fait d'être sur un nœud spécifique à l'étape t. On remarque qu'une marche aléatoire sur ce réseau est un processus de Markov irréductible et apériodique. Par conséquent, il existe une distribution stationnaire $\boldsymbol{\pi}$ qui satisfait les propriétés suivantes :

$$\begin{cases} \boldsymbol{\pi}(i) > 0\,; \quad \forall i \in V \\ \sum_{i\in V} \boldsymbol{\pi}(i) = 1 \quad \Leftrightarrow \quad \boldsymbol{\pi}\boldsymbol{e}^T = 1 \end{cases} \tag{3.9}$$

Nous pouvons naturellement introduire la probabilité définissant la marche allant d'un nœud à l'autre. Dans ce cas, on peut imaginer que la variable aléatoire x est un marcheur aléatoire, et donc x_t est sa position à l'étape t, et x_{t+1}, sa position à l'étape $t+1$. Nous considérons les nœuds v_i et v_j, notés i et j. La probabilité s'écrit donc :

$$\mathbb{P}(x_{t+1} = j \mid x_t = i) = \begin{cases} \frac{1}{d_i} & \text{si } (i,j) \in E \\ 0 & \text{sinon} \end{cases} \tag{3.10}$$

avec d_i le degré du nœud i. Toutes les transitions possibles entre nœuds définissent la matrice de transition, qui est à cette la matrice d'adjacence du réseau, notée A. On peut normaliser la matrice d'adjacence afin d'avoir une matrice stochastique et primitive qui définit la matrice de probabilité, notée M. Cette matrice de probabilité peut être vue comme la matrice des degrés de liberté des marcheurs aléatoires au sein du réseau. Le vecteur de probabilité est défini par $\boldsymbol{p}_t = (\boldsymbol{p}_t(i))_{i\in V}$ qui décrit la probabilité d'être dans un nœud i à l'étape t. Et le vecteur de probabilité décrivant la probabilité à l'étape $t+1$ est donné par l'équation de différence linéaire et homogène [131, 132] suivante :

$$\boldsymbol{p}_{t+1}^T = M\boldsymbol{p}_t^T \tag{3.11}$$

La convergence de ce processus itératif permet d'obtenir le vecteur stationnaire $\boldsymbol{\pi}^T$.

3.1.2. *PageRank*

Un exemple bien connu de marche aléatoire sur les réseaux est la méthode *PageRank* [38], que nous avons présenté dans la section 1.2.1. Nous allons ici reformuler la méthode *PageRank* d'une manière plus détaillée et plus proche de sa construction

initiale [132]. L'idée de base de *PageRank* est de simuler le comportement d'un utilisateur de service internet qui navigue d'une page internet à une autre à travers des hyperliens. L'utilisateur peut aussi recommencer le processus de navigation à partir d'une nouvelle page choisie arbitrairement (*restart*). Cette stratégie de marche aléatoire, qui introduit un *restart*, permet d'éviter les impasses (des pages internet qui ne pointent vers aucune autre).
Nous allons formuler mathématiquement la méthode *PageRank*. Comme nous l'avons vu en section 1.2.1, l'idée derrière la méthode *PageRank* (et la mesure de centralité qu'elle définit) est de développer une méthode qui permet de mettre en avant les nœuds qui présentent un fort intérêt, c'est-à-dire les nœuds qui ont un degré élevé et qui sont eux-mêmes connectés à des nœuds de degré élevé. Dans le cas des pages internet, cela revient à mettre en avant les pages qui ont un grand intérêt, c'est-à-dire les pages qui sont référencées par des pages internet présentant elles-mêmes un intérêt ; le tout pondéré par le nombre de pages internet vers lesquelles elles renvoient (degré sortant en d'autres termes). Ainsi, si on considère la page internet v_i, son score *PageRank*, noté $r(v_i)$, s'écrit de la manière suivante :

$$r(v_i) = \sum_{v_j \in V_{v_i}} \frac{r(v_j)}{k_j^{\text{out}}} \tag{3.12}$$

avec V_{v_i} l'ensemble des pages qui pointent vers la page v_i, k_j^{out} le nombre de pages vers lesquelles pointe v_j.
La méthode *PageRank* peut se définir de manière itérative par le processus suivant. Initialement, on démarre sur une page choisie arbitrairement parmi l'ensemble des pages de manière uniforme. Puis, le score *PageRank* associé aux nœuds v_i est défini à chaque étape comme dans l'équation (3.12). Ainsi, on obtient le couple d'équations suivantes :

$$\begin{cases} r_0(v_i) = \frac{1}{n} \\ r_{t+1}(v_i) = \sum\limits_{v_j \in V_{v_i}} \frac{r_t(v_j)}{k_j^{\text{out}}} \end{cases} \tag{3.13}$$

où n définit le nombre total de pages dans le réseau internet et r_0, la condition initiale. On peut ensuite réécrire ce couple d'équations de manière matricielle, en introduisant la matrice des hyperliens, notée H, qui est normalisée et qui contient les probabilités de passer d'une page internet à une autre page internet. Cette matrice est la matrice d'adjacence normalisée du réseau internet, un réseau dirigé. On obtient donc :

$$H_{ij} = \frac{1}{k_i^{\text{out}}} \tag{3.14}$$

Il est bon de noter que H_{ij} définit le lien de la page v_j vers la page v_i. On introduit le vecteur ligne $\boldsymbol{\pi}_t$ qui définit le vecteur de *PageRank* à l'étape t. On obtient donc la

relation de récurrence suivante :

$$\begin{cases} \boldsymbol{\pi}_0^T = \frac{1}{n}\boldsymbol{e}^T \\ \boldsymbol{\pi}_{t+1}^T = H\boldsymbol{\pi}_t^T \end{cases} \tag{3.15}$$

avec $\boldsymbol{e}$ le vecteur constitué uniquement de 1 et de taille $(1 * n)$.
On peut observer que la matrice H définie précédemment pose quelques problèmes. Par exemple, s'il existe une page qui ne pointe vers aucune autre, le navigateur se trouve dans une impasse, et cela se traduit dans la matrice H par une rangée de zéros. Ainsi, la matrice ne sera pas stochastique et la convergence vers un unique état stationnaire est non garantie. Par conséquent, pour avoir un processus de marche aléatoire qui certifie l'existence d'un unique état stationnaire, il est nécessaire d'introduire une nouvelle matrice, qui, elle, est stochastique. On la notera S. On la définit de la manière suivante :

$$S = H + \boldsymbol{a} * (\frac{1}{n}\boldsymbol{e}^T) \tag{3.16}$$

avec $\boldsymbol{a}$ définie telle que a_i est égale à 1 si la page v_i ne pointe vers aucune autre page et 0 autrement. Ainsi, on se retrouve avec une matrice stochastique et primitive, ce qui conduit à ce que notre chaîne de Markov associée soit irréductible et apériodique, et donc qu'il existe une unique distribution stationnaire, notée $\boldsymbol{\pi}$.
De plus, comme nous l'avons mentionné précédemment, nous pouvons ajouter une probabilité de *restart*, notée α, ce qui conduit à définir la matrice de Google, notée G. Cette matrice est la matrice associée à la méthode *PageRank*. Le vecteur de Perron de la matrice de Google définit la centralité de *PageRank*. La matrice de Google s'écrit de la manière suivante :

$$G = \alpha S + (1-\alpha)\frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T \tag{3.17}$$

Ainsi, comme nous l'avons vu dans l'équation (1.51), le vecteur stationnaire correspondant au vecteur *PageRank* peut être obtenu par le couple d'équations suivant :

$$\begin{cases} \boldsymbol{\pi}^T = G\boldsymbol{\pi}^T \\ \boldsymbol{\pi}^T\boldsymbol{e} = 1 \end{cases} \tag{3.18}$$

où la seconde équation correspond à la condition de normalisation. Pour obtenir la centralité de *PageRank*, cela revient à déterminer le vecteur associé à la valeur propre 1, autrement dit, dans le cas des matrices stochastiques, le vecteur de Perron. Le couple d'équations précédent peut être réécrit de la manière suivante :

$$\begin{cases} \boldsymbol{\pi}^T(1-G) = \mathbf{0}^T \\ \boldsymbol{\pi}^T\boldsymbol{e} = 1 \end{cases} \tag{3.19}$$

Dans ce cas, la détermination du vecteur de *PageRank* revient à déterminer le vecteur

qui annule la matrice $(1-G)$. Les deux couples d'équations définis précédemment font appel d'un point de vue numérique à des algorithmes de résolution ayant une forte complexité temporelle. Par conséquent, il est, en pratique, plus efficace de résoudre une version itérative du problème et de définir un critère numérique de convergence. Ainsi, le couple d'équation se réécrit comme étant :

$$\boldsymbol{\pi}_{t+1}^T = G\boldsymbol{\pi}_t^T \tag{3.20}$$

où l'équation de normalisation est vérifiée si le vecteur d'initialisation est lui-même normalisé ; cela est une propriété des matrices stochastiques.

Il existe une version alternative du processus *PageRank*, nommée *personalized PageRank*. Dans ce processus, le vecteur d'initialisation n'est plus forcément uniformément distribué. En d'autre termes, le vecteur d'initialisation $\frac{1}{n}\boldsymbol{e}$ devient simplement un vecteur normalisé $\boldsymbol{v}$. Et ainsi, la matrice de Google s'écrit de la manière suivante :

$$G = \alpha S + (1-\alpha)\boldsymbol{e}\boldsymbol{v}^T \tag{3.21}$$

Il est bon de noter que le changement de vecteur d'initialisation dans la matrice de Google ne change aucunement le processus de détermination du vecteur stationnaire. La méthode *PageRank* est l'une des méthodes les plus répandues de marche aléatoire sur les réseaux. Cette méthode a ouvert la voie à une grande quantité de méthodes qui s'en inspirent, comme la méthode *personalized PageRank*. Dans le cadre de ma thèse, je me suis intéressé au cas des marches aléatoires avec *restart* (RWR), qui est une méthode comparable à la méthode *personalized PageRank*.

3.2. Marche aléatoire avec *restart*

Il existe de nombreuses autres méthodes de marche aléatoire sur les réseaux [130, 133]. Dans le cadre de ma thèse, je me suis concentré sur une méthode directement inspirée de la méthode *PageRank*. Il s'agit de la méthode de marche aléatoire avec *restart* (RWR). Les marches aléatoires avec *restart* sont couramment utilisées pour explorer les réseaux à grande échelle, notamment en bioinformatique. Les stratégies de marche aléatoire avec *restart* ont montré des performances bien supérieures aux méthodes classiques, basées sur des mesures de distances locales, en particulier dans le cas de priorisation d'association entre gènes et maladies [134].

3.2.1. Marche aléatoire avec *restart*, une introduction

Nous avons vu avec l'équation (3.11) qu'une chaîne de Markov sur un réseau peut

s'écrire comme une équation de différence linéaire et homogène. Cela est notamment le cas pour la méthode *PageRank*, comme nous pouvons le remarquer avec l'équation (3.20). Cependant, l'homogénéité n'est pas une condition nécessaire aux processus markoviens. On peut donc définir une équation avec un second membre. En d'autres termes, on peut définir une équation de différence linéaire et non homogène [131], qui correspond à l'équation des marches aléatoires avec *restart*. La marche aléatoire avec *restart* (RWR) est une stratégie alternative à la méthode *PageRank*. Dans le cas des RWR, le *restart* est restreint à un petit nombre de nœuds du réseau, appelés les graines (*seeds*) [135]. Dans cette stratégie, le vecteur stationnaire issu du processus de marche aléatoire peut être vu comme une mesure de similarité entre tous les nœuds du réseau et la ou les graines. Le processus de marche aléatoire avec *restart* peut être également décrit comme un processus de diffusion, où l'objectif est de déterminer l'état stationnaire d'une distribution de probabilité initiale (vecteur d'initialisation) [136]. Mathématiquement, l'équation de marche aléatoire avec *restart* s'écrit de la manière suivante :

$$\boldsymbol{p}_{t+1}^T = (1-r)M\boldsymbol{p}_t^T + r\,\boldsymbol{p}_0^T \tag{3.22}$$

avec M la matrice de transition, qui est stochastique et primitive. Dans le cas des réseaux non dirigés, elle correspond à la matrice d'adjacence normalisée. Le vecteur $\boldsymbol{p}_0$ est défini comme la distribution de probabilité initiale (vecteur d'initialisation), où tous les éléments du vecteur sont nuls, sauf ceux correspondant aux graines. Le paramètre $r \in [0,1]$ représente la probabilité de *restart*. Il est montré dans la littérature que ce paramètre modifie peu les résultats des marches aléatoires, du moins lorsque l'on évite les cas extrêmes (proches de 0 ou de 1). Par conséquent, il est régulièrement fixé, avec comme valeur consensuelle 0.7 [134, 137, 138].

3.2.2. Marche aléatoire avec *restart*, les extensions

De nombreuses extensions aux marches aléatoires avec *restart* ont été développées au cours de la dernière décennie. On peut citer l'extension aux réseaux hétérogènes [137], l'extension aux réseaux multiplexes [86] et l'extension aux réseaux multiplexes-hétérogènes [138]. Dans les marches aléatoires avec *restart*, les degrés de liberté des marcheurs aléatoires sont intégrés dans la matrice de transition, qui correspond aux transitions autorisées entre les différents nœuds du réseaux. L'extension des marches aléatoires avec *restart* à des réseaux multi-couches plus complexes ont été un défi. Il a en effet fallu déterminer cette matrice de transition, ainsi que la normalisation associée, afin de garantir que la matrice soit stochastique et primitive. Nous allons détailler l'extension aux réseaux hétérogènes et mentionner brièvement celle aux réseaux multiplexes-hétérogènes.

Une étude menée par Li et Patra [137] a permis d'étendre les marches aléatoires

avec *restart* aux réseaux hétérogènes. Les réseaux hétérogènes sont des réseaux multi-couches constitués de deux réseaux monoplexes (réseaux simples constitués d'une seule couche) et d'un réseau bipartite, contenant les arêtes reliant les deux types de nœuds des deux réseaux monoplexes. Il est bon de noter que dans leur article, Li et Patra utilisent des réseaux biologiques non dirigés : un réseau d'interaction entre gènes (noté G) et un réseau de relations entre phénotypes (noté P). Le réseau bipartite est, quant à lui, un réseau dans lequel les arêtes représentent les associations bipartites entre gènes et phénotypes. Ces réseaux bipartites sont définis par leurs matrices d'adjacences bipartites. Ainsi, la matrice d'adjacence A_G est associée au réseau de gènes, alors que la matrice d'adjacence A_P est associée au réseau de phénotypes. Dans le cas du réseau bipartite, on définit deux matrices bipartites $B_{G,P}$ et $B_{P,G}$ telles que $B_{G,P} = B_{P,G}^T$, dans le cas où le réseau bipartite est non dirigé. La difficulté est de définir une matrice de transition qui intègre les trois réseaux ainsi définis. On introduit une nouvelle matrice, notée H, qui est la matrice de transition non normalisée du réseau. Elle est définie de la manière suivante :

$$\mathscr{H} = \begin{bmatrix} \mathscr{A}_G & B_{G,P} \\ B_{P,G} & \mathscr{A}_P \end{bmatrix} \tag{3.23}$$

On peut préciser la définition du réseau G_H, en termes d'ensemble de nœuds et d'arêtes de la manière suivante :

$$\begin{cases} G_H = (V_H, E_H) \\ V_H = \{v_{G,i}, i = 1, ..., n_G\} \cup \{v_{P,i}, i = 1, ..., n_P\} \\ E_H = \{e_{i,j}^{1,G}, i, j = 1, ..., n_G, (A_G^{[\alpha_k]})_{i,j} \neq 0\,,\ \alpha_k = 1, ..., L_k\} \\ \qquad \cup \{e_{i,j}^{P,2}, i, j = 1, ..., n_P, (A_P^{[\alpha_k]})_{i,j} \neq 0\,,\ \alpha_k = 1, ..., L_k\}\} \\ \qquad \cup \{e_{i,j}^{G,P}, i = 1, ..., n_G, j = 1, ..., n_P, (B_{i,j}^{[G,P]}) \neq 0\} \end{cases} \tag{3.24}$$

La normalisation de la matrice H permet d'obtenir la matrice de transition normalisée du réseau. Elle est notée $\widehat{S}$ et elle est définie par les équations de normalisation suivantes :

$$\widehat{S}_{GG}(i_G, j_G) = \begin{cases} \frac{A_G(i_G, j_G)}{\sum_{k_G=1}^{n_G} A_G(i_G, k_G)} & \text{si } \sum_{k_G=1}^{n_G} B_{G,P}(i_G, k_P) = 0 \\ \frac{\lambda_{GG} A_G(i_G, j_G)}{\sum_{k_G=1}^{n_G} A_G(i_G, k_G)} & \text{sinon} \end{cases} \tag{3.25}$$

$$\widehat{S}_{PP}(i_P, j_P) = \begin{cases} \frac{A_P(i_P, j_P)}{\sum_{k_P=1}^{n_P} A_P(i_P, k_P)} & \text{si } \sum_{k_P=1}^{n_P} B_{P,G}(i_P, k_G) = 0 \\ \frac{\lambda_{PP} A_P(i_P, j_P)}{\sum_{k_P=1}^{n_P} A_P(i_P, k_P)} & \text{sinon} \end{cases} \tag{3.26}$$

$$\widehat{S}_{GP}(i_G, j_P) = \begin{cases} \frac{\lambda_{GP} B_{G,P}(i_G, j_P)}{\sum_{k_P=1}^{n_P} B_{G,P}(i_G, k_P)} & \text{si} \sum_{k_P=1}^{n_P} B_{G,P}(i_G, k_P) \neq 0 \\ 0 & \text{sinon} \end{cases} \tag{3.27}$$

$$\widehat{S}_{PG}(i_P, j_G) = \begin{cases} \frac{\lambda_{PG} B_{P,G}(i_P, j_G)}{\sum_{k_G=1}^{n_G} B_{P,G}(i_P, k_G)} & \text{si} \sum_{k_G=1}^{n_G} B_{P,G}(i_P, k_G) \neq 0 \\ 0 & \text{sinon} \end{cases} \tag{3.28}$$

où les paramètres $\lambda_{PG}, \lambda_{GP}$ représentent les probabilités de transition entre le réseau de phénotypes et le réseau de gènes, et inversement. Les paramètres $\lambda_{PP}, \lambda_{GG}$ représentent les probabilités de rester, soit dans le réseau de phénotypes, soit dans le réseau de gènes. Ainsi, la matrice de transition normalisée est définie comme étant :

$$\widehat{S} = \begin{bmatrix} \widehat{S}_{GG} & \widehat{S}_{GP} \\ \widehat{S}_{PG} & \widehat{S}_{PP} \end{bmatrix} \tag{3.29}$$

Par conséquent, l'équation de marche aléatoire avec *restart* est donnée par l'équation suivante :

$$\boldsymbol{p}_{t+1}^T = (1-r)\widehat{S}\boldsymbol{p}_t^T + r\boldsymbol{p}_0^T \tag{3.30}$$

Une autre étude menée par Valdeolivas et al. [138] a permis d'étendre les marches aléatoires avec *restart* à certains types de réseaux multi-couches. Dans ce cas, le formalisme, ainsi que l'outil numérique, permettent d'explorer un réseau multiplex associé à un réseau monoplex hétérogène, grâce à un réseau bipartite. Dans cet article, le réseau multiplex est un réseau contenant plusieurs types d'interactions entre gènes et protéines. Le réseau monoplex est un réseau de maladies et le réseau bipartite est un réseau où les arêtes représentent les associations entre gènes et maladies. Dans leur étude, les auteurs se sont restreints au cas de réseaux non dirigés et non pondérés.

Comme nous le voyons, les extensions des marches aléatoires avec *restart* existantes sont limitées en ce qui concerne les types de réseaux qu'elles peuvent explorer, aussi bien en termes de variété, qu'en termes d'hétérogénéité. À ma connaissance, au début de ma thèse, il n'existait pas de méthode de marche aléatoire avec *restart* fonctionnant sur un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites (que nous avons appelé réseaux multi-couches universels), ni de formalisme déterminant une matrice de transition normalisée permettant l'intégration de tels réseaux. De plus, les méthodes existantes se restreignent généralement au cas de réseaux non dirigés et non pondérés.
Le développement d'une méthode permettant l'exploration d'un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites, a été au cœur de mon travail de thèse. Il est bon de noter que ce formalisme permet l'intégration

de réseaux multiplexes et de réseaux bipartites pouvant être dirigés et pondérés. Le formalisme que j'ai développé, ainsi que la librairie numérique associée Multi-Xrank, permettent en effet de répondre à ces problématiques et sont présentés dans le chapitre 6 dédié à l'article *Universal Multilayer Exploration by Random Walk with Restart.*

4. Des réseaux biologiques aux réseaux biologiques multi-couches

Sommaire

Dans les chapitres précédents, nous avons détaillé les bases théoriques de la théorie des graphes, ainsi que les méthodes utilisées pour intégrer, explorer et analyser des réseaux. Nous avons aussi distingué les réseaux simples des réseaux multi-couches, en précisant que ces derniers faisaient écho aux données multidimensionnelles disponibles de nos jours. Dans cette section, nous allons introduire les réseaux biologiques sur lesquels les méthodes vues précédemment peuvent s'appliquer. Nous verrons aussi que l'analyse des systèmes biologiques nécessite l'intégration de données multidimensionnelles et que, par conséquent, les approches "réseaux" en biologie nécessitent l'utilisation de réseaux multi-couches. En effet, l'intégration d'une grande diversité de données et d'échelles permet de décrire les systèmes biologiques de manière plus complète et, ainsi, d'en avoir une meilleure compréhension.

4.1. L'émergence des premiers réseaux en biologie

La biologie des systèmes est une approche holistique de la biologie qui considère les systèmes biologiques comme des systèmes complexes constitués d'une grande et hétérogène variété de composants. Cette approche a connu son essor à la suite du projet génome humain (*Human Genome Project*), qui amena la communauté des biologistes à adopter une vision plus systémique de la recherche en génétique [139]. Cette vision de la biologie est une approche complémentaire aux approches réductionnistes qui ont fait le succès de la biologie moléculaire du XX^{e} siècle, en étudiant les gènes ou les protéines un à un. Cependant, aussi fructueuses qu'ont pu être les approches réductionnistes, elles sont insuffisantes pour déchiffrer la complexité des

systèmes biologiques. Les gènes, les cellules et les organismes n'évoluent pas indépendamment les uns des autres ; ils sont en perpétuelles interactions. Autrement dit, ils appartiennent à un tout et ce tout est plus grand que l'ensemble des parties. De plus, les phénomènes émergents tiennent un rôle central dans les systèmes complexes que sont les systèmes biologiques. Selon Sir Paul Nurse [140], la biologie repose sur quatre grandes idées communément acceptées : (i) le gène est la base de l'hérédité, (ii) la cellule est l'unité fondamentale des organismes, (iii) la biologie est basée sur la chimie, (iv) les espèces évoluent par sélection naturelle. Ainsi, à ces quatre grandes idées qui définissent la biologie, il est aujourd'hui accepté d'ajouter que les systèmes biologiques sont des systèmes complexes multi-échelles [141]. Cette idée est la base de la biologie des systèmes [142]. Par ailleurs, nous pourrions ajouter que la biologie n'est pas uniquement basée sur la chimie, mais aussi sur la physique, comme l'amènent à penser le développement et les succès récents de la biophysique.

Une manière naturelle de représenter et d'analyser la complexité en biologie est de faire appel aux réseaux. Par exemple, les réseaux métaboliques ou les réseaux de régulation génétique ont été un mode de représentation particulièrement pertinent pour visualiser les processus biologiques. Au début du XXI^{e} siècle, la représentation en réseaux est devenue centrale avec l'émergence des technologies de criblage d'interactions à grande échelle. La technologie des cribles double-hybrides, par exemple, a permis l'identification massive d'interactions physiques entre protéines chez plusieurs organismes modèles. Ainsi, les réseaux d'interactions ont vu leurs tailles croître jusqu'à devenir ce que l'on appelle des interactomes. Historiquement, le premier interactome obtenu à partir des cribles double-hybrides à grande échelle a été celui de la levure en 2000, par Uetz et al. [143]. D'autres études similaires ont été publiées peu de temps après.

En 2005, le premier interactome protéine-protéine humain a été publié par Rual et al. [144], peu de temps après celui de la drosophile [145]. La construction de ces interactomes protéine-protéine, et de l'interactome humain en particulier, est toujours un sujet de recherche actif. Des mises à jour régulières des jeux de données d'interaction protéine-protéine sont publiées, en suivant l'évolution des technologies [146, 147]. Il faut noter que la taille de l'interactome protéine-protéine humain est actuellement inconnue. Cela s'explique par la découverte de nouvelles interactions au fil des études menées. Il est ainsi compliqué de déterminer la taille de l'interactome humain, même si des estimations existent [148].

Les interactomes sont de nos jours définis comme des jeux de données d'interactions que l'on représente naturellement sous forme de réseaux. Lorsque l'on s'intéresse aux réseaux moléculaires, dans lesquels les nœuds représentent les gènes, les protéines, ou d'autres molécules biologiques, les interactomes ne représentent plus uniquement les données d'interaction protéine-protéine à grande échelle, mais une grande variété d'interactions, physiques ou fonctionnelles. Nous pouvons distinguer trois grandes classes d'interactomes : i) les réseaux issus de données expérimentales, comme les

réseaux d'interactions protéine-protéine ou les réseaux des complexes moléculaires ; dans ce cas, le réseau est défini naturellement directement par les données expérimentales, sans inférences ni expertise annexe, ii) les réseaux issus de l'expertise, comme par exemple les réseaux des voies de signalisation, ou les réseaux métaboliques, qui sont définis à partir de la littérature et de l'expertise d'une communauté travaillant sur les composants biologiques des réseaux, gènes, protéines, ou métabolites, par exemple, et enfin iii) les réseaux inférés, qui sont issus de l'analyse de données biologiques, en général de type omique. Les réseaux inférés les plus célèbres sont les réseaux de co-expressions, issus en général de la corrélations entre les profils d'expression des gènes, le plus souvent obtenus par (*RNA-seq*).

Il faut également noter qu'en plus de ces réseaux dans lesquels les nœuds représentent des molécules biologiques, d'autres types de réseaux ont été construits ces dernières années. On peut noter en particulier un réseau de maladies dans lequel les nœuds représentent les maladies humaines et les arêtes, leurs associations inférées à partir des gènes causatifs partagés [149]. Il est remarquable que ce réseau possède un chevauchement important avec l'interactome protéine-protéine humain [149, 150]. Ce réseau de maladies a été utilisé, entre autres, pour définir des relations de comorbidités entre maladies, comme nous le verrons dans la section suivante. D'autres réseaux de maladies sont également construits en utilisant les proximités phénotypiques entre les maladies [138]. Des réseaux d'interactions entre médicaments sont également construits à grande échelle, à partir de la similarité chimique entre les composés, ou de la similarité entre leurs effets secondaires [151]. De manière importante, ces nœuds hétérogènes ont également été à la base de la création de réseaux bipartites, qui rassemblent les interactions entre des types de nœuds différents. Ainsi, un réseau bipartite gène-maladie rassemble les associations entre les maladies et leurs gènes causaux. Un réseau bipartite médicament-protéine rassemble les associations entre les médicaments et leurs cibles thérapeutiques.

Les différents interactomes biologiques créés ces dernières années montrent que, du point de vue des données, les réseaux à grande échelle sont une représentation naturelle de la génération grandissante de données. Mais les réseaux ne sont pas intéressants uniquement pour la représentation de données. En effet, la représentation sous forme de réseaux permet l'utilisation des outils de la théorie des graphes pour leurs explorations et analyses. Ainsi, les approches "réseaux" sont devenues omniprésentes dans différents domaines de la biologie [152], particulièrement dans l'étude des maladies humaines [153], où l'on parle de médecine des réseaux (*network medicine*).

4.2. L'émergence des réseaux multi-couches

Nous avons vu que nous pouvions définir une grande variété de réseaux. Par exemple, en biologie moléculaire, les différents réseaux peuvent être classés en trois types :

les réseaux expérimentaux, les réseaux issus de l'expertise et de la littérature et les réseaux inférés. Se pose alors la question de l'intégration de ces différents réseaux. En effet, l'intégration de données est essentielle pour réduire le fossé entre la quantité de données disponibles et notre compréhension de la biologie. En ce qui concerne les données multi-omiques, les différentes omiques capturent différents aspects du fonctionnement cellulaire. Ainsi, leur combinaison permet d'obtenir une description plus complète des systèmes biologiques. De plus, l'utilisation de données omiques complémentaires permet aussi d'obtenir une description des systèmes biologiques moins bruités par les processus expérimentaux inhérents à certains omiques [154]. Dans le cadre des approches "réseaux", il n'est donc plus seulement nécessaire de définir des interactomes, mais aussi des integromes [155], qui sont la représentation issue de la combinaison de plusieurs interactomes. Un autre exemple d'utilisation de plusieurs jeux de données peut être défini dans le cas de l'exposome, qui est l'ensemble des réponses biologiques aux expositions extérieurs aux systèmes biologiques considérés. Il est nécessaire d'intégrer différents jeux de données qui peuvent être représentés sous forme de réseaux, afin de pouvoir capturer la diversité des interactions [156]. Ainsi, les réseaux multi-couches apparaissent comme une réponse naturelle pour représenter des systèmes où plusieurs jeux de données différents sont nécessaires pour capturer la diversité des interactions. Ces dernières années, les réseaux multi-couches ont été utilisés dans de nombreux contextes : pour intégrer des réseaux tissus-spécifiques [110], des réseaux hétérogènes tels que des réseaux métaboliques associés à des réseaux de gènes et de protéines [157], ou bien des couches issues de données de flux métabolique (fluxomique) et de données de transcriptomique [109].

5. Construire des réseaux biologiques

Sommaire

Dans le chapitre précédent, nous avons vu la genèse des approches "réseaux" dans le cadre des données biologiques. Nous avons vu également que nous pouvons définir trois grandes classes de réseaux en biologie : les réseaux expérimentaux, les réseaux issus de l'expertise et de la littérature et les réseaux inférés, en général à partir de données omiques. Il faut cependant noter qu'en général, la construction de ces réseaux biologiques n'est pas évidente, et ce, pour plusieurs raisons. Dans le cas des réseaux expérimentaux, la construction peut paraître plus simple que dans les deux autres cas, puisque les interactions sont directement identifiées expérimentalement. Néanmoins, il se pose la question de trouver des bases de données permettant l'accès aux données voulues.

De plus, il est parfois nécessaire d'associer plusieurs bases de données pour obtenir un réseau plus complet, ce qui peut causer de nombreux problèmes de correspondance entre les éléments des différentes bases de données. Dans le cas des réseaux issus de la littérature ou de l'expertise, il est parfois nécessaire de compiler de nombreuses sources bibliographiques, afin de construire les réseaux. Dans le cas des réseaux inférés, si les données n'ont pas été produites en interne, les mêmes difficultés d'accès aux bases de données se présentent. Difficultés auxquelles s'ajoute celle de l'application ou la création de méthodes statistiques permettant d'inférer un réseau à partir des données. Différentes méthodes existent pour inférer des réseaux. Dans ce cas, se pose la question de l'ajustement des paramètres de la méthode en fonction des données. Nous verrons également par la suite l'utilisation de mesures de similarité pour inférer des réseaux à partir de données.

Ainsi, la construction des réseaux biologiques est un véritable enjeu. La construction de réseaux a été une partie non négligeable de mon travail de thèse, menée en

amont et parallèlement à mes travaux sur le développement de nouvelles méthodes d'intégration, d'exploration et d'analyse de réseaux multi-couches. Je vais donc décrire dans cette section de type "matériel et méthodes" les différents réseaux biologiques auxquels je me suis intéressé, ainsi que les protocoles que j'ai développés pour les construire.

Je vais décrire dans un premier temps les réseaux biologiques monoplexes et multiplexes que j'ai construits. Ces réseaux monoplex et multiplex sont composés, chacun, d'un seul type de nœuds : un réseau multiplex de gènes et de protéines, un réseau monoplex de maladies, et un réseau multiplex de médicaments. Ces trois réseaux multiplexes et monoplexes ont été fondamentaux dans l'application des méthodes que j'ai développées dans ma thèse.

De plus, pour définir des réseaux multi-couches universels, il est nécessaire de construire des réseaux bipartites permettant de connecter les différents types de nœuds des différents réseaux. Les réseaux bipartites seront présentés dans la deuxième section : réseau bipartite entre gènes/protéines et maladies, entre gènes/protéines et médicaments, et enfin entre maladies et médicaments.

Enfin, il est bon de noter que ces différents réseaux, monoplexes, multiplexes et bipartites ne représentent pas l'ensemble des réseaux biologiques existants [158]. Ainsi, dans la troisième section, nous présenterons des réseaux génomiques qui intègrent des informations concernant le génome humain. Ces réseaux génomiques ont été développés dans le cadre de mon second article de thèse (chapitre 7) et viennent compléter les premiers réseaux biologiques présentés.

5.1. Construction de réseaux biologiques monoplexes et multiplexes

Dans cette section, nous allons détailler les différents points considérés pour la création des trois réseaux biologiques monoplexes et multiplexes à la base des différentes applications des méthodes que j'ai développées au cours de ma thèse et qui sont présentées aux chapitres 6, 7 et 9. Ces réseaux sont soit des réseaux multiplexes, comme le réseau de gènes ou de médicaments, soit un réseau monoplex, dans le cas du réseau de maladies.

5.1.1. Réseau multiplex de gènes et de protéines

Il est important tout d'abord de noter que, dans le cas des réseaux de gènes et de protéines, on suppose qu'il y a une correspondance parfaite entre ces deux entités biologiques, c'est-à-dire qu'un gène code une protéine. Ainsi, les nœuds correspondant aux gènes ou aux protéines seront étiquetés par le même nom. Cette

hypothèse est forte et ignore donc les phénomènes d'épissage ou de modification post-traductionnelle qui amènent à ce qu'un gène puisse coder plusieurs protéines. Cependant, les interactions n'étant cependant pas définies, avec une granularité prenant en compte ces phénomènes, les réseaux restent construits à l'échelle des gènes et protéines considérés indifféremment. Une conséquence directe de cette hypothèse est que l'on peut définir des réseaux multiplexes dans lesquels certaines couches contiennent des interactions entre protéines et d'autres, des interactions entre gènes.

Les réseaux d'interaction protéine-protéine ont été les premiers réseaux biologiques à grande échelle construits ; ils ont permis de démontrer l'intérêt de la théorie des graphes pour leur exploration et leur analyse. Cependant, il existe de nombreux autres réseaux d'interactions entre gènes ou protéines : les réseaux de co-expression, ou les réseaux de voies de signalisation, par exemple.
Je vais brièvement détailler les réseaux ainsi que les bases de données que j'ai utilisées dans le cadre de ma thèse et de mes articles de recherche. J'ai défini et utilisé le réseau multiplex de gènes/protéines composé des trois couches suivantes :

- Interactome protéine-protéine (*Protein-Protein Interaction (PPI)* : Cette couche est constituée de la fusion de 3 jeux de données : APID (homo sapiens niveau 2, sans les interactions inter-espèces), Hi-Union et Lit-BM (www.interactome-atlas.org/download). Il est important de noter que plusieurs protéines n'ont pas de correspondance dans l'annotation des noms de gènes. Dans ces cas, nous avons choisi d'utiliser l'identifiant Uniprot.

- Complexome : Cette couche est composée des interactions issues des complexes protéiques. Elle résulte de la fusion des jeux de données suivants : Hu.map [159] et Corum [160], en utilisant OmniPathR [161]. Les complexes sont définis sous forme de cliques (*matrix model*) dans le réseau ; chaque clique est constituée de tous les nœuds correspondant aux protéines d'un complexe protéique.

- Reactome : Cette couche est composée des voies de signalisation. Le réseau a été extrait depuis NDEx [162] et elle correspond aux données du Reactome de Croft et al. [163], pour l'humain. Les directions des interactions, présentes dans les voies de signalisation de Reactome, ont été supprimées pour obtenir un réseau non dirigé.

Comme indiqué plus haut, il est essentiel que les formats des noms des nœuds des différents réseaux correspondent les uns aux autres. Il existe plusieurs formats pour les noms de gènes, comme par exemple le format utilisant les identifiants Ensembl (sous la forme ENSGXXXXXXXXXXX), ou bien les noms des gènes officiels. Pour les protéines, il est possible d'utiliser les identifiants UniProt. Dans le cadre de mon travail de thèse, nous avons choisi l'annotation officielle des noms de gènes pour

définir les gènes, ainsi que les protéines correspondantes.

5.1.2. Réseau monoplex de maladies

Les maladies peuvent être définies comme des ensembles de phénotypes. Ces phénotypes sont notamment répertoriés dans l'ontologie des phénotypes humains (*The Human Phenotype Ontology (HPO)*) [164–166]. Les ontologies sont une manière qui permet, entre autres, d'organiser des termes appartenant à une arborescence, comme le cas des phénotypes. Les ontologies jouent un rôle essentiel dans la recherche biomédicale, en raison de leurs nombreuses caractéristiques. En effet, elles fournissent des identifiants et un vocabulaire pour les différentes classes et relations au sein du domaine ontologique. Elles fournissent aussi des métadonnées décrivant la signification des classes et des relations. Enfin, elles fournissent des assertions et des définitions lisibles par machine, ce qui permet l'intégration et l'analyse des classes et des relations de manière automatisée [167]. L'ontologie *HPO* est une unification sémantique des phénotypes associés aux maladies humaines [168]. Elle permet l'exploration des termes de manière ordonnée. Ainsi, l'ontologie *HPO*, avec l'ensemble des associations entre maladies et phénotypes qu'elle fournit, a été fondamentale dans la création du réseau de maladies que j'ai utilisé dans le cadre de ma thèse.

J'ai construit et utilisé un réseau monoplex de maladies construit à partir de leurs proximités phénotypiques. Ces similarités phénotypiques sont calculées à partir de l'information mutuelle entre les maladies au sein de l'ontologie *HPO*. En d'autres termes, une maladie est décrite comme étant un ensemble de phénotypes. La similarité phénotypique entre deux maladies peut être calculée comme le nombre de phénotypes qu'elles partagent. Cependant, certains phénotypes sont plus pertinents que d'autres. Par exemple, deux maladies partageant un phénotype rare sont sans doute plus similaires que deux maladies partageant un phénotype fréquent [169]. Ainsi, nous pouvons définir la pertinence d'un phénotype comme étant liée à la fréquence de ce phénotype parmi l'ensemble des phénotypes des maladies de l'ontologie. On la définit comme la quantité d'informations, et cela s'écrit :

$$I_c(i) = -\ln\left(f_i\right) \tag{5.1}$$

avec f_i la fréquence du phénotype i, parmi l'ensemble des maladies de l'ontologie. Et la similarité entre deux phénotypes i et j s'écrit comme étant :

$$S(i,j) = \max_{t \in anc(i) \cap anc(j)} I_c(t) \tag{5.2}$$

avec $anc(i)$ l'ancêtre du phénotype i au sein de l'arbre de l'ontologie. Par conséquent, la similarité entre deux maladies D_a et D_b est définie comme la somme pondérée des

quantités d'informations des phénotypes partagés entre les deux maladies [170].

$$S(D_a, D_b) = \frac{1}{|D_a|} \sum_{i \in D_a} \max_{j \in D_b} \big(S(i,j)\big) + \frac{1}{|D_b|} \sum_{j \in D_b} \max_{i \in D_a} \big(S(i,j)\big) \tag{5.3}$$

Il est bon de noter que dans le cas où l'on applique cette définition de la similarité, la grande majorité des paires de maladies auront une similarité non nulle, à l'exception du cas où les deux maladies ne partagent aucun phénotype. Par conséquent, pour construire un réseau à partir de ces similarités, il faut définir un critère permettant de réduire le nombre de voisins pour chaque nœud. Pour ce faire, deux choix sont possibles. Le premier consiste à définir un seuil et enlever l'ensemble des interactions inférieures à ce seuil. Cependant, la détermination d'un seuil reste arbitraire. Un autre choix consiste à garder le top-k des maladies les plus similaires à chaque maladie. Dans notre cas, nous avons adopté cette deuxième solution, avec $k = 5$, de manière similaire à d'autres études [137, 138].

Par ailleurs, il est possible de définir une mesure de similarité plus précise entre deux maladies, en prenant en compte la fréquence du phénotype pour la maladie, en plus de la fréquence du phénotype dans la base de données, si elle est connue. En effet, les maladies sont associées à un ensemble de phénotypes, mais ces différents phénotypes ne sont pas systématiquement présents chez tous les patients : certains phénotypes sont systématiquement présents alors que d'autres apparaissent plus rarement. Ainsi, deux maladies sont d'autant plus proches qu'elles partagent un phénotype rare et que ce phénotype est fréquent dans chacune des deux maladies. Cette nouvelle similarité entre les deux maladies s'écrit de la manière suivante :

$$S(D_a, D_b) = \frac{1}{|D_a|} \sum_{i \in D_a} \max_{j \in D_b} \big(S_f(i,j)\big) + \frac{1}{|D_b|} \sum_{j \in D_b} \max_{i \in D_a} \big(S_f(i,j)\big) \tag{5.4}$$

$$S_f(i,j) = \max_{t \in anc(i) \cap anc(j)} I_c(t) \cdot \Big[\big(-\ln\big(1 - f_i^a\big)\big) \cdot \Big(-\ln\Big(1 - f_i^b\Big)\Big)\Big] \tag{5.5}$$

avec f_i^a la fréquence du phénotype i pour la maladie a et f_i^b la fréquence du phénotype i pour la maladie b. Dans notre cas, il était pour le moment pas envisageable d'utiliser cette formulation puisque dans *HPO*, le nombre de maladies pour lesquelles la fréquence de chaque phénotype est connue restait très faible. Cependant, il serait pertinent de reconstruire le réseau entre maladies avec cette nouvelle formule lorsque les informations concernant la fréquence des phénotypes pour chaque maladie seront plus accessibles.

Nous avons choisi le format *UMLS* (*Unified Medical Language System*) [171] pour nommer les nœuds de maladies au sein du réseau. Le choix de ce format plutôt que du format *OMIM* (*Online Mendelian Inheritance in Man*) [172], très répandu également, vient du fait que la base de données *OMIM* est axée sur les maladies rares. Sachant que l'on souhaite construire un réseau représentant les interactions des maladies rares,

mais également des maladies communes, cela aurait pu être limitant. De plus, une partie des réseaux bipartites que nous détaillerons dans la section suivante utilisent d'autres formats que *OMIM*. *UMLS* propose également une large librairie sémantique dans laquelle la grande majorité des identifiants *OMIM* ou d'autres bases de données ont leurs correspondants.

Dans le cadre de mon travail, j'ai utilisé le réseau monoplex de maladies défini précédemment. Cependant, nous pouvons noter que des données accessibles récemment pourraient en théorie permettre de construire un réseau multiplex de maladies. Un tel réseau multiplex pourrait être constitué de deux couches, une première couche où les arêtes représenteraient une proximité phénotypique comme précédemment, et une deuxième couche où les arêtes représenteraient une proximité génotypique [111]. Deux maladies sont dites proches d'un point de vue génotypique, si elles sont associées au moins à un gène en commun. En conclusion, l'utilisation d'un réseau de maladies (et de sa combinaison aux réseaux de gènes et de protéines que nous décrirons dans la section dédiée aux réseaux bipartites) permet de capturer une information biologique au niveau des phénotypes, ce qui est complémentaire à l'information moléculaire.

5.1.3. Réseau multiplex de médicaments

Dans la littérature, il existe plusieurs manières de définir une similarité entre médicaments :

1. **Une similarité basée sur la structure chimique** [151, 173] de deux médicaments. Cette similarité est définie comme le recouvrement des deux clés *MACCS* représentant chacune une molécule médicamenteuse. Une clé *MACCS* est la représentation binaire de la structure chimique d'une molécule. Deux médicaments possèdent une structure chimique d'autant plus similaire que leur coefficient de Tanimoto (équivalent de l'indice de Jaccard) est élevé. Ce coefficient se définit de la manière suivante :

$$T = \frac{c}{a+b-c} \tag{5.6}$$

   avec a et b le nombre de bits de chacune des clés *MACCS* des deux médicaments et c le nombre de bits communs entre les deux médicaments. Si $T = 0$, les deux molécules ne partagent aucun bit.

2. **Une similarité basée sur les cibles (protéines)**, où il est possible de calculer le coefficient de Tanimoto, non pas entre les structures chimiques des deux molécules médicamenteuses, mais entre les ensembles de protéines avec lesquelles elles interagissent [174]. Il est aussi possible de déterminer des similarités entre médicaments, à partir du réseau bipartite protéine-médicament, où deux médicaments

sont liés s'ils ont la même cible protéique [173, 175].

3. **Une similarité basée sur l'expression des gènes** est envisageable à l'aide des données de réponse d'expression génique aux médicaments [173]. Ces données sont accessibles depuis le projet *Connectivity map* [176] (https://clue.io). À partir de ces données, il est possible de définir des mesures de similarité entre les différents profils d'expression des gènes en réponse aux médicaments. Différentes mesures existent, par exemple :
   - La corrélation de Spearman
   - L'indice de Jaccard entre les 500 gènes les plus différentiellement exprimés (250 sur-exprimés et 250 sous-exprimés) [173]
   - L'analyse d'enrichissement de gènes (*Gene set enrichment analysis (GSEA)*) utilisée comme mesure de similarité [177]

4. **Une similarité basée sur les effets secondaires**. Par exemple, si l'on considère deux médicaments, notés D_a et D_b, et que l'on note s_a et s_b l'ensemble des effets secondaires de chacun des deux médicaments, on peut définir que ces deux médicaments sont d'autant plus similaires que l'indice de Jaccard entre les deux ensembles d'effets secondaires s_a et s_b est élevé [173].

5. **Une similarité basée sur la classification anatomique** [173]. La classification hiérarchique *ATC (Anatomical Therapeutic Chemical)* est un système de classification qui associe à chaque médicament l'organe qu'il affecte, ainsi que les caractéristiques chimiques et thérapeutiques du médicament. Il est donc possible de définir une similarité entre deux médicaments, en utilisant un algorithme de similarité sémantique [170] qui explore cette classification hiérarchique. Cet algorithme est analogue à celui présenté dans le section 5.1.2 dédié aux réseaux de maladies, et où la similarité entre deux maladies est calculée à partir de l'exploration de l'ontologie *HPO*, qui est également une classification hiérarchique.

Il existe bien d'autres méthodes, comme l'utilisation de méthodes hybrides qui font appel à la fois aux structures chimiques des médicaments et aux profils d'expression des gènes [178]. Il est aussi possible de construire un réseau à partir des interactions identifiées dans la littérature [151].

Il est bien entendu possible de combiner plusieurs réseaux de médicaments au sein d'un réseau multiplex. Il s'agit d'une approche que j'ai adoptée dans le cadre de ma thèse et de mes articles de recherche. J'ai défini et utilisé un réseau multiplex de médicaments composé de quatre couches. Trois des quatre couches sont extraites de données provenant de l'étude de Cheng et al. [151][†], et la dernière couche est issue des interactions pharmacologiques entre médicaments provenant de snap.stanford.edu[‡].

- Réseau 1[†] : Cette couche représente les interactions entre médicaments rapportées dans des études cliniques. Cette couche est issue de données répertoriées manuellement à partir de dizaines de milliers d'articles cliniques.

- Réseau 2[†] : Cette couche représente les combinaisons entre médicaments obtenues expérimentalement et répertoriées par Drugbank.

- Réseau 3[†] : Cette couche représente les combinaisons entre médicaments prédites à l'aide d'une mesure de distance sur les réseaux (*separation measure*). Cette mesure de distance détermine des similarités entre médicaments. Les similarités obtenues sont corrélées à celles obtenues en utilisant des similarités basées sur les structures chimiques (1), sur les cibles (2), sur l'expression des gènes (3) et sur la classification anatomique (5) [151].

- Réseau 4[‡] : Cette couche représente les interactions entre médicaments déterminées par les effets pharmacologiques de l'action d'un médicament sur un autre médicament.

Le format des nœuds du réseau a été choisi en suivant le format de la base de données DrugBank [179], une des bases de données de référence pour les médicaments.

5.2. Construction des réseaux bipartites

Au cours de ma thèse et de mes travaux de recherche, nous avons eu pour objectif d'assembler et d'explorer des réseaux multi-couches universels. Ces réseaux intègrent réseaux multiplexes ou monoplexes, connectés grâce à des réseaux bipartites. Dans la première section de ce chapitre dédié à la construction des réseaux biologiques, nous avons défini les réseaux multiplexes et monoplexes de gènes/protéines, de maladies et de médicaments. Il est ensuite nécessaire de construire des réseaux d'interactions bipartites pour relier ces différents types de nœuds : réseaux bipartites entre gènes et maladies, entre gènes et médicaments, ou entre maladies et médicaments. Dans cette section, nous allons détailler ces trois types de réseaux bipartites.

Le réseau bipartite entre protéines/gènes et maladies est défini tel qu'une maladie et un gène sont connectés si le gène est responsable de la maladie, ou du moins joue un rôle dans l'apparition de cette maladie. Ce réseau bipartite est essentiel pour le diagnostic, la compréhension des mécanismes et la thérapie des maladies [180]. Le réseau bipartite entre protéines/gènes et maladies permet, entre autres, de s'intéresser à la priorisation de gènes responsables de maladies, ainsi que de capturer la complexité des relations entre génotype et phénotype. En intégrant des données d'interaction

entre protéines, de similarité entre phénotype et maladies et entre phénotype et gènes [181], ces approches offrent aussi de nouvelles pistes pour explorer les relations de comorbidités entre maladies [153]. Dans le cadre de ma thèse, j'ai construit un réseau bipartite gène-maladie à partir de la version 2020 de la base de données DisGeNET (v7.0) [182]. Cette base de données rassemble des associations entre gènes et maladies à partir de différentes sources, notamment OMIM [172]. Ces associations sont pondérées avec des scores calculés à l'aide de formules définies sur le site de la base de données https://www.DisGeNET.org/dbinfo. Ces scores sont utilisés pour pondérer les arêtes du réseau construit à partir de la base de données.

Le réseau bipartite entre gènes/protéines et médicaments est défini tel qu'un médicament et une protéine sont connectés, si la protéine est une des cibles du médicament [183]. Dans le cadre de ma thèse, j'ai défini un réseau bipartite gène-médicament grâce à la fusion de plusieurs bases de données : la version 5.1.8 des associations médicament-cible de la base de données DrugBank go.drugbank.com/releases/latest, la version v10.12 des associations médicament-cible de la base de données drugcentral drugcentral.org/download, et la base de données reportant les associations médicament-cible extraite de l'article Cheng et al. [151].

Le réseau bipartite entre maladies et médicaments est défini tel qu'un médicament et une maladie sont connectés si le médicament est utilisé pour traiter la maladie.
Il est possible de définir ces associations à partir de bases de données comme DrugBank. Récemment, Guney et al. [184] ont pu déterminer l'efficacité des traitements et distinguer ceux qui sont palliatifs de ceux qui sont curatifs, en utilisant des associations issues d'un réseau bipartite entre maladies et médicaments. Dans le cadre de ma thèse, j'ai défini un réseau bipartite maladie-médicament en utilisant des associations entre maladies et médicaments issues de la base de données repoDB, définie dans Brown et al. [185].

Les différents réseaux décrits jusqu'ici ont permis de construire un réseau multi-couche universel constitué de différents réseaux biologiques. Ce réseau multi-couche universel est constitué de trois types de nœuds : des nœuds correspondant aux gènes/protéines inclus dans le réseau multiplex de gènes/protéines décrit en section 5.1.1, des nœuds correspondant aux maladies dans le réseau monoplex de maladies décrit en section 5.1.2, et des nœuds correspondant aux médicaments inclus dans le réseau multiplex de médicaments décrit en section 5.1.3. Les trois réseaux multiplexes ou monoplexes constitués de ces différents types de nœuds sont connectés entre eux par les réseaux bipartites décrits en section 5.2 (un réseau bipartite gène-maladie, un réseau bipartite gène-médicament et un réseau bipartite maladie-médicament). Dans la littérature, il existe d'autres exemples de réseaux biologiques multi-couches universels. Un exemple particulièrement intéressant est le réseau *Hetionet* [186]. Ce réseau multi-couche universel intègre des réseaux construits à partir d'informations

issues de plusieurs millions d'articles de recherches biomédicales et de plusieurs dizaines de bases de données. Ce réseau contient neuf types différents de nœuds qui sont : des gènes, des maladies, des médicaments, des parties anatomiques, des symptômes, des effets secondaires, des classes pharmacologiques, des voies de signalisations et des processus biologiques. Les différentes couches du réseau sont connectées par une dizaine de réseaux bipartites, comme par exemple des réseaux bipartites gène-maladie, gène-médicament, maladie-médicament, mais aussi un réseau bipartite entre les gènes et les parties anatomiques. Hetionet est une illustration remarquable de la richesse et de la complexité des réseaux biologiques.

5.3. Construction des réseaux génomiques

Les réseaux biologiques ne se restreignent pas aux réseaux de gènes, de maladies ou de médicaments. En particulier, à l'échelle moléculaire, malgré la diversité des réseaux présentés, nous avons jusqu'ici uniquement pris en compte les régions codantes du génome. Dans ces réseaux, les nœuds sont les gènes ou les protéines. Cependant, il est connu que les régions non-codantes du génome, ainsi que son organisation tridimensionnelle, jouent un rôle central dans la régulation de l'expression des gènes [187, 188] et les maladies [189, 190]. Un exemple de mécanisme tridimensionnel pouvant affecter la régulation est celui du repliement de la chromatine. Ce repliement peut amener à ce qu'un élément de régulation, comme un *enhancer*, soit proche de son gène cible et active sa transcription [191]. Par conséquent, pouvoir considérer ce type d'information génomique au sein des approches "réseaux" semble être pertinent pour mieux comprendre les traits phénotypiques et les maladies. De plus, utiliser des réseaux construits à partir de données génomiques représente l'opportunité de compléter les réseaux moléculaires déjà construits à l'échelle des gènes et des protéines, et d'obtenir ainsi une représentation des systèmes biologiques plus réalistes. Dans cette section, nous détaillerons les moyens d'accéder à la conformation tridimensionnelle du génome et comment, à partir de ces informations, construire des réseaux génomiques.

Il est connu que le noyau des cellules humaines est constitué de quarante-six chromosomes densément compactés. Au sein de ces chromosomes, une structure hiérarchiques existe pour empaqueter et organiser le génome de manière fonctionnelle. Les territoires chromosomiques sont définis comme étant les régions préférentiellement occupées par les différents chromosomes au sein du noyau. Chaque chromosome est partitionné en sous-compartiments chromosomiques dans lesquels se trouvent des structures appelées *TADs (topologically associating domains)* [192, 193]. Les TADs sont eux-mêmes constitués de boucles de chromatine assurés par des sites de liaisons CTCF [194] ou des contacts entre promoteurs et *enhancers* [187] (Fig. 5.1.A). Un *TAD* est une région génomique dans laquelle les fragments de chromatine interagissent plus

fréquemment entre eux qu'avec les fragments de chromatines se situant en dehors du TAD (Fig. 5.1.B). De plus, les interactions entre régions régulatrices et gènes ont lieu préférentiellement au sein de ces *TADs* [195]. Les *TADs* peuvent être déterminés à partir d'une matrice de contact (Fig. 5.1.B et Fig. 5.2) qui est une représentation matricielle des contacts entre les fragments de chromatine.

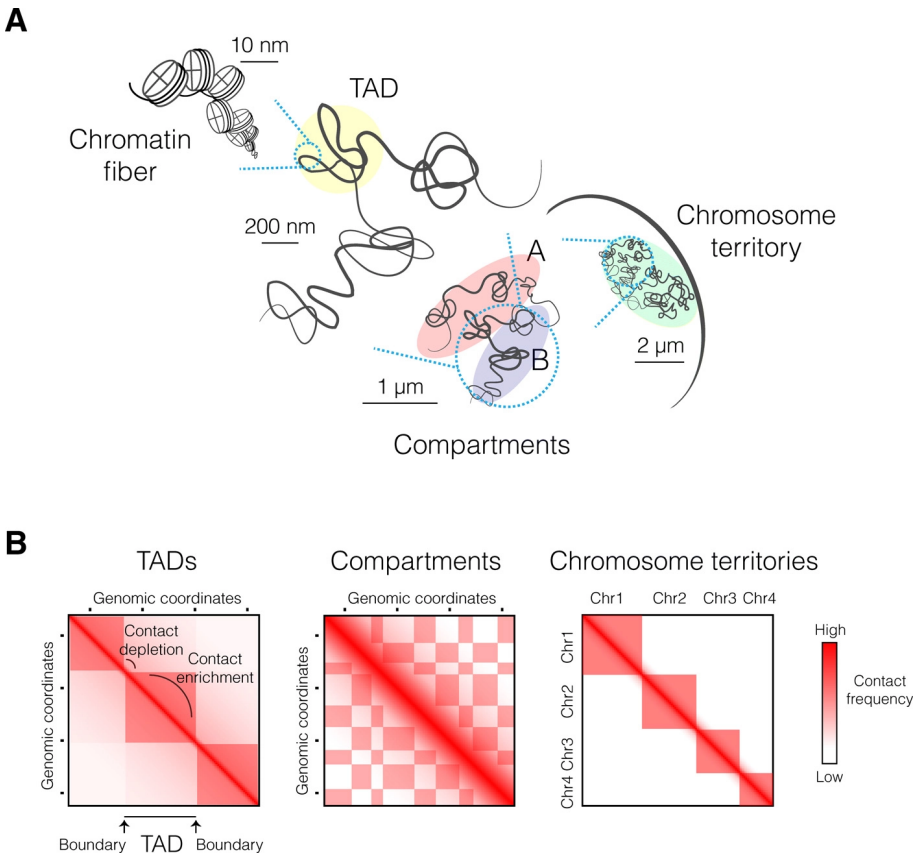


Figure 5.1. – A : Illustration du repliement des chromosomes au sein du noyau des cellules. La fibre de chromatine est compactée à différents niveaux de granularité. Elle se replie en sous-domaines enrichis en contacts et appelés *TADs*. À l'échelle chromosomique, la chromatine est séparée en deux compartiments : un compartiment actif "A" et un compartiment réprimé "B". Ces compartiments reflètent les contacts préférentiels entre les régions chromatiniennes. Les chromosomes occupent leur propre espace dans le noyau, formant les territoires chromosomiques. B : Illustration des matrices de contacts à différentes échelles génomiques (ici les matrices ont été obtenues à partir de données *Hi-C*). Les coordonnées génomiques sont indiquées sur les deux axes et la fréquence de contact entre les régions est représentée par un code couleur. Les *TADs* apparaissent comme des carrés enrichis en contacts le long de la diagonale, séparés par des zones de déplétion de contacts délimitées par les frontières des *TADs*. À l'échelle chromosomique, les interactions à longue portée de la chromatine forment un motif à carreaux caractéristique de deux compartiments A et B mutuellement exclus. Enfin, les interactions intrachromosomiques sont sur-représentées par rapport aux contacts interchromosomiques, ce qui est cohérent avec la formation de territoires chromosomiques. Extrait de Szabo et al. *Principles of genome folding into topologically associating domains* (2019) [192]. Reproduit en accord avec les termes de licence d'utilisation CC BY-NC 4.0.

La détection de contacts au sein du noyau permet de définir des cartes de l'organisation tridimensionnelle du génome à l'échelle du tissu ou même de la cellule unique. Nous pouvons distinguer trois grandes catégories de méthodes de détection de contacts [195] :

- Détection de contacts basée sur l'imagerie : Il est possible de déterminer l'organisation tridimensionnelle du génome à l'aide de méthodes de microscopie optiques ou électroniques. Les méthodes les plus répandues sont les méthodes du type *DNA-FISH*, telles que le 2D-FISH [196], ou le 3D-FISH [197] développé plus récemment.

- Détection de contacts basée sur le séquençage avec ligature : Ces méthodes sont directement inspirées de la méthode *3C* [198]. La méthode *3C* consiste à déterminer les fréquences des contacts entre un locus de référence et un locus cible qui doit être défini au préalable (méthode du type *'one versus one'*). La capture des contacts est effectuée après une fixation au formaldéhyde. Suite à la fixation, la fibre de chromatine est fragmentée par des enzymes de restriction. Les fragments de chromatine obtenus sont ligaturés, afin d'obtenir ce que l'on appelle une "librairie" *3C*. Enfin, la fréquence des contacts entre les deux locus peut être quantifiée, en utilisant une *PCR* (*polymerase chain reaction*) avec les paires d'amorces appropriées. Suite au développement de la méthode *3C*, des méthodes permettant de déterminer les contacts entre un locus et le reste du génome ont été développées, par exemple, la méthode *4C* [199, 200] (méthode du type *'one versus all'*). De plus, récemment, des méthodes du type *'all-versus-all'*, comme la méthode *Hi-C* [201] et son optimisation *in situ Hi-C* [202] ont permis d'obtenir une capture des contacts à travers l'ensemble du génome. Il est bon de noter que la résolution des contacts obtenue est dépendante du type d'enzyme de restriction choisie pour découper le génome. Plus le motif sera court, plus il sera fréquent dans le génome, et donc plus les fragments de restriction de la librairie *Hi-C* seront petits et plus la cartographie des contacts sera précise. Finalement, des méthodes permettent de capturer les contacts entre éléments de régulation en particulier (méthode du type *"many versus all"*), comme par exemple la méthode *PCHI-C* (*Promoter Capture Hi-C*) [203] qui capture les contacts des promoteurs au sein du génome.

- Détection de contacts basée sur le séquençage sans ligature : Ces méthodes ont été développées récemment et elles sont du type *'all-versus-all'*. Ces méthodes permettent donc d'avoir une cartographie des contacts de l'ensemble du génome. On peut citer comme exemple, les méthodes suivantes : GAM [204], SPRITE [205], ChIA-Drop [206].

Comme nous venons de le voir, de nombreuses méthodes existent pour détecter les contacts entre les différentes régions du génome et ainsi définir son organisation

tridimensionnelle. Dans le cadre de ma thèse, je me suis intéressé uniquement à la méthode *Hi-C* et des méthodes associées, en particulier le *PCHi-C*. La méthode *Hi-C* permet d'obtenir des matrices de contacts (Fig. 5.1.B), qui sont des représentations matricielles où l'enrichissement des interactions entre régions du génome est facilement lisible. Il est bon de noter que la méthode de *Hi-C* définit la matrice de contact à partir de millions de cellules, par conséquent la matrice de contact est tissu-spécifique. En d'autres termes, une matrice de contact permet de traduire la réalité tridimensionnelle de la conformation du génome sous une forme matricielle (Fig. 5.2). Cette matrice de contacts peut servir de base à la création de réseaux. En effet, elle peut être vue comme la matrice d'adjacence d'un réseau pondéré, dans lequel les nœuds sont les fragments du génome (définis à partir des fragments de restrictions) et les arêtes sont les contacts obtenus à partir de la méthode *Hi-C*. De manière importante, contrairement aux réseaux de gènes et de protéines qui sont génériques, on rappelle que l'on est ici à l'échelle des tissus. Il faut noter également que la résolution des méthodes *Hi-C* sur un génome entier est rarement suffisante pour obtenir un réseau satisfaisant. Deux raisons majeures expliquent ce problème : premièrement, le bruit peut rendre difficile la détermination entre de véritables contacts et des artefacts expérimentaux. Deuxièmement, les réseaux obtenus sont souvent creux (peu denses), faute d'interactions fiables, ce qui les rend inexplorables, notamment par des marches aléatoires. Afin d'éviter ces difficultés, je me suis plutôt intéressé à la méthode du type *PCHi-C*. Comme son nom l'indique, la méthode *PCHi-C* permet de capturer les interactions entre les promoteurs et les différentes régions du génomes, c'est-à-dire à la fois les autres promoteurs, mais aussi d'autres régions. Ces autres régions sont appelées *other-ending*. Ces régions *other-ending* peuvent contenir également des éléments régulateurs, tels que par exemple des *enhancers* (Fig. 5.2). L'intérêt des interactions entre promoteurs et *enhancers* a été montré comme influent dans l'apparition de pathologies [207].

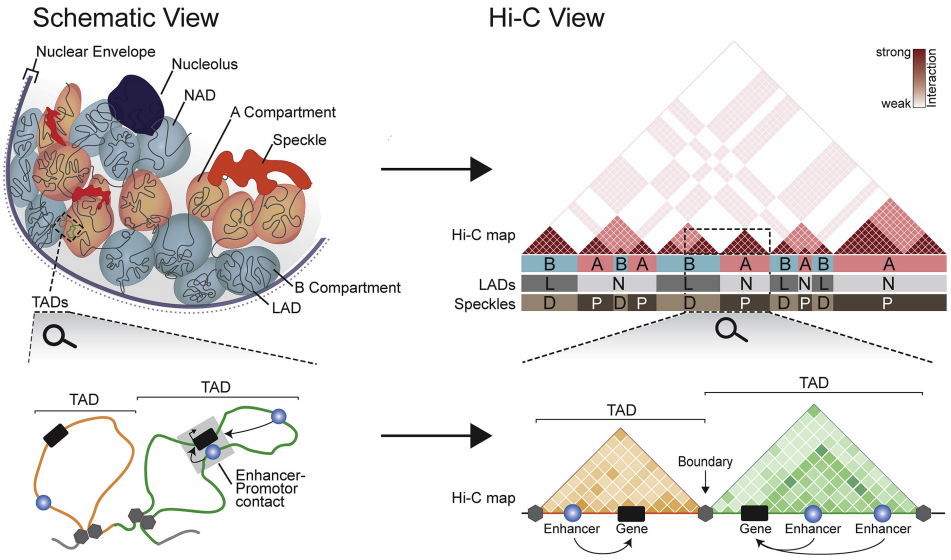


Figure 5.2. – Organisation hiérarchique de l'organisation tridimensionnelle du génome. Représentation schématique (à gauche) et *Hi-C* (à droite) de l'organisation du génome. Panneau supérieur : aux échelles d'ordre supérieur, la chromatine est séparée en compartiments d'interactions : un compartiment actif "A" (en rouge) et un compartiment réprimé "B" (en bleu). Les compartiments "B" chevauchent fréquemment les domaines associés aux nucléoles (*NADs*) et *LAD* (L) mais sont éloignés des *speckles* (D). Les compartiments "A" coïncident avec les domaines *non-LADs* (N) et sont proches des *speckles* (P). Panneau inférieur : à plus petite échelle, les *enhancers* transmettent des informations de régulations aux gènes par proximité physique au sein des *TADs*, mais pas entre *TADs*. Les *TADs* sont séparés par des frontières. Les fragments de chromatine au sein des *TADs* s'associent préférentiellement entre eux, afin de créer des blocs fonctionnels. Extrait de Robson et al. *Regulatory Landscaping: How Enhancer-Promoter Communication Is Sculpted in 3D* (2019) [208]. Reproduit avec la permission de l'éditeur Elsevier, licence numéro 5330761170746

Il existe plusieurs outils afin d'analyser les données issues de *PCHi-C*. On peut notamment citer l'outil *CHICAGO* [209] qui permet d'extraire uniquement les contacts pertinents issus des expériences de *PCHi-C*. Les données obtenues par l'outil CHICAGO sont présentées sous la forme d'une liste d'interactions entre les différents fragments de restrictions intégrant des promoteurs et des fragments de restrictions intégrant d'autres régions du génomes (*other-ending*). Ainsi, à cette étape, nous avons un jeu de données réunissant l'ensemble des interactions entre fragments de restrictions intégrant des promoteurs, que l'on peut considérer comme l'ensemble des interactions entre promoteurs. Mais nous avons également l'ensemble des interactions entre les promoteurs et les *other-ending*. Il est bon de noter qu'un poids

est associé à chaque interaction. Ce poids représente la confiance en l'existence de cette interaction. À partir de ces données, il est donc possible de créer deux types de réseaux : un réseau d'interaction de promoteurs et un réseau d'interactions entre promoteurs et *other-ending*. Néanmoins, il n'est pas aussi simple de générer des réseaux de bonne qualité avec des données de *PCHi-C*, puisque la résolution des expériences (définie par la longueur du motif de l'enzyme de restriction choisie) est un paramètre influant dans la génération de la "librairie" *PCHi-C*. Si la résolution des expériences est faible, les nœuds du réseau obtenus à partir de ces données seront très peu connectés, ce qui empêchera l'utilisation pertinente d'approches "réseaux". Cette situation est similaire pour les réseaux générés à partir des données issues de la méthode *Hi-C*. Ainsi, le choix d'un jeu de données offrant des expériences de *PCHi-C*, ayant une résolution suffisamment grande pour la construction de réseaux, est le paramètre limitant. Nous verrons au chapitre 7 comment nous avons dépassé cette limitation et construit des réseaux issus de *PCHi-C*. Nous avons aussi construit des réseaux à l'aide de données de *TADs*. Nous détaillerons aussi dans le chapitre 7 les réseaux bipartites entre fragments de restrictions et *TADs*, entre gènes et fragments de restrictions, ainsi qu'entre gènes et *TADs*. L'ensemble de ces nouveaux réseaux nous ont permis de construire de nouveaux réseaux multi-couches universels, ayant à la fois des informations sur les régions codantes du génome (avec le réseau multiplex de gènes) et des informations sur les régions non codantes du génome (avec les réseaux génomiques).

6. Article 1 : Exploration de réseaux multi-couches universels à l'aide de marche aléatoire avec *restart*

Sommaire

6.1. Introduction

La quantité de données disponibles, ainsi que leur variété et leur hétérogénéité, augmente depuis des décennies. Cette disponibilité des données à grande échelle représente une opportunité sans précédent pour mieux comprendre les systèmes complexes. Comme nous l'avons vu, les réseaux multi-couches apparaissent dans ce contexte comme un outil particulièrement efficace pour la représentation et l'analyse des données complexes, en particulier dans le cas des données biologiques. De plus, l'extension récente des méthodes d'exploration de réseaux permet de tirer profit des formalismes multi-couches. Les marches aléatoires sont une stratégie très utilisée pour explorer la topologie des réseaux à grande échelle. Dans le cadre de ma thèse, je me suis intéressé au cas particulier des marches aléatoires avec *restart*. Les marches aléatoires avec *restart* permettent de mesurer une similarité entre un nœud donné et les autres nœuds du réseau. Cependant, les méthodes actuelles sont limitées par le nombre et la variété des combinaisons de réseaux qu'elles peuvent explorer. Par conséquent, le développement de nouvelles méthodes analytiques et numériques est nécessaire afin de faire face à l'augmentation de la diversité et de la complexité des réseaux multi-couches. Mon travail de thèse, et en particulier mon premier article, s'inscrivent dans ce contexte.

Dans le cadre de ma thèse, j'ai construit plusieurs réseaux multi-couches. Un réseau biologique multi-couche biologique, constitué de deux multiplexes et d'un monoplex

associés à trois réseaux bipartites permettant de connecter chacun des trois réseaux, multiplexes et monoplex, entre eux. Les réseaux multiplexes et monoplexes, ainsi que les réseaux bipartites, sont détaillés dans les sections 5.1 et 5.2. J'ai également construit un réseau multi-couche d'aéroports, constitué de trois réseaux multiplexes et de trois réseaux bipartites permettant de connecter ces trois réseaux multiplexes. Il est intéressant de noter que ces deux réseaux multi-couches possèdent des complexités différentes. Le réseau biologique multi-couche est constitué de plusieurs dizaines de milliers de nœuds et de millions d'arêtes. Le réseau multi-couche d'aéroports est constitué de quelques dizaines de nœuds et de quelques centaines d'arêtes. Enfin, j'ai également construit un réseau multi-couche "modèle" afin de servir d'illustration au formalisme mathématique. Dans ce modèle, nous avons trois réseaux multiplexes, chacun constitué de deux couches et connecté par trois réseaux bipartites.

L'étape de la construction des réseaux n'est que la première étape, lorsque l'on considère le développement d'approches "réseaux". Il est nécessaire, dans un second temps, de développer les outils permettant l'exploration de ces réseaux, afin d'en extraire des connaissances. Le premier article réalisé dans le cadre de ma thèse, intitulé *Universal Multilayer Exploration by Random Walk with Restart*, s'inscrit dans cette volonté de développer une méthode d'exploration de réseaux multi-couches universels. Ce travail fait suite à des travaux antérieurs qui étendaient les marches aléatoires avec *restart* (introduites dans la section 3.2) aux réseaux hétérogènes [137] et aux réseaux multi-couches, ceux-ci étant constitués d'un réseau multiplex et d'un réseau monoplex hétérogène connectés au premier par un réseau bipartite [138]. Comme nous l'avons vu au chapitre 3, les marches aléatoires ont la propriété de permettre l'exploration de toute la topologie des réseaux. Dans le cadre de la priorisation d'association entre gènes et maladies, cette propriété a permis aux méthodes de marche aléatoire avec *restart* d'obtenir des performances supérieures aux méthodes classiques basées sur des mesures de distances locales [134]. Mon article généralise les marches aléatoires avec *restart* (RWR pour *Random Walk with Restart*) aux réseaux multi-couches, sans condition de complexité, c'est-à-dire à des réseaux multi-couches intégrant un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites. De plus, aussi bien les réseaux multiplexes que les réseaux bipartites peuvent être pondérés ou dirigés. Nous les avons baptisés réseaux multi-couches universels. Cette généralisation, des RWR aux réseaux multi-couches universels, introduit un formalisme mathématique qui résout en particulier le problème de la normalisation de la matrice de transition. Cette normalisation est essentielle pour obtenir une matrice stochastique et certifier la convergence vers un unique état stationnaire, comme décrit dans la section 3.2. Cet état stationnaire correspond à la mesure de similarité de tous les nœuds du réseau par rapport aux nœuds de référence, nommés graines (*seeds*). Ce nouveau formalisme mathématique est accompagné d'un package Python, nommé MultiXrank (Fig. 5.1) installable depuis *GitHub* (https://github.com/anthbapt/multixrank) et depuis pip https://pypi.org/project/multixrank. Ce package est assorti d'une documentation

accessible : https://multixrank-doc.readthedocs.io.

Ce travail est le premier à notre connaissance permettant d'obtenir une mesure de similarité entre les nœuds d'un réseau multi-couche universel. Par conséquent, nous n'avons pas pu comparer notre méthode à d'autres types de méthodes ou algorithmes. Cependant, nous avons mené une étude de validation croisée, afin de déterminer si l'ajout de réseaux multiplexes au système multi-couche, ou de couches de réseaux à l'intérieur d'un réseau multiplex, améliorait diverses prédictions (d'association entre nœuds ou de prédiction de liens). Ces tests ont été menés sur les deux réseaux multi-couches universels présentés plus haut : le réseau biologique multi-couche universel et le réseau multi-couche universel d'aéroports. Cette étude montre que la quantité n'est pas un gage de qualité. En d'autres termes, ajouter des données d'interaction ne permet pas systématiquement d'améliorer les performances des prédictions. Nous avons montré en particulier l'importance de la qualité des réseaux bipartites pour avoir un réseau multi-couche où l'information circule et, qu'ainsi, les marches aléatoires explorent correctement le réseau.
Enfin, j'ai également prêté une attention particulière à l'exploration de l'espace des paramètres, très nombreux dans le cas de notre algorithme de marche aléatoire avec *restart*. Cette exploration montre de nouveau l'importance d'avoir des réseaux multi-couches où les réseaux hétérogènes sont bien connectés entre eux par les réseaux bipartites.

De plus, nous avons introduit de nouvelles procédures permettant d'analyser l'influence de ces paramètres sur les sorties de l'algorithme. De manière générale, l'exploration de l'espace des paramètres et les validations croisées offrent de nouvelles pistes d'analyses du rapport signal sur bruit des réseaux à grande échelle.

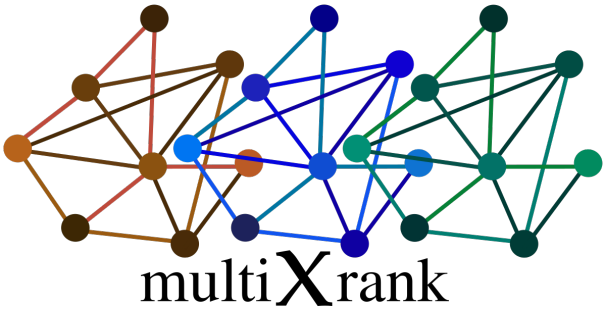


Figure 6.1. – Logo du package Python MultiXrank.

6.2. *Universal Multilayer Exploration by Random Walk with Restart*

Le présent article a été publié dans *Communications Physics*, le 01/07/2022 [210]. Le matériel supplémentaire de l'article est disponible dans l'annexe B, en fin de manuscrit.

communications physics

ARTICLE

https://doi.org/10.1038/s42005-022-00937-9 OPEN

Universal multilayer network exploration by random walk with restart

Anthony Baptista [1,2]✉, Aitor Gonzalez [2] & Anaïs Baudot [1,3]✉

The amount and variety of data have been increasing drastically for several years. These data are often represented as networks and explored with approaches arising from network theory. Recent years have witnessed the extension of network exploration approaches to capitalize on more complex and richer network frameworks. Random walks, for instance, have been extended to explore multilayer networks. However, current random walk approaches are limited in the combination and heterogeneity of networks they can handle. New analytical and numerical random walk methods are needed to cope with the increasing diversity and complexity of multilayer networks. We propose here MultiXrank, a method and associated Python package that enables Random Walk with Restart on any kind of multilayer network. We evaluate MultiXrank with leave-one-out cross-validation and link prediction, and measure the impact of the addition or removal of network data on prediction performances. Finally, we measure the sensitivity of MultiXrank to input parameters by in-depth exploration of the parameter space.

[1] Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, CNRS, Marseille, France. [2] Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France. [3] Barcelona Supercomputing Center, Barcelona, Spain. ✉email: anthony.baptista@univ-amu.fr; anais.baudot@univ-amu.fr

Data amount and variety have soared as never seen before, offering a unique opportunity to better understand complex systems. Among the different modes of representation of data, networks appear as particularly successful. Networks are indeed interesting to refine raw data and extract relevant features, patterns, and classes. They are exploited for years to study complex systems, and a wide and powerful range of tools from graph theory are available for their exploration.

However, the integrated exploration of large multidimensional datasets remains a major challenge in many scientific fields. For instance, a comprehensive understanding of biological systems would require the integrated analysis of dozens of different datasets produced at different molecular, cellular or tissular scales. Recently, multilayer networks emerged as essential players in the analysis of such complex systems. Multilayer networks allow integrating more than one network in a unified formalism, in which the different networks are considered as layers[1]. For instance, Duran-Frigola et al.[2] combined 25 different networks of chemical compounds and their relationships, gathering relationships from chemical structures to clinical outcomes. This multilayer framework allows an integrated study of chemical compounds and their biological activities. Another example is given by the Hetionet project. The authors collected dozen of heterogeneous networks, i.e networks with various types of nodes such as genes, drugs or diseases, to prioritize drugs for repurposing[3].

Several definitions of multilayer networks have been proposed, based on the (in)homogeneity of the layers and the properties of the connections between layers[4–6]. For instance, multiplex networks are multilayer networks composed of different layers containing the same nodes (called replica nodes) but different types of edges, and thereby different topologies. Heterogeneous networks link networks composed of different types of nodes thanks to bipartite interactions. Temporal networks follow the dynamic of a network over time: all the layers have the same nodes, but each layer represents the interaction state at a given time[7]. We will here consider universal multilayer networks, which can be defined as multilayer networks composed of any number of multiplex (or monoplex) networks (with edges that can be directed and/or weighted), linked by bipartite networks (with edges that can be directed and/or weighted) (Fig. 1). A wide range of methods have been developed in the recent years to analyze multilayer networks. For instance, different network metrics have been adapted to multilayer networks[8], as well as various network clustering algorithms for community detection[9–11] or random walk for network exploration[12–15].

Random walks are iterative stochastic processes widely used to explore network topologies. They can be described as simulated particles that walk iteratively from one node to one of its neighbors with some probability[16]. The PageRank algorithm, for instance, is based on a random walk simulating the behavior of an internet user walking from one page to another thanks to hyperlinks. The user can also restart the walk on any arbitrary page[17]. In this particular random walk strategy, the restart prevents the random walker from being trapped in dead-ends[18]. An interesting alternative strategy restricts the restart to specific node(s), called the seed(s)[19]. In this strategy, named Random Walk with Restart (RWR) or Personalized PageRank, the random walk represents a measure of proximity from all the nodes in the network to the seed(s). RWR can also be described as a diffusion process, in which the objective is to determine the steady-state of an initial probability distribution[20].

RWR are widely used to exploit large-scale networks. In computational biology, for instance, RWR strategies have been shown to significantly outperform methods based on local distance measures for the prioritization of gene-disease associations[21]. Importantly, different upgrades of the RWR approach have been implemented during the last decade, including its extension to (i) heterogeneous networks[12], (ii) multiplex networks[13] and (iii) multiplex-heterogeneous networks[15]. In RWR, the degrees of freedom are summarized in the Transition rate matrix, and correspond to the available transitions between the different nodes of the graph. The extensions of RWR are challenging because the Transition rate matrices need to be normalized. To the best of our knowledge, this normalization is currently only solved for multilayer networks composed of two heterogeneous multiplex networks[15,22] and the more universal case of N multiplex networks remains unsolved.

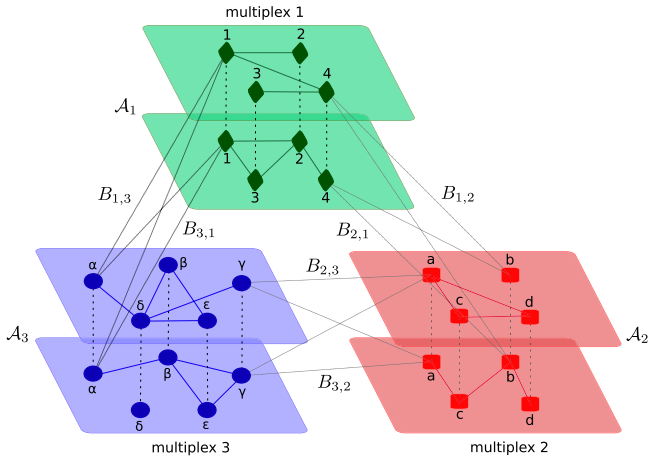


Fig. 1 A universal multilayer network. A universal multilayer network composed of three multiplex networks (green, blue and red multiplex networks). Each multiplex network contains different types of nodes (denoted 1 to 4, α to ϵ, and a to d, respectively). Their corresponding Supra-adjacency matrices are denoted by $\mathcal{A}_i$. The three multiplex networks are linked by six bipartite networks (represented here as bipartite interactions for the sake of visualization). The corresponding Bipartite network matrices are denoted by $B_{i,j}$. It is to note that a connection between a node i in a first multiplex network and α and a node j in a second multiplex network β imposes the creation of edges between all replicas of node i present in the different layers of the multiplex network α and all replicas of node j present in the different layers of multiplex network β. All the edges of the universal multilayer networks can be weighted and/or directed.

We propose here MultiXrank, a framework composed of a method and a Python package to execute RWR on universal multilayer networks. We first introduce the mathematical bases of this RWR for universal multilayer networks, which correspond to a generalization of the approach from[12]. We evaluate MultiXrank with leave-one-out cross-validation and link prediction protocols. These evaluations reveal that more network data is not always better and highlight the critical influence of the bipartite networks. We finally present an in-depth exploration of the parameter space to measure the stability of the RWR output scores under variations of the input parameters. The MultiXrank Python package is freely available at https://github.com/anthbapt/multixrank, with an optimized implementation allowing its application to large multilayer networks.

Results

Random walk with restart (RWR). Let us consider an irreducible and aperiodic Markov chain, for instance a network composed of a giant component with undirected edges, $G = (V, E)$, where V is the set of vertices and $E \subseteq (V \times V)$ is the set of edges. In the case of irreducible and aperiodic Markov chains, a stationary probability

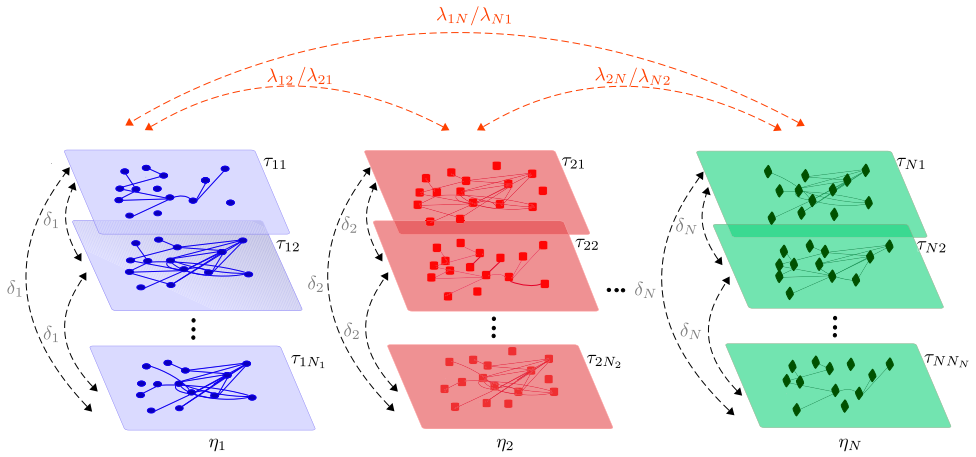


Fig. 2 MultiXrank Random Walk with Restart parameters. Parameters of the Random Walk with Restart allowing to explore universal multilayer networks composed of N multiplex networks (each composed of several layers containing the same set of (replica) nodes but different edges). The parameters δ are associated with the probability to jump from one layer to another in a given multiplex network, λ with the probability to jump from one multiplex network to another multiplex network, τ with the probability to restart in a given layer of a given multiplex network, and η with the probability to restart in a given multiplex network.

$\mathbf{p}^*$ exists and satisfies the following properties:

$$\begin{cases} \mathbf{p}^*(i) > 0; \forall i \in V \\ \sum_{i \in V} \mathbf{p}^*(i) = 1 \end{cases} \quad (1)$$

We next introduce the probability defining the walk from one node to another. Let us define x, a particle that explores the network, x_t its position at time t and x_{t+1} its position at time $t+1$. Considering two nodes i and j:

$$\mathbb{P}(x_{t+1} = j \,|\, x_t = i) = \begin{cases} \frac{1}{d_i} & \text{if } (i,j) \in E \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

with d_i being the degree of the node i. All the normalized possible transitions can be included in the Transition rate matrix. This Transition rate matrix, noted M, can be seen as the matrix of the degrees of freedom of the particle in the system. It is useful to note that the Transition rate matrix is equal to the column-normalized Adjacency matrix. The distribution denoted by $\mathbf{p}_t = (\mathbf{p}_t(i))_{i \in V}$ describes the probability of being in the node i at time t, and the stationary distribution $\mathbf{p}^*$ is obtained thanks to the homogeneous linear difference equation [3][18,23]:

$$\mathbf{p}_{t+1}^T = M\mathbf{p}_t^T \quad (3)$$

with $\mathbf{p}_t^T$ denoting the transpose of the vector $\mathbf{p}_t$. Moreover, we can introduce a non-homogeneous linear difference equation [4][23] to take into account the restart on the seed(s). When the Transition rate matrix is a Stochastic matrix, the stationary distribution is reached[18] (Supplementary Note 1.A.1 for elements of proof of convergence) and this distribution can be seen as a measure of proximity of all the network nodes with respect to the seed(s).

$$\mathbf{p}_{t+1}^T = (1-r)M\mathbf{p}_t^T + r\mathbf{p}_0^T \quad (4)$$

The distribution $\mathbf{p}_0$ corresponds to the initial probability distribution, where only the seed(s) have non-zero values; r represents the restart probability.

RWR on multiplex networks. The RWR method has been extended to multiplex networks, i.e., multilayer networks with a one-to-one mapping between the (replica) nodes of the different layers (Fig. 1)[1,13,14]. Multiplex networks can be represented by Supra-adjacency matrices, which correspond to a generalization of the standard Adjacency matrix. In the following, we will use several multiplex networks, indexed by k. We denoted by $\mathcal{A}_k$ the Supra-adjacency matrix of the multiplex network indexed by k. The Adjacency matrix of the layer l of the multiplex network k is denoted by $A_k^{[l]}$. The element of this adjacency matrix from node i to node j is defined as $(A_k^{[l]})_{i,j} \geq 0$. The dimension of the Supra-adjacency matrix $\mathcal{A}_k$ of the multiplex network k is equal to $(L_k{}^*n_k)^*(L_k{}^*n_k)$, with n_k the number of nodes in each layer of the multiplex network k and L_k the number of layers in the multiplex network k. The Supra-adjacency matrix $\mathcal{A}_k$ is defined as follows:

$$(\mathcal{A}_k)_{i_l,j_m} = \begin{cases} \left(A_k^{[l]}\right)_{i,j} & \text{if } l = m \\ \delta_{i,j} & \text{if } l \neq m \end{cases} \quad (5)$$

where δ defines the Kronecker delta (i.e., 1 if i equal j and 0 otherwise), and l and m represent the layers of the multiplex network k. We can also define a multiplex network as a set of nodes, $V_{\mathcal{A}_k}$ and a set of edges, $E_{\mathcal{A}_k}$:

$$\begin{cases} G_{\mathcal{A}_k} = (V_{\mathcal{A}_k}, E_{\mathcal{A}_k}) \\ V_{\mathcal{A}_k} = \{v_i^l, i = 1, \ldots, n_k, l = 1, \ldots, L_k\} \\ E_{\mathcal{A}_k} = \{e_{i,j}^{ll}, i,j = 1, \ldots, n_k, l = 1, \ldots, L_k, (A_k^{[l]})_{i,j} \neq 0\} \\ \cup \{e_{i,i}^{lm}, i = 1, \ldots, n_k, l \neq m\} \end{cases} \quad (6)$$

Importantly, we need to column-normalize the Supra-adjacency matrix defined in the equations [5–6] in order to converge to the steady-state, as defined in[15]. This normalization requires including the parameters δ_k related to the jumps from one layer to another inside the matrix representation, as described in[13] (Fig. 2). In the next section, we need to index by k all the parameters that are dedicated to the multiplex network k. The Supra-adjacency matrix representing the multiplex network k can be written as described in equation [7]. The matrix I_k represents the Identity matrix of size n_k.

$$\mathcal{A}_k = \begin{bmatrix} (1-\delta_k)A_k^{[1]} & \frac{\delta_k}{(L_k-1)}I_k & \cdots & \frac{\delta_k}{(L_k-1)}I_k \\ \frac{\delta_k}{(L_k-1)}I_k & (1-\delta_k)A_k^{[2]} & \cdots & \frac{\delta_k}{(L_k-1)}I_k \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta_k}{(L_k-1)}I_k & \frac{\delta_k}{(L_k-1)}I_k & \cdots & (1-\delta_k)A_k^{[L_k]} \end{bmatrix} \quad (7)$$

RWR on universal multilayer networks. We here define a RWR method that can be applied to universal multilayer networks. Universal multilayer networks are composed of any combination of multiplex networks, linked by any combination of bipartite networks (Fig. 1). All network edges can also be weighted and/or directed. The formalism for the application of RWR on multiplex networks is described in the previous section. We will now detail the Bipartite network matrices, and how to combine intra- and inter- multiplex networks information to obtain the Supra-heterogeneous adjacency matrix. The Supra-heterogeneous adjacency matrix will embed all the possible transitions in a universal multilayer network.

Bipartite networks connect heterogeneous nodes. The Bipartite network matrices contain the transitions between different types of nodes present in different networks. If the network α has n_α nodes, and the network β has n_β nodes, the Bipartite network matrix denoted $b_{\alpha,\beta}$ has a size equal to $n_\alpha{}^*n_\beta$. Now, let us define $\mathcal{A}_\alpha$ and $\mathcal{A}_\beta$, two Supra-adjacency matrices representing the multiplex networks α and β. The Bipartite network matrix $B_{\alpha,\beta}$ represents the transitions from the nodes of the multiplex network α to the nodes of the multiplex network β. The size of the Bipartite network matrix $B_{\alpha,\beta}$ is equal to $(L_\alpha{}^*n_\alpha)^*(L_\beta{}^*n_\beta)$. The Bipartite network matrices are composed of $(L_\alpha{}^*L_\beta)$ times the Bipartite network matrix $b_{\alpha,\beta}$ (equation [8]). The matrix $b_{\alpha,\beta}$ is composed of all the transitions from one layer of the multiplex network α to one layer of the multiplex network β. We extended the formalism used in[15] in order to consider more than two different multiplex networks.

$$B_{\alpha,\beta} = \underbrace{\left.\begin{bmatrix} b_{\alpha,\beta} & b_{\alpha,\beta} & \cdots & b_{\alpha,\beta} \\ b_{\alpha,\beta} & b_{\alpha,\beta} & \cdots & b_{\alpha,\beta} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\alpha,\beta} & b_{\alpha,\beta} & \cdots & b_{\alpha,\beta} \end{bmatrix}\right\}}_{L_\beta \text{ times}} L_\alpha \text{ times} \tag{8}$$

The representation of the bipartite networks as a set of nodes $V_\mathcal{B}$ and a set of edges $E_\mathcal{B}$ can be written as:

$$\begin{cases} G_\mathcal{B} = (V_\mathcal{B}, E_\mathcal{B}) \\ V_\mathcal{B} = \{v_k^\alpha, k = 1, \ldots, n_\alpha\} \cup \{v_l^\beta, l = 1, \ldots, n_\beta\} \\ E_\mathcal{B} = \{e_{k,l}^{\alpha\beta} k = 1, \ldots, n_\alpha, l = 1, \ldots, n_\beta; (b_{\alpha,\beta})_{k,l} \neq 0\} \end{cases} \tag{9}$$

It is to note that if the bipartite networks are undirected, $b_{\beta,\alpha}^T = b_{\alpha,\beta}$ and $B_{\beta,\alpha}^T = B_{\alpha,\beta}$.

Universal multilayer networks unify the representation of heterogeneous multiplex networks. We previously defined the Supra-adjacency matrices of each multiplex network and the Bipartite network matrices connecting the different multiplex networks. We now introduce the Supra-heterogeneous adjacency matrix, denoted by $\mathcal{S}$. This matrix, defined in equation [10], collects the N Supra-adjacency matrices representing each multiplex network, $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_N$, and the $N^*(N-1)$ Bipartite network matrices connecting each multiplex network, $B_{1,2}$, $B_{1,3}$, …, $B_{1,N}$, $B_{2,1}$, …, $B_{N,N-1}$.

$$\mathcal{S} = \begin{bmatrix} \mathcal{A}_1 & B_{1,2} & \cdots & B_{1,N} \\ B_{2,1} & \mathcal{A}_2 & \cdots & B_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N,1} & B_{N,2} & \cdots & \mathcal{A}_N \end{bmatrix} \tag{10}$$

We can also define the Supra-heterogeneous adjacency matrix as a set of nodes and edges:

$$\begin{cases} G_\mathcal{S} = (V_\mathcal{S}, E_\mathcal{S}) \\ V_\mathcal{S} = \bigcup_{k=1}^{N} \{v_{k,i}^{\alpha_k}, i = 1, \ldots, n_k, \alpha_k = 1, \ldots, L_k\} \\ E_\mathcal{S} = \bigcup_{k=1}^{N} \left(\{e_{i,j}^{\alpha_k,\alpha_k}, i,j = 1, \ldots, n_k, \left(A_k^{[\alpha_k]}\right)_{i,j} \neq 0\} \right. \\ \qquad \left. \cup \{e_{i,i}^{\alpha_k,\beta_k}, i = 1, \ldots, n_k, \alpha_k \neq \beta_k, \alpha_k, \beta_k = 1, \ldots, L_k\} \right) \\ \qquad \cup \bigcup_{k,l=1;k\neq l}^{N} \{e_{i,j}^{\alpha_k,\alpha_l}, i = 1, \ldots, n_k, j = 1, \ldots, n_l, \left(B_{k,l}\right)_{i,j} \neq 0\} \end{cases} \tag{11}$$

The normalization of the Supra-heterogeneous adjacency matrix ensures the convergence of the RWR to the steady-state. The most complex issue is the normalization of the Supra-heterogeneous adjacency matrix into a Transition rate matrix that can be used in equation [4]. The normalization allows obtaining a Stochastic matrix that guarantees the convergence of the RWR to the steady-state[18] (see elements of proof in Supplementary Note 1.A.1). It is important to note that we have chosen a column normalization. The resulting normalized matrix, denoted by $\widehat{\mathcal{S}}$ is defined in equation [12]. We generalized the formalism of Li and Patra[12] established for two heterogeneous monoplex networks (Supplementary Note 1.D). This generalization to universal multilayer networks is done thanks to the intra- and inter- multiplex network normalizations defined in equations [13–14], with $\alpha \in [[1, N]]$, $\beta \in [[1, N]]$. In addition, c_{i_α} is the number of bipartite networks in which the node i_α appears as source of the multiplex network α denoted by M_α.

$$\widehat{\mathcal{S}} = \begin{bmatrix} \widehat{S}_{11} & \widehat{S}_{12} & \cdots & \widehat{S}_{1N} \\ \widehat{S}_{21} & \widehat{S}_{22} & \cdots & \widehat{S}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{S}_{N2} & \widehat{S}_{N2} & \cdots & \widehat{S}_{NN} \end{bmatrix} \tag{12}$$

In equation [13], $\widehat{S}_{\alpha\alpha}$ defines the transition probabilities inside a given multiplex network. In the case of a multiplex network, if a node has no bipartite interactions with nodes from another multiplex networks, we can use the standard normalization. If bipartite interactions exist, then the normalization takes into account the probability that the walker can stay in the multiplex network $(1 - \sum_{\beta=1}^{c_{i_\alpha}} \lambda_{\alpha\beta})$. In equation [14], $\widehat{S}_{\alpha\beta}$ defines the transition probability between two different multiplex networks. There are here three possibilities. If the node has no bipartite interactions, the transition probability is equal to zero. If the node has bipartite interactions, the transition probability is equal to the standard normalization weighted by the jump probability ($\lambda_{\alpha\beta}$). Finally, if the node exists only in the bipartite network, the normalization corresponds to the standard normalization weighted by a modified jump probability. This normalization takes into account all the bipartite interactions of the considered node.

$$\widehat{S}_{\alpha\alpha}(i_\alpha, j_\alpha) = \begin{cases} \dfrac{A_\alpha(i_\alpha, j_\alpha)}{\sum_{k_\alpha=1}^{n_\alpha} A_\alpha(i_\alpha, k_\alpha)} & \text{if } \forall\beta : \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta) = 0 \\ \dfrac{\left(1 - \sum_{\beta=1}^{c_{i_\alpha}} \lambda_{\alpha\beta}\right) * A_\alpha(i_\alpha, j_\alpha)}{\sum_{k_\alpha=1}^{n_\alpha} A_\alpha(i_\alpha, k_\alpha)} & \text{Otherwise} \end{cases} \tag{13}$$

$$\widehat{S}_{\alpha\beta}(i_\alpha, j_\beta) = \begin{cases} \frac{\lambda_{\alpha\beta} B_{\alpha,\beta}(i_\alpha, j_\beta)}{\sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta)} & \text{if } \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta) \neq 0 \\ \frac{\lambda_{\alpha\beta} \sum_{i_\alpha=1}^{c} B_{\alpha,\beta}(i_\alpha, j_\beta)}{\sum_{\beta=1}^{c} \lambda_{\alpha\beta} \sum_{i_\alpha=1}^{c} \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta)} & \text{if } i_\alpha \text{ not in } M_\alpha \\ 0 \quad \text{Otherwise} \end{cases} \tag{14}$$

The normalization allows including the parameters $\lambda_{\alpha\beta}$ to jump between the multiplex networks (Fig. 2). In other words, these parameters weight the jumps from one multiplex network α to another multiplex network β, if the bipartite interaction exists. Moreover, the standard probability condition of normalization imposes that $\sum_{\alpha=1}^{N} \lambda_{\alpha\beta} = 1, \forall\, \beta$, where N represents the number of multiplex networks. Finally, the RWR equation on universal multilayer networks is defined as:

$$\mathbf{p}_{t+1}^T = (1 - r)\widehat{S}\mathbf{p}_t^T + r\mathbf{p}_0^T. \tag{15}$$

RWR initial probability distribution in universal multilayer networks. The initial probability distribution $\mathbf{p}_0$ from equation [15], which contains the probabilities to restart on the seed(s), can be written in its general form as follows:

$$\mathbf{p}_0^T = \begin{bmatrix} \eta_1 \bar{\mathbf{v}}_0^1 \\ \eta_2 \bar{\mathbf{v}}_0^2 \\ \cdots \\ \eta_N \bar{\mathbf{v}}_0^N \end{bmatrix} \tag{16}$$

where η_k is the probability to restart in one of the layers of the multiplex network k, and $\bar{\mathbf{v}}_0^k$ is the initial probability distribution of the multiplex network k. The size of $\bar{\mathbf{v}}_0^k$ is equal to $(L_k{}^* n_k)$, where L_k is the number of layers in the multiplex network k and n_k is the number of nodes in the multiplex network k. We constraint the parameter η with the standard condition of normalization of the probability that imposes $\sum_{k=1}^{N} \eta_k = 1$. We defined another parameter, τ, to take into account the probability of restarting in the different layers of a given multiplex network. This parameter includes τ_{kj}, where k corresponds to the index of the multiplex network, and j to the index of the layer of the multiplex network k (Fig. 2). In other words, τ_{kj} corresponds to the probability to restart in the j^{th} layer of the multiplex network k. Finally, $\bar{\mathbf{v}}_0^k$ is defined as follows: $\bar{\mathbf{v}}_0^k = [\tau_{k1}\mathbf{v}_0^k, \tau_{k2}\mathbf{v}_0^k, \ldots, \tau_{kL_k}\mathbf{v}_0^k]^T$, with $\mathbf{v}_0^k$ being a vector with $1/\omega_k$ in the position(s) of seed(s) and zeros elsewhere, and ω_k being the number of seeds in the multiplex network k. The standard condition of normalization of the probability gives the constraint: $\sum_{j=1}^{L_k} \tau_{kj} = 1, \forall\ k$.

Numerical implementation: multiXrank. Our RWR on universal multilayer networks is implemented as a Python package called MultiXrank (Supplementary Note 2). MultiXrank has an optimized implementation. Default parameters allow exploring homogeneously the multilayer network (Supplementary Note 1.B). The running time of the package depends on the number of edges of the multilayer network (complexity analyses in Supplementary Note 2.A). The package is available on GitHub https://github.com/anthbapt/multixrank, and can be installed with standard pip installation command: https://pypi.org/project/MultiXrank.

Evaluations. We evaluated the performances of MultiXrank using two different multilayer networks. The first one is a large biological multilayer network composed of two multiplex networks and one monoplex network. It contains a gene multiplex network gathering gene physical and functional relationships, a drug multiplex network containing drug clinical and chemical relationships, and a disease monoplex network representing disease phenotypic similarities. Each monoplex/multiplex network is connected to the others thanks to bipartite networks containing gene-disease, drug-gene, and drug-disease interactions (Supplementary Note 3.B). The second multilayer network is composed of three multiplex networks. It contains a French airports multiplex network, a British airports multiplex network, and a German airports multiplex network. In each multiplex network, the nodes represent the airports of each country and the edges represent the national flight connections between these airports for three different airline companies. The three multiplex networks are linked with bipartite networks corresponding to transnational flight connections (Supplementary Note 3.A).

We designed a Leave-One-Out Cross-Validation (LOOCV) protocol inspired by F.Mordelet and J.P.Vert[24] and A.Valdeolivas et al.[15]. In this protocol, we systematically leave-out some known associations and assess the reconstruction of this left-out data using the data remaining in the network (Supplementary Note 4.A and Fig. S9). In the case of the biological multilayer network,we systematically left-out known gene-disease associations. More specifically, for each disease associated with at least two genes, each gene is remove one-by-one and considered as the left-out gene. The remaining gene(s) associated with the same disease are used as seed(s). When the disease network is considered in the evaluation, the disease node is used as seed together with the gene node(s). The RWR algorithm is then applied, and all the network nodes are scored according to their proximity to the seed(s). The rank of the gene node that was left-out in the ongoing run is recorded. The perfect ranking for the left-out gene is 1; the closer the rank is to 1, the better the prediction. The gene left-out process is repeated iteratively for all the genes. Finally, the Cumulative Distribution Function (CDF) of the ranks of the left-out genes is plotted (Fig. 3). The CDF displays the ratio of left-out genes that are ranked by the RWR within the top-K ranked gene nodes. The CDFs are used to evaluate and compare the performance of the RWR applied to different combinations of biological networks: the protein-protein interactions (PPI) network alone, the gene multiplex network, the multilayer network composed of the gene multiplex and the disease monoplex networks, and the multilayer network composed of the gene and drug multiplex networks and the disease monoplex network (Fig. 3a).

We observed that considering multiple sources of network data is always better than considering the PPI alone. In addition, considering multilayer information is better than considering only the gene multiplex network. However, the increased performances in the LOOCV seem to arise only from combining the gene multiplex network with the disease monoplex network (and associated gene-disease bipartite network). Indeed, the addition of the drug multiplex network (and associated drug-gene and drug-disease bipartite networks) to the multilayer system does not increase the performances (Fig. 3a).

We repeated the same LOOCV protocol for the airports multilayer network, in which the left-out nodes are French airport nodes associated with a given British airport node. Here, the behavior is different, as adding the third multiplex network containing German airports connections (and associated French-German and British-German bipartite networks) increases the performances of the RWR to predict the associations between French and British airports (Fig. 3b).

To better understand these different behaviors, we examined in detail the amount of common nodes (called overlaps) existing between the nodes of the different bipartite networks. We

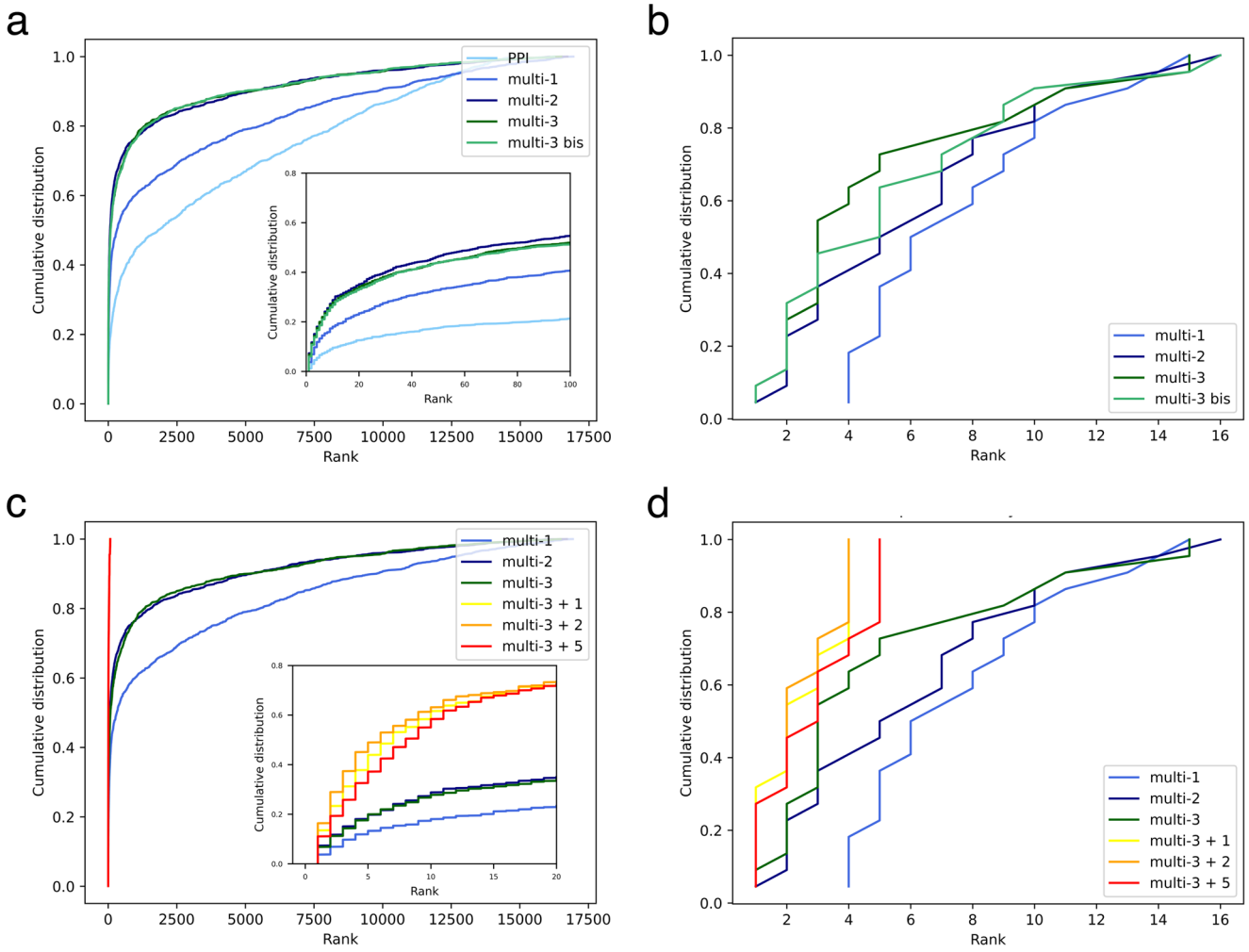


Fig. 3 Evaluation and comparison of multiXrank performances on different combinations of multilayer networks. a, **b** Cumulative Distribution Functions (CDFs) representing the ranks of the left-out nodes in the Leave-One-Out Cross-Validation (LOOCV) protocol. **a**: focus on different combinations of biological networks: protein-protein interactions network alone (PPI), gene multiplex network (multi-1), multilayer network composed of the gene multiplex network and the disease monoplex network (multi-2), and multilayer network composed of the gene and drug multiplex networks and the disease monoplex network, for two different sets of parameters (multi-3, multi-3 bis). The multilayer networks are connected by the bipartite networks described in the Evaluations section. **b**: focus on different combinations of airports networks: French multiplex network (multi-1), multilayer network composed of the French and British airports multiplex networks (multi-2), and multilayer network composed of the French, British, and German airports multiplex networks, for two different sets of parameters (multi-3, multi-3 bis). These multilayer networks are connected by the bipartite networks described in the Evaluations section. **c**, **d** CDFs representing the ranks of the left-out nodes in the LOOCV protocol for the multi-3 multilayer networks described previously with artificially increased connectivity in the gene-drug and disease-drug bipartite networks. **c** The connectivity is artificially increased thanks to the addition of 1 (multi3+1), 2 (multi3+2) or 5 (multi3+5) transit drug nodes for each gene-disease association. **d**: In the airport multilayer network, the connectivity is artificially increased in the French-German and British-German bipartite networks thanks to the addition of 1 (multi3+1), 2 (multi3+2) or 5 (multi3+5) transit German nodes for each French-British airports association. The parameters of the Random Walk with Restart (RWR) are detailed in Supplementary Tables S5–S6.

observed that only 23% of the genes from the gene-disease bipartite network are present in the drug-gene bipartite network. Similarly, only 5% of the diseases from the gene-disease bipartite network are present in the disease-drug bipartite network (Fig. S10). Given these low overlaps, the drug multiplex network might not contribute significantly to connecting gene and disease nodes during the random walks. This might explain why adding the drug multiplex network does not improve the performances of the LOOCV. Contrarily, the bipartite networks of the airport multilayer network displays high overlaps (Fig. S10). These high overlaps might explain why the addition of the third multiplex network in this case increases the predictive power (Fig. 3b).

To validate the proposed central role of bipartite networks in the RWR performances, we artificially increased the connectivity of the gene-drug and disease-drug bipartite networks before applying the same LOOCV protocol. To this goal, we added artificial transit drug nodes linking existing gene-disease associations (strategy described in Supplementary Note 4C and Fig. S12). We observed that these artificially added transit nodes increased drastically the performances of the LOOCV (Fig. 3c). The same phenomenon is observed for the airports multilayer network (Fig. 3d). In addition, we checked if random perturbations in these artificially enhanced bipartite networks would decrease the performances of the LOOCV. To do so, we progressively randomized the edges in the bipartite networks with artificially increased connectivity, until obtaining completely random bipartite networks. We observed that the progressive randomization of the bipartite networks continuously decreases the

predictive power of the RWR up to obtaining the same performances as with only two multiplex networks (Fig. S13.A for the airport multilayer networks and S13.B for the biological multilayer networks).

Finally, we repeated all these evaluations using a standard Link Prediction (LP) protocol (Supplementary Note 4.B). LP has already been used to measure the predictive power of RWR methods[25]. In the LP protocol, we systematically removed gene-disease edges from the gene-disease bipartite network, and predicted the rank of the removed gene using the disease as seed in the RWR. The LP protocol is applied on the airport multilayer network by removing a French-British edge from the French-British bipartite network, and predicting the rank of the French airport using the British airport node as a seed in the RWR. We overall observed similar behaviors as in the LOOCV (Fig. S11 and S14).

Importantly, the LOOCV and LP protocols can be used to evaluate the pertinence of adding new multiplex networks in a multilayer network or new network layers in a multiplex network. Both evaluation protocols are available within the MultiXrank package.

Parameter space exploration. We next evaluated the stability of MultiXrank output scores upon variations of the input parameters. We illustrate this exploration of the parameter space with the biological multilayer network composed of the gene multiplex network and the disease monoplex network. We first compared the top-5 and top-100 gene and disease nodes prioritized by MultiXrank using 125 different sets of parameters (see Supplementary Note 5 for the definition of the sets of parameters). We observed that the top-ranked gene nodes vary more depending on the input parameters than the top-ranked disease nodes (Fig. 4a).

To better understand the stability of the output scores upon variations of the input parameters, we proposed a protocol based on 5 successive steps: (i) definition of the sets of parameters, (ii) construction of a matrix containing the similarities of the RWR output scores obtained with each set of input parameters, using a the similarity measure defined in equation [17]. The similarities are computed for each type of node independently (i.e., for gene and disease nodes independently).

$$\Theta^k_{\gamma\sigma} = \sum_{j=1}^{n_k} \frac{\sqrt{\left(\frac{1}{\left[\left(\mathbf{r}^k_\gamma\right)_j - \left(\mathbf{r}^k_{\gamma\sigma}\right)_j\right]}\right)^2 + \left(\frac{1}{\left[\left(\mathbf{r}^k_\sigma\right)_j - \left(\mathbf{r}^k_{\sigma\gamma}\right)_j\right]}\right)^2}}{\left(\frac{\left(\mathbf{r}^k_\gamma\right)_j + \left(\mathbf{r}^k_\sigma\right)_j}{2}\right)^2} \quad (17)$$

where γ and σ define two sets of parameters, n_k is the number of nodes associated with the multiplex network k. In addition, $\mathbf{r}^k_\gamma$ (resp. $\mathbf{r}^k_\sigma$) is the rank output scores distribution that associates with each node its rank given by the RWR with the set of parameters γ (resp. σ) for the multiplex network k. Finally, $\mathbf{r}^k_{\gamma\sigma}$ (resp. $\mathbf{r}^k_{\sigma\gamma}$) gives to each node of the output scores distribution obtained by the set of parameters γ (resp. σ) (in the multiplex network k) their rank in the distribution σ (resp. γ).

We next computed a consensus Similarity matrix with a normalized euclidean norm of each individual Similarity matrix (equation [18]).

$$\Theta_{\gamma\sigma} = \sqrt{\sum_{k=1}^{N} \frac{\left(\Theta^k_{\gamma\sigma}\right)^2}{n_k}} \quad (18)$$

where N is the number of multiplex networks.

The next step is (iii) projection of the consensus Similarity matrix into a Principal Component Analysis (PCA) space (Fig. 4b). In this PCA space, each dot represents the output scores resulting from a set of parameters. Then, (iv) clustering (using k-means on the two first principal components) to identify sub-regions containing similar RWR output scores. Finally, (v) comparing the top-ranked nodes obtained with the set of parameters belonging to each cluster (Fig 4c, Supplementary Note 5).

We applied this protocol to evaluate the output scores obtained by MultiXrank on the previously defined biological multilayer network composed of the gene multiplex network and the disease monoplex network, using 125 different combinations of parameters (Fig. 4, supplementary Fig. S16). We projected the consensus Similarity matrix into a PCA space and identified 8 clusters (Fig. 4b). To illustrate the behavior inside clusters, we concentrated our analyses on the two clusters defined in the bottom left subspace (clusters number 4 and 6, zoom-in Fig. 4b). The top-100 ranked gene and disease nodes inside each of the two clusters are overall similar (Fig. 4c). This means that, even if the node prioritization can be sensitive to input parameters, we can identify regions of stability in the parameter space. Moreover, the protocol allows identifying the monoplex/multiplex networks that generate most variability in the output scores upon changes in the input parameters.

We applied the parameter space exploration protocol to other multilayer networks and observed diverse behaviors, from highly variable top-rankings and scattered projections in the PCA space for the airport multilayer network (Supplementary Fig. S15) to robust top-rankings with well-clustered projections in the PCA space for the biological multilayer network composed of 3 types of nodes (genes, diseases and drugs, Supplementary Fig. S16). Overall, our parameter space study reveals different sensitivities to input parameters depending on the multilayer network explored. The protocol is available within the MultiXrank package and can be used to characterize in-depth the sensitivity to input parameters of any multilayer network.

Discussion

Multilayer networks are nowadays very popular, in particular because they allow capturing a larger part of real and engineered systems. In biology, multilayer networks integrating multiscale sources of heterogeneous interactions provide a more comprehensive picture of biological system functionalities. However, data representation as multilayer networks must be accompanied by the development of tools allowing their exploration. Many efforts are thereby dedicated to extend classical network theory algorithms to multilayer systems[5,26]. These algorithms include for instance clustering algorithms[27], Graph Convolutional Networks[28,29] or meta-path based methods[3,30]. Other important network exploration algorithms, such as diffusion kernels or methods based on random walk, are based on the principle of network propagation[26]. The methods based on random walk, such as PageRank, biased random walk or Random Walk with Restart (RWR), are widely used in network science. They are indeed versatile: the random walk output scores can be used directly for node prioritization and subnetwork extraction, but can also be used as input for downstream analyses, for instance for supervised classification or node embedding[22].

Different random walk methods have been adapted to consider multilayer networks. However, a large variety of multilayer networks exist, from multiplex to temporal networks, for instance. To the best of our knowledge, network exploration algorithms that have been adapted to handle multilayer networks can usually be applied only to specific categories of

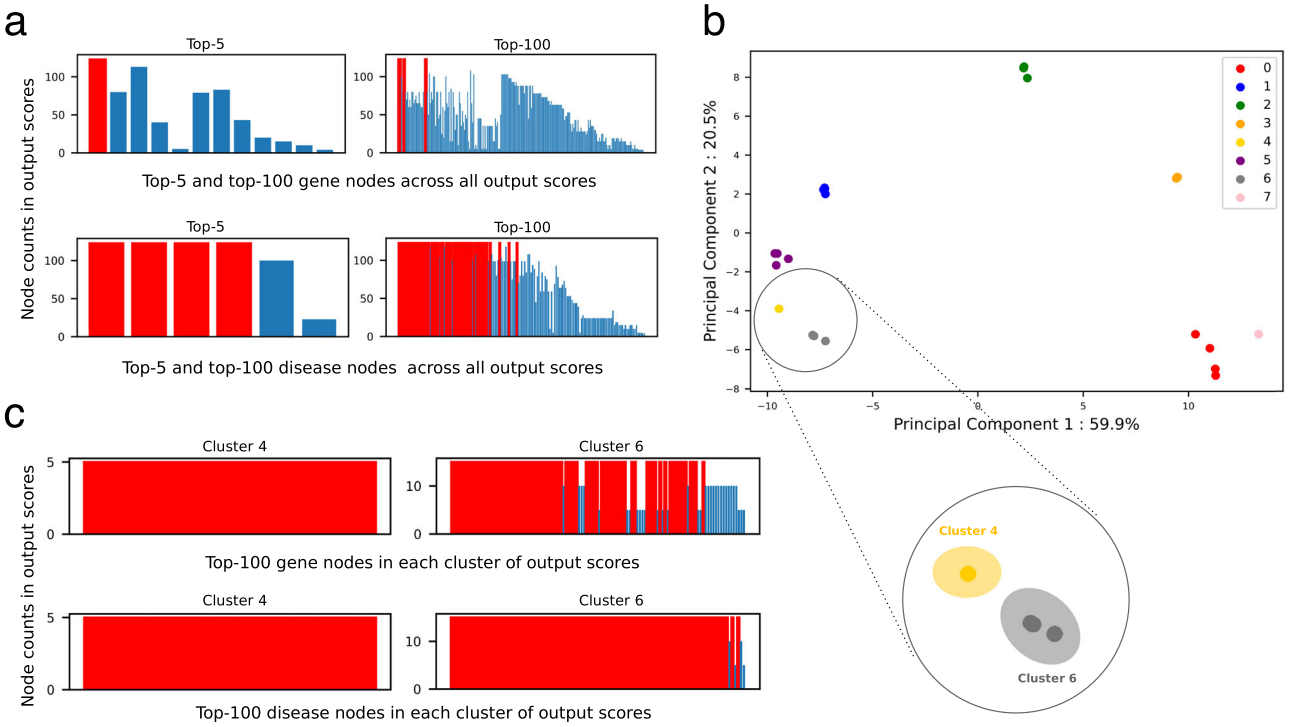


Fig. 4 Exploration of multiXrank parameter space. a Comparison of the top-5 and top-100 nodes ranked by MultiXrank using a biological multilayer networks composed of the gene multiplex network and the disease monoplex network for 125 different sets of parameters. The top-5 or top-100 ranked nodes for each set of parameters are merged, and the number of occurrences of each node are counted. The nodes are represented in bars colored in red when the node is found in all top-5 or top-100 scores, and in blue otherwise. **b** Clustering in the Principal Component Analysis (PCA) space of the output scores obtained with MultiXrank on the biological multilayer network composed of the gene multiplex network and the disease monoplex network using 125 different sets of parameters. The zoom-in emphasizes the clusters number 4 and 6. **c** Comparison of the top-100 nodes retrieved for the sets of parameters belonging to clusters 4 and 6 defined in (**b**). The bar is colored in red when a node is found in all top-100 scores, and in blue otherwise. The parameters of the Random Walk with Restart (RWR) are detailed in Supplementary Table S7.

multilayer networks, such as multiplex networks composed of the same set of nodes.

We present here MultiXrank, a tool that proposes an optimized and general formalism for RWR on universal multilayer networks. MultiXrank can be applied to explore multilayer networks composed of any combination of multiplex, monoplex or bipartite networks, and all the network edges can be directed and/or weighted. To the best of our knowledge, any type of multilayer networks could be represented with our formalism, even if it might sometimes require some adaptations. We illustrated the use of MultiXrank with RWR on biological and airport multilayer networks and thereby provide guidelines for users. Even if one's initial intuition in data analysis could be that "more data is better", the addition of interaction network layers also brings additional degrees of freedom[5]. To evaluate the pertinence of the addition of multiplex networks or the addition of layers in a multilayer system, MultiXrank includes a systematic evaluation protocol based on Leave-One-Out-Cross-Validation and Link Prediction. Overall, our results show that adding networks data does not always increase the predictive power of the RWR, as already suggested by previous studies[11]. Our evaluation protocol can be used, for the first time to our knowledge, to evaluate in-depth the signal-to-noise of multilayer system combinations. Finally, we complemented MultiXrank with a parameter space exploration protocol to measure the influence of varying the input parameters on the global stability of the output scores. It is to note that this parameter space exploration protocol is universal and can be used to study any complex system exploration approach providing scores as outputs.

The output scores of MultiXrank can be used in a wide variety of downstream analyses. For instance, shallow embedding methods need similarity measures for the optimization of the loss function[22,31]. MultiXrank can produce such a similarity measure respecting the global topology of the multilayer network. An interesting application could be to use MultiXrank output scores for embedding and evaluate the predictive power of the gene-disease association prediction task. Indeed, the embedding is expected to be more robust to the noise than the direct network space[32].

The MultiXrank package can be applied to any kind of multilayer network such as social, economic, or ecological multilayer networks. MultiXrank is optimized and can handle multilayer networks containing up to millions edges. To consider billion-scale network problems, several strategies could be considered, such as the Block Elimination Approach for RWR (BEAR) that can be exact or approximate[33] or the Best of Preprocessing and Iterative approaches (BEPI) that is an approximate approach[34].

Data availability

All the data and the code used in the article are available on an OSF repository: https://osf.io/zsmua (DOI 10.17605/OSF.IO/ZSMUA). This repository includes all the results obtained in the article.

Code availability

The package is available on GitHub https://github.com/anthbapt/multixrank, can be installed with standard pip installation command: https://pypi.org/project/MultiXrank, and is associated with complete documentation: https://multixrank-doc.readthedocs.io/en/latest.

Received: 13 September 2021; Accepted: 8 June 2022;
Published online: 01 July 2022

References

1. Bianconi, G. *Multilayer Networks: Structure and Function*. (Oxford University Press, Oxford, 2018).
2. Duran-Frigola, M. et al. Extending the small-molecule similarity principle to all levels of biology with the chemical checker. *Nat. Biotechnol.* **38**, 1087–1096 (2020).
3. Himmelstein, D. S. et al. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife* **6**, e26726 (2017).
4. De Domenico, M. et al. Mathematical formulation of multilayer networks. *Phys. Rev. X* **3**, 041022 (2013).
5. Kivelä, M. et al. Multilayer networks. *J. Complex Netw.* **2**, 203–271 (2014).
6. Lee, B., Zhang, S., Poleksic, A. & Xie, L. Heterogeneous multi-layered network model for omics data integration and analysis. *Front. Genet.* **10**, 1381 (2020).
7. Holme, P. & Saramäki, J. Temporal networks. *Phys. Rep.* **519**, 97–125 (2012).
8. Battiston, F., Nicosia, V. & Latora, V. Structural measures for multiplex networks. *Phys. Rev. E* **89**, 032804 (2014).
9. Mucha, P. J., Richardson, T., Macon, K., Porter, M. A. & Onnela, J.-P. Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328**, 876–878 (2010).
10. Didier, G., Brun, C., Baudot, A. & Gomez, S. Identifying communities from multiplex biological networks. *PeerJ* **3**, e1525 (2015).
11. Choobdar, S. et al. Assessment of network module identification across complex diseases. *Nat. Methods* **16**, 843–852 (2019).
12. Li, Y. & Patra, J. C. Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network. *Bioinformatics* **26**, 1219–1224 (2010).
13. De Domenico, M., Solé-Ribalta, A., Gómez, S. & Arenas, A. Navigability of interconnected networks under random failures. *Proc. Natl Acad. Sci.* **111**, 8351–8356 (2014).
14. Cho, H., Berger, B. & Peng, J. Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.* **3**, 540–548.e5 (2016).
15. Valdeolivas, A. et al. Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* **35**, 497–505 (2018).
16. Lovász, L. Random walks on graphs: a survey. *Combinatorics, Paul. Erdos is. Eighty* **2**, 1–46 (1993).
17. Brin, S. & Page, L. The anatomy of a large-scale hypertextual web search engine. *Computer Netw. ISDN Syst.* **30**, 107–117 (1998). Proceedings of the Seventh International World Wide Web Conference.
18. Langville, A. N. & Meyer, C. D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. (Princeton University Press, USA, 2006).
19. Pan, J.-Y., Yang, H.-J., Faloutsos, C. & Duygulu, P. Automatic multimedia cross-modal correlation discovery. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, 653–658 (Association for Computing Machinery, New York, NY, USA, 2004). https://doi.org/10.1145/1014052.1014135.
20. Gómez, S. et al. Diffusion dynamics on multiplex networks. *Phys. Rev. Lett.* **110**, 028701 (2013).
21. Köhler, S., Bauer, S., Horn, D. & Robinson, P. N. Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.* **82**, 949–958 (2008).
22. Pio-Lopez, L., Valdeolivas, A., Tichit, L., Remy, E. & Baudot, A. Multiverse: a multiplex and multiplex-heterogeneous network embedding approach. *Sci. Rep.* **11**, 8794 (2021).
23. Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. (Society for Industrial and Applied Mathematics, USA, 2000).
24. Mordelet, F. & Vert, J.-P. Prodige: prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinforma.* **12**, 389 (2011).
25. Zhou, M., Zheng, C. & Xu, R. Combining phenome-driven drug-target interaction prediction with patients' electronic health records-based clinical corroboration toward drug discovery. *Bioinformatics* **36**, i436–i444 (2020).
26. Boccaletti, S. et al. The structure and dynamics of multilayer networks. *Phys. Rep.* **544**, 1–122 (2014).
27. Huang, X., Chen, D., Ren, T. & Wang, D. A survey of community detection methods in multilayer networks. *Data Min. Knowl. Discov.* **35**, 1–45 (2021).
28. Ghorbani, M., Baghshah, M. S. & Rabiee, H. R. Mgcn: Semi-supervised classification in multi-layer graphs with graph convolutional networks. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19, 208-211 (Association for Computing Machinery, New York, NY, USA, 2019). https://doi.org/10.1145/3341161.3342942.
29. Shanthamallu, U. S., Thiagarajan, J. J., Song, H. & Spanias, A. Gramme: Semisupervised learning using multilayered graph attention models. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 3977–3988 (2020).
30. Zhang, X., Zou, Q., Rodríguez-Patón, A. & ZENG, X. Meta-path methods for prioritizing candidate disease mirnas. *IEEE/ACM Trans. Computational Biol. Bioinforma.* **16**, 283–291 (2019).
31. Hamilton, L., Ying, W., R. & Leskovec, J. Representation learning on graphs: Methods and applications (v3). https://arxiv.org/abs/1709.05584 (2018).
32. Nelson, W. et al. To embed or not: Network embedding as a paradigm in computational biology. *Front. Genet.* **10**, 381–381 (2019).
33. Shin, K., Jung, J., Lee, S. & Kang, U. Bear: Block elimination approach for random walk with restart on large graphs. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15, 1571-1585 (Association for Computing Machinery, New York, NY, USA, 2015). https://doi.org/10.1145/2723372.2723716.
34. Jung, J., Park, N., Lee, S. & Kang, U. Bepi: Fast and memory-efficient method for billion-scale random walk with restart. In Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, 789-804 (Association for Computing Machinery, New York, NY, USA, 2017). https://doi.org/10.1145/3035918.3035950.

Acknowledgements

The project leading to this preprint has received funding from the ≪ Investissements d'Avenir ≫ French Government program managed by the French National Research Agency (ANR-16-CONV-0001), from Excellence Initiative of Aix-Marseille University - A*MIDEX and from the Inserm Cross-Cutting Project GOLD.

Author contributions

A.Bap. and A.Bau. designed research; A.Bap. performed research; A.Bap. analyzed data; A.Bap. and A.G. contributed to packaged code; A.Bap. and A.Bau. wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-022-00937-9.

Correspondence and requests for materials should be addressed to Anthony Baptista or Anaïs. Baudot.

Peer review information *Communications Physics* thanks Albert Solé-Ribalta, Joao Gama Oliveira and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

6.3. Discussion

Le formalisme mathématique et l'algorithme présentés dans cet article représentent la partie centrale de mon travail de thèse. Ce travail a permis de résoudre le problème de l'exploration de réseaux multi-couches intégrant un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites. En dehors de ces aspects techniques, ce travail ouvre des portes dans l'analyse et l'interprétation des résultats en science des réseaux, notamment sur les questions du rapport signal sur bruit et de l'intégration de toujours plus de données (*"Is more always better?"*). Les questions, liées à l'interprétabilité des résultats, étaient centrales, en particulier dans le cas des données biologiques et d'autant plus lorsque l'on intègre des données multi-dimensionnelles. Ce travail, se plaçant dans le cadre des approches "réseaux", permet d'offrir une interprétabilité satisfaisante. Cette interprétabilité est la conséquence de la possibilité de visualiser les sous-réseaux correspondants aux résultats obtenus. Dans le cas des réseaux multi-couches, il existe aussi des logiciels de visualisations qui peuvent s'appliquer aux sous-réseaux [211–213]. Cependant, ces outils sont perfectibles et visualiser des réseaux multi-couches demeure un défi. En effet, représenter dans un espace à deux dimensions trois réseaux multiplexes constitués chacun de quatre couches et connectés entre eux par des réseaux bipartites, n'est pas une tâche aisée. D'autant plus lorsque l'on considère que les réseaux biologiques sont régulièrement constitués de milliers de nœuds et de plusieurs centaines de milliers d'arêtes. On peut noter, toutefois, la tentative récente d'utiliser la réalité augmentée [214] afin de visualiser et de naviguer au sein de réseaux complexes.

Le développement de MultiXrank a été motivé par la quantité et la diversité grandissantes des données d'interactions disponibles, notamment en biologie, ainsi que par la volonté de pouvoir exploiter ces données afin d'obtenir des connaissances plus complètes sur les systèmes étudiés. Partant de ce point, nous avons démontré l'efficacité de notre méthode pour l'exploration de telles données. Nous avons utilisé comme cas d'étude un réseau biologique multi-couche universel, ainsi qu'un réseau multi-couche universel constitué de nœuds d'aéroports. Ce réseau multi-couche universel d'aéroports nous a permis de nous extraire du strict contexte des réseaux biologiques et de tester notre méthode sur un réseau présentant une topologie et une taille différentes. Par ailleurs, il est particulièrement difficile de trouver des réseaux présentant une complexité comparable à celle des réseaux biologiques, c'est-à-dire des réseaux multi-couches constitués de plusieurs réseaux multiplexes hétérogènes connectés entre eux par des réseaux bipartites. Les réseaux hétérogènes, les réseaux bipartites, ainsi que les réseaux multiplexes sont couramment construits et explorés, et ce, dans de nombreux domaines : en sciences sociales, en écologie, ou en neurosciences, entre autres. Cependant, ces réseaux sont encore très souvent construits indépendamment les uns des autres, sans possibilité de les combiner dans un même système multi-couche. Il est bon de noter, que les réseaux économiques ou financiers

tendent à utiliser des données de plus en plus variées et hétérogènes [215, 216]. De nombreux efforts restent à faire en matière d'accessibilité aux données, mais le formalisme multi-couche pourra probablement prochainement être appliqué dans ce contexte.

Les réseaux biologiques offrent un cadre où une multitude de réseaux existent et où utiliser une approche universelle comme celle de MultiXrank est aisée. Cependant, la qualité des réseaux disponibles est disparate. Les réseaux biologiques historiques, comme les interactomes protéine-protéine ou de maladies, sont de bonne qualité et sont facilement accessibles. Mais d'autres réseaux sont d'une qualité moindre, comme le montre notre étude avec le cas des réseaux bipartites médicament-maladie et médicament-gène, ainsi qu'avec le cas du réseau de médicaments. Nous avions eu des difficultés à construire ces réseaux et ils sont, en effet, peu utilisés dans la littérature. De plus, beaucoup de réseaux biologiques possèdent des biais d'études, par exemple certains réseaux peuvent être tissu-spécifique ou maladie-spécifique, sans que cela ne soit spécifié. Par exemple, certains réseaux de maladies incluent presque uniquement des maladies rares, puisqu'ils sont construits avec une base de données dédiée aux maladies Mendéliennes, comme *OMIM*. Intégrer au système multi-couche des réseaux de mauvaise qualité n'améliore pas les performances de l'algorithme (comme nous pouvons l'observer avec des validations croisées et la prédiction de liens). Par conséquent, il n'est pas seulement nécessaire d'avoir accès à des réseaux, il est essentiel d'avoir accès à des réseaux qui soient les moins biaisés et les moins bruités possible, ce qui n'est pas encore le cas pour l'ensemble des réseaux pertinents en biologie. Ainsi, il est nécessaire d'avoir accès à de nouvelles données de bonne qualité, notamment issues de domaines différents de la biologie, afin de pouvoir utiliser MultiXrank sur de nouvelles données et mener des analyses complémentaires sur le comportement des réseaux multi-couches.
Nous conclurons en rappelant que MultiXrank est un outil qui ouvre la voie au développement de nouvelles méthodes basées sur les marches aléatoires, ainsi qu'à une multitude d'applications. Tout d'abord, en ce qui concerne les applications en biologie, MultiXrank peut être utile pour la priorisation de gènes et de médicaments, la prédiction d'associations entre gènes et maladies, ainsi que l'exploration de nouveaux types de réseaux biologiques tels que les réseaux génomiques. Ces différentes applications de MultiXrank seront détaillées dans le chapitre suivant et font l'objet du second article réalisé au cours de ma thèse. En plus d'être central pour les applications biologiques du second article, MultiXrank est également le point de départ de deux nouvelles méthodes que nous détaillerons au chapitre 9 : une nouvelle approche de détection de communautés pour identifier des communautés de nœuds à partir de réseaux multi-couches et une méthode d'*embedding* qui ouvre la porte de l'*embedding* aux réseaux multi-couches universels.

7. Article 2 : Applications de Multixrank sur différents cas biologiques

Sommaire

7.1. Introduction

Dans le chapitre précédent, nous avons introduit l'outil développé au cours de ma thèse, MultiXrank. MultiXrank permet d'explorer des réseaux multi-couches universels à l'aide de marches aléatoires avec *restart*. Dans le premier article, nous avons évalué l'algorithme à l'aide de méthodes de validations croisées, nous avons aussi exploré l'espace des paramètres. Cependant, nous n'avons pas détaillé les applications permises par MultiXrank, en particulier les applications biologiques.
Dans ce chapitre, nous allons détailler trois applications de MultiXrank. Ces applications sont basées sur l'utilisation des scores de marches aléatoires obtenus par l'exploration de réseaux biologiques multi-couches universels avec MultiXrank. La première application s'intéresse à la priorisation des nœuds. Nous avons ainsi utilisé MultiXrank pour prioritiser des gènes, candidats pour être impliqués dans la leucémie, et des médicaments, candidats pour soigner l'épilepsie. La seconde application s'intéresse à la prédiction d'associations entre gènes et maladies. Les scores de marches aléatoires sont ici utilisés pour entraîner un classifieur supervisé. La troisième application s'intéresse à la détection de comorbidités entre les maladies auto-immunes, en utilisant des données génomiques, issues en particulier de *Promoter Capture Hi-C (PCHi-C)* [203] et obtenues sur des cellules hématopoïétiques [217].

7.2. *Biological Applications of Random Walk with Restart on Multilayer Networks*

Le matériel supplémentaire de l'article est disponible dans l'annexe C, en fin de manuscrit.

RESEARCH

Biological Applications of Random Walk with Restart on Multilayer Networks

Anthony Baptista[1,2*] and Anaïs Baudot[1,3*]

*Correspondence:
anthony.baptista@univ-amu.fr;
anais.baudot@univ-amu.fr
[1]INSERM, MMG, Turing Center for Living Systems, Aix-Marseille Univ, Marseille, France
Full list of author information is available at the end of the article

Abstract

Background: Random Walk with Restart (RWR) is a strategy allowing the exploration of large networks such as biological networks. RWR has been successfully applied to a wide variety of tasks, including node prioritization or link prediction. However, the increasing diversity and complexity of network data, calls for the development and application of new RWR frameworks. A recent RWR algorithm, named MultiXrank, has been developed to explore universal multilayer networks, i.e. networks composed of any combinations of multiplex and heterogeneous networks connected by bipartite networks.

Results: MultiXrank output RWR scores from the exploration of multilayer networks. We propose here three biological applications using these RWR output scores. The first application is dedicated to node prioritization; we prioritize genes and drugs of interest in the context of Leukemia. We also use the scores to predict candidate drugs for epilepsy. Our second application is dedicated to the training of a supervised classifier for the prediction of gene-disease associations. Finally, our third application concerns the prediction of comorbidities between immune diseases, using genomic information extracted from Promoter Capture Hi-C (PCHi-C).

Conclusion: We show that MultiXrank-based RWR on multilayer networks offers new opportunities in computational biology, as RWR scores are versatile enough to be used in a wide variety of biological applications.

Keywords: Multilayer Network; Random Walk; Multi-omics Data; Biological Network

Introduction

Random Walk with Restart (RWR) is an algorithm dedicated to the exploration of the whole topology of networks. This algorithm is inspired by the PageRank algorithm. The PageRank algorithm uses random walks to simulate the behavior of an internet user walking from one page to another thanks to hyperlinks. The user can also restart the walk on any arbitrary page [1]. Importantly, the restart prevents the random walker from being trapped in dead-ends [2]. In the RWR strategy, the restart is restricted to one or several specific node(s), called the seed(s) [3]. RWR can also be described as a diffusion process, in which the objective is to determine the steady-state of an initial probability distribution [4]. This steady-state represents a measure of proximity from all the nodes in the network to the seed(s). RWR is widely used in computational biology to explore large-scale networks. For instance, RWR strategies have shown to significantly outperform methods based on local distance measures for the prioritization of gene-disease associations [5].

Several extensions of the RWR algorithm have been developed to improve the prediction of gene-disease associations, mostly by considering additional interaction data such as interactions built from phenotype data. For instance, Li and Patra [6] defined a RWR on heterogeneous networks. The heterogeneous network is composed of two networks, a gene network, and a phenotype network, both connected by a gene-phenotype bipartite network. Li and Li [7] defined a RWR on a multigraph containing both gene and phenotype information. A recent extension uses RWR on a multiplex-heterogeneous network (i.e., a multiplex network connected to a monoplex network with bipartite interactions) [8].

RWR algorithm is also used in a wide variety of biological applications beyond the prediction of gene-disease associations, such as for protein function prediction [9], for the identification of disease comorbidity [10], or for drug-target interaction prediction [11]. More recently, in the context of SARS-CoV-2, RWR has been used to prioritize antiviral drugs [12] and to repurpose drugs by identifying SARS-CoV-2-induced pathways [13]. RWR algorithm is a useful tool to exploit data represented as networks. We have seen that the RWR algorithms have been extended to consider more complex network frameworks, such as multiplex or heterogeneous networks. This is particularly important to better leverage the amount, variety, and heterogeneity of data, which have been increasing drastically for several years.

Recently, a RWR algorithm able to explore universal multilayer networks, named MultiXrank, has been published [14]. A universal multilayer network is defined as a multilayer network composed of any number of multiplex (or monoplex) networks, connected by bipartite networks. All the network layers can also be weighted and/or directed. MultiXrank offers the opportunity to apply RWR in the context of the previously mentioned biological applications while leveraging the richness of data present in multilayer networks. In practice, MultiXrank gives access to RWR output scores obtained from exploring the multilayer networks. These RWR output scores can then be used in several applications. We implemented here three applications. The first one is dedicated to node prioritization. We used MultiXrank output scores to prioritize genes and drugs of interest in leukemia, we also predict candidate drugs for epilepsy. The second application intends to predict gene-disease associations with a supervised classifier trained with RWR output scores. Finally, the last application is devoted to the prediction of comorbidities between immune diseases. This last application uses genomic information extracted from Promoter Capture Hi-C (PCHi-C) [15] experiments in different hematopoietic cells [16]. Some of the identified comorbidities are further confirmed by bibliographic research.

1 Node prioritization in leukemia and epilepsy

The first biological application consists in using MultiXrank for node prioritization. This is done in two different contexts, leukemia, and epilepsy.

1.1 Node prioritization in leukemia

In order to illustrate the value of MultiXrank for node prioritization, we analyzed two nodes of interest for leukemia, a well-studied disease for which we can confront our predictions with the knowledge existing in the literature. Our goal is to prioritize nodes around i) Tipifarnib (DB04960), a drug investigated for the treatment of Acute Myeloid Leukemia and other types of cancer [17–19], and ii) HRAS, as mutations in the RAS gene family have been described in a wide variety of tumors, in particular in Myeloid Leukemia [20]. The association between these two nodes is particularly relevant as HRAS is a farnesylated protein and Tipifarnib is a farnesyltransferase inhibitor [21]. We used HRAS and Tipifarnib as gene and drug seeds, respectively, in MultiXrank applied on a multilayer network composed of a gene multiplex network and a drug multiplex network (Fig. 1). The networks are detailed in supplementary section 1.A and the parameters of MultiXrank in supplementary Table S1. A literature survey of the top-10 prioritized genes and drugs (supplementary Table S2) demonstrates established or suspected connections with leukemia (supplementary section 2.A). For instance, the second highest-scoring gene is FNTB, a gene coding the farnesyltransferase and a target of Tripifarnib. Different genes related to signal transduction and known to be relevant for cancer, including RAF1, RASGRP1, RASA1, or ARAF are also identified among the top-scoring genes. The details for the top-10 genes and diseases prioritized are detailed in supplementary section 2.A.
In conclusion, all the top-ranked genes and drugs prioritized with the HRAS and Tipifarnib seeds have links with leukemia. These results illustrate the efficiency of MultiXrank in a prioritization task.

1.2 Node Prioritization to predict candidate drugs for epilepsy and comparison of MultiXrank with the Hetionet framework

We applied MultiXrank to prioritize candidate drugs for epilepsy, using as a seed the epilepsy disease node (OID:1826) in the large and heterogeneous multilayer network assembled in the Hetionet project [22]. This multilayer network (Fig. S1) is composed of nine different types of nodes distributed in multiplex and monoplex networks (see supplementary section 1.B for details). We compared the drugs top-predicted by MultiXrank (an unsupervised approach) with the drugs top-predicted by the Hetionet approach using a supervised machine learning approach based on regularized logistic regression [22]. Many drugs are predicted by both approaches. Indeed, for one of the sets of parameters tested in MultiXrank (set of parameters number 4), 59% of the top-100 Hetionet predictions are also in the top-100 of MultiXrank predictions, 80% in the top-200 of MultiXrank, and 99% in the top-500 of MultiXrank (Fig. 2). We further checked the 41 drugs that are found in the top-100 of candidate drugs for epilepsy by MultiXrank but are not Hetionet predictions. Supplementary Table S3 displays the classes of drugs that are corresponding to at least seven of the 41 drugs. We next investigated to what extent these drugs are

relevant for epilepsy. We focused on the 27 different drugs that belong to these three classes of drugs (supplementary Table S3): Cytochrome P-450 Substrates, Analgesics, and Indoles. We remind that some drugs have several classifications. For instance, among the 27 different drugs belonging to these three classes, 24 are Cytochrome P-450 Substrates, 17 are Analgesics (14 are both Cytochrome P-450 Substrates and Analgesics), 8 are Indoles (5 are both Cytochrome P-450 Substrates and Indoles), and all the Indoles drugs are also Analgesics. A recent study has shown in mice that spontaneous recurrent seizures modify Cytochrome P-450 expression in the liver and the hippocampus. The authors hypothesize that nuclear receptors or inflammatory pathways can be considered as candidate mechanisms of Cytochrome P-450 regulation during seizures [23]. A recent publication reviews preclinical evidence that tends to show that anti-epileptic drugs can be used as Analgesics in patients with inflammatory pain, and can contribute to the improvement of the treatment of various inflammatory pain states [24]. Lately, another study on rats evaluated the anti-epileptic effect of new Indole derivatives. This study demonstrates that some Indole derivatives induce a decreasing susceptibility to seizures [25].

These results indicate that MultiXrank can provide predictions complementary to the Hetionet machine learning approach. In addition, MultiXrank predictions can be easily interpreted thanks to the extraction of the subnetworks underlying the prioritization.

2 Supervised classification of gene-disease associations

In this section, we will introduce a new supervised method to predict gene-disease associations based on RWR exploration of multilayer networks. The prediction of gene-disease associations is crucial for the diagnosis, understanding, and treatment of genetic diseases. Among the approaches available for gene-disease predictions, network-based methods have been particularly employed and have demonstrated good performances [26]. These network approaches were initially mainly based on unsupervised strategies, but an increasing number of methods are currently implementing supervised strategies [26]. Here, we use the RWR output scores of MultiXrank to train a supervised binary random forest classifier to predict gene-disease associations.

We applied MultiXrank (using the parameters described in supplementary Table S4) to a biological multilayer network composed of a gene multiplex network and a disease monoplex network (supplementary section 1.A). These monoplex/multiplex networks are connected with a gene-disease bipartite network constructed with an outdated version of DisGeNET (v2.0, 2014 [27]). In brief, we trained a binary random forest classifier with different parameters using the MultiXrank output scores. We tested the performance of the classifiers to predict gene-disease associations that have been added in an updated version of the same gene-disease bipartite network (DisGeNET v7.0, 2020 [28]). A more detailed protocol is described below.

1 **Creation of the training dataset**
The training dataset was created based on the 2014 gene-disease associations of DisGeNET database (v2.0). Each association has a score reflecting the number of sources that report the association. We choose a threshold equal to 0.5, and we obtained 1980 gene-disease associations. These associations constitute our positive dataset (labels equal to 1). On the other hand, we generated a negative set of 1980 associations (labels equal to 0) defined as a subset of the following one: $\{(i,j) \in (\text{Gene node}, \text{Disease nodes})\} \backslash \{(i,j) \in \text{Disgnet v2.0}\}$. The training dataset is composed of the merging of the two previous datasets.

2 **Running MultiXrank for all associations**
Then, all the associations defined in the training dataset are systematically considered as seeds for MultiXrank. We run MultiXrank on the multilayer network composed of two multiplex networks, one composed of three layers of genes (gene multiplex network) and a second composed of a diseases similarity network (disease monoplex network). These monoplex/multiplex networks are connected by a bipartite network composed of the 1980 gene-disease association defined by DisGeNET v2.0 (see section 3.B for more details on the networks). The parameters chosen for the MultiXrank are defined in supplementary Table S4. For each positive and negative gene-disease association defined in the training dataset, we used both gene and disease nodes as seeds, and we save the scores defined by MultiXrank.

3 **Training the binary classifier (Random forest)**
The scores obtained with MultiXrank are used to train a binary classifier. The scores encode the similarity of all the nodes from the multilayer networks to the seeds of each positive and negative training gene-disease association. The training of the classifier from RWR output scores is the same as in Liu et al. [29], except that we used a random forest instead of a gradient tree boosting algorithm (Fig S2). We decided to use random forest as a good compromise between stability, over-fitting, and interpretability. The parameters of the random forest are described in supplementary Table S4.

4 **Creation of the test dataset**
We used as a test dataset the bipartite network containing the gene-disease associations defined by an updated version of DisGeNET v7.0 (2020) instead of DisGeNET v2.0 (2014). The positive associations are extracted from the DisGeNET v7.0 (threshold equal to 0.5) and the negative associations are defined as a subset of the following set: $\{(i,j) \in (\text{Gene node}, \text{Disease nodes})\} \backslash \{(i,j) \in$ Disgnet v7.0$\}$

5 **Evaluation of the binary classifier on the test dataset**
Finally, we run MultiXrank for each positive/negative gene-disease association of the test dataset described in the fourth step, containing gene-disease association from DisGeNET v7.0 (2020). We use MultiXrank output scores as input of the random forest binary classifier training on the data obtained

with the gene-disease association of DisGeNET v2.0 (2014). Finally, we compare the classification obtained with the random forest to the expected results.

The best random forest classifier according to the F1 score had a score equal to 0.90 (supplementary Table S5), which demonstrates that MultiXrank outputs can be used to train a classifier to predict new gene-disease associations.

In a recent study, it has been shown that "more is not always better" and that the quality of the bipartite networks is of high importance [14]. However, this initial study leaves an open question regarding the impact of the addition of a new multiplex network in a multilayer system. To test this, we applied the same supervised random forest approach to another biological multilayer network, this time composed of two multiplex and a monoplex networks: a gene and a drug multiplex networks, and a disease monoplex network. The goal is to determine if the predictions are better when using MultiXrank output scores from three multiplex/monoplex networks rather than using output scores from two multiplex/monoplex networks. The biological networks are described in supplementary section 1.A.
We used the best random forest classifiers (highlighted in orange in supplementary Tables S5 and S6). The results show that the second multilayer network, containing one multiplex network more than the first multilayer network, does not increase the predictive power of the classifier. It even decreases the performance (F1 score = 0.63, supplementary Table S6) compared to a score of 0.90 (supplementary Table S5) with the use of the multilayer system composed of two multiplex/monoplex networks. These results complement our previous study in which we have shown that poorly connected bipartite networks do not increase the prediction performances of the RWR [14].

3 MultiXrank exploration of genomic networks to predict comorbidities

In this section, we will present a way to unveil comorbidities between diseases by exploring multilayer networks. We created a multilayer network composed of several types of networks: a gene multiplex network, a disease monoplex network, a PCHi-C (Promoter Capture Hi-C) fragment network, and a TAD (Topologically Associating Domain) network. Details about the construction of the different networks are available in supplementary section 1.A and 1.C.
The consideration of genomic information with the PCHi-C and the TAD networks enables to complement the gene multiplex network with data that represents the 3D conformation of DNA. This 3D conformation of DNA is a key to understanding, for instance, the structural variations. The structural variations of the genome regroup deletions, duplications, inversions, and translocations, all contributing to a large extent to the genetic variability of the human genome [30]. The structural variations are key players in the study of diseases [31]. These networks allow considering the non-coding regions of the genome.

3.1 PCHi-C experiment reflects the tree lineage of hematopoietic cells

The tree lineage of hematopoietic cells is defined in Fig. S3, as well as in the original paper from which we extracted the PCHi-C dataset [16]. We extracted the PCHi-C and the TAD of eight different hematopoietic cells. A simplified vision of the tree lineage of the eight hematopoietic cells defines two major branches. The first branch contains the lymphoid cells and the second branch contains the myeloid cells. The lymphoid cells branch is further divided into two other branches: the first one contains the B-cells with the naive B cells (nB), and the second one contains the T-cells with the Naive CD4+ T cells (nCD4) and the Naive CD8+ T cells (nCD8). The myeloid cells branch is also further divided into two other branches: the first one contains the Monocytes (Mon) and the Neutrophils (Neu), and the second one contains the Megakaryocytes (MK), and the Erythroblasts (Ery). The Macrophage M0 (Mac0) are differentiated from monocytes [32].
Before controlling that the tree lineage of the hematopoietic cells is reflected by the exploration of a multilayer network with RWR, we need to check that this tree lineage can be found directly from the PCHi-C datasets and from the TAD datasets of the eight hematopoietic cells. The structure of the different datasets is defined in supplementary section 1.C. In the following text, we will refer to the PCHi-C datasets as the PCHi-C fragment omic, and to the TAD datasets as the TAD omic. For both omics, we used the Jaccard index on the different datasets to define a similarity measure between the different hematopoietic cells. Based on the Jaccard index, we created a similarity matrix defined as follows:

$$J_{i,j} = \frac{D_i \cap D_j}{D_i \cup D_j} \tag{1}$$

where D_i and D_j the datasets (PCHi-C or TAD) corresponding with the hematopoietic cells i and j. Then, we projected this similarity matrix into a 2D PCA (Principal Component Analysis) space (Fig. 3). The projection preserved correctly the tree lineage of the hematopoietic cells. We indeed observe a separation between lymphoid cells on the right (red and blue dots) and myeloid cells on the left (green, yellow, and purple dots). Second, we observe that the nCD4 and the nCD8 are very close to each other (blue dots), and the nB (red dot) are close to both of these hematopoietic cells. Third, the Mon and the Neu cells are closer to each other (yellow and purple dots) than to the other nodes. Finally, the Ery and MK are very close to each other (green dots). In conclusion, the projection shows that both PCHi-C and TAD omics reflect the tree lineage of the hematopoietic cells.

3.2 RWR exploration of a multilayer network including PCHi-C and TAD omics data also reflects the tree lineage of the hematopoietic cells

The previous section proves that the PCHi-C and the TAD genomic omics reflect the tree lineage of the hematopoietic cells. Now, we will determine if the tree lineage of the hematopoietic cells can still be unveiled through the construction of a

multilayer network containing the genomic omics, and its exploration with RWR. We first need to represent both PCHi-C and TAD omics as networks. This network representation will allow the integration of the PCHi-C and TAD omics into a multilayer network (Fig. 4). It is important to note that the PCHi-C and TAD omics are obtained for the different hematopoietic cell types independently. We hence defined a multilayer network for each cell type. Each cell-type specific multilayer network contains the same disease monoplex and gene multiplex networks but a PCHi-C fragment network and TAD network specific to the cell type. The PCHi-C fragment network represents the interaction between chromatin fragments, with at least one of the chromatin fragments including a promoter. The TAD network connects adjacent TADs. The idea is to use a linear approximation of the genome as the edges of this TAD network; the hypothesis is that it would complete the 3D conformation information from the PCHi-C fragment network. The details of the construction of the networks are defined in supplementary section 1.C. The PCHi-C fragment networks of the eight different hematopoietic cells are detailed in supplementary Table S7. We illustrated the PCHi-C fragment networks with the network produced for the nB cells (Fig. S4).

We used MultiXrank to explore the different multilayer networks. MultiXrank produces RWR output score for each type of node: the gene/protein nodes, the disease nodes, the PCHI-C fragment nodes, and the TAD nodes. We will refer to each network and the set of nodes associated as omics. Hence, we defined four omics: the gene/protein omics, the disease omics, the PCHi-C fragment omics, and the TAD omics. The RWR needs a starting seed to explore the network and to define a similarity measure between this seed and all the other nodes. We selected seeds among the 138 different immune diseases iteratively (see supplementary Table S8 for the list of immune diseases). This produces 138 different RWR output scores, one for each immune disease. Then we considered the four types of RWR output scores, one for each omics. Thus, we obtain 138 RWR output scores separated into four RWR vectors, each RWR vector being associated with one omics. Finally, the protocol is reproduced for each hematopoietic cell type, leading to $8 \times 4 \times 138$ RWR vectors.

Based on these RWR vectors, we defined a similarity measure between the different hematopoietic cell types, one measure for each omics. To do so, for each omics and each hematopoietic cell type, we computed a similarity matrix between every pair of the 138 RWR vectors with a homemade similarity measure detailed in Fig. S5. This produces 8 similarity matrices of size 138×138. These 8 similarity matrices are in addition produced for each omics. Then, based on these similarity matrices, we constructed a proximity matrix obtained with the average Pearson correlation between every pair of the 8 similarity matrices defined previously, again for each omics separately. The average Pearson correlation corresponds to the mean of the Pearson correlation between each pair of columns of the matrices. The size of the proximity matrix is equal to 8×8, and represents the proximity between all pairs of hematopoietic cell types, for each omics. Finally, we projected the proximity matrix into a 2D PCA space, to check if the tree lineage of the hematopoietic cell types is reflected using a multilayer network exploration with RWR.

Fig. 5 represents the 2D PCA projection of the proximity matrix obtained for each omics. We observe first that the gene/protein omics and the disease omics do not

reflect the tree lineage of the hematopoietic cell types. This result was expected because the gene multiplex network and the disease monoplex network are not cell-type specific. The TAD omics reflect the tree lineage of the hematopoietic cell types. We see a clear separation between the lymphoid and the myeloid cells. However, the Neu and the Mon proximity have disappeared. Finally, the PCHi-C omics reflect the tree lineage of the hematopoietic cells to the same extent as the original datasets represented in Fig. 3. In conclusion, the tree lineage of the hematopoietic cells is preserved through the RWR process on multilayer networks constructed with PCHi-C omics and the TAD omics.

3.3 RWR exploration on multilayer network to study comorbidities between immune diseases

In the previous section, we have seen that the tree lineage of the hematopoietic cell types is reflected in the RWR exploration of genomic networks built from PCHI-C and TAD omics data. The PCHi-C and the TAD omics allow gathering information concerning the non-coding region of the genome, which is complementary to the gene/protein omics gathering genetic information contained in the coding region of the genome. Both information are completed by the disease omics, which define similarity measures between diseases. All these different pieces of information are contained in the RWR output scores. So, these RWR output scores provide a good opportunity to define similarities between immune diseases. We hypothesize that the most similar diseases according to all the gathered network information could represent potential comorbidity relationships, i.e. diseases occurring more frequently together in patients than expected by chance.
To predict comorbidity relationships between immune diseases, we use a similar process as described in the previous section: we used each immune disease iteratively as a seed in a MultiXrank exploration of the multilayer networks, iteratively for each one of the 8 cell-type specific multilayer networks. MultiXrank returns the four types of RWR output scores, one for each omics, for the 138 different RWR output scores, one for each immune disease. Thus, we obtain 138 RWR output scores separate into four vectors, one vector associated with each omics. Overall, we obtain $8 \times 4 \times 138$ RWR vectors. For each omics and each hematopoietic cell type, we computed a similarity matrix between the pairs of RWR output scores vectors of the 138 immune disease, using the homemade similarity measure (detailed in Fig. S5). This produces 8 similarity matrices of size 138×138, one for each omic. For each omics, we constructed a consensual similarity matrix, noted S, based on the eight hematopoietic cell type similarity matrices, such that:

$$S = \sqrt{\sum_{i=1}^{m} S_i^2} \tag{2}$$

with S_i representing the similarity matrix for a given hematopoietic cell type, and m the number of different hematopoietic cell types (8). Then, this consensual similarity matrix S, associated with an omics, is projected into a 2D t-SNE (t-distributed

stochastic neighbor embedding) [33] space (Fig. 6 and Fig. S7-9). Fig. 6 represents the projection of the consensual similarity matrix for the PCHi-C omic, into a 2D t-SNE. Then a partition of the nodes in the 2D t-SNE space is done with the k-means method, leading to 30 clusters. We represented the same data with the alternative UMAP (Uniform Manifold Approximation and Projection) [34] non-linear dimension reduction methods in Fig. S6. We also computed the t-SNE projection of the three other omics: the gene/protein omics (Fig. S7), the disease omics (Fig. S8), and the TAD omics (Fig. S9).
In the 2D t-SNE projection represented in Fig. 6, we can focus on some diseases close in the projection and belonging to the same cluster. We hypothesize that these close diseases could represent comorbidity predictions. Some of the comorbidity predictions can be used as controls because some immune diseases are duplicated. They are indeed named by several UMLS identifiers (see supplementary Table S8). In other words, some immune diseases correspond to different nodes in the disease network, and therefore different nodes in the 2D t-SNE projection. So, for these disease nodes, we expect that there are close in the 2D t-SNE projection. However, they can have different interactions in the disease monoplex network and the gene-disease bipartite network. So, these nodes in the t-SNE space are expected to be close but not identical.

- 22q11.2 deletion syndrome (DiGeorge syndrome) (nodes 46, 47, 48): cluster 1
- Rheumatoid arthritis (nodes 28 and 35): cluster 14
- Giant cell arteritis (temporal arteritis) (nodes 7 and 33): cluster 27
- Cicatricial pemphigoid (node 3) and Ocular cicatricial pemphigoid (node 23): cluster 28

In another control, we considered nodes corresponding to diseases that are expected to be similar based on their description. We hypothesized that these nodes are also close to each other in the 2D t-SNE space. We consider the three following examples:

- Immune dysfunction with T-cell inactivation due to calcium entry defect 1 (node 90) and Immune dysfunction with T-cell inactivation due to calcium entry defect 2 (node 91): cluster 20
- Hypogammaglobulinemia AGM3 (CD79A) (node 12) and Hypogammaglobulinemia AGM6 (CD79B) (node 15): cluster 6
- Hypogammaglobulinemia AGM2 (IGLL1) (node 11), Hypogammaglobulinemia AGM4 (BLNK) (node 13), and Hypogammaglobulinemia AGM5 (LRRC8A) (node 14): cluster 19

For these sets of similar diseases, the hypothesis is validated: similar disease nodes are close in the projected space and belong to the same cluster.

We can now consider the reverse experiment. Instead of taking similar diseases and looking if there are close in the 2D t-SNE projection, we considered close nodes in the 2D t-SNE projection. We considered these close diseases as predicted comorbid diseases and checked if they have known relationships. The three following

examples represent close nodes that correspond to different diseases but that could have comorbidity relationships.

- Autoimmune lymphoproliferative syndrome due to CTLA4 haploinsuffiency (node 55), cyclic neutropenia (node 74) and Felty's syndrome (node 81): cluster 29. Neutropenia has relationships with the two other diseases. For instance, the Felty syndrome is a severe form of rheumatoid arthritis, characterized by a triad of rheumatoid arthritis, splenomegaly, and neutropenia (https://www.orpha.net). Moreover, the autoimmune lymphoproliferative syndrome can cause neutropenia (https://www.niaid.nih.gov/autoimmune-lymphoproliferative-syndrome).
- Netherton syndrome (node 109) and Papillon Lefevre syndrome (node 112): cluster 18. It has been shown that the Lympho-epithelial Kazal-type-related inhibitor (LEKTI) encoded by the gene SPINK5, a defective gene in Netherton syndrome, presents mutations in the Kazal-type 1 identified Papillon–Lefevre syndrome [35, 36].
- Burkitt lymphoma (nodes 36 and 38), Chronic lymphocytic leukemia (node 39), Chronic myelogenous leukemia (node 40), Hodgkin's lymphoma (node 41), Ataxia telangiectasia (node 53): cluster 16. It is known that patients with Ataxia telangiectasia present high risk of developing leukemia or lymphoma [37, 38] due to a weakened immune system (https://medlineplus.gov/ataxia-telangiectasia).

In conclusion, this new approach that uses both genetic and genomic information associated with disease relationships seems to be promising to discover new relationships, and potential comorbidities, between diseases.

Discussion

Multilayer networks allow the integration of a wide variety of information. This is particularly useful in biology where different omics provide complementary views of biological systems. MultiXrank [14] is a method that we proposed recently to explore universal multilayer networks with RWR. MultiXrank gives access to RWR output scores, which measure similarities between a seed node to all the other nodes of the multilayer networks. These RWR output scores can be used in several contexts. We have illustrated the use of the RWR in three different biological applications: the prioritization of genes or drugs of interest (node prioritization), the prediction of gene-disease associations in a supervised context (link prediction), and the prediction of comorbidities between diseases. The applications can also be seen as guidelines for further analysis for users.

Funding
The project leading to this preprint has received funding from the "Investissements d'Avenir" French Government program managed by the French National Research Agency (ANR-16-CONV-0001), from Excellence Initiative of Aix-Marseille University - A*MIDEX and from the Inserm Cross-Cutting Project GOLD.

Availability of data and materials
The MultiXrank package is available on GitHub github/MultiXrank, can be installed with standard pip installation command: pypi/MultiXrank, and is associated with complete documentation: https://multixrank-doc.readthedocs.io/en/latest.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Authors' contributions
A.Bap. and A.Bau. designed research; A.Bap. performed research; A.Bap. analyzed data; A.Bap. created numerical code; A.Bap. and A.Bau. wrote the paper.

Author details
[1]INSERM, MMG, Turing Center for Living Systems, Aix-Marseille Univ, Marseille, France. [2]INSERM, TAGC, Turing Center for Living Systems, Aix-Marseille Univ, Marseille, France. [3]Barcelona Supercomputing Center, Barcelona, Spain.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30**(1), 107–117 (1998). doi:10.1016/S0169-7552(98)00110-X. Proceedings of the Seventh International World Wide Web Conference
2. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, USA (2006)
3. Pan, J.-Y., Yang, H.-J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '04, pp. 653–658. Association for Computing Machinery, New York, NY, USA (2004). doi:10.1145/1014052.1014135. https://doi.org/10.1145/1014052.1014135
4. Gómez, S., Díaz-Guilera, A., Gómez-Gardeñes, J., Pérez-Vicente, C.J., Moreno, Y., Arenas, A.: Diffusion dynamics on multiplex networks. Phys. Rev. Lett. **110**, 028701 (2013). doi:10.1103/PhysRevLett.110.028701
5. Köhler, S., Bauer, S., Horn, D., Robinson, P.N.: Walking the interactome for prioritization of candidate disease genes. The American Journal of Human Genetics **82**(4), 949–958 (2008). doi:10.1016/j.ajhg.2008.02.013
6. Li, Y., Patra, J.C.: Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. Bioinformatics **26**(9), 1219–1224 (2010). doi:10.1093/bioinformatics/btq108. https://academic.oup.com/bioinformatics/article-pdf/26/9/1219/29012989/btq108.pdf
7. Li, Y., Li, J.: Disease gene identification by random walk on multigraphs merging heterogeneous genomic and phenotype data. BMC Genomics **13**(7), 27 (2012). doi:10.1186/1471-2164-13-S7-S27
8. Valdeolivas, A., Tichit, L., Navarro, C., Perrin, S., Odelin, G., Levy, N., Cau, P., Remy, E., Baudot, A.: Random walk with restart on multiplex and heterogeneous biological networks. Bioinformatics **35**(3), 497–505 (2018). doi:10.1093/bioinformatics/bty637. https://academic.oup.com/bioinformatics/article-pdf/35/3/497/27699899/bty637.pdf
9. Cho, H., Berger, B., Peng, J.: Compact integration of multi-network topology for functional analysis of genes. Cell Systems **3**(6), 540–5485 (2016). doi:10.1016/j.cels.2016.10.017
10. Ko, Y., Cho, M., Lee, J.-S., Kim, J.: Identification of disease comorbidity through hidden molecular mechanisms. Scientific Reports **6**(1), 39433 (2016). doi:10.1038/srep39433
11. Chen, X., Liu, M.-X., Yan, G.-Y.: Drug–target interaction prediction by random walk on the heterogeneous network. Mol. BioSyst. **8**, 1970–1978 (2012). doi:10.1039/C2MB00002D
12. Peng, L., Shen, L., Xu, J., Tian, X., Liu, F., Wang, J., Tian, G., Yang, J., Zhou, L.: Prioritizing antiviral drugs against sars-cov-2 by integrating viral complete genome sequences and drug chemical structures. Scientific Reports **11**(1), 6248 (2021). doi:10.1038/s41598-021-83737-5
13. Han, N., Hwang, W., Tzelepis, K., Schmerer, P., Yankova, E., MacMahon, M., Lei, W., Katritsis, N.M., Liu, A., Felgenhauer, U., Schuldt, A., Harris, R., Chapman, K., McCaughan, F., Weber, F., Kouzarides, T.: Identification of sars-cov-2-induced pathways reveals drug repurposing strategies. Science Advances **7**(27), 3032 (2021). doi:10.1126/sciadv.abh3032. https://www.science.org/doi/pdf/10.1126/sciadv.abh3032
14. Baptista, A., Gonzalez, A., Baudot, A.: Universal multilayer network exploration by random walk with restart. Communications Physics **5**(1), 170 (2022). doi:10.1038/s42005-022-00937-9
15. Schoenfelder, S., Furlan-Magaril, M., Mifsud, B., Tavares-Cadete, F., Sugar, R., Javierre, B.-M., Nagano, T., Katsman, Y., Sakthidevi, M., Wingett, S.W., Dimitrova, E., Dimond, A., Edelman, L.B., Elderkin, S., Tabbada, K., Darbo, E., Andrews, S., Herman, B., Higgs, A., LeProust, E., Osborne, C.S., Mitchell, J.A., Luscombe, N.M., Fraser, P.: The pluripotent regulatory circuitry connecting promoters to their long-range interacting elements. Genome research **25**, 582–97 (2015)
16. Javierre, B.M., Burren, O.S., Wilder, S.P., Kreuzhuber, R., Hill, S.M., Sewitz, S., Cairns, J., Wingett, S.W., Várnai, C., Thiecke, M.J., Burden, F., Farrow, S., Cutler, A.J., Rehnström, K., Downes, K., Grassi, L., Kostadima, M., Freire-Pritchett, P., Wang, F., Martens, J.H., Kim, B., Sharifi, N., Janssen-Megens, E.M., Yaspo, M.-L., Linser, M., Kovacsovics, A., Clarke, L., Richardson, D., Datta, A., Flicek, P., Stunnenberg, H.G., Todd, J.A., Zerbino, D.R., Stegle, O., Ouwehand, W.H., Frontini, M., Wallace, C., Spivakov, M., Fraser, P.: Lineage-specific genome architecture links enhancers and non-coding disease variants to target gene promoters. Cell **167**(5), 1369–138419 (2016). doi:10.1016/j.cell.2016.09.037
17. Thomas, X., Elhamri, M.: Tipifarnib in the treatment of acute myeloid leukemia. Biologics : targets & therapy **1**(19707311), 415–424 (2007)
18. Yanamandra, N., Buzzeo, R.W., Gabriel, M., Hazlehurst, L.A., Mari, Y., Beaupre, D.M., Cuevas, J.: Tipifarnib-induced apoptosis in acute myeloid leukemia and multiple myeloma cells depends on ca2+ influx through plasma membrane ca2+ channels. J Pharmacol Exp Ther **337**(3), 636 (2011)

19. Luger, S., Wang, V.X., Paietta, E., Ketterling, R.P., Rybka, W., Lazarus, H.M., Litzow, M.R., Rowe, J.M., Larson, R.A., Appelbaum, F.R., Tallman, M.S.: Tipifarnib as maintenance therapy in acute myeloid leukemia (aml) improves survival in a subgroup of patients with high risk disease. results of the phase iii intergroup trial e2902. Blood **126**(23), 1308–1308 (2015). doi:10.1182/blood.V126.23.1308.1308
20. Tyner, J.W., Erickson, H., Deininger, M.W.N., Willis, S.G., Eide, C.A., Levine, R.L., Heinrich, M.C., Gattermann, N., Gilliland, D.G., Druker, B.J., Loriaux, M.M.: High-throughput sequencing screen reveals novel, transforming ras mutations in myeloid leukemia patients. Blood **113**(19075190), 1749–1755 (2009)
21. McGeady, P., Kuroda, S., Shimizu, K., Takai, Y., Gelb, M.H.: The farnesyl group of h-ras facilitates the activation of a soluble upstream activator of mitogen-activated protein kinase. The Journal of biological chemistry **270**, 26347–51 (1995)
22. Himmelstein, D.S., Lizee, A., Hessler, C., Brueggeman, L., Chen, S.L., Hadley, D., Green, A., Khankhanian, P., Baranzini, S.E.: Systematic integration of biomedical knowledge prioritizes drugs for repurposing. eLife **6**, 26726 (2017). doi:10.7554/eLife.26726
23. Runtz, L., Girard, B., Toussenot, M., Espallergues, J., Fayd'Herbe De Maudave, A., Milman, A., deBock, F., Ghosh, C., Guérineau, N.C., Pascussi, J.-M., Bertaso, F., Marchi, N.: Hepatic and hippocampal cytochrome p450 enzyme overexpression during spontaneous recurrent seizures. Epilepsia **59**, 123–134 (2018)
24. Tomić, M., Pecikoza, U., Micov, A., Vučković, S., Stepanović-Petrović, R.: Antiepileptic drugs as analgesics/adjuvants in inflammatory pain: current preclinical evidence. Pharmacology & therapeutics **192**, 42–64 (2018)
25. Swathi, K., Sarangapani, M.: Evaluation of anti-epileptic effect of new indole derivatives by estimation of biogenic amines concentrations in rat brain. Advances in experimental medicine and biology **988**, 39–48 (2017)
26. Ata, S.K., Wu, M., Fang, Y., Ou-Yang, L., Kwoh, C.K., Li, X.-L.: Recent advances in network-based methods for disease gene prediction. Briefings in bioinformatics (2020). doi:10.1093/bib/bbaa303
27. Piñero, J., Queralt-Rosinach, N., Bravo, A., Deu-Pons, J., Bauer-Mehren, A., Baron, M., Sanz, F., Furlong, L.I.: Disgenet: a discovery platform for the dynamical exploration of human diseases and their genes. Database (Oxford) **2015** (2015)
28. Piñero, J., Ramírez-Anguita, J.M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., Furlong, L.I.: The disgenet knowledge platform for disease genomics: 2019 update. Nucleic Acids Res **48**(D1), 845–855 (2020)
29. Liu, H., Zhang, W., Nie, L., Ding, X., Luo, J., Zou, L.: Predicting effective drug combinations using gradient tree boosting based on features extracted from drug-protein heterogeneous network. BMC Bioinformatics **20**(1), 645 (2019). doi:10.1186/s12859-019-3288-1
30. Weischenfeldt, J., Symmons, O., Spitz, F., Korbel, J.O.: Phenotypic impact of genomic structural variation: insights from and for human disease. Nature Reviews Genetics **14**(2), 125–138 (2013). doi:10.1038/nrg3373
31. Spielmann, M., Lupiáñez, D.G., Mundlos, S.: Structural variation in the 3d genome. Nature Reviews Genetics **19**(7), 453–467 (2018). doi:10.1038/s41576-018-0007-0
32. Martinez, F.O., Gordon, S., Locati, M., Mantovani, A.: Transcriptional profiling of the human monocyte-to-macrophage differentiation and polarization: New molecules and patterns of gene expression. The Journal of Immunology **177**(10), 7303–7311 (2006). doi:10.4049/jimmunol.177.10.7303. https://www.jimmunol.org/content/177/10/7303.full.pdf
33. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(86), 2579–2605 (2008)
34. McInnes, L., Healy, J., Saul, N., Großberger, L.: Umap: Uniform manifold approximation and projection. Journal of Open Source Software **3**(29), 861 (2018). doi:10.21105/joss.00861
35. Toomes, C., James, J., Wood, A.J., Wu, C.L., McCormick, D., Lench, N., Hewitt, C., Moynihan, L., Roberts, E., Woods, C.G., Markham, A., Wong, M., Widmer, R., Ghaffar, K.A., Pemberton, M., Hussein, I.R., Temtamy, S.A., Davies, R., Read, A.P., Sloan, P., Dixon, M.J., Thakker, N.S.: Loss-of-function mutations in the cathepsin c gene result in periodontal disease and palmoplantar keratosis. Nature Genetics **23**(4), 421–424 (1999). doi:10.1038/70525
36. Bitoun, E., Chavanas, S., Irvine, A.D., Lonie, L., Bodemer, C., Paradisi, M., Hamel-Teillac, D., Ansai, S.-i., Mitsuhashi, Y., Taïeb, A., de Prost, Y., Zambruno, G., Harper, J.I., Hovnanian, A.: Netherton syndrome: Disease expression and spectrum of spink5 mutations in 21 families. Journal of Investigative Dermatology **118**(2), 352–361 (2002). doi:10.1046/j.1523-1747.2002.01603.x
37. Taylor, A.M., Metcalfe, J.A., Thick, J., Mak, Y.F.: Leukemia and lymphoma in ataxia telangiectasia. Blood **87**, 423–38 (1996)
38. Boultwood, J.: Ataxia telangiectasia gene mutations in leukaemia and lymphoma. Journal of clinical pathology **54**, 512–6 (2001)

Additional Files

Additional file 1 — Supplementary information for: Biological Multilayer Networks Applications
Descriptions of multilayer networks used
Figs. S1 to S11
Tables S1 to S7

Figures

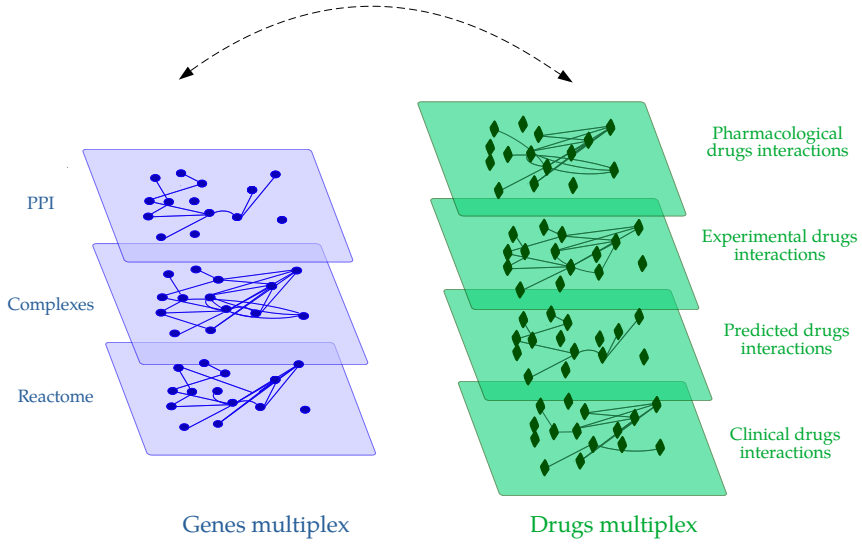


Figure 1 Biological multilayer network composes of a gene and a drug multiplex networks and their associated bipartite network. For the sake of clarity, bipartite networks are not represented.

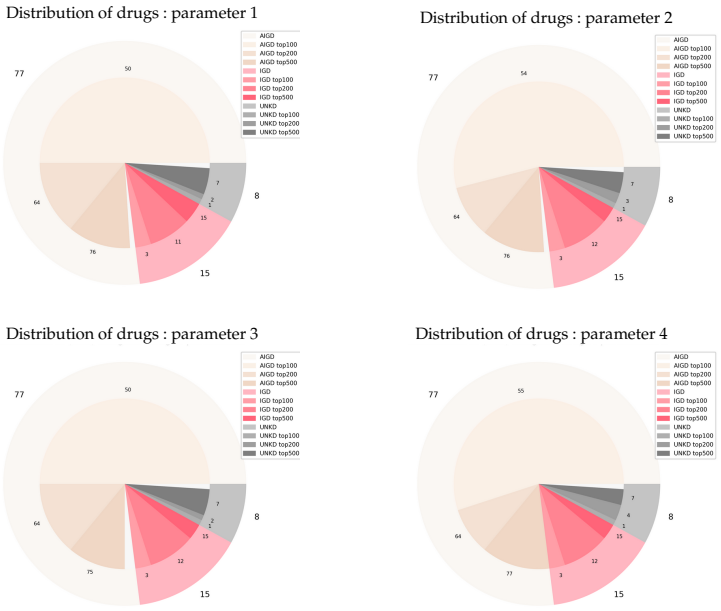


Figure 2 Distribution of MultiXrank top-predicted drugs, for different sets of parameters, and comparison with Hetionet predictions. AIGD represents anti-ictogenic drugs that have a seizure suppressor effect (in beige), IGD represents ictogenic drugs (in pink), and UKND represents drugs with unknown effects (in grey). For each set of parameters (1, 2, 3, or 4), we display in the outer circle the number of AIGD, IGD, and UNKD drugs found by Hetionet, and in the inner pie chart, the number of the same AIGD, IGD, and UNKD found in the top-100, top-200, and top-500 predicted drugs by MultiXrank. The nodes of the top-500 include the nodes found in the top-200, and the top-200 includes the nodes found in the top-100. It is to note that the white part of the pie chart corresponds to the fraction of the Hetionet predicted drugs that are not in our MultiXrank top-500 drugs.

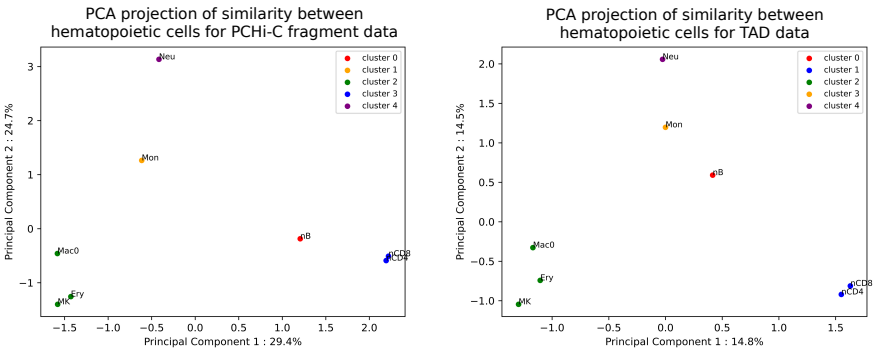


Figure 3 2D PCA (Principal Component Analysis) projection of the similarities between the different hematopoietic cell types. The similarities are computed with the Jaccard index. Each hematopoietic cell type is associated with a vector that represents the Jaccard index between the PCHi-C fragment omics of this hematopoietic cell type and all the other PCHi-C fragment omics corresponding to all the other hematopoietic cell types (on the left). On the right, the similarities are computed with the Jaccard index, each hematopoietic cell type is associated with a vector that represents the Jaccard index between the TAD omics of this hematopoietic cell type and all the other TAD omics corresponding to all the others hematopoietic cell types.

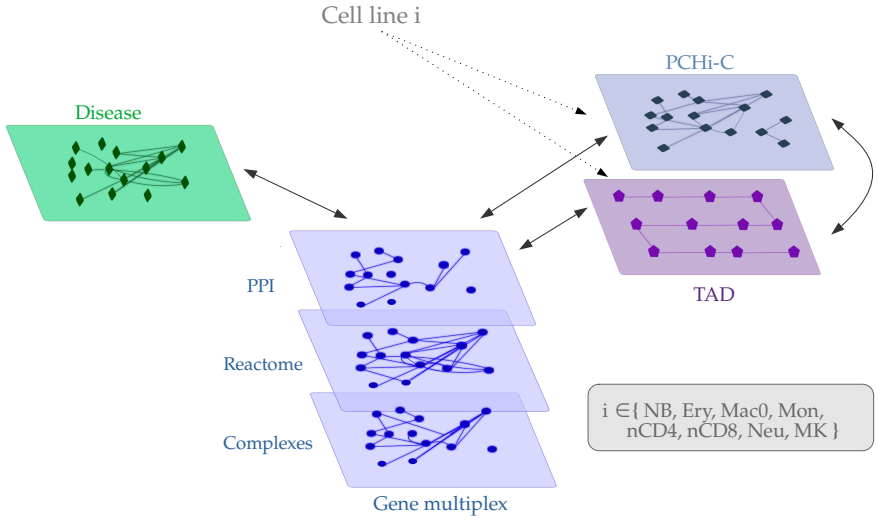


Figure 4 Representation of the multilayer networks constructed with genomic information from PCHi-C and TAD omics (detailed in supplementary section 1.C). The networks generated from these genomic omics are cell-type-specific. The disease network and the gene multiplex network are the same for all cell types.

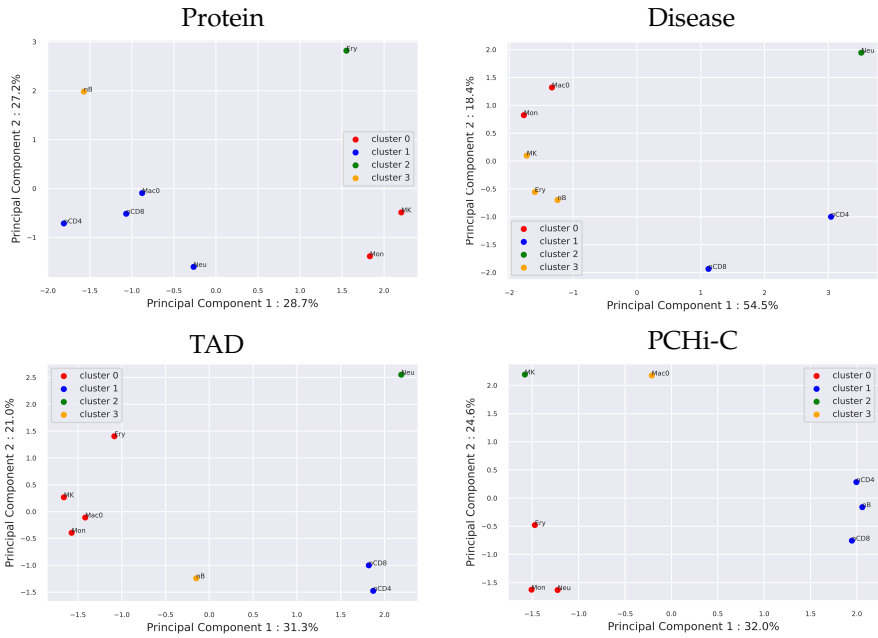


Figure 5 2D PCA (Principal Component Analysis) projection of the similarity between the hematopoietic cell types. The similarity is based on the proximity matrix, obtained with the protocol described in Fig. S6. The tree lineage of hematopoietic cells is correctly found with the PCHi-C fragment output score of RWR, and to a lesser extent with the TAD and the disease output scores of RWR. The tree lineage of hematopoietic cells is however not observed using the gene and disease output scores of the RWR.

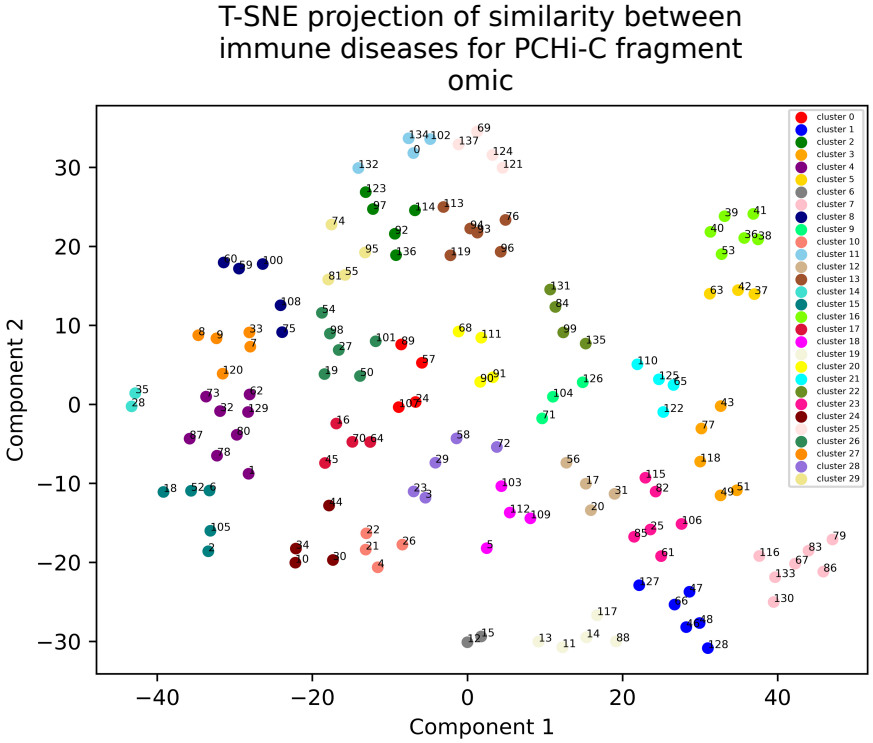


Figure 6 The t-SNE (t-distributed stochastic neighbor embedding) [33] projection in a 2D space of the consensual similarity between the 138 different immune diseases (see supplementary Table S8) obtained for the PCHi-C fragment omics. For this omics, we obtained eight different similarity matrices of size 138×138 (see Fig. S5), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then, the consensual similarity matrix used for the projection is obtained with equation (2).

7.3. Discussion

Une des perspectives que nous souhaiterions donner à ce travail est l'étude des comorbidités entre maladies rares et maladies communes. Cette perspective pourrait s'appuyer sur les résultats encourageants obtenus sur la détection de comorbidités présentés dans le papier. Nous rappelons brièvement que les maladies génétiques rares sont souvent causées par une quantité réduite de mutations ayant une forte pénétrance. De plus, ces mutations sont le plus souvent localisées dans des gènes codant des protéines. Dans le cas où la maladie rare est monogénique, on parle de maladie rare Mendélienne. Les estimations récentes font état d'environ 7 000 maladies rares [218, 219]. D'un autre côté, les maladies communes sont, elles, causées par une grande quantité de variants tout le long du génome et possèdent une forte dépendance vis-à-vis des facteurs environnementaux. De plus, la pénétrance de ces variations est très variable. Une seconde perspective de ce travail est de pouvoir intégrer un réseau monoplex ou multiplex de médicaments (comme ceux définis dans la section 5.1.3) aux réseaux déjà intégrés ayant permis la détection de comorbidités. Un réseau multi-couche universel intégrant des nœuds de gènes, de maladies, de médicaments, de fragments de *PCHi-C* et de *TADs* permettrait de reproduire les résultats de l'article, tout en ayant une information sur les médicaments, ce qui laisse entrevoir la possibilité de définir des similarités non plus seulement entre maladies, mais aussi entre maladies et médicaments. Les pistes de repositionnement de médicaments permettraient de proposer des médicaments développés dans le cadre des maladies communes pour les maladies rares, et, de manière symétrique, d'utiliser les connaissances pathophysiologiques que l'on possède sur les maladies rares afin de mieux expliquer les mécanismes jouant un rôle dans les maladies communes.
Une autre perspective importante concerne le réseau de *TADs*, ce réseau semble moins robuste que le réseau de fragments de *PCHi-C*, cela peut être dû à la manière dont nous l'avons construit. Nous rappelons que deux *TADs* sont connectés s'ils sont adjacents au sein du génome. Adopter une vision tridimensionnelle du réseau de *TADs* en s'appuyant sur les données de *PCHi-C* peut être une piste d'amélioration de ce réseau. Cependant, il est bon de noter que l'existence des *TADs* est sujet à controverse [193]. Sachant que les *TADs* sont générés à partir du séquençage de centaines de millions de cellules, il est tout à fait envisageable de supposer que les *TADs* ne sont que des reliquats statistiques. Cependant, les expériences de Hi-C à cellule unique (*single-cell Hi-C* tendent à confirmer l'existence de structures de grande échelle similaires aux *TADs* [220–222]. Il est aussi envisageable de compléter l'analyse que nous avons faite avec d'autres de données de *PCHi-C* sur des cellules hématopoïétiques, issues de la même plateforme expérimentale, mais publiées dans un autre article [223]. Ces données permettraient d'avoir accès à un plus grand nombre de types cellulaires, et, par conséquent, elles permettraient d'obtenir des prédictions de comorbidités plus complètes.
Finalement, les applications vues dans ce chapitre ne sont pas les seules que nous

avons développées à partir de l'exploration des réseaux multi-couches universels avec les marches aléatoires. Nous verrons en effet, au cours des chapitres suivants, l'utilisation de MultiXrank dans le cadre de la détection de communautés et de l'*embedding* de réseaux.

8. Article 3 : Revue de la littérature pour l'*embedding* de réseaux

Sommaire

8.1. Introduction

Nous avons détaillé la notion d'*embedding* de réseaux en section 1.3.2, et ses applications aux réseaux multi-couches en section 2.3.2. Dans ce chapitre, je vais présenter une revue de la littérature des différentes méthodes d'*embedding* de réseaux, à travers une taxonomie que nous avons développée. Une partie de la revue est de plus dédiée aux applications des méthodes d'*embedding* de réseaux. Cette revue de la littérature a été faite dans le cadre de la préparation d'un article sur une méthode d'*embedding* pour les réseaux multi-couches universels que j'ai développée au cours de ma thèse (voir section 9.3). Au vu de la richesse et de la complexité de ce domaine, j'ai souhaité étudier la littérature afin de replacer mon travail dans le contexte de l'état de l'art. Cette revue de la littérature a pour but de guider les néophytes s'intéressant aux méthodes d'*embedding* et d'homogénéiser la sémantique et le vocabulaire utilisé par les différentes communautés scientifiques.

8.2. *Zoo Guide for Network Embedding*

Zoo Guide of Network Embedding

Anthony Baptista[1,2,*] **and Anaïs Baudot**[1,3, 4,*]

[1]Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, Marseille, France; [2]Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France; [3]Barcelona Supercomputing Center, Barcelona, Spain; [4]CNRS, Marseille, France

This manuscript was compiled on July 20, 2022

Data integration is a hot topic in several fields for decades. The large number of frameworks, models, and methods dedicated to data integration question the visibility of these achievements across communities. This issue exists also for network embedding (a.k.a representation learning). This review aims to enlighten the rich variety of methods and applications of network embedding and propose a user-friendly guideline to explore its rich associated literature. We expect it to help reconciling the semantic used across research communities.

Network Embedding | Representation Learning | Data Integration

Networks are a ubiquitous way to represent and analyse data in a wide variety of research fields. In social sciences, user classification from social network is useful for different tasks such as recommendations, user research, or targeted advertising. Using communications networks, community detection and link prediction could help to better understand the spreading process during rumors or epidemics. In Biology, link prediction in biological networks is also a common task for predicting new therapeutic applications for existing drugs, predicting new gene-disease associations, or just inferring interactions between proteins. The embedding of networks is expected to improve the performances of all these different network analyses. Indeed, network embedding (a.k.a representation learning) presents several advantages. The algorithms built upon network embedding and working in a lower dimension space offer faster and more robust results than the ones on the original (direct) network space. The embedding can be used for a wide variety of downstream analyses, either by direct interpretation of the embedding space or by coupling the vectorial embedding representation with machine learning techniques (1). However, one weakness of network embedding is the interpretability of the results as compared to the methods that work in the direct space. Indeed, vectorial representation is a powerful representation, but the interpretation of the results can be difficult (2).

Network embedding goal is to learn a low dimension vectorial representation from a high dimensional network. To this goal, the relationships between nodes in the graphs are represented by a distance in the vectorial space (a.k.a latent space or embedding space). To make it useful for downstream analysis, network embedding also needs to support network inference, such as link prediction, node classification, community detection, and so on.

Classical approaches to extract topological information from networks were initially associated to node-, edge-, or subgraph-level statistics. Such statistics include for instance node centrality (degree, betweenness (3), closeness (4)), clustering coefficients (5), kernel functions (6), features engineering measuring local neighborhood structures (7), or motifs analyses (8). In parallel, for decades, methods based on dimension reduction appeared as a relevant way to encode topological network information (9–11). The idea is to learn a mapping to embed/project nodes, links, or (sub)graphs in a low dimension vectorial space. These initial methods are the first set of network embedding methods. Many new embedding methods appeared recently, based, for instance, on random walks or deep-learning strategies.

Network embedding, to be efficient, needs to gather several characteristics, that can be summed up as follows:

- Adaptable: Embedding methods need to be applicable in different contexts without repeating a learning step;
- Scalable: Embedding methods need to process large-scale networks in a reasonable time;
- Topology awareness: The distance between nodes in latent space should measure the homophily of the nodes the original network;
- Low dimensional: Embedding methods are expected to reduce the noise present in the original network, to be executed faster than direct methods, and to open the accessibility to machine learning methods designed for vectorial space (1);
- Continuous: The latent space is continuous, which could be beneficial in some tasks like classification (12)

Network embedding methods raise a lot of challenges. First of all, we may wonder which network properties the embedding space needs to preserve. The embedding space may indeed preserve the intra-community similarity, the structural role similarity, or the similarity between node labels and annotation. Then, we may wonder which dimension the embedding space needs to get. The choice of the space dimension is a compromise between correspondence with the original network (choice of a high-dimensional space) and reducing the noise in the original network (choice of low-dimensional space). The

A.Bap. and A.Bau. designed research; A.Bap. performed research; A.Bap. analyzed data; A.Bap. and A.G contributed to packaged code; A.Bap. and A.Bau. wrote the paper.

The authors declare no conflict of interest.

*To whom correspondence should be addressed. E-mail: anthony.baptista@univ-amu.fr, anais.baudot@univ-amu.fr

space needs to be high enough to extract all the relevant information, but not too large to avoid using redundant information and the risk of using just the original network. Furthermore, the scalability of network embedding methods is challenging. Indeed, these methods need to be efficient on large-scale networks, i.e., networks with millions of nodes and billions of edges. Moreover, embedding methods applied to real networks face low parallelizability and data sparsity issues.

1. Definition and preliminaries

A network, defined as $G = (V, E)$, is composed of a set of vertices (nodes) $V = \{v_1, v_2, ..., v_i, ...\}$, and a set of edges (links) $E = \{e_{ij},\ (i,j) \in V \times V\}$. The edge e_{ij} is connected to the two vertices v_i and v_j. We denote the number of vertices by n. In the case of directed networks, $e_{ij} \neq e_{ji}$. The network embedding aims to map each node to a vector in which a certain property between nodes is preserved. in other words, for a network G, we want to find the mapping function :

$$f : V \to \mathbb{R}^d$$
$$v_i \mapsto z_i$$

where the vector z_i is the embedding vector of dimension $d \ll n$ of the node v_i. The embedding vector is expected to capture properties of the node in the original network while reducing the dimension.

As we have seen, the embedding space reduces the space dimension. However, the embedding space needs to preserve some network properties. The choice of the properties to be preserved is not automatic and must be imposed. We will now define the most common properties preserved by network embedding methods.

- The first-order similarity between two vertices is associated with the pairwise similarity between vertices. In other words, it is the weight of the edge between vertices.
- The second-order similarity between two vertices is associated with the similarity of both vertices' neighborhood structures. Let define s_{v_i} (resp. s_{v_j}) the first-order similarity associated with the node v_i (resp. v_j) to the other nodes. The second-order similarity between the nodes v_i and v_j is defined as the similarity between s_{v_i} and s_{v_j}. Higher-order similarities are based on the same idea. These similarities defines structural equivalence between nodes.
- The regular equivalence similarity defines the similarity between vertices that share common roles in their neighborhood. For instance, if a node is a bridge between two communities, or if a node belongs to a clique. The regular equivalence tries to unveil similarity between distant vertices which share common roles.
- The intra-community similarity defines similarity between vertices in the same community. The intra-community similarity tries to preserve the cluster structure information of the networks.

Network embedding methods take as input a network and give as output a vectorial representation. Nonetheless, different types of networks exist, and the network embedding methods have to adapt. Most embedding methods only deal with specific types of networks. However, network embedding methods nowadays tend to integrate more complex and heterogeneous network as input.

- Heterogeneous network: Let define $G = (V, E)$, and $\phi : V \to A$, $\psi : E \to R$, two type-mapping functions that associated each node and each edge to its type. It is to note that if $|A| = 1$ and $|R| = 1$, the network is homogeneous (i.e., composed of only one type of node).
- Signed network: Let define $G = (V, E)$, and $\tau : V \to \{-1, 1\}$. τ is a mapping function that associates a sign to each edge. Positive edges are associated with 1 and negative edges are associated with -1.
- Multilayer network (13–17): They are defined as a triplet $\mathcal{M} = (Y, G, \mathcal{G})$, with $Y = \{\alpha, \alpha \in [\![1, M]\!]\}$ and M the number of layers of the multilayer network. The variable G is defined as a list of layers of the multilayer network $G = (G_1, G_2, ..., G_M)$, such that $G_\alpha = (E_\alpha, V_\alpha)$, and n_α the number of nodes in the network (layer) G_α (defined as $n_\alpha = |V_\alpha|$). The networks, composing G, define intra-layer links of the multilayer network. The variable $\mathcal{G}$ defines the list of the $M(M-1)$ bipartite networks of the multilayer network. Each bipartite network $\mathcal{G}_{\alpha,\beta}$is defined by $\mathcal{G}_{\alpha,\beta} = (V_\alpha, V_\beta, E_{\alpha,\beta})$. The networks, composing $\mathcal{G}$, define inter-layer links of the multilayer network. There is a rich literature on multilayer networks, with different special cases such as multiplex or temporal networks. The interested reader can refer to (18) for an extended overview.
- Temporal networks: It is a specific case of multilayer network where the layers are ordered by time. So $G = \{G^1, G^2, ..., G^T\}$, where T is the number of temporal steps.
- Attributed graph: we have defined homogeneous networks as $G = (V, E)$ with E the set of edges. Each edge is associated with a weight, which can be seen as a one-dimensional attribute. Attributed networks are the generalization to multi-dimensional attributes of homogeneous networks.
- Knowledge graph: It is defined as a set of triplets $(u, r, v) \in V \times R \times V$, where the nodes u and v belong to the node type A, and they are connected by edges of type $r \in R$.

Network embedding methods can also embed different parts of the network. The most common kind of method is the node embedding method, in which each node of the network is projected/embedded into a reduced vectorial representation. However, some methods handle edge embedding. In this case, each edge of the network is projected/embedded into a reduced vectorial representation. Others embedding methods target subgraph or whole-network embedding : the whole network or some of its parts are projected into a vector.

We have witnessed the emergence of general frameworks that propose to group different embedding methods under a common mathematical formulation. We can cite the work of Hamilton et al. (19) that proposed an encoder-decoder framework to organize a wide variety of embedding methods

under the same notations and concepts. This framework organizes the embedding methods following four components:

1. A pairwise similarity function: $s_{\mathcal{G}} : V \times V \to \mathbb{R}^+$.
   This function defines the similarity measure between the nodes in original network space (direct space).

2. An encoder function: $\text{Enc} : V \to \mathbb{R}^d$.
   This function encodes the nodes into the embedding space. For instance, the node $v_i \in V$ is embedded into the vector $z_i \in \mathbb{R}^d$.

3. A decoder function: $\text{Dec} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$.
   This function associates each pair of embedding vectors, a similarity measure in the embedding space.

4. A loss function: $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
   This function measures the quality of the pairwise reconstruction. In other words, it minimizes the errors of the reconstruction as following: $\text{Dec}(\text{Enc}(v_i), \text{Enc}(v_j)) = \text{Dec}(z_i, z_j) \approx s_G(v_i, v_j)$. Most approaches minimize an empirical loss function around a set of training nodes (noted $\mathcal{D}$) to perform the reconstruction objective.

$$\mathcal{L} = \sum_{(v_i,v_j)\in\mathcal{D}} l(\text{Dec}(z_i, z_j), s_{\mathcal{G}}(v_i, v_j))$$

Some classes of embedding methods fit well this formalism. This is the case of shallow embedding methods. However, other methods do not fit exactly with this framework. This is the case for instance of hypergraph embedding methods (20–22) that do not necessarily use a pairwise similarity function.

Another general framework has been proposed by Yang et al. (23) to organize all the existing heterogeneous network embedding (HNE) methods. The idea of the framework is to convert the homophily principle (similar nodes in a network will be close in the embedding space) into a generic objective function.

$$\mathcal{J} = \sum_{v_i,v_j\in V} w_{v_i v_j} d(z_i, z_j) + \mathcal{J}_R$$

The term $w_{v_i v_j}$ denotes the proximity weight, $d(z_i, z_j)$ is the embedding distance function between the embedding vectors associated to the nodes v_i and v_j, $\mathcal{J}_R$ represents some additional objectives like regularizers.

2. Existing classifications of network embedding methods

The huge amount and variety of existing embedding methods (24) make their classification into categories a difficult and not consensual task. Several criteria exist to sort the different methods. We will briefly present some of the most common classifications.

The first way to classify network embedding methods is based on the type of embedded networks. Some authors (24) split the methods using homogeneous networks from the ones using heterogeneous networks. The same idea may be used to separate static network and temporal network embedding methods, or single network and multilayer network embedding methods. Based on this classification, it is possible to add a layer of complexity with the type of embedding, such as node, subgraph or whole network embedding.
Other authors (25) use some properties of the network embedding process to classify the different methods. For instance, the network embedding methods may be divided depending on the network properties they preserve. Three different types of property preservations can be defined, at the "microscopic", the "mesoscopic" properties, or the "macroscopic" scales. Methods preserving "microscopic" properties retain structural equivalences between nodes, such as the first-order, second-order, or high-order similarities between nodes. They hence try to preserve the homophily existing in the original network. Methods preserving "mesoscopic" properties focus on the regular equivalence between nodes, on intra-community similarity, or on any property that is in between the close node neighborhood and the whole network. Finally, methods preserving "macroscopic" properties conserve whole network properties, like the scale-free property (26). A different classification based on a similar idea has been adopted by P. Cui et al. (27). The authors split the network embedding methods into three classes: methods preserving structure and property, methods using classic information, and method preserving advanced information. The first class preserves structural information like the neighborhood or the community structures. The second class constructs the embedding with complementary information like node labels and types (for heterogeneous networks) or edge attributes. The third class gathers supervised methods that propose an end-to-end solution and use complementary information to learn the embedding space.

In conclusion, some authors considered a classification according to the properties preserved by the network embedding methods (23, 25, 27), other authors classified the network embedding methods based on the type and the properties of input networks (12, 24), and other authors classify the network embedding methods based on mathematical consideration (28–30). In our review, we propose a slightly different taxonomy; we intend to be fined-grained and based on a mathematical point of view.

3. A new taxonomy of network embedding methods

The taxonomy that we adopted is based on a mathematical point of view. A classification based on mathematical considerations offers a consensual view and is defined as a fine-grained level. In addition, such a classification is independent of the scientific domain in which the methods have been developed and are applied. This would also allow an easy integration of new methods. In the next section, we adopt the following notation: $G = (V, E)$ is a network composed of a vertices set $V = \{v_1, v_2, ..., v_i, ...\}$ and an edges set $E = \{e_{ij},\ (i, j) \in V \times V\}$. In this context, the edge e_{ij} connects the two vertices v_i and v_j. We denote the number of vertices by $n = |V|$. The dimension of the embedding space is d. The notation $\|.\|_F$ defines the Frobenius norm and $\|.\|_2$ is the euclidean norm.

A. Shallow network embedding methods.

In this section, we will consider the general framework proposed by Hamilton et al. (31). According to this framework, shallow network embedding methods are a set of methods with an encoder function that can be written as follows:

$$\text{Enc}(v_i) = \mathbf{Z}\mathbf{v_i}$$

where $\mathbf{Z}$ corresponds to the matrix with the embedding vectors of all nodes, and $\mathbf{v_i}$ corresponds to the indicator vector associated with each node v_i. In this case, the objective of the embedding process is to optimize the embedding matrix $\mathbf{Z}$ in order to have the best mapping between the nodes and the embedding vectors.

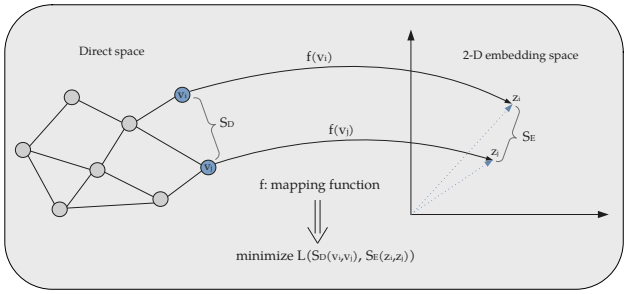


Fig. 1. Shallow network embedding: A network is projected into a low dimensional vectorial space (here in a 2-dimensional space). The mapping function f allows passing from the direct space to the embedding space. The mapping function is obtained thanks to the optimization of a loss function L that minimizes the error of a similarity measure between nodes in the direct space (S_D) and their counterparts into the embedded space (S_E).

A.1. Matrix factorization methods.

Matrix factorization is based on the property that a network can be defined as a matrix. This property implies that all the existing methods in matrix algebra, in particular matrix factorization, can be used for network embedding. Network embedding methods based on matrix factorization are directly inspired by linear dimensionality reduction methods such as PCA (32), LDA (11), or MDS (12). Other methods are inspired by non-linear dimensionality reduction methods such as Isomap (33), which is an extension of MDS (12), LLE (34), t-SNE (35), or more recently UMAP (36). The factorization process depends on the properties of the matrices. If the matrix is positive and semi-definite, the embedding can be obtained by eigenvalue decomposition. However, if the matrix is unstructured, gradient descent or Singular Value Decomposition (SVD) should be used to obtain the network embedding.

- **Laplacian Eigenmaps (LE)** (37) aims to keep in the low-dimensional embedding space two nodes close to each other, when these two nodes are also close on the original data according to a similarity measure. The similarity measure can be for instance the Weight matrix, denoted W, where W_{ij} encode the weight between the nodes i and j. The learning process is done by optimizing the following objective function:

$$\mathcal{L} = \sum_{v_i,v_j \in V} \text{Dec}(z_i, z_j), s_{\mathcal{G}}(v_i, v_j)$$

with $\text{Dec}(z_i, z_j) = \|z_i - z_j\|_2^2$, and $s_{\mathcal{G}}(v_i, v_j) = W_{ij}$.
We can introduce the Laplacian matrix L, defined as $L = D - W$, with $D_{ii} = \sum_j W_{ji}$. The previous equation can be written as:

$$\mathcal{L} = \frac{1}{2}\sum_{i,j}\|z_i - z_j\|_2^2 W_{ij}$$

$$\mathcal{L} = \frac{1}{2}\sum_{i,j}(\|z_i\|_2^2 W_{ij} + \|z_j\|_2^2 W_{ij} + \|z_i z_j\|_2^2 W_{ij})$$

$$\mathcal{L} = \frac{1}{2}(\sum_{i,i}\|z_i\|_2^2 D_{ii} + \sum_{j,j}\|z_j\|_2^2 D_{jj} + 2\sum_{i,j}\|z_i z_j\|_2^2 W_{ij})$$

$$\mathcal{L} = \sum_{i,i}\|z_i\|_2^2 D_{ii} + \sum_{i,j}\|z_i z_j\|_2^2 W_{ij}$$

$$\mathcal{L} = \sum_{i,i}\|z_i z_j\|_2^2 (D_{ii} - W_{ij})$$

$$\mathcal{L} = \sum_{i,i}\|z_i z_j\|_2^2 (L_{ij})$$

$$\mathcal{L} = \text{Tr}(ZZ^T L) = \text{Tr}(Z^T L Z)$$

with $Z = (z_1, z_2, ..., z_n) \in \mathbb{R}^{d \times n}$. The loss function needs to respect the constraint $Z^T D Z = I$ to avoid trivial solutions. The solution can be obtained by finding the matrix composed of the eigenvectors associated with the d smallest eigenvalues of the generalized eigenvalue problem $LZ = \Lambda DZ$, with $\Lambda = diag([\lambda_1, \lambda_2, ..., \lambda_n])$ (38).

- **Cauchy Graph Embedding** (39) aims to improve the previous Laplacian eigenmaps, which used a quadratic decoder function. This function doesn't preserve the local topology because the quadratic penalty de-emphasizes the small distance between embedded nodes. So, these authors propose to adopt a new decoder function equal to $\frac{\|z_i - z_j\|_2^2}{\|z_i - z_j\|_2^2 + \sigma^2} = 1 - \frac{\sigma^2}{\|z_i - z_j\|_2^2 + \sigma^2}$. Consequently, the loss function can be written as follows:

$$\mathcal{L} = \sum_{i,j}\frac{1}{\|z_i - z_j\|_2^2 + \sigma^2} W_{ij}$$

with the following constraints: $\sum_i z_i = 0$, and $Z^T Z = I$, where $Z = (z_1, z_2, ..., z_n) \in \mathbb{R}^{d \times n}$. The solution is obtained by an algorithm that mix gradient descent and SVD.

- **Graph Factorization** (40) proposes a factorization method that is designed for network partitioning. It learns an embedding representation that minimizes the number of neighboring vertices. The loss function can be written as follows:

$$\mathcal{L} = \sum_{v_i,v_j \in V}\|\text{Dec}(z_i, z_j) - s_{\mathcal{G}}(v_i, v_j)\|_2^2 + \frac{\lambda}{2}\sum_{v_i \in V}\|z_i\|_2^2$$

with $\text{Dec}(z_i, z_j) = z_i^T z_j$, $s_{\mathcal{G}}(v_i, v_j) = W_{ij}$, and λ a regularization parameter. It is to note that this method is scalable and can deal with networks with millions of vertices and billions of edges.

- **GraRep** (41) extends the skip-gram model (42) to capture higher-order similarity, i.e., nodes that share common k-step neighbors. The value of k is chosen such as $1 \leq k \leq K$, with K the highest order. GraRep is also motivated by the Noise-Contrastive Estimation (NCE) approximation (43). Grarep defines its k-step loss function as follows:

$$\mathcal{L}_k = \sum_{v_i \in V} (\sum_{v_j \in V} T_{i,j}^k \log(\sigma(x_i^T x_j) + \lambda \mathbb{E}_{j' \sim p_k(V)} [\log(\sigma(-x_i^T x_j))])$$

where the matrix T represents the transition matrix, defined as $T = D^{-1}A$, with A the adjacency matrix, and D the degree matrix. The vectors x_i and x_j are the vector representations of the nodes v_i and v_j in the direct space. The term $\mathbb{E}_{j' \sim p_k(V)}$ is the expectation of the node $j^{'}$, obtained by negative sampling. The expectation follows the distribution over the vertices in the network, denoted by $p_k(V)$. The parameter λ is a parameter that indicates the number of negative samples, and $\sigma(.)$ is the sigmoid function defined as $\sigma(x) = (1 + e^{-x})^{-1}$. GraRep reformulates its loss function, minimization into a matrix factorization problem. Each k-step term is computed from the matrix X^k defined as $X_{ij}^k = \max([\log(\frac{T_{ij}^k}{\sum_m T_{mj}^k}) - \log(\beta)], 0)$. Then, the matrix W^k is constructed, and is the low-dimensional representation. This matrix comes from the singular value decomposition: $\text{SVD}(X^k)$. Finally, the final representation is obtained by concatenating all different order terms, $W = [W^1, W^2, ..., W^K]$.

- **High-Order Proximity preserved Embedding (HOPE)** (44) has been developed to capture higher-order similarity of large-scale networks while also capturing the asymmetric transitivity. HOPE can hence deal with directed networks. The loss function is equal to:

$$\mathcal{L} = \sum_{v_i, v_j \in V} \|\text{Dec}(z_i, z_j) - s_{\mathcal{G}}(v_i, v_j)\|_2^2$$

with $\text{Dec}(z_i, z_j) = z_i^T z_j$. $s_{\mathcal{G}}(v_i, v_j)$ can be any similarity measure. The authors introduce a general factorization in which each similarity measure can be factorized as two matrices, one associated with the global similarity M_g and the other associated with the local similarity M_l. So, the similarity matrix can be expressed as $S = M_g^{-1} M_l$, where both are polynomial sparse matrices. This also enables HOPE to use efficient SVD decomposition for embedding large-scale networks. The authors considered different similarity measures such as Katz index ($S^{katz} = (I - \beta A)^{-1}(\beta A)$), Rooted PageRank ($S^{RPR} = (I - \alpha T)^{-1}((1 - \alpha)I)$), common neighbors ($S^{CN} = I(A^2)$), or Adamic-Adar ($S^{AA} = I(ADA)$). In the previous expression, A represents the adjacency matrix, T represents the transition matrix, α a value $\in [0, 1[$, and β a value inferior to the spectral radius of the adjacency matrix.

- **Modularized Nonnegative Matrix Factorization (M-NMF)** (45) aims to obtain an embedding representation aware of the community structure of the original network, while maintaining the microscopic information from the first-order and second-order similarity. Let's define the similarity measure $S = S^{(1)} + \eta S^{(2)} \in \mathbb{R}^{n \times n}$. $S^{(1)}$ is the first-order similarity matrix, for instance, $S_{ij}^{(1)} = A_{ij}$ with A the adjacency matrix. $S^{(2)}$ is the second-order similarity matrix. It can be defined as follows: $S_{ij}^{(2)} = \frac{\mathcal{N}_i \mathcal{N}_j}{\|\mathcal{N}_i\|_2 \|\mathcal{N}_j\|_2}$, with $\mathcal{N}_i = (S_{i1}^{(1)}, S_{i2}^{(1)}, ..., S_{in}^{(1)})$ the first-order similarity vector of the node i. The parameter η is the weight of the second-order term (often chosen equal to 5 (45)). The embedding of the microscopic structure can be expressed in the NMF framework as the following optimization problem:

$$\min_{M,U} \|S - MU^T\|_F^2 \; ; \quad M > 0 \; , \; U > 0$$

with $M \in \mathbb{R}^{n \times d}$ the non-negative basis matrix, and $U \in \mathbb{R}^{n \times d}$ the non-negative representation matrix; U_i is the representation of the node i.
The community structure is obtained with the modularity maximization that is expressed for two communities as $Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) h_i h_j$, with k_i the degree of the node i, h_i is equal to 1 if the node i belongs to the first community, otherwise is equal to -1, and m is the total number of edges. Let's define B such as $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$, so the modularity becomes $Q = \frac{1}{4m} h^T B h$, where $h \in \mathbb{R}^n$. To generalize the modularity to k communities, we define $H \in \mathbb{R}^{n \times k}$, and the modularity optimization problem is equal to:

$$\min_H \; -\beta \, \text{Tr}\,(H^T B H) \; ; \quad \text{Tr}\,(H^T H) = n$$

The second equation imposes to all the nodes to be associated to one community, β is a positive parameter. Moreover, to combine the two models, the authors add a term that uses the community structure to guide the nodes' representation learning process. To do so, let us define $C \in \mathbb{R}^{k \times d}$ the community representation matrix; $C_r$is the representation of the community r. Thus, $U_i C_r$ represents the propensity of the node i to belong to the community r. So the last term to optimize is equal to: $\alpha \|H - UC^T\|_F^2$, with the constraint that $C > 0$, and α a positive parameter. Finally, the expression to be optimized is the following one:

$$\min_{M,U,H,C} \|S - MU^T\|_F^2 - \beta \, \text{Tr}\,(H^T B H) + \alpha \|H - UC^T\|_F^2$$
$$M > 0 \; , \; U > 0 \; , \; C > 0 \; , \; \text{Tr}\,(H^T H) = n$$

Due to the non-convex behavior of the previous function, a complex optimization process has been developed (45).

- **Text-Associated DeepWalk (TADW)** (46) aims to integrate text data information into the network embedding process. The authors first prove that the learning process used in the Deepwalk embedding method (see the section about Random walk network embedding methods) is equivalent to the optimization of a matrix factorization problem, $M = W^T H$, with $M \in \mathbb{R}^{n \times n}$ the matrix of the original network, $W \in \mathbb{R}^{d \times n}$ the weight matrix, and

$H \in \mathbb{R}^{d \times n}$ the factor matrix. The factorization matrix problem is the following:

$$\min_{W,H} \|M - W^T H\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)$$

The idea of TADW is to take into account a text factor matrix T into the decomposition, such that $M = W^T HT$, with $M \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{d \times n}$, $H \in \mathbb{R}^{d \times k}$, and $T \in \mathbb{R}^{k \times n}$. The new factorization matrix problem is:

$$\min_{W,H} \|M - W^T HT\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2)$$

The optimization process is obtained with a gradient descent algorithm from H. Yu et al. (47)

- **Other methods**: A wide variety of methods have been developed. We have here just presented the most common, but we can briefly mention some other related methods. Some methods are variations around the LE method. For instance, the method named Locality Preserving Properties (LPP) (48) uses a linear approximation of LE. The method Structure-Preserving Embedding (SPE) (49) extends LE by including connectivity structure similarity as a constraint during the learning process. The method Augmented Relational embedding (ARE) (50) modifies the Laplacian matrix to integrate feature information. Spectral techniques to obtain a latent space is also an idea used by Label informed Attributed Network Embedding (LANE) (51) to preserve nodes' structure similarities and the correlations between their labels. Moreover, we can mention methods based on variations around the TADW method. All the following methods are dedicated to multi-class node classification. The method Homophily, structure, and content augmented (HSCA) (52) add a regularization term to the objective function of TADW to enforce the structure homophily existing between nodes in the network. The method Max-Margin DeepWalk (MMDW) (53) adds a multi-class SVM to integrate labeling information of the nodes. The method Discriminative Matrix Factorization (DMF) (54) uses a linear classifier that is trained on labeled nodes to complement the TADW objective function. Finally, several embedding methods, applied to knowledge graphs, use matrix factorization (or tensor factorization). These methods can be defined as relation learning methods. We can mention some of the most common ones, such as RESCAL (55), DistMult (56), which is a special case of RESCAL developed to reduce overfitting, and ComplEx (57) which extent DistMult to complex matrices.

A.2. Random walk-based methods.

The idea behind random walk embedding methods is to encode the scores of the random walk into an embedding space. Most methods use the basic ideas developed in the Deepwalk paper (58). We will present here the most common methods and some of their extensions.

- **Deepwalk** (58) is a scalable network embedding method that uses local information obtained from truncated random walks to learn latent representations. Deepwalk treats the walks as the equivalent to sentences. The process is inspired by the well-known word2vec method (42, 59), in which short sequences of words from a text corpus are embedded into a vectorial space. The first step of Deepwalk consists in generating sequences of nodes obtained from truncated random walks on the graph. Then, the update procedure consists in applying the skip-gram model on sequences of nodes to maximize the probability of observing a node's neighbors conditioned by the embedding representation of the node. The loss function is defined as follows:

$$\min_{\phi} \; -\log(\mathbb{P}(\{v_{i-w}, ..., v_{i+w}\} \setminus v_i \mid \phi(v_i)))$$

the skip-gram model (42) allows to transform the equation as follows:

$$\min_{\phi} \; -\log(\prod_{j=i-w}^{i+w} \mathbb{P}(v_j \mid \phi(v_i))$$

Then, the hierarchical softmax function (60) is defined to approximate the joint probability distribution as:

$$\begin{aligned}\mathbb{P}(v_j \mid \phi(v_i)) &= \prod_{l=1}^{\log(n)} \mathbb{P}(b_l \mid \phi(v_i)) \\ &= \prod_{l=1}^{log(n)} \frac{1}{1 + \exp(-\phi(v_i) * \psi(b_l))}\end{aligned}$$

where v_j is defined by a sequence of tree nodes $(b_0, b_1, ..., b_{\log(n)})$, with b_0 the root of the tree, and $b_{\log(n)}$ the node v_i. To note, similarly to the TADW method, Deepwalk is equivalent to the following matrix factorization problem: $M = W^T H$ (46).

- **node2vec** (61) is a modified version of the Deepwalk method, with two main changes. First, node2vec uses a negative sampling instead of a hierarchical softmax for the normalization. Second, the random walk process used is different. Deepwalk uses a biased random walk, which offers more flexible learning with control parameters. The biased random walk can be described as:

$$\mathbb{P}(c_i = x \mid c_{i-1} = y) = \begin{cases} \frac{\pi_{yx}}{Z} & \text{if } (y,x) \in E \\ 0 & \text{Otherwise} \end{cases}$$

where π_{yx} is the unnormalized transition probability between node y and node x, Z is the normalizing constant. Moreover, the variable π is defined as follows:

$$\pi_{yx} = \begin{cases} \frac{1}{p}\omega_{yx} & \text{if } d_{tx} = 0 \\ \omega_{yx} & \text{if } d_{tx} = 1 \\ \frac{1}{q}\omega_{yx} & \text{if } d_{tx} = 2 \end{cases}$$

where ω_{yx} is the weight of the edge between the node y and the node x, and d_{tx} is the shortest path between the node x and the node t, which is the node reached before the node y. The parameters p and q are two control parameters of the random walk. The parameter p is known as the return parameter; it controls the likelihood of immediately revisiting a node in the walk. The parameter q is known as the in-out parameter; it controls the likelihood of visiting a node in the neighborhood of the nodes that was just visited. Both parameters somehow control if the random

walk will follow a Breadth-first Sampling (BFS) strategy or a Depth-first Sampling (DFS) strategy. Recently, an extension of node2vec to multilayer networks, named Multinode2vec has been developed (62).

- **HARP** (63) proposes a way to improve network embedding methods, including Deepwalk, node2vec, and LINE. The idea is to capture the global structure of an input network by recursively coalescing edges and nodes of the network into smaller networks with similar structures (see section 4.B about network compression). The hierarchy of small networks is a good initialization because it directly proposes a reduced dimension version of the input network while preserving the global structure. The final embedding is obtained by propagating the embedding of the smallest network through the hierarchy.

- **Discriminative Deep Random Walk (DDRW)** (64) aims to capture the topology of the input network. This method is particularly adapted for the network classification task. It can be seen as a Deepwalk extension by taking node attributes information (like TriDNR). To do so, it jointly optimizes the Deepwalk embedding loss function and a classification loss function. The final loss function to optimize is defined as:

$$\mathcal{L} = \eta \mathcal{L}_{\mathcal{DW}} + \mathcal{L}_{\mathcal{C}}$$
$$\mathcal{L}_{\mathcal{C}} = C \sum_{i=1}^{n} (\sigma(1 - y_i \beta^T \theta_i))^2 + \frac{1}{2} \beta^T \beta$$

  with η a weight parameter, σ is defined as $\sigma(x)$ is equal to x, if $x > 0$ and equal to 0 otherwise. The variable θ_i is the embedding vector of the node v_i, y_i is the label of the node v_i, C is the regularizer parameter, and β the subsequent classifier.

- **Walklets** (65). Starting from the assessment that Deepwalk can be derived from a matrix factorization containing the powers of the adjacency matrix (66), it appears that Deepwalk is biased towards lower-powers of the adjacency matrix corresponding to short walks. This can be limiting when the higher-order powers are the appropriate representations to embed the regular equivalence between nodes. To bypass this issue, Walklets propose to learn the embedding directly from the multi-scale representation. This multi-scale representation is sampled from successive higher powers of the adjacency matrix obtained from random walks. Then, after partitioning the relationships by scale, Walklets learns the representation of each node generated for each scale.

- **Struct2vec** (67) aims to capture the regular equivalence between nodes in a network. In other words, two nodes that have identical local network structures should have the same embedding representation. Moreover, Struct2vec also imposes that the latent representation does not depend on the node or edge attributes. The construction of the embedding representation is based on four steps: The first step is to determine the structural similarity between each pair of nodes for different neighborhood sizes. The structural similarity between the nodes v_i and v_j, when considering their k-hop neighborhoods (all nodes at a distance inferior or equal to k and all edges among them), is defined as follows:

$$d_k(v_i, v_j) = d_{k-1}(v_i, v_j) + g(s(R_k(v_i)), s(R_k(v_j))) \; ;$$
$$k \geq 0 \;\text{ and }\; |R_k(v_i)|, |R_k(v_j)| > 0$$

  with $R_k(v_i)$ denoting the set of nodes at a distance inferior or equal to k from the node v_i, $s(S)$ representing the ordered degree sequence of a set of nodes S, $g(S_1, S_2)$ denoting the distance measure between the two ordered degree sequences S_1 and S_2. The distance used is the Dynamic Time Warping (68). Finally, $d_{-1} = 0$ by convention.
  So, a hierarchy of structural similarities between nodes is obtained. Then, from this hierarchy, a weighted multilayer network is created, in which every node is present in each layer, and each layer corresponds to a level of the hierarchy previously defined. The edge weights between all the pairs of nodes are inversely proportional to their structural similarity. After that, a biased random walk process is applied to the multilayer network to generate sequences of nodes which are used to learn the latent representation with the skip-gram model.

- **Other methods**: SemiNE (69) is a semi-supervised extension of Deepwalk. GENE (70) and PPNE (71) are three methods that integrate supplementary information: node labels for GENE and SemiNE, and node attributes for PPNE. It is to note that node labels and attributes are also integrated in TriDNR (72). SNS (73) is another method that aims to preserve structural similarity in the embedding representation. SNS measures the regular equivalence between nodes by representing them as a graphlet degree vector: each element represents the number of times the given node is touched by the corresponding orbit of graphlets.
  A rich literature about Heterogeneous Network Embedding (HNE) has been developed, and random walks are widely used in this context. We can cite: MRWNN (74), SHNE (75), HHNE (76), GHE (77), (78), JUST (79), HeteSpaceyWalk (80), and TapEm (81). The interested reader can refer to Yang et al. (23) for a more detailed review on the HNE embedding. Finally, the metapath-based methods are another set of methods that is usually used for network embedding. This set of methods is often based on random walks such as: Metapath2vec (82), HIN2vec (83), HINE (84), or more recently HERec (85).

A.3. Optimization-based methods.

The most important step in optimization-based methods is to define a loss function that encodes all the properties that we wish to preserve through the embedding. This loss function often gathers similarities between nodes in the direct space, together with some regularizer terms that depend on network features that we want to preserve. The embedding representation is obtained based on the optimization of this loss function. We will present the most common methods and some of their extensions.

- **VERtex Similarity Embeddings (VERSE)** (86) is a versatile network embedding method because *VERSE* accepts any similarity measure in the direct space. We

define a network, noted G, associated with a similarity measure in the direct (i.e., original network) space $sim_G : V \times V \rightarrow \mathbb{R}^+$. The VERSE method constructs the embedding representation of the network G, noted $Z \in \mathbb{R}^{n \times d}$, associated with a similarity measure in the embedding space $sim_E : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$. The embedding representation is based on the optimization of a loss function, noted $\mathcal{L}$, corresponding to the Kullback-Leibler divergence between the similarity matrix in the direct space and the similarity matrix in the embedding space.

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{n} sim_G(v_i, .) \cdot \ln(\frac{sim_G(v_i, .)}{sim_E(v_i, .)}) \\ &= -\sum_{i=1}^{n} sim_G(v_i, .) \cdot \ln(sim_E(v_i, .)) + C\end{aligned}$$

with $C = \sum_{i=1}^{n} sim_G(v_i, .) \cdot \ln(sim_G(v_i, .))$ a constant that can be removed for the optimization. The vector $sim_G(v_i, .)$ corresponds to the vector associated with the node v_i in the similarity matrix defined in the direct space. The similarity matrix in the direct space can be defined by several measures. The authors proposed three different similarity matrices: the adjacency matrix, the SimRank similarity matrix (87), and a similarity matrix based on Random Walk with Restart. The vector $sim_E(v_i, .)$ corresponds to the vector associated with the node v_i in the similarity matrix defined in the embedding space. The vector $sim_E(v_i, .)$ can also be seen as the similarity vector between the vectors z_i and z_j with $j \neq i, j \in [\![1, n]\!]$, where n is the number of nodes in the network. The vectors gathered in the similarity matrix in the embedding space are defined by the following equation:

$$sim_E(v_i, .) = \frac{\exp(z_i \cdot Z)}{\sum_{j=1}^{n} \exp(z_i \cdot z_j^T)}.$$

The node embedding is obtained by optimizing the loss function with a gradient descent algorithm. We suppose that the embedding vectors are initialized with a normal distribution with zero mean. It is to note that the Kullback-Leibler optimization is a time-consuming process. Hence, a negative sampling like NCE (Noise Contrastive Estimation) (43, 88) is often chosen. Recently, an extension of VERSE to heterogeneous multiplex networks, named MultiVERSE, has been developed (89).

- **Large Scale Information Network Embedding (LINE)** (90) aims are to embed both first-order similarity and second-order similarity.

  1. The embedding space of the first-order similarity is obtained with the optimization process described as follows:
     The theoretical probability is defined as follows:

     $$p_1(v_i, v_j) = \frac{1}{1 + \exp(-z_i^T z_j)} \; ; \; z_i \in \mathbb{R}^d$$

     The empirical probability is defined as follows:

     $$p_1^*(v_i, v_j) = \frac{w_{ij}}{W} \; ; \; W = \sum_{(i,j) \in E} w_{ij}$$

     The idea is to minimize the error between the theoretical probability p_1 and the empirical probability p_1^*. To do so, we define the loss function O_1 that minimizes a distance $d(p_1^*(.,.), p_1(.,.))$. LINE uses the Kullback-Leibler divergence. Hence, O_1 can be written as follows:

     $$O_1 = -\sum_{(i,j) \in E} w_{ij} \log(p_1(v_i, v_j))$$

  2. The embedding space of the second-order similarity is obtained with the optimization process described as follows:

     $$p_2(v_j \mid v_i) = \frac{\exp(z_j^T z_i}{\sum_{k=1}^{n} \exp(z_k^T z_i)} \; ; \; z_i \in \mathbb{R}^d$$

     The empirical probability is defined as:

     $$p_2^*(v_j \mid v_i) = \frac{w_{ij}}{d_i} \; ; \; d_i = \sum_{i \in N(i)} w_{ik}$$

     where $N(i)$ is the neighborhood of the node i. It is to note that d_i defines the out-degree of the node i. The idea is again to minimize the error between the theoretical probability and the empirical probability. We define the loss function as $O_2 = d(p_2^*(. \mid v_i), p_2(. \mid v_i))$. LINE use the Kullback-Leibler divergence. Hence, O_2 can be written as follows:

     $$O_2 = -\sum_{(i,j) \in E} w_{ij} \log(p_2(v_j \mid v_i))$$

  As we mentioned, LINE aims to embed first-order and second-order similarity. The two different node representations are computed separately with the loss function described just above. Then, both embedding representations are concatenated for each node.

- **Transductive LINE (TLINE)** (91) is a transductive version of LINE that uses SVM (Support Vector Machine) as a classifier. Both node embedding and the SVM classifier are optimized simultaneously, in order to make full use of the label information existing in the network. TLINE permits fast embedding of large-scale networks by using edge sampling and negative sampling in the stochastic gradient descent process. In more details, the method optimizes a loss function O_T composed of the same loss functions as LINE, O_1 and O_2 to embed both first and second-order similarity, and the SVM loss function O_{SVM}.

  $$\begin{aligned}O_T &= O + \beta O_{SVM} \\ O_{SVM} &= \sum_{i=1}^{n} \sum_{k=1}^{K} \max(0, 1 - y_i^k w_k^T u_i) + \lambda \|w_k\|^2\end{aligned}$$

  with β a trade-off that allows balancing between LINE and SVM, n the number of nodes, K the number of

label types in the network, u_i the embedding vector representation of the node v_i, w_k the parameter vector of the label class k, and y_i^k is equal to 1 if the node v_i is in the class k.

- **Other methods**: A wide range of optimization-based methods are applied to heterogeneous networks. Many of them are similar to LINE, and optimize first and second order similarity. In the case of the method named PTE (Predictive Text Embedding) (92), the loss function is divided into several loss functions: each loss function is associated with one network of the heterogeneous network. APP (Asymmetric Proximity Preserving) (93) network embedding method is similar to VERSE. This method captures both asymmetric and high-order similarities between node pairs thanks to a random walk with restart process.Finally, several embedding methods applied to knowledge graphs are optimization methods. These methods are often called relation learning methods. We can mention some of the most common ones, like the translation-based methods, first defined by Bordes et al. (94). This method, named TransE, embeds multi-relational data that uses directed graphs. Edges can be defined by three elements: the head node (h), the tail node (t), and the edge label (l). The embedding vector of the tail node t should be close to the embedding vector of the head node h, plus some vector that depends on the relationship l. This approach constructs the embedding representation by optimizing a loss function that integrates these three elements. This method has given rise to several alternative methods: TransH (95) which improves TranSE for reflexive/one-to-many/many-to-one/many-to-many relationships. TransR (96) builds entity and relation embeddings in separate entity space and relation space, contrarily to the two previous methods. TransD (97) which is an improvement of TransR for large-scale networks. Recently, the RotatE method has been developed (98). RotatE is a knowledge graph embedding method that can model and infer various relation patterns such as symmetry, inversion, and composition.

B. Deep learning methods.

In recent years, deep learning methods became unavoidable in data analysis. The analysis of network data is no exception. The success of deep learning based methods can be explained by their ability to capture non-linearity in networks and complex features. The first network embedding methods based on deep learning used conventional deep learning techniques, like autoencoders: DNGR (Deep Neural Networks for Learning Graph Representations) (99), SDNE (Structural deep network embedding) (100), VGAE (Variational graph auto-encoders) (101)). Some other methods used Convolutional Neural Network (CNN): DKRL (Description-Embodied Knowledge Representation Learning) (102), PSCN (PATCHY-SAN) (103)). Among the other techniques of conventional deep learning, we can mention the Multi-Layer Perceptron (MLP), for instance used by the method PALE (Predicting Anchor Links via Embedding) (104)), and the Recurrent Neural Network (RNN), for example used by the method Deepcas (105)).

Recently, an important class of deep learning methods for network embedding has been developed: Graph Neural Network (GNN) (106). GNN generalizes the notion of CNN typically applied to image datasets to network. GNN can be seen as an embedding process that encodes high-dimensional information about each node's neighborhood into a dense vector embedding, without feature engineering. A GNN algorithm can be divided into two main components. The encoder that maps a node v_i into a low-dimensional embedding vector z_i, based on the local neighborhood and the attributes of the node, and a decoder, which extracts from the embedding vector user-specified predictions. This kind of method is suitable for end-to-end learning, and offers state-of-the-art performance (106, 107). GNN and their applications to network embedding can be divided into different classes of methods.
The Graph Convolutional Network (GCN) (106), which can be applied for classification (108, 109), or applied on signed networks (SGCN as in Signed graph convolutional network) (110)).
Another important class of embedding methods based on GNN is the Graph Attention Network (GAT) (111–113). Several other embedding methods based on alternative architectures of GNN exist (114–116). The interested reader can refer to the review of Zhou et al. on GNN methods (117).
Finally, graph generative methods are other examples of deep learning methods. These methods are mostly known for Generative Adversarial Networks (GAN) (118). GAN have two components: a generator and a discriminator. The idea of GAN is to train a generator until it is efficient enough to mislead the discriminator. The discriminator is misled when it can't discriminate real data from the data generated by the generator. Based on this idea, several embedding methods appeared, including GraphGAN (119), Adversarial Network Embedding (ANE) (120), and ProGAN (121). An alternative method to GAN is the Restricted Boltzmann Machine (122), which inspired different embedding methods (pRBM (123)).
The different methods detailed above can be classified as follows:

1. Conventional Neural Network
   - Autoencoder
   - Convolutional Neural Network (CNN)
   - Other Neural networks: Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), ...
2. Graph Neural Network (GNN)
   - Graph Convolutional Network (GCN)
   - Graph Attention Network (GAT)
   - Other Graph Neural Networks
3. Graph generative methods

C. Hypergraph methods.

We have seen in section 1 that shallow embedding methods use a pairwise similarity function. This choice is imposed by the structure of the graphs, which connects the nodes by pairwise interactions. However, the graphs have been generalized, and they overcome this limitation. For instance, hypergraphs

connect nodes by hyperedges, which have any number of endpoints. Recently, network embedding adapted to this new formalism, and a wide range of hypergraph embedding methods have been developed. In particular, some of the existing methods (described in sections 3.A-B) have been extended to hypergraphs. We can mention some of these hypergraph embedding methods: Spectral hypergraph embedding (20), HyperEdge-Based Embedding (Hebe) (21, 124) , Deep hypergraph network embedding (DHNE) (125), LBSN2Vec++ (126), Hypergraph neural network (HGNN) (22), Hypergraph Wavelet Neural Networks (HWNN) (127).

4. Evaluations and applications

A. Classical applications.

- **Node classification:** The aim here is to associate a label to each node, based on the information learned with the labels of the other nodes. Network embedding methods embed each node into a vector. These vectors can be used in an unsupervised setting. In this case, the nodes associated with similar node embedding vectors, have similar labels. In a supervised setting, the classifier is trained with the vectors associated with labeled nodes. The classifier is then applied to predict the labels of query nodes.

- **Link prediction:** The aim is to infer interactions between pairs of nodes in a network. The similarity between nodes encodes the propensity of the nodes to be linked. It can be computed, for instance, with an inner product or a cosine similarity between each pair of node embedding vectors. In the embedding space, several operators exist to compute edges between every pair of embedding vectors (61). For instance, it can be obtained from binary operators (Table 1). It is to note that heuristic scores are used to predict the edge between pairs of nodes in the direct space (Table 2).

- **Visualization:** Network embedding methods, as the other reduction dimension methods, can be used to visualize high-dimensional data (with n dimensions) in a lower-dimensional space (with d dimensions). We suppose that similar nodes are close to each other in the representation. However, network embedding methods that were not designed for this specific task propose poor results when directly projected into a two-dimensional embedding space (25, 90). Hence, to visualize the result of network embedding methods, it is frequent to project the low-dimensional embedding space into a two-dimensional space. The two-dimensional space is obtained with dimension reduction methods suitable for visualization, like Principle Component Analysis (PCA) (9, 32), t-distributed Stochastic Neighbor Embedding (t-SNE) (35), or Uniform Manifold Approximation and Projection (UMAP) (36). It is useful to note that $n \gg d \geq 2$.

- **Node clustering/Community detection:** The aim is to determine a partition of the network such that the nodes belonging to a cluster are more similar to each other than between the nodes belonging to different clusters. In practice, any classic clustering method can be directly applied in the latent space to cluster the nodes. K-means (128) is often used for this purpose.

Operator	Definition
Average	$\frac{z_i(k)+z_j(k)}{2}$
Hadamard	$z_i(k) * z_j(k)$
Weighted-L1	$\|z_i(k) - z_j(k)\|$
Weighted-L2	$\|z_i(k) - z_j(k)\|^2$
Cosine	$\frac{z_i(k)\cdot z_j(k)}{\|\|z_i\|\|\|z_j\|\|}$

Table 1. Binary operators to compute edges features between all the pairs of embedding vectors. The variable z_i defines the embedding vector associated with the node v_i, and $z_i(k)$ defines the k-th element of the embedding vectorz_i.

Score	Definition
Common Neighbors	$\|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)\|$
Jaccard's Coefficient	$\|\frac{\mathcal{N}(v_i)\cap\mathcal{N}(v_j)}{\mathcal{N}(v_i)\cup\mathcal{N}(v_j)}\|$
Adamic-Adar Score	$\sum_{t\in\mathcal{N}(v_i)\cap\mathcal{N}(v_j)} \frac{1}{\ln \|\mathcal{N}(v_t)\|}$
Preferential Attachment	$\|\mathcal{N}(v_i)\| \cdot \|\mathcal{N}(v_j)\|$

Table 2. Heuristic scores are used to predict edges between pairs of nodes in the direct space. The variable $\mathcal{N}(v_i)$ defines the neighbor set of nodes associated with the node v_i.

- **Network reconstruction:** The aim here is to reconstruct the whole network based on the learned embedding representation. Let define N, the total number of nodes in the network. The reconstruction imposes $N(N-1)/2$ evaluations to test each potential edge, and each evaluation is equivalent to a link prediction.

B. Emerging applications.

- **Network compression and coarsening:** The aim of network compression is to convert a large network into a smaller network containing a reduced number of edges. This compression is expected to store the network more efficiently and to allow running the network algorithms faster. Network coarsening is often used as a preliminary step in the network embedding process to produce a compression by collapsing pairs of nodes and edges with appropriate criteria.

- **Network classification:** The aim is to associate a label to a whole network. Network classification can easily be applied in the context of whole network embedding. A wide range of applications has been proposed, such as predicting therapeutic effects of candidate drugs based on molecular networks (129), or classifying images that have been converted into networks representation (130).

- **Applications in Knowledge graph:** Let us consider a knowledge graph defined by triplets (u, r, v). There are three main applications of embedding for knowledge graphs. Link prediction is used to infer the interaction between given u and v. Triplet classification, which is a

standard binary classification task, determines if a given triplet (u, r, v) is correct. And finally, knowledge completion aims to determine the missing element in a triplet where only two of the information is known (24, 131).

- **Biological applications:** Network embedding is an active research topic in bioinformatics, and all the classical applications previously mentioned also flourish in the context of biological networks. However, some applications specific to network biology seem to emerge. We will mention some of them and the interested reader can refer to (1, 30) for more detailed reviews. Network alignment aims to find correspondences between nodes in different networks. For instance, we can consider the alignment of Protein-Protein interactions (PPI) networks from two different species in order to identify similar subnetworks (132, 133). Network denoising consists in projecting a graph into an embedding space to reduce the noise by only preserving relevant properties of the original network. For instance, a diffusion process can preserve high-order structures of networks. Network embedding methods can also be used to predict the functions of proteins, or to detect modules in chromosome conformation networks (1).Finally, knowledge graphs are also used in biomedical contexts. For example, Electronic Health Record (EHR) can be represented as a knowledge network that is embedded with other networks which integrate proteins, diseases, or drug information to predict patient outcomes (134, 135).

Conclusion

This guide provides a comprehensive review of the variety of network embedding algorithms, which aim to produce a low-dimensional vector representation of the networks, while preserving different properties of the original networks. We further propose a tentative taxonomy based on their mathematical formulations, to help the readers navigate this quickly evolving field. We defined three classes of methods: the shallow embedding methods, the deep learning methods, and the hypergraph methods. The shallow embedding methods are further divided into three main sub-categories: the matrix factorization methods, the random walk-based methods, and the optimization-based methods. The main part of our review focuses on shallow embedding methods, which are, to the best of our knowledge, the most widely used. The deep learning methods are divided into three sub-categories: The conventional neural network methods, the graph neural network methods, and the graph generative methods. However, some classification choices can be discussed. Moreover, some methods could be better classified as hybrid methods. For instance, some factorization matrix methods use optimization processes, or some deep learning methods can be based on random walks. Overall, we tried here to provide some hints about the network embedding field to the curious readers, but we were not able to solve all the classification and reviewing issues imposed by the huge variety of approaches.

ACKNOWLEDGMENTS. The project leading to this preprint has received funding from the « Investissements d'Avenir » French Government program managed by the French National Research Agency (ANR-16-CONV-0001 and ANR-21-CE45-0001-01).

1. Nelson W, et al. (2019) To embed or not: Network embedding as a paradigm in computational biology. *Frontiers in genetics* 10(31118945):381–381.
2. Chari T, Banerjee J, Pachter L (2021) The specious art of single-cell genomics. *bioRxiv*.
3. Freeman LC (1977) A Set of Measures of Centrality Based on Betweenness. *Sociometry* 40(1):35–41.
4. Bavelas A (1950) Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America* 22(6):725–730.
5. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442.
6. Vishwanathan S, Schraudolph NN, Kondor R, Borgwardt KM (2010) Graph kernels. *Journal of Machine Learning Research* 11(40):1201–1242.
7. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58(7):1019–1031.
8. Pržulj N, Corneil DG, Jurisica I (2004) Modeling interactome: scale-free or geometric? *Bioinformatics* 20(18):3508–3515.
9. Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374(2065):20150202.
10. Robinson SL, Bennett RJ (1995) A typology of deviant workplace behaviors: A multidimensional scaling study. *Academy of Management Journal* 38(2):555–572.
11. Ye J, Janardan R, Li Q (2005) Two-dimensional linear discriminant analysis in *Advances in Neural Information Processing Systems*, eds. Saul L, Weiss Y, Bottou L. (MIT Press), Vol. 17.
12. Chen H, Perozzi B, Al-Rfou R, Skiena S (2018) A tutorial on network embeddings. *CoRR* abs/1808.02590.
13. De Domenico M, et al. (2013) Mathematical formulation of multilayer networks. *Phys. Rev. X* 3(4):041022.
14. De Domenico M, Solé-Ribalta A, Gómez S, Arenas A (2014) Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences* 111(23):8351–8356.
15. Boccaletti S, et al. (2014) The structure and dynamics of multilayer networks. *Physics Reports* 544(1):1–122.
16. De Domenico M, Granell C, Porter MA, Arenas A (2016) The physics of spreading processes in multilayer networks. *Nature Physics* 12(10):901–906.
17. Bianconi G (2018) *Multilayer Networks: Structure and Function*. (Oxford University Press, Oxford), p. 416.
18. Kivelä M, et al. (2014) Multilayer networks. *Journal of Complex Networks* 2(3):203–271.
19. Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: Methods and applications. *CoRR* abs/1709.05584.
20. Zhou D, Huang J, Schölkopf B (2006) Learning with hypergraphs: Clustering, classification, and embedding in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06. (MIT Press, Cambridge, MA, USA), p. 1601–1608.
21. Gui H, et al. (2016) Large-scale embedding learning in heterogeneous event data in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. pp. 907–912.
22. Feng Y, You H, Zhang Z, Ji R, Gao Y (2019) Hypergraph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01):3558–3565.
23. Yang C, Xiao Y, Zhang Y, Sun Y, Han J (2020) Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering* pp. 1–1.
24. Li B, Pi D (2020) Network representation learning: a systematic literature review. *Neural Computing and Applications* 32(21):16647–16679.
25. Zhang D, Yin J, Zhu X, Zhang C (2020) Network representation learning: A survey. *IEEE Transactions on Big Data* 6(01):3–28.
26. Feng R, Yang Y, Hu W, Wu F, Zhuang Y (2018) Representation learning for scale-free networks. *ArXiv* abs/1711.10755.
27. Cui P, Wang X, Pei J, Zhu W (2019) A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* 31(5):833–852.
28. Goyal P, Ferrara E (2018) Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151:78–94.
29. Chen F, Wang YC, Wang B, Kuo CCJ (2020) Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing* 9:e15.
30. Li MM, Huang K, Zitnik M (2021) Representation learning for networks in biology and medicine: Advancements, challenges, and opportunities. *CoRR* abs/2104.04883.
31. Hamilton WL, Ying R, Leskovec J (2018) Representation learning on graphs: Methods and applications.
32. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2(1):37–52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.
33. Samko O, Marshall A, Rosin P (2006) Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognition Letters* 27(9):968–979.
34. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290 5500:2323–6.
35. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. *Journal of Machine Learning Research* 9(86):2579–2605.
36. McInnes L, Healy J, Saul N, Großberger L (2018) Umap: Uniform manifold approximation and projection. *Journal of Open Source Software* 3(29):861.
37. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15(6):1373–1396.
38. Ghojogh B, Karray F, Crowley M (2019) Eigenvalue and generalized eigenvalue problems: Tutorial.
39. Luo D, Ding CHQ, Nie F, Huang H (2011) Cauchy graph embedding in *ICML*. pp. 553–560.

40. Ahmed A, Shervashidze N, Narayanamurthy S, Josifovski V, Smola AJ (2013) Distributed large-scale natural graph factorization in *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13. (Association for Computing Machinery, New York, NY, USA), p. 37–48.
41. Cao S, Lu W, Xu Q (2015) Grarep: Learning graph representations with global structural information in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15. (Association for Computing Machinery, New York, NY, USA), p. 891–900.
42. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.
43. Gutmann M, Hyvärinen A (2010) Noise-contrastive estimation: A new estimation principle for unnormalized statistical models in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, eds. Teh YW, Titterington M. (PMLR, Chia Laguna Resort, Sardinia, Italy), Vol. 9, pp. 297–304.
44. Ou M, Cui P, Pei J, Zhang Z, Zhu W (2016) Asymmetric transitivity preserving graph embedding in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. (Association for Computing Machinery, New York, NY, USA), p. 1105–1114.
45. Wang X, et al. (2017) Community preserving network embedding in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17. (AAAI Press, San Francisco, California, USA), p. 203–209.
46. Yang C, Liu Z, Zhao D, Sun M, Chang EY (2015) Network representation learning with rich text information in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15. (AAAI Press, Buenos Aires, Argentina), p. 2111–2117.
47. Yu HF, Jain P, Kar P, Dhillon I (2014) Large-scale multi-label learning with missing labels in *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, eds. Xing EP, Jebara T. (PMLR, Bejing, China), Vol. 32, pp. 593–601.
48. He X, Niyogi P (2004) Locality preserving projections. *Advances in neural information processing systems* 16(16):153–160.
49. Shaw B, Jebara T (2009) Structure preserving embedding in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09. (Association for Computing Machinery, New York, NY, USA), p. 937–944.
50. Lin YY, Liu TL, Chen HT (2005) Semantic manifold learning for image retrieval in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05. (Association for Computing Machinery, New York, NY, USA), p. 249–258.
51. Huang X, Li J, Hu X (2017) Label informed attributed network embedding in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17. (Association for Computing Machinery, New York, NY, USA), p. 731–739.
52. Zhang D, Yin J, Zhu X, Zhang C (2016) Homophily, structure, and content augmented network representation learning in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. pp. 609–618.
53. Tu C, Zhang W, Liu Z, Sun M (2016) Max-margin deepwalk: Discriminative learning of network representation in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16. (AAAI Press, New York, New York, USA), p. 3889–3895.
54. Zhang D, Yin J, Zhu X, Zhang C (2016) Collective classification via discriminative matrix factorization on sparsely labeled networks in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16. (Association for Computing Machinery, New York, NY, USA), p. 1563–1572.
55. Nickel M, Tresp V, Kriegel HP (2012) Factorizing yago: Scalable machine learning for linked data in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12. (Association for Computing Machinery, New York, NY, USA), p. 271–280.
56. Yang B, tau Yih W, He X, Gao J, Deng L (2015) Embedding entities and relations for learning and inference in knowledge bases. *CoRR* abs/1412.6575.
57. Trouillon T, Welbl J, Riedel S, Gaussier E, Bouchard G (2016) Complex embeddings for simple link prediction in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16. (JMLR.org, New York, NY, USA), p. 2071–2080.
58. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14. (Association for Computing Machinery, New York, NY, USA), p. 701–710.
59. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality in *Advances in Neural Information Processing Systems*, eds. Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ. (Curran Associates, Inc.), Vol. 26.
60. Mnih A, Hinton GE (2008) A scalable hierarchical distributed language model in *NIPS*.
61. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks.
62. Wilson JD, Baybay M, Sankar R, Stillman PE (2018) Fast embedding of multilayer networks: An algorithm and application to group fmri. *ArXiv* abs/1809.06437.
63. Chen H, Perozzi B, Hu Y, Skiena S (2018) Harp: Hierarchical representation learning for networks in *AAAI*.
64. Li J, Zhu J, Zhang B (2016) Discriminative deep random walk for network classification in *ACL*.
65. Perozzi B, Kulkarni V, Chen H, Skiena S (2017) Don't walk, skip! online learning of multi-scale network embeddings in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17. (Association for Computing Machinery, New York, NY, USA), p. 258–265.
66. Yang C, Liu Z (2015) Comprehend deepwalk as matrix factorization. *ArXiv* abs/1501.00358.
67. Ribeiro LF, Saverese PH, Figueiredo DR (2017) struc2vec: Learning node representations from structural identity in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17. (Association for Computing Machinery, New York, NY, USA), p. 385–394.
68. Salvador S, Chan P (2007) Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* 11(5):561–580.
69. Gong M, Yao C, Xie Y, Xu M (2020) Semi-supervised network embedding with text information. *Pattern Recognition* 104:107347.
70. Chen J, Zhang Q, Huang X (2016) Incorporate group information to enhance network embedding in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16. (Association for Computing Machinery, New York, NY, USA), p. 1901–1904.
71. Li C, et al. (2017) Ppne: Property preserving network embedding in *Database Systems for Advanced Applications*, eds. Candan S, Chen L, Pedersen TB, Chang L, Hua W. (Springer International Publishing, Cham), pp. 163–179.
72. Pan S, Wu J, Zhu X, Zhang C, Wang Y (2016) Tri-party deep network representation in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16. (AAAI Press, New York, New York, USA), p. 1895–1901.
73. Lyu T, Zhang Y, Zhang Y (2017) Enhancing the network embedding quality with structural similarity in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17. (Association for Computing Machinery, New York, NY, USA), p. 147–156.
74. Wu F, et al. (2016) Learning of multimodal representations with random walks on the click graph. *IEEE Transactions on Image Processing* 25(2):630–642.
75. Zhang C, Swami A, Chawla NV (2019) Shne: Representation learning for semantic-associated heterogeneous networks in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19. (Association for Computing Machinery, New York, NY, USA), p. 690–698.
76. Wang X, Zhang Y, Shi C (2019) Hyperbolic heterogeneous information network embedding. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01):5337–5344.
77. Chen T, Sun Y (2017) Task-guided and path-augmented heterogeneous network embedding for author identification in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17. (Association for Computing Machinery, New York, NY, USA), p. 295–304.
78. Zhang H, Qiu L, Yi L, Song Y (2018) Scalable multiplex network embedding in *IJCAI*.
79. Hussein R, Yang D, Cudré-Mauroux P (2018) Are meta-paths necessary? revisiting heterogeneous graph embeddings in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18. (Association for Computing Machinery, New York, NY, USA), p. 437–446.
80. He Y, et al. (2019) Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19. (Association for Computing Machinery, New York, NY, USA), p. 639–648.
81. Park C, Kim D, Zhu Q, Han J, Yu H (2019) Task-guided pair embedding in heterogeneous network in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19. (Association for Computing Machinery, New York, NY, USA), p. 489–498.
82. Dong Y, Chawla NV, Swami A (2017) Metapath2vec: Scalable representation learning for heterogeneous networks in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17. (Association for Computing Machinery, New York, NY, USA), p. 135–144.
83. Fu Ty, Lee WC, Lei Z (2017) Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17. (Association for Computing Machinery, New York, NY, USA), p. 1797–1806.
84. Huang Z, Mamoulis N (2017) Heterogeneous information network embedding for meta path based proximity. *ArXiv* abs/1701.05291.
85. Shi C, Hu B, Zhao WX, Yu PS (2019) Heterogeneous information network embedding for recommendation. *IEEE Trans. on Knowl. and Data Eng.* 31(2):357–370.
86. Tsitsulin A, Mottin D, Karras P, Müller E (2018) Verse: Versatile graph embeddings from similarity measures in *Proceedings of the 2018 World Wide Web Conference*, WWW '18. (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE), p. 539–548.
87. Jeh G, Widom J (2002) Simrank: A measure of structural-context similarity in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02. (Association for Computing Machinery, New York, NY, USA), p. 538–543.
88. Mnih A, Teh YW (2012) A fast and simple algorithm for training neural probabilistic language models in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12. (Omnipress, Madison, WI, USA), p. 419–426.
89. Pio-Lopez L, Valdeolivas A, Tichit L, Élisabeth Remy, Baudot A (2020) Multiverse: a multiplex and multiplex-heterogeneous network embedding approach. *arXiv:2008.10085*.
90. Tang J, et al. (2015) Line: Large-scale information network embedding in *Proceedings of the 24th International Conference on World Wide Web*, WWW '15. (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE), p. 1067–1077.
91. Zhang X, Chen W, Yan H (2016) Tline: Scalable transductive network embedding in *Information Retrieval Technology*, eds. Ma S, et al. (Springer International Publishing, Cham), pp. 98–110.
92. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15. (Association for Computing Machinery, New York, NY, USA), p. 1165–1174.
93. Zhou C, Liu Y, Liu X, Liu Z, Gao J (2017) Scalable graph embedding for asymmetric proximity. *Proceedings of the AAAI Conference on Artificial Intelligence* 31(1).
94. Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13. (Curran Associates Inc., Red Hook, NY, USA), p. 2787–2795.

95. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14. (AAAI Press, Québec City, Québec, Canada), p. 1112–1119.
96. Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015) Learning entity and relation embeddings for knowledge graph completion in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15. (AAAI Press, Austin, Texas), p. 2181–2187.
97. Ji G, He S, Xu L, Liu K, Zhao J (2015) Knowledge graph embedding via dynamic mapping matrix in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. (Association for Computational Linguistics, Beijing, China), pp. 687–696.
98. Sun Z, Deng Z, Nie JY, Tang J (2019) Rotate: Knowledge graph embedding by relational rotation in complex space. *ArXiv* abs/1902.10197.
99. Cao S, Lu W, Xu Q (2016) Deep neural networks for learning graph representations in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16. (AAAI Press, Phoenix, Arizona), p. 1145–1152.
100. Wang D, Cui P, Zhu W (2016) Structural deep network embedding in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. (Association for Computing Machinery, New York, NY, USA), p. 1225–1234.
101. Kipf T, Welling M (2016) Variational graph auto-encoders. *ArXiv* abs/1611.07308.
102. Xie R, Liu Z, Jia J, Luan H, Sun M (2016) Representation learning of knowledge graphs with entity descriptions in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16. (AAAI Press, Phoenix, Arizona), p. 2659–2665.
103. Niepert M, Ahmed M, Kutzkov K (2016) Learning convolutional neural networks for graphs in *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, eds. Balcan MF, Weinberger KQ. (PMLR, New York, New York, USA), Vol. 48, pp. 2014–2023.
104. Man T, Shen H, Liu S, Jin X, Cheng X (2016) Predict anchor links across social networks via an embedding approach in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16. (AAAI Press, New York, New York, USA), p. 1823–1829.
105. Li C, Ma J, Guo X, Mei Q (2017) Deepcas: An end-to-end predictor of information cascades in *Proceedings of the 26th International Conference on World Wide Web*, WWW '17. (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE), p. 577–586.
106. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16. (Curran Associates Inc., Red Hook, NY, USA), p. 3844–3852.
107. Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34(13):i457–i466.
108. Kipf T, Welling M (2017) Semi-supervised classification with graph convolutional networks. *ArXiv* abs/1609.02907.
109. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. (Curran Associates Inc., Red Hook, NY, USA), p. 1025–1035.
110. Derr T, Ma Y, Tang J (2018) Signed graph convolutional networks in *2018 IEEE International Conference on Data Mining (ICDM)*. pp. 929–934.
111. Veličković P, et al. (2017) Graph attention networks. *6th International Conference on Learning Representations*.
112. Xu Q, Wang Q, Xu C, Qu L (2017) Attentive graph-based recursive neural network for collective vertex classification in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17. (Association for Computing Machinery, New York, NY, USA), p. 2403–2406.
113. Abu-El-Haija S, Perozzi B, Al-Rfou R, Alemi AA (2018) Watch your step: Learning node embeddings via graph attention in *Advances in Neural Information Processing Systems*, eds. Bengio S, et al. (Curran Associates, Inc.), Vol. 31.
114. Xu K, et al. (2018) Representation learning on graphs with jumping knowledge networks in *ICML*.
115. Ying R, et al. (2018) Hierarchical graph representation learning with differentiable pooling in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18. (Curran Associates Inc., Red Hook, NY, USA), p. 4805–4815.
116. Zhang C, Song D, Huang C, Swami A, Chawla NV (2019) Heterogeneous graph neural network in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19. (Association for Computing Machinery, New York, NY, USA), p. 793–803.
117. Zhou J, et al. (2020) Graph neural networks: A review of methods and applications. *AI Open* 1:57–81.
118. Goodfellow I, et al. (2014) Generative adversarial nets in *Advances in Neural Information Processing Systems*, eds. Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger K. (Curran Associates, Inc.), Vol. 27.
119. Wang H, et al. (2018) Graphgan: Graph representation learning with generative adversarial nets. *Proceedings of the AAAI Conference on Artificial Intelligence* 32(1).
120. Dai Q, Li Q, Tang J, Wang D (2018) Adversarial network embedding in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. (AAAI Press, New Orleans, Louisiana, USA).
121. Gao H, Pei J, Huang H (2019) Progan: Network embedding via proximity generative adversarial network in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19. (Association for Computing Machinery, New York, NY, USA), p. 1308–1316.
122. McClelland JL, Rumelhart DE, Group PR, , et al. (1987) *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. (MIT press) Vol. 2.
123. Wang S, Tang J, Morstatter F, Liu H (2016) Paired restricted boltzmann machine for linked data in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16. (Association for Computing Machinery, New York, NY, USA), p. 1753–1762.
124. Gui H, et al. (2017) Embedding learning with events in heterogeneous information networks. *IEEE Transactions on Knowledge and Data Engineering* 29(11):2428–2441.
125. Tu K, Cui P, Wang X, Wang F, Zhu W (2018) Structural deep embedding for hyper-networks in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. (AAAI Press, New Orleans, Louisiana, USA).
126. Yang D, Qu B, Yang J, Cudre-Mauroux P (2020) Lbsn2vec++: Heterogeneous hypergraph embedding for location-based social networks. *IEEE Transactions on Knowledge and Data Engineering* pp. 1–1.
127. Sun X, et al. (2021) Heterogeneous hypergraph embedding for graph classification. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.
128. Macqueen J (1967) Some methods for classification and analysis of multivariate observations in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. pp. 281–297.
129. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P (2016) Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* 30(8):595–608.
130. Bruna J, Zaremba W, Szlam AD, LeCun Y (2014) Spectral networks and locally connected networks on graphs. *CoRR* abs/1312.6203.
131. Feng J, Huang M, Yang Y, Zhu X (2016) GAKE: Graph aware knowledge embedding in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. (The COLING 2016 Organizing Committee, Osaka, Japan), pp. 641–651.
132. Fan J, et al. (2018) A multi-species functional embedding integrating sequence and network structure. *bioRxiv*.
133. Heimann M, Shen H, Safavi T, Koutra D (2018) Regal: Representation learning-based graph alignment in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18. (Association for Computing Machinery, New York, NY, USA), p. 117–126.
134. Rotmensch M, Halpern Y, Tlimat A, Horng S, Sontag D (2017) Learning a health knowledge graph from electronic medical records. *Scientific Reports* 7(1):5994.
135. Wu T, et al. (2019) Representation learning of ehr data via graph-based medical entity embedding. *ArXiv* abs/1910.02574.

8.3. Discussion

Cette revue présente une taxonomie pour unifier la présentation des différentes méthodes d'*embedding*. Cependant, cette taxonomie ne résout pas toutes les difficultés. En effet, bon nombre de méthodes sont hybrides et mélangent plusieurs aspects de notre taxonomie. Par exemple, il existe des méthodes d'apprentissage machine qui utilisent les marches aléatoires, ou bien des méthodes de factorisation de matrice qui utilisent des méthodes d'optimisation. Néanmoins, il semble difficile de résoudre ces difficultés de classification, sans entrer dans une classification trop spécifique.

9. Autres méthodes et projets développés

Sommaire

Mon travail de thèse ne se restreint pas aux travaux présentés aux chapitres 6, 7 et 8. J'ai également travaillé au développement d'autres méthodes : des méthodes de détection de communautés et de partitionnement sur des réseaux multi-couches universels et une nouvelle méthode d'*embedding* de réseaux qui ouvre la voie de l'*embedding* aux réseaux multi-couches universels. Ces méthodes sont basées sur les scores obtenus à l'aide des marches aléatoires avec *restart* sur les réseaux, y compris sur les réseaux multi-couches universels, grâce à mon algorithme MultiXrank présenté au chapitre 6. Nous avons aussi généralisé la similarité de Katz au cas des réseaux multi-couches universels. Dans ce chapitre, on présentera dans un premier temps les méthodes de détection de communautés et de partitionnement de réseaux, puis la généralisation de la similarité de Katz aux réseaux multi-couches universels. Enfin, nous conclurons ce chapitre avec la méthode d'*embedding* adaptée aux réseaux multi-couches universels.

9.1. Méthode de détection de communautés et de partitionnement de réseaux à l'aide de marche aléatoire avec *restart*

La méthode de marche aléatoire avec *restart* que j'ai développée au cours de ma thèse et qui est présentée au chapitre 6 ouvre la voie à une grande variété de méthodes reposant sur les scores obtenus à partir de ces marches aléatoires. Ainsi, l'outil MultiXrank est le point de départ de deux méthodes que j'ai développées au cours de ma thèse, une pour la détection de communautés et une pour le partitionnement de réseaux. Ces projets sont également basés sur la méthode proposée par Macropol et al. [224] (*Repeated Random Walk (RRW)*), qui utilise les marches aléatoires avec *restart* afin de définir une communauté de nœuds de taille k autour d'un nœud initialement choisi comme graine de la marche aléatoire avec *restart*. La méthode de *RRW* utilise les scores des marches aléatoires afin de déterminer le nœud le plus proche de la graine de départ. Ce nœud est ensuite ajouté à l'ensemble des graines, pour le moment constitué uniquement de la graine de départ. Ensuite, le processus de marche aléatoire avec *restart* est relancé avec comme graines les deux graines de l'ensemble. Les scores des marches aléatoires permet de déterminer le nœud le plus proche de ces deux graines. Ce nœud est ajouté à l'ensemble des graines, puis le processus est répété itérativement jusqu'à obtenir un ensemble de graines de taille k qui correspond à une communauté. D'une certaine manière, ce processus peut être vu comme un processus de nucléation autour d'une graine de départ. Cette idée a été utilisée par Alberto Valdeolivas dans le cadre de son doctorat dans mon équipe d'accueil. Alberto a travaillé sur une méthode de marche aléatoire avec *restart* pour les réseaux multiplex et multiplex-hétérogènes [138]. Au cours de sa thèse, il a également utilisé cette méthode de *RRW* afin de définir des communautés au sein de réseaux. Cependant, ces *RRW*, étant basés sur la marche aléatoire avec *restart*, ils étaient limités aux cas des réseaux monoplexes, multiplexes ou d'un réseau multiplex associé à un réseau hétérogène par un réseau bipartite. Il n'était pas possible d'intégrer plus de deux réseaux hétérogènes ni d'explorer un réseau multi-couche universel. Cette limitation de la marche aléatoire avec *restart* ayant été levée par notre méthode MultiXrank, l'opportunité d'utiliser l'approche de RRW pour identifier des communautés sur des réseaux multi-couches universels s'est présentée. Dans le court article qui suit, nous avons démontré l'efficacité de cette approche de détection de communautés sur les réseaux multi-couches universels. Nous avons aussi développé une extension permettant le partitionnement de réseaux [210].

Le matériel supplémentaire de l'article est disponible dans l'annexe D, en fin de manuscrit.

Clustering Multilayer Networks with Random Walk with Restart

Anthony Baptista[1,2,*], **Alberto Valdeolivas**[3,*], **Ozan Ozisik**[1], **and Anaïs Baudot**[1,4,*]

[1]Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, Marseille, France; [2]Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France; [3]Roche Pharma Research and Early Development, Basel, Switzerland; [4]Barcelona Supercomputing Center, Barcelona, Spain

*** anthony.baptista@univ-amu.fr, alvaldeolivas@gmail.com, anais.baudot@univ-amu.fr**

Abstract

Clustering is a common task in network analysis. The aim of graph clustering is to identify subgraphs, named clusters, such that nodes share more interactions within their cluster than with the rest of the graph. Community detection is a specific case of clustering, in which a cluster is defined around a node of interest. Network partitioning is another specific case of clustering, in which enough communities are defined to cover the whole network. In this report, we will present new methods based on Random Walk with Restart for both community detection and network partitioning from multilayer networks.

Multilayer Networks | Random walk | Clustering | Community detection | Network partitioning

Introduction

Networks are particularly successful to represent and exploit complex data. Among the different methods developed to analyse networks, clustering methods have been a focus of research for years. Clustering methods aim to define subgraphs enriched in interactions. We can identify two different clustering strategies: Community detection, which aims to define a subgraph (called community or cluster) around a node of interest; and network partitioning, which aims to define enough clusters to cover the whole network. A wide range of methods exist for network clustering. These methods are based on diverse assumptions and algorithms (1). Random walk based methods, such as Walktrap (2) and Infomap (3), are among the most prominent approaches. Recently, Macropol et al. have developed a clustering method based on Repeated Random Walks (RRW) (4). This method relies on Random Walk with Restart (RWR), an alternative of the PageRank algorithm (5). In RWR, instead of restarting in random nodes, the restart is restricted to one or several specific node(s), called seed(s) (6). In this case, the random walk represents a measure of proximity from all the nodes in the network with respect to the seed(s). The authors of the RRW clustering approach used RWR to define a cluster of size k from a seed. The proximity measure obtained from the RWR process is used to select the node the closest to the seed. This closest node is then added to a set composed of the seed. Then, the RWR process is run again with the new set of two seeds to obtain the closest node to these two seeds. This closest node is again added to the set of seeds. This process is repeated iteratively until obtaining a set composed of k seeds. This final set forms the cluster.

The original RRW methods can consider as input only a single *monoplex* network, and only identifies one community (4). However, clustering algorithms able to leverage more complex networks are becoming necessary. Indeed, the amount and heterogeneity of data available have been increasing drastically for several years, offering a unique opportunity to better understand complex systems. These more complex datasets are better represented by multilayer networks, which allow the integration of more than one network in a common framework. In the multilayer framework, different networks are considered as layers and are connected by bipartite networks (7). Multilayer networks have been shown to provide a more complete and accurate description of complex systems in several fields such as economy and finance (8, 9), social science (10–12), ecology (13–15), or biology (16, 17). Various network exploration algorithms have been extended to consider these more complex but richer network frameworks. In particular, RWR can nowadays navigate universal multilayer networks composed of several multiplex networks connected with bipartite networks (18).

In biology, defining clusters from networks is fundamental. For instance, in molecular networks containing interactions between genes and proteins, groups of tightly connected nodes (often called modules) contain genes/proteins likely to be involved in the same cellular functions or processes (19). Defining topological modules from multilayer networks that integrate a wide variety of information may improve the identification of clusters.

We present here two clustering methods in universal multilayer networks. These methods are dedicated to i) community identification; ii) network partitioning. Importantly, the universal multilayer networks that can be clustered by our methods can be composed of any combination of monoplex and multiplex networks connected by bipartite networks, and all interactions can be weighted or directed. The two clustering methods are based on MultiXrank, a tool allowing the running of RWR on universal multilayer networks (18), combined with the idea of clustering using RRW (4). We explore the behavior of these clustering methods thanks to their application

to explore an airports multilayer network composed of three multiplex networks with their associated bipartite networks, and a large biological disease network.

Results

New multilayer network clustering approaches for community detection and network partitioning.

Community detection The community detection algorithm aims to determine a community around a node of interest, called the seed. We use Random Walk with Restart (RWR), which returns a list of ranked scores corresponding to proximity measures of all the nodes of the network with respect to this seed. The basic idea, coming from the RRW algorithm (4), is to grow the size of the set of seeds, which will ultimately correspond to the community when the process will have converged. Contrarily to the previous RRW approach, which was based on a fixed community size, we introduce a variable convergence condition, based on the threshold defined below. We start the process with a given seed. Then, we run RWR, which returns the node scores associated with this seed. We select the top-ranking node for all the node types, each associated with a monoplex/multiplex network. We include these top-ranked nodes in the set of seeds for the next iteration if and only if $|\frac{\text{score}}{\text{old score}}| > \epsilon$, with ϵ a threshold. Therefore, the set of seeds grows until none of the top-ranking nodes satisfy the previous condition. In this case, the set of seeds reaches its final size, and is considered as the community.

Network partitioning We also propose an approach for the partitioning of full multilayer networks. In this case, we apply the process described above but instead of stopping the process when the community converges and reaches its final size, we select randomly one of the remaining nodes of the network (i.e., a node not part of the defined community) as a new seed, and we apply again the same community detection process until convergence using this new seed. We iterate the process until obtaining a global network partition, i.e. until all the nodes are associated with at least one cluster.

Application to the clustering of small multilayer network.

Community detection begins in a starting seed. We can wonder to what extend the degree of the starting seed would affect the growing process of the set of seeds and the obtained community. We computed the node degree distribution in the multiplex networks composing the airports multilayer network (Fig. 1A). We selected the highest and lowest degree nodes (or one of the highest/lowest degree nodes, if several exists) of each multiplex network, i.e., nodes #7 and #394 for the French airports multiplex network, nodes #18 and #383 for the British airports multiplex network, and nodes #38 and #56 for the German airports multiplex network. Fig. 1B-C show the sizes communities obtained when starting the community identification process using the node #7 from the French airports multiplex network as a seed (the results obtained using the other highest/lowest degree nodes as seeds are available in Fig. S1). Fig. 1B represents the evolution of the final community size depending on threshold values ranging from 0.1 to 2.0. We observe that the final community size ranges from the whole network size when a low threshold is selected to a tiny community when a high threshold is selected. Indeed, when the threshold is high, the addition of a new node to the community is hard. The green curve represents the Gaussian fit. Its inflection point gives the value for which we observe an abrupt behavior. We can expect to have the best threshold value around this value. Indeed, this parameter value can help select enough nodes in the community to avoid dealing with a very small community, but not too much, to avoid having a final community size that equals the whole network size. Importantly, every starting seed we tested gave the same inflection threshold value equal to 0.8. So, for following analyses dedicated to network partitioning, we selected threshold values around 0.8. Fig. 1C illustrates the evolution of the community size through the different multiplex networks for different threshold values. This figure represents the community size evolution as in Fig. 1B, but in Fig. 1C we associate each node of the community to its multiplex network of origin. The different threshold values are represented by the color shades, and each color represents the nodes of a specific multiplex network. The external part of the pie chart gives the total number of nodes in each multiplex network. The inner part gives the number of nodes in the final community depending on their multiplex network of origin. It is to note that low threshold values lead to communities, including all the nodes of the communities obtained with higher threshold values. High threshold values (1.0 or 1.3) allow gathering mainly nodes from the multiplex network containing the starting seed (the blue one for node 7). Contrarily, lower threshold value (0.8) seems to remove this bias and, finally, for very low threshold values (0.3) almost no selection appears and the community reaches the whole network size with nodes from all the multiplex networks. Hence, when a threshold around 0.8 is selected, the community gathers the different types of nodes from the different multiplex networks.

Network partitioning: the total number of clusters and the size of their intersections depends on the convergence threshold value.

We now apply the network partitioning algorithm to partition the complete airport multilayer network into clusters. Fig. 2 represents an upset plot comparing the clusters obtained for a threshold value of 0.8. This figure shows that the 69 nodes of the three multiplex networks are clustered into 10 clusters of different sizes, ranging from 7 to 67 nodes. Several nodes are redundant, i.e. they are associated with several clusters. We observe that the higher the threshold value is, the tinier the size of the clusters is. When the threshold value is high, we hence need more clusters to partition the whole network (Fig. S2). The same whole network partitioning was repeated using systematically all the network nodes as a starting seed (Fig. S3-S6). This network partitioning approach is different from the previous one, which was selecting nodes as seeds until all the nodes were associated with at least one cluster. In the context of the systematic selection of nodes as seeds, we observe that the distribution of the sizes of the cluster intersections also depends on the threshold value. More precisely, a large variability of the cluster intersection sizes is observed for low threshold value (0.7, 0.8, Fig. S3-S4), and a more continuous distribution for high threshold value (0.9, 1.0, Fig. S5-S6).

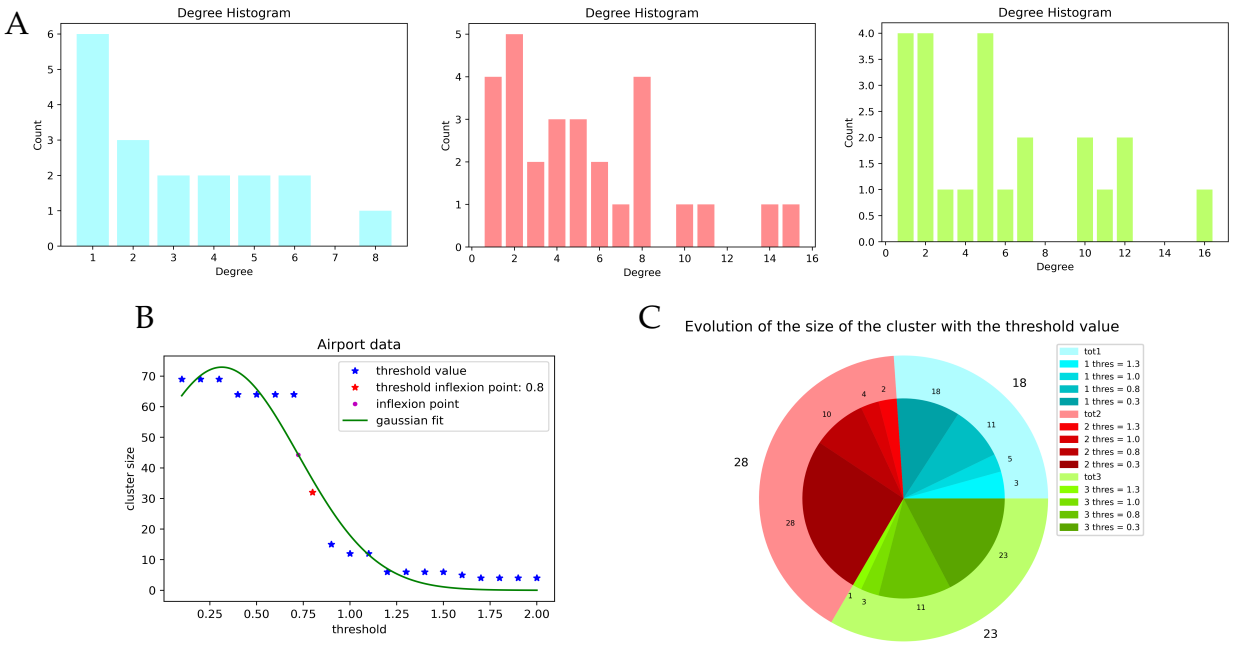


Fig. 1. Clustering analysis of the airports multilayer network. A: Degree distribution of the different multiplex networks which constitute the airports multilayer network: French airports (in blue, multiplex network 1), British airports (in red, multiplex network 2) and German airports (in green, multiplex network 3). B-C: Both charts represent the evolution of the community's final size depending on the threshold value. The chart B represents the evolution of the final community size depending on a threshold value ranging from 0.1 to 2.0, when the node #7 from the French airports multiplex network is selected as a seed. The green curve represents the Gaussian fit with its inflection (pink point) that gives the value for which we observe an abrupt behavior. We expect the best threshold value around this inflection point. The red point represents the closest experimental point (here associated with a threshold value equal to 0.8). The chart C represents the evolution of the community size for different threshold values represented by shades of colors. The colors represent the nodes of a specific multiplex network, blue for the French airports multiplex network, red for the British airports multiplex network, and green for the German airports multiplex network. The external part of the pie chart gives the total number of nodes in each multiplex network, and the inner part gives the number of nodes in the final community depending on the threshold value.

Application to the clustering of a large monoplex networks.

In order to test the approach on a large network, we consider a biological network containing disease-disease relationships. In the case of this large network, the best threshold values are slightly different than in the case of the small multilayer network. We can also determine the best threshold values with the inflection point of the Gaussian fit of the curve that represents the evolution of the cluster's final size depending on the threshold value. This might indicate a common response of the algorithm whatever the input networks. Fig. 3A represents the degree distribution of the disease monoplex network, and Fig. 3B illustrates the influence of the threshold values on the final cluster size (the starting seed is UMLS:C0033300, the Progeria disease). The range of interesting threshold values is here close to 0.4. Fig. 3C shows the distribution of the sizes of the clusters obtained using the network partitioning algorithm with three different threshold values (0.50, 0.55, 0.60), as well as their power-law fits. We see that the number of clusters retrieved increases with the threshold values, and also that the average size of the clusters is noticeably lower for high threshold value. These two behaviours were also described in the case of the small airport network.

Materials and Methods

A. Codes.

The following codes correspond to the different analyses presented in this report. The first three codes concern the community detection algorithm, the others concern the clustering algorithm.

- **choosing_node.py:** It computes the degree distribution histogram of the different multiplex networks (Fig. 1A, Fig. 3A). It also returns the nodes with the highest and lowest degrees in the different multiplex networks. Both nodes are useful to test the community identification algorithm starting with seed nodes with different degrees.
- **community.py:** It generates the community detection algorithm described in section A. It can also make the exploration of the threshold value. It was usually performed from 0.1 to 2.0, with various steps, a 0.1 step for the airports multilayer network analysis (Fig. 1B). The start and the end value for the exploration of the threshold values can be changed.
- **analysis_community_size.py:** Based on the results obtained with the threshold value exploration process, we can generate the curve that corresponds to the size of the final community depending on the value of the threshold. We estimate the experimental dots curve with a Gaussian fit to extract the inflection point that gives us a range of values that allows us to expect to obtain a threshold value. This threshold value corresponds with a value that gives a community that is large enough to be useful and not too large to avoid corresponding with the whole network size (Fig. 1B). The second representation given by the

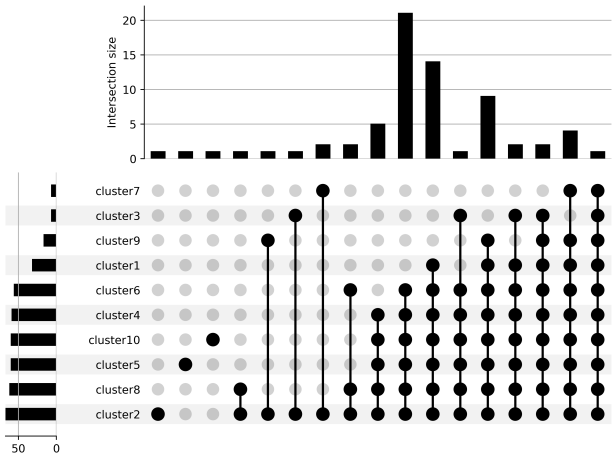


Fig. 2. Upset plot comparing the clusters obtained from the partitioning of the airports multilayer network with a threshold value equal to 0.8. The barplot on the left represents the total size (i.e., number of nodes) of each cluster. All the existing intersections are represented in the combination matrix, and the sizes of the intersections are shown on the top barplot.

code is a pie chart that represents the evolution of the size of the final community for different threshold values. We display separately nodes from the different multiplex networks (Fig. 1C, Fig. 3D).

- **clustering_all.py:** It generates a partition of the networks according to the community detection algorithm described in section A. In the beginning, we start from a starting seed that grows until convergence. Then, we select a seed from the remaining nodes not present in the cluster defined in the previous step, and we repeat the same process. When it converges, we take again a seed from the remaining nodes, and we iterate this process until the set of remaining nodes is empty. In the end, all the nodes are associated with at least one cluster. This partitioning algorithm enable multiple associations of nodes to clusters. It is to note that the analysis is performed for a specified threshold value.
- **clustering_all_each_node.py:** This code runs the community detection algorithm of section A for a specific threshold value. All the nodes are considered as the starting seed iteratively. At the end, it returns all the clusters obtained from all the possible starting seed. This code could be seen as a generalization of the clustering_all.py code: it does not stop when we have a global partition (every node associated with at least one cluster), but rather proposes an exhaustive view (Fig. S3-S6).
- **upsetplot_clusters.py:** It generates an upset plot (Fig. 2, a representation to visualize the intersection of sets). This allows comparing the composition of the different clusters. It also allows checking if the partition generates redundant nodes, i.e. nodes affected to different clusters.
- **distribution_clustering_all.py:** It generates the distribution of the size of the different clusters for a predefined threshold value. It is possible to display the distributions obtained from different threshold values (Fig. 3C).

B. Airport networks.

The different multiplex networks were generated from the connections between airports given by data published in Cardillo et al (20). The mapping of the airports to their respective countries has been done thanks to the database https://openflights.org/data.html.
The companies were chosen based on their number of connections. We took the top-3 most connected companies for each country.

1. **French airports multiplex network:**
   - **FR3:** Easyjet connections between French airports.
   - **FR7:** Air France connections between French airports.
   - **FR26:** Netjets connections between French airports.
2. **British airports multiplex network:**
   - **UK3:** Easyjet connections between British airports.
   - **UK15:** Flybe connections between British airports.
   - **UK26:** Netjets connections between British airports.
3. **German airports multiplex network:**
   - **G1:** Lufthansa connections between German airports.
   - **G6:** Air Berlin connections between German airports.
   - **G24:** Germanwings connections between German airports.

The bipartite networks correspond to the transnational flights operated by these different companies between France and UK, France and Germany, and British and Germany.

C. Disease monoplex network.

The network was built on phenotypic proximity between diseases calculated thanks to the information content. The whole protocol is described in (17). We chose the UMLS annotation for diseases in this monoplex network and its associated bipartite networks.

ACKNOWLEDGMENTS. The project leading to this report has received funding from the « Investissements d'Avenir » French Government program managed by the French National Research Agency (ANR-16-CONV-0001), from Excellence Initiative of Aix-Marseille University - A*MIDEX and from the Inserm Cross-Cutting Project GOLD.

1. Choobdar S, et al. (2019) Assessment of network module identification across complex diseases. *Nature Methods* 16(9):843–852.
2. Pons P, Latapy M (2005) Computing communities in large networks using random walks in *Computer and Information Sciences - ISCIS 2005*, eds. Yolum p, Güngör T, Gürgen F, Özturan C. (Springer Berlin Heidelberg, Berlin, Heidelberg), pp. 284–293.
3. Rosvall M, Axelsson D, Bergstrom CT (2009) The map equation. *The European Physical Journal Special Topics* 178(1):13–23.
4. Macropol K, Can T, Singh AK (2009) Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics* 10(1):283.
5. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30(1):107 – 117. Proceedings of the Seventh International World Wide Web Conference.

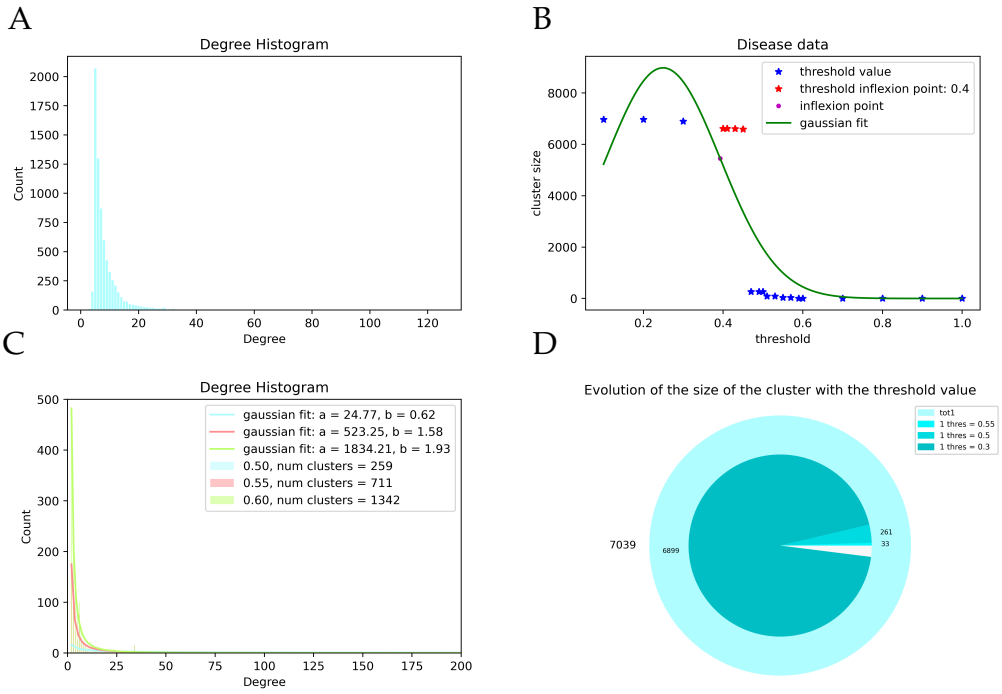


Fig. 3. The figures represent the analyses for the disease monoplex network. A: Degree distribution of the disease monoplex network. B,D: Both charts represent the evolution of the community's final size, depending on the threshold value. B: represents the evolution of the final community size when starting from the seed node UMLS:C0033300, for threshold values from 0.1 to 1.0. The green curve represents the Gaussian fit with its inflection (pink point) that gives the value, for which we observe an abrupt behavior (here associated with a threshold value equal to 0.4). We expect the best threshold value around this inflection point. The red point represents the nearest experimental points. D: represents the evolution of the community size for different threshold values represented by shades of blue. The external part of the pie chart gives the total number of nodes in the disease monoplex network and the inner part gives the number of nodes in the final community, depending on the threshold. C: Distribution of the size of the clusters obtained with the network partitioning algorithm for three different threshold values (0.50, 0.55, 0.60), as well as the power-law fit. The parameters of the fit are given in the label.

6. Pan JY, Yang HJ, Faloutsos C, Duygulu P (2004) Automatic multimedia cross-modal correlation discovery in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04. (Association for Computing Machinery, New York, NY, USA), p. 653–658.
7. Bianconi G (2018) *Multilayer Networks: Structure and Function*. (Oxford University Press, Oxford), p. 416.
8. van Lidth de Jeude JA, Aste T, Caldarelli G (2019) The multilayer structure of corporate networks. *New Journal of Physics* 21(2):025002.
9. Bardoscia M, Bianconi G, Ferrara G (2019) Multiplex network analysis of the uk over-the-counter derivatives market. *International Journal of Finance & Economics* 24(4):1520–1544.
10. Szell M, Lambiotte R, Thurner S (2010) Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences* 107(31):13636–13641.
11. Dickison ME, Magnani M, Rossi L (2016) *Multilayer Social Networks*. (Cambridge University Press).
12. Battiston F, Nicosia V, Latora V (2016) Efficient exploration of multiplex networks. *New Journal of Physics* 18(4):043035.
13. Baggio JA, et al. (2016) Multiplex social ecological network analysis reveals how social changes affect community robustness more than resource depletion. *Proceedings of the National Academy of Sciences*.
14. Stella M, Andreazzi CS, Selakovic S, Goudarzi A, Antonioni A (2016) Parasite spreading in spatial ecological multiplex networks. *Journal of Complex Networks* 5(3):486–511.
15. Pilosof S, Porter MA, Pascual M, Kéfi S (2017) The multilayer nature of ecological networks. *Nature Ecology & Evolution* 1(4):0101.
16. Didier G, Brun C, Baudot A, Gomez S (2015) Identifying communities from multiplex biological networks. *PeerJ* 3:e1525.
17. Valdeolivas A, et al. (2018) Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* 35(3):497–505.
18. Baptista A, Gonzalez A, Baudot A (2022) Universal multilayer network exploration by random walk with restart. *Communications Physics* 5(1):170.
19. Hartwell LH, Hopfield JJ, Leibler S, Murray AW (1999) From molecular to modular cell biology. *Nature* 402(6761):C47–C52.
20. Cardillo A, et al. (2013) Emergence of network features from multiplexity. *Scientific Reports* 3(1):1344.

Ce travail constitue une partie d'un projet plus ambitieux dans le cadre d'une collaboration avec Alberto Valdeolivas, maintenant chercheur à Roche Pharma en Suisse, Ozan Ozisik, chercheur post-doctorant dans mon équipe d'accueil et Anaïs Baudot, ma directrice de thèse. Ce projet a pour objectif d'utiliser des approches de détection de communautés pour étudier les maladies génétiques liées à un phénotype de vieillissement prématuré, via une approche systémique. Mon travail dans le cadre de ce projet correspond à ces extensions des méthodes de détection de communautés et de partitionnement des réseaux aux réseaux multi-couches universels. Les autres collaborateurs du projet s'occupent de la conception du projet dans son ensemble, des aspects bioinformatiques et de l'interprétation des résultats biologiques. Ce travail n'est pas encore soumis, puisqu'il reste des analyses bioinformatiques et des interprétations biologiques à effectuer afin de finaliser l'étude.

9.2. Généralisation de la similarité de Katz aux réseaux multi-couches universels

Nous avons détaillé la similarité de Katz en section 1.2.2 pour les réseaux monoplexes, et en section 2.2.2 pour certains types de réseaux multi-couches, notamment ceux constitués de deux réseaux monoplexes connectés par un réseau bipartite [120–124], ainsi que pour les réseaux multiplexes [125]. La similarité de Katz est couramment utilisée en bioinformatique pour prédire des associations entre maladies et d'autres composantes biologiques comme des gènes [120], des ARNs [121, 123, 124], ou des microbes responsables de maladies non infectieuses [122]. La similarité de Katz est aussi largement utilisée dans le cadre des réseaux sociaux, notamment pour prédire des liens entre nœuds d'un même réseau, qu'il soit monoplex [225] ou multiplex [125]. Cependant, ces extensions de la similarité de Katz aux réseaux multi-couches ne sont pas assez générales pour intégrer des réseaux multi-couches universels, c'est-à-dire des réseaux multi-couches composés d'un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites. Nous rappelons qu'un réseau monoplex est un cas particulier de réseau multiplex composé d'une unique couche. Dans le cadre de ma thèse, j'ai développé une extension de la similarité de Katz aux réseaux multi-couches universels. Ce travail a été réalisé en collaboration avec un étudiant de Master en informatique de l'université d'Aix-Marseille, Loumi Trémas, que j'ai encadré.

9.2.1. Formalisme mathématique de l'extension

Nous rappelons que la similarité de Katz pour un réseau monoplex s'écrit de la

manière suivante :

$$\sigma = (I - \alpha A)^{-1} - I = \sum_{k=1}^{\infty} (\alpha A)^k \\ = \alpha A + \alpha^2 A^2 + \alpha^3 A^3 + \alpha^4 A^4 + \cdots \tag{9.1}$$

avec A la matrice d'adjacence du réseau, $\alpha \in [0,1]$ un paramètre qui contrôle la contribution des termes en fonction de leur ordre dans le développement, et I la matrice identité. Le paramètre α est souvent choisi dans la littérature [120, 124] de la manière suivante :

$$\alpha < \frac{1}{\|A\|_2} \tag{9.2}$$

avec $\|.\|_2$, la norme 2 de la matrice (voir annexe A.2 pour plus de détails).

Nous avons vu dans la section 2.2.2 que, dans le cas de réseaux multi-couches, il est possible de ne pas directement développer dans la similarité de Katz la matrice de transition non normalisée (dans le cas des réseaux monoplexes, il s'agit de la matrice d'adjacence) mais de développer des combinaisons de blocs de cette matrice. Cette stratégie permet de calculer plusieurs produits de matrices de petites tailles au lieu de calculer le produit de matrices de grandes tailles. De plus, cette stratégie permet de voir les contributions de chacun des chemins possibles. Nous allons détailler ce point avec le réseau multi-couche universel représenté dans la Fig. 9.1 gauche (similaire à l'exemple de la section 2.2.2).
On définit la matrice de transition non normalisée, notée S telle que :

$$S = \begin{bmatrix} \mathscr{A}_1 & B_{1,2} \\ B_{2,1} & \mathscr{A}_2 \end{bmatrix} \tag{9.3}$$

On rappelle que, dans le cas d'un réseau dirigé, on définit les éléments de matrice S_{ij} comme représentant l'arête du nœud v_j vers le nœud v_i. Ainsi, la matrice bipartite $B_{i,j}$ définit les transitions du réseau j vers le réseau i (voir Fig. 9.1). Dans l'exemple de la section 2.2.2, tous les réseaux étaient non dirigés, aussi bien les réseaux bipartites que les réseaux monoplexes ou multiplexes. Ceci implique que les combinaisons de matrices permettant de définir la similarité de Katz de j vers i, sont les mêmes que les combinaisons de matrices allant de i vers j, à une transposition du produit de matrices près. En reprenant l'exemple de la section 2.2.2, on considère dorénavant le cas de réseaux dirigés, puisque le cas non dirigé peut être déduit de manière triviale, à partir de cet exemple.

On définit donc la similarité de Katz entre les nœuds du réseau multi-couche universel de la manière suivante :

$$\sigma = \sum_{k=1}^{\infty} (\alpha S)^k = \sum_{k=1}^{\infty} \alpha^k S_k \tag{9.4}$$

avec S_k la combinaison de produits de matrices représentant les chemins de taille k. Par exemple, si on choisit le réseau bleu comme source (réseau 1) et le réseau rouge comme cible (réseau 2) , on obtient les combinaisons de produits matriciels suivants pour les termes S_k :

$$S_1 = B_{2,1} \tag{9.5}$$

$$S_2 = B_{2,1}\mathscr{A}_1 + \mathscr{A}_2 B_{2,1} \tag{9.6}$$

$$S_3 = B_{2,1}\mathscr{A}_1^2 + \mathscr{A}_2^2 B_{2,1} + \mathscr{A}_2 B_{2,1}\mathscr{A}_1 + B_{2,1}B_{1,2}B_{2,1} \tag{9.7}$$

$$\begin{aligned} S_4 = {} & B_{2,1}\mathscr{A}_1^3 + \mathscr{A}_2^2 B_{2,1} + \mathscr{A}_2^2 B_{2,1}\mathscr{A}_1 + \mathscr{A}_2 B_{2,1}\mathscr{A}_1^2 + \mathscr{A}_2 B_{2,1}B_{1,2}B_{2,1} \\ & + B_{2,1}B_{1,2}B_{2,1}\mathscr{A}_1 + B_{2,1}B_{1,2}\mathscr{A}_2 B_{2,1} + B_{2,1}\mathscr{A}_1 B_{1,2}B_{2,1} \end{aligned} \tag{9.8}$$

On s'arrête au quatrième terme puisque les termes d'ordres supérieurs ne contribuent que très faiblement à la série, comme nous le montrerons dans la section suivante. Ainsi, une bonne approximation de la similarité de Katz peut s'écrire à l'aide des équations (9.5-9.8) de la manière suivante :

$$\sigma = \alpha S_1 + \alpha^2 S_2 + \alpha^3 S_3 + \alpha^4 S_4 \tag{9.9}$$

Nous retrouvons le résultat obtenu dans la section 2.2.2, en adoptant la convention suivante : l'élément de la matrice de transition non normalisée dirigée S_{ij} définit l'arête du nœud v_j vers le nœud v_i. Ainsi, on se retrouve avec les transposées des différents produits matriciels de l'équation (2.20). Cependant, on peut constater que le processus est difficile à généraliser pour plus de deux réseaux hétérogènes, puisqu'il n'existe pas de formule permettant de définir l'ensemble des combinaisons d'un nombre arbitraire de réseaux pour une longueur arbitraire de chemins entre un réseau source et un réseau cible. Par conséquent, la généralisation de la similarité de Katz entre un réseau source et un réseau cible passe par la création d'une méthode permettant de dénombrer toutes les combinaisons possibles de longueurs de chemins k entre ces deux réseaux.

On rappelle que la matrice de transition non normalisée d'un réseau multi-couche universel composé de N réseaux multiplexes et de $N*(N-1)$ réseaux bipartites s'écrit de la manière suivante :

$$S = \begin{bmatrix} \mathscr{A}_1 & B_{1,2} & \cdots & B_{1,N} \\ B_{2,1} & \mathscr{A}_2 & \cdots & B_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N,1} & B_{n,2} & \cdots & \mathscr{A}_N \end{bmatrix} \tag{9.10}$$

On note le réseau source s et le réseau cible t, tels que $s,t \in 1,2,..,N$. L'ensemble des combinaisons de chemins entre le réseau s et le réseau t de longueur k est défini dans la Table 9.1.

Cible	Combinaisons			Source	
	11	⋯	11		
	11	⋯	22		
		⋮			
	11	⋯	NN		
	22	⋯	11		
	22	⋯	22		
		⋮			n^{k-1} termes
t	22	⋯	NN	s	
		⋮			
	NN	⋯	11		
	NN	⋯	22		
		⋮			
	NN	⋯	NN		

Table 9.1. – Les variables s et t définissent le numéro des réseaux source et cible. Les combinaisons sont constituées de toutes les suites possibles résultant de $k-1$ tirages avec remises au sein de l'ensemble {11, 22, ..., NN}.

Ainsi, si on considère le cas d'un réseau multi-couche universel constitué de trois réseaux multiplexes et des réseaux bipartites associés (Fig. 9.1, droite), on peut définir l'ensemble des chemins de longueur $k = 3$ associé à la variable S_3. En considérant le réseau rouge comme source (réseau 2) et le réseau vert comme cible (réseau 3). On obtient la table de combinaisons suivante à l'aide de la méthode précédente : Cette

Cible	Combinaisons	Source
	11 11	
	11 22	
	11 33	
	22 11	
3	22 22	2
	22 33	
	33 11	
	33 22	
	33 33	

Table 9.2. – Adaptation de la Table 9.1, dans le cas d'un réseau multi-couche universel constitué de trois réseaux multiplexes (numérotés 1, 2 et 3) et des réseaux bipartites associés. Le réseau source est numéroté 2 et le réseau cible 3, les combinaisons permettent de définir l'ensemble des chemins de longueur $k = 3$.

table qui permet d'obtenir l'ensemble des termes constituant S_3, en reconstituant la suite de termes deux à deux et en leur associant les matrices correspondantes. Ainsi, la première suite est : 31 11 12, ce qui donne le produit matriciel : $B_{3,1}\mathscr{A}_1 B_{1,2}$. L'ensemble des termes constituant S_3 est obtenu par la même lecture de la table de combinaisons, ce qui permet d'obtenir l'équation suivante :

$$\begin{aligned} S_3 =& B_{3,1}\mathscr{A}_1 B_{1,2} + B_{3,1}B_{1,2}\mathscr{A}_2 + B_{3,1}B_{1,3}B_{3,2} \\ &+ B_{3,2}B_{2,1}B_{1,2} + B_{3,2}\mathscr{A}_2^2 + B_{3,2}B_{2,3}B_{3,2} \\ &+ \mathscr{A}_3 B_{3,1}B_{1,2} + \mathscr{A}_3 B_{3,2}\mathscr{A}_2 + \mathscr{A}_3^2 B_{3,2} \end{aligned} \tag{9.11}$$

Cette méthode permet ainsi de déterminer l'ensemble des combinaisons de chemins possibles permettant d'aller d'un réseau source vers un réseau cible, qu'importe le nombre de réseaux constituant le réseau multi-couche universel. Le lecteur intéressé pourra se référer à l'annexe A.3 pour l'obtention de l'équation (9.8) avec cette méthode. Finalement, il est bon de noter qu'il est possible de reconstituer la matrice de similarité de Katz globale en la recomposant par blocs, avec des blocs composés de la similarité de Katz entre un réseau source et un réseau cible. Ainsi, dans le cas du réseau multi-couche universel précédent constitué de trois réseaux multiplexes et des réseaux bipartites associés, la similarité de Katz peut être obtenue à partir du développement de Katz défini à l'équation (9.9) avec la matrice de transition non normalisée, ou bien par blocs de matrices de la manière suivante :

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{9.12}$$

avec chaque σ_{ij} définissant la similarité de Katz entre le réseau source j et le réseau cible i.

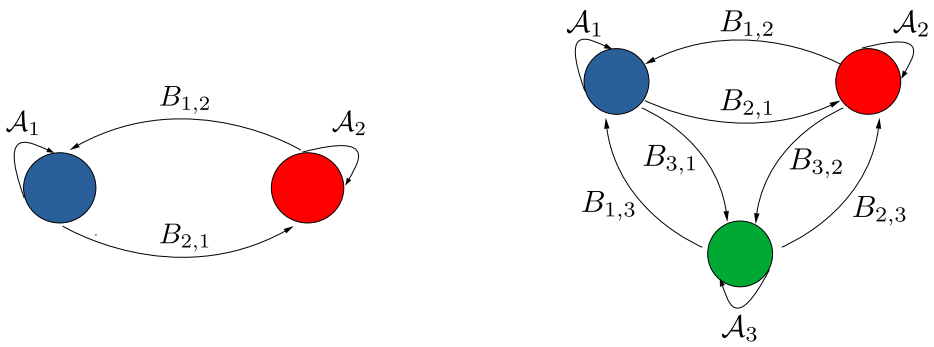


Figure 9.1. – Illustration de réseaux multi-couches universels. À gauche, on a un réseau multi-couche universel constitué de deux réseaux multiplexes (ou monoplexes) représentés par des ronds : bleu pour le premier réseau, rouge pour le second. Les matrices de supra-adjacences $\mathscr{A}_i$ (ou d'adjacences pour les réseaux monoplexes) de chacun des réseaux sont représentées par la flèche allant du nœud vers le nœud lui-même. Les matrices bipartites $B_{i,j}$ (et les réseaux bipartites associés) entre les deux réseaux multiplexes (ou monoplexes) rouge et bleu sont représentées par les flèches allant de l'un vers l'autre des réseaux. À droite, on a un réseau multi-couche universel constitué de trois réseaux multiplexes (ou monoplexes) représentés par des ronds : bleu pour le premier réseau, rouge pour le second, vert pour le troisième. Les matrices de supra-adjacences (ou d'adjacences), ainsi que les matrices bipartites sont représentées de la même manière que dans la figure de gauche.

9.2.2. Résultats préliminaires

À partir de la méthode définie dans la section dédiée au formalisme mathématique, nous avons pu obtenir des résultats préliminaires sur un réseau multi-couche universel de taille réduite (réseau multi-couche universel d'aéroports) et un réseau multi-couche universel à grande échelle (réseau biologique multi-couche universel). Ces deux réseaux ont été détaillés dans le chapitre 6. Les deux figures suivantes ont été générées à partir du réseau multi-couche universel d'aéroports.

Dans un premier temps, on calcule la similarité de Katz à partir de la méthode exacte, c'est-à-dire de la première égalité de l'équation (9.1) (définition purement matricielle). Pour chaque valeur de $\alpha \in [0.1, 1]$ avec un pas de 0.1, on calcule la similarité de Katz associée au réseau, puis on définit la matrice de corrélation (Fig. 9.2) qui représente la corrélation de Spearman moyenne. Cette corrélation est calculée entre deux matrices représentant la similarité de Katz obtenue pour une valeur spécifique de α. Chaque élément de la matrice de corrélation est calculé à partir de la moyenne de toutes les corrélations de Spearman, calculée entre chaque vecteur des deux matrices corre-

spondantes. À partir de cette matrice de corrélation, on observe que la similarité de Katz est peu affectée par le changement de valeur de α (Fig. 9.2).

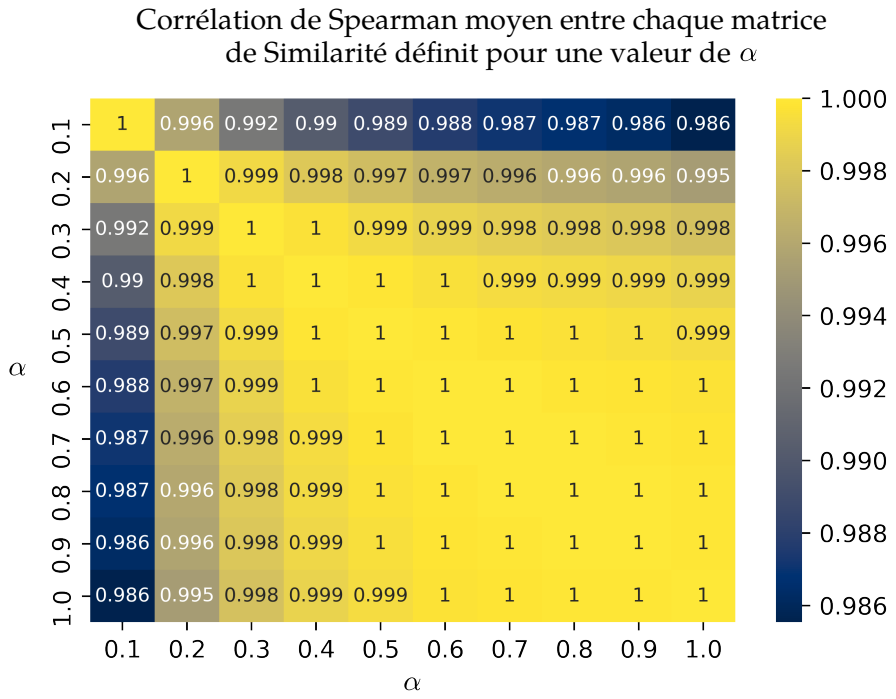


Figure 9.2. – Matrice de corrélation de Spearman moyenne. Elle est calculée entre deux matrices représentant la similarité de Katz obtenue pour une valeur spécifique de α. Chaque élément de la matrice de corrélation est calculé à partir de la moyenne de toutes les corrélations de Spearman, calculée entre chaque vecteur des deux matrices correspondantes. La barre de couleur indique l'intensité de la corrélation, très forte en jaune (proche de 1), moins forte en bleu foncé.

Dans un second temps, on fixe la valeur de α, puis on calcule la similarité de Katz pour différentes valeurs de troncature de la série (ordre du dernier terme de la série). On définit la longueur de convergence de la série comme étant l'ordre au bout duquel il n'est plus nécessaire de pousser le développement plus loin pour avoir toute la mesure de similarité. On suppose que le critère de convergence est réalisé lorsque la corrélation de Spearman moyenne (définie plus haut) entre deux matrices de similarité de Katz issues de deux troncatures successives est strictement suftpérieure ou égale à 99 % (matrice de corrélation de la Fig.9.3). La corrélation de Spearman moyenne est calculée pour différentes valeurs successives de α (Fig.9.3). On observe que la série associée à la similarité de Katz converge rapidement et qu'il suffit de développer jusqu'au quatrième ou cinquième terme pour obtenir toutes les contributions. Autrement dit, comme le terme d'ordre k de la série correspond à l'ensemble des chemins en k étapes séparant le nœud source et le nœud cible, la

similarité de Katz entre deux nœuds est définie par l'ensemble des chemins d'une longueur égale à quatre ou cinq arêtes séparant les deux au sein du réseau (Fig. 9.3). Ce résultat avait déjà été obtenu dans de précédentes études [120, 124]. Une deuxième observation montre que la vitesse de convergence de la méthode varie cependant en fonction de la valeur de α. Les valeurs de α faibles amènent la série à converger plus vite, à partir du quatrième terme, tandis que, pour de grandes valeurs de α, il est nécessaire de développer la série jusqu'au cinquième terme (Fig. 9.3). On voit que la série converge plus rapidement lorsque l'on se restreint à une valeur de α satisfaisant la condition de l'équation (9.2).

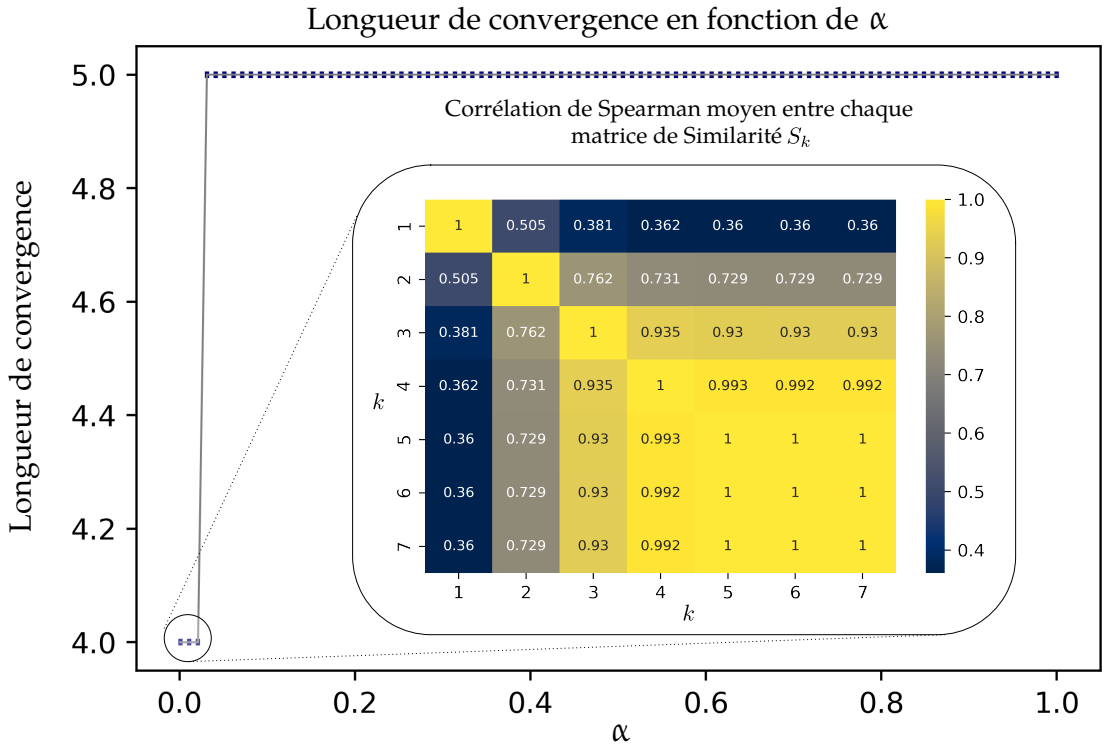


Figure 9.3. – La courbe représente la longueur de convergence (ordre de troncature k de la série), en fonction de la valeur de α. La valeur de convergence est obtenue quand la corrélation de Spearman, moyenne entre deux matrices de similarité de Katz issues de deux troncatures successives, est strictement supérieure ou égale à 99 %. Le zoom correspond à une matrice de corrélation de Spearman moyenne obtenue pour une faible valeur de α, valeur associée à une longueur de convergence égale à quatre. Chaque élément de la matrice est calculé à partir de la corrélation de Spearman moyenne entre deux matrices de similarité de Katz, issues de deux troncatures de valeur k. La barre de couleur indique l'intensité de la corrélation, très forte en jaune (près de 1), moins forte en bleu foncé.

9.2.3. Perspectives et discussion

Nous avons donc étendu la similarité de Katz aux réseaux multi-couches universels. Cette extension ouvre la porte à toutes les méthodes s'appuyant sur la similarité de Katz pour tirer parti des réseaux multi-couches universels. Par exemple, la prédiction d'associations entre nœuds de différents types, ou la prédiction de liens entre nœuds du même type. À titre d'exemple, une méthode de prédiction d'associations entre gènes et maladies basée sur la similarité de Katz existe déjà [120]. Cependant, cette méthode n'utilise que l'information sur des réseaux de gènes, de maladies et leurs réseaux bipartites pour calculer la similarité entre les nœuds. Avec notre méthode, il est dorénavant possible d'ajouter autant de réseaux que l'on souhaite, et ainsi de prendre en compte des informations complémentaires pour calculer la similarité entre les nœuds de gènes et de maladies. Cependant, il est bon de noter qu'ajouter des réseaux au sein d'un réseau multi-couche, sans prendre en considération les réseaux bipartites permettant de les connecter, peut ne pas améliorer les prédictions [210]. Enfin, il est possible de s'intéresser à la prédiction d'associations entre médicaments et cibles (protéines ou gènes), ce qui peut être utile dans le cadre du repositionnement de médicaments.

Il existe d'autres points à explorer à l'aide de la nouvelle mesure de similarité de Katz sur les réseaux multi-couches universels. Nous pouvons notamment mentionner la comparaison avec l'autre mesure de similarité définie dans le cadre de ma thèse, c'est-à-dire avec la similarité basée sur les marches aléatoires. Ils serait particulièrement intéressant de comparer ces deux méthodes sur des validations croisées et de prédictions de liens, comme nous l'avons fait dans le cadre du premier article avec différents réseaux multi-couches universels. De plus, comme nous le verrons dans la prochaine section, toute méthode qui utilise des mesures de similarité comme élément de base peut utiliser cette nouvelle mesure de similarité de Katz afin d'explorer des réseaux multi-couches universels. Cela est notamment le cas pour certaines méthodes d'*embedding*, comme la méthode *VERSE* détaillée en section 1.3.2.
Finalement, il est possible d'associer à la similarité de Katz des méthodes supervisées, comme des méthodes de classification supervisée. Cette association permettrait de construire une méthode de prédiction supervisée qui serait entraînée à l'aide des mesures de similarité entre les nœuds définies à partir de la similarité de Katz.

9.3. *Embedding* de réseaux à l'aide de MultiXrank

Dans la section 1.3.2, nous avons introduit une méthode d'*embedding* sur réseaux nommée *VERSE* (*VERtex Similarity Embeddings*) [81]. Cette méthode présente plusieurs avantages. Elle est versatile, c'est-à-dire qu'elle peut utiliser différentes mesures de

similarité en entrée, y compris une similarité basée sur les marches aléatoires avec *restart*. Elle est aussi adaptée aux réseaux à grande échelle grâce à l'utilisation de matrices creuses (*sparse*), et d'une stratégie d'échantillonnage négatif (*negative sampling*) qui permet d'optimiser la fonction de perte plus rapidement. Nous rappelons que la méthode *VERSE* utilise comme fonction de similarité de paire dans l'espace direct n'importe quelle mesure de similarité. Nous rappelons également que la fonction de similarité dans l'espace d'*embedding*, aussi appelé espace latent (fonction *decoder*), est une fonction softmax normalisée et que la fonction de perte utilisée est la divergence de Kullback-Leibler. De plus, l'encodage des nœuds de l'espace direct vers l'espace d'*embedding* est obtenu en optimisant la fonction de perte à l'aide de la méthode de descente de gradient. Les vecteurs de la représentation matricielle de l'*embedding* du réseau sont initialement distribués selon une distribution centrée en zéro.
Comme nous venons de le décrire, la méthode d'*embedding VERSE* accepte une grande variété de mesures de similarité de paire, notamment une mesure de similarité provenant des marches aléatoires avec *restart*. Par conséquent, il est possible, sans grande difficulté, de définir l'*embedding* d'un réseau multi-couche universel, si nous pouvons définir une mesure de similarité entre les nœuds de ce réseau multi-couche universel. Il est possible de définir une telle mesure à l'aide de MultiXrank, qui est capable de calculer une similarité entre tous les nœuds du réseau et un nœud de référence (la graine). Ainsi, nous pouvons calculer pour chaque nœud, considéré tour à tour comme graine, la mesure de similarité entre ce nœud et les autres nœuds du réseau. Les mesures associées à chaque nœud "graine" formeront les vecteurs de la matrice de similarité entre les différents nœuds du réseau. Il est important de remarquer qu'une mesure de similarité obtenue à partir des marches aléatoires avec *restart* possède deux avantages majeurs : ces similarités sont non linéaires et asymétriques. La non-linéarité permet d'obtenir une mesure capturant une géométrie plus complexe et l'asymétrie permet de capturer la nature asymétrique des relations entre nœuds d'un réseau [226]. En d'autres termes, aller du nœud v_i au nœud v_j peut être différent, en termes de probabilité, que d'aller du nœud v_j au nœud v_i, sachant que les voisinages de ces deux nœuds sont différents. Ainsi, cette asymétrie se définit telle que la similarité s_{ij} définie entre le nœud v_i et le nœud v_j est différente de la similarité s_{ji} entre le nœud v_j et le nœud v_i.
En m'appuyant sur MultiXrank, j'ai initié le développement d'un nouvel outil d'*embedding* de réseaux s'inspirant de la méthode *VERSE*. Cette démarche a été aussi à la base d'une précédente étude au sein de mon équipe d'accueil [129]. Cependant, cette méthode initiale ne permet l'*embedding* que de réseaux multi-couches possédant aux plus deux réseaux hétérogènes. Notre méthode peut, elle, intégrer un nombre arbitraire de réseaux multiplexes connectés entre eux par des réseaux bipartites. Néanmoins, le point central de mon travail en cours de réalisation concerne, en plus de cette nouvelle méthode d'*embedding* de réseaux multi-couches universels, l'étude de l'apport de l'*embedding* par rapport aux méthodes directes. En effet, l'*embedding* peut être une méthode puissante pour les prédictions, notamment dans le cas de la détection

de communautés. Cependant, sur certaines tâches, l'*embedding* ne présente pas de meilleures performances que les méthodes directes [227]. De plus, les méthodes d'*embedding* sont souvent des boîtes noires et l'interprétation des résultats est très difficile. Développer une méthode systématique permettant de comparer les performances d'approches dans l'espace direct et dans l'espace d'*embedding* est ainsi fondamental, notamment dans le contexte où ces méthodes deviennent de plus en plus utilisées. L'article de Zhang et al. [228] présente trois différentes tâches permettant de déterminer la qualité d'une méthode d'*embedding* : la prédiction de liens, la correspondance entre les similarités entre les nœuds dans l'espace d'*embedding* et la similarité entre les nœuds dans l'espace direct (*mapping accuracy*), la navigabilité dans l'espace d'*embedding* (*greedy routing*), c'est-à-dire dans quelle mesure il est possible d'utiliser l'*embedding* pour déterminer le plus court chemin entre deux nœuds du réseau. Je propose d'utiliser ces différentes tâches pour comparer la performance des approches entre la méthode dans l'espace direct et la méthode dans l'espace d'*embedding*. Cela permettra de trancher sur le gain qu'apportent les méthodes d'*embedding*, par rapport à la perte d'interprétabilité qu'elles génèrent, et cela, en fonction des différentes tâches.
De plus, sachant que la méthode d'*embedding* basée sur *VERSE* est versatile, il est donc possible d'utiliser différentes mesures de similarité dans l'espace direct. Ainsi, il sera possible de comparer les performances de la méthode d'*embedding* sur les différentes tâches décrites au-dessus en utilisant les deux méthodes de similarité sur les réseaux multi-couches universels que j'ai développés au cours de ma thèse. C'est-à-dire d'un côté, la similarité de Katz développée en section 9.2, et de l'autre côté, la similarité basée sur les marches aléatoires avec *restart* développée dans le chapitre 6 (qui s'appuie sur MultiXrank).

Conclusion

Mon travail de thèse s'inscrit dans une double filiation : d'un côté la communauté de la théorie des graphes, et de l'autre côté la communauté des réseaux biologiques associée à la biologie des systèmes. Le développement de méthodes "réseaux" permet à la communauté des réseaux biologiques d'intégrer et d'explorer les réseaux biologiques et les réseaux biologiques fournissent des données particulièrement riches pour la communauté de la théorie des graphes. Ce dialogue entre les deux communautés a été au cœur de ma thèse, et cela m'a permis de contribuer aux deux communautés.
Dans un premier temps, j'ai développé un nouveau formalisme et de nouveaux algorithmes dédiés aux réseaux multi-couches universels, avec notamment l'extension des marches aléatoires avec *restart* et de la similarité de Katz aux réseaux multi-couches universels, ainsi que le développement d'une méthode de détection de communautés basée sur les réseaux multi-couches universels. Ma contribution à la communauté de la théorie des graphes a aussi été menée à travers l'étude approfondie du domaine de l'*embedding* de réseaux, cette étude a donné lieu à une revue de la littérature qui a pour but de définir une taxonomie transdisciplinaire des méthodes d'*embedding* de réseaux.
Dans un second temps, j'ai appliqué les méthodes que j'ai développé aux réseaux biologiques. J'ai, en effet, appliqué les marches aléatoires avec *restart* à des réseaux multi-couches biologiques. Cela m'a permis de proposer de nouvelles méthodes de priorisation de médicaments ou de prédiction d'associations entre gènes et maladies. J'ai aussi développé une nouvelle manière d'intégrer des informations génomiques grâce à des réseaux multi-couches, avec pour objectif de détecter des comorbidités. L'ensemble de ces méthodes est accessible à la communauté de la biologie des systèmes, et est adaptable à n'importe quelles associations de réseaux biologiques.
Mon travail a également eu pour objectif de mieux comprendre le rapport signal sur bruit dans les réseaux complexes, une question majeure dans la communauté des réseaux. Dans mon article *Universal Multilayer Exploration by Random Walk with Restart*, je propose des pistes de réflexion et des outils pour évaluer l'intérêt de l'ajout de couches d'interactions à des systèmes multi-couches. En plus de fournir quelques réponses à cette problématique, j'ai amené des pistes de réflexion sur l'importance des réseaux bipartites dans l'exploration des réseaux multi-couches.
Finalement, de nombreuses pistes de recherches restent encore à explorer. Sur le sujet de l'*embedding* de réseaux, il reste important de définir des critères permettant de juger de la qualité d'un *embedding* de réseaux, ainsi que de définir dans quelle mesure les méthodes d'*embedding* d'un réseau sont plus efficaces que sur les méthodes utilisant le réseau directement. Je me suis également intéressé aux méthodes

de percolation et de *cascading failure*, notamment dans le cadre de la détection de communautés. Définir une méthode de détection de communautés basée sur une mesure de similarité régulière, mesure de similarité régulière qui s'appuie sur la robustesse du réseau calculé depuis un processus de *cascading failure*, est une piste de recherche inédite. J'aimerais continuer à développer cette piste au-delà de ma thèse afin de continuer à contribuer à la communauté des réseaux.

Bibliographie

[1] Anthony Baptista and Aurélien Perera. “Modeling micro-heterogeneity in mixtures: The role of many body correlations”. In: *The Journal of Chemical Physics* 150.6 (2019), p. 064504. DOI: 10.1063/1.5066598. eprint: https://doi.org/10.1063/1.5066598. URL: https://doi.org/10.1063/1.5066598 (cit. on p. 18).

[2] László Almásy et al. “Microscopic origin of the scattering pre-peak in aqueous propylamine mixtures: X-ray and neutron experiments versus simulations”. In: *Phys. Chem. Chem. Phys.* 21 (18 2019), pp. 9317–9325. DOI: 10.1039/C9CP01137D. URL: http://dx.doi.org/10.1039/C9CP01137D (cit. on p. 18).

[3] Leonhard Euler. “Solutio problematis ad geometriam situs pertinentis”. In: *Commentarii Academiae Scientiarum Petropolitanae, Volume 8, pp. 128-140.* (1741) (cit. on p. 20).

[4] Carl Hierholzer. “Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren”. In: *Mathematische Annalen VI, 30–32.* (1873) (cit. on p. 20).

[5] Édouard Lucas. “Récréations mathématique (2ème éd.)” In: *Librairie Scientifique et Technique Albert Blanchard* (1891) (cit. on p. 20).

[6] Ernesto Estrada. *Graph and Network Theory in Physics*. 2013. DOI: 10.48550/ARXIV.1302.4378. URL: https://arxiv.org/abs/1302.4378 (cit. on p. 21).

[7] Nenad Trinajstic. “Chemical graph theory”. In: *CRC Press* (1983) (cit. on p. 21).

[8] Narsingh Deo. *Graph Theory with Applications to Engineering and Computer Science (Prentice Hall Series in Automatic Computation)*. USA: Prentice-Hall, Inc., 1974. ISBN: 0133634736 (cit. on p. 21).

[9] O. Mason and M. Verwoerd. “Graph theory and networks in Biology.” eng. In: *IET systems biology* 1 (2 Mar. 2007), pp. 89–119 (cit. on p. 21).

[10] Jordi Bascompte. “Networks in ecology”. In: *Basic and Applied Ecology* 8.6 (2007), pp. 485–490. ISSN: 1439-1791. DOI: https://doi.org/10.1016/j.baae.2007.06.003. URL: https://www.sciencedirect.com/science/article/pii/S1439179107000576 (cit. on p. 21).

[11] Jonathan D. Phillips et al. “Graph theory in the geosciences”. In: *Earth-Science Reviews* 143 (2015), pp. 147–160. ISSN: 0012-8252. DOI: https://doi.org/10.1016/j.earscirev.2015.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0012825215000239 (cit. on p. 21).

[12] Fernando Vega-Redondo. *Complex Social Networks*. Econometric Society Monographs. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511804052 (cit. on p. 21).

[13] Stephen P. Borgatti et al. "Network Analysis in the Social Sciences". In: *Science* 323 (2009), pp. 892–895 (cit. on p. 21).

[14] Frank Schweitzer et al. "Economic Networks: The New Challenges". In: *Science* 325.5939 (2009), pp. 422–425. DOI: 10.1126/science.1173644. eprint: https://www.science.org/doi/pdf/10.1126/science.1173644. URL: https://www.science.org/doi/abs/10.1126/science.1173644 (cit. on p. 21).

[15] Katja Filippova. "Multi-Sentence Compression: Finding Shortest Paths in Word Graphs". In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 322–330. URL: https://aclanthology.org/C10-1037 (cit. on p. 21).

[16] Alexei Vazquez et al. "Global protein function prediction from protein-protein interaction networks". In: *Nature Biotechnology* 21.6 (2003), pp. 697–700. ISSN: 1546-1696. DOI: 10.1038/nbt825. URL: https://doi.org/10.1038/nbt825 (cit. on p. 21).

[17] Koji Tsuda et al. "Fast protein classification with multiple networks". In: *Bioinformatics* 21 (Sept. 2005), pp. ii59–ii65. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti1110. eprint: https://academic.oup.com/bioinformatics/article-pdf/21/suppl_2/ii59/6685982/bti1110.pdf. URL: https://doi.org/10.1093/bioinformatics/bti1110 (cit. on p. 21).

[18] Sinan Aral and Christos Nicolaides. "Exercise contagion in a global social network". In: *Nature Communications* 8.1 (2017), p. 14753. ISSN: 2041-1723. DOI: 10.1038/ncomms14753. URL: https://doi.org/10.1038/ncomms14753 (cit. on p. 21).

[19] Nicholas A. Christakis and James H. Fowler. "The Collective Dynamics of Smoking in a Large Social Network". In: *New England Journal of Medicine* 358.21 (2008). PMID: 18499567, pp. 2249–2258. DOI: 10.1056/NEJMsa0706154. eprint: https://doi.org/10.1056/NEJMsa0706154. URL: https://doi.org/10.1056/NEJMsa0706154 (cit. on p. 21).

[20] Dan Braha and Marcus A. M. de Aguiar. "Voting contagion: Modeling and analysis of a century of U.S. presidential elections". In: *PLOS ONE* 12.5 (May 2017), pp. 1–30. DOI: 10.1371/journal.pone.0177970. URL: https://doi.org/10.1371/journal.pone.0177970 (cit. on p. 21).

[21] Anna D. Broido and Aaron Clauset. "Scale-free networks are rare". In: *Nature Communications* 10.1 (2019), p. 1017. ISSN: 2041-1723. DOI: 10.1038/s41467-019-08746-5. URL: https://doi.org/10.1038/s41467-019-08746-5 (cit. on p. 27).

[22] Marcus Berliant and Axel H. Watanabe. “A scale-free transportation network explains the city-size distribution”. eng. In: *Quantitative Economics* 9.3 (2018), pp. 1419–1451. ISSN: 1759-7331. DOI: 10.3982/QE619. URL: http://hdl.handle.net/10419/217132 (cit. on p. 27).

[23] David Lane and Arnold Levine. “p53 Research: the past thirty years and the next thirty years.” eng. In: *Cold Spring Harbor perspectives in biology* 2 (12 Dec. 2010), a000893 (cit. on p. 27).

[24] Leland H. Hartwell et al. “From molecular to modular cell biology”. In: *Nature* 402.6761 (1999), pp. C47–C52. ISSN: 1476-4687. DOI: 10.1038/35011540. URL: https://doi.org/10.1038/35011540 (cit. on pp. 28, 42).

[25] Stanley Wasserman, Katherine Faust, et al. “Social network analysis: Methods and applications”. In: (1994) (cit. on p. 29).

[26] Xuanyao Liu et al. “Trans Effects on Gene Expression Can Drive Omnigenic Inheritance”. In: *Cell* 177.4 (May 2019), 1022–1034.e6. ISSN: 0092-8674. DOI: 10.1016/j.cell.2019.04.014. URL: https://doi.org/10.1016/j.cell.2019.04.014 (cit. on p. 29).

[27] H. Jeong et al. “Lethality and centrality in protein networks”. In: *Nature* 411.6833 (2001), pp. 41–42. ISSN: 1476-4687. DOI: 10.1038/35075138. URL: https://doi.org/10.1038/35075138 (cit. on p. 30).

[28] Xionglei He and Jianzhi Zhang. “Why Do Hubs Tend to Be Essential in Protein Networks?” In: *PLOS Genetics* 2.6 (June 2006), pp. 1–9. DOI: 10.1371/journal.pgen.0020088. URL: https://doi.org/10.1371/journal.pgen.0020088 (cit. on p. 30).

[29] Stefan Wuchty and Eivind Almaas. “Peeling the yeast protein network”. In: *PROTEOMICS* 5.2 (2005), pp. 444–449. DOI: https://doi.org/10.1002/pmic.200400962. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/pmic.200400962. URL: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/pmic.200400962 (cit. on p. 30).

[30] Réka Albert et al. “Error and attack tolerance of complex networks”. In: *Nature* 406.6794 (2000), pp. 378–382. ISSN: 1476-4687. DOI: 10.1038/35019019. URL: https://doi.org/10.1038/35019019 (cit. on p. 30).

[31] Jing-Dong J. Han et al. “Evidence for dynamically organized modularity in the yeast protein-protein interaction network”. In: *Nature* 430.6995 (2004), pp. 88–93. ISSN: 1476-4687. DOI: 10.1038/nature02555. URL: https://doi.org/10.1038/nature02555 (cit. on p. 30).

[32] Tao Zhou et al. “Predicting missing links via local information”. In: *The European Physical Journal B* 71.4 (2009), pp. 623–630. ISSN: 1434-6036. DOI: 10.1140/epjb/e2009-00335-8. URL: https://doi.org/10.1140/epjb/e2009-00335-8 (cit. on p. 30).

[33] Yu Chen et al. “Protein Interface Complementarity and Gene Duplication Improve Link Prediction of Protein-Protein Interaction Network”. In: *Frontiers in Genetics* 11 (2020). ISSN: 1664-8021. DOI: 10.3389/fgene.2020.00291. URL: https://www.frontiersin.org/article/10.3389/fgene.2020.00291 (cit. on p. 31).

[34] Benno Schwikowski et al. “A network of protein-protein interactions in yeast”. In: *Nature Biotechnology* 18.12 (2000), pp. 1257–1261. ISSN: 1546-1696. DOI: 10.1038/82360. URL: https://doi.org/10.1038/82360 (cit. on p. 31).

[35] Phillip Bonacich. “Power and Centrality: A Family of Measures”. In: *American Journal of Sociology* 92.5 (1987), pp. 1170–1182. DOI: 10.1086/228631. eprint: https://doi.org/10.1086/228631. URL: https://doi.org/10.1086/228631 (cit. on p. 32).

[36] Leo Katz. “A new status index derived from sociometric analysis”. In: *Psychometrika* 18.1 (1953), pp. 39–43. ISSN: 1860-0980. DOI: 10.1007/BF02289026. URL: https://doi.org/10.1007/BF02289026 (cit. on pp. 32, 37).

[37] Jing Zhao et al. “Ranking Candidate Disease Genes from Gene Expression and Protein Interaction: A Katz-Centrality Based Approach”. In: *PLOS ONE* 6.9 (Sept. 2011), pp. 1–9. DOI: 10.1371/journal.pone.0024306. URL: https://doi.org/10.1371/journal.pone.0024306 (cit. on p. 32).

[38] Sergey Brin and Lawrence Page. “The anatomy of a large-scale hypertextual Web search engine”. In: *Computer Networks and ISDN Systems* 30.1 (1998). Proceedings of the Seventh International World Wide Web Conference, pp. 107–117. ISSN: 0169-7552. DOI: https://doi.org/10.1016/S0169-7552(98)00110-X. URL: https://www.sciencedirect.com/science/article/pii/S016975529800110X (cit. on pp. 32, 67).

[39] Stephen P. Borgatti and Martin G. Everett. “A Graph-theoretic perspective on centrality”. In: *Social Networks* 28.4 (2006), pp. 466–484. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2005.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0378873305000833 (cit. on p. 33).

[40] Kousik Das et al. “Study on centrality measures in social networks: a survey”. In: *Social Network Analysis and Mining* 8.1 (2018), p. 13. ISSN: 1869-5469. DOI: 10.1007/s13278-018-0493-2. URL: https://doi.org/10.1007/s13278-018-0493-2 (cit. on p. 33).

[41] David F. Gleich. “PageRank beyond the Web”. In: *SIAM Rev.* 57 (2015), pp. 321–363 (cit. on p. 33).

[42] Mark Newman. *Networks: An Introduction*. eng. Oxford: Oxford University Press, 2010, p. 784. DOI: 10.1093/acprof:oso/9780199206650.001.0001. URL: https://oxford.universitypressscholarship.com/10.1093/acprof:oso/9780199206650.001.0001/acprof-9780199206650 (cit. on p. 33).

[43] Alex Bavelas. "Communication Patterns in Task-Oriented Groups". In: *The Journal of the Acoustical Society of America* 22.6 (1950), pp. 725–730. DOI: 10.1121/1.1906679. eprint: https://doi.org/10.1121/1.1906679. URL: https://doi.org/10.1121/1.1906679 (cit. on p. 33).

[44] G. Sabidussi. "The centrality of a graph." eng. In: *Psychometrika* 31 (4 Dec. 1966), pp. 581–603 (cit. on p. 33).

[45] Tien-Dzung Tran and Yung-Keun Kwon. "Hierarchical Closeness Efficiently Predicts Disease Genes in a Directed Signaling Network". In: *Comput. Biol. Chem.* 53.PB (Dec. 2014), pp. 191–197. ISSN: 1476-9271. DOI: 10.1016/j.compbiolchem.2014.08.023. URL: https://doi.org/10.1016/j.compbiolchem.2014.08.023 (cit. on p. 34).

[46] Linton C. Freeman. "A Set of Measures of Centrality Based on Betweenness". In: *Sociometry* 40.1 (1977), pp. 35–41. ISSN: 00380431. URL: http://www.jstor.org/stable/3033543 (visited on 05/22/2022) (cit. on p. 34).

[47] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Phys. Rev. E* 69 (2 Feb. 2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113 (cit. on pp. 34, 42).

[48] Jeongah Yoon et al. "An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality". In: *Bioinformatics* 22.24 (Oct. 2006), pp. 3106–3108. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btl533. eprint: https://academic.oup.com/bioinformatics/article-pdf/22/24/3106/16852098/btl533.pdf. URL: https://doi.org/10.1093/bioinformatics/btl533 (cit. on p. 34).

[49] Paul Jaccard. "THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1". In: *New Phytologist* 11.2 (1912), pp. 37–50. DOI: https://doi.org/10.1111/j.1469-8137.1912.tb05611.x. eprint: https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.1912.tb05611.x. URL: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x (cit. on p. 35).

[50] Gerard Salton. "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer". In: 1989 (cit. on p. 35).

[51] Karl Pearson and Francis Galton. “VII. Note on regression and inheritance in the case of two parents”. In: *Proceedings of the Royal Society of London* 58.347-352 (1895), pp. 240–242. DOI: 10.1098/rspl.1895.0041. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspl.1895.0041. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspl.1895.0041 (cit. on p. 36).

[52] Juan I. Fuxman Bass et al. “Using networks to measure similarity between genes: association index selection”. In: *Nature Methods* 10.12 (2013), pp. 1169–1176. ISSN: 1548-7105. DOI: 10.1038/nmeth.2728. URL: https://doi.org/10.1038/nmeth.2728 (cit. on p. 36).

[53] Vincent D. Blondel et al. “A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching”. In: *SIAM Rev.* 46 (2004), pp. 647–666 (cit. on p. 36).

[54] Glen Jeh and Jennifer Widom. “SimRank: A Measure of Structural-Context Similarity”. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 538–543. ISBN: 158113567X. DOI: 10.1145/775047.775126. URL: https://doi.org/10.1145/775047.775126 (cit. on pp. 36, 49).

[55] E. A. Leicht et al. “Vertex similarity in networks”. In: *Phys. Rev. E* 73 (2 Feb. 2006), p. 026120. DOI: 10.1103/PhysRevE.73.026120. URL: https://link.aps.org/doi/10.1103/PhysRevE.73.026120 (cit. on p. 36).

[56] David Liben-Nowell and Jon Kleinberg. “The Link Prediction Problem for Social Networks”. In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. CIKM ’03. New Orleans, LA, USA: Association for Computing Machinery, 2003, pp. 556–559. ISBN: 1581137230. DOI: 10.1145/956863.956972. URL: https://doi.org/10.1145/956863.956972 (cit. on pp. 37, 38).

[57] Duncan J. Watts and Steven H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684 (1998), pp. 440–442. ISSN: 1476-4687. DOI: 10.1038/30918. URL: https://doi.org/10.1038/30918 (cit. on p. 39).

[58] Stanley Milgram. “The Small-World Problem”. In: *Psychology Today* 1.1 (1967), pp. 61–67 (cit. on p. 40).

[59] Jeffrey Travers and Stanley Milgram. “An Experimental Study of the Small World Problem”. In: *SOCIOMETRY* 32.4 (1969), pp. 425–443 (cit. on p. 40).

[60] Eman Yasser Daraghmi and Yuan Shyan Ming. “Using Graph Theory to Re-Verify the Small World Theory in an Online Social Network Word”. In: *Proceedings of the 14th International Conference on Information Integration and Web-Based Applications and Services*. IIWAS ’12. Bali, Indonesia: Association for Computing Machinery, 2012, pp. 407–410. ISBN: 9781450313063. DOI: 10.1145/

2428736.2428811. URL: https://doi.org/10.1145/2428736.2428811 (cit. on p. 40).

[61] Antonio del Sol et al. "Topology of small-world networks of protein–protein complex structures". In: *Bioinformatics* 21.8 (Jan. 2005), pp. 1311–1315. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti167. eprint: https://academic.oup.com/bioinformatics/article-pdf/21/8/1311/691592/bti167.pdf. URL: https://doi.org/10.1093/bioinformatics/bti167 (cit. on p. 40).

[62] L. A. N. Amaral et al. "Classes of small-world networks". In: *Proceedings of the National Academy of Sciences* 97.21 (2000), pp. 11149–11152. DOI: 10.1073/pnas.200327197. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.200327197. URL: https://www.pnas.org/doi/abs/10.1073/pnas.200327197 (cit. on p. 40).

[63] Mason A Porter et al. "Communities in networks". In: *Notices of the AMS* 56.9 (2009), pp. 1082–1097 (cit. on p. 41).

[64] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3 (2010), pp. 75–174. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2009.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0370157309002841 (cit. on p. 41).

[65] Wayne W. Zachary. "An Information Flow Model for Conflict and Fission in Small Groups". In: *Journal of Anthropological Research* 33.4 (1977), pp. 452–473. DOI: 10.1086/jar.33.4.3629752. eprint: https://doi.org/10.1086/jar.33.4.3629752. URL: https://doi.org/10.1086/jar.33.4.3629752 (cit. on p. 41).

[66] M. Girvan and M. E. J. Newman. "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.122653799. URL: https://www.pnas.org/doi/abs/10.1073/pnas.122653799 (cit. on p. 41).

[67] Sarvenaz Choobdar et al. "Assessment of network module identification across complex diseases". In: *Nature Methods* 16.9 (2019), pp. 843–852. ISSN: 1548-7105. URL: https://doi.org/10.1038/s41592-019-0509-5 (cit. on pp. 42, 62).

[68] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. DOI: 10.1088/1742-5468/2008/10/p10008. URL: https://doi.org/10.1088/1742-5468/2008/10/p10008 (cit. on pp. 42, 44).

[69] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489 (cit. on p. 42).

[70] M. E. J. Newman. “Analysis of weighted networks”. In: *Phys. Rev. E* 70 (5 Nov. 2004), p. 056131. DOI: 10.1103/PhysRevE.70.056131. URL: https://link.aps.org/doi/10.1103/PhysRevE.70.056131 (cit. on p. 42).

[71] Aaron Clauset et al. “Finding community structure in very large networks”. In: *Phys. Rev. E* 70 (6 Dec. 2004), p. 066111. DOI: 10.1103/PhysRevE.70.066111. URL: https://link.aps.org/doi/10.1103/PhysRevE.70.066111 (cit. on p. 42).

[72] M. E. J. Newman. “Modularity and community structure in networks”. In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582. DOI: 10.1073/pnas.0601602103. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0601602103. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0601602103 (cit. on p. 42).

[73] Giuseppe Di Battista et al. *Graph Drawing: Algorithms for the Visualization of Graphs*. 1st. USA: Prentice Hall PTR, 1998. ISBN: 0133016153 (cit. on p. 44).

[74] Thomas M. J. Fruchterman and Edward M. Reingold. “Graph drawing by force-directed placement”. In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. DOI: https://doi.org/10.1002/spe.4380211102. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.4380211102. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102 (cit. on p. 44).

[75] Svante Wold et al. “Principal component analysis”. In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. URL: https://www.sciencedirect.com/science/article/pii/0169743987800849 (cit. on p. 45).

[76] David Arthur and Sergei Vassilvitskii. “K-Means++: The Advantages of Careful Seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245 (cit. on p. 45).

[77] Satu Elisa Schaeffer. “Graph clustering”. In: *Computer Science Review* 1.1 (2007), pp. 27–64. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2007.05.001. URL: https://www.sciencedirect.com/science/article/pii/S1574013707000020 (cit. on p. 45).

[78] Santo Fortunato and Darko Hric. “Community detection in networks: A user guide”. In: *Physics Reports* 659 (2016). Community detection in networks: A user guide, pp. 1–44. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2016.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0370157316302964 (cit. on p. 45).

[79] Amit Saxena et al. “A review of clustering techniques and developments”. In: *Neurocomputing* 267 (2017), pp. 664–681. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.053. URL: https://www.sciencedirect.com/science/article/pii/S0925231217311815 (cit. on p. 45).

[80] William L. Hamilton et al. *Representation Learning on Graphs: Methods and Applications*. 2018. arXiv: 1709.05584 [cs.SI] (cit. on pp. 47, 48).

[81] Anton Tsitsulin et al. “VERSE: Versatile Graph Embeddings from Similarity Measures”. In: *Proceedings of the 2018 World Wide Web Conference*. WWW ’18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 539–548. ISBN: 9781450356398. DOI: 10.1145/3178876.3186120. URL: https://doi.org/10.1145/3178876.3186120 (cit. on pp. 48, 49, 161).

[82] Michael Gutmann and Aapo Hyvärinen. “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 297–304. URL: https://proceedings.mlr.press/v9/gutmann10a.html (cit. on p. 49).

[83] Andriy Mnih and Yee Whye Teh. “A Fast and Simple Algorithm for Training Neural Probabilistic Language Models”. In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML’12. Edinburgh, Scotland: Omnipress, 2012, pp. 419–426. ISBN: 9781450312851 (cit. on p. 49).

[84] Gary Chartrand and Ping Zhang. *Chromatic Graph Theory*. 1st. Chapman and Hall/CRC, 2008. ISBN: 1584888008 (cit. on p. 53).

[85] Manlio De Domenico et al. “Mathematical Formulation of Multilayer Networks”. In: *Phys. Rev. X* 3 (4 Dec. 2013), p. 041022. DOI: 10.1103/PhysRevX.3.041022. URL: https://link.aps.org/doi/10.1103/PhysRevX.3.041022 (cit. on p. 54).

[86] Manlio De Domenico et al. “Navigability of interconnected networks under random failures”. In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8351–8356. ISSN: 0027-8424. DOI: 10.1073/pnas.1318469111. eprint: https://www.pnas.org/content/111/23/8351.full.pdf. URL: https://www.pnas.org/content/111/23/8351 (cit. on pp. 54, 71).

[87] S. Boccaletti et al. “The structure and dynamics of multilayer networks”. In: *Physics Reports* 544.1 (2014). The structure and dynamics of multilayer networks, pp. 1–122. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2014.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0370157314002105 (cit. on p. 54).

[88] Manlio De Domenico et al. “The physics of spreading processes in multilayer networks”. In: *Nature Physics* 12.10 (2016), pp. 901–906. ISSN: 1745-2481. DOI: 10.1038/nphys3865. URL: https://doi.org/10.1038/nphys3865 (cit. on p. 54).

[89] Ginestra Bianconi. *Multilayer Networks: Structure and Function*. eng. Oxford: Oxford University Press, 2018, p. 416. DOI: 10.1093/oso/9780198753919.001.0001. URL: https://oxford.universitypressscholarship.com/10.1093/oso/9780198753919.001.0001/oso-9780198753919 (cit. on pp. 54, 56).

[90] Petter Holme and Jari Saramäki. “Temporal networks”. In: *Physics Reports* 519.3 (2012). Temporal Networks, pp. 97–125. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2012.03.001. URL: http://www.sciencedirect.com/science/article/pii/S0370157312000841 (cit. on p. 55).

[91] Mikko Kivelä et al. “Multilayer networks”. In: *Journal of Complex Networks* 2.3 (July 2014), pp. 203–271. ISSN: 2051-1310. DOI: 10.1093/comnet/cnu016. eprint: https://academic.oup.com/comnet/article-pdf/2/3/203/9130906/cnu016.pdf. URL: https://doi.org/10.1093/comnet/cnu016 (cit. on pp. 55, 56).

[92] J A van Lidth de Jeude et al. “The multilayer structure of corporate networks”. In: *New Journal of Physics* 21.2 (Feb. 2019), p. 025002. DOI: 10.1088/1367-2630/ab022d (cit. on p. 55).

[93] Marco Bardoscia et al. “Multiplex network analysis of the UK over-the-counter derivatives market”. In: *International Journal of Finance & Economics* 24.4 (2019), pp. 1520–1544. DOI: https://doi.org/10.1002/ijfe.1745. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ijfe.1745. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ijfe.1745 (cit. on p. 55).

[94] Michael Szell et al. “Multirelational organization of large-scale social networks in an online world”. In: *Proceedings of the National Academy of Sciences* 107.31 (2010), pp. 13636–13641. DOI: 10.1073/pnas.1004008107. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1004008107. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1004008107 (cit. on p. 55).

[95] Arkadiusz Stopczynski et al. “Measuring Large-Scale Social Networks with High Resolution”. In: *PLOS ONE* 9.4 (Apr. 2014), pp. 1–24. DOI: 10.1371/journal.pone.0095978. URL: https://doi.org/10.1371/journal.pone.0095978 (cit. on p. 55).

[96] Mark E. Dickison et al. *Multilayer Social Networks*. Cambridge University Press, 2016. DOI: 10.1017/CBO9781139941907 (cit. on p. 55).

[97] Federico Battiston et al. “Efficient exploration of multiplex networks”. In: *New Journal of Physics* 18.4 (Apr. 2016), p. 043035. DOI: 10.1088/1367-2630/18/4/043035. URL: https://doi.org/10.1088/1367-2630/18/4/043035 (cit. on p. 55).

[98] Kelly R. Finn et al. “The use of multilayer network analysis in animal behaviour”. In: *Animal Behaviour* 149 (2019), pp. 7–22. ISSN: 0003-3472. DOI: https://doi.org/10.1016/j.anbehav.2018.12.016. URL: http://www.sciencedirect.com/science/article/pii/S0003347218304020 (cit. on p. 55).

[99] Jacopo A. Baggio et al. “Multiplex social ecological network analysis reveals how social changes affect community robustness more than resource depletion”. In: *Proceedings of the National Academy of Sciences* (2016). ISSN: 0027-8424. DOI: 10.1073/pnas.1604401113. eprint: https://www.pnas.org/content/early/2016/11/14/1604401113.full.pdf. URL: https://www.pnas.org/content/early/2016/11/14/1604401113 (cit. on p. 55).

[100] Massimo Stella et al. “Parasite spreading in spatial ecological multiplex networks”. In: *Journal of Complex Networks* 5.3 (Oct. 2016), pp. 486–511. ISSN: 2051-1310. DOI: 10.1093/comnet/cnw028. eprint: https://academic.oup.com/comnet/article-pdf/5/3/486/17654926/cnw028.pdf. URL: https://doi.org/10.1093/comnet/cnw028 (cit. on p. 55).

[101] Shai Pilosof et al. “The multilayer nature of ecological networks”. In: *Nature Ecology & Evolution* 1.4 (2017), p. 0101. ISSN: 2397-334X. URL: https://doi.org/10.1038/s41559-017-0101 (cit. on p. 55).

[102] Aixia Feng et al. “Three-dimensional air-sea interactions investigated with bilayer networks”. In: *Theoretical and Applied Climatology* 109.3 (2012), pp. 635–643. ISSN: 1434-4483. DOI: 10.1007/s00704-012-0600-7. URL: https://doi.org/10.1007/s00704-012-0600-7 (cit. on p. 55).

[103] Michael W. Cole et al. “Intrinsic and Task-Evoked Network Architectures of the Human Brain”. In: *Neuron* 83.1 (July 2014), pp. 238–251. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2014.05.014. URL: https://doi.org/10.1016/j.neuron.2014.05.014 (cit. on p. 55).

[104] Barry Bentley et al. “The Multilayer Connectome of Caenorhabditis elegans”. In: *PLOS Computational Biology* 12.12 (Dec. 2016), pp. 1–31. DOI: 10.1371/journal.pcbi.1005283. URL: https://doi.org/10.1371/journal.pcbi.1005283 (cit. on p. 55).

[105] Danielle S. Bassett and Olaf Sporns. “Network neuroscience”. In: *Nature Neuroscience* 20.3 (2017), pp. 353–364. ISSN: 1546-1726. URL: https://doi.org/10.1038/nn.4502 (cit. on p. 55).

[106] Richard F. Betzel et al. “The community structure of functional brain networks exhibits scale-specific patterns of inter- and intra-subject variability”. In: *NeuroImage* 202 (2019), p. 115990. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2019.07.003. URL: http://www.sciencedirect.com/science/article/pii/S1053811919305658 (cit. on p. 55).

[107] Riccardo Gallotti and Marc Barthelemy. “The multilayer temporal network of public transport in Great Britain”. In: *Scientific Data* 2.1 (2015), p. 140056. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.56. URL: https://doi.org/10.1038/sdata.2014.56 (cit. on p. 55).

[108] Alberto Aleta et al. “A Multilayer perspective for the analysis of urban transportation systems”. In: *Scientific Reports* 7.1 (2017), p. 44359. ISSN: 2045-2322. DOI: 10.1038/srep44359. URL: https://doi.org/10.1038/srep44359 (cit. on p. 55).

[109] Claudio Angione et al. “Multiplex methods provide effective integration of multi-omic data in genome-scale models”. In: *BMC Bioinformatics* 17.4 (2016), p. 83. ISSN: 1471-2105. DOI: 10.1186/s12859-016-0912-1. URL: https://doi.org/10.1186/s12859-016-0912-1 (cit. on pp. 55, 78).

[110] Marinka Zitnik and Jure Leskovec. “Predicting multicellular function through multi-layer tissue networks”. In: *Bioinformatics* 33.14 (July 2017), pp. i190–i198. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx252. eprint: https://academic.oup.com/bioinformatics/article-pdf/33/14/i190/25157097/btx252.pdf. URL: https://doi.org/10.1093/bioinformatics/btx252 (cit. on pp. 55, 78).

[111] Arda Halu et al. “The multiplex network of human diseases”. In: *npj Systems Biology and Applications* 5.1 (2019), p. 15. ISSN: 2056-7189. DOI: 10.1038/s41540-019-0092-5. URL: https://doi.org/10.1038/s41540-019-0092-5 (cit. on pp. 55, 84).

[112] Bohyun Lee et al. “Heterogeneous Multi-Layered Network Model for Omics Data Integration and Analysis.” eng. In: *Frontiers in genetics* 10 (2019), p. 1381 (cit. on p. 55).

[113] Alberto Aleta and Yamir Moreno. “Multilayer Networks in a Nutshell”. In: *Annual Review of Condensed Matter Physics* 10.1 (2019), pp. 45–62. DOI: 10.1146/annurev-conmatphys-031218-013259. eprint: https://doi.org/10.1146/annurev-conmatphys-031218-013259. URL: https://doi.org/10.1146/annurev-conmatphys-031218-013259 (cit. on p. 56).

[114] Federico Battiston et al. “Structural measures for multiplex networks”. In: *Phys. Rev. E* 89 (3 Mar. 2014), p. 032804. DOI: 10.1103/PhysRevE.89.032804. URL: https://link.aps.org/doi/10.1103/PhysRevE.89.032804 (cit. on pp. 57, 58).

[115] Federico Battiston et al. “The new challenges of multiplex networks: Measures and models”. In: *The European Physical Journal Special Topics* 226.3 (2017), pp. 401–416. ISSN: 1951-6401. DOI: 10.1140/epjst/e2016-60274-8. URL: https://doi.org/10.1140/epjst/e2016-60274-8 (cit. on pp. 57, 58).

[116] Luis Solá et al. “Eigenvector centrality of nodes in multiplex networks”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 23.3 (2013), p. 033131. DOI: 10.1063/1.4818544. eprint: https://doi.org/10.1063/1.4818544. URL: https://doi.org/10.1063/1.4818544 (cit. on p. 57).

[117] Manlio De Domenico et al. “Ranking in interconnected multilayer networks reveals versatile nodes”. In: *Nature Communications* 6.1 (2015), p. 6868. ISSN: 2041-1723. DOI: 10.1038/ncomms7868. URL: https://doi.org/10.1038/ncomms7868 (cit. on p. 59).

[118] Albert Solé-Ribalta et al. “Random walk centrality in interconnected multilayer networks”. In: *Physica D: Nonlinear Phenomena* 323-324 (2016). Nonlinear Dynamics on Interconnected Networks, pp. 73–79. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2016.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0167278916000026 (cit. on p. 59).

[119] Piotr Bródka et al. “Quantifying layer similarity in multiplex networks: a systematic study”. In: *Royal Society Open Science* 5.8 (2018), p. 171747. DOI: 10.1098/rsos.171747. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsos.171747. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsos.171747 (cit. on p. 60).

[120] U. Martin Singh-Blom et al. “Prediction and Validation of Gene-Disease Associations Using Methods Inspired by Social Network Analyses”. In: *PLOS ONE* 8.5 (May 2013), pp. 1–17. DOI: 10.1371/journal.pone.0058977. URL: https://doi.org/10.1371/journal.pone.0058977 (cit. on pp. 60, 61, 153, 154, 160, 161).

[121] Xing Chen. “KATZLDA: KATZ measure for the lncRNA-disease association prediction”. In: *Scientific Reports* 5.1 (2015), p. 16840. ISSN: 2045-2322. DOI: 10.1038/srep16840. URL: https://doi.org/10.1038/srep16840 (cit. on pp. 60, 153).

[122] Xing Chen et al. “A novel approach based on KATZ measure to predict associations of human microbiota with non-infectious diseases”. In: *Bioinformatics* 33.5 (Dec. 2016), pp. 733–739. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw715. eprint: https://academic.oup.com/bioinformatics/article-pdf/33/5/733/25148393/btw715.pdf. URL: https://doi.org/10.1093/bioinformatics/btw715 (cit. on pp. 60, 153).

[123] Chunyan Fan et al. “Prediction of CircRNA-Disease Associations Using KATZ Model Based on Heterogeneous Networks”. In: *Int J Biol Sci* 14 (2018), pp. 1950–1959. DOI: 10.7150/ijbs.28260. URL: https://www.ijbs.com/v14p1950.htm (cit. on pp. 60, 153).

[124] Xuan Zhang et al. “Meta-Path Methods for Prioritizing Candidate Disease miRNAs”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.1 (2019), pp. 283–291. DOI: 10.1109/TCBB.2017.2776280 (cit. on pp. 60, 61, 153, 154, 160).

[125] Shikhar Sharma and Anurag Singh. “An efficient method for link prediction in weighted multiplex networks”. In: *Computational Social Networks* 3.1 (2016), p. 7. ISSN: 2197-4314. DOI: 10.1186/s40649-016-0034-y. URL: https://doi.org/10.1186/s40649-016-0034-y (cit. on pp. 60, 153).

[126] Gilles Didier et al. “Identifying communities from multiplex biological networks”. In: *PeerJ* 3 (2015), e1525. ISSN: 2167-8359. URL: https://doi.org/10.7717/peerj.1525 (cit. on p. 62).

[127] Peter J. Mucha et al. “Community Structure in Time-Dependent, Multiscale, and Multiplex Networks”. In: *Science* 328.5980 (2010), pp. 876–878. ISSN: 0036-8075. DOI: 10.1126/science.1184819. eprint: https://science.sciencemag.org/content/328/5980/876.full.pdf. URL: https://science.sciencemag.org/content/328/5980/876 (cit. on p. 62).

[128] Zhana Kuncheva and Giovanni Montana. “Community Detection in Multiplex Networks Using Locally Adaptive Random Walks”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ASONAM ’15. Paris, France: Association for Computing Machinery, 2015, pp. 1308–1315. ISBN: 9781450338547. DOI: 10.1145/2808797.2808852. URL: https://doi.org/10.1145/2808797.2808852 (cit. on p. 62).

[129] Léo Pio-Lopez et al. “MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach”. In: *Scientific Reports* 11.1 (2021), p. 8794. ISSN: 2045-2322. DOI: 10.1038/s41598-021-87987-1. URL: https://doi.org/10.1038/s41598-021-87987-1 (cit. on pp. 63, 162).

[130] L. Lovász. “Random walks on graphs: A survey”. In: *Combinatorics, Paul Erdos is Eighty* 2.1 (1993), pp. 1–46 (cit. on pp. 64, 70).

[131] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. USA: Society for Industrial and Applied Mathematics, 2000. ISBN: 0898714540 (cit. on pp. 67, 71).

[132] Amy N. Langville and Carl D. Meyer. *Google’s PageRank and Beyond: The Science of Search Engine Rankings*. USA: Princeton University Press, 2006. ISBN: 0691122024 (cit. on pp. 67, 68).

[133] Naoki Masuda et al. “Random walks and diffusion on networks”. In: *Physics Reports* 716-717 (2017). Random walks and diffusion on networks, pp. 1–58. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2017.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0370157317302946 (cit. on p. 70).

[134] Sebastian Köhler et al. “Walking the Interactome for Prioritization of Candidate Disease Genes”. In: *The American Journal of Human Genetics* 82.4 (Apr. 2008), pp. 949–958. ISSN: 0002-9297. DOI: 10.1016/j.ajhg.2008.02.013. URL: https://doi.org/10.1016/j.ajhg.2008.02.013 (cit. on pp. 70, 71, 96).

[135] Jia-Yu Pan et al. “Automatic Multimedia Cross-Modal Correlation Discovery”. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’04. Seattle, WA, USA: Association for Computing Machinery, 2004, pp. 653–658. ISBN: 1581138881. DOI: 10.1145/1014052.1014135. URL: https://doi.org/10.1145/1014052.1014135 (cit. on p. 71).

[136] S. Gómez et al. “Diffusion Dynamics on Multiplex Networks”. In: *Phys. Rev. Lett.* 110 (2 Jan. 2013), p. 028701. DOI: 10.1103/PhysRevLett.110.028701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.028701 (cit. on p. 71).

[137] Yongjin Li and Jagdish C. Patra. “Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network”. In: *Bioinformatics* 26.9 (Mar. 2010), pp. 1219–1224. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq108. eprint: https://academic.oup.com/bioinformatics/article-pdf/26/9/1219/29012989/btq108.pdf. URL: https://doi.org/10.1093/bioinformatics/btq108 (cit. on pp. 71, 83, 96).

[138] Alberto Valdeolivas et al. “Random walk with restart on multiplex and heterogeneous biological networks”. In: *Bioinformatics* 35.3 (July 2018), pp. 497–505. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty637. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/3/497/27699899/bty637.pdf. URL: https://doi.org/10.1093/bioinformatics/bty637 (cit. on pp. 71, 73, 77, 83, 96, 147).

[139] Trey Ideker et al. “A NEW APPROACH TO DECODING LIFE: Systems Biology”. In: *Annual Review of Genomics and Human Genetics* 2.1 (2001). PMID: 11701654, pp. 343–372. DOI: 10.1146/annurev.genom.2.1.343. eprint: https://doi.org/10.1146/annurev.genom.2.1.343. URL: https://doi.org/10.1146/annurev.genom.2.1.343 (cit. on p. 75).

[140] Paul Nurse. “The great ideas of biology.” eng. In: *Clinical medicine (London, England)* 3 (6 Nov. 2003), pp. 560–8 (cit. on p. 76).

[141] Frank J. Bruggeman and Hans V. Westerhoff. "The nature of systems biology". In: *Trends in Microbiology* 15.1 (Jan. 2007), pp. 45–50. ISSN: 0966-842X. DOI: 10.1016/j.tim.2006.11.003. URL: https://doi.org/10.1016/j.tim.2006.11.003 (cit. on p. 76).

[142] Marc Vidal. "A unifying view of 21st century systems biology". In: *FEBS Letters* 583.24 (2009), pp. 3891–3894. DOI: https://doi.org/10.1016/j.febslet.2009.11.024. eprint: https://febs.onlinelibrary.wiley.com/doi/pdf/10.1016/j.febslet.2009.11.024. URL: https://febs.onlinelibrary.wiley.com/doi/abs/10.1016/j.febslet.2009.11.024 (cit. on p. 76).

[143] Peter Uetz et al. "A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae". In: *Nature* 403.6770 (2000), pp. 623–627. ISSN: 1476-4687. DOI: 10.1038/35001009. URL: https://doi.org/10.1038/35001009 (cit. on p. 76).

[144] Jean-François Rual et al. "Towards a proteome-scale map of the human protein-protein interaction network". In: *Nature* 437.7062 (2005), pp. 1173–1178. ISSN: 1476-4687. DOI: 10.1038/nature04209. URL: https://doi.org/10.1038/nature04209 (cit. on p. 76).

[145] L. Giot et al. "A Protein Interaction Map of Drosophila melanogaster". In: *Science* 302.5651 (2003), pp. 1727–1736. DOI: 10.1126/science.1090289. eprint: https://www.science.org/doi/pdf/10.1126/science.1090289. URL: https://www.science.org/doi/abs/10.1126/science.1090289 (cit. on p. 76).

[146] Thomas Rolland et al. "A Proteome-Scale Map of the Human Interactome Network". In: *Cell* 159.5 (Nov. 2014), pp. 1212–1226. ISSN: 0092-8674. DOI: 10.1016/j.cell.2014.10.050. URL: https://doi.org/10.1016/j.cell.2014.10.050 (cit. on p. 76).

[147] Katja Luck et al. "A reference map of the human binary protein interactome". In: *Nature* 580.7803 (2020), pp. 402–408. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2188-x. URL: https://doi.org/10.1038/s41586-020-2188-x (cit. on p. 76).

[148] Michael P. H. Stumpf et al. "Estimating the size of the human interactome". In: *Proceedings of the National Academy of Sciences* 105.19 (2008), pp. 6959–6964. DOI: 10.1073/pnas.0708078105. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0708078105. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0708078105 (cit. on p. 76).

[149] Kwang-Il Goh et al. "The human disease network". In: *Proceedings of the National Academy of Sciences* 104.21 (2007), pp. 8685–8690. DOI: 10.1073/pnas.0701361104. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0701361104. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0701361104 (cit. on p. 77).

[150] Marc Vidal et al. “Interactome Networks and Human Disease”. In: *Cell* 144.6 (2011), pp. 986–998. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2011.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0092867411001309 (cit. on p. 77).

[151] Feixiong Cheng et al. “Network-based prediction of drug combinations”. In: *Nature Communications* 10.1 (2019), p. 1197. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09186-x. URL: https://doi.org/10.1038/s41467-019-09186-x (cit. on pp. 77, 84–87).

[152] Trey Ideker and Ruth Nussinov. “Network approaches and applications in biology”. In: *PLOS Computational Biology* 13.10 (Oct. 2017), pp. 1–3. DOI: 10.1371/journal.pcbi.1005771. URL: https://doi.org/10.1371/journal.pcbi.1005771 (cit. on p. 77).

[153] Albert-László Barabási et al. “Network medicine: a network-based approach to human disease”. In: *Nature Reviews Genetics* 12.1 (2011), pp. 56–68. ISSN: 1471-0064. DOI: 10.1038/nrg2918. URL: https://doi.org/10.1038/nrg2918 (cit. on pp. 77, 87).

[154] Laura Cantini et al. “Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer”. In: *Nature Communications* 12.1 (2021), p. 124. ISSN: 2041-1723. DOI: 10.1038/s41467-020-20430-7. URL: https://doi.org/10.1038/s41467-020-20430-7 (cit. on p. 78).

[155] Monya Baker. “Big biology: The ’omes puzzle”. In: *Nature* 494.7438 (2013), pp. 416–419. ISSN: 1476-4687. DOI: 10.1038/494416a. URL: https://doi.org/10.1038/494416a (cit. on p. 78).

[156] Roel Vermeulen et al. “The exposome and health: Where chemistry meets biology”. In: *Science* 367.6476 (2020), pp. 392–396. DOI: 10.1126/science.aay3164. eprint: https://www.science.org/doi/pdf/10.1126/science.aay3164. URL: https://www.science.org/doi/abs/10.1126/science.aay3164 (cit. on p. 78).

[157] Xueming Liu et al. “Robustness and lethality in multilayer biological molecular networks”. In: *Nature Communications* 11.1 (2020), p. 6043. ISSN: 2041-1723. DOI: 10.1038/s41467-020-19841-3. URL: https://doi.org/10.1038/s41467-020-19841-3 (cit. on pp. 78, 196).

[158] Mikaela Koutrouli et al. “A Guide to Conquer the Biological Network Era Using Graph Theory”. In: *Frontiers in Bioengineering and Biotechnology* 8 (2020). ISSN: 2296-4185. DOI: 10.3389/fbioe.2020.00034. URL: https://www.frontiersin.org/article/10.3389/fbioe.2020.00034 (cit. on p. 80).

[159] Kevin Drew et al. “Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes”. In: *Mol Syst Biol* 13.6 (June 2017), p. 932. ISSN: 1744-4292. DOI: 10.15252/msb.20167490. URL: https://doi.org/10.15252/msb.20167490 (cit. on p. 81).

[160] Madalina Giurgiu et al. “CORUM: the comprehensive resource of mammalian protein complexes–2019”. In: *Nucleic Acids Res* 47.D1 (Jan. 2019), pp. D559–D563. ISSN: 0305-1048. DOI: 10.1093/nar/gky973. URL: https://doi.org/10.1093/nar/gky973 (cit. on p. 81).

[161] Dénes Türei et al. “Integrated intra- and intercellular signaling knowledge for multicellular omics analysis”. In: *Molecular Systems Biology* 17.3 (2021), e9923. DOI: https://doi.org/10.15252/msb.20209923. eprint: https://www.embopress.org/doi/pdf/10.15252/msb.20209923. URL: https://www.embopress.org/doi/abs/10.15252/msb.20209923 (cit. on p. 81).

[162] Dexter Pratt et al. “NDEx, the Network Data Exchange”. In: *Cell Systems* 1.4 (Oct. 2015), pp. 302–305. ISSN: 2405-4712. DOI: 10.1016/j.cels.2015.10.001. URL: https://doi.org/10.1016/j.cels.2015.10.001 (cit. on p. 81).

[163] David Croft et al. “The Reactome pathway knowledgebase”. In: *Nucleic Acids Res* 42.D1 (Jan. 2014), pp. D472–D477. ISSN: 0305-1048. DOI: 10.1093/nar/gkt1102. URL: https://doi.org/10.1093/nar/gkt1102 (cit. on p. 81).

[164] Sebastian Köhler et al. “The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data”. In: *Nucleic Acids Research* 42.D1 (Nov. 2013), pp. D966–D974. ISSN: 0305-1048. DOI: 10.1093/nar/gkt1026. eprint: https://academic.oup.com/nar/article-pdf/42/D1/D966/3529478/gkt1026.pdf. URL: https://doi.org/10.1093/nar/gkt1026 (cit. on p. 82).

[165] Sebastian Köhler et al. “The Human Phenotype Ontology in 2017”. In: *Nucleic Acids Research* 45.D1 (Nov. 2016), pp. D865–D876. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1039. eprint: https://academic.oup.com/nar/article-pdf/45/D1/D865/8846656/gkw1039.pdf. URL: https://doi.org/10.1093/nar/gkw1039 (cit. on p. 82).

[166] Sebastian Köhler et al. “The Human Phenotype Ontology in 2021”. In: *Nucleic Acids Research* 49.D1 (Dec. 2020), pp. D1207–D1217. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa1043. eprint: https://academic.oup.com/nar/article-pdf/49/D1/D1207/35364524/gkaa1043.pdf. URL: https://doi.org/10.1093/nar/gkaa1043 (cit. on p. 82).

[167] Robert Hoehndorf et al. “The role of ontologies in biological and biomedical research: a functional perspective”. In: *Briefings in Bioinformatics* 16.6 (Apr. 2015), pp. 1069–1080. ISSN: 1467-5463. DOI: 10.1093/bib/bbv011. eprint: https://academic.oup.com/bib/article-pdf/16/6/1069/607091/bbv011.pdf. URL: https://doi.org/10.1093/bib/bbv011 (cit. on p. 82).

[168] Tudor Groza et al. “The Human Phenotype Ontology: Semantic Unification of Common and Rare Disease”. In: *The American Journal of Human Genetics* 97.1 (July 2015), pp. 111–124. ISSN: 0002-9297. DOI: 10.1016/j.ajhg.2015.05.020. URL: https://doi.org/10.1016/j.ajhg.2015.05.020 (cit. on p. 82).

[169] Sarah K. Westbury et al. “Human phenotype ontology annotation and cluster analysis to unravel genetic defects in 707 cases with unexplained bleeding and platelet disorders”. In: *Genome Medicine* 7.1 (2015), p. 36. ISSN: 1756-994X. DOI: 10.1186/s13073-015-0151-5. URL: https://doi.org/10.1186/s13073-015-0151-5 (cit. on p. 82).

[170] Philip Resnik. “Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language”. In: *J. Artif. Int. Res.* 11.1 (July 1999), pp. 95–130. ISSN: 1076-9757 (cit. on pp. 83, 85).

[171] Olivier Bodenreider. “The Unified Medical Language System (UMLS): integrating biomedical terminology”. In: *Nucleic Acids Research* 32.suppl_1 (Jan. 2004), pp. D267–D270. ISSN: 0305-1048. DOI: 10.1093/nar/gkh061. eprint: https://academic.oup.com/nar/article-pdf/32/suppl_1/D267/7621558/gkh061.pdf. URL: https://doi.org/10.1093/nar/gkh061 (cit. on p. 83).

[172] Joanna S. Amberger et al. “OMIM.org: Online Mendelian Inheritance in Man (OMIM), an online catalog of human genes and genetic disorders”. In: *Nucleic Acids Research* 43.D1 (Nov. 2014), pp. D789–D798. ISSN: 0305-1048. DOI: 10.1093/nar/gku1205. eprint: https://academic.oup.com/nar/article-pdf/43/D1/D789/7329486/gku1205.pdf. URL: https://doi.org/10.1093/nar/gku1205 (cit. on pp. 83, 87).

[173] Liat Perlman et al. “Combining Drug and Gene Similarity Measures for Drug-Target Elucidation”. In: *Journal of Computational Biology* 18.2 (2011). PMID: 21314453, pp. 133–145. DOI: 10.1089/cmb.2010.0213. eprint: https://doi.org/10.1089/cmb.2010.0213. URL: https://doi.org/10.1089/cmb.2010.0213 (cit. on pp. 84, 85).

[174] Hui Huang et al. “DMAP: a connectivity map database to enable identification of novel drug repositioning candidates”. In: *BMC Bioinformatics* 16.13 (2015), S4. ISSN: 1471-2105. DOI: 10.1186/1471-2105-16-S13-S4. URL: https://doi.org/10.1186/1471-2105-16-S13-S4 (cit. on p. 84).

[175] Xian Liu et al. “In Silicotarget fishing: addressing a “Big Data” problem by ligand-based similarity rankings with data fusion”. In: *Journal of Cheminformatics* 6.1 (2014), p. 33. ISSN: 1758-2946. DOI: 10.1186/1758-2946-6-33. URL: https://doi.org/10.1186/1758-2946-6-33 (cit. on p. 85).

[176] Justin Lamb et al. “The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease.” eng. In: *Science (New York, N.Y.)* 313 (5795 Sept. 2006), pp. 1929–35 (cit. on p. 85).

[177] Francesco Iorio et al. “Identifying network of drug mode of action by gene expression profiling.” eng. In: *Journal of computational biology : a journal of computational molecular cell biology* 16 (2 Feb. 2009), pp. 241–51 (cit. on p. 85).

[178] Fujian Tan et al. “Drug repositioning by applying ‘expression profiles’ generated by integrating chemical structure similarity and gene semantic similarity”. In: *Mol. BioSyst.* 10 (5 2014), pp. 1126–1138. DOI: 10.1039/C3MB70554D. URL: http://dx.doi.org/10.1039/C3MB70554D (cit. on p. 85).

[179] David S. Wishart et al. “DrugBank: a comprehensive resource for in silico drug discovery and exploration”. In: *Nucleic Acids Research* 34 (Jan. 2006), pp. D668–D672. ISSN: 0305-1048. DOI: 10.1093/nar/gkj067. eprint: https://academic.oup.com/nar/article-pdf/34/suppl_1/D668/3924741/gkj067.pdf. URL: https://doi.org/10.1093/nar/gkj067 (cit. on p. 86).

[180] Oron Vanunu et al. “Associating Genes and Protein Complexes with Disease via Network Propagation”. In: *PLOS Computational Biology* 6.1 (Jan. 2010), pp. 1–9. DOI: 10.1371/journal.pcbi.1000641. URL: https://doi.org/10.1371/journal.pcbi.1000641 (cit. on p. 86).

[181] Xuebing Wu et al. “Network-based global inference of human disease genes”. In: *Molecular Systems Biology* 4.1 (2008), p. 189. DOI: https://doi.org/10.1038/msb.2008.27. eprint: https://www.embopress.org/doi/pdf/10.1038/msb.2008.27. URL: https://www.embopress.org/doi/abs/10.1038/msb.2008.27 (cit. on p. 87).

[182] Janet Piñero et al. “The DisGeNET knowledge platform for disease genomics: 2019 update”. In: *Nucleic Acids Res* 48.D1 (Jan. 2020), pp. D845–D855. ISSN: 0305-1048. URL: https://doi.org/10.1093/nar/gkz1021 (cit. on p. 87).

[183] Muhammed A. Yıldırım et al. “Drug–target network”. In: *Nature Biotechnology* 25.10 (2007), pp. 1119–1126. ISSN: 1546-1696. DOI: 10.1038/nbt1338. URL: https://doi.org/10.1038/nbt1338 (cit. on p. 87).

[184] Emre Guney et al. “Network-based in silico drug efficacy screening”. In: *Nature Communications* 7.1 (2016), p. 10331. ISSN: 2041-1723. DOI: 10.1038/ncomms10331. URL: https://doi.org/10.1038/ncomms10331 (cit. on p. 87).

[185] Adam S. Brown and Chirag J. Patel. “A standard database for drug repositioning”. In: *Scientific Data* 4.1 (2017), p. 170029. ISSN: 2052-4463. DOI: 10.1038/sdata.2017.29. URL: https://doi.org/10.1038/sdata.2017.29 (cit. on p. 87).

[186] Daniel Scott Himmelstein et al. “Systematic integration of biomedical knowledge prioritizes drugs for repurposing”. In: *eLife* 6 (Sept. 2017). Ed. by Alfonso Valencia, e26726. ISSN: 2050-084X. DOI: 10.7554/eLife.26726. URL: https://doi.org/10.7554/eLife.26726 (cit. on p. 87).

[187] Ana Pombo and Niall Dillon. “Three-dimensional genome architecture: players and mechanisms”. In: *Nature Reviews Molecular Cell Biology* 16.4 (2015), pp. 245–257. ISSN: 1471-0080. DOI: 10.1038/nrm3965. URL: https://doi.org/10.1038/nrm3965 (cit. on p. 88).

[188] David U. Gorkin et al. “Common DNA sequence variation influences 3-dimensional conformation of the human genome”. In: *Genome Biology* 20.1 (2019), p. 255. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1855-4. URL: https://doi.org/10.1186/s13059-019-1855-4 (cit. on p. 88).

[189] Peter Hugo Lodewijk Krijger and Wouter de Laat. “Regulation of disease-associated gene expression in the 3D genome”. In: *Nature Reviews Molecular Cell Biology* 17.12 (2016), pp. 771–782. ISSN: 1471-0080. DOI: 10.1038/nrm.2016.138. URL: https://doi.org/10.1038/nrm.2016.138 (cit. on p. 88).

[190] Chiara Anania and Darío G Lupiáñez. “Order and disorder: abnormal 3D chromatin organization in human disease”. In: *Briefings in Functional Genomics* 19.2 (Feb. 2020), pp. 128–138. ISSN: 2041-2657. DOI: 10.1093/bfgp/elz028. eprint: https://academic.oup.com/bfg/article-pdf/19/2/128/32989783/elz028.pdf. URL: https://doi.org/10.1093/bfgp/elz028 (cit. on p. 88).

[191] Chin-Tong Ong and Victor G. Corces. “Enhancer function: new insights into the regulation of tissue-specific gene expression”. In: *Nature Reviews Genetics* 12.4 (2011), pp. 283–293. ISSN: 1471-0064. DOI: 10.1038/nrg2957. URL: https://doi.org/10.1038/nrg2957 (cit. on p. 88).

[192] Quentin Szabo et al. “Principles of genome folding into topologically associating domains”. In: *Science Advances* 5.4 (2019), eaaw1668. DOI: 10.1126/sciadv.aaw1668. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.aaw1668. URL: https://www.science.org/doi/abs/10.1126/sciadv.aaw1668 (cit. on pp. 88, 90).

[193] Jonathan A. Beagan and Jennifer E. Phillips-Cremins. “On the existence and functionality of topologically associating domains”. In: *Nature Genetics* 52.1 (2020), pp. 8–16. ISSN: 1546-1718. DOI: 10.1038/s41588-019-0561-1. URL: https://doi.org/10.1038/s41588-019-0561-1 (cit. on pp. 88, 129).

[194] Elissavet Kentepozidou et al. “Clustered CTCF binding is an evolutionary mechanism to maintain topologically associating domains”. In: *Genome Biology* 21.1 (2020), p. 5. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1894-x. URL: https://doi.org/10.1186/s13059-019-1894-x (cit. on p. 88).

[195] Rieke Kempfer and Ana Pombo. “Methods for mapping 3D chromosome architecture”. In: *Nature Reviews Genetics* 21.4 (2020), pp. 207–226. ISSN: 1471-0064. DOI: 10.1038/s41576-019-0195-2. URL: https://doi.org/10.1038/s41576-019-0195-2 (cit. on pp. 89, 91).

[196] Jenny A. Croft et al. “Differences in the Localization and Morphology of Chromosomes in the Human Nucleus ”. In: *Journal of Cell Biology* 145.6 (June 1999), pp. 1119–1131. ISSN: 0021-9525. DOI: 10.1083/jcb.145.6.1119. eprint: https://rupress.org/jcb/article-pdf/145/6/1119/1285017/9812016.pdf. URL: https://doi.org/10.1083/jcb.145.6.1119 (cit. on p. 91).

[197] Marion Cremer et al. “Multicolor 3D Fluorescence In Situ Hybridization for Imaging Interphase Chromosomes”. In: *The Nucleus: Volume 1: Nuclei and Subnuclear Components*. Ed. by Ronald Hancock. Totowa, NJ: Humana Press, 2008, pp. 205–239. ISBN: 978-1-59745-406-3. DOI: 10.1007/978-1-59745-406-3_15. URL: https://doi.org/10.1007/978-1-59745-406-3_15 (cit. on p. 91).

[198] Job Dekker et al. “Capturing Chromosome Conformation”. In: *Science* 295.5558 (2002), pp. 1306–1311. DOI: 10.1126/science.1067799. eprint: https://www.science.org/doi/pdf/10.1126/science.1067799. URL: https://www.science.org/doi/abs/10.1126/science.1067799 (cit. on p. 91).

[199] Marieke Simonis et al. “Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4C)”. In: *Nature Genetics* 38.11 (2006), pp. 1348–1354. ISSN: 1546-1718. DOI: 10.1038/ng1896. URL: https://doi.org/10.1038/ng1896 (cit. on p. 91).

[200] Zhihu Zhao et al. “Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions”. In: *Nature Genetics* 38.11 (2006), pp. 1341–1347. ISSN: 1546-1718. DOI: 10.1038/ng1891. URL: https://doi.org/10.1038/ng1891 (cit. on p. 91).

[201] Erez Lieberman-Aiden et al. “Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome”. In: *Science* 326.5950 (2009), pp. 289–293. DOI: 10.1126/science.1181369. eprint: https://www.science.org/doi/pdf/10.1126/science.1181369. URL: https://www.science.org/doi/abs/10.1126/science.1181369 (cit. on p. 91).

[202] Suhas S. P. Rao et al. “A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping”. In: *Cell* 159.7 (Dec. 2014), pp. 1665–1680. ISSN: 0092-8674. DOI: 10.1016/j.cell.2014.11.021. URL: https://doi.org/10.1016/j.cell.2014.11.021 (cit. on p. 91).

[203] Stefan Schoenfelder et al. “The pluripotent regulatory circuitry connecting promoters to their long-range interacting elements.” eng. In: *Genome research* 25 (4 Apr. 2015), pp. 582–97 (cit. on pp. 91, 110).

[204] Robert A. Beagrie et al. “Complex multi-enhancer contacts captured by genome architecture mapping”. In: *Nature* 543.7646 (2017), pp. 519–524. ISSN: 1476-4687. DOI: 10.1038/nature21411. URL: https://doi.org/10.1038/nature21411 (cit. on p. 91).

[205] Sofia A. Quinodoz et al. “Higher-Order Inter-chromosomal Hubs Shape 3D Genome Organization in the Nucleus”. In: *Cell* 174.3 (2018), 744–757.e24. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2018.05.024. URL: https://www.sciencedirect.com/science/article/pii/S0092867418306366 (cit. on p. 91).

[206] Meizhen Zheng et al. “Multiplex chromatin interactions with single-molecule precision”. In: *Nature* 566.7745 (2019), pp. 558–562. ISSN: 1476-4687. DOI: 10.1038/s41586-019-0949-1. URL: https://doi.org/10.1038/s41586-019-0949-1 (cit. on p. 91).

[207] Leina Lu et al. “Robust Hi-C Maps of Enhancer-Promoter Interactions Reveal the Function of Non-coding Genome in Neural Development and Diseases”. In: *Molecular Cell* 79.3 (2020), 521–534.e15. ISSN: 1097-2765. DOI: https://doi.org/10.1016/j.molcel.2020.06.007. URL: https://www.sciencedirect.com/science/article/pii/S1097276520303920 (cit. on p. 92).

[208] Michael I. Robson et al. “Regulatory Landscaping: How Enhancer-Promoter Communication Is Sculpted in 3D”. In: *Molecular Cell* 74.6 (2019), pp. 1110–1122. ISSN: 1097-2765. DOI: https://doi.org/10.1016/j.molcel.2019.05.032. URL: https://www.sciencedirect.com/science/article/pii/S1097276519304046 (cit. on p. 93).

[209] Jonathan Cairns et al. “CHiCAGO: robust detection of DNA looping interactions in Capture Hi-C data”. In: *Genome Biology* 17.1 (2016), p. 127. ISSN: 1474-760X. DOI: 10.1186/s13059-016-0992-2. URL: https://doi.org/10.1186/s13059-016-0992-2 (cit. on p. 93).

[210] Anthony Baptista et al. “Universal multilayer network exploration by random walk with restart”. In: *Communications Physics* 5.1 (2022), p. 170. ISSN: 2399-3650. DOI: 10.1038/s42005-022-00937-9. URL: https://doi.org/10.1038/s42005-022-00937-9 (cit. on pp. 98, 147, 161).

[211] Manlio De Domenico et al. “MuxViz: a tool for multilayer analysis and visualization of networks”. In: *Journal of Complex Networks* 3.2 (Oct. 2014), pp. 159–176. ISSN: 2051-1310. DOI: 10.1093/comnet/cnu038. eprint: https://academic.oup.com/comnet/article-pdf/3/2/159/1070864/cnu038.pdf. URL: https://doi.org/10.1093/comnet/cnu038 (cit. on p. 108).

[212] Luca Rossi and Matteo Magnani. “Towards effective visual analytics on multiplex and multilayer networks”. In: *Chaos, Solitons & Fractals* 72 (2015). Multiplex Networks: Structure, Dynamics and Applications, pp. 68–76. ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2014.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0960077914002422 (cit. on p. 108).

[213] F. McGee et al. “The State of the Art in Multilayer Network Visualization”. In: *Computer Graphics Forum* 38.6 (2019), pp. 125–149. DOI: https://doi.org/10.1111/cgf.13610. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13610. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13610 (cit. on p. 108).

[214] Sebastian Pirch et al. “The VRNetzer platform enables interactive network analysis in Virtual Reality”. In: *Nature Communications* 12.1 (2021), p. 2432. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22570-w. URL: https://doi.org/10.1038/s41467-021-22570-w (cit. on p. 108).

[215] Richard Bookstaber and Dror Kenett. *Looking Deeper, Seeing More: A Multilayer Map of the Financial System*. Briefs 16-06. Office of Financial Research, US Department of the Treasury, July 2016. URL: https://ideas.repec.org/p/ofr/briefs/16-06.html (cit. on p. 109).

[216] Marco Bardoscia et al. “The physics of financial networks”. In: *Nature Reviews Physics* 3.7 (2021), pp. 490–507. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00322-5. URL: https://doi.org/10.1038/s42254-021-00322-5 (cit. on p. 109).

[217] Biola M. Javierre et al. “Lineage-Specific Genome Architecture Links Enhancers and Non-coding Disease Variants to Target Gene Promoters”. In: *Cell* 167.5 (2016), 1369–1384.e19. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2016.09.037. URL: https://www.sciencedirect.com/science/article/pii/S0092867416313228 (cit. on p. 110).

[218] Melissa Haendel et al. “How many rare diseases are there?” eng. In: *Nature reviews. Drug discovery* 19.32020066 (Feb. 2020), pp. 77–78. ISSN: 1474-1776. DOI: 10.1038/d41573-019-00180-y. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7771654/ (cit. on p. 129).

[219] Melina Claussnitzer et al. “A brief history of human disease genetics”. In: *Nature* 577.7789 (2020), pp. 179–189. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1879-7. URL: https://doi.org/10.1038/s41586-019-1879-7 (cit. on p. 129).

[220] Takashi Nagano et al. “Single-cell Hi-C reveals cell-to-cell variability in chromosome structure”. In: *Nature* 502.7469 (2013), pp. 59–64. ISSN: 1476-4687. DOI: 10.1038/nature12593. URL: https://doi.org/10.1038/nature12593 (cit. on p. 129).

[221] Ilya M. Flyamer et al. “Single-nucleus Hi-C reveals unique chromatin reorganization at oocyte-to-zygote transition”. In: *Nature* 544.7648 (2017), pp. 110–114. ISSN: 1476-4687. DOI: 10.1038/nature21711. URL: https://doi.org/10.1038/nature21711 (cit. on p. 129).

[222] Bogdan Bintu et al. “Super-resolution chromatin tracing reveals domains and cooperative interactions in single cells”. In: *Science* 362.6413 (2018), eaau1783. DOI: 10.1126/science.aau1783. eprint: https://www.science.org/doi/pdf/10.1126/science.aau1783. URL: https://www.science.org/doi/abs/10.1126/science.aau1783 (cit. on p. 129).

[223] Renée Beekman et al. “The reference epigenome and regulatory chromatin landscape of chronic lymphocytic leukemia”. In: *Nature Medicine* 24.6 (2018), pp. 868–880. ISSN: 1546-170X. DOI: 10.1038/s41591-018-0028-4. URL: https://doi.org/10.1038/s41591-018-0028-4 (cit. on p. 129).

[224] Kathy Macropol et al. “RRW: repeated random walks on genome-scale protein networks for local cluster discovery”. In: *BMC Bioinformatics* 10.1 (2009), p. 283. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-283. URL: https://doi.org/10.1186/1471-2105-10-283 (cit. on p. 147).

[225] Alexis Papadimitriou et al. “Fast and accurate link prediction in social networking systems”. In: *Journal of Systems and Software* 85.9 (2012). Selected papers from the 2011 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA 2011), pp. 2119–2132. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2012.04.019. URL: https://www.sciencedirect.com/science/article/pii/S0164121212001069 (cit. on p. 153).

[226] Chang Zhou et al. “Scalable Graph Embedding for Asymmetric Proximity”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (Feb. 2017). DOI: 10.1609/aaai.v31i1.10878. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10878 (cit. on p. 162).

[227] Walter Nelson et al. “To Embed or Not: Network Embedding as a Paradigm in Computational Biology”. In: *Frontiers in Genetics* 10 (2019). ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00381. URL: https://www.frontiersin.org/article/10.3389/fgene.2019.00381 (cit. on p. 163).

[228] Yi-Jiao Zhang et al. “Systematic comparison of graph embedding methods in practical tasks”. In: *Phys. Rev. E* 104 (4 Oct. 2021), p. 044315. DOI: 10.1103/PhysRevE.104.044315. URL: https://link.aps.org/doi/10.1103/PhysRevE.104.044315 (cit. on p. 163).

[229] Ginestra Bianconi and Sergey N. Dorogovtsev. “Multiple percolation transitions in a configuration model of a network of networks”. In: *Phys. Rev. E* 89 (6 June 2014), p. 062814. DOI: 10.1103/PhysRevE.89.062814. URL: https://link.aps.org/doi/10.1103/PhysRevE.89.062814 (cit. on p. 193).

[230] Béla Bollobás. “A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs”. In: *European Journal of Combinatorics* 1.4 (1980), pp. 311–316. ISSN: 0195-6698. DOI: https://doi.org/10.1016/S0195-6698(80)80030-8. URL: https://www.sciencedirect.com/science/article/pii/S0195669880800308 (cit. on p. 193).

[231] Ginestra Bianconi and Filippo Radicchi. “Percolation in real multiplex networks”. In: *Phys. Rev. E* 94 (6 Dec. 2016), p. 060301. DOI: 10.1103/PhysRevE.94.060301. URL: https://link.aps.org/doi/10.1103/PhysRevE.94.060301 (cit. on p. 193).

[232] Sergey V. Buldyrev et al. “Catastrophic cascade of failures in interdependent networks”. In: *Nature* 464.7291 (2010), pp. 1025–1028. ISSN: 1476-4687. DOI: 10.1038/nature08932. URL: https://doi.org/10.1038/nature08932 (cit. on pp. 193, 195).

[233] Filippo Radicchi. “Percolation in real interdependent networks”. In: *Nature Physics* 11.7 (2015), pp. 597–602. ISSN: 1745-2481. DOI: 10.1038/nphys3374. URL: https://doi.org/10.1038/nphys3374 (cit. on p. 193).

[234] M. E. J. Newman. “Spread of epidemic disease on networks”. In: *Phys. Rev. E* 66 (1 July 2002), p. 016128. DOI: 10.1103/PhysRevE.66.016128. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.016128 (cit. on p. 194).

[235] Jianxi Gao et al. “Networks formed from interdependent networks”. In: *Nature Physics* 8.1 (2012), pp. 40–48. ISSN: 1745-2481. DOI: 10.1038/nphys2180. URL: https://doi.org/10.1038/nphys2180 (cit. on p. 195).

[236] Charles D. Brummitt et al. “Suppressing cascades of load in interdependent networks”. In: *Proceedings of the National Academy of Sciences* 109.12 (2012), E680–E689. DOI: 10.1073/pnas.1110586109. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1110586109. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1110586109 (cit. on p. 195).

[237] Jörg Stelling et al. “Robustness of Cellular Functions”. In: *Cell* 118.6 (Sept. 2004), pp. 675–685. ISSN: 0092-8674. DOI: 10.1016/j.cell.2004.09.008. URL: https://doi.org/10.1016/j.cell.2004.09.008 (cit. on p. 196).

[238] Hiroaki Kitano. “Biological robustness”. In: *Nature Reviews Genetics* 5.11 (2004), pp. 826–837. ISSN: 1471-0064. DOI: 10.1038/nrg1471. URL: https://doi.org/10.1038/nrg1471 (cit. on p. 196).

[239] Ashley G. Smart et al. “Cascading failure and robustness in metabolic networks”. In: *Proceedings of the National Academy of Sciences* 105.36 (2008), pp. 13223–13228. DOI: 10.1073/pnas.0803571105. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0803571105. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0803571105 (cit. on p. 196).

[240] Daniel Marbach et al. “Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases”. In: *Nature Methods* 13.4 (2016), pp. 366–370. ISSN: 1548-7105. DOI: 10.1038/nmeth.3799. URL: https://doi.org/10.1038/nmeth.3799 (cit. on p. 196).

ANNEXES

A. Matériel supplémentaire du manuscrit

A.1. Autres algorithmes

Nous allons expliciter quelques autres algorithmes sur les réseaux multi-couches. Ces méthodes existent aussi sur les réseaux simples. Cependant, dans le cadre de ma thèse, je me suis intéressé à ces méthodes uniquement dans un cadre multi-couche, en particulier sur les réseaux inter-dépendants. Deux réseaux sont inter-dépendants s'ils ont des interactions entre eux. Il s'agit d'un cas particulier de réseau multi-couche.

- **La Percolation :** La théorie de la percolation permet d'étudier le comportement de systèmes complexes au cours de changements dans leur structure. Dans le cas de la science des réseaux, le processus le plus usuel de percolation consiste à la suppression successive de nœuds (*site percolation*), ou d'arêtes (*bond percolation*). L'analyse des conséquences de ces suppressions successives permet d'obtenir de nombreuses informations concernant le réseau, notamment sa robustesse face aux attaques. Ce genre de mécanismes permet entre autres d'étudier la résilience d'un réseau, ainsi que de définir des diagrammes de transition de phases. Le processus de percolation a été étendu à des réseaux multi-couches [229], où chaque couche du réseau est définie à partir du *configuration model* [230] (méthode pour générer des réseaux aléatoires où le degré des nœuds est prédéfini), ainsi qu'à des réseaux multiplexes réels [231], ou encore à des réseaux réels inter-dépendants [232, 233].
  Nous allons décrire brièvement le processus de percolation pour un réseau simple. Il faut introduire quelques notions au préalable. On définit tout d'abord la fonction génératrice du réseau i de la manière suivante :

$$G_i = \sum_{k=0}^{\infty} P_i(k) x^k \tag{.1}$$

  avec $P_i(k)$ une distribution de probabilité qui associe à chacun des n_i nœuds du réseau i un degré k. x est une variable aléatoire arbitraire. On peut définir également le degré moyen du réseau i :

$$\langle k \rangle_i = \sum_{k=0}^{\infty} k P_i(k) = \left. \frac{\partial G_i}{\partial x} \right|_{x \to 1} = G_i'(1) \tag{.2}$$

  Si on suppose que l'on a un réseau infiniment grand, i.e $n_i \to \infty$, alors le processus de connections aléatoires entre les nœuds peut être décrit comme un processus de branchement dans lequel la probabilité qu'une arête sortant d'un nœud soit connectée à un nœud de degré k est égale à $kP_i/\langle k \rangle_i$. Ainsi, il reste

au nœud connecté $k-1$ arêtes sortantes de libres. Par conséquent, la fonction génératrice de ce processus de branchement est la suivante :

$$H_i(x) = \sum_{k=0}^{\infty} \frac{P_i(k)kx^{k-1}}{\langle k \rangle_i} = \frac{G_i'(x)}{G_i'(1)} \tag{.3}$$

De plus, on définit f_i comme étant la probabilité qu'une arête sélectionnée au hasard ne soit pas connectée à la plus grande composante connexe du réseau. Cette probabilité est égale à f_i^{k-1} si le nœud a $k-1$ arêtes sortantes. Ainsi, on remarque que l'on peut définir la relation de récurrence suivante :

$$f_i = H_i(f_i) \tag{.4}$$

Nous pouvons aussi définir la relation de récurrence déterminant la probabilité qu'un nœud soit connecté à la composante géante du réseau par l'équation suivante :

$$g_i = G_i(f_i) \tag{.5}$$

Maintenant, si on suppose que l'on retire une fraction $(1-p)$ de nœuds au réseau, nous définissons donc une nouvelle fonction génératrice $G_i(x;p)$. Elle définit la probabilité qu'exactement m des k arêtes soient connectées à un nœud restant. Cette probabilité est simplement la distribution binomiale $\binom{k}{m}(p)^m(1-p)^{k-m}$ et donc la distribution de probabilité de m est générée par la fonction génératrice suivante [234] :

$$\begin{aligned} G_i(x;p) &= \sum_{m=0}^{\infty} p_m x^m = \sum_{m=0}^{\infty}\sum_{k=m}^{\infty} p_k \binom{k}{m} p^m(1-p)^{k-m}x^m \\ &= \sum_{k=0}^{\infty} p_k \sum_{m=0}^{k} \binom{k}{m}(xp)^m(1-p)^{k-m} \\ &= \sum_{k=0}^{\infty} p_k(1-p+xp)^k \\ &= G_i(1-p+xp) \end{aligned} \tag{.6}$$

Ainsi, en utilisant l'équation (2.29) avec les équations (2.27 et 2.28) on obtient le couple d'équations suivant :

$$\begin{cases} g_i(p) = 1 - G_i(pf_i(p) + 1 - p) \\ f_i(p) = H_i(pf_i(p) + 1 - p) \end{cases} \tag{.7}$$

À partir des relations de récurrence définies précédemment, on peut définir une variable pertinente : la fraction de nœuds qui sont connectés à la composante

géante du réseau i, notée $P_{\infty,i}$. Cette variable est donnée par le produit suivant :

$$P_{\infty,i} = pg_i(p) \tag{.8}$$

On remarque que plus p décroît, plus on s'approche du cas où $f_i = 1$, c'est à dire qu'aucun nœud n'est connecté à la composante géante.

- **Processus de *cascading failure* :** Dans le cas où le processus de percolation se passe dans des réseaux inter-dépendants, on parle de processus de *cascading failure*. Ce terme s'explique par le comportement des défaillances couplées et en série que l'on observe [232]. En d'autres termes, la défaillance d'une partie d'un réseau entraîne la défaillance d'une partie d'un autre réseau, et cette défaillance entraîne elle-même la défaillance d'une autre partie du premier réseau. Bien entendu, ce processus peut être envisagé dans le cas où il y a plus de deux réseaux inter-dépendants [235]. Le processus de *cascading failure* est une méthode privilégiée pour étudier la robustesse des réseaux lorsqu'ils sont soumis à des défaillances en série. Un exemple est le *black-out* électrique du 14 au 16 août 2003 qui affecta une partie de l'Amérique du Nord [236], ou celui du 28 septembre 2003 qui affecta l'Italie [232].
  Nous allons décrire le processus pour le cas de deux réseaux inter-dépendants (qui peuvent être deux couches d'un réseau hétérogène), notés réseaux α et réseau β, ayant respectivement n_α et n_β nœuds, et ayant pour distribution de degré des nœuds les distributions $P_\alpha(k)$ et $P_\beta(k)$. De plus, on définit par q_α la fraction de nœuds du réseaux α connectés au réseau β, et par q_β la fraction de nœuds du réseau β connectés au réseau α. On suppose aussi que chaque nœud connecté à l'autre réseau n'est pas connecté à plus d'un nœud dans cet autre réseau. Donc, si un nœud du réseau α, noté $(v_\alpha)_i$ est connecté à un nœud du réseau β, noté $(v_\beta)_j$, et que ce même nœud $(v_\beta)_j$ est connecté à un second nœud du réseau α, noté $(v_\alpha)_l$, alors $l = i$. Cette condition est appelée condition de non-rétroaction.
  On suppose que l'on enlève $(1-p)$ nœuds du réseau α. Ainsi, la fraction de nœuds restants dans le réseau α est donnée par $\psi'_1 = p$, et la fraction de nœuds qui reste connectée à la composante géante est donnée par $\psi_1 = \psi'_1 g_\alpha(\psi'_1)$. La fraction de nœuds $(1-\psi_1)q_\beta$ correspond aux nœuds du réseau β étant connectés aux nœuds enlevés du réseau α. Par conséquent, la fraction de nœuds restant fonctionnelle (donc connectée à la composante géante) dans le réseau β est égale à $\phi'_1 = 1 - q_\beta[1 - \psi'_1 g_\alpha(\psi'_1)]$, et la fraction de nœuds du réseau β qui reste connectée à la composante géante est donnée par $\phi_1 = \phi'_1 g_\beta(\phi'_1)$.
  En s'appuyant sur cette démarche, on peut définir le processus de *cascading*

failures, qui définit la séquence suivante :

$$\begin{cases} \psi'_1 = p \\ \phi'_1 = 1 - q_\beta[1 - p g_\alpha(\psi'_1)] \\ \psi'_t = p[1 - q_\alpha(1 - g_\beta(\phi'_{t-1}))] \\ \phi'_t = 1 - q_\beta[1 - p g_\alpha(\psi'_{t-1})] \end{cases} \tag{.9}$$

De plus, on peut définir, pour chaque étape t du processus, la fraction de nœuds qui reste connectée à la composante géante, pour chacun des deux réseaux α et β, de la manière suivante :

$$\begin{cases} \psi_t = \psi'_t g_\alpha(\psi'_t) \\ \phi_t = \phi'_t g_\beta(\phi'_t) \end{cases} \tag{.10}$$

La fin du processus de *cascading failures* est obtenue lorsque les conditions suivantes sont réalisées : $\psi'_{t+1} = \psi'_t$ et $\phi'_{t+1} = \phi'_t$, c'est-à-dire que les états stationnaires sont atteints. On les note $\psi^{*'} = \psi'_{t\to\infty}$ et $\phi^{*'} = \phi'_{t\to\infty}$. On obtient le couple d'équations stationnaires suivant :

$$\begin{cases} \psi^{*'} = p[1 - q_\alpha(1 - g_\beta(\psi^{*'}))] \\ \phi^{*'} = 1 - q_\beta[1 - p g_\alpha(\phi^{*'})] \end{cases} \tag{.11}$$

On peut définir la fraction de nœuds finale qui reste connectée à la composante géante, pour chacun des deux réseaux α et β, de la manières suivante :

$$\begin{cases} P_{\infty,\alpha} = \psi^* = \psi^{*'} g_\alpha(\psi^{*'}) \\ P_{\infty,\beta} = \phi^* = \phi^{*'} g_\beta(\phi^{*'}) \end{cases} \tag{.12}$$

La robustesse d'un système biologique est définie comme étant sa faculté de maintenir stables ses fonctions malgré les perturbations [237, 238]. Dans le cas des réseaux biologiques, la faculté des organismes vivants de survivre à une large gamme de conditions environnementales a pu être expliquée à l'aide de la méthode de *cascading failure* appliquée aux réseaux métaboliques [239]. De plus, l'analyse de la robustesse des réseaux moléculaires soumis à des perturbations est une méthode efficace pour découvrir les mécanismes moléculaires à l'origine des maladies [240]. Récemment, une étude [157] s'est inspirée de la méthode de *cascading failure* afin d'étudier la robustesse de réseaux biologiques multi-couches. Le réseau multi-couche étudié intègre des réseaux de gènes, de protéines et de métabolites et tend à confirmer l'importance des couplages entre les différentes couches dans la dynamique cellulaire résultant d'une perturbation génétique.

A.2. Norme 2 d'une matrice

On définit une matrice $M \in \mathbb{R}^{n\times n}$. Le rayon spectral, noté $\rho(.)$, est défini pour une matrice M telle que :

$$\rho(M) = \max(\{|\lambda|, \lambda \in \mathbb{C}, \text{valeurs propres de } M\}) \tag{A.1.1}$$

La norme 2 d'une matrice, notée $\|.\|_2$, est définie telle que :

$$\|M\|_2 = (\rho(M^T \cdot M))^{\frac{1}{2}} \tag{A.1.2}$$

Une propriété de la norme 2 est que cette norme est égale ou inférieure à la norme de Frobenius, notée $\|.\|_F$. Cela se traduit de la manière suivante :

$$\|M\|_2 \leq \|M\|_F = (\sum_{i,j=1}^{n} (M_{ij})^2)^{\frac{1}{2}} \tag{A.1.3}$$

A.3. Exemple de table de combinaisons dans un réseau multi-couche universel constitué de deux réseaux multiplexes

On considère le réseau multi-couche universel défini dans la Fig. 9.2 gauche. On rappelle que la similarité de Katz peut s'écrire dans une bonne approximation de la manière suivante :

$$\sigma = \alpha S_1 + \alpha^2 S_2 + \alpha^3 S_3 + \alpha^4 S_4 \tag{A.2.1}$$

De plus, on peut définir la similarité de Katz entre un réseau source j et un réseau cible i , notée σ_{ij}, qui est un bloc de la matrice de similarité de Katz σ, défini de la manière suivante :

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} \tag{A.2.2}$$

Dans notre cas, on reprend l'exemple de la section 9.2.1, avec comme réseau source, le réseau 1 et comme réseau cible, le réseau 2. En d'autres termes, on cherche à déterminer la similarité de Katz $\sigma_{2,1}$. On se restreint uniquement au terme S_4 de cette similarité. Donc, la table des combinaisons est la suivante : Cette table de combinaisons permet d'obtenir l'ensemble des termes constituant S_4, en lisant la suite de termes deux à deux et en leur associant les matrices, ainsi la première suite est : 21 11 11 11, ce qui donne le produit matriciel : $B_{2,1}\mathscr{A}_1\mathscr{A}_1\mathscr{A}_1 = B_{2,1}\mathscr{A}_1^3$. L'ensemble des termes constituant S_4 est obtenu par la même lecture de la table de combinaisons,

Cible	Combinaisons	Source
2	11 11 11 11 11 22 11 22 11 11 22 22 22 11 11 22 11 22 22 22 11 22 22 22	1

ce qui permet d'obtenir l'équation suivante :

$$S_4 = B_{2,1}\mathscr{A}_1^3 + B_{2,1}\mathscr{A}_1 B_{1,2}B_{2,1} + B_{2,1}B_{1,2}B_{2,1}\mathscr{A}_1 + B_{2,1}B_{1,2}\mathscr{A}_2 B_{2,1} + \mathscr{A}_2 B_{2,1}\mathscr{A}_1^2 + \mathscr{A}_2 B_{2,1}B_{1,2}B_{2,1} + \mathscr{A}_2^2 B_{2,1}\mathscr{A}_1 + \mathscr{A}_2^3 B_{2,1} \tag{A.2.3}$$

Cette équation est la même que celle obtenue dans l'équation (9.8).

B. *Universal Multilayer Exploration by Random Walk with Restart* : matériel supplémentaire

Supplementary Information for

Universal Multilayer Network Exploration by Random Walk with Restart

Anthony Baptista[1, 2, *], Aitor Gonzalez[2] and Anaïs Baudot[1, 3, *]

[1] Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, CNRS, Marseille, France, [2] Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France, [3] Barcelona Supercomputing Center, Barcelona, Spain

* anthony.baptista@univ-amu.fr, anais.baudot@univ-amu.fr

This PDF file includes:

Contents

1. Supplementary Note 1: MultiXrank mathematical formulation

A. Elements of proof and rate of convergence.

We hypothesize that S is normalized in a way that makes it a Stochastic matrix (1).
We recall that the radius of convergence is defined as follows:

$$\mathrm{R} = \sup\{|z| : z \in \mathbb{C}, \sum a_n z^n \text{ converge}\}$$

A.1. Elements of proof of convergence.

$\mathbf{p}_t = \alpha S \mathbf{p}_{t-1} + b$
$R(S) = 1$; because S is a Stochastic matrix
$R(\alpha S) < 1; \alpha \in [0, 1[$ (†)
Because the rate of convergence of the new matrix αS is strictly inferior to 1, $\exists\ \mathbf{p}^*$ such as:
$\mathbf{p}_t \xrightarrow[t\to\infty]{} \mathbf{p}^*$

Proof of (†):

$R(S) = max(|\lambda i|)$, λi = eigenvalue of S
So $R(\alpha S) = \alpha max(|\lambda i|)$
The Perron-Frobenius gives that $\lambda i \in [0,\ 1]\ \forall i$, and because $\alpha \in [0, 1[$
So, $R(\alpha S) = max(|\alpha \lambda i|)$
Thus, $0 < R(\alpha S) < R(S) = 1$
In conclusion, $R(\alpha S) < 1$

A.2. Rate of convergence.

$Ax = b$
Let define: $A = M - N$ and $H = M^{-1}N$
So: $x = Hx + d \quad \rightarrow \quad x(t) = Hx(t-1) + d$

The spectrum of H is equal to:
$\sigma(H) = \{\lambda_1, \lambda_2, ..., \lambda_s\}$ and $1 > |\lambda_1| > ... > |\lambda_s|$ (Perron-Frobenius)

We define the error at the k^{th} step as:
$\epsilon(k) = x(k) - x^*$
$\epsilon(k) = Hx(k-1) + d - x^*$
Moreover: $x^* = Hx^* + d$
And: $\epsilon(k-1) = x(k-1) - x^*$
So: $\epsilon(k) = Hx(k-1) + d - Hx^* - d$
$= H(x(k-1) - x^*)$
$= H\epsilon(k-1)$
$= H^k\epsilon(0) = (\lambda_1^k G_1 + ... + \lambda_s^k G_s)\epsilon(0)$
$\approx \lambda_1^k G_1 \epsilon(0)$ (In first order)

with G_i = Spectral projection occurring in the spectral decomposition, defined such that:

$$G_i = \frac{\prod_{j=1, j\neq i}^{k} (H - \lambda_j I)}{\prod_{j=1, j\neq i}^{k} (\lambda_i - \lambda_j)}$$

So, $\forall\ i\ \frac{\epsilon_i(k-1)}{\epsilon_i(k)} \approx \frac{1}{|\lambda_1|}$
In our case, $\lambda_i = (1 - r)$
In conclusion, the rate of convergence at the first order is governed by: $\frac{1}{(1-r)}$
In other words, the greater r is, the higher the rate of convergence is. This can be observed in Fig. S4 (see section 2.A).

B. MultiXrank parameters.

In order to simplify the use of MultiXrank, the user may choose default parameters. These default parameters intend to produce a homogeneous exploration of the multilayer network. They are based on three structural properties of the multilayer networks: the number of multiplex networks (N), the number of layers in each multiplex network (n), and the number of seeds in each multiplex network (α).
It is important to note that the number of top nodes (K) given back by MultiXrank is chosen by the user. This parameter K is an integer between 1 and the total number of nodes. If the user wants all the nodes to be scored, the K variable must be equal to 0.

B.1. Default parameter. : r

The parameter r corresponds to the global restart probability. By default, we set this parameter to 0.7, which is a value often used in the literature (2).

B.2. Default parameter. : δ

The parameter δ is associated with the probability to jump from one layer to another layer in a given multiplex network.
In the case of a monoplex network, δ is equal to 0. The random walker stays in the layer with a probability equal to 1 (1-δ). However, if the random walker explores a multiplex network, the probability is defined as equal to 0.5. In other words, the random walker has the same probability to jump from one layer to another than to stay in the layer. It is to note that, if several multiplex networks are considered, δ_i corresponds to the δ parameter of the multiplex network i and is equal to 0.5 for each multiplex network.

B.3. Default parameter. : τ

The parameter τ is associated with the probability to restart in a specific layer of a multiplex network (Fig. 2).
The homogeneous choice is $\tau_{ij} = \frac{1}{n_i}\ \forall j$, where τ_{ij} represent the probability to restart in the layer j within the multiplex network i, and n_i is the number of layers of the multiplex network i.

B.4. Default parameter. : η

The parameter η is associated with the probability to restart in a specific multiplex network.
Let us define N the number of multiplex networks, n_i the number of layers in the multiplex network i, and α_i the number of seeds in the multiplex network i. It is important to remind that $\tau_{ij} = \frac{1}{n_i}$ [1].
To have a homogeneous exploration, the parameter η is constrained by the following equations:

$$\alpha_k \sum_{i=1}^{n_k} \eta_k \tau_{ki} = \alpha_l \sum_{i=1}^{n_l} \eta_l \tau_{li} \quad \forall k, l \text{ [2]}$$

$$\sum_{j=1}^{N} \alpha_j [\sum_{i=1}^{n_j} \eta_j \tau_{ji}] = 1 \text{ [3]}$$

Both equations [2] and [3] can be simplified thanks to [1] as:

$$\alpha_k \eta_k = \alpha_l \eta_l \quad \forall k, l \text{ [4]}$$

$$\sum_{j=1}^{N} \alpha_j \eta_j = 1 \text{ [5]}$$

To solve this set of equations, we can use an algebraic point of view and intend to solve $\mathbf{X} = A^{-1}\mathbf{Y}$ in order to determine $\mathbf{X} = [\eta_1, \eta_2, ..., \eta_N]^T$:

$$\mathbf{Y} = \begin{pmatrix} 1 \\ 0 \\ ... \\ 0 \end{pmatrix} \; ; \; A = \begin{pmatrix} 1 & 1 & ... & 1 & 1 \\ \alpha_1 & -\alpha_2 & 0 & ... & 0 \\ \alpha_1 & 0 & -\alpha_3 & ... & 0 \\ & & ... & & \\ \alpha_1 & 0 & ... & 0 & -\alpha_N \end{pmatrix}$$

It is important to note that if some α_i terms are equal to zero, the size of the linear system is reduced to take only the non-zero α_i terms. In other words, if there are N_s non-zero α_i, the linear system to solve for the matrix A has a size equal to $N_s{*}N_s$. There are N_s terms η_i associated with the N_s non-zero α_i, and $(N - N_s)$ terms η_i associated with zero α_i which are equal to zero.

B.5. Default parameter. : λ

The parameter λ is associated with the probability to jump from one multiplex network to another one. Let us define N the number of multiplex networks, n_i the number of layers in the multiplex network i, and $\zeta_i = \frac{1}{n_i}$ the reverse of the number of layers in the multiplex network i. ζ_i is defined as a weight to penalize networks with a large number of layers. To have a homogeneous exploration, the parameter λ is constrained by the following equations:

$\sum_{i=1}^{N} \lambda_{ij} = 1 \quad \forall j$

$\zeta_k \lambda_{km} = \zeta_l \lambda_{lm} \quad \forall m, k, l \mid m = k, k \neq l$

To solve this set of equations, we can use an algebraic point of view and intend to solve $\mathbf{X} = A^{-1}\mathbf{Y}$, in order to determine $\mathbf{X} = [\lambda_{11}, \lambda_{21}, .., \lambda_{N1}, \lambda_{12}, .., \lambda_{N2}, \lambda_{13}, ..., \lambda_{1N}, \lambda_{2N}, .., \lambda_{NN}]^T$:

$$\mathbf{Y} = \begin{pmatrix} 1 \\ .. \\ 1 \\ 0 \\ ... \\ 0 \end{pmatrix} \; ; \; \mathrm{A} = \begin{pmatrix} 1_1 & & & & 0 \\ & 2_1 & & & \\ & & ... & & \\ & 0 & & ... & \\ & & & & N_1 \\ \boxed{1_2} & \boxed{0} & \boxed{0} & ... & \boxed{0} \\ \boxed{0} & \boxed{2_2} & \boxed{0} & ... & \boxed{0} \\ & & ... & & \\ \boxed{0} & \boxed{0} & ... & \boxed{0} & \boxed{N_2} \end{pmatrix}$$

$$\boxed{i_1} = \begin{pmatrix} 1 & 1 & ... & 1 & 1 \end{pmatrix} ; \boxed{i_2} = \begin{pmatrix} -\zeta_1 & 0 & 0 & ... & 0 & \zeta_i & 0 & ... & 0 & 0 \\ 0 & -\zeta_2 & 0 & ... & 0 & \zeta_i & 0 & ... & 0 & 0 \\ 0 & 0 & -\zeta_3 & ... & 0 & \zeta_i & 0 & ... & 0 & 0 \\ & & & ... & & & & & & \\ 0 & 0 & 0 & ... & 0 & \zeta_i & 0 & ... & 0 & -\zeta_N \end{pmatrix} ; \boxed{0} = \text{matrix of zero}$$

The size of $\boxed{i_1}$ is $1 * N$, the sizes of $\boxed{i_2}$ and $\boxed{0}$ are $N * (N-1)$, $\forall i \in [|1, N|]$
The size of $\mathbf{Y}$ is N^2, there are N times the value 1 and $N * (N-1)$ times the value 0.
It is important to note that $\mathbf{X}$ is a vector with a size equal to N^2 but it is necessary to convert it into a matrix of size $N * N$, because the λ parameter is a $N * N$ matrix.

C. Toy model of a universal multilayer network composed of three multiplex networks and three bipartite networks.

We propose here the full description of a toy model (Fig. S1) composed of three multiplex networks and their six associated bipartite networks. We detail all the matrices and the parameters chosen for the RWR exploration of this universal multilayer network (Fig. S2).

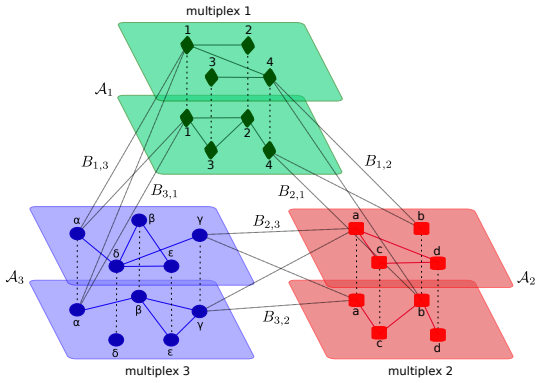


Fig. S1. Toy model of a universal multilayer network composed of three multiplex networks (green, red and blue multiplex networks). Each multiplex network contains different types of nodes (denoted 1 to 4, α to ϵ, and a to d, respectively). Their corresponding Supra-adjacency matrices are denoted by $\mathcal{A}_i$. The three multiplex networks are linked by six bipartite networks (represented here as bipartite interactions for the sake of visualization). The corresponding Bipartite network matrices are denoted by $B_{i,j}$. It is to note that a connection between a node i in the multiplex network α and node j in multiplex network β imposes the creation of edges between all replicas of node i present in the different layers of the multiplex network α and all replicas of node j present in the different layers of multiplex network β. All the edges of the universal multilayer networks can be weighted and/or directed. This figure is the same as Fig. 1. from the main manuscript.

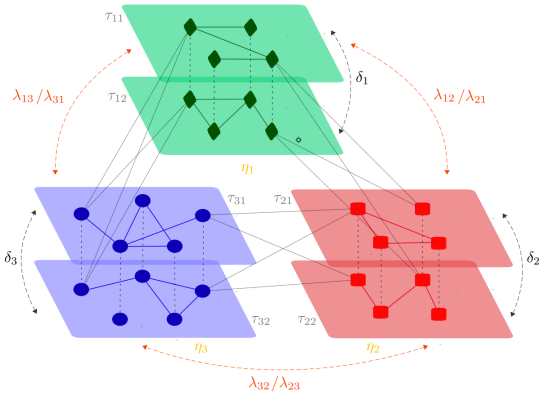


Fig. S2. RWR parameters of the toy model composed of three multiplex networks and their associated bipartite networks.

C.1. Supra-adjacency matrices of the toy model.

Let's define:

$\mathcal{A}_1$ = Supra-adjacency matrix of the multiplex network 1
$A_1^{[1]}$ = Adjacency matrix of the first layer of multiplex network 1
$A_1^{[2]}$ = Adjacency matrix of the second layer of multiplex network 1

$\mathcal{A}_2$ = Supra-adjacency matrix of the multiplex network 2
$A_2^{[1]}$ = Adjacency matrix of the first layer of multiplex network 2
$A_2^{[2]}$ = Adjacency matrix of the second layer of multiplex network 2

$\mathcal{A}_3$ = Supra-adjacency matrix of the multiplex network 3
$A_3^{[1]}$ = Adjacency matrix of the first layer of multiplex network 3

$A_3^{[2]}$ = Adjacency matrix of the second layer of multiplex network 3

I_4 = Identity matrix matrix of size n_4 (4*4)

I_5 = Identity matrix matrix of size n_5 (5*5)

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad I_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [1]$$

$$\mathcal{A}_1 = \begin{bmatrix} A_1^{[1]} & I_4 \\ I_4 & A_1^{[2]} \end{bmatrix} \qquad A_1^{[1]} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad A_1^{[2]} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad [2]$$

$$\mathcal{A}_2 = \begin{bmatrix} A_2^{[1]} & I_4 \\ I_4 & A_2^{[2]} \end{bmatrix} \qquad A_2^{[1]} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad A_2^{[2]} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad [3]$$

$$\mathcal{A}_3 = \begin{bmatrix} A_3^{[1]} & I_5 \\ I_5 & A_3^{[2]} \end{bmatrix} \qquad A_3^{[1]} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad A_3^{[2]} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad [4]$$

C.2. Bipartite network matrices of the toy model.

The bipartite networks are undirected, so the Bipartite network matrices are symmetric:

$B_{2,1} = B_{1,2}^T$ and $b_{2,1} = b_{1,2}^T$
$B_{3,1} = B_{1,3}^T$ and $b_{3,1} = b_{1,3}^T$
$B_{3,2} = B_{2,3}^T$ and $b_{3,2} = b_{2,3}^T$

$$B_{1,2} = \begin{bmatrix} b_{1,2} & b_{1,2} \\ b_{1,2} & b_{1,2} \end{bmatrix} \qquad b_{1,2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad [5]$$

$$B_{1,3} = \begin{bmatrix} b_{1,3} & b_{1,3} \\ b_{1,3} & b_{1,3} \end{bmatrix} \qquad b_{1,3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad [6]$$

$$B_{2,3} = \begin{bmatrix} b_{2,3} & b_{2,3} \\ b_{2,3} & b_{2,3} \end{bmatrix} \qquad b_{2,3} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad [7]$$

$$\mathcal{S} = \begin{bmatrix} A_1^{[1]} & I_4 & b_{1,2} & b_{1,2} & b_{1,3} & b_{1,3} \\ I_4 & A_1^{[2]} & b_{1,2} & b_{1,2} & b_{1,3} & b_{1,3} \\ b_{2,1} & b_{2,1} & A_2^{[1]} & I_4 & b_{2,3} & b_{2,3} \\ b_{2,1} & b_{2,1} & I_4 & A_2^{[2]} & b_{2,3} & b_{2,3} \\ b_{3,1} & b_{3,1} & b_{3,2} & b_{3,2} & A_3^{[1]} & I_5 \\ b_{3,1} & b_{3,1} & b_{3,2} & b_{3,2} & I_5 & A_3^{[2]} \end{bmatrix} \quad [8]$$

C.3. Random Walk with Restart (RWR) applied to the toy model.

We included the RWR framework into the Supra-adjacency matrices in order to take into account the possibility to jump from one layer to another inside a multiplex network. The parameter δ_k control the probability to jump from one layer to another inside the multiplex network k.

$$\mathcal{A}_1 = \begin{bmatrix} (1-\delta_1)A_1^{[1]} & \delta_1 I_4 \\ \delta_1 I_4 & (1-\delta_1)A_1^{[2]} \end{bmatrix} \quad [9]$$

$$\mathcal{A}_2 = \begin{bmatrix} (1-\delta_2)A_2^{[1]} & \delta_2 I_4 \\ \delta_2 I_4 & (1-\delta_2)A_2^{[2]} \end{bmatrix} \quad [10]$$

$$\mathcal{A}_3 = \begin{bmatrix} (1-\delta_3)A_3^{[1]} & \delta_3 I_5 \\ \delta_3 I_5 & (1-\delta_3)A_3^{[2]} \end{bmatrix} \quad [11]$$

$$\mathcal{S} = \begin{bmatrix} (1-\delta_1)A_1^{[1]} & \delta_1 I_4 & b_{1,2} & b_{1,2} & b_{1,3} & b_{1,3} \\ \delta_1 I_4 & (1-\delta_1)A_1^{[2]} & b_{1,2} & b_{1,2} & b_{1,3} & b_{1,3} \\ b_{2,1} & b_{2,1} & (1-\delta_2)A_2^{[1]} & \delta_2 I_4 & b_{2,3} & b_{2,3} \\ b_{2,1} & b_{2,1} & \delta_2 I_4 & (1-\delta_2)A_2^{[2]} & b_{2,3} & b_{2,3} \\ b_{3,1} & b_{3,1} & b_{3,2} & b_{3,2} & (1-\delta_3)A_3^{[1]} & \delta_3 I_5 \\ b_{3,1} & b_{3,1} & b_{3,2} & b_{3,2} & \delta_3 I_5 & (1-\delta_3)A_3^{[2]} \end{bmatrix} \quad [12]$$

C.4. Initial probability distribution.

The initial probability distribution $\mathbf{p}_0^T$ is associated with the restart probability distribution. It corresponds to the distribution of probabilities to restart on specific seeds. We can write the vector $\mathbf{p}_0^T$ as follows:

$$\mathbf{p}_0^T = \begin{bmatrix} \eta_1 \bar{\mathbf{v}}_0^1 \\ \eta_2 \bar{\mathbf{v}}_0^2 \\ ... \\ \eta_N \bar{\mathbf{v}}_0^N \end{bmatrix} \quad [13]$$

with η_k the probability to restart in a specific multiplex network k, and $\bar{\mathbf{v}}_0^k$ the initial probability distribution of the multiplex network k. The initial probability distribution of the multiplex network k size is equal to $L_k * n_k$, with L_k the number of layers of the multiplex network k and n_k the number of nodes in the multiplex network k. We constraint the parameter η with the standard normalization condition, $\sum_{k=1}^{3} \eta_k = 1$. In this toy model application, we choose the default parameters $\eta_1 = \eta_2 = \eta_3 = 1/3$, i.e equal probability to restart in each multiplex network.
In the toy model, we choose as seeds the node 1 of multiplex network 1, a of multiplex network 2, and α of multiplex network 3. We need to add other parameters to take into account the restart probability in the different layers of each multiplex network. These parameters are defined as τ parameters. τ_{11} (resp. τ_{12}) is the probability to restart in the first layer (resp. the second layer) of the first multiplex network, τ_{21} (resp. τ_{22}) is the probability to restart in the first layer (resp. the second layer) of the second multiplex network, and τ_{31} (resp. τ_{32}) is the probability to restart in the first layer (resp. the second layer) of the third multiplex network. We added the η parameters to $\bar{\mathbf{v}}_0^i$ such as $\bar{\mathbf{v}}_0^k = [\tau_{k1}\mathbf{v}_0^k, \tau_{i2}\mathbf{v}_0^k, ..., \tau_{iL_k}\mathbf{v}_0^k]^T$, with L_k = number of layers in multiplex network k.

In this toy model application, we used the default parameters, i.e. an equal probability to restart in each layer of each multiplex network:

$\tau_{11} = \tau_{12} = 1/2$
$\tau_{21} = \tau_{22} = 1/2$
$\tau_{31} = \tau_{32} = 1/2$

Thus,

$$\mathbf{p}_0^T = \begin{bmatrix} \eta_1 \bar{\mathbf{v}}_0^1 \\ \eta_2 \bar{\mathbf{v}}_0^2 \\ \eta_3 \bar{\mathbf{v}}_0^3 \end{bmatrix} \qquad \bar{\mathbf{v}}_0^1 = \begin{pmatrix} \tau_{11} * 1 \\ 0 \\ 0 \\ 0 \\ \tau_{12} * 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \bar{\mathbf{v}}_0^2 = \begin{pmatrix} \tau_{21} * 1 \\ 0 \\ 0 \\ 0 \\ \tau_{22} * 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \bar{\mathbf{v}}_0^3 = \begin{pmatrix} \tau_{31} * 1 \\ 0 \\ 0 \\ 0 \\ \tau_{32} * 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad [14]$$

Finally, this initial probability distribution $\mathbf{p}_0^T$ is normalized.

C.5. Normalization of the Supra-heterogeneous adjacency matrix (S).

The last part concerns the inter-multiplex network parameters λ, which are linked to the normalization of the Supra-heterogeneous adjacency matrix. We need to convert the Supra-heterogeneous adjacency matrix into a Stochastic matrix to ensure convergence (see section 1.A.1):

$$\widehat{S} = \begin{bmatrix} \widehat{S}_{11} & \widehat{S}_{12} & \widehat{S}_{13} \\ \widehat{S}_{21} & \widehat{S}_{22} & \widehat{S}_{23} \\ \widehat{S}_{31} & \widehat{S}_{32} & \widehat{S}_{33} \end{bmatrix} \quad [15]$$

We will use the formula described below with $\alpha \in [|1,3|]$ and $\beta \in [|1,3|]$:

$$\widehat{S}_{\alpha\alpha}(i_\alpha, j_\alpha) = \begin{cases} \frac{\mathcal{A}_\alpha(i_\alpha, j_\alpha)}{\sum_{k_\alpha=1}^{n_\alpha} \mathcal{A}_\alpha}(i_\alpha, k_\alpha) & \text{if } \forall\, \beta : \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta) = 0 \\ \frac{(1-\sum_{\beta=1}^{c_{i_\alpha}} \lambda_{\alpha\beta}) * \mathcal{A}_\alpha(i_\alpha, j_\alpha)}{\sum_{k_\alpha=1}^{n_\alpha} \mathcal{A}_\alpha}(i_\alpha, k_\alpha) & \text{Otherwise} \end{cases} \quad [16]$$

$$\widehat{S}_{\alpha\beta}(i_\alpha, j_\beta) = \begin{cases} \frac{\lambda_{\alpha\beta} B_{\alpha,\beta}(i_\alpha, j_\beta)}{\sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta)} & \text{if } \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta) \neq 0 \\ \frac{\frac{\lambda_{\alpha\beta}}{\sum_{\beta=1}^{c} \lambda_{\alpha\beta}} \sum_{i_\alpha=1}^{c} B_{\alpha,\beta}(i_\alpha, j_\beta)}{\sum_{i_\alpha=1}^{c} \sum_{k_\beta=1}^{n_\beta} B_{\alpha,\beta}(i_\alpha, k_\beta)} & \text{if } i_\alpha \text{ not in } M_\alpha \\ 0 & \text{Otherwise} \end{cases} \quad [17]$$

So, if we describe each term, we obtain:

$$\widehat{S}_{11}(i_1, j_1) = \begin{cases} \frac{\mathcal{A}_1(i_1, j_1)}{\sum_{k_1=1}^{n_1} \mathcal{A}_1}(i_1, k_1) & \text{if } \forall\, \beta : \sum_{k_\beta=1}^{n_\beta} B_{1,\beta}(i_1, k_\beta) = 0 \\ \frac{(1-\sum_{\beta=1}^{c_{i_1}} \lambda_{\alpha\beta})) \mathcal{A}_1(i_1, j_1)}{\sum_{k_1=1}^{n_1} \mathcal{A}_1}(i_1, k_1) & \text{Otherwise} \end{cases} \quad [18]$$

$$\widehat{S}_{22}(i_2, j_2) = \begin{cases} \frac{\mathcal{A}_2(i_2, j_2)}{\sum_{k_2=1}^{n_2} \mathcal{A}_2}(i_2, k_2) & \text{if } \forall\, \beta : \sum_{k_\beta=1}^{n_\beta} B_{2,\beta}(i_2, k_\beta) = 0 \\ \frac{(1-\sum_{\beta=1}^{c_{i_2}} \lambda_{\alpha\beta}) \mathcal{A}_2(i_2, j_2)}{\sum_{k_2=1}^{n_2} \mathcal{A}_2}(i_2, k_2) & \text{Otherwise} \end{cases} \quad [19]$$

$$\widehat{S}_{33}(i_3, j_3) = \begin{cases} \frac{\mathcal{A}_3(i_3, j_3)}{\sum_{k_3=1}^{n_3} \mathcal{A}_3}(i_3, k_3) & \text{if } \forall\, \beta : \sum_{k_\beta=1}^{n_\beta} B_{3,\beta}(i_3, k_\beta) = 0 \\ \frac{(1-\sum_{\beta=1}^{c_{i_3}} \lambda_{\alpha\beta}) \mathcal{A}_3(i_3, j_3)}{\sum_{k_3=1}^{n_3} \mathcal{A}_3}(i_3, k_3) & \text{Otherwise} \end{cases} \quad [20]$$

$$\widehat{S}_{12}(i_1, j_2) = \begin{cases} \frac{\lambda_{12} B_{1,2}(i_1, j_2)}{\sum_{k_2=1}^{n_2} B_{1,2}(i_1, k_2)} & \text{if } \sum_{k_2=1}^{n_2} B_{1,2}(i_1, k_2) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad [21]$$

$$\widehat{S}_{13}(i_1, j_3) = \begin{cases} \frac{\lambda_{13} B_{1,3}(i_1, j_3)}{\sum_{k_3=1}^{n_3} B_{1,3}(i_1, k_3)} & \text{if } \sum_{k_3=1}^{n_3} B_{1,3}(i_1, k_3) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad [22]$$

$$\widehat{S}_{23}(i_2, j_3) = \begin{cases} \frac{\lambda_{23} B_{2,3}(i_2, j_3)}{\sum_{k_3=1}^{n_3} B_{2,3}(i_2, k_3)} & \text{if } \sum_{k_3=1}^{n_3} B_{2,3}(i_2, k_3) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad [23]$$

We computed $\widehat{S}_{21}$, $\widehat{S}_{23}$, and $\widehat{S}_{31}$ with similar formulas.
In this toy model application, we choose the default parameters, which are an equal probability to jump from one multiplex network to another or to stay in the same multiplex network.

$\lambda_{11} = \lambda_{21} = \lambda_{31} = 1/3$
$\lambda_{12} = \lambda_{22} = \lambda_{32} = 1/3$
$\lambda_{13} = \lambda_{23} = \lambda_{33} = 1/3$

D. Comparison with previously published formulas for RWR on heterogeneous networks.

A previous study about RWR on heterogeneous networks (i.e., two monoplex networks connected by a bipartite network) was published by Li and Patra (3). The challenging point concerned the Transition rate matrix and more precisely the normalization to obtain this Transition rate matrix. The normalization is a *sine qua non* condition because it is mandatory to assure the convergence to the steady-state (see section 1.A.1). Let us define $\mathcal{H}$, the Adjacency matrix of the heterogeneous networks, and $\mathcal{A}_1, \mathcal{A}_2$ the Adjacency matrices of the monoplex networks. In addition, let us define $B_{1,2}, B_{2,1}$, the Bipartite network matrices. If the bipartite network is undirected $B_{2,1} = (B_{1,2})^T$:

$$\mathcal{H} = \begin{bmatrix} \mathcal{A}_1 & B_{1,2} \\ B_{2,1} & \mathcal{A}_2 \end{bmatrix} \quad [24]$$

As previously, we can defined the heterogeneous graph G_H as sets of nodes and edges:

$$\begin{cases} G_H = (V_H, E_H) \\ V_H = \{v_{1,i}, i = 1, ..., n_1\} \cup \{v_{2,i}, i = 1, ..., n_2\} \\ E_H = \{e^{1,1}_{i,j}, i, j = 1, ..., n_1, (A_1^{[\alpha_k]})_{i,j} \neq 0\ ,\ \alpha_k = 1, ..., L_k\} \\ \qquad \cup \{e^{2,2}_{i,j}, i, j = 1, ..., n_2, (A_2^{[\alpha_k]})_{i,j} \neq 0\ ,\ \alpha_k = 1, ..., L_k\}\} \\ \qquad \cup \{e^{1,2}_{i,j}, i = 1, ..., n_1, j = 1, ..., n_2, (B^{[1,2]}_{i,j}) \neq 0\} \end{cases} \quad [25]$$

The normalization of this Adjacency matrix allowing to obtain the Transition rate matrix is based on the transformation described in Li and Patra (3).

To adopt the same notation as in Li and Patra (3), we defined:

$\alpha = G$
$\beta = P$
$\lambda = \lambda_{PG} = \lambda_{GP}$
$(1 - \lambda) = \lambda_{PP} = \lambda_{GG}$
$\widehat{S}_{GG} = M_G$
$\widehat{S}_{PP} = M_P$
$\widehat{S}_{GP}(i_G, j_P) = (M_{GP})_{i,j}$

$\widehat{S}_{PG}(i_P, j_G) = (M_{PG})_{i,j}$
$B_{G,P} = B$
$B_{P,G} = B_{G,P}^T = B^T$

With the previous notation and our general formulation of the normalization the following equations, we obtain:

$$\widehat{S}_{GG}(i_G, j_G) = \begin{cases} \frac{A_G(i_G, j_G)}{\sum_{k_G=1}^{n_G} A_G(i_G, k_G)} & \text{if } \sum_{k_G=1}^{n_G} B_{G,P}(i_G, k_P) = 0 \\ \frac{\lambda_{GG} A_G(i_G, j_G)}{\sum_{k_G=1}^{n_G} A_G(i_G, k_G)} & \text{Otherwise} \end{cases} \quad [26]$$

$$\widehat{S}_{PP}(i_P, j_P) = \begin{cases} \frac{A_P(i_P, j_P)}{\sum_{k_P=1}^{n_P} A_P(i_P, k_P)} & \text{if } \sum_{k_P=1}^{n_P} B_{P,G}(i_P, k_G) = 0 \\ \frac{\lambda_{PP} A_P(i_P, j_P)}{\sum_{k_P=1}^{n_P} A_P(i_P, k_P)} & \text{Otherwise} \end{cases} \quad [27]$$

$$\widehat{S}_{GP}(i_G, j_P) = \begin{cases} \frac{\lambda_{GP} B_{G,P}(i_G, j_P)}{\sum_{k_P=1}^{n_P} B_{G,P}(i_G, k_P)} & \text{if } \sum_{k_P=1}^{n_P} B_{G,P}(i_G, k_P) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad [28]$$

$$\widehat{S}_{PG}(i_P, j_G) = \begin{cases} \frac{\lambda_{PG} B_{P,G}(i_P, j_G)}{\sum_{k_G=1}^{n_G} B_{P,G}(i_P, k_G)} & \text{if } \sum_{k_G=1}^{n_G} B_{P,G}(i_P, k_G) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad [29]$$

These formulas are the ones obtained in the Li and Patra (3).
Another study applied RWR to a network composed of a multiplex network and a heterogeneous layer (2). If we choose the same notations, we also retrieve the same formulas (data not shown).
Overall, our approach can reproduce the results obtained in previous studies on RWR exploration of (multiplex)-heterogeneous networks.

2. Supplementary Note 2: MultiXrank numerical framework

MultiXrank is the Python package implementing our method that enables Random Walk with Restart on any kind of multilayer network (Fig. S3).

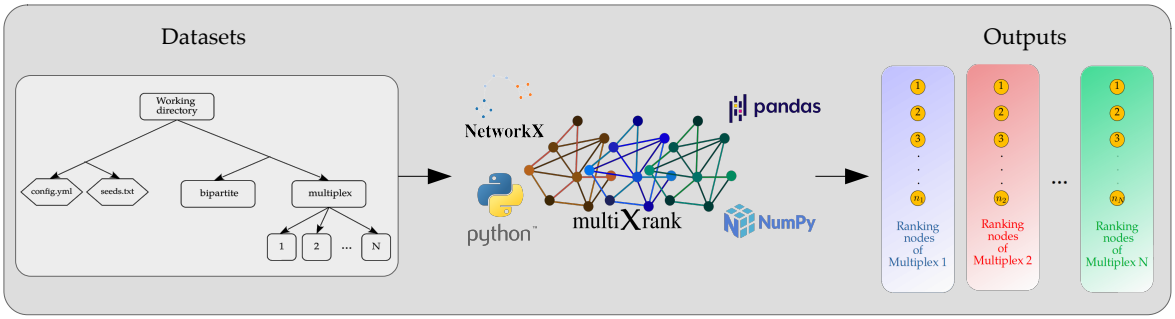


Fig. S3. Workflow of the MultiXrank Python package. The network and the parameter datasets need to be ordered as in the tree file in the left panel. This tree file is used by MultiXrank to run RWR on the multilayer network defined in the bipartite and multiplex folders, with the parameters given in the config and seeds files. MultiXrank returns the output scores in an output folder, as shown in the right panel. A Conda environment is available on GitHub (https://github.com/anthbapt/multixrank) to install all dependencies needed by MultiXrank.

A. Information on Time complexity.

Determining the time complexity of the algorithm depending on the input parameters is an essential point to evaluate the running time. We here present some information about MultiXrank complexity. Indeed, some fundamental functions of MultiXrank are based on a high-level Python library, from which the complexity is hard to estimate precisely. Thereby, the exact time complexity of MultiXrank is difficult to estimate. We used the standard bounded Landau notation O to represent the dependencies of the function to the input parameters in our code. However, we still give the dependencies on the input parameters that drive the running time of the different functions of MultiXrank class. The most important part concerns the random_walk_restart function that computes the Markovian iteration process:

$$\mathbf{p}_{t+1}^T = (1-r)\widehat{S}\mathbf{p}_t^T + r\mathbf{p}_0^T.$$

At each iteration, we compute a sparse matrix-vector multiplication and a sum of vectors until the convergence to the steady-state. These processes have a time complexity of $O(m)$ and $O(1)$, with m the number of edges in the Transition rate matrix. So this function has a total complexity of $O(\delta m)$, where δ is the number of iterations.
In conclusion, the time complexity of this part is driven by the number of edges, if the number of nodes is negligible compared to the number of edges, as already described in W.Jin et al. (4).

We computed the average running time over 10 runs for the different functions of the total algorithm. To this goal, we used the multilayer network described in Table S1. The results are listed in Table S2, where N is the number of layers, n the number of nodes, and m the number of edges.
We can identify five time-consuming functions: read_layers, get_supra_adj_multiplex, get_bipartite, get_transition, and random_walk_restart. These five functions represent 98.5% of the running time.
If we make the hypothesis that the number of nodes is negligible compared to the number of edges ($m >> n$), the total complexity of MultiXrank is $O\Delta m$, with Δ a constant. The complexity of MultiXrank is thereby depending on the total number of edges.

Remarks:

1. A remaining question concerns the evolution of the running time according to the input parameters. To partially answer this question, we display the evolution of the running time in function of the global restart probability (r) (from 0.01 to 0.99). We can see that a correlation exists between MultiXrank running time and the RWR function (random_walk_restart) running time (Fig. S4). This correlation reveals that the global restart parameter (r) does not change the running time of the functions, except for the random_walk_restart function.

2. Moreover, the running time increases with the inverse of the global restart probability (r). This can be explained as the more the parameter r decreases, the more we explore the network. This is an expected result given by the elements of proof about the rate of convergence in the section 1.A.2.

multiplex network	Layer	Nodes	# Interactions
1	PPI	genes	143 653
1	Complexes	genes	63 561
1	Reactome	genes	194 500
2	Diseases Similarity	diseases	29 200
3	Clinical drug interactions	drugs	14 822
3	Experimental drug interactions	drugs	737
3	Predicted drug interactions	drugs	2 080
3	Pharmacological drugs interactions	drugs	48 514

multiplex network	# Nodes
1	16 947
2	7 039
3	1 559

Bipartite networks	# Interactions
1-2	1 135 037
2-3	1 418 248
1-3	30 895

Table S1. Networks chosen to test time complexity.

B. Input requirements.

B.1. File tree.

To use MultiXrank on universal multilayer networks, one needs to respect a file tree (Fig. S5). The user needs to choose a working directory and create two folders and two files. The files are defined as follows: a file dedicated to the input parameters (config.yml) and a second file dedicated to the seeds (seeds.txt). Then, two folders are dedicated to the networks files. The different networks are separated into the bipartite networks folder and the multiplex networks folder.
In the multiplex networks folder, the different multiplex networks are each associated with a different folder, named: $1, 2, ..., N$. Inside these folders, the user needs to put all the layers of the multiplex networks, with a key-name reported in the input parameters file (config.yml).
In the bipartite networks folder, the user needs to put all the bipartite networks with a specific name described as follows: NumberOfMultiplexNetwork_NumberOfOtherMultiplexNetwork.tsv. For example, for a bipartite network between the multiplex network 1 and the multiplex network 2, the name will be 1_2.tsv and for a

Function	Running time	Time complexity
multiplex networks_length	$2.15 * 10^{-6}$ s	$O(1)$
check_parameters	$1.02 * 10^{-3}$ s	$O(1)$
read_layers	$7.40 * 10^{-1}$ s	$O(\delta_1 m)$
pool_of_nodes	$9.33 * 10^{-3}$ s	$O(n)$
check_seeds	$7.79 * 10^{-4}$ s	$O(N)$
add_missing_nodes	$2.12 * 10^{-2}$ s	$O(n)$
get_number_nodes	$1.38 * 10^{-4}$ s	$O(N)$
get_supra_adj_multiplex	1.60 s	$O(\delta_2 m)$
get_bipartite	6.84 s	$O(\delta_3 m)$
get_transition	3.96 s	$O(\delta_4 m)$
get_seed_scores	$4.57 * 10^{-3}$ s	$O(N)$
random_walk_restart	$6.55 * 10^{-1}$ s	$O(\delta_5 m)$
ranking	$2.40 * 10^{-1}$ s	$O(n)$
top_K_rank	$6.94 * 10^{-4}$ s	$O(n)$
total	14.0 s	$O(\Delta m + \Gamma n) \sim O(\Delta m)$

Table S2. Different functions associated with their running time and elements of time complexity for MultiXrank applied to the multilayer network described in Table S1.

bipartite network between the multiplex network 2 and the multiplex network 1, it will be 2_1.tsv. In the case of a directed bipartite network, these two bipartite networks are different. To simplify the creation of the file tree, a python script named hierarchy.py is available on Github (https://github.com/anthbapt/multixrank). This script creates all the folders needed (for a specific number of multiplex networks) in a chosen working directory.

B.2. Network files.

We use the standard edgelist format, with a tab separation between each column (Fig. S6). The extension used is TSV, but any extension with edgelist format can be chosen.

B.3. Seeds file.

This file contains all the seed nodes. Each row contains one seed, with no preferential order.

B.4. Input parameters file.

We choose the YML format for the input parameters file in order to facilitate the import of each parameter into a user-friendly Python type. This file is divided into two parts. The first one is dedicated to MultiXrank parameters $(r, \delta, \tau, \lambda, \eta, K)$, and the second one is dedicated to the network property parameters used by MultiXrank.

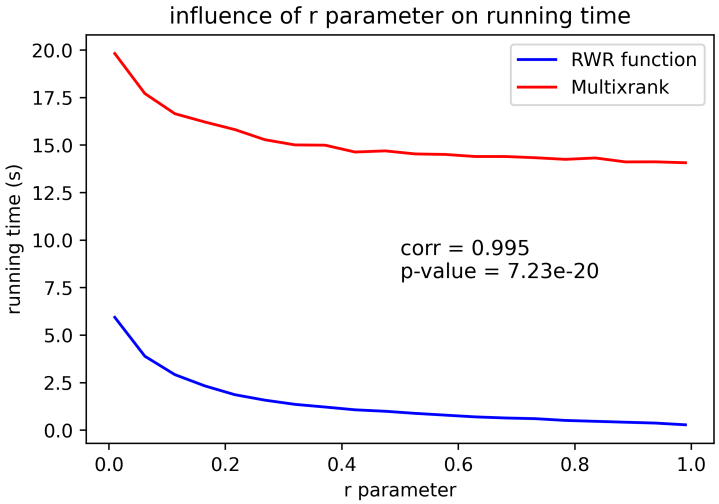


Fig. S4. Running time of MultiXrank (red) and of the RWR function (random_walk_restart, blue) depending on the value of the global restart probability (r), the variation of the running time of MultiXrank is correlated (Spearman correlation equal to 0.995) with the running time of the RWR function.

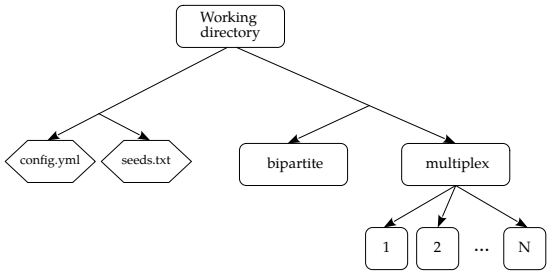


Fig. S5. File tree of Network data and parameters needed to run MultiXrank.

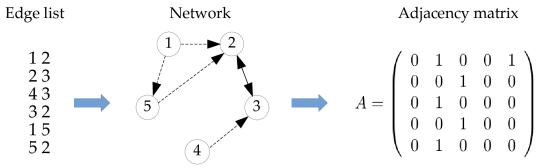


Fig. S6. Network file format and representations. The edgelist format is used to input networks in MultiXrank.

Parameters of the RWR:

- Global restart probability: $r \rightarrow$ float
- Inter-layers jump probability: $\delta \rightarrow$ np.array (1D)
  δ_i = probability to jump across the different layer of the multiplex network i
- Layers restart probability: $\tau \rightarrow$ np.array (2D)
  τ_{ij} = probability to restart in the layer j of the multiplex network i
- Inter-multiplex networks jump probability: $\lambda \rightarrow$ np.array (2D)
  λ_{ij} = probability to jump from the multiplex network i to the multiplex network j
- multiplex network restart probability: $\eta \rightarrow$ np.array (1D)
  η_i = probability to restart in the multiplex network i
- Number of top nodes: $K \rightarrow$ int (0 for all nodes)
  This parameter K is an integer between 1 and the total number of nodes. If the user wants all the nodes to be scored, the K variable must be equal to 0.

Parameters for Network properties:

- seed: Name of the seeds file.
- self_loops: Boolean value to take into account the self-loops in the network datasets for the network python object. It is preferable to choose self_loops equal to 0 (without self-loops). Some nodes are present in the bipartite networks but absent from the corresponding monoplex/multiplex network. In order to avoid normalization problems (column of zeros), we have to add artificially these nodes to the multiplex networks by adding self-loops in one of the layers.
- Multiplex:
  - layers: list of the paths to the layers of the multiplex network considered
  - graph_type: types of the different interactions in the considered multiplex network according to the Table S3.
- Bipartite: list of the information concerning each bipartite network: the source (source multiplex network number), the target (target multiplex network number), and the type of the bipartite network (graph_type) according to the Table S3.

Notations for the type of interactions:

The notation to specify the different types of interactions in each multiplex and bipartite network (i.e., graph_type) is based on a Boolean code (Table S3), where the first digit is dedicated to the weight and the second one to the directed property.

name	directed	weighted	Boolean code
undirweighted	0	0	00
weighted	0	1	01
directed	1	0	10
dirweighted	1	1	11

Table S3. Boolean code for the different types of interactions (edges) in multiplex and bipartite networks.

B.5. Example for a multilayer network composed of two multiplex networks:.

Remarks:

If undirected bipartite networks are considered, the Bipartite network matrices are symmetric. The example in Fig. S7 belongs to this case, so Bipartite_interactions is equivalent to $[[0, "00"], ["00", 0]]$.

```yaml
# Name of the seeds file
seed: seeds.txt
# Global Restart Probability (r)
r: 0.7
# Multiplex Networks Restart Probability (eta)
eta: [1/2, 1/2]
# Inter Multiplex Networks Jump Probability (lambda)
lamb:
    - [1/2, 1/2]
    - [1/2, 1/2]

multiplex:
    1:
        layers:
            - multiplex/1/PPI.tsv
            - multiplex/1/Reactome.tsv
            - multiplex/1/Complexes.tsv
        # Inter Layers Jump Probability (delta) : for multiplex 1
        delta: 0.5
        # Graph type: (un)weighted, (un)directed : for multiplex 1
        graph_type: [00, 00, 00]
        # Layers Restart Probability (tau) : for multiplex 1
        tau: [1/3, 1/3, 1/3]
    2:
        layers:
            - multiplex/2/disease_disease_final.tsv
        # Inter Layers Jump Probability (delta) : for multiplex 2
        delta: 0
        # Graph type: (un)weighted, (un)directed : for multiplex 2
        graph_type: [00]
        # Layers Restart Probability (tau) : for multiplex 2
        tau: [1]

bipartite:
    # Bipartite network information
    bipartite/1_2.tsv: {source: 1, 'target': 2, graph_type: 00}
```

Fig. S7. Example of an input parameters file for a multilayer network composed of a multiplex and a monoplex network (gene and disease networks).

3. Supplementary Note 3: multilayer networks

The different networks described in Table S4 are available on GitHub in edgelist format: https://github.com/anthbapt/multixrank

System	# Multiplexes	# Layers in each multiplex	Nodes	# Nodes	Edges	# interactions	Bipartites
Toy model	3	2, 2, 2	\	4 ; 4 ; 5	\	1 : 7 ; 2 : 6 ; 3 : 9	1-2 : 1 ; 1-3 : 1 ; 2-3 : 1
Airport	3	3, 3, 3	French, UK, German Airports	18 ; 28 ; 23	Company connections	1 : 15, 5, 9 2 : 23, 43, 9 3 : 41, 35, 12	1-2 : 51 1-3 : 39 2-3 : 61
Biological system	3	3, 1, 4	Genes, Diseases, Drugs	16 967 ; 7 039 ; 1 559	PPI, Complexes, Reactome ; Diseases similarity ; drug1, drug2, drug3, drug4	1 : 143 653, 63 561, 194 500 2 : 29 200 3 : 14 822 ; 737 ; 2 080 ; 48514	1-2 : 1 135 037 1-3 : 30 895 2-3 : 1 418 248

Table S4. The different multilayer network systems used in the manuscript and their associated properties.

A. Airport networks.

The different multiplex networks were generated from the connections between airports given by data published in Cardillo et al. (5). The mapping of the airports to their respective countries has been done thanks to the database https://openflights.org/data.html.
The companies were chosen based on their number of connections. We took the top-3 most connected companies in each country.

1. French airports multiplex network
   - FR3: Easyjet connections between French airports.
   - FR7: Air France connections between French airports.
   - FR26: Netjets connections between French airports.
2. British airports multiplex network
   - UK3: Easyjet connections between British airports.
   - UK15: Flybe connections between British airports.
   - UK26: Netjets connections between British airports.
3. German airports multiplex network
   - G1: Lufthansa connections between German airports.
   - G6: Air Berlin connections between German airports.
   - G24: Germanwings connections between German airports.

The bipartite networks correspond to the transnational flights operated by these different companies between France and UK, France and Germany, and UK and Germany.

B. Biological networks.

1. Gene multiplex network: This multiplex network is composed of three layers of interactions between gene nodes. Please note that we consider here genes and proteins indifferently. We choose the gene names for the nodes of this multiplex network and its associated bipartite networks.

- Complexes: A molecular complexes layer constructed from the fusion of Hu.map (6) and Corum (7), using OmniPathR (8).
- PPI: A Protein-Protein interaction (PPI) layer corresponding to the fusion of 3 datasets: APID (homo sapiens level 2, without inter-species interactions), Hi-Union and Lit-BM (www.interactome-atlas.org/download). It is important to notice that 166 over 13346 APID proteins do not match to any gene name. We choose to use the Uniprot IDs in these cases.
- Reactome: A pathways layer extracted from NDEx (9) and corresponding to human Reactome data (10).

2. Disease monoplex network: The network was built based on phenotypic proximities between diseases calculated with the information content. The whole protocol is described in (2). We choose the UMLS annotation for diseases in this monoplex network and its associated bipartite networks.
3. Drug multiplex network: The data come from Cheng et al. (11)† and pharmacological drugs interaction network from snap.stanford.edu‡. We used the DrugBank name format for the drugs in this multiplex network and its associated bipartite networks.
   - drug1†: Clinical drug interactions (clinically reported adverse drug–drug interactions, 14822 clinically reported adverse drug–drug interactions connecting 667 drugs were retained)
   - drug2†: Experimental drug combinations (experimentally validated drug combinations, 737 unique pairwise drug combinations connecting 376 drugs were retained)
   - drug3†: Predicted drug combinations (network-predicted hypertensive drug combinations, 2080 potential combinations involving 65 hypertensive drugs were retained)
   - drug4‡: Drug-Drug interactions determined from the pharmacologic effect of the action of one drug on another drug (48514 interactions involving 1514 drugs).

Bipartite networks are defined as follows:

- Gene-Disease (1-2): The bipartite network come from the 2020 version (v7.0) of DisGeNET (12). This network is weighted with score computed according to the formula reported in https://www.DisGeNET.org/dbinfo.
- Gene-Drug (1-3): Several bipartite networks are available: drug-target association from DrugBank Release Version 5.1.8 go.drugbank.com/releases/latest, drug-target associations from drugcentral release v10.12 drugcentral.org/download, and the drug-target association from Cheng et al (11). We merged all of these bipartite interactions to obtain the gene-drug bipartite network.
- Disease-Drug (2-3): The disease-drug association network come from repoDB from (13).

It is important to note that some nodes are present only in the bipartite networks but absent from the corresponding monoplex/multiplex networks. We have to artificially add these nodes to the multiplex networks by adding self-loops in one of the layers (see supplementary section 2.B.4). They will then be ranked by MultiXrank.

4. Supplementary Note 4: Evaluation protocols

Importantly, the two evaluation protocols presented here can be used to compare the performances or to evaluate the predictive power of RWR using different combinations of network layers in a multilayer network framework. In other words, they can be used to test the signal-to-noise ratio of a multilayer system.

A. Leave-One-Out Cross-Validation (LOOCV) protocol.

To evaluate the performances of the RWR on different combinations of networks, we designed a Leave-One-Out Cross-Validation (LOOCV) protocol inspired by F.Mordelet and J.P.Vert (14) and A.Valdeolivas et al. (2). The supplementary Figure S9 presents a LOOCV workflow on an example multilayer network composed of two monoplex networks linked by a bipartite network. For the real biological multilayer network evaluated in Figure 3, we considered gene-disease associations with a score superior or equal to 0.3 from DisGeNET v7.0. We further considered only diseases associated with at least two genes. Overall, these gene-disease associations are a subset of the gene-disease bipartite network constructed from DisGeNET, which is part of the multilayer biological network (described in section 3.B). We focused on diseases associated with at least two genes. For each of these diseases, we selected the set of genes. In each gene set, the genes are considered as the left-out genes one-by-one. Each time a gene is left-out, the remaining genes associated with the same disease are used as seed(s), as well as the disease node when the disease network is considered for the RWR exploration. The RWR algorithm is then applied, and all the network nodes are scored and ranked according to their proximity to the seed(s). The rank of the gene that was left-out in the current run is recorded. The process is repeated iteratively for all the left-out genes. Finally, the Cumulative Distribution Function (CDF) of the ranks of the left-out genes is plotted. The CDF displays the percentage of left-out genes that are ranked within the top-K gene nodes. The CDFs are used to evaluate and compare the performance of the RWR applied to different combinations of networks: multiplex networks, multilayer networks composed of two or three monoplex/multiplex networks...

Importantly, the LOOCV to evaluate the RWR performances on the monoplex PPI network and the gene multiplex network (composed of only of gene nodes) only use gene nodes as seed(s). The LOOCV on heterogeneous and multilayer network composed of two or three monoplex/multiplex networks and including both gene and disease node types use both gene and disease nodes as seeds. It is to note that in this case, in order to simulate an unknown association, we also removed the bipartite interaction linking the left-out gene node and the disease node of the current run from the bipartite network. Doing so, we avoid the artificial top-ranking of the left-out gene nodes.

For the evaluations using the airports universal multilayer network, the left-out nodes are the French airports. The LOOCV focuses on all the British airports associated with at least two French airports. The French airports are removed one-by-one and considered as the left-out nodes. The remaining French airport associated with the same British airport are used as seed(s) for the RWR. When the British airport network is considered in the evaluation, the British airport node is considered as a seed together with the French airports nodes that are not left-out in the current run.

B. Link Prediction (LP) protocol.

Another way to evaluate the performances of the RWR framework is to use a Link Prediction (LP) protocol. In LP, we use all the associations, including the nodes from the disease network associated with only one node from the gene network. We systematically remove associations between gene nodes and disease nodes from the bipartite network, and we predict the rank of the corresponding gene node using the disease node as a seed in the RWR. All the network nodes are scored and ranked according to their proximity to the seed. The rank of the gene node from the association that was left-out in the current run is recorded. Finally, the Cumulative Distribution Function (CDF) of the ranks of the left-out gene nodes is plotted as described above.

The process is similar for the airport network, where we systematically remove associations between French and British airport nodes. We here predict the rank of the corresponding French airport using the British

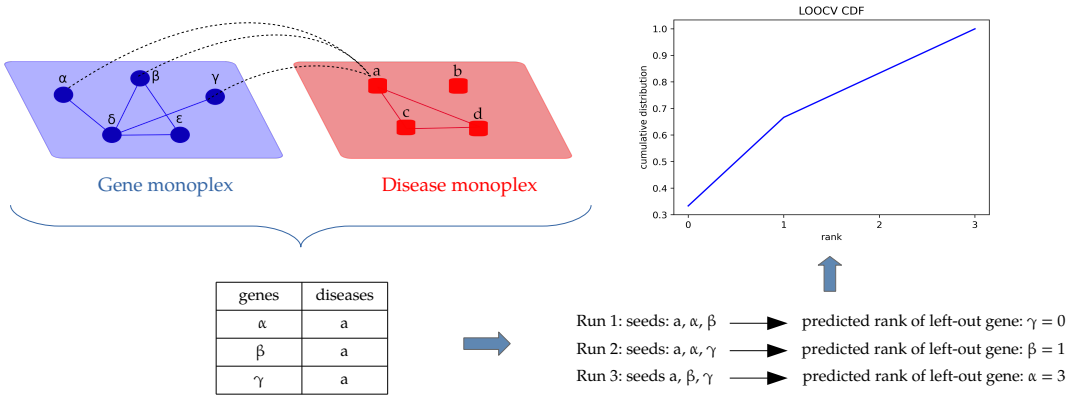


Fig. S8. Workflow of the Leave-One-Out Cross-Validation (LOOCV) protocol. The example is a heterogeneous network composed of two monoplex networks: a gene monoplex network and a disease monoplex network. Both networks are connected by a bipartite network containing three bipartite interactions. The bipartite interactions link the gene nodes α, β, and γ to the disease node a. For the LOOCV, we considered all the disease nodes linked to at least two genes. In the example case, we hence selected the interactions displayed in the table. Each gene is left-out one by one, for each disease iteratively. The other genes associated with the given disease and the disease node itself are considered as seeds for the RWR. In the example, we have one disease associated with three genes. We intend to evaluate the performance of RWR exploring both the gene and the disease monoplex networks. Hence, each gene node α, β, and γ will be left-out, and the remaining genes plus the disease node a will be considered as seeds in the RWR. The rank of the left-out gene in a given RWR run is recorded. The results are displayed as a Cumulative Distribution Function (CDF).

airport node as seed.

C. Artificial increase of the connectivity in bipartite networks.

We observed that adding new multiplex networks into a multilayer system may not systematically increase the predictive power of the RWR, as evaluated with LOOCV or LP. The reason is that the walk from one node to another may not exist or may correspond to a very long path. For instance, takingan example from Fig. S10, if we want to predict a gene-disease link between a node in the disease monoplex network and a node in the gene multiplex network, we need to have bipartite connections between neighboring nodes. The data from the third (drug) multiplex network could help predicting such gene-disease association, if short walks from the disease node to the drug multiplex network and from the drug multiplex network to the gene multiplex network exist. If such walks do not exist or if they are too long, the addition of the third multiplex network does not increase the predictive power.

We checked if the gene and disease nodes present in the gene-disease bipartite network are also present in the gene-drug and disease-drug bipartite networks. We observed that the overlaps between the different bipartite networks are very low (Fig. S10). We hypothesize that these low overlaps explain why the addition of the drug multiplex network does not increase the predictive power of the RWR in biological multilayer networks (Fig. 3.A.1 for LOOCV and Fig. S11.A.1 for LP). However, we can see an increase in the predictive power for the airport multilayer when a third multiplex network is considered (Fig. 3.A.2 and S11.A.2). The bipartite networks of the airport multilayer network display high node overlaps (Fig. S10).

To check this hypothesis, we artificially increased the connectivity of the bipartite networks between the drug multiplex network and the gene multiplex network, and between the drug multiplex network and the disease monoplex network. To do so, we added artificial transit drug nodes linking diseases to genes (Fig. S12). In the airport multilayer network, we artificially increased the connectivity of the bipartite networks between the German airports multiplex network and the French airports multiplex network, and between the German airports multiplex network and the British airports multiplex network. To do so, we added artificial transit German nodes liking British to French airports (Fig. S12). We show that that these artificially added transit nodes increase drastically the prediction performance for both LOOCV (Fig. 3.B.1-2) and LP (Fig. S11.B.1-2).

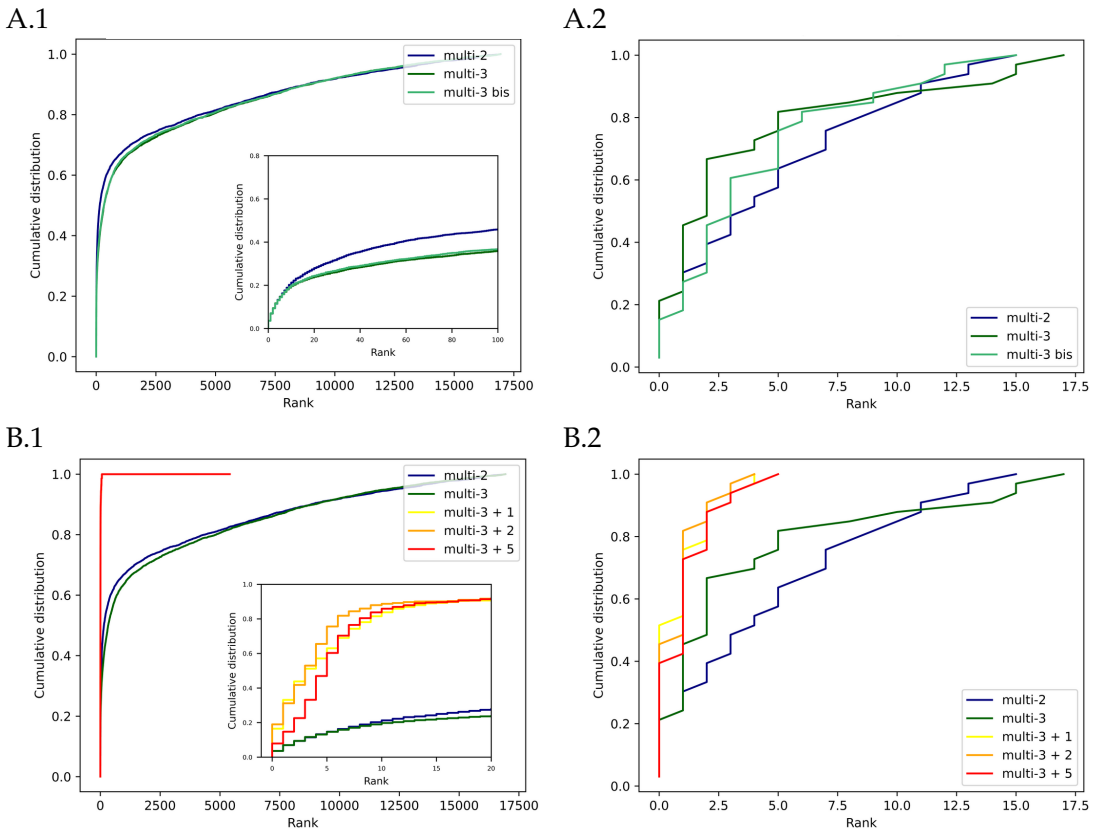


Fig. S9. A.1-2: Cumulative Distribution Functions (CDFs) representing the ranks of the left-out nodes in the LP protocol. A.1: focus on different combinations of biological networks: multilayer network composed of the gene multiplex network and the disease monoplex network (multi-2), and multilayer network composed of the gene and the drug multiplex networks and the disease monoplex network, for two different sets of parameters (multi-3, multi-3 bis). A.2: focus on different combinations of airports networks: French and British airports multiplex networks (multi-2), and French, British, and German airports multiplex networks, for two different sets of parameters (multi-3, multi-3 bis). The multiplex networks are connected with the bipartite networks described in the Evaluations section. B.1-2: Cumulative Distribution Functions (CDFs) representing the ranks of the left-out nodes in the LP protocol for the 3-multiplex networks described previously with artificially increased connectivity in the gene-drug and disease-drug bipartite networks. The connectivity is artificially increased thanks to the addition of 1 (multi3+1), 2 (multi3+2) or 5 (multi3+5) transit nodes for each gene-disease association (B.1). In the airport multilayer network, the connectivity is artificially increased in the French-German and British-German bipartite networks thanks to the addition of 1 (multi3+1), 2 (multi3+2) or 5 (multi3+5) transit German nodes for each French-British airports association (B.2). The parameters of the RWR are detailed in Table S5-S6.

D. Perturbations in the context of artificially increased connectivity.

To confirm that artificially highly connected bipartite networks increase the predictive power of the RWR, we need to check if the reverse process decreases the predictive power. In other words, if perturbations of the bipartite networks designed previously (Fig. S12) will change the predictive power. To do so, we shuffled the nodes in the bipartite networks. This approach is similar to using random networks as bipartite networks. By replacing highly connected bipartite networks by random bipartite networks, we expect to reduce the predictive power until observing the same results as obtained with only two multiplex networks.
Step by step, we replaced parts of the bipartite networks with shuffled (random) networks. We started with 10% of random links, then 20%, then 50%, and finally 100% of random links. We observed very clearly that the more the bipartite networks are perturbed, the more the predictive power decrease. At the end, with fully shuffled bipartite networks, we observed results similar to the one obtained with only two multiplex networks. The details of the results are given for the airports multilayer network in Fig. S13.A (LOOCV) and Fig S14.A (LP) and for the biological multilayer network in the Fig. S13.B (LOOCV) and Fig. S14.B (LP).

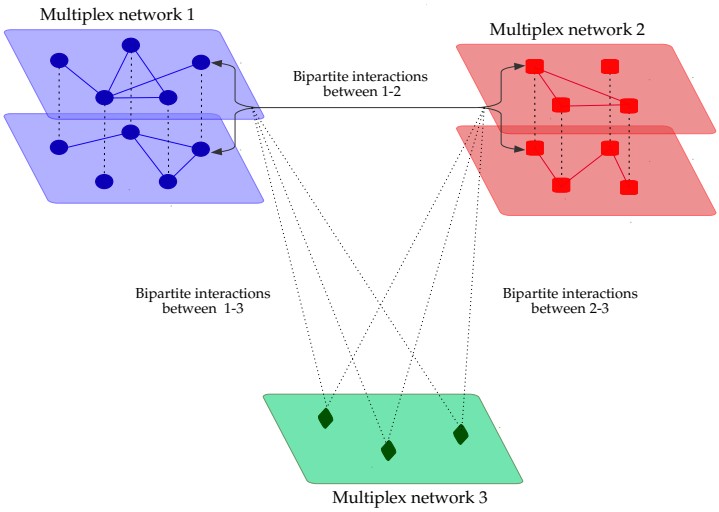


Fig. S10. Strategy used to increase artificially the connectivity of the bipartite networks. For instance, for the biological network, we increased artificially the gene-drug and disease-drug bipartite networks. We added drug nodes as transit nodes (green nodes in the green layer) linking diseases to genes. For each association between the multiplex network 1 (blue) and multiplex network 2 (red), we simulate the addition of 1, 2, 3, and 5 transit nodes (here, the addition of three transit nodes are represented for one association between multiplex network 1 and 2). A similar strategy is applied for the airport multilayer network.

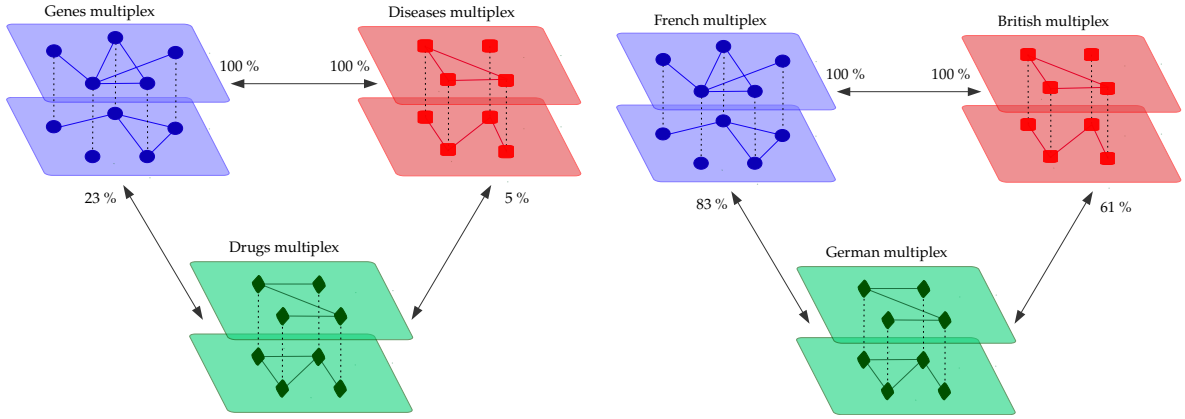


Fig. S11. Overlaps between the different bipartite network nodes. We took as reference all the gene and disease nodes present in the gene-disease bipartite network (i.e., these genes corresponds to 100% of the genes and these diseases to 100% of the diseases). We then checked for the presence of these genes in the gene-drug bipartite network (23% of the genes from the gene-disease bipartite network are present in the drug-gene bipartite network), and for the presence of these diseases in the drug-disease bipartite network (5% of the diseases from the gene-disease bipartite network are present in the disease-drug bipartite network).Similar interpretation should be given to the airports multilayer network. In this case, we took as references all the nodes present in the bipartite network connecting the French and the British multiplex networks.

5. Supplementary Note 5: Exploration of parameter space

Our exploration of the parameter space focuses on two different multilayer networks. The first one is a small airports multilayer network composed of three multiplex networks. These three multiplex networks represent a French airports multiplex network, a British airports multiplex network, and a German airports multiplex network. In each multiplex network, the nodes correspond to airports, and the edges to the national flight connections between these airports. All these multiplex networks are linked by bipartite networks that correspond to transnational flight (for details see supplementary section 3.A). The second multilayer network

r	δ	τ	λ	η
0.7	0.5	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$	0	1

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 0 & 1/3 \\ 0 & 1/3 & 1/3 \\ 2/3 & 2/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$

Table S5. Parameters of the RWR for the different airport networks used for LOOCV and LP protocols. 1) The French multiplex network (multi-1). 2) The multilayer network composed of the French and British airports multiplex networks (multi-2). 3-4) The multilayer network composed of the French, British, and German airports multiplex networks for two different sets of parameters (multi-3, multi-3 bis). All parameter sets expect multi-3 correspond to defaults parameters.

is a large biological multilayer network composed of a gene multiplex network, a disease monoplex network, and a drug multiplex network. Each multiplex network is connected to the others thanks to bipartite networks association (for details see supplementary section 3.B).
To better understand the stability of the output scores upon variations of the input parameters, we propose a protocol based on 5 successive steps.

A. First step: Definition of the sets of parameters.

The choice of the sets of input parameters is delicate. Indeed, for each parameter, we need to define step sizes small enough to explore all the space. But not too small to avoid a drastic increase in the running time. We decided to explore only some parameters. Indeed, the global restart (r) is not expected to affect the solutions (3, 15). In addition, we imposed the probability to restart in specific multiplex networks (η) because it only depends on seed(s) that do not change across the exploration of the parameter space. We also choose to impose the probability to restart in specific layers on the multiplex network (τ). Finally, the column normalization of the matrix of the probability of jumping from one multiplex network to another (λ) imposes the value of the diagonal elements. This value is equal to one minus the sum of all the other elements of the same column.
Overall, we decided to explore the anti-diagonal terms of the λ parameter and the parameter that control the probability to jump from one layer to another in a specific multiplex network, (δ).
We run MultiXrank on each set of parameters, from 0.1 to 1 for the δ parameter and from 0 to 0.9 for the λ parameters. We defined the step sizes according to the size of the multilayer networks, in to avoid running time soaring. Each set of parameters gives output scores for each node in each multiplex network.
The seed chosen for the airport multilayer network exploration is the node '7'. The seed chosen for the biological multilayer network exploration is the gene 'LMNA'.

r	δ	τ	λ	η
0.7	0.5	1	0	1

r	δ	τ	λ	η
0.7	0.5	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$	0	1

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 0 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\begin{bmatrix} 1/2 & 0 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/4 & 1/3 \\ 1/4 & 1/2 & 1/3 \\ 1/4 & 1/4 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}$

Table S6. Parameters of the RWR for the different biological networks used for LOOCV and LP protocols. 1) The protein-protein interactions network (PPI). 2) The gene multiplex network (multi-1). 3) The multilayer network composed of the gene multiplex network and the disease monoplex network (multi-2). 4-5) The multilayer network composed of the gene multiplex network, the drug multiplex networks and the disease monoplex network for two different sets of parameters (multi-3, multi-3 bis). All parameter sets expect multi-3 bis correspond to defaults parameters.

B. Second step: Construction of the Similarity matrix.

To measure the similarity between the distributions of the output scores of two sets of parameters, we can use two different types of information: the ranking of the node and its value. In other words, we have access to the ranking distribution and to the standard distribution. We first tried to combine the Kullback-Leibler measure with a measure about the ranking of the nodes. However, the Kullback-Leibler divergence drives the similarity measure from a distribution form point of view. This seems to be less efficient to identify clusters at step fourth than only using the rank information. We hence adopted a simple formulation for the similarity measure, based on a weighted ranking score:

Let us define two sets of parameters: γ, σ

N is the number of multiplex network

n_k is the number of nodes associated to the multiplex network k

$\mathbf{r}_\gamma^k$ resp. $\mathbf{r}_\sigma^k$ are the rank output scores distributions that associate each node with their rank in the RWR associated with the set of parameters γ (resp. σ), for the multiplex network k.

$\mathbf{r}_{\gamma\sigma}^k$ (resp. $\mathbf{r}_{\sigma\gamma}^k$) are the distributions that gives to each node of the output score distributions given by the RWR associated with the set of parameters γ (resp. σ) (in the multiplex network k) their rank in the distributions σ (resp. γ.)

r	δ	τ	λ	η
0.7	$\delta_1 = (0.1, 0.4, 0.7, 1)$ $\delta_2 = (0.1, 0.4, 0.7, 1)$ $\delta_3 = (0.1, 0.4, 0.7, 1)$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\lambda_{12} = (0.1, 0.3, 0.5, 0.7, 0.9)$ $\lambda_{13} = (0.1, 0.3, 0.5, 0.7, 0.9)$ $\lambda_{23} = (0.1, 0.3, 0.5, 0.7, 0.9)$ $\lambda_{21} = \lambda_{12}$ $\lambda_{23} = \lambda_{32}$ $\lambda_{31} = \lambda_{13}$ $\lambda_{11} = 1 - \lambda_{21} - \lambda_{31}$ $\lambda_{22} = 1 - \lambda_{12} - \lambda_{32}$ $\lambda_{33} = 1 - \lambda_{13} - \lambda_{23}$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\delta_1 = (0.1, 0.55, 1)$ $\delta_2 = (0.1, 0.55, 1)$ $\delta_3 = (0.1, 0.55, 1)$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$	$\lambda_{12} = (0.1, 0.5, 0.9)$ $\lambda_{13} = (0.1, 0.5, 0.9)$ $\lambda_{23} = (0.1, 0.5, 0.9)$ $\lambda_{21} = \lambda_{12}$ $\lambda_{23} = \lambda_{32}$ $\lambda_{31} = \lambda_{13}$ $\lambda_{11} = 1 - \lambda_{21} - \lambda_{31}$ $\lambda_{22} = 1 - \lambda_{12} - \lambda_{32}$ $\lambda_{33} = 1 - \lambda_{13} - \lambda_{23}$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$

r	δ	τ	λ	η
0.7	$\delta_1 = (0.1, 0.325, 0.55, 0.775, 1)$ $\delta_2 = (0.1, 0.325, 0.55, 0.775, 1)$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 \end{bmatrix}$	$\lambda_{12} = (0.1, 0.3, 0.5, 0.7, 0.9)$ $\lambda_{21} = \lambda_{12}$ $\lambda_{11} = 1 - \lambda_{21}$ $\lambda_{22} = 1 - \lambda_{12}$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$

Table S7. Top: Airport multilayer network composed of 3 multiplex networks British, German nodes). Corresponds to the 1472 different sets of parameters used for the Fig.S15. Middle: Biological multilayer network composed of 3 multiplex networks (gene, disease, drug nodes). Corresponds to the 108 different sets of parameters used for the Fig.S16. Bottom: Biological multilayer network composed of 2 multiplex networks (gene, disease nodes). Corresponds to the 125 different sets of parameters used for the Fig.S16. The parameters are explored with respect to the list defined for each δ_i and each λ_{ij}. For both Tables 1) and 2), the exploration need also to respect that $(\lambda_{12} + \lambda_{13}) < 1$, $(\lambda_{12} + \lambda_{23}) < 1$, and $(\lambda_{13} + \lambda_{23}) < 1$.

So, $(\mathbf{r}_{\alpha}^{k})_j$ is the rank of the node in position j of the distribution $\mathbf{r}_{\gamma}^{k}$, and $(\mathbf{r}_{\gamma\sigma}^{k})_j$ is the rank in the distribution $\mathbf{r}_{\sigma}^{k}$ of the node associated to the position j in the distribution $\mathbf{r}_{\gamma}^{k}$.

$$\Theta_{\gamma\sigma}^{k} = \sum_{j=1}^{n_k} \frac{\sqrt{\left(\frac{1}{[(\mathbf{r}_{\gamma}^{k})_j - (\mathbf{r}_{\gamma\sigma}^{k})_j]}\right)^2 + \left(\frac{1}{[(\mathbf{r}_{\sigma}^{k})_j - (\mathbf{r}_{\sigma\gamma}^{k})_j]}\right)^2}}{\left(\frac{(\mathbf{r}_{\gamma}^{k})_j + (\mathbf{r}_{\sigma}^{k})_j}{2}\right)^2} \quad [30]$$

$$\Theta_{\gamma\sigma} = \sqrt{\sum_{k=1}^{N} \frac{(\Theta_{\gamma\sigma}^{k})^2}{n_k}} \quad [31]$$

C. Third step: Projection of the Similarity matrix into the Principal Component Analysis space.

We aim to represent the similarity vectors, which are M-dimensional representations (with M the number of sets of parameters) into a unified and two-dimensional space. To this goal, we used Principal Component

Analysis (PCA). PCA is a suitable way to represent in low dimension the spreading of the data thanks to variance optimization.

D. Fourth step: Clustering of the PCA space into sub-regions of stability.

After representing the data in a two-dimension space, we need to cluster it. This this goal, we applied a k-means clustering algorithm on the nodes represented in the PCA space. The nodes gathered in a specific cluster are expected to correspond to similar output scores obtained from different sets of parameters.
The number of clusters chosen is an important parameter that has to be tuned depending on the dispersion of the data in the PCA space. The choice of the clustering method can also be discussed. We did not observed remarkable difference between k-means, spectral clustering, and agglomerative clustering. We hence adopted the simplest one.

E. Fifth step: Comparisons of the top-ranked nodes in the clusters.

After clustering the output scores on the PCA space, we need to evaluate if the output scores aggregated in the same cluster share a common ranking of the nodes. To this goal, we selected the top-K nodes (with K equal to 1, 3, 5, 10, 20, 100) of all the RWR outputs in each cluster (in the main manuscript, the results are presented for K equal to 5 and 100). We checked if these nodes are present in the outputs of all the set of parameters from a given cluster. The ideal case would be that all the nodes are the sames in the outputs of a given cluster.
A particular difficulty with multilayer networks comes from the different contributions of the different multiplex networks. Some multiplex networks may have a well-clustered PCA representation. Others can be scattered.

The projection into the PCA space of the output scores obtained from different sets of parameters shows different behavior depending on the network. For instance, in Fig. S15 we see that the two first components gather 72.2% of the total variance. We observe a scattered distribution of the output scores in the PCA space. This reflects a large variability in the results under variations of the input parameters. In this case, the parameters affect drastically the output scores. In this context, it could be interesting to use a consensus ranking by averaging all the ranks given by the different sets of parameters.A different behavior is observed in Fig. S16, where the two firsts components gathered 99.5% of the total variance. The cluster are well separated, and a specific behavior is observed inside each cluster. The top-ranked nodes are the same between each set of parameters belonging to a given cluster.
The two different behavior observed in Fig S15 and S16 hence reflect two very different behaviors possibly associated with different types of multilayer networks. On the one hand, multilayer networks associated with scattered behavior, with a high impact on the input parameters. On the other hand, multilayer network associated with stable behavior, with well-defined clusters and robustness to changes in input parameters in each cluster.

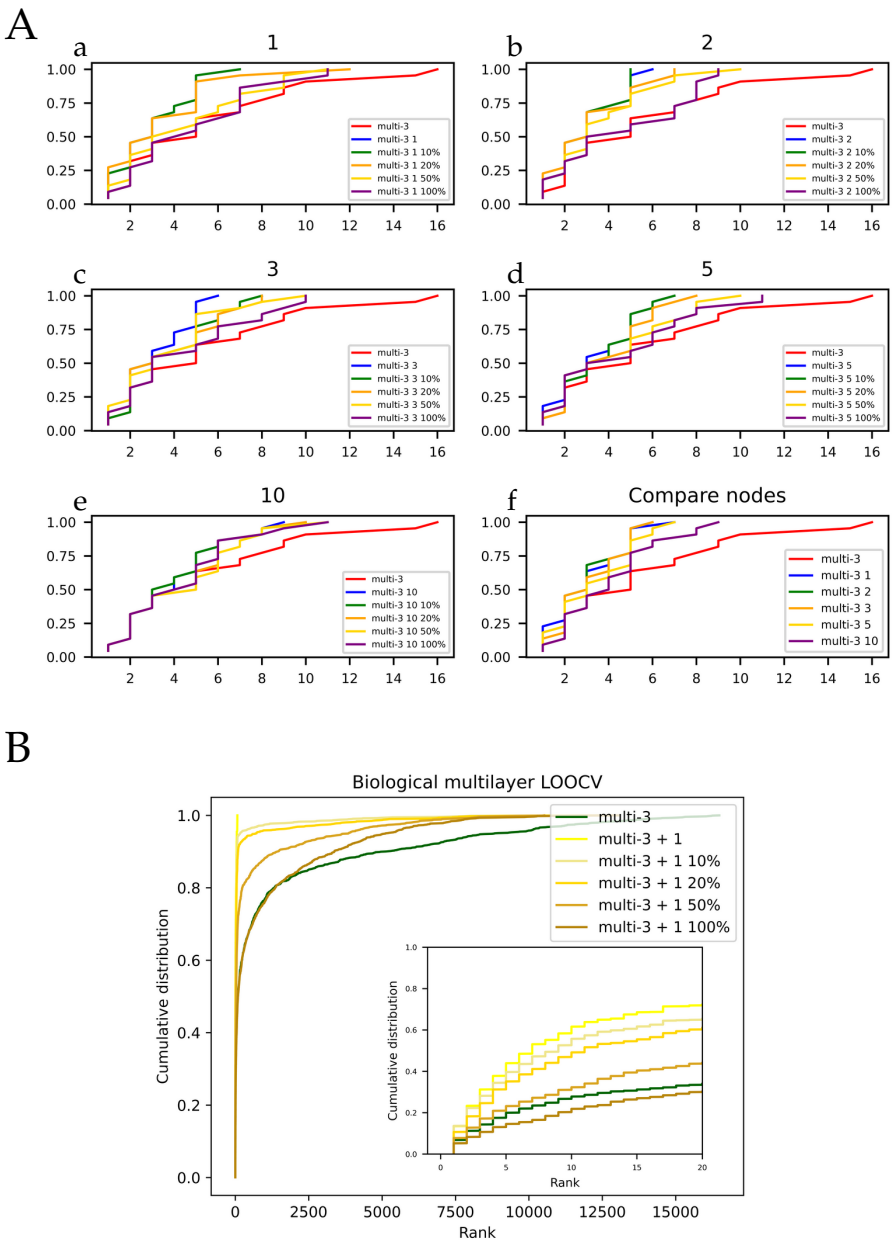


Fig. S12. Cumulative Distribution Functions (CDFs) representing the ranks of the left-out nodes during perturbations of the bipartite networks with artificially increased connectivity, for Leave-One-Out Cross-Validation (LOOCV). A: CDFs of the left-out nodes in the airports multilayer network, for 5 case studies of artificially increased connectivity. These 5 case studies correspond to the addition of 1, 2, 3, 5, or 10 transit nodes in the German airports connecting French-German and British-German airports. This addition of transit nodes allows to artificially increase the connectivity of the nodes of the bipartite networks between the French airports multiplex network and the British airports multiplex network (Fig.A.a-e). In each figure, we display the CDF associated with the multilayer networks composed of the three multiplex networks: French, British, and German multiplex networks (multi3), the CDF associated with the artificially increased connectivity, and the CDFs resulting from the perturbation of the French-German airports bipartite network and the British-German airports multiplex network with artificially increased the connectivity. We randomized, 10%, 20%, 50%, and 100% of the edges of these bipartite networks. Fig. A.f represents the CDFs for the 3-multiplex network associated with the 5 artificially increased connectivity cases described previously compared with the 3-multiplex network without transit nodes. B: CDFs of the left-out nodes in the biological multilayer network composed of the gene and drug multiplex networks, and the disease monoplex network. These multilayer network is considered without transit node (multi3) or with one transit node (which artificially increases the connectivity) in the gene-drug bipartite network and the disease-drug bipartite network for each gene-disease associations (multi3+1). Perturbations of the multi3+1 system are done with 10% of randomized edges in the gene-drug bipartite network and the disease-drug bipartite network (multi+1 10%), with 20% (multi+1 20%), with 50% (multi+1 50%), and with 100% (multi+1 100%).The parameters of the RWR are detailed in Table S5-S6.

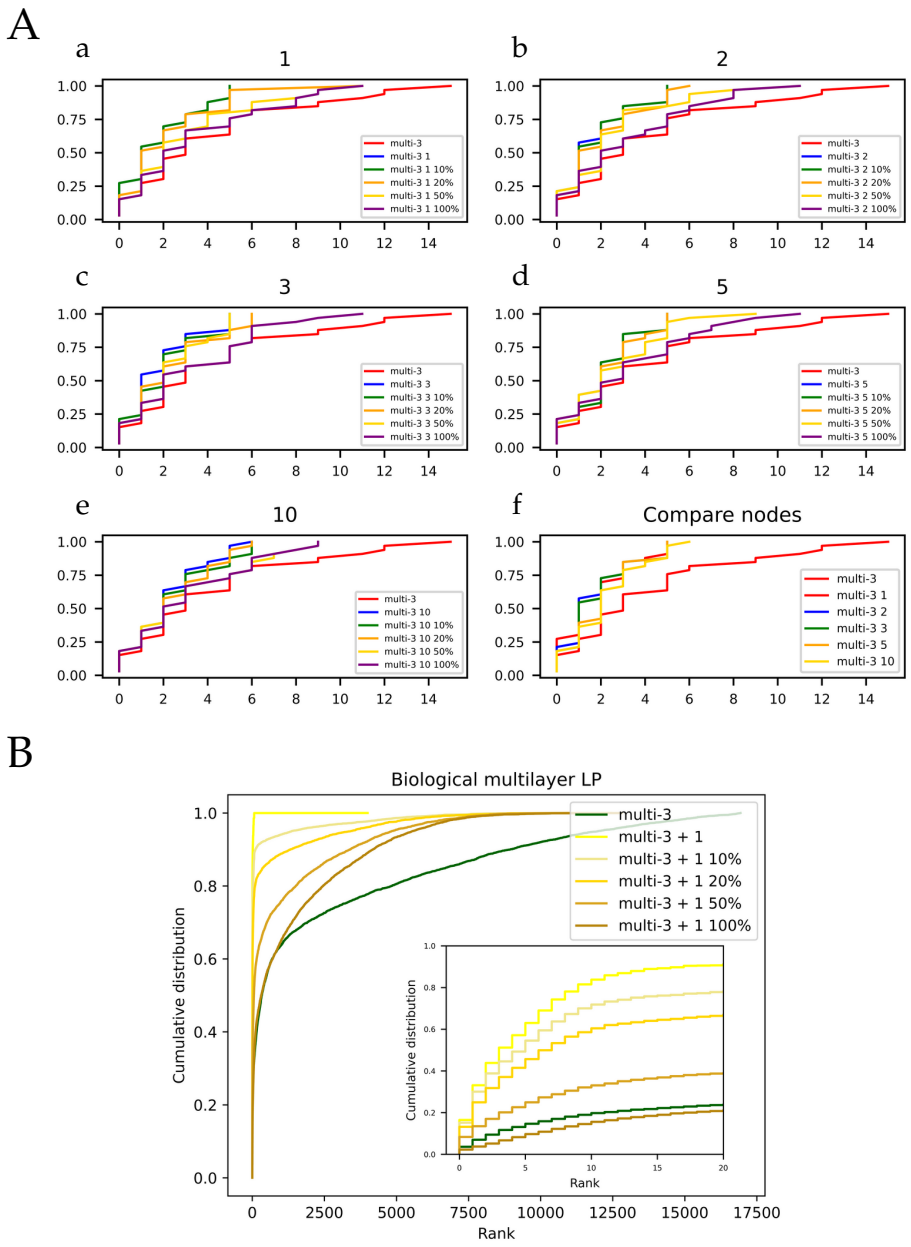


Fig. S13. Cumulative Distribution Functions (CDFs) representing the ranks of the left-out nodes during perturbations of the bipartite networks with artificially increased connectivity, for Link Prediction (LP). A: CDFs of the left-out nodes in the airports multilayer network, for 5 case studies of artificially increased connectivity. These 5 case studies correspond to the addition of 1, 2, 3, 5, or 10 transit nodes in the German airports connecting French-German and British-German airports bipartite networks. This addition of transit nodes allows to artificially increase the connectivity of the nodes of the bipartite networks between the French airports multiplex network and the British airports multiplex network (Fig.A.a-e). In each figure, we display the CDF associated with the multilayer networks composed of the three multiplex networks: French, British, and German multiplex networks (multi3), the CDF associated with the artificially increased connectivity, and the CDFs resulting from the perturbation of the French-German airports bipartite network and the British-German airports multiplex network with artificially increased connectivity. We randomized, 10%, 20%, 50%, and 100% of the edges of these bipartite networks. Fig. A.f represents the CDFs for the 3-multiplex network associated with the 5 artificially increased connectivity cases described previously compared with the 3-multiplex network without transit nodes. B: CDFs of the left-out nodes in the biological multilayer network composed of the gene and drug multiplex networks, and the disease monoplex network. This multilayer network is considered without transit node (multi3) or with one transit node (which artificially increases the connectivity) in the gene-drug bipartite network and the disease-drug bipartite network for each gene-disease associations (multi3+1). Perturbations of the multi3+1 system are done with 10% of randomized edges in the gene-drug bipartite network and the disease-drug bipartite network (multi+1 10%), with 20% (multi+1 20%), with 50% (multi+1 50%), and with 100% in these bipartite networks (multi+1 100%). The parameters of the RWR are detailed in Table S5-S6.

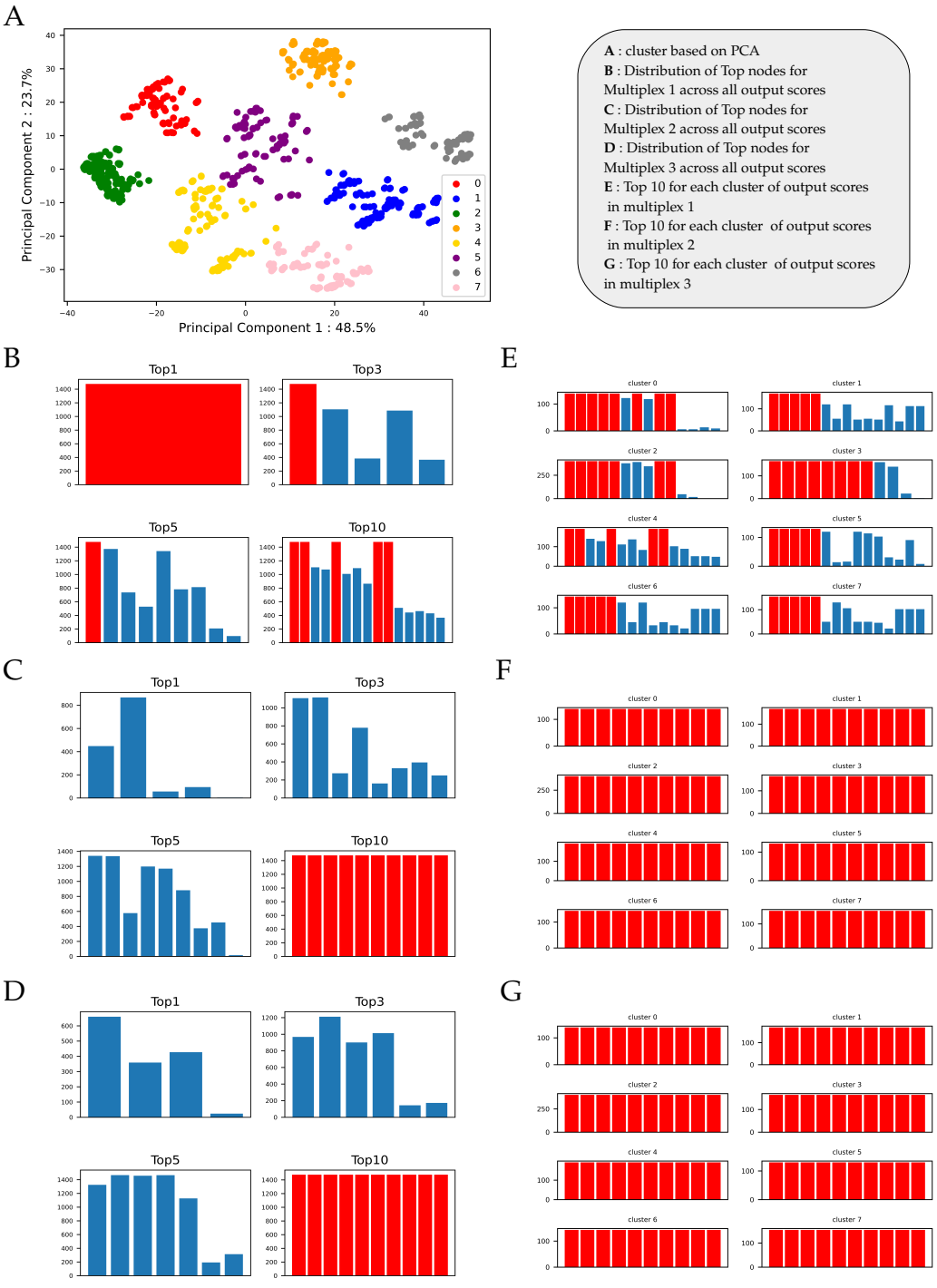


Fig. S14. A: Clustering in the PCA space of the output scores obtained with MultiXrank on the airports multilayer network (composed of the French, British, and German airports multiplex networks described in section 3.A) using 1472 different sets of parameters. B-D: Comparisons of the top-1, top-3, top-5, and top-10 nodes obtained for the French airports multiplex network (B), for the British airports multiplex network (C), and for the German airports multiplex network (D), for 1472 different sets of parameters. E-G: Top-10 nodes for the different sets of parameter in each cluster defined from the PCA space (A), for the French airports multiplex network (E), for the British airports multiplex network (F), and for the German airports multiplex network (G). The bar is colored in red when a node is found in all top-K scores (K is equal to 1, 3, 5, and 10), and in blue otherwise. The parameters of the RWR are detailed in Table S7.

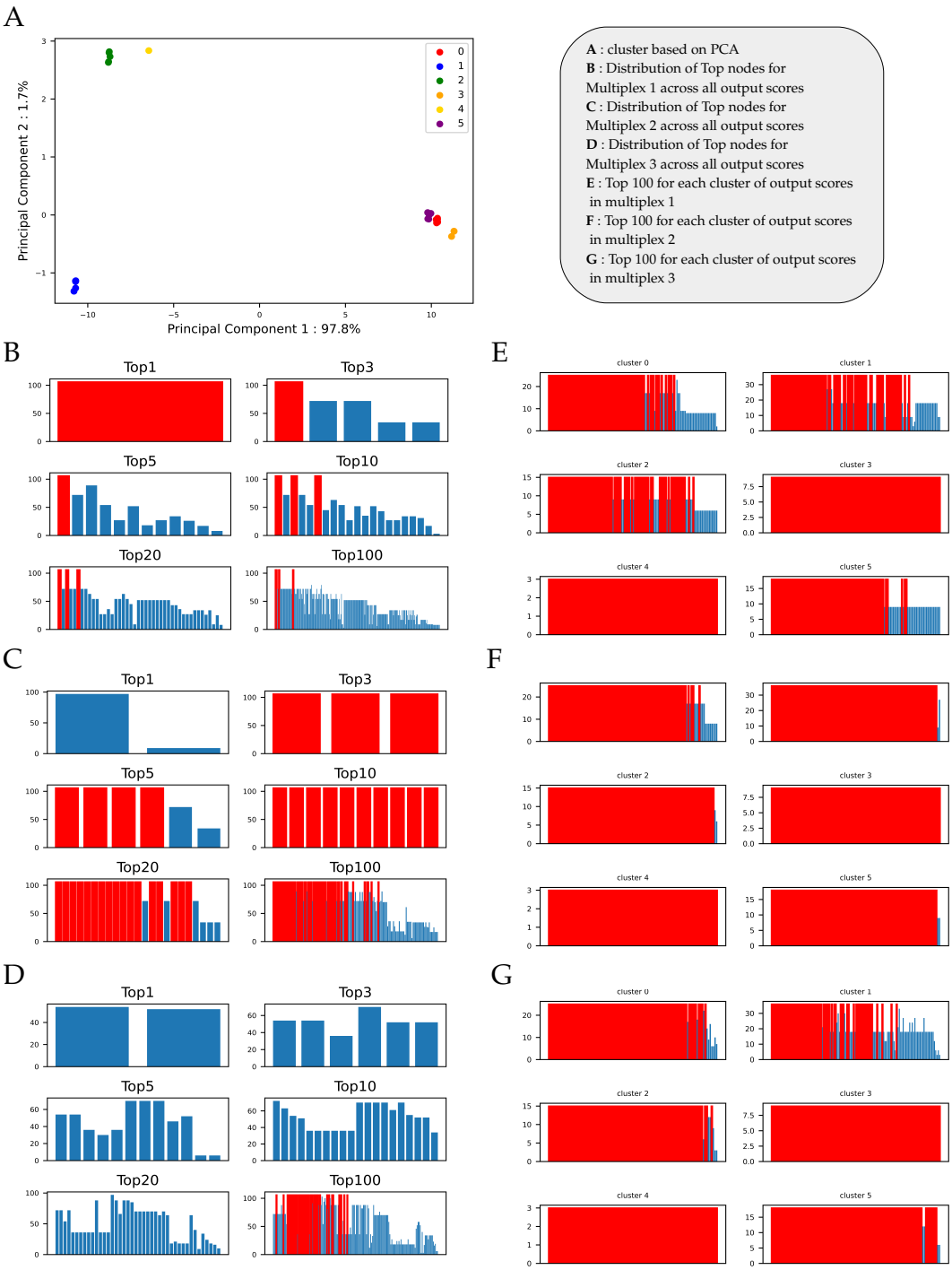


Fig. S15. A: Clustering in the PCA space of the output scores obtained with MultiXrank on biological multilayer network composed of genes, drug multiplex networks, and a disease monoplex network, as described in section 3.B, using 108 different sets of parameters. B-D: Comparisons of the top-1, top-3, top-5, top-10, top-20 and top-100 nodes obtained for the gene multiplex network (B), for the disease monoplex network (C), and for the drug multiplex network (D), for 108 different sets of parameters. E-G: Top-100 nodes for the different sets of parameter in each cluster defined from the PCA space (A), for the gene multiplex network (E), for the disease monoplex network (F), and for the drug multiplex network (G). The bar is colored in red when a node is found in all top-K scores (K is equal to 1, 3, 5, 10, 20, and 100), and in blue otherwise. The parameters of the RWR are detailed in Table S7.

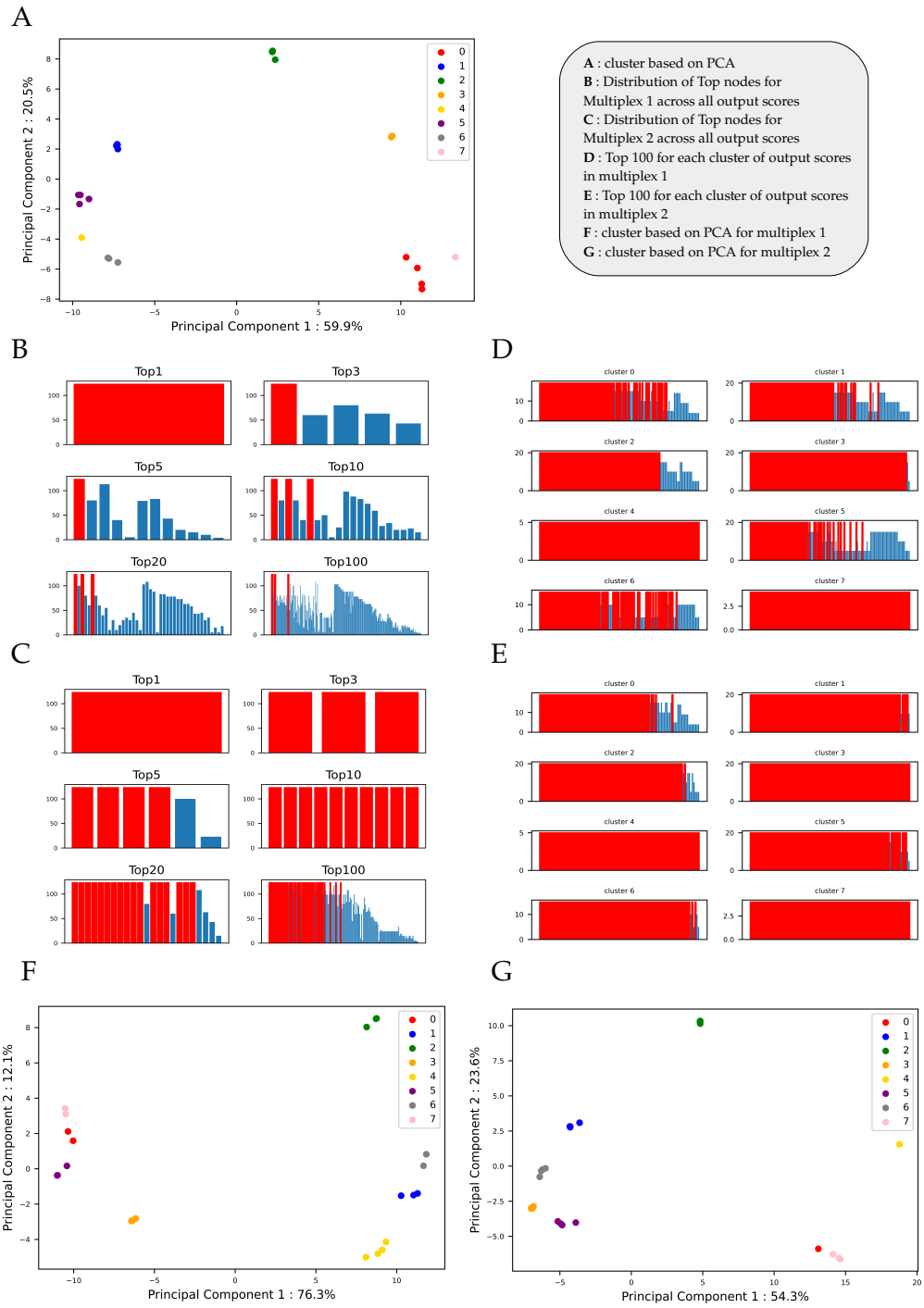


Fig. S16. A: Clustering in the PCA space of the output scores obtained with MultiXrank on a biological multilayer network composed of a gene multiplex network and a disease monoplex network, as described in section 3.B, using 125 different sets of parameters. B-C: Comparisons of the top-1, top-3, top-5, top-10, top-20 and top-100 nodes obtained for the gene multiplex network (B), for the disease monoplex network (C), for 125 different sets of parameters. D-E: Top-100 nodes for the different sets of parameter in each cluster defined from the PCA space (A), for the gene multiplex network (D), for the disease monoplex network (E). The bar is colored in red when a node is found in all top-K scores (K is equal to 1, 3, 5, 10, 20, and 100) and in blue otherwise. F-G: Clustering in the PCA space of the output scores obtained with MultiXrank on the biological multilayer when we only consider the genes output scores (F) or only the diseases output scores (G). The parameters of the RWR are detailed in Table S7.

Supplementary References

1. Langville AN, Meyer CD (2006) *Google's PageRank and Beyond: The Science of Search Engine Rankings.* (Princeton University Press, USA).
2. Valdeolivas A, et al. (2018) Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* 35(3):497–505.
3. Li Y, Patra JC (2010) Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics* 26(9):1219–1224.
4. Jin W, Jung J, Kang U (2019) Supervised and extended restart in random walks for ranking and link prediction in networks. *PLOS ONE* 14(3):e0213857.
5. Cardillo A, et al. (2013) Emergence of network features from multiplexity. *Scientific Reports* 3(1):1344.
6. Drew K, et al. (2017) Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. *Mol Syst Biol* 13(6):932.
7. Giurgiu M, et al. (2019) Corum: the comprehensive resource of mammalian protein complexes–2019. *Nucleic Acids Res* 47(D1):D559–D563.
8. Türei D, et al. (2021) Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Molecular Systems Biology* 17(3):e9923.
9. Pratt D, et al. (2015) Ndex, the network data exchange. *Cell Systems* 1(4):302–305.
10. Croft D, et al. (2014) The reactome pathway knowledgebase. *Nucleic Acids Res* 42(D1):D472–D477.
11. Cheng F, Kovács IA, Barabási AL (2019) Network-based prediction of drug combinations. *Nature Communications* 10(1):1197.
12. Piñero J, et al. (2020) The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic Acids Res* 48(D1):D845–D855.
13. Brown AS, Patel CJ (2017) A standard database for drug repositioning. *Scientific Data* 4(1):170029.
14. Mordelet F, Vert JP (2011) Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics* 12(1):389.
15. Köhler S, Bauer S, Horn D, Robinson PN (2008) Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics* 82(4):949–958.

C. *Biological Applications of Random Walk with Restart on Multilayer Networks* : matériel supplémentaire

Supplementary Information for

Biological Applications of Random Walk with Restart on Multilayer Networks

Anthony Baptista[1, 2, *] **and Anaïs Baudot**[1, 3, *]

[1] Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, CNRS, Marseille, France, [2] Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France, [3] Barcelona Supercomputing Center, Barcelona, Spain

[*] anthony.baptista@univ-amu.fr, anais.baudot@univ-amu.fr

This PDF file includes:

Contents

1. Multilayer networks

A. Biological networks.

1. Gene multiplex network: this multiplex network is composed of three layers of interactions between gene nodes. Please note that we consider here genes and proteins indifferently. We choose the gene names for the nodes of this multiplex network and its associated bipartite networks.
    - Complexes: A molecular complexes layer constructed from the fusion of Hu.map (1) and Corum (2), using OmniPathR (3).
    - PPI: A Protein-Protein interaction (PPI) layer corresponding to the fusion of 3 datasets: APID (homo sapiens level 2, without inter-species interactions), Hi-Union and Lit-BM (www.interactome-atlas.org/download). It is important to notice that 166 over 13346 APID proteins do not match any gene name. We choose to use the Uniprot IDs in these cases.
    - Reactome: a pathways layer extracted from NDEx (4) and corresponding to human Reactome data (5).
2. Disease monoplex network: The network was built based on phenotypic proximities between diseases calculated with the information content. The whole protocol is described in (6). We choose the UMLS annotation for diseases in this monoplex network and its associated bipartite networks.
3. Drug multiplex network: The data come from Cheng et al. (7)[†] and pharmacological drugs interaction network from snap.stanford.edu[‡]. We used the DrugBank name format for the drugs in this multiplex network and its associated bipartite networks.
    - Drug1[†]: Clinical drug interactions (clinically reported adverse drug-drug interactions, 14822 clinically reported adverse drug–drug interactions connecting 667 drugs were retained)
    - Drug2[†]: Experimental drug combinations (experimentally validated drug combinations, 737 unique pairwise drug combinations connecting 376 drugs were retained)
    - Drug3[†]: Predicted drug combinations (network-predicted hypertensive drug combinations, 2080 potential combinations involving 65 hypertensive drugs were retained)
    - Drug4[‡]: Drug-Drug interactions determined from the pharmacologic effect of the action of one drug on another drug (48514 interactions involving 1514 drugs).

Bipartite networks are defined as follows:

- Gene-Disease (1-2): The bipartite network comes from the 2020 version (v7.0) of DisGeNET (8). This network is weighted with score computed according to the formula reported in https://www.DisGeNET.org/dbinfo.
- Gene-Drug (1-3): Several bipartite networks are available: drug-target association from DrugBank Release Version 5.1.8 go.drugbank.com/releases/latest, drug-target associations from drugcentral release v10.12 drugcentral.org/download, and the drug-target association from Cheng et al (7). We merged all of these bipartite interactions to obtain the gene-drug bipartite network.
- Disease-Drug (2-3): The disease-drug association network comes from repoDB from (9).

It is important to note that some nodes are present only in the bipartite networks but absent from the corresponding monoplex/multiplex networks. We have to artificially add these nodes to the multiplex networks by adding self-loops in one of the layers (see supplementary section 2.B.4). They will then be ranked by MultiXrank.

B. Hetionet networks.

Hetionet (10) is composed of heterogeneous networks constructed by integrating knowledge and experimental findings from millions of biomedical research pubications. This heterogeneous network contains nine kinds of nodes: genes, diseases, compounds, anatomy parts, symptoms, side effects, pharmacologic classes, pathways, biological processes. In our formalism, it is necessary to represent the original data suitably for MultiXrank, i.e, multiplex networks (or monoplex networks) and bipartite networks. However, some kind of nodes are just present in the bipartite networks, thus it could be needed to create an artificial self-loop network to take them as specific nodes.

1. Gene multiplex network: Three layers Gene–covaries–Gene (GcG), Gene–interacts–Gene (GiG), Gene-regulates-Gene (GrG)
2. Disease monoplex network: Disease–resembles–Disease (DrD)
3. Compound monoplex network: Compound–resembles–Compound (CrC)
4. Anatomy part monoplex network: The nodes were extracted from bipartite networks 2-4 and 4-1 (self-loops network)
5. Symptom monoplex network: The nodes were extracted from bipartite networks 2-5 (self-loops network)
6. Side effect monoplex network: The nodes were extracted from bipartite networks 3-6 (self-loops network)
7. Pharmacologic classe monoplex network: The nodes were extracted from bipartite networks 7-3 (self-loops network)
8. Pathway monoplex network: The nodes were extracted from bipartite networks 1-8 (self-loops network)
9. Biological processe monoplex network: The nodes were extracted from bipartite networks 1-9 (self-loops network)

Bipartite networks are defined as follows (For more details see the Table 2 from D.S.Himmelstein et al. (10)):

- Gene-Pathway (1-8): Gene–participates–Pathway (GpPW)
- Gene-Biological Process (1-9): Gene–participates–Biological Process (GpBP)
- Anatomy Part-Gene (4-1): Merged of three layers, Anatomy–downregulates–Gene (AdG), Anatomy–expresses–Gene (AeG), Anatomy–upregulates–Gene (AuG)
- Disease-Anatomy part (2-4): Disease–localizes–Anatomy (DlA)
- Disease-Symptom (2-5): Disease–presents–Symptom (DpS)
- Compound-Side effect (3-6): Compound–causes–Side Effect (CcSE)
- Pharmalogic classes-Compound (7-3): Pharmacologic Class–includes–Compound (PCiC)
- Disease-Gene (2-1): Merged of three layers, Disease–associates–Gene (DaG), Disease–downregulates–Gene (DdG), Disease–upregulates–Gene (DuG)
- Coumpound-Disease (3-2): Merged of two layers, Compound–palliates–Disease (CpD), Compound–treats–Disease (CtD)
- Coumpound-Gene (3-1): Merged of two layers, Compound–downregulates–Gene (CdG), Compound–upregulates–Gene (CuG)

C. Multilayer network with genomic information.

The multilayer network is represented in Fig. 4. It is to note that the gene multiplex network, the disease monoplex network and the bipartite network associated are described in section 1.A of the supplementary information. These networks are the same used in the studies in the first and second parts of the article.
The networks that represent the PCHi-C fragment for the eight different hematopoietic cell types are described in supplementary Table S7. These networks were generated with datasets from the study of Javierre et al. (11). The original datasets defined the interactions detected by the PCHi-C experiment (12), after processing by the CHICAGO pipeline (score > 5) (13), to only keep the significant interactions according to the scores computed by the pipeline. In our case, we use these processed datasets that are organized as follows:

fragments 1			fragments 2			Score
chr	start	end	chr	start	end	
⋮			⋮			⋮
⋮			⋮			⋮
⋮			⋮			⋮

The networks that represent the TADs for the eight different hematopoietic cell types, are constructed by creating an edge between two adjacent TADs. The idea behind this is to use the linear approximation of the genome in the edges of these networks, to complete the 3D conformation information that drives the edges between PCHi-C fragments in the corresponding networks. In the case of the TADs, they are determined using the directionality index score (14). In our case, we use these processed datasets that are organized as follows:

TADs		
chr	start	end
	⋮	
	⋮	
	⋮	

Bipartite networks are defined as follows (fragment defined PCHi-C fragment in the following section):

- Gene-Fragment: This bipartite network is created with the reported gene to the promoter associated with the fragment, defined in the article of Javierre et al. (11). They used the version GRCh37.p13 of the genome.
- Gene-TAD: We created an edge between a gene and a TAD, if the gene is included in the TAD region. To define the position of genes we used gencode V19, with the version GRCh37.p13 of the genome, to correspond with the version used in the article of Javierre et al. (11).
- Fragment-TAD: We created an edge between a fragment and a TAD, if the fragment is included in the TAD region.

2. Node prioritization

A. Node prioritization in leukemia.

We take the genes and the drug multiplex networks described in Fig. 1 and supplementary Table S1. We focus on the study of Leukemia, which is a well-studied disease. We choose as seeds one gene and one drug: **HRAS**, which known to be involved in Acute Myeloid Leukemia (AML) (15), and Tipifarnib (**DB04960**), a drug that is being studied in the treatment of AML and other types of cancer (16–18).
We checked the top-10 scoring genes and drugs (supplementary Table S2) obtained from MultiXrank, and compared to the literature and the DrugBank database information, to determine if predictions are relevant regarding to the current knowledge.

Top-10 genes (at least one reference was found in the literature for each gene):

1. CYP3A4 is known to be implicated in Leukemia [https://doi.org/10.1073/pnas.95.22.13176].
2. FNTB is the target of Tipifarnib (our seed drug) according to DrugBank.
3. RAF1 is a well-known Leukemia-associated gene [ORPHA → RAF1 - Raf-1 proto-oncogene, serine/threonine kinase].
4. RASGRP1 has been linked with Leukemia previously [https://doi.org/10.1038/leu.2011.328].
5. DGKZ knockdown can induce apoptosis in human AML HL-60 cells through the MAPK/survivin/caspase pathway [https://doi.org/10.1691/ph.2019.9386].
6. RIN1 plays a role in the maintenance of the abnormal RTK signaling in Chronic Myeloid Leukemia [https://doi.org/10.1007/s13277-015-3772-9].
7. BRAF mutations were identified in AML patients with monocytic differentiation [https://doi.org/10.1080/10428194.2016.1
8. RASA1 is involved in several cancer types, including Leukemia [https://doi.org/10.3892/or.2020.7807].
9. ARAF: A study [https://doi.org/10.1111/j.1600-0463.2005.apm1130108.x] found ARAF gene mutation in MOLT-4 Leukemia cell line.
10. AURKA: A significant association between over-expression of AURKA and cytogenetic abnormalities was found in AML patients [https://doi.org/10.1016/j.leukres.2010.07.034].

Top-10 drugs (we first parsed DrugBank to identify the indications for each drug and further surveyed the literature to find additional information):

1. DB00637 (Astemizole) belongs to a drug class that was validated for its activities against human primary AML samples [https://doi.org/10.1038/s41408-018-0087-2].
2. DB01380 (Cortisone acetate): This drug has two targets, ANXA1 and NR3C1, which are part of the top-10 predicted genes. Moreover, cortisone acetate was shown to dramatically reduce the size of lymphoid tumors or Leukemia [doi:10.1172/JCI102317].
3. DB00630 (Alendronate): Some preliminary experiments show a benefit for the treatment of osteopenia/osteoporosis with Alendronate in children with Acute Lymphoblastic Leukemia [https://doi.org/10.1097/mph.0b013e31
4. DB00398 (Sorafenib) is a drug used for the treatment of advanced renal cell carcinoma, and some studies mention its impact on Leukemia [https://doi.org/10.1038/sj.leu.2404508].
5. DB00399 (Zoledronic acid) is a drug used to prevent skeletal fractures in patients with cancers such as multiple myeloma and prostate cancer, and some studies show interesting links with Leukemia [https://doi.org/10.3892/mmr.2016.5957].

6. DB00773 (Etoposide) is used for first-line treatment in patients with small-cell lung cancer. It is also used to treat other malignancies such as Lymphoma, Nonlymphocytic Leukemia (source:DrugBank).
7. DB01254 (Dasatinib) is used in patients with Chronic Myelogenous Leukemia (source:DrugBank).
8. DB01268 (Sunitinib): A phase I/II study of sunitinib and intensive chemotherapy for AML patients with FLT3 mutations has been achieved with encouraging results [https://doi.org/10.1111/bjh.13353].
9. DB06589 (Pazopanib): This drug is in clinical trial phase II for patients with AML [https://doi.org/10.1007/s00277-019-03651-9].
10. DB00530 (Erlotinib) is an inhibitor of the epidermal growth factor receptor (EGFR) tyrosine kinase that is used in the treatment of several types of cancer, and some references exist for Leukemia [https://doi.org/10.4161/cc.22382].

r	δ	τ	λ	η	K
0.7	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$	10

multiplex networks	Layer	nodes	# interactions
1	PPI	genes	143 652
1	Complexes	genes	63 561
1	Reactome	genes	194 500
2	clinical drugs interactions	drugs	14 909
2	experimental drugs interactions	drugs	843
2	predicted drugs interactions	drugs	2 080
2	Pharmacological drugs interactions	drugs	48 514

multiplex networks	# nodes
1	16 967
2	1559

Bipartite network	# interactions
2-1	30 895

Table S1. Networks specificities and parameters for the genes and drug multiplex networks.

genes	scores	drugs	scores
CYP3A4	0.009444	DB00637	0.000357
FNTB	0.009263	DB01380	0.000350
RAF1	0.001001	DB00630	0.000326
RASGRP1	0.000982	DB00398	0.000313
DGKZ	0.000780	DB00399	0.000252
RIN1	0.000745	DB00773	0.000248
BRAF	0.000225	DB01254	0.000184
RASA1	0.000204	DB01268	0.000165
ARAF	0.000196	DB06589	0.000114
AURKA	0.000195	DB00530	0.000096

Table S2. MultiXrank scores for multilayer network composed of a gene and drug multiplex networks.

B. Node prioritization to predict candidate drugs for epilepsy and comparison of MultiXrank with the Hetionet framework.

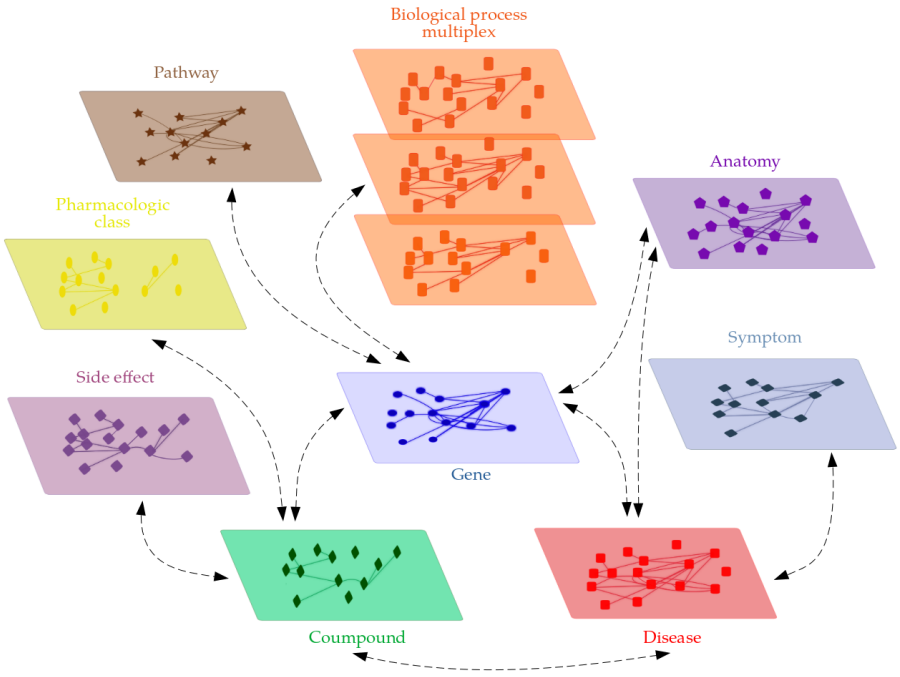


Fig. S1. Hetionet heterogeneous network adapted from D.S.Himmelstein et al. (10)

Drugs class	N	Drugs class	N	Drugs class	N
Cytochrome P-450 Substrates	24	Peripheral Nervous System Agents	10	Agents causing hyperkalemia	8
Agents that produce hypertension	18	Antimigraine Preparations	10	Cytochrome P-450 CYP2C19 Substrates	7
Central Nervous System Depressants	18	Cardiovascular Agents	10	Cytochrome P-450 CYP3A4 Inhibitors	7
Neurotransmitter Agents	17	Sensory System Agents	10	Enzyme Inhibitors	7
Analgesics	17	P-glycoprotein substrates	9	Biogenic Amines	7
Heterocyclic Compounds Fused-Ring	15	Drugs that are Mainly Renally Excreted	9	Biogenic Monoamines	7
Cytochrome P-450 Enzyme Inhibitors	14	Serotonin Modulators	9	Selective Serotonin 5-HT1 Receptor Agonists	7
Cytochrome P-450 CYP3A4 Substrates	14	Serotonin Receptor Agonists	9	Selective Serotonin Agonists	7
Amines	12	Serotonin 5-HT1 Receptor Agonists	9	Serotonin 1b Receptor Agonists	7
Cytochrome P-450 CYP1A2 Substrates	12	Cytochrome P-450 CYP2D6 Substrates	9	Serotonin 1d Receptor Agonists	7
Serotonergic Drugs	11	Cytochrome P-450 CYP2C9 Substrates	8	Triptans	7
Antidepressive Agents	10	Indoles	8	P-glycoprotein inhibitors	7
Serotonin Agents	10	Analgesics Non-Narcotic	8		

Table S3. Drugs classification of the 41 drugs predicted in the top-100 of MultiXrank with the set of parameters 4 that are not predicted in the top-100 predicted drugs of Hetionet. The first column of the table corresponds to the drugs class and the second column is associated with the number of times (N) where this class is found among the 41 drugs only predicted by MultiXrank. It is important to note that we only take the drugs class that is associated with 7 or more drugs.

3. Supervised classification of gene-disease associations

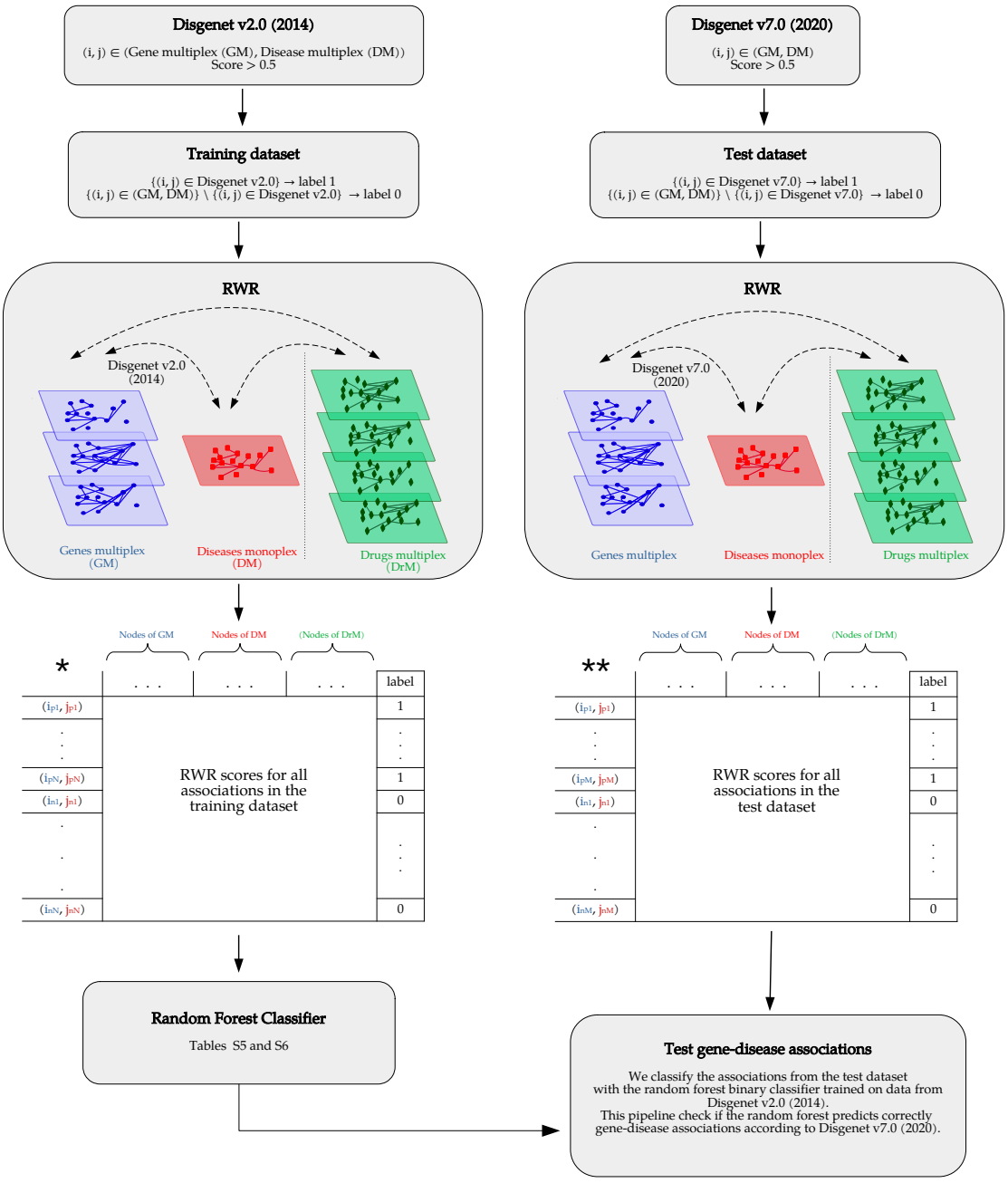


Fig. S2. Workflow for the training and the testing of the random forest binary classifier training with MultiXrank output scores. The left workflow represents the training step with the 2014 gene-disease bipartite network (DisGeNET v2.0). After creating both positive (from DisGeNET v2.0) and negative gene-disease associations, we run for each association MultiXrank, and we save the output scores as described in the matrix *. Then we use this matrix with the label to train a random forest binary classifier. The right workflow represents the test step with the updated (2020) gene-disease bipartite network (DisGeNET v7.0). After creating the positive (from DisGeNET v7.0) and negative gene-disease associations, we run MultiXrank, and we save the output scores as described in the matrix **. The last step consists of using this matrix as an input of the previous trained random forest classifier to predict the labels. Finally, we compare the predictions of the labels by the random forest classifier and the known labels (supplementary Tables S5 and S6). The workflow is the same for two or three multiplex networks.

r	δ	τ	λ	η	K
0.7	$\begin{bmatrix} 1/2 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$	'all'

r	δ	τ	λ	η	K
0.7	$\begin{bmatrix} 1/2 & 0 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 1/2 & 0 \end{bmatrix}$	'all'

Table S4. Parameters of the RWR for the universal multilayer network used to train and test the classifier. Top: parameters for the multilayer network composed of the gene multiplex network and disease monoplex network (2-multiplex network). Bottom: parameters for the multilayer network composed of the gene, drug multiplex networks, and disease monoplex network (3-multiplex network).

model classifier	Y == 0	Y == 1	Y	F1 score	weights
Random Forest (100 estimators)	88.28%	35.54%	61.91%	0.4827	none
Random Forest (200 estimators)	90.80%	30.56%	60.68%	0.4373	none
Random Forest (1000 estimators)	90.93%	31.57%	61.25%	0.4489	none
Gradient Boosting	80.40%	49.91%	65.15%	0.5888	none
Decision Tree weight 0	89.60%	25.52%	57.56%	0.3755	0 : 0.1, 1 : 0.9
Decision Tree weight 1	77.63%	31.82%	54.73%	0.4128	0 : 0.2, 1 : 0.8
Decision Tree weight 2	81.10%	29.24%	55.17%	0.3947	0 : 0.3, 1 : 0.7
Decision Tree weight 3	81.73%	20.29%	51.01%	0.2929	0 : 0.4, 1 : 0.6
Decision Tree weight 4	67.55%	58.29%	62.92%	0.6112	0 : 0.5, 1 : 0.5
Decision Tree weight 5	63.96%	59.80%	61.88%	0.6107	0 : 0.6, 1 : 0.4
Decision Tree weight 6	65.60%	42.28%	53.94%	0.4786	0 : 0.7, 1 : 0.3
Decision Tree weight 7	25.33%	75.24%	50.28%	0.6021	0 : 0.8, 1 : 0.2
Decision Tree weight 8	12.98%	81.29%	47.13%	0.6059	0 : 0.9, 1 : 0.1
Random Forest (100 estimators) 0	99.05%	2.02%	50.54%	0.0392	0 : 0.1, 1 : 0.9
Random Forest (100 estimators) 1	96.66%	8.88%	52.77%	0.1583	0 : 0.2, 1 : 0.8
Random Forest (100 estimators) 2	96.16%	10.84%	53.50%	0.1890	0 : 0.3, 1 : 0.7
Random Forest (100 estimators) 3	91.43%	26.97%	59.20%	0.3980	0 : 0.4, 1 : 0.6
Random Forest (100 estimators) 4	88.97%	38.44%	63.71%	0.5143	0 : 0.5, 1 : 0.5
Random Forest (100 estimators) 5	83.55%	48.96%	66.26%	0.5920	0 : 0.6, 1 : 0.4
Random Forest (100 estimators) 6	68.43%	73.53%	70.98%	0.7171	0 : 0.7, 1 : 0.3
Random Forest(100 estimators) 7	36.80%	93.57%	65.19%	0.7288	0 : 0.8, 1 : 0.2
Random Forest (100 estimators) 8	13.36%	99.24%	56,30 %	0.6943	0 : 0.9, 1 : 0.1
Random Forest (1000 estimators) 6	93.01%	87.78%	90.39%	0.9013	0 : 0.7, 1 : 0.3
Random Forest (100 estimators) balanced	90.42%	31.19%	60.81%	0.4432	balanced
Random Forest (100 estimators) balanced subsamble	90.04%	31.88%	60.96%	0.4496	balanced subsample

Table S5. Different models of binary classifiers and their accuracies. There are trained and tested according to the process described in Fig. S2. Here we display the results for the multilayer network composed of a gene multiplex network and the disease monoplex network. The classifier highlighted in orange is the most accurate, because it offers high F1 score with balance predictions for the associations labeled 0 or 1. The classifier highlighted in yellow is a good alternative classifier. For each classifier: the columns Y==0, Y==1, display the percentage of negative (Y==0) and positive (Y==1) gene-disease associations correctly predicted by the classifier, the column Y is the percentage of gene-disease associations correctly predicted by the classifier, and the two last columns display the F1 score and the weights chosen for each class of the classifier.

model classifier	Y == 0	Y == 1	Y	F1 score	weights
Random Forest (100 estimators) 0	91.05 %	16.26 %	53.65 %	0.2597	0 : 0,1, 1 : 0,9
Random Forest (100 estimators) 1	89.41 %	19.28 %	54.35 %	0.2969	0 : 0,2, 1 : 0,8
Random Forest (100 estimators) 2	88.78 %	19.47 %	54.13 %	0.2980	0 : 0,3, 1 : 0,7
Random Forest (100 estimators) 3	88.15 %	21.80 %	54.98 %	0.3263	0 : 0,4, 1 : 0,6
Random Forest (100 estimators) 4	85.32 %	22.94 %	54.13 %	0.3333	0 : 0,5, 1 : 0,5
Random Forest (100 estimators) 5	84.37 %	26.21 %	55.29 %	0.3696	0 : 0,6, 1 : 0,4
Random Forest (100 estimators) 6	80.91 %	30.81 %	55.86 %	0.4111	0 : 0,7, 1 : 0,3
Random Forest(100 estimators) 7	74.10 %	35.54 %	54.82 %	0.4403	0 : 0,8, 1 : 0,2
Random Forest (100 estimators) 8	60.49 %	64.27 %	62,38 %	0.6308	0 : 0,9, 1 : 0,1

Table S6. Different models of binary classifiers and their accuracies. There are training and testing according to the process described in Fig. S2. Here we display the results for the multilayer network composed of genes, drug multiplex networks and a disease monoplex network. The classifier highlighted in orange is the most accurate, because it offers high F1 score with balance predictions for the associations labeled 0 or 1. For each classifier: the columns Y==0, Y==1, display the percentage of negative (Y==0) and positive (Y==1) gene-disease associations correctly predicted by the classifier, the column Y is the percentage of gene-disease associations correctly predicted by the classifier, and the two last columns display the F1 score and the weights chosen for each class of the classifier.

4. Integration of genomics information from PCHi-C with multilayer network approach

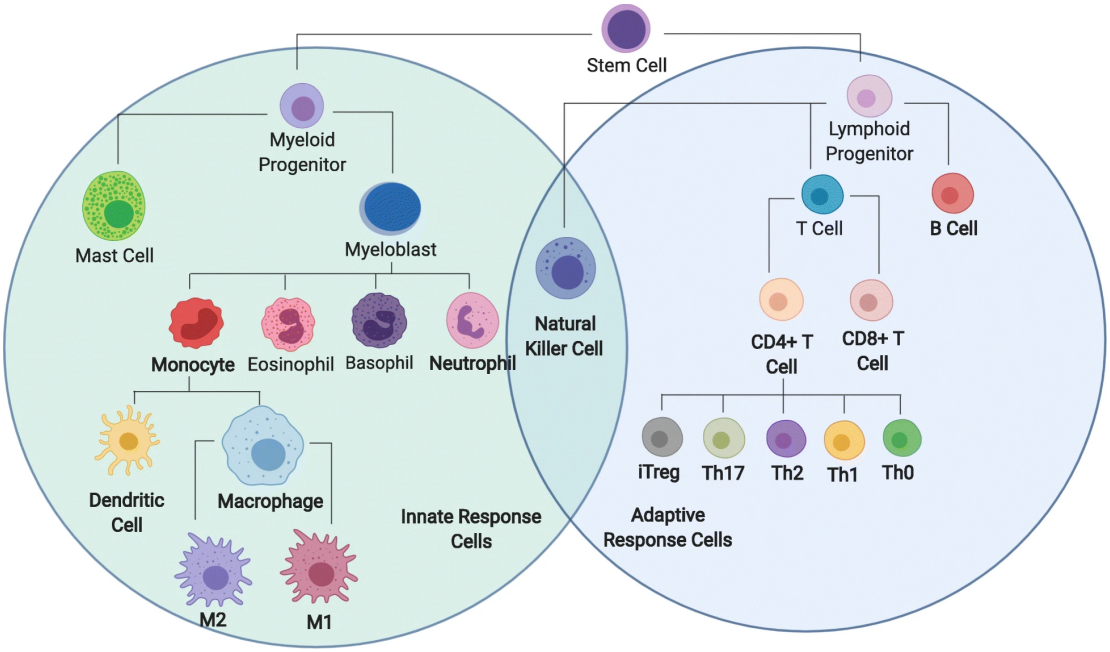


Fig. S3. The tree lineage of the hematopoetic cells, extracted from Torang et al. (19). Reproduce with the rights according to the user licence CC BY-NC 4.0.

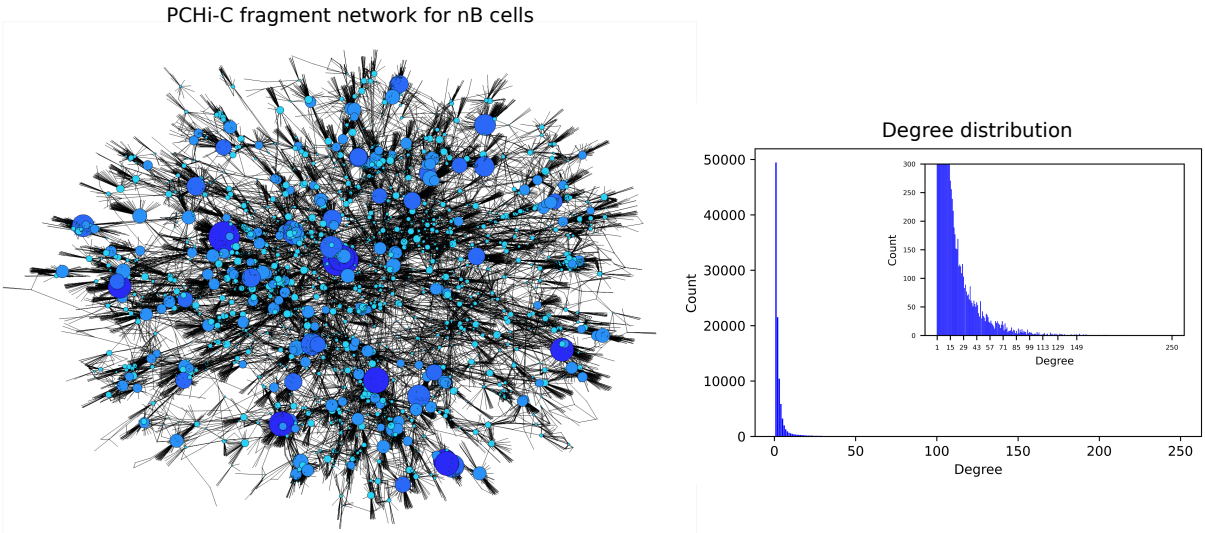


Fig. S4. On the left: PCHi-C fragment network for the nB cells data from Javierre et al. (11). On the right: Degree distribution of the PCHi-C fragment network defined on the left, with a zoom on the low degree nodes.

cellular type	**NB (Naive B cells)**	**Ery (Erythroblasts)**	**Mac0 (Macrophages M0)**	**Mon (Monocytes)**
# nodes	101973	90184	109566	96443
# edges	192104	152617	182250	167275
# isolated nodes	0	0	0	0
#self loops	0	0	0	0
density	0.000037	0.000038	0.000030	0.000036
avg clustering coeff	0.152	0.092	0.094	0.140
avg degree	3.768	3.385	3.327	3.469
# connected components	**1663**	**2082**	**2004**	**1938**
giant component (GC) size	**20003**	**4090**	**16136**	**8105**
diameter of GC	36	34	63	47
# nodes in periphery of GC	42	23	4	17
density of GC	0.000191	0.001281	0.000236	0.000467
avg clustering coeff of GC	0.165	0.139	0.120	0.172
avg degree of GC	3.826	5.237	3.812	3.782

cellular type	**nCD4 (Naive CD4 + T cells)**	**nCD8 (Naive CD8 + T cells)**	**Neu (Neutrophils)**	**MK (Megakaryocytes)**
# nodes	106849	108016	81244	98159
# edges	212106	218293	143679	152347
# isolated nodes	0	0	0	0
#self loops	0	0	0	0
density	0.000037	0.000037	0.000044	0.000032
avg clustering coeff	0.188	0.180	0.151	0.091
avg degree	3.970	4.042	3.537	3.104
# connected components	**1375**	**1277**	**2136**	**2633**
giant component (GC) size	**13252**	**31825**	**3676**	**2007**
diameter of GC	38	62	32	33.000
# nodes in periphery of GC	4	13	19	2
density of GC	0.000289	0.000133	0.001259	0.002590
avg clustering coeff of GC	0.201	0.199	0.162	0.162
avg degree of GC	3.826	4.232	4.628	5.196

Table S7. PCHi-C networks properties for the eight different hematopoietic cell types. The rows in bold defined relevant properties of the network : the number of connected components, and the size of the giant component. Networks with giant component of large size can be explored with random walks. The PCHi-C data was extracted from the study of Javierre et al. (11).

ID	Name	UMLS
0	Agammaglobulinemia	UMLS:C0221026
1	Behcet's disease	UMLS:C0004943
2	Chronic recurrent multifocal osteomyelitis (CRMO)	UMLS:C0410422
3	Cicatricial pemphigoid	UMLS:C1282359
4	Congenital heart block	UMLS:C0149530
5	Eosinophilic esophagitis (EoE)	UMLS:C0341106
6	Eosinophilic fasciitis	UMLS:C0264005
7	Giant cell arteritis (temporal arteritis)	UMLS:C1956391
8	Goodpasture's syndrome	UMLS:C0403529
9	Granulomatosis with Polyangiitis	UMLS:C3495801
10	Hashimoto's thyroiditis	UMLS:C0677607
11	Hypogammaglobulinemia AGM2 (IGLL1)	UMLS:C3150750
12	Hypogammaglobulinemia AGM3 (CD79A)	UMLS:C3150751
13	Hypogammaglobulinemia AGM4 (BLNK)	UMLS:C3150752
14	Hypogammaglobulinemia AGM5 (LRRC8A)	UMLS:C3150753
15	Hypogammaglobulinemia AGM6 (CD79B)	UMLS:C3150207
16	Immune thrombocytopenic purpura (ITP)	UMLS:C0398650
17	Inclusion body myositis (IBM)	UMLS:C0238190
18	Kawasaki disease (Mucocutaneous Lymph Node Syndrome)	UMLS:C0026691
19	Lichen sclerosus	UMLS:C0023652
20	Meniere's disease	UMLS:C0025281
21	Myasthenia gravis	UMLS:C0026896
22	Myositis	UMLS:C0027121
23	Ocular cicatricial pemphigoid	UMLS:C1282359
24	Parry Romberg syndrome	UMLS:C0015458
25	Pernicious anemia (PA)	UMLS:C0002892
26	Primary biliary cirrhosis	UMLS:C0008312
27	Raynaud's phenomenon	UMLS:C0034734
28	Rheumatoid arthritis	UMLS:C0003873
29	Schmidt syndrome	UMLS:C0085860
30	Sjögren's syndrome	UMLS:C1527336
31	Stiff person syndrome (SPS)	UMLS:C0085292
32	Takayasu's arteritis	UMLS:C0039263
33	Temporal arteritis, Giant cell arteritis	UMLS:C1956391
34	Hashimoto's syndrome	UMLS:C0677607
35	Rheumatoid arthritis	UMLS:C0003873
36	Burkitt lymphoma	UMLS:C0006413
37	Acute myeloid leukemia	UMLS:C0023467
38	Burkitt's lymphoma	UMLS:C0006413
39	Chronic lymphocytic leukemia	UMLS:C0023434
40	Chronic myelogenous leukemia	UMLS:C0023473
41	Hodgkin's lymphoma	UMLS:C0019829
42	Myelodysplastic syndromes	UMLS:C3463824
43	Non-Hodgkin lymphoma	UMLS:C0024305
44	Mycosis fungoides	UMLS:C0026948
45	Type 1 diabetes	UMLS:C0011854
46	22q11.2 deletion syndrome (DiGeorge syndrome)	UMLS:C0220704
47	22q11.2 deletion syndrome (DiGeorge syndrome)	UMLS:C0431406
48	22q11.2 deletion syndrome (DiGeorge syndrome)	UMLS:C0012236
49	Aicardi-Goutieres syndrome	UMLS:C3489725

ID	Name	UMLS
50	Aicardi-Goutieres syndrome	UMLS:C0796126
51	Aicardi-Goutieres syndrome	UMLS:C3489724
52	Amyloidosis familial visceral	UMLS:C0268389
53	Ataxia telangiectasia	UMLS:C0004135
54	Autoimmune lymphoproliferative syndrome	UMLS:C1328840
55	Autoimmune lymphoproliferative syndrome due to CTLA4 haploinsuffiency	UMLS:C4015214
56	Autoimmune polyglandular syndrome type 1	UMLS:C0085859
57	Autosomal dominant hyper IgE syndrome	UMLS:C3489795
58	Autosomal recessive candidiasis familial chronic mucocutaneous	UMLS:C3714992
59	Autosomal recessive early-onset inflammatory bowel disease 28	UMLS:C2751053
60	Autosomal recessive early-onset inflammatory bowel disease 25	UMLS:C2675508
61	Barth syndrome	UMLS:C0574083
62	Blau syndrome	UMLS:C1861303
63	Bloom syndrome	UMLS:C0005859
64	C1q deficiency	UMLS:C3150902
65	Cartilage-hair hypoplasia	UMLS:C0220748
66	CHARGE syndrome	UMLS:C0265354
67	Chediak-Higashi syndrome	UMLS:C0007965
68	Cherubism	UMLS:C0008029
69	Combined immunodeficiency with skin granulomas	UMLS:C2673536
70	Complement component 2 deficiency	UMLS:C3150275
71	Complement component 8 deficiency type 1	UMLS:C3151081
72	Complement component 8 deficiency type 2	UMLS:C3151080
73	Cryoglobulinemic vasculitis	UMLS:C1852456
74	Cyclic neutropenia	UMLS:C0221023
75	Deficiency of interleukin-1 receptor antagonist	UMLS:C2748507
76	Dendritic cell, monocyte, B lymphocyte, and natural killer lymphocyte deficiency	UMLS:C3280030
77	Epidermodysplasia verruciformis	UMLS:C0014522
78	Familial cold autoinflammatory syndrome	UMLS:C0343068
79	Familial hemophagocytic lymphohistiocytosis	UMLS:C0272199
80	Familial Mediterranean fever	UMLS:C0031069
81	Felty's syndrome	UMLS:C0015773
82	Glycogen storage disease type 1B	UMLS:C0268146
83	Griscelli syndrome type 2	UMLS:C1868679
84	Hepatic venoocclusive disease with immunodeficiency	UMLS:C1856128
85	Hereditary folate malabsorption	UMLS:C0342705
86	Hermansky Pudlak syndrome 2	UMLS:C1842362
87	Hyper-IgD syndrome	UMLS:C0398691
88	ICF syndrome	UMLS:C3279748
89	IL12RB1 deficiency	UMLS:C4013949
90	Immune dysfunction with T-cell inactivation due to calcium entry defect 1	UMLS:C2748568
91	Immune dysfunction with T-cell inactivation due to calcium entry defect 2	UMLS:C2748568
92	Immunodeficiency with hyper IgM type 1	UMLS:C0398689
93	Immunodeficiency with hyper IgM type 2	UMLS:C1720956
94	Immunodeficiency with hyper IgM type 3	UMLS:C1720957
95	Immunodeficiency with hyper IgM type 4	UMLS:C1842413
96	Immunodeficiency with hyper IgM type 5	UMLS:C1720958
97	Immunodeficiency without anhidrotic ectodermal dysplasia	UMLS:C1845117
98	Immune dysregulation, polyendocrinopathy and enteropathy X-linked	UMLS:C0342288
99	Immunoglobulin A deficiency 2	UMLS:C1836032

ID	Name	UMLS
100	Intestinal atresia multiple	UMLS:C0220744
101	IRAK-4 deficiency	UMLS:C1843256
102	Isolated growth hormone deficiency type 3	UMLS:C0472813
103	Leukocyte adhesion deficiency type 1	UMLS:C0398738
104	LRBA deficiency	UMLS:C3553512
105	Majeed syndrome	UMLS:C1864997
106	Melkersson-Rosenthal syndrome	UMLS:C0025235
107	MHC class 1 deficiency	UMLS:C1858266
108	MYD88 deficiency	UMLS:C2677092
109	Netherton syndrome	UMLS:C0265962
110	Neutrophil-specific granule deficiency	UMLS:C0398593
111	Osteopetrosis autosomal recessive 7	UMLS:C2676766
112	Papillon Lefevre syndrome	UMLS:C0030360
113	PASLI disease	UMLS:C4014934
114	PASLI disease	UMLS:C3714976
115	Pearson syndrome	UMLS:C0342784
116	PGM3-CDG	UMLS:C4014371
117	Poikiloderma with neutropenia	UMLS:C1858723
118	Pruritic urticarial papules plaques of pregnancy	UMLS:C0269680
119	Purine nucleoside phosphorylase deficiency	UMLS:C0268125
120	Pyogenic arthritis, pyoderma gangrenosum and acne	UMLS:C1858361
121	Reticular dysgenesis	UMLS:C0272167
122	Schimke immuno-osseous dysplasia	UMLS:C0877024
123	Severe congenital neutropenia X-linked	UMLS:C1845987
124	Short-limb skeletal dysplasia with severe combined immunodeficiency	UMLS:C1860168
125	Shwachman-Diamond syndrome	UMLS:C0272170
126	Spondyloenchondrodysplasia with immune dysregulation	UMLS:C1842763
127	T-cell immunodeficiency, congenital alopecia and nail dystrophy	UMLS:C1866426
128	TARP syndrome	UMLS:C1839463
129	Tumor necrosis factor receptor-associated periodic syndrome	UMLS:C1275126
130	Vici syndrome	UMLS:C1855772
131	WHIM syndrome	UMLS:C0472817
132	Wiskott Aldrich syndrome	UMLS:C0043194
133	Woods Black Norbury syndrome	UMLS:C1848144
134	X-linked agammaglobulinemia	UMLS:C0221026
135	X-linked immunodeficiency with magnesium defect, Epstein-Barr virus infection and neoplasia	UMLS:C3275445
136	X-linked lymphoproliferative syndrome	UMLS:C0549463
137	ZAP-70 deficiency	UMLS:C2931299

Table S8. List of the immune diseases. The first column represents the number to identify each disease in the t-SNE (t-distributed stochastic neighbor embedding) (20) and UMAP (Uniform Manifold Approximation and Projection) (21) projection in a 2D space, the second column represents the name of the disease, and the third column is the UMLS identifier of the disease. It is to note that some diseases can have several UMLS identifiers, so some diseases appear several times in the table associated with their different identifiers.

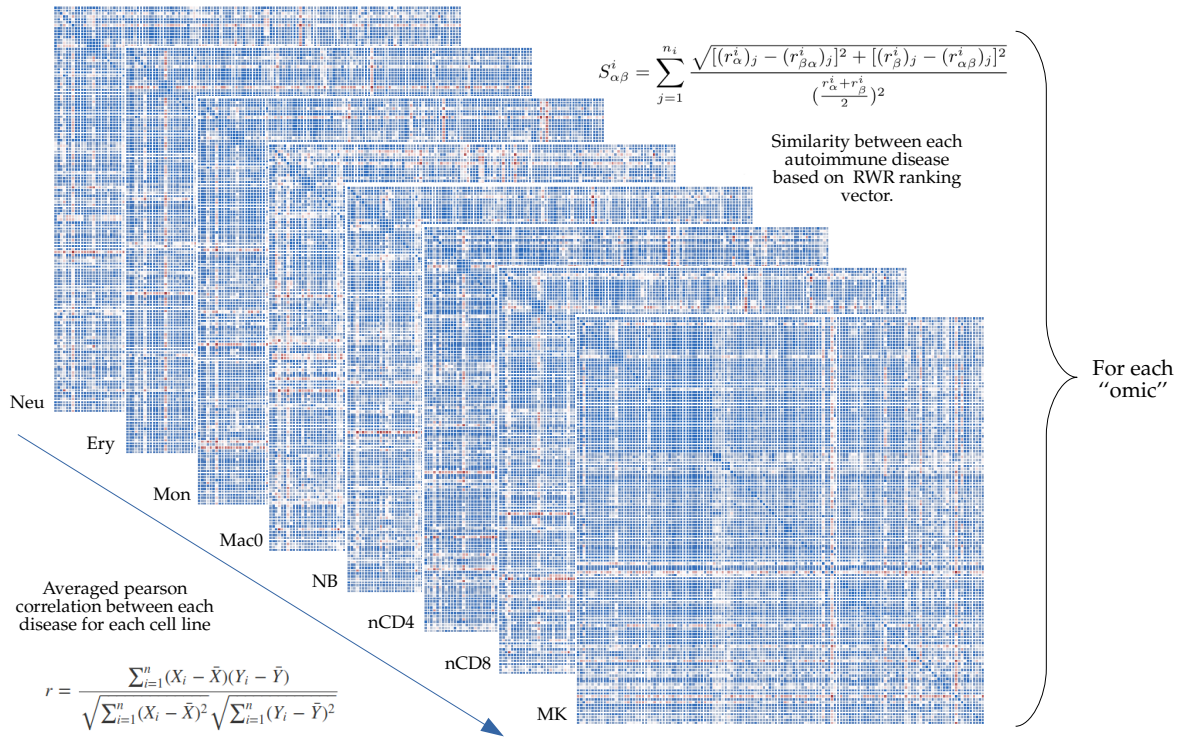


Fig. S5. Illustration of the protocol to produce Fig. 5-6 and Fig. S6-S9. For each of the four following omics, the gene/protein omic, the disease omic, the PCHi-C fragment omic, and the TAD omic: We obtain eight different similarity matrices of size 138×138 (number of immune diseases considered, supplementary Table S8), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then we constructed a proximity matrix obtained with the averaged Pearson correlation between each pair of the eight matrices previously defined. The averaged Pearson correlation corresponded to the mean of the Pearson correlation between each pair of columns of both matrices. Then we project the proximity matrix into a 2D PCA (Principal Component Analysis) space.

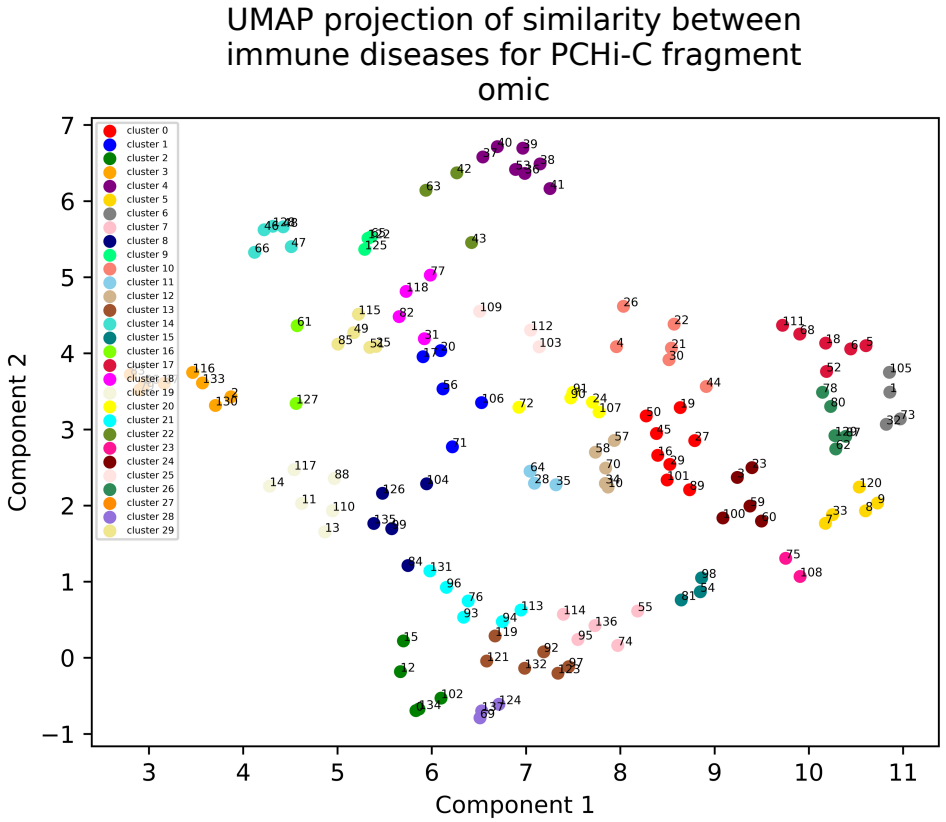


Fig. S6. The UMAP (21) projection in a 2D space of the consensual similarity between the 138 different immune diseases (see supplementary Table S8) obtained for the PCHi-C fragment omic. For this omic, we obtained eight different similarity matrices of size 138×138 (see Fig. S5), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then the consensual similarity matrix used for the projection, is obtained with equation (2). The t-SNE (20) projection in a 2D space of the consensual similarity proximity is represent in Fig. 6.

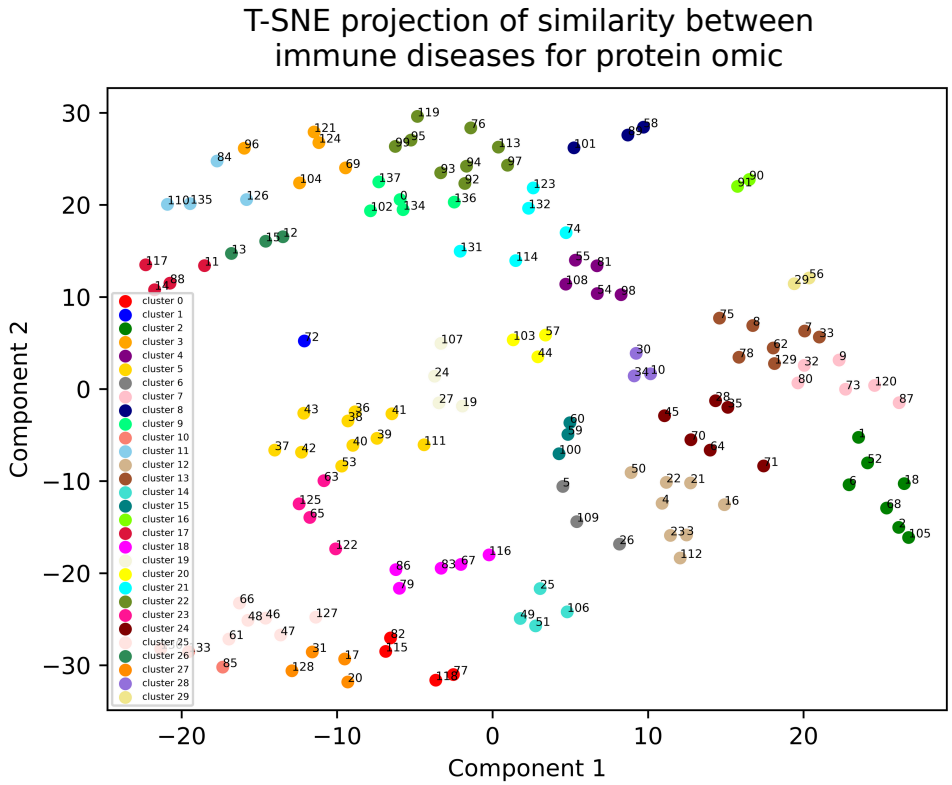


Fig. S7. The t-SNE (20) projection in a 2D space of the consensual similarity between the 138 different immune diseases (see supplementary Table S8) obtained for the gene/protein omic. For this omic, we obtained eight different similarity matrices of size 138×138 (see Fig. S5), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then the consensual similarity matrix used for the projection, is obtained with equation (2).

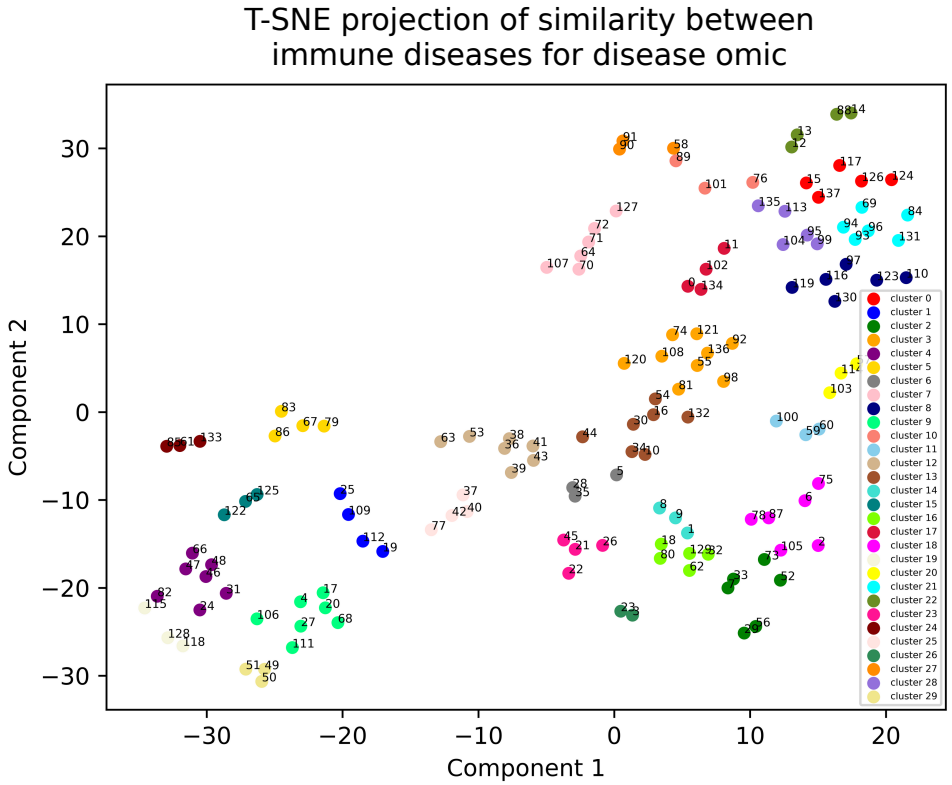


Fig. S8. The t-SNE (20) projection in a 2D space of the consensual similarity between the 138 different immune diseases (see supplementary Table S8) obtained for the disease omic. For this omic, we obtained eight different similarity matrices of size 138×138 (see Fig. S5), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then the consensual similarity matrix used for the projection, is obtained with equation (2).

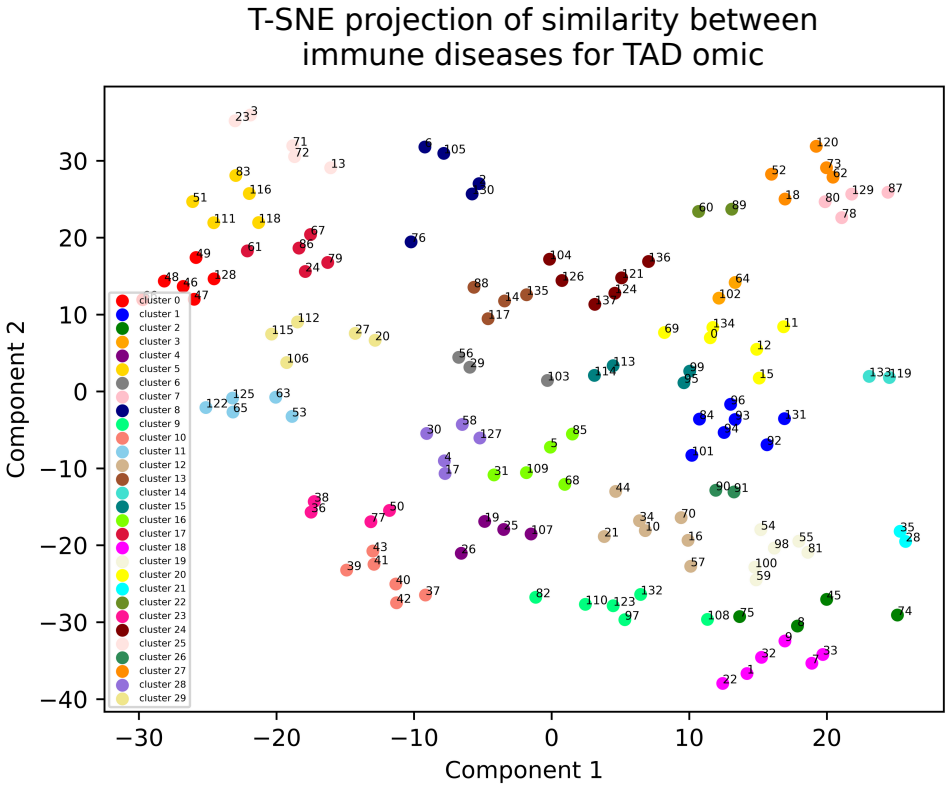


Fig. S9. The t-SNE (20) projection in a 2D space of the consensual similarity between the 138 different immune diseases (see supplementary Table S8) obtained for the TAD omic. For this omic, we obtained eight different similarity matrices of size 138×138 (see Fig. S5), one for each hematopoietic cell type. The similarity matrix is based on the similarity measure (defined on the top right of the figure) between the output scores of MultiXrank obtained for each disease considered as a seed of the RWR (Random Walk with Restart) process. Then the consensual similarity matrix used for the projection,is obtained with equation (2).

Supplementary References

1. Drew K, et al. (2017) Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. *Mol Syst Biol* 13(6):932.
2. Giurgiu M, et al. (2019) Corum: the comprehensive resource of mammalian protein complexes–2019. *Nucleic Acids Res* 47(D1):D559–D563.
3. Türei D, et al. (2021) Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Molecular Systems Biology* 17(3):e9923.
4. Pratt D, et al. (2015) Ndex, the network data exchange. *Cell Systems* 1(4):302–305.
5. Croft D, et al. (2014) The reactome pathway knowledgebase. *Nucleic Acids Res* 42(D1):D472–D477.
6. Valdeolivas A, et al. (2018) Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* 35(3):497–505.
7. Cheng F, Kovács IA, Barabási AL (2019) Network-based prediction of drug combinations. *Nature Communications* 10(1):1197.
8. Piñero J, et al. (2020) The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic Acids Res* 48(D1):D845–D855.
9. Brown AS, Patel CJ (2017) A standard database for drug repositioning. *Scientific Data* 4(1):170029.
10. Himmelstein DS, et al. (2017) Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife* 6:e26726.
11. Javierre BM, et al. (2016) Lineage-specific genome architecture links enhancers and non-coding disease variants to target gene promoters. *Cell* 167(5):1369–1384.e19.
12. Schoenfelder S, et al. (2015) The pluripotent regulatory circuitry connecting promoters to their long-range interacting elements. *Genome research* 25(4):582–97.
13. Cairns J, et al. (2016) Chicago: robust detection of dna looping interactions in capture hi-c data. *Genome Biology* 17(1):127.
14. Dixon JR, et al. (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485(7398):376–380.
15. Tyner JW, et al. (2009) High-throughput sequencing screen reveals novel, transforming ras mutations in myeloid leukemia patients. *Blood* 113(19075190):1749–1755.
16. Thomas X, Elhamri M (2007) Tipifarnib in the treatment of acute myeloid leukemia. *Biologics : targets & therapy* 1(19707311):415–424.
17. Karp JE, Lancet JE (2008) Tipifarnib in the treatment of newly diagnosed acute myelogenous leukemia. *Biologics : targets & therapy* 2(3):491–500.
18. Yanamandra N, et al. (2011) Tipifarnib-induced apoptosis in acute myeloid leukemia and multiple myeloma cells depends on ca2+ influx through plasma membrane ca2+ channels. *J Pharmacol Exp Ther* 337(3):636.
19. Torang A, Gupta P, Klinke DJ (2019) An elastic-net logistic regression approach to generate classifiers and gene signatures for types of immune cells and t helper cell subsets. *BMC Bioinformatics* 20(1):433.
20. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. *Journal of Machine Learning Research* 9(86):2579–2605.
21. McInnes L, Healy J, Saul N, Großberger L (2018) Umap: Uniform manifold approximation and projection. *Journal of Open Source Software* 3(29):861.

D. *Clustering in Multilayer Networks with Random Walk with Restart* : matériel supplémentaire

Supplementary Information for

Clustering in Multilayer Networks with Random Walk with Restart

Anthony Baptista[1, 2, *], Alberto Valdeolivas[3, *], Ozan Ozisik[1], and Anaïs Baudot[1, 4, *]

[1] Aix-Marseille Univ, INSERM, MMG, Turing Center for Living Systems, CNRS, Marseille, France, [2] Aix-Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, Marseille, France, [3] Roche Pharma Research and Early Development, Basel, Switzerland, [4] Barcelona Supercomputing Center, Barcelona, Spain

* anthony.baptista@univ-amu.fr, alvaldeolivas@gmail.com, anais.baudot@univ-amu.fr

This PDF file includes:

Figs. S1 to S6

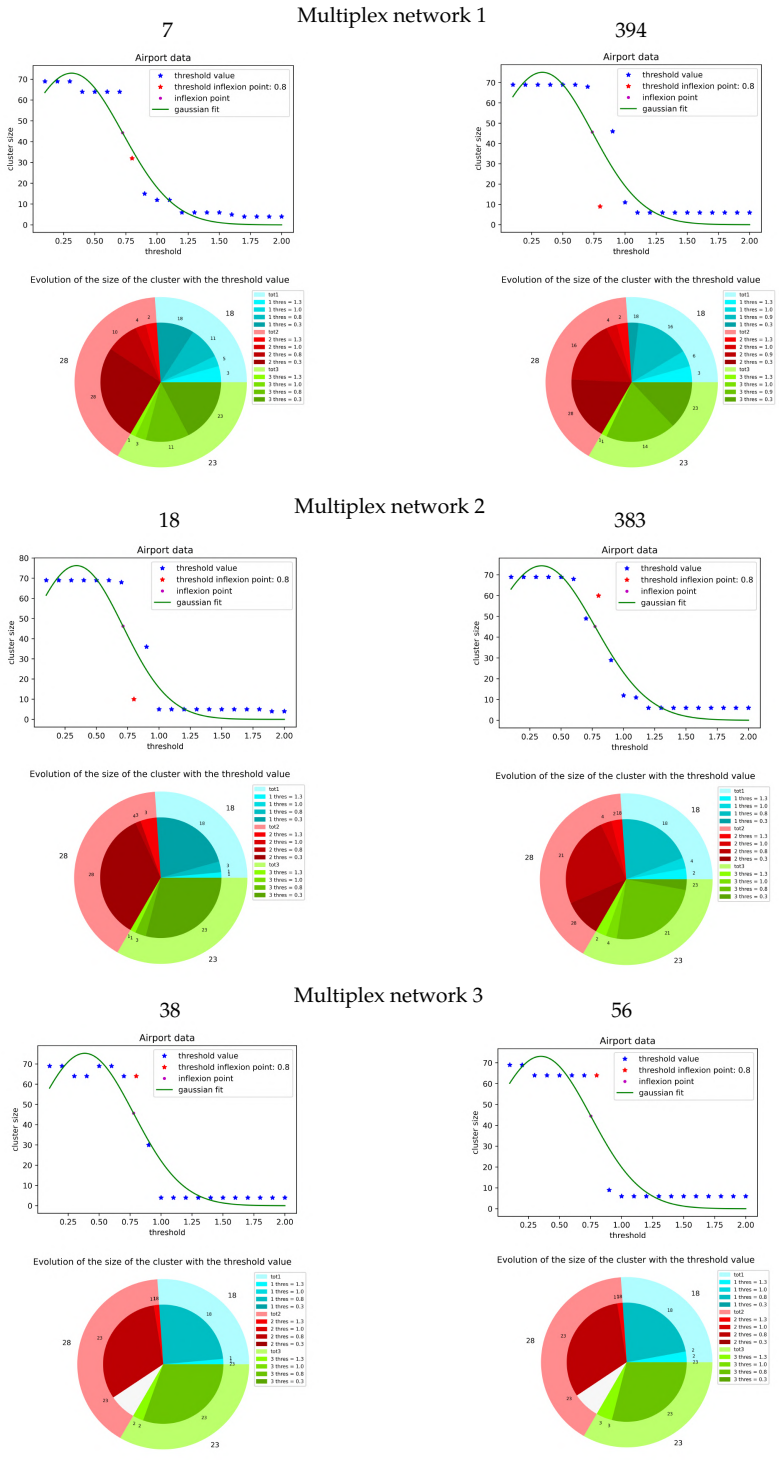
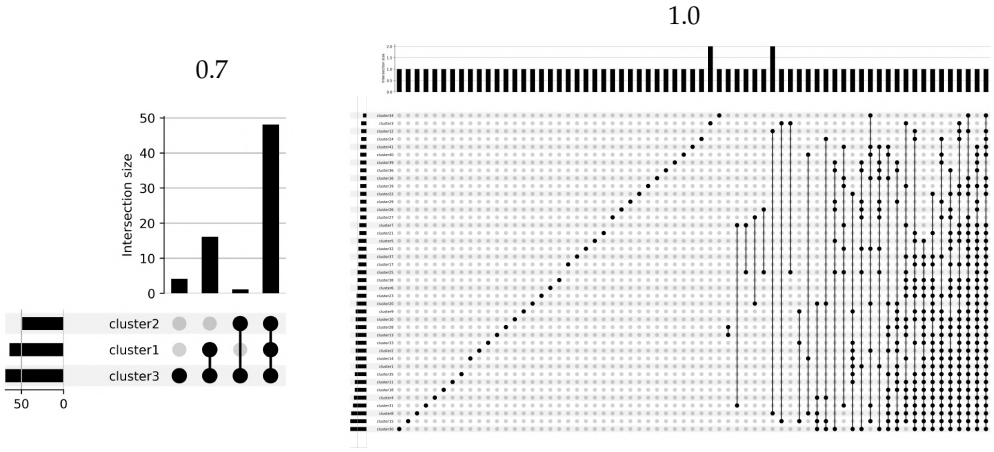
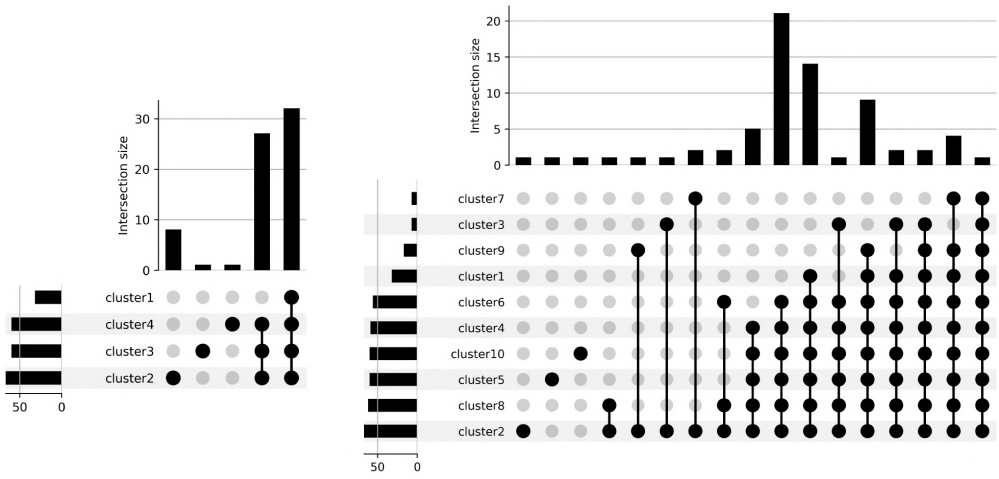


Fig. S1. Each pair of two charts represent the evolution of the community's final size depending on the threshold value. The top chart represents the evolution of the final community size depending on the threshold value from 0.1 to 2.0. The green curve represents the Gaussian fit with its inflection (pink point) point that gives us the value for which we observe an abrupt behavior, we expect to observe a better threshold value around this value. The red point represents the nearest experimental point (here always associated with a threshold value equal to 0.8). The bottom chart represents the evolution of the community size for different threshold values represented by the shades of colors. The colors represent the nodes of a specific multiplex network, blue for the first multiplex network (French multiplex network), red for the second multiplex network (British multiplex network), and green for the third multiplex network (German multiplex network). The external part of the pie chart gives the total number of nodes in each multiplex network, and the inner part gives the number of nodes in the final community depending on the threshold value and concerning their multiplex network of origin. The left figures represent the highest degree node selected as a seed in each multiplex network (#7, #18, #38), and the right figures represent the lowest degree node selected as a seed in each multiplex network (#394, #383, #56).

Upsetplot for two different threshold values, with seed is node 7



Upsetplot for two different runs, with threshold equal to 0.8 and seed is node 7



Upsetplot for two different runs, with threshold equal to 0.9 and seed is node 7

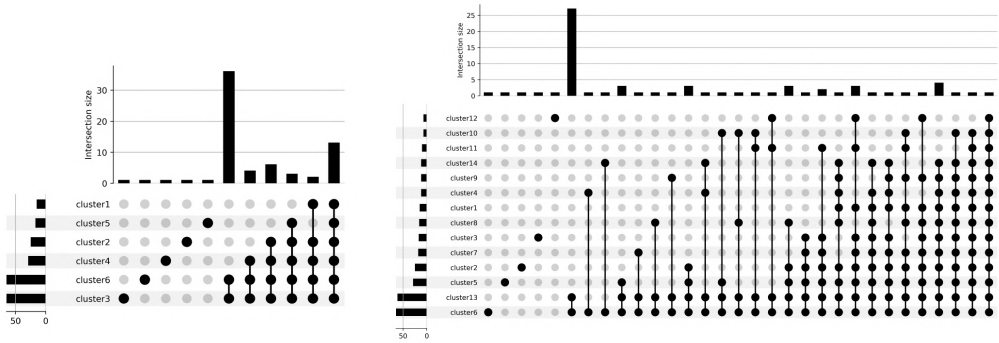


Fig. S2. Upset plot comparing the clusters obtained from the network partitioning of the airports multilayer network with different threshold values. For the threshold values equal to 0.7 and 0.9 we display one run of the partition algorithm (top charts). For the threshold values equal to 0.8 and 0.9, we display the upset plot from two different runs of the partition algorithm (middle and bottom charts). For each upset plot, the left barplot represents the total size of each cluster. Every existing intersection is represented by the bottom plot, which represents the combination matrix, and their occurrence is shown on the top barplot.

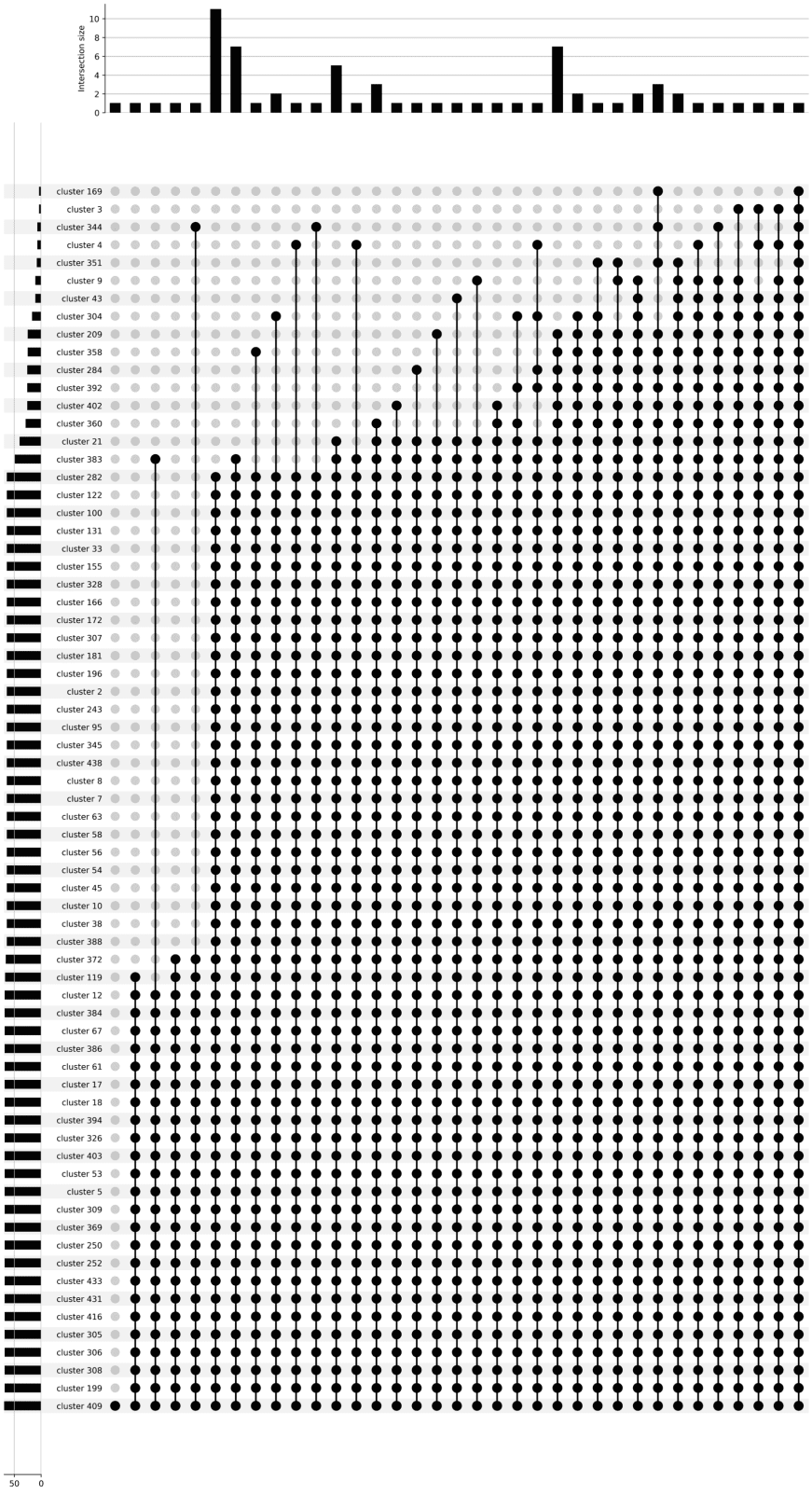


Fig. S3. Upset plot comparing the clusters obtained from the network partitioning of the disease monoplex network with a threshold value equal to 0.7. We use the exhaustive partition algorithm, where we defined clusters from each node considered as a starting seed. The left barplot represents the total size of each cluster. Every existing intersection is represented by the bottom plot, which represents the combination matrix, and their occurrence is shown on the top barplot. The name of the cluster is based on the name of the starting seed, in other words: cluster + name of the starting seed.

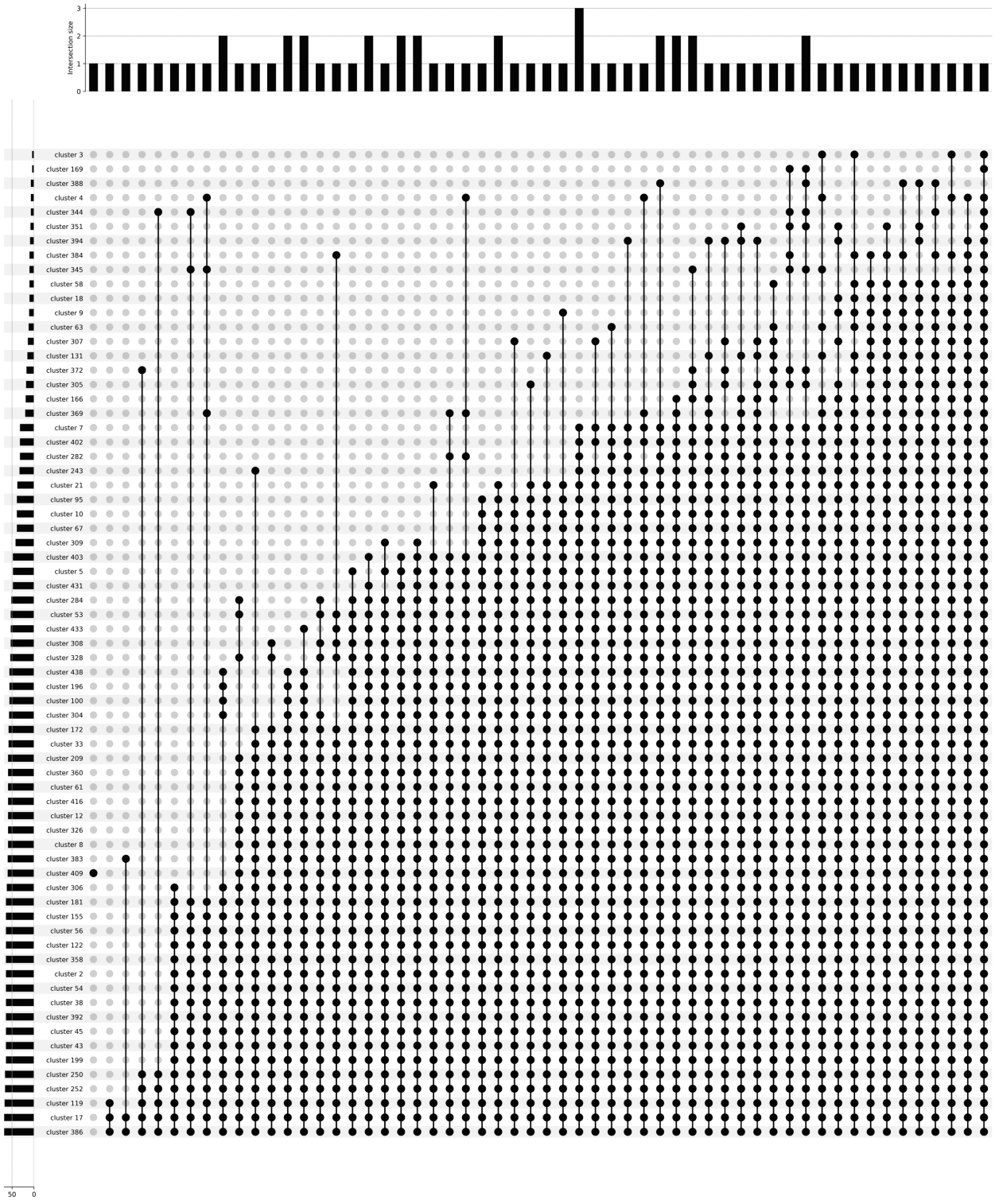


Fig. S4. Upset plot comparing the clusters obtained from the network partitioning of the disease monoplex network with a threshold value equal to 0.8. We use the exhaustive partition algorithm, where we defined clusters from each node considered as a starting seed. The left barplot represents the total size of each cluster. Every existing intersection is represented by the bottom plot, which represents the combination matrix, and their occurrence is shown on the top barplot. The name of the cluster is based on the name of the starting seed, in other words: cluster + name of the starting seed.

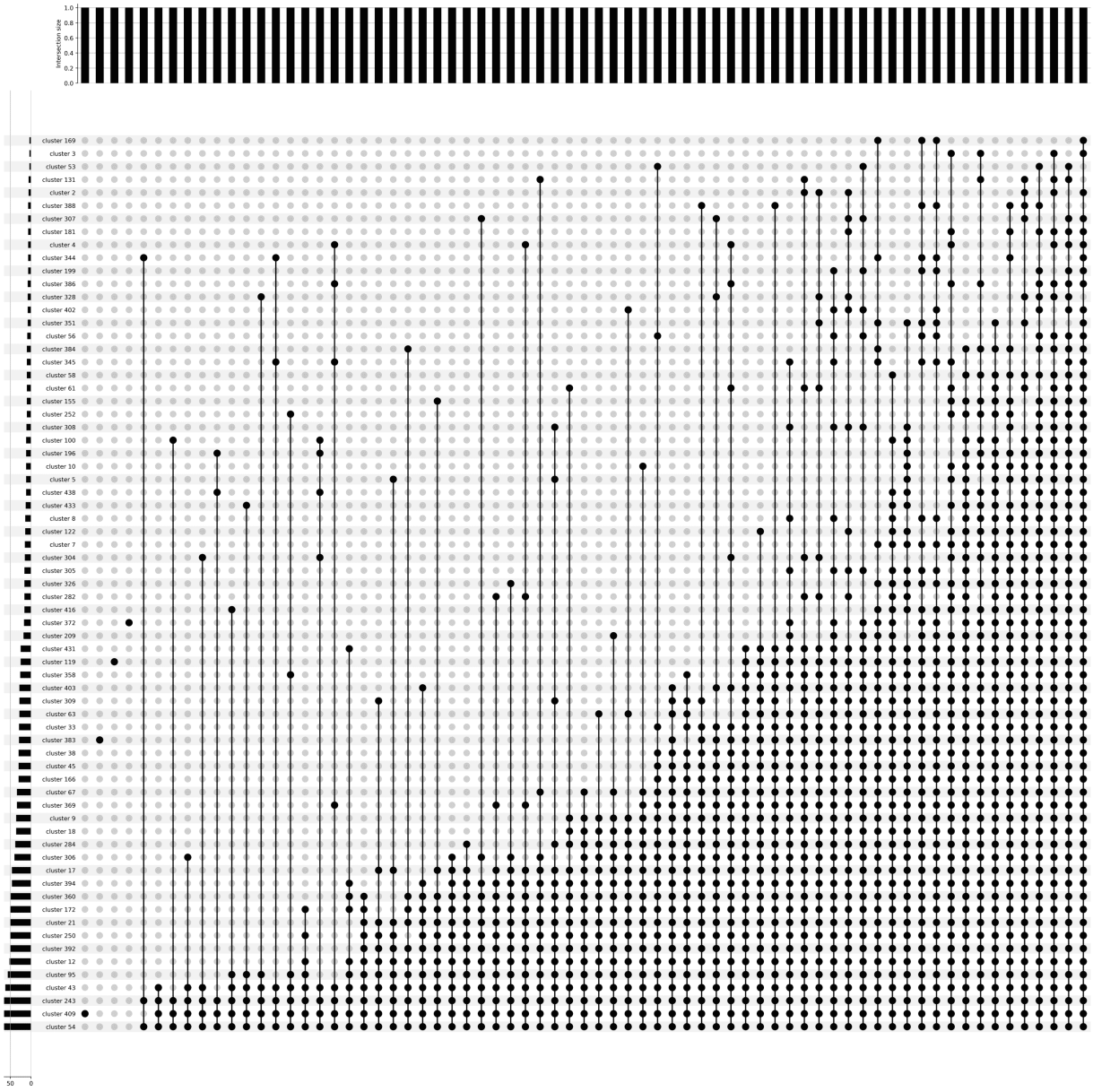


Fig. S5. Upset plot comparing the clusters obtained from the network partitioning of the disease monoplex network with a threshold value equal to 0.9. We use the exhaustive partition algorithm, where we defined clusters from each node considered as a starting seed. The left barplot represents the total size of each cluster. Every existing intersection is represented by the bottom plot, which represents the combination matrix, and their occurrence is shown on the top barplot. The name of the cluster is based on the name of the starting seed, in other words: cluster + name of the starting seed.

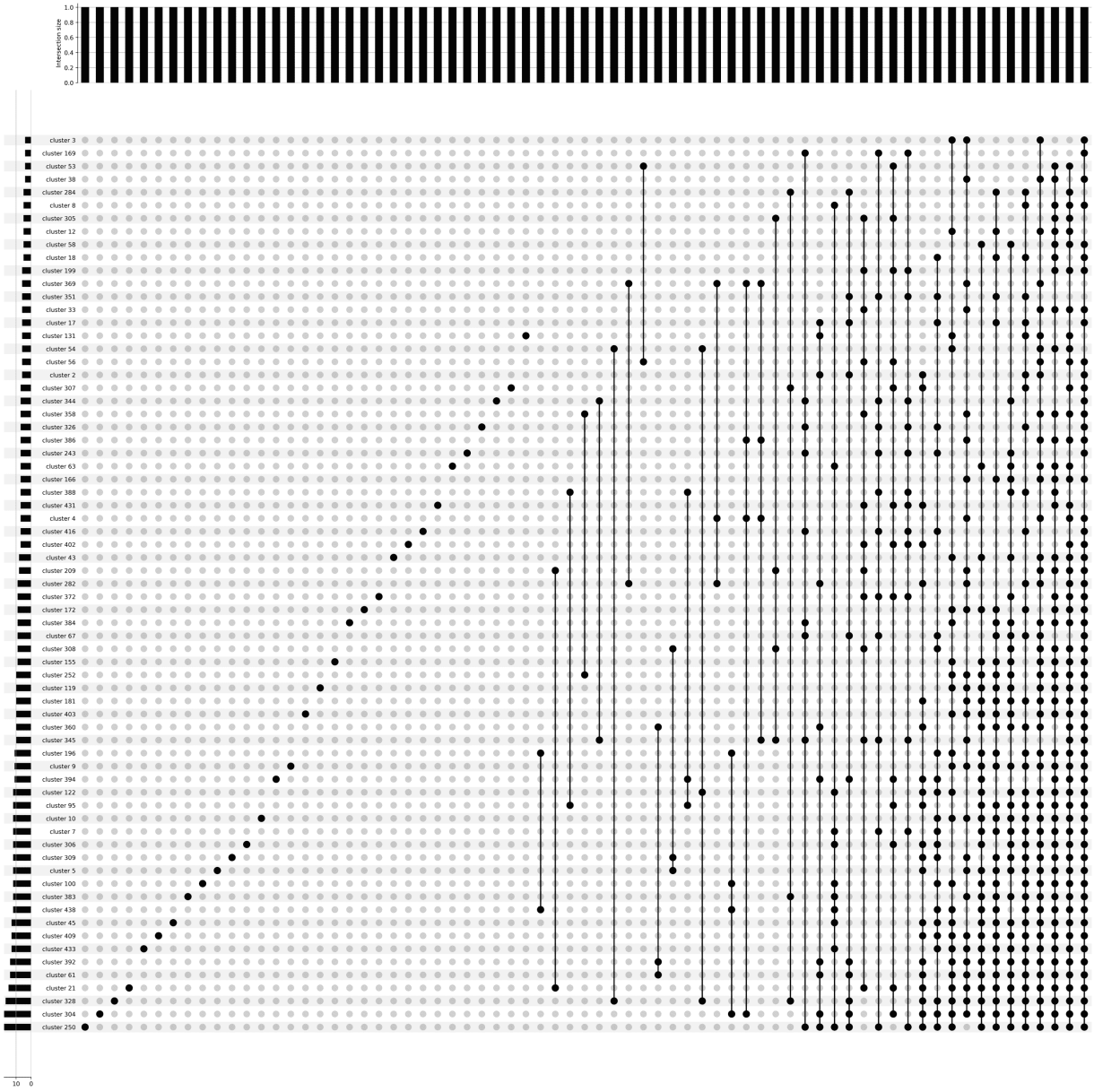


Fig. S6. Upset plot comparing the clusters obtained from the network partitioning of the disease monoplex network with a threshold value equal to 1.0. We use the exhaustive partition algorithm, where we defined clusters from each node considered as a starting seed. The left barplot represents the total size of each cluster. Every existing intersection is represented by the bottom plot, which represents the combination matrix, and their occurrence is shown on the top barplot. The name of the cluster is based on the name of the starting seed, in other words: cluster + name of the starting seed.