UNIVERSITÉ DU QUÉBEC À MONTRÉAL

L'INDICE DE TRANSFERT, NETUNIFRAC ET QUELQUES DISTANCES DE PLUS COURTS CHEMINS UTILES POUR L'ANALYSE DE COMMUNAUTÉS DANS LES RÉSEAUX DE SIMILARITÉ DE SÉQUENCES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

HENRY XING

JUILLET 2020

REMERCIEMENTS

La réalisation de ce mémoire a été rendue possible grâce à mon entourage à qui je voudrais témoigner ma reconnaissance.

En premier lieu, j'aimerais remercier mon directeur, Vladimir Makarenkov, pour sa disponibilité et son effort à me pousser dans ce projet ainsi que pour son soutien financier, et mon co-directeur, Steven Kembel, pour ses conseils précieux et pour son soutien financier.

Je désire aussi remercier mes collègues et membres du laboratoire pour le temps passé ensemble et leur support intellectuel, qui m'ont aidé à avancer tout au long de ce parcours.

J'exprime ma reconnaissance envers ma famille et mes amis qui m'ont apporté leur soutien moral et leur confiance. Un grand merci à ma compagne Cindy Zheng, qui m'a donné un appui émotionnel indispensable pour achever cet ouvrage.

TABLE DES MATIÈRES

LISTE DES FIGURES vii

LISTE DES TABLEAUX xi

LISTE DES ABRÉVIATIONS, DES SIGLES et DES ACRONYMES xiii

RÉSUMÉ xv

ABSTRACT xvi

INTRODUCTION 1

CHAPITRE I NOTIONS DE BASE 8

1.1 Théorie des graphes 8

1.2 Réseaux de similarité de séquences en biologie 10

1.3 Plus court chemin 14

CHAPITRE II MÉTHODES 17

2.1 Définitions des nouvelles distances et de l'indice de *Transfert* 18

2.2 Exemple des calculs de distances 31

2.3 Exemple de calculs des distances et indices sur des arbres phylogénétiques et RSS simples 33

2.4 Algorithme pour un calcul rapide de l'indice de *Transfert* 36

CHAPITRE III UNE ÉTUDE DE SIMULATIONS ET APPLICATIONS DES NOUVELLES DISTANCES ET INDICES À DES DONNÉES RÉELLES 40

3.1 Études de simulations 40

3.2 Études de simulations sur de larges données synthétiques 44

CHAPITRE IV Résultats 46

4.1 Le réseau d'Esophagus 46

4.2 Analyse des regroupements TetA et CAT dans un réseau de gènes de résistance aux antibiotiques 49

4.3 Reconstruction des réseaux TetA et CAT 54

CONCLUSION 63

APPENDICE A CODE DU PACKAGE R NETFRAC 67

APPENDICE B LES ARTICLES PUBLIÉS LORS DE CE PROJET DE MAÎTRISE 89

RÉFÉRENCES 113

LISTE DES FIGURES

Figure 1.1 : Un graphe à 5 nœuds 9

Figure 1.2 : Représentation de 3 sous-graphes provenant du graphe de la Figure 1.1. En A, un sous-graphe. En B, il s'agit d'un sous-graphe induit car toutes les arêtes entre les nœuds 2 à 5 du graphe de la Figure 1.1 sont représentées dans ce sous-graphe. En C, un sous-graphe couvrant, avec tous les nœuds du graphe original, mais manquant certaines arêtes. 10

Figure 1.3 : Sous-graphes induits de la communauté jaune (A) et bleue (B) de la Figure 1.1. 11

Figure 1.4: Création d'un RSS à partir de BLAST (Forster *et al.*, 2015). Le seuil de similarité détermine la connectivité du réseau final. 13

Figure 1.5 : Plus court chemin dans un arbre. Nous partons du nœud 8 vers le nœud 13 en passant par la racine, qui est le plus proche commun ancêtre des deux nœuds. 15

Figure 2.1 : Exemple de calcul de l'indice de *Transfert* pour un réseau de similarité de séquences contenant *N* espèces de la communauté bleue et *N* espèces de la communauté jaune. Cas A: les communautés bleue et jaune sont complètement séparées (e.g., les sous-graphes sont connectés par une seule arête, représentée par la ligne pointillée, ou ne sont pas connectés du tout); Cas B: une espèce de la communauté bleue est affectée par un transfert de gènes d'une espèce de la

communauté jaune; Cas C: *n*/2 espèces de la communauté bleue sont affectées par des transferts de gènes de espèces de la communauté jaune. Cas D: toutes les *N* espèces de la communauté bleue sont affectées par des transferts de gènes des espèces de la communauté jaune. 26

Figure 2.2: Exemple de calcul de l'indice de *Transfert alternatif* pour un réseau de similarité de séquences contenant *N* espèces de la communauté bleue et *N* espèces de la communauté jaune. Cas A: les communautés bleue et jaune sont complètement séparées (e.g., les sous-graphes sont connectés par une seule arête, représentée par la ligne pointillée, ou ne sont pas connectés du tout); Cas B: une espèce de la communauté bleue est affectée par un transfert de gènes d'une espèce de la communauté jaune; Cas C: *n*/2 espèces de la communauté bleue sont affectées par des transferts de gènes de espèces de la communauté jaune. Cas D: toutes les *N* espèces de la communauté bleue sont affectées par des transferts de gènes des espèces de la communauté jaune 30

Figure 2.3 : RSS avec cinq nœuds (deux nœuds bleus et trois nœuds jaunes, représentant deux communautés d'espèces différentes) et six arêtes. Les longueurs d'arêtes sont indiquées à côté de chaque arête. 31

Figure 2.4 : Cinq RSS simples contenant des espèces des communautés jaune (Y) et bleu (B). Le nombre à côté des arêtes sont leur poids, et les nœuds sont numérotés. 33

Figure 2.5 : Six arbres phylogénétiques enracinés utilisés comme exemples pour les calculs de distances et indices définis dans ce mémoire. Les poids des arêtes sont indiqués en noir. Les espèces appartenant aux communautés d'espèces bleues (B) et rouges (R) sont initialement associées aux feuilles des arbres. Par la suite, chaque nœud interne est coloré soit en en bleu, en rouge, ou en bleu et rouge,

dépendamment des feuilles qui descendent de ce nœud. Nous remontons ainsi jusqu'au premier branchement en bas de la racine. 35

Figure 3.1: (a) erreur de détection de THG, qui est la différence absolue entre le nombre de transferts générés et retrouvés, pour RIATA-HGT (losanges blancs), HGT-Detection (carrés blancs) et l'Algorithme 1 (triangles gris) par rapport au nombre de feuilles de l'arbre ou de nœuds du réseau. Chaque valeur représente une moyenne obtenue sur 100 arbres pour chaque taille considérée; (b) temps d'exécution moyen (en secondes) pour chaque algorithme par rapport au nombre de feuilles ou de nœuds. 42

Figure 3.2: Diagrammes en boîte du score F1 (a) et du rappel (b) obtenus pour l'Algorithme 1 sur des RSS aléatoires (de 10 à 100 nœuds). L'axe des abscisses indiquent le nombre de nœud dans le réseau. 43

Figure 3.3 : Diagrammes en boîte du score F1 (a) et du rappel (b) obtenus dans la deuxième simulation quand l'Algorithme 1, calculant l'indice de *Transfert*, a été utilisé sur des RSS de 1000 nœuds. L'axe des abscisses indique les différents pourcentages de séquences transférées (i.e., nœuds du réseau) entre deux communautés d'espèces (i.e., 0% et 10% de transferts; 0% d'espèces de la première communauté étaient affectées par les THG des espèces provenant de la deuxième communauté, et 10% de la deuxième communauté étaient affectées par THG des espèces de la première communauté). 45

Figure 4.1 : Les RSS pour les gènes de résistance aux antibiotiques de TetA et CAT de la section *Supporting information* S4 dans Fondi et Fani (2010). Les nœuds représentent des protéines (GRA) d'organismes bactériens et les arêtes proviennent de la mesure de WIV. Les nœuds sont colorés selon l'habitat assigné par la base de données GOLD: les nœuds rouges pour les organismes dans les

hôtes, les nœuds bleus pour les organismes aquatiques, les nœuds verts pour les organismes dans plusieurs ou tous les habitats (omniprésent) et les nœuds gris pour les organismes manquant dans la base de données GOLD. Les nœuds ou arêtes redondants ont été enlevés du réseau. 52

Figure 4.2: Chaque graphique montre la relation entre le nombre d'arêtes dans le RSS et le seuil choisi pour les réseaux de TetA (A) et CAT (B). Pour les deux réseaux 4 points pivots sont présentés comme des candidats possibles pour choisir le seuil des réseaux. La valeur de seuil de 0.904 a été choisie pour les deux réseaux TetA et CAT. 55

Figure 4.3 : Les réseaux taxonomiques TetA et CAT pour les communautés d'espèces Actinobactéries (turquoise), Bacilli (jaune), γ-protéobactéries (marron), et Autres bactéries (rose) construits en utilisant de la matrice de distances JTT, qui a été calculée à partir de séquences alignées avec *Muscle*. Le seuil de distance de 0.904 pour les réseaux TetA et CAT a été utilisé pour déterminer les arêtes du réseau (i.e., une arête entre deux nœuds a été ajoutée si la distance est plus petite que ce seuil). 58

Figure 4.4 : Les réseaux environnementaux TetA et CAT pour les communautés d'espèces Hôte (rouge), Omniprésent (vert), Aquatique (bleu), et Inconnu (gris) construit en utilisant la matrice de distances JTT, qui a été calculée à partir de séquences alignées avec *Muscle*. Le seuil de distance de 0.904 pour les réseaux TetA et CAT a été utilisé pour déterminer les arêtes du réseau. 60

LISTE DES TABLEAUX

Table 1.1 : Exemple de matrice de distance entre des séquences de gènes. Le nombre sur la première ligne indique le nombre d'entrées dans la matrice. La matrice est carrée, les colonnes représentent les séquences dans le même ordre que les rangées. 12

Table 2.1 : Numérateur, dénominateur et résultat de chaque nouvelle distance et nouvel indice pour le RSS présenté dans la Fig. 2.3. 32

Table 2.2 : Valeurs des distances et indices obtenus pour les RSS représentés dans la Figure 2.4. 34

Table 2.3 : Valeurs des distances et indices obtenues pour les arbres de la Figure 2.5. 35

Table 4.1 : Nombre de nœuds et d'arêtes pour chaque comparaison d'une paire de communautés, ainsi que le nombre total de nœuds et d'arêtes (construits en utilisant un BLAST local sur un seuil de 97%)..... 47

Table 4.2 : Les valeurs des distances entre les communautés bactériennes B-C, C-D et B-D pour le jeu de données d'Esophagus pour notre RSS..... 48

Table 4.3 : Valeurs des distances et des indices obtenues pour le réseau TetA de la Figure 4.1. La direction des mesures de transferts est donnée par *reverse* et *direct*. 53

Table 4.4 : Valeurs des distances et indices obtenues pour le réseau CAT de la Figure 4.1. 53

Table 4.5 : Indice de *Transfert* dans les communautés de TetA selon la classification taxonomique. La valeur de 1 indique qu'aucun transfert n'a eu lieu de la communauté de la ligne vers la communauté de la colonne. L'écart-type des valeurs de l'indice obtenu pour les 4 points pivots présentés sur la Figure 4.2 est indiqué entre parenthèses après chaque valeur. 57

Table 4.6 : Indice de *Transfert* pour les communautés taxonomiques du réseau CAT. 59

Table 4.7 : Indice de *Transfert* pour les communautés environnementales du réseau TetA..... 61

Table 4.8 : Indice de *Transfert* pour les communautés environnementales du réseau CAT. 61

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

GRA : gène de résistance aux antibiotiques

RSS : réseau de similarité de séquences

Spelp : Shortest path edge length proportions

Spep: Shortest path edge proportions

Spinp: Shortest path internal nodes proportions

Spp : Shortest path proportions

THG : transfert horizontal de gène

RÉSUMÉ

Les arbres phylogénétiques et leurs méthodes d'analyse ont joué un rôle-clé dans plusieurs études d'évolution, d'écologie et de bioinformatique. Alternativement, les réseaux phylogénétiques ont aussi été largement utilisés dans l'analyse et la représentation des processus d'évolution complexes, qui ne peuvent être étudiés adéquatement par les méthodes phylogénétiques traditionnelles. Ces processus incluent, entres autres, l'hybridation, les transferts horizontaux de gènes (THG) et la recombinaison génétique. De nos jours, les réseaux de similarités de séquences (RSS) et similarités de génomes deviennent des outils efficaces pour l'analyse de communautés dans des jeux de données moléculaires. Ces réseaux peuvent être utilisés pour une variété de problèmes d'évolution, tels que l'identification des THG, des gènes et génomes mosaïques, et l'étude des holobiontes. Les plus courts chemins dans un arbre phylogénétique permettent d'estimer la distance d'évolution entre les espèces. Nous montrons comment le concept des plus courts chemins peut être étendu aux RSS avec cinq nouvelles distances, soient *NetUniFrac*, *Spp*, *Spep*, *Spelp* et *Spinp*, et l'indice de *Transfert*. Ces nouvelles distances peuvent être vues comme analogues de la distance traditionnelle *UniFrac* appliquée sur les arbres pour mesurer la dissimilarité entre des communautés d'espèces, alors que l'indice de *Transfert* sert à estimer le ratio et la direction des transferts, ou la dispersion des espèces, entre différentes communautés d'espèces phylogénétiques ou écologiques. De plus, *NetUniFrac* et l'indice de *Transfert* peuvent être calculés dans un temps linéaire par rapport au nombre d'arêtes dans le réseau. Nous montrons comment ces nouvelles mesures sont utilisées pour analyser des réseaux de microbiomes et de gènes de résistance aux antibiotiques (GRA). Notre programme *NetFrac*, implémenté en R et en C, avec les codes sources, est disponible gratuitement sur GitHub à l'adresse URL suivante : https://github.com/XPHenry/Netfrac.

Mots clés : Phylogénie, réseau de similarités de séquences, réseau phylogénétique, plus court chemin, transferts horizontaux de gènes.

ABSTRACT

Phylogenetic trees and the methods for their analysis have played a key role in many evolutionary, ecological and bioinformatics studies. Alternatively, phylogenetic networks have been widely used to analyze and represent complex reticulate evolutionary processes that cannot be adequately studied using traditional phylogenetic methods. These processes include, among others, hybridization, horizontal gene transfer and genetic recombination. Nowadays, sequence-similarity and genome-similarity networks have become an efficient tool for community analysis of large molecular datasets in comparative studies. These networks can be used for tackling a variety of complex evolutionary problems such as the identification of horizontal gene transfer events, the recovery of mosaic genes and genomes, and the study of holobionts. The shortest path in a phylogenetic tree is used to estimate evolutionary distances between species. We show how the shortest path concept can be extended to sequence similarity networks by defining five new distances, NetUniFrac, Spp, Spep, Spelp and Spinp, and the Transfer index, between species communities present in the network. These new distances can be seen as network analogues of the traditional UniFrac distance used to assess dissimilarity between species communities in a phylogenetic tree, while the Transfer index is intended for estimating the rate and direction of gene transfers, or species dispersal, between different phylogenetic, or ecological, species communities. Moreover, NetUniFrac and the Transfer index can be computed in linear time with respect to the number of edges in the network. We show how these new measures can be used to analyze microbiota and antibiotic resistance gene similarity networks. Our NetFrac program, implemented in R and C, along with its source code, is freely available on GitHub at the following URL address: https://github.com/XPHenry/Netfrac.

Keywords: Phylogeny, sequence similarity network, shortest path distance, horizontal gene transfer.

INTRODUCTION

Dans le domaine de la biologie, et plus particulièrement de l'écologie, les arbres phylogénétiques (arbres additifs ou X-arbres) sont utilisés dans de nombreuses applications, telles que la visualisation de données et le calcul de mesures statistiques. Différentes mesures de distance entre les arbres, telles que la distance de *Robinson et Foulds* ou la distance des quartets (Felsenstein, 2004), et de biodiversité entre les communautés d'espèces présentes dans les arbres, telles que la distance *UniFrac* et ses variantes (Lozupone et al., 2011), ont été proposées et amplement utilisées. Cependant, il est bien connu que les arbres ont leurs limites quant à la représentation de l'évolution, ainsi que de la biodiversité des espèces (Huson et Bryant, 2005). Les réseaux phylogénétiques et les réseaux de similarité de séquences (RSS) sont souvent mieux adaptés que les arbres pour représenter des phénomènes évolutifs complexes, tels que l'hybridation, l'endosymbiose, la recombinaison ou le transfert horizontal de gènes (Bapteste et *al.,* 2012).

Dans un RSS, un nœud représente une séquence d'un gène ou d'un génome d'intérêt, alors qu'une arête connecte deux nœuds avec une similarité qui satisfait un seuil donné. Le réseau peut être défini de différentes façons, le plus souvent en utilisant les matrices de distances produites par des outils de calcul, tels que BLAST (*basic local alignment search tool*) (Altschul *et al.*, 1990) et Mothur (Schloss *et al.*, 2009) ou des modèles évolutifs de mutations comme JTT, Kimura ou Jukes-Cantor (Xiong, 2006). Deux nœuds sont connectés par une arête s'ils montrent une similarité supérieure à un seuil *h*. Si les arêtes sont pondérées, qui correspond habituellement à la valeur des distances entre les séquences, alors le réseau est appelé un RSS *pondéré*. Les réseaux de similarité sont complémentaires aux arbres et réseaux phylogénétiques, i.e., les split-

graphes ou les réseaux de transferts horizontaux de gènes (THG). Ils offrent non seulement un moyen différent de représenter des similarités entre les séquences/espèces, mais proposent également un cadre de recherche souple d'analyse de données métagénomiques (Bapteste et al., 2013, Jachiet *et al.*, 2013, Pathmanathan *et al.,* 2018). Des mesures de biodiversité équivalentes à la métrique *UniFrac*, qui permet d'estimer la distance entre les communautés d'espèces dans un arbre phylogénétique, ont récemment été proposées pour des split-graphes (Parks et Beiko, 2012) et des réseaux phylogénétiques généraux (Wicke et Fischer, 2018). Plusieurs autres travaux traitant des RSS dans l'analyse de données phylogénétiques démontrent aussi la possibilité d'utiliser les RSS pour améliorer le traitement des données biologiques.

Le travail de Park et Beiko (2012) se concentre sur les split-graphes, un type de réseau implicite, soit un réseau qui permet d'illustrer plusieurs, voire tous les scénarios d'évolution possibles. En effet, tel que le nom le suggère, les split-graphes sont des réseaux qui séparent les nœuds en plusieurs bipartitions possibles avec les mêmes nœuds d'un bord et de l'autre. Graphiquement, il s'agit d'arêtes parallèles où la jonction des lignes n'est plus un nœud, car ce nœud serait répété à travers ces arêtes. En utilisant les concepts inhérents des split-graphes, tels que les "splits" uniques, partagés, enracinés et externes, les auteurs créent une version adaptée des mesures de distance en phylogénie, comme la distance de *Manhattan* ou *UniFrac*. Cela reste cependant limité aux split-graphes, et ne s'applique pas à d'autres types de réseaux.

Pour les événements d'introgression, soit de transfert de gène d'une espèce vers une autre (transferts horizontaux), Bapteste *et al.* (2012) suggèrent d'utiliser des RSS. Ils y introduisent les distances par motifs, où une certaine structure du graphe, par exemple 3 nœuds reliés par deux arêtes, constitue un motif de base que nous cherchons dans l'ensemble du réseau. Les introgressions, sont groupées en deux grandes familles: les gènes fusionnés et les membres de lignées multiples, qui forment les deux spectres

opposés. Les gènes fusionnés, beaucoup plus fréquent, découlent des gènes de deux ou plusieurs espèces pour finir par n'en former qu'un seul. Les membres de lignées multiples sont des interacteurs différents qui forment une coalition structurée, tel qu'un biofilm, mais qui se réplique indépendamment les uns des autres. Bapteste *et al.* (2012) décrivent les gènes mosaïques en M-P3, ceux où deux nœuds (représentant les interacteurs) sont reliés par un troisième, où les arêtes proviennent cependant d'unités différentes. Les espèces A et B sont reliées à C par les gènes *X* et *Y* respectivement, ce qui constitue un lien indirect entre A et C, car ce ne sont pas les mêmes gènes qui représentent la similarité de A vers B et de B vers C. Ces M-P3 permettent de détecter les événements d'introgression dans les réseaux de similarités et d'interactions.

Dans son étude sur les origines et l'évolution des eucaryotes, Alvarez-Ponce *et al.* (2013) préfèrent aussi se baser sur les RSS, car la complexité et la profondeur des relations entre les domaines de la vie est difficilement analysable par les méthodes phylogénétiques traditionnelles. Entre autres, l'utilisation des arbres phylogénétiques ne permet pas de distinguer les séquences hautement divergentes de la même famille, puisque les méthodes de regroupement utilisées détectent majoritairement les séquences qui sont proches dans une sortie de BLAST (Gribaldo et Philippe, 2002). De plus, l'alignement des séquences multiples ne permet pas de construire adéquatement un arbre avec la présence d'un nombre élevé de substitutions. La divergence accumulée avec le temps efface ou minimise beaucoup de signaux phylogénétiques entre des séquences homologues. Finalement, les hypothèses générées sur des séquences très divergentes sont toujours dépendantes du modèle d'évolution utilisé (Cox *et al.* 2008). En suivant un modèle très similaire aux gènes mosaïques décrit par Bapteste *et al.* (2012), Alvarez-Ponce construit son RSS en se basant sur des composantes connexes, qui permettent de garder les séquences divergentes mais homologues dans un même regroupement, tout en soulignant les différences entre ces mêmes gènes par une similarité fluctuante. Leurs résultats montrent la robustesse des RSS quant à l'analyse de l'évolution des gènes avec des phénomènes d'évolution complexes.

Cette pensée s'applique aussi en écologie, alors que Forster *et al.* (2015) étudient la dispersion entres des organismes marins. Traditionnellement basé sur l'analyse des regroupements de séquences, les données écologiques cachent des motifs de diversité qui ne sont pas apparents seulement par regroupement de séquences analogues. Les RSS permettent de rajouter de l'information à l'aide de la topologie des connections pondérées entre les séquences de différentes communautés. Un compte de phylotypes dans deux jeux de données séparés ne montre pas d'information supplémentaire sur la similarité des phylotypes, alors qu'ils peuvent être comparés directement dans un réseau de similarité unique combinant les deux jeux de données. De plus, la quantité de données étant souvent massive dans les jeux de données écologiques, l'alignement multiples, la reconstruction d'arbres et la visualisation des arbres se font beaucoup plus difficilement (Halary *et al.*, 2010).

Dans un article récent de Wicke et Fischer (2018), les efforts sont plutôt dirigés sur des réseaux phylogénétiques. Les réseaux phylogénétiques sont un autre type de réseaux, qui se rapprochent plus de la structure des arbres, cependant ils sont permissifs au niveau des structures cycliques. Ils articulent les événements d'hybridation ou de transfert horizontaux de gènes sans perdre l'information évolutive, ou la transmission verticale. Les auteurs y présentent aussi leur programme qui permet de calculer leur distance entre les réseaux phylogénétiques. Ces réseaux présentent aussi davantage de possibilités d'analyses par rapport aux arbres phylogénétiques.

En général, il y a un besoin de trouver des alternatives aux méthodes traditionnelles de phylogénie pour concaténer, visualiser et analyser des grands jeux de données moléculaires (Gligorijević et Pržulj, 2015). L'avènement des NGS (*Next-generation sequencing*) permet le traitement massif des séquences de données, donnant lieu à des nouvelles catégories de biologie moléculaires – les « -omiques », transcriptomique, génomique, protéomique, épigénomique, etc. (Hawkins *et al.*, 2010). Cela provoque un goulot d'étranglement (*bottleneck*) où la quantité de données brutes dépasse largement

la capacité des méthodes d'analyse, qui sont trop peu efficaces ou lents pour traiter ce volume de données (Gomez-Cabrero *et al.*, 2014). Cela mène donc au concept de l'intégration des données, où ces différents domaines « -omiques » sont regroupés sous une même structure pour permettre une vue d'ensemble des données, qui sont autrement trop volumineuses à analyser séparément, mais permettent d'extraire des informations cruciales sur le fonctionnement biologique d'un système. Ce domaine d'étude est connu sous le nom de la biologie des systèmes (Joyce et Palsson, 2006). Une solution est l'utilisation de réseaux, que ce soit des RSS, des réseaux d'interactions moléculaires ou des réseaux d'associations fonctionnelles (Winterbach *et al.*, 2013). Il est aussi possible d'utiliser des réseaux bayésiens, qui combine des éléments de la théorie des graphes avec les probabilités (Ben-Gal *et al.*, 2005), plus souvent utilisés dans la prédiction des fonctions des gènes et la modélisation de la transduction des signaux des protéines (Sachs *et al.*, 2005). En apprentissage machine, les méthodes basées sur l'astuce du noyau (*kernel-based*), tels que le SVM (*support vector machine*) et le PCA (*principal component analysis*) peuvent aussi être utilisées pour la prédiction de fonctions de gènes ou la recherche des médicaments potentiels pour une maladie, ou l'utilisation de médicament existant sur une maladie différente (Wang *et al.*, 2013).

En se basant sur les informations collectées dans ces différents travaux, nous avons décidé de concentrer nos efforts sur des distances qui se calculent plus spécifiquement sur des RSS, et possiblement sur des réseaux d'interactions. Dans l'analyse des RSS, les concepts de centralité, connectivité et modularité sont souvent utilisés pour comparer des graphes ou leurs parties (Girvan et Newman, 2002). En revanche, lorsqu'il s'agit de l'analyse de communauté à l'aide des RSS, peu de méthodes permettent de faire la comparaison entre les différentes communautés. Il s'agit d'une analyse qui serait intéressante à faire, surtout dans un contexte biologique, car différentes communautés biologiques ou écologiques peuvent avoir différents rôles à jouer dans un écosystème tout en partageant des caractéristiques similaires.

Classiquement, cette analyse est faite sur des arbres phylogénétiques à l'aide de la mesure d'*UniFrac* (Lozupone *et al.*, 2005), très connue dans ce domaine.

En calculant la distance entre les communautés, nous obtenons des informations plus précises sur la composition de chaque communauté. Dans le cas de grands réseaux, qui sont plus difficiles à visualiser, cela permet d'obtenir une statistique supplémentaire sur le comportement du réseau du point de vue des communautés qui la composent (Newman, 2010). Par exemple, si nous voulons comparer la flore microbienne de plusieurs patients ou personnes saines, nous pouvons calculer la distance entre les communautés de gènes, ou chaque communauté représente une partie du microbiome d'une personne. Bien que cette information ne permette pas de conclure sur l'état d'une personne, elle aide définitivement à l'analyse des microbiomes intestinaux en sachant quelle direction prend chaque microbiome par rapport à une personne saine. Par la suite, il est possible de comparer les microbiomes des patients sains à des patients malades, et soutirer des informations qui permettraient de comprendre les maladies causées par un changement dans la flore microbienne ou l'impact causé par la diète de la personne (Walter et Ley, 2011). Il est ensuite possible de comparer différentes communautés de microbiome résidant dans des parties distinctes du tract digestifs, toujours en utilisant un RSS.

Dans ce mémoire, nous proposons des nouvelles mesures de distances, dont l'extension directe de la distance *UniFrac*, qui peuvent être utilisées pour estimer la dissimilarité entre différentes communautés d'espèces (de gènes, d'individus ou d'objets) associées à des nœuds d'un RSS. À la différence de la distance des *Motifs*, qui est basée sur le calcul des proportions des motifs spécifiques impliquant les espèces de la même communauté (Bapteste et al., 2012), la plupart de nos distances utilisent le concept du plus court chemin entre les espèces de la même communauté. Le plus court chemin en phylogénie est étroitement associé avec la distance d'évolution entre les espèces considérées.

Bien que nous utilisions des réseaux de gènes comme exemples, nos mesures peuvent avoir des interprétations différentes selon le type de réseaux. Dans le cas des réseaux écologiques, les distances représentent des phénomènes écologiques, telle que la dispersion des espèces entre communautés, indépendamment des transferts des gènes. L'interprétation des distances et indices introduits dans ce mémoire varie selon le contexte, qui est déterminé par le contenu des jeux de données. Les études mentionnées considèrent des données différentes, mais tant que les données restent comparables, il est possible de les structurer dans un RSS.

CHAPITRE I

NOTIONS DE BASE

Dans ce chapitre, quelques notions de base sur la théorie des graphes seront présentées. Entre autres, les définitions qui entourent les graphes (ou réseaux), la création de sous-graphes à partir de communautés. Ensuite, nous présenterons plus précisément des réseaux de similarité de séquences dans un contexte biologique, et comment passer de données biologiques à une structure de RSS. Enfin, nous considérons avec le concept du plus-court chemin, plus connu avec les arbres phylogénétiques, mais tout aussi présent dans les RSS.

1.1 Théorie des graphes

Un graphe G est composé des ensembles N et E, $G = (N, E)$, où N est l'ensemble des nœuds et E est l'ensemble des arêtes. Les ensembles N et E sont toujours finis, c'est-à-dire que nous considérons seulement des graphes finis. Une arête $\{i,j\}$, où $i,j \in N$ et $i \neq j$, relie deux nœuds i et j et peut être dénotée ij. De ce fait, ij et ji représentent exactement la même arête; les nœuds i et j sont les extrémités de l'arête. Pour les RSS avec lesquels nous travaillons, nous ne pouvons pas avoir plusieurs copies de la même arête, c'est-à-dire deux nœuds reliés avec plus qu'une arête, autrement existant dans les multigraphes. Si $ij \in E$, alors les nœuds i et $j \in N$, sont dits adjacents ou voisins, et incidents avec l'arête ij. Similairement, deux arêtes sont adjacentes si elles possèdent exactement une extrémité commune (Bollobás, 2013).

Les graphes sont donc décrits par deux ensembles, l'ensemble des nœuds et des arêtes, mais il est évidemment plus facile de décrire un graphe en le dessinant. En fait, le graphe avec les nœuds 1,2,3,4,5 et les arêtes 12, 23, 24, 34, 35, 45 est beaucoup plus compréhensible en regardant la Figure 1.1 :

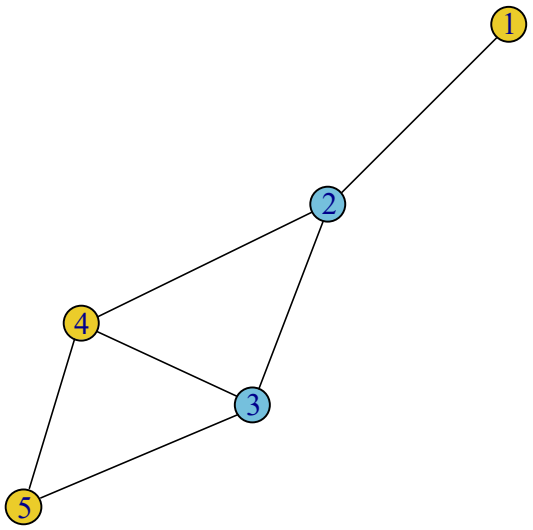


Figure 1.1 : Un graphe à 5 nœuds

Souvent, de nouveaux graphes sont construits à partir d'un graphe de départ, en utilisant une partie ou tous les éléments existants. Ces nouveaux graphes sont alors appelés des sous-graphes. Nous disons que $G' = (N', E')$ est un sous-graphe de $G = (N, E)$ si $N' \subset N$ et $E' \subset E$. Dans ce cas, nous pouvons écrire $G' \subset$ G. Si G' contient toutes les arêtes de G qui joignent les paires de nœuds dans N', alors G' est dit *induit* par N' et est dénoté par $G[N']$. Donc, un sous-graphe G' de G est un sous-graphe induit si $G' = G[N(G')]$. Si $N' = N$, alors G' est un sous-graphe *couvrant* de G (Bollobás, 2013). Ces concepts sont illustrés dans la Fig. 1.2.

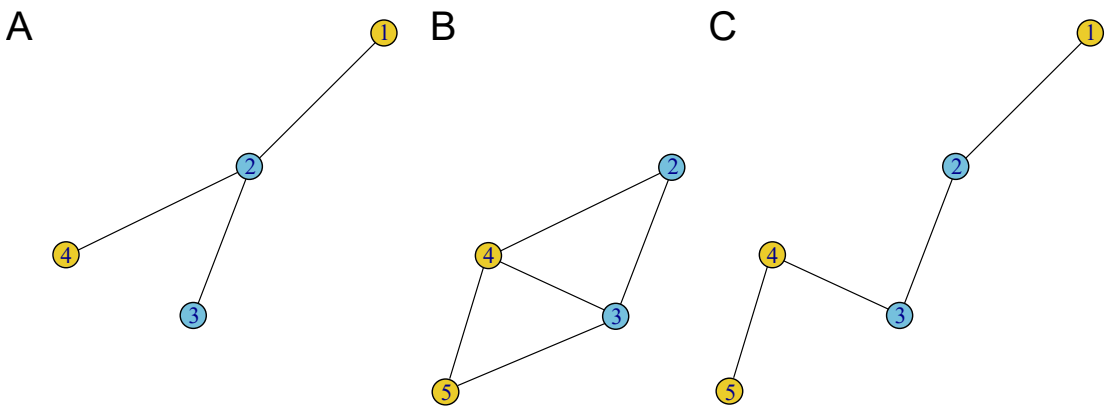


Figure 1.2 : Représentation de 3 sous-graphes provenant du graphe de la Figure 1.1. En A, un sous-graphe. En B, il s'agit d'un sous-graphe induit car toutes les arêtes entre les nœuds 2 à 5 du graphe de la Figure 1.1 sont représentées dans ce sous-graphe. En C, un sous-graphe couvrant, avec tous les nœuds du graphe original, mais manquant certaines arêtes.

1.2 Réseaux de similarité de séquences en biologie

Pour donner une profondeur à l'analyse, ainsi qu'à l'utilité générale des réseaux, les nœuds et les arêtes peuvent posséder des caractéristiques supplémentaires, tels que le(s) groupe(s) au(x)quel(s) ils appartiennent. Dans les réseaux biologiques, les groupes sont souvent des communautés, par exemple des communautés biologiques sur un certain territoire ou biome, un regroupement taxonomique ou parfois un regroupement fonctionnel, souvent la fonction de protéines semblables dans différents organismes (Stelzl *et al.*, 2005). Nous pouvons aussi créer des sous-graphes induits à partir d'un nombre limité de communautés pour mieux isoler les relations entre des communautés qui nous intéressent. Dans la Figure 1.3, nous avons isolé chaque communauté, mais dans un réseau plus grand, il est possible de créer des sous-graphes induits de plus de communautés à la fois.

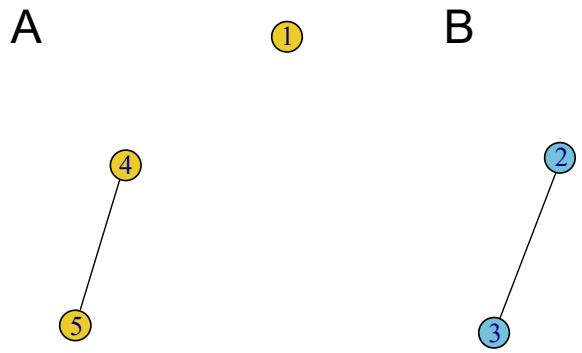


Figure 1.3 : Sous-graphes induits de la communauté jaune (A) et bleue (B) de la Figure 1.1.

Dans les RSS, qui sont des graphes, la similarité est ce qui détermine la présence et la longueur des arêtes dans le réseau. En biologie moléculaire, la similarité mesurée est celle entre les séquences génétiques (ADN ou ARN) ou protéiques d'un certain gène ou certaine protéine. Plus deux séquences de gènes se ressemblent, plus elles ont de fortes chances d'être reliées par une arête dans un RSS (Korf *et al.*, 2003). Il faut aussi déterminer un seuil de similarité, une certaine valeur qui doit être satisfaite par la similarité de deux nœuds. Par exemple, les arêtes dans un RSS sont le résultat du calcul de similarité (ou distance) entre deux nœuds. Le calcul de ces similarités peut sembler simple, puisque nous faisons affaire avec des bases nucléotidiques (A,T,G,C) ou des acides aminés, mais en réalité plusieurs facteurs doivent être considérés lors de la comparaison des séquences génétiques/protéiques, qui proviennent justement de la complexité des processus biologiques menant aux différents gènes actuels (Edgar et Batzoglou, 2006). Pour cela, il existe plusieurs outils et méthodes déjà développés, qui permettent de comparer plusieurs séquences entre elles, ou une séquence à une autre séquence prédéfinie, et en ressortir un résultat (une valeur) qui représente le mieux la similarité (ou distance) entre les différentes séquences (Chatzou *et al.*, 2016). En effet, en biologie moléculaire, un terme qui revient souvent est la distance d'évolution. Cette distance est souvent égale à 1 – similarité, et vice-versa, donc la conversion ne change

pas la définition d'un RSS. Parmi ces outils, BLAST est fortement populaire (Altschul *et al.*, 1990), mais il existe plusieurs alternatives, avec des paramètres réglables pour adapter différents contextes. La Table 1.1 montre un exemple de matrice de distance, qui compare chaque paire de nœuds et en présente une distance.

Table 1.1 : Exemple de matrice de distance entre des séquences de gènes. Le nombre sur la première ligne indique le nombre d'entrées dans la matrice. La matrice est carrée, les colonnes représentent les séquences dans le même ordre que les rangées.

6						
90962917_Lactobacillus_pMP118	0	2.075	2.092	2.159	4.579	5.608
183219418_Lactobacillus_pLR581	2.075	0	0.450	0.425	4.704	4.939
187729652_Lactococcus_pKL0018	2.092	0.450	0	0.265	4.300	5.244
32455514_Lactobacillus_pMD5057	2.159	0.425	0.265	0	4.612	4.776
190015737_Clostridium_pCW3	4.579	4.704	4.300	4.612	0	3.741
62945228_Mannheimia_pCCK3259	5.608	4.939	5.244	4.776	3.741	0

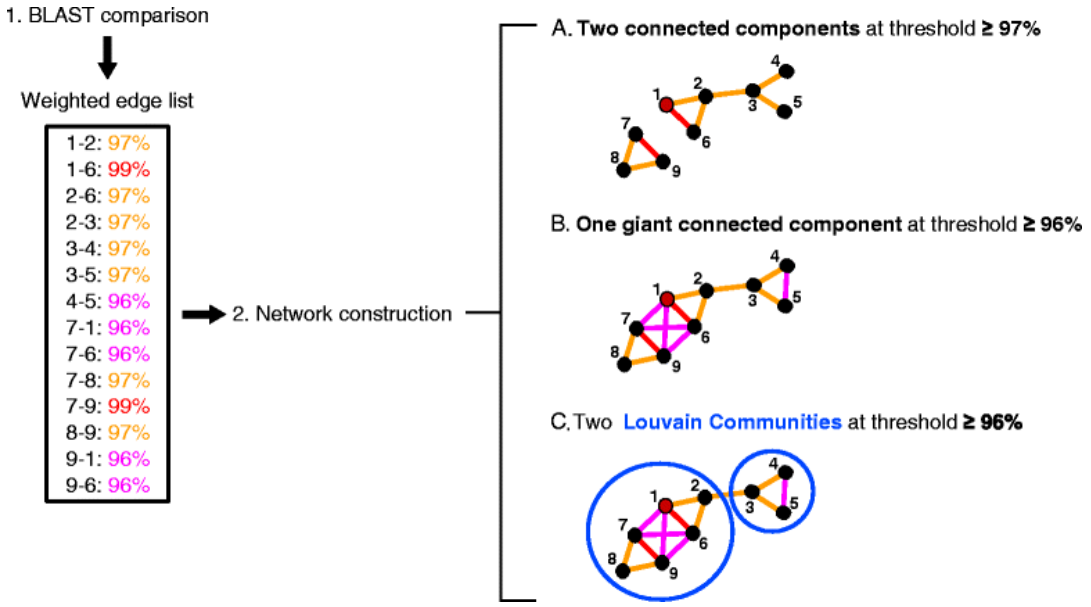


Figure 1.4: Création d'un RSS à partir de BLAST (Forster *et al.*, 2015). Le seuil de similarité détermine la connectivité du réseau final.

Tant que nous avons une matrice de distance, il est possible de créer notre RSS. À l'instar des arbres phylogénétiques, il est possible de passer par un modèle de substitutions (Jukes-Cantor, JTT, Kimura) pour créer une matrice de distance. Pour cette même raison, il est possible de transformer un arbre phylogénétique en un RSS (Atkinson *et al.*, 2009). La Figure 1.4 montre un exemple de RSS crée avec la sortie de BLAST, qui est à la base une matrice de distance.

En résumé, les nœuds du réseau représentent les espèces par une séquence de gène, protéine, parfois le génome entier, alors que les longueurs des arêtes sont des valeurs obtenues par un calcul de similarité effectué sur les séquences en question. Dans la majorité des cas, pour alléger les informations du réseau, les séquences ne sont pas stockées dans les nœuds; le calcul des similarités est terminé et les arêtes sont définies. À la place, il s'agit d'un nom ou d'un code qui permet d'identifier de manière unique la séquence qui est représentée par le nœud. Si nous utilisons les figures précédentes

pour appliquer les définitions, la Figure 1.1 serait donc le produit de ces étapes de calcul, avec les arêtes définies et les nœuds simplifiés en numéro. Le Cas C de la Figure 1.2 serait obtenu avec un seuil de similarité plus restrictif, ce qui diminue le nombre d'arêtes dans le réseau.

1.3 Plus court chemin

Pour nous, il ne s'agit pas de calculer la distance entre les nœuds, mais plutôt la distance entre les communautés auxquelles ces nœuds appartiennent. Pour cela, nous utilisons le concept des plus courts chemins, qui est déjà bien définis avec les arbres phylogénétiques (Yang et Rannala, 2012). Il est possible d'approximer la distance d'évolution entre des taxons avec le plus court chemin dans un arbre qui représente leur relation évolutive, ce que nous voulons appliquer dans les RSS. Un plus court chemin est le chemin pour se rendre d'un nœud *i* vers un nœud *j*. Le plus court chemin est toujours unique dans un arbre phylogénétique, mais pas dans un réseau phylogénétique ou un RSS (Semple et Steel, 2003). Dans un arbre phylogénétique, sans les longueurs d'arêtes, il faut naviguer vers l'ancêtre commun le plus proche des deux nœuds à partir d'un nœud de départ, puis compter le nombre de nœuds à partir de cet ancêtre commun vers le nœud d'arrivée, tel qu'illustré dans la Figure 1.5.

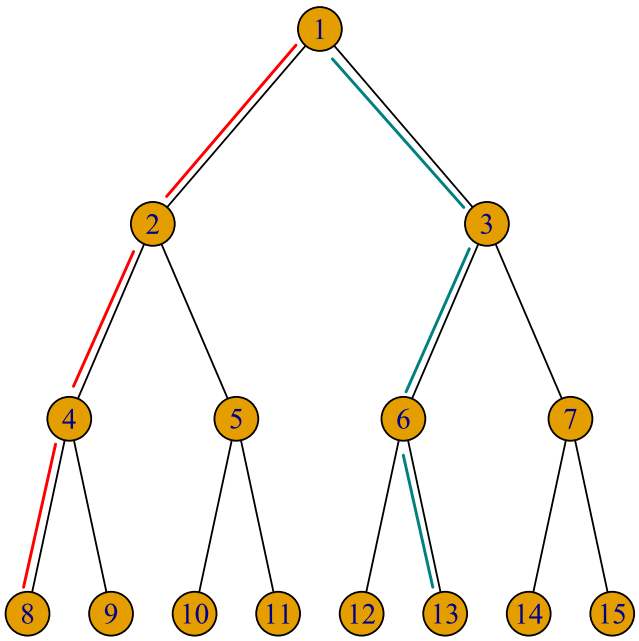


Figure 1.5 : Plus court chemin dans un arbre. Nous partons du nœud 8 vers le nœud 13 en passant par la racine, qui est le plus proche commun ancêtre des deux nœuds.

Mais un arbre est un graphe acyclique et dirigé, ce qui n'est pas le cas des RSS. Il est possible de retrouver de multiples plus courts chemins entre deux nœuds dans un RSS. Pour le calcul des plus courts chemins dans les réseaux, un algorithme connu est l'algorithme de Djikstra (Dijkstra, 1959), qui a été depuis optimisé (Deng *et al.*, 2012). Il existe plusieurs algorithmes différents qui permettent de calculer des plus courts chemins (Zhan, 1997), mais nous utiliserons seulement l'algorithme de Dijkstra au cours de ce travail. Nous rechercherons le ou les plus courts chemin(s) qui existent entre une paire de nœuds *ij* dans un RSS, représentés par tous les nœuds qu'il faut traverser pour se rendre du nœud *i* vers le nœud *j*.

CHAPITRE II

MÉTHODES

Dans ce chapitre, nous donnons les définitions de nouvelles distances et indice entre les communautés d'espèces présentes dans un RSS. La première de ces distances, *NetUnifrac*, est une extension directe de la distance *Unifrac* traditionnelle (Lozupone et *al.*, 2011), pour les RSS. Quatre autres distances, *Spp*, *Spep*, *Spelp* et *Spinp*, sont des distances de communauté symétriques basées sur le calcul des plus courts chemins. Nous proposons aussi l'indice de *Transfert*, qui est une mesure directionnel asymétrique pour estimer la direction et le taux de transfert horizontal de matériel génétique entre des communautés d'espèces différentes d'un RSS. Dans les arbres phylogénétiques (e.g., les feuilles externes des arbres qui sont généralement annotées avec un taxon, et des feuilles internes qui sont leurs ancêtres), il existe un chemin évolutif unique entre deux nœuds donnés, souvent utilisé pour estimer la distance d'évolution entre eux. Les RSS montrent plus de flexibilité quant au nombre de chemins évolutifs possibles. C'est pourquoi nous utilisons les plus courts chemins multiples pour définir nos distances. Dans les deux cas, des arbres ainsi que des RSS, la similarité/dissimilarité des séquences de gène ou génome des espèces étudiées est utilisée pour construire la structure. Bien que le concept des plus courts chemins dans un RSS n'ait pas tout à fait la même explication évolutive les chemins dans un arbre phylogénétique, il peut être exploité pour découvrir différents types de relation entre les communautés d'espèces en utilisant les nouvelles mesures décrites dans ce travail.

Nous avons aussi inclus dans notre étude la distance de *Motifs* présenté par Bapteste *et al.* (2012), même s'il ne s'agit pas d'une nouvelle métrique en soi.

2.1 Définitions des nouvelles distances et de l'indice de *Transfert*

Nous proposons quelques nouvelles distances entre différentes communautés d'espèces. Il est important de noter que toutes ces distances sont des comparaisons par paire, et donc lorsqu'il y a plus de deux communautés, les distances sont calculées sur des sous-graphes qui regroupent seulement les nœuds associés à deux communautés qui sont comparés. L'idée générale derrière les plus courts chemins est comme suit : si une espèce de la communauté *X* est située sur le plus court chemin de deux espèces de la communauté Y, cela signifie que l'évolution de ces deux espèces ou de leur ancêtre a été affectée par le transfert de matériel génétique de la communauté X.

Une arête est dite *monochrome* si ses deux extrémités appartiennent à la même communauté. Similairement, un chemin est *monochrome* si toutes les arêtes sont monochromes. Les arbres phylogénétiques et les RSS gardent une certaine ressemblance puisqu'ils traitent des mêmes objets, soit des séquences homologues, et puisqu'ils sont tous les deux des graphes. Cependant, les arbres phylogénétiques sont utilisés pour représenter l'évolution des espèces, alors que les RSS, tel que le nom le suggère, représentent plutôt la similarité/dissimilarité entre ces mêmes espèces. À l'opposé des arbres, les RSS sont des graphes non-enracinés qui peuvent aussi être déconnectés. Transformer un arbre en RSS est simple, mais le contraire n'est pas évident (Atkinson *et al.*, 2009).

Plus bas, nous présenterons de nouvelles distances et indices entre les communautés d'espèces, sachant que plus d'un plus court chemin peut exister entre une paire d'espèce (*i*,*j*) dans un RSS. Une communauté consiste en un sous-graphe du réseau dont les nœuds appartiennent à une catégorie donnée au préalable, par exemple un

groupe relié phylogénétiquement, un habitat géographique ou n'importe quelle autre catégorie à étudier. L'idée générale derrière nos mesures basées sur les plus courts chemins des RSS, définies par les distances *Spp*, *Spep*, *Spelp*, *Spinp* et l'indice de *Transfert* est que si un nœud représentant une espèce d'une communauté *Y* est localisé sur le ou un des plus courts chemins entre deux nœuds de la communauté *X*, il est possible que :

1. Dans un RSS de gènes : l'évolution du gène d'un ou des deux nœuds de la communauté *X*, ou un de leurs ancêtres, a été affectée par un THG provenant de la communauté *Y*, ou d'un ancêtre proche de *Y* (dans ce cas, le regroupement du réseau à laquelle appartient l'espèce de *Y* est habituellement plus large que l'une des deux composantes connexes de *X*). Cela correspond au cas traditionnel de THG qui assume que le gène transféré supplante le gène orthologue ou est rajouté lorsqu'il était inexistant dans cet organisme;
2. Dans un RSS de gènes : La séquence du gène de l'espèce de *Y*, ou un des ancêtres proches, est un gène mosaïque qui a été créé à partir de la recombinaison intragénique des gènes des deux espèces *X*, ou de leurs ancêtres proches (dans ce cas, le regroupement à laquelle appartient l'espèce de *Y* est habituellement très petit, voire constitué de cet unique nœud);
3. Dans un RSS de génome : Cette espèce de *Y*, ou un ancêtre proche, est un hybride des deux espèces de *X*, ou d'ancêtres proches (dans ce cas, le regroupement à laquelle appartient cette espèce de *Y* est habituellement très petit, voire constitué de cet unique nœud).

La distance *NetUnifrac* pour un RSS est un analogue de la distance *Unifrac* classique (Lozupone et *al*., 2011) définie pour les arbres phylogénétiques :

$$NetUniFrac = \frac{\text{Longueur totale des arêtes monochromes du RSS}}{\text{Longueur totale de toutes les arêtes du RSS}} \quad (1)$$

Un des avantages de cette distance est qu'elle peut être calculée dans un temps linéaire en fonction du nombre d'arêtes dans le réseau. Il est important de mentionner que *NetUniFrac* peut être calculé en tant que proportion pour tout le réseau ou pour une paire de communautés présentes dans le réseau (comme la distance *UniFrac* classique). Cette distance reflète l'homogénéité du réseau ou d'une paire de communautés. La distance *NetUniFrac* peut aussi être vue comme une variante simplifiée de l'assortativité (Newman, 2003), qui représente aussi une fraction des arêtes qui connectent les nœuds de la même communauté. Pour un RSS non-dirigé :

$$Assortativity = (f_M - \sum_{C_i} f_{NM}^2(C_i))/(1 - \sum_{C_i} f_{NM}^2(C_i)),$$

où f_M est la fraction des arêtes monochromes dans le réseau et $f_{NM}(C_i)$ est la fraction des arêtes non-monochromatiques dans le réseau qui connecte un nœud appartenant à la communauté C_i à un nœud d'une communauté différente. Par exemple, quand le nombre d'arêtes non-monochromes tend vers 0, l'assortativité et *NetUniFrac* tendent vers 1.

Quand les longueurs d'arêtes sont égales (réseau non-pondéré), *NetUniFrac* devient équivalent à la distance de *Motifs* avec les motifs de longueurs 2 (i.e., les arêtes) définie dans Bapteste *et al*. (2012). La distance de *Motifs* est calculée comme une proportion d'une structure de sous-graphes spécifiques qui appartiennent à une communauté d'espèces donnée :

$$Motif = \frac{\text{Nombre de motifs spécifiques appartenant à une communauté}}{\text{Nombre total de motifs spécifique de même taille}} \quad (2)$$

Habituellement, la taille du motif est limitée à 2 ou 3 dû au temps de calcul demandé pour identifier tous les motifs de plus grande taille.

La distance *Spp* : en anglais *Shortest path proportion* est la proportion des plus court chemins monochromes pour chaque paire de communautés du réseau :

$$Spp(X,Y) = \frac{\sum_{i,j \in X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k} + \sum_{i,j \in Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}}{\frac{1}{2}(N_X(N_X - 1) + N_Y(N_Y - 1))}, \quad (3)$$

où N_X est le nombre de nœuds (représentant des espèces) appartenant à la communauté X, N_Y est le nombre de nœuds appartenant à la communauté Y, $K(ij)$ est le nombre de plus courts chemins entre les nœuds i et j appartenant à la même communauté. La variable binaire $\sigma_{ij}^{k} = 1$, si le $k^{\text{ème}}$ plus court chemin entre the nœuds i et j est monochrome; sinon, $\sigma_{ij}^{k} = 0$. La quantité $\frac{1}{2}(N_X(N_X - 1) + N_Y(N_Y - 1))$, qui apparaît dans le dénominateur est le nombre maximum de plus court chemins monochromes uniques dans un RSS avec les communautés X et Y. L'utilisation de σ_{ij}^{k} dans le numérateur nous permet de mesurer la contribution du plus court chemin k entre les nœuds i et j au compte total des plus courts chemins dans le réseau.

Dans un RSS de transferts, cette distance peut être utilisée pour estimer la proportion des connections évolutives entre des communautés d'espèces qui sont dues à des THG, alors que dans un RSS d'hybridation, cette distance peut être utilisée pour identifier des hybrides potentiels et leurs parents.

La distance *Spep* : en anglais *Shortest path edge proportion*, est basée sur le nombre d'arêtes monochromes dans tous les plus courts chemins entre les nœuds de la même communauté:

$$Spep(X,Y) = \frac{\sum_{i,j \in X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k} + \sum_{i,j \in Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k}}{\frac{1}{2}(N_X(N_X - 1) + N_Y(N_Y - 1))}, \quad (4)$$

où E_{ij}^{k} est la proportion d'arêtes monochromes dans le $k^{\text{ème}}$ plus court chemin entre les nœuds *i* et *j* (appartenant à la même communauté d'espèces). Que ce soit dans les RSS de transferts ou d'hybridation, quand le nombre de receveurs du matériel génétique est connu, cette distance peut être utilisée pour estimer les espèces receveurs qui ont été affectées par un transfert récent, tel que montré dans la Figure 1b et d, quand tous les receveurs de transferts sont bien séparés les uns des autres (les nœuds bleus à la droite de la Figure 1b), ou par des transferts plus anciens, lorsque les receveurs forment un grand regroupement.

La distance *Spelp* : en anglais *Shortest path edge length proportion*, est basée sur la longueur totale des arêtes monochromes dans tous les plus courts chemins entre les nœuds de la même communauté:

$$Spelp(X,Y) = \frac{\sum_{i,j \in X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k} + \sum_{i,j \in Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k}}{\frac{1}{2}(N_X(N_X-1) + N_Y(N_Y-1))}, \quad (5)$$

où EL_{ij}^{k} est la proportion de la longueur des arêtes monochromes dans le $k^{\text{ème}}$ plus court chemin entre les nœuds *i* et *j*. Cette distance est très proche de *Spelp*. La différence principale réside dans le poids des arêtes dans les plus courts chemins, ce qui donnent plus d'importance lorsque le temps d'évolution entre deux nœuds est plus long.

La distance *Spinp* : en anglais *Shortest path internal nœuds proportion*, est calculée selon la proportion de nœuds internes d'un plus court chemin, qui appartiennent à la même communauté d'espèces que ses extrémités :

$$Spinp(X,Y) = \frac{\sum_{i,j \in X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k} + \sum_{i,j \in Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k}}{\frac{1}{2}(N_X(N_X - 1) + N_Y(N_Y - 1))}, \quad (6)$$

où N_{ij}^{k} est la proportion de nœuds internes de la même communauté, à laquelle les nœuds *i* et *j* appartiennent, dans le $k^{\text{ème}}$ plus court chemin entre *i* et *j*. Cette distance peut être sensible aux poids des nœuds du réseau (si présents), et donc considérer l'abondance ou la pertinence statistique des espèces associées.

L'indice de *Transfert* : Aussi basé sur le calcul des plus courts chemins, cette mesure est cependant asymétrique et directionnelle. Elle peut être utilisée pour calculer la proportion d'espèces de la communauté *X* qui sont affectées par des THG provenant d'espèces/séquences de la communauté *Y* lorsque nous considérons un RSS de transferts, ou la dispersion physique des espèces entre les communautés lorsque nous considérons un RSS en écologie. L'interprétation de l'indice est flexible, mais nous l'avons tout d'abord créé pour des RSS de transferts. La variable $p_i(\alpha)$ est égale à 0 si le nœud *i* de *X* a reçu du matériel génétique d'un nœud de *Y*, et est égale à 1 sinon:

$$p_i(\alpha(Y,X)) = \begin{cases} 1, \text{ if } \sum_{j \in X(j \neq i)} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k} \geq \alpha(N_X - 1) > 0, \\ 0, \text{ sinon,} \end{cases}, \quad (7)$$

où la variable binaire $\sigma_{ij}^{k} = 1$, si le $k^{\text{ème}}$ plus court chemin entre *i* et *j* est monochrome, et $\sigma_{ij}^{k} = 0$, sinon, $K(ij)$ $(0 \leq \alpha \leq 1)$ est le nombre de plus courts chemins entre les nœuds *i* et *j*, et $\alpha(Y,X)$ est le seuil choisi selon la proportion de plus courts chemins monochromes pour décider si un nœud de *X* a été affecté par un transfert de *Y* ou non.

Ensuite, l'indice de *Transfert* de la communauté *Y* à la communauté *X*, selon le seuil *α*, peut être défini comme suit:

$$Tr(Y \to X, \alpha(Y,X)) = \frac{\sum_{i \in X} p_i(\alpha(Y,X))}{N_X}, \quad (8)$$

L'indice de *Transfert* de *X* à *Y*, $T_d(X \to Y, \alpha)$, est définie de la même manière. Bien sûr, $T_d(Y \to X, \alpha)$ n'est pas nécessairement égale à $T_d(X \to Y, \alpha)$. La question ici est de sélectionner une valeur appropriée pour le paramètre $\alpha(Y,X)$. Si nous considérons la suite ordonnée de la manière suivante : $\frac{1}{N_X - 1} \sum_{j \in X (j \neq i)} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^k$, i =1, …, N_X, qui est calculée pour chaque nœud $i \in X$, où i =1, …, N_X, un choix intuitif serait de sélectionner la valeur de $\alpha(Y,X)$ comme le milieu du plus grand intervalle entre deux valeurs de la suite. Étant donné les résultats de simulations que nous montrerons plus tard (voir Chapitre III), cette hypothèse semble être efficace pour choisir le seuil $\alpha(Y,X)$. La valeur de $\alpha(Y,X)$ doit changer selon les paires de communautés d'espèces analysées, (*X*, *Y*), et de la direction du transfert.

La Figure 2.1 présente un exemple de calcul de l'indice de *Transfert*. Le RSS montré inclut *n* espèces de la communauté bleue (*B*) et *n* espèces de la communauté jaune (*Y*). La valeur de *α* pour les cas représentés sur la Figure 2.1 est choisie comme expliqué plus haut. Dans le cas A, les composantes des espèces bleues et jaunes sont reliées par une arête unique, sans THG. Donc, $T_d(B \to Y, \alpha = 0.5) = T_d(B \to Y, \alpha = 0.5) = 1$. Dans le cas B, une espèce de la communauté bleue (*B*) est affectée par le transfert de la communauté jaune (*Y*) et les valeurs de l'indice de *Transfert* pour les communautés jaune et bleue sont respectivement: $T_d(B \to Y, \alpha = 0.5) = 1$ et

$T_d(Y \to B, \alpha = \frac{n-2}{2(n-1)}) = 1/n$. Quand la moitié des espèces de *B* sont affectées par des transferts de *Y*, les valeurs de l'indice seront comme suit: $T_d(B \to Y, \alpha = 0.5) = 1$ et $T_d(Y \to B, \alpha = \frac{n-2}{4(n-1)}) = 1/2$. Quand toutes les espèces de *B* sont affectées par des transferts de *Y*, les valeurs de l'indice seront comme suit: $T_d(B \to Y, \alpha = 0.5) = 1$ et $T_d(Y \to B, \alpha = 0) = 0$.

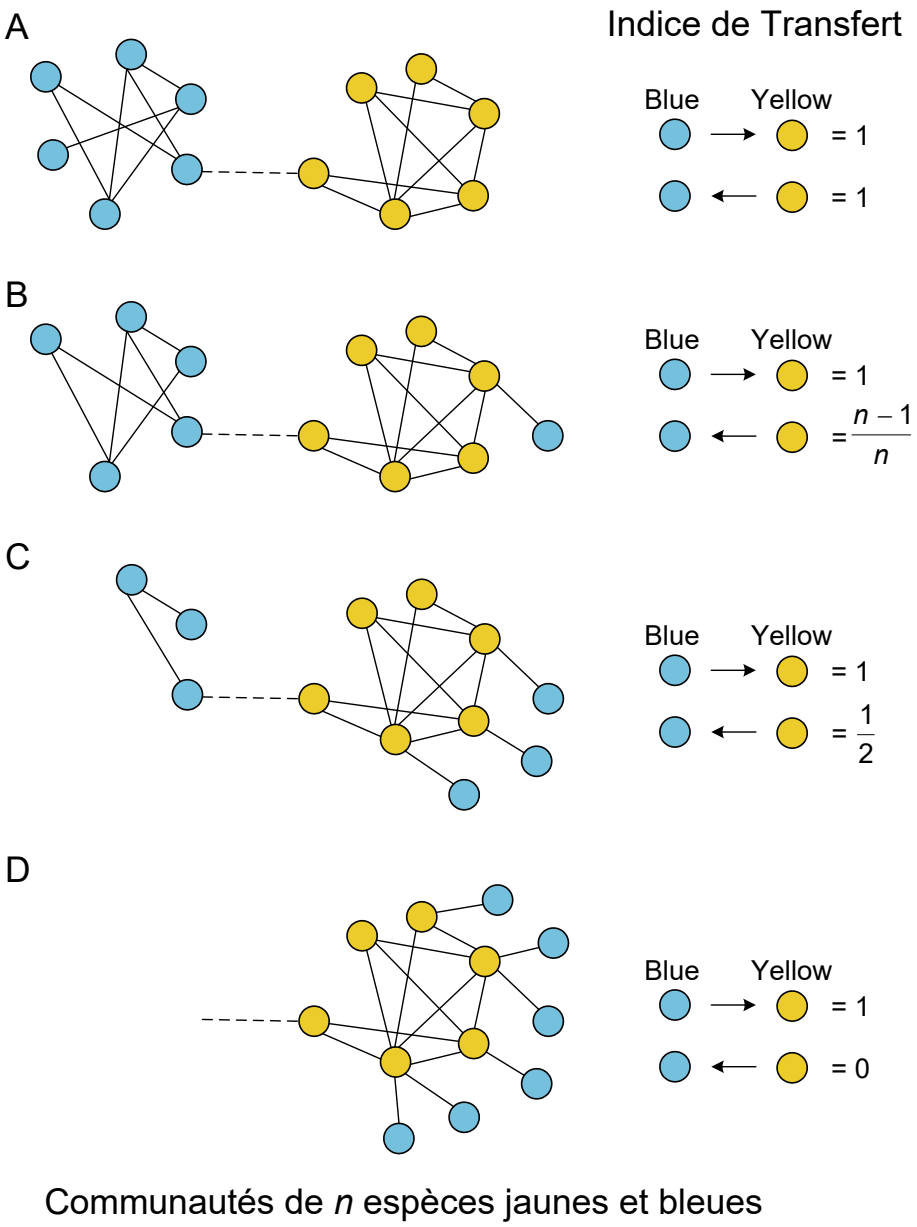


Figure 2.1 : Exemple de calcul de l'indice de *Transfert* pour un réseau de similarité de séquences contenant *N* espèces de la communauté bleue et *N* espèces de la communauté jaune. Cas A: les communautés bleue et jaune sont complètement séparées (e.g., les sous-graphes sont connectés par une seule arête, représentée par la ligne pointillée, ou ne sont pas connectés du tout); Cas B: une espèce de la communauté bleue est affectée par un transfert de gènes d'une espèce de la communauté jaune; Cas C: *n*/2 espèces de la communauté bleue sont affectées par des transferts de gènes de espèces de la communauté jaune. Cas D: toutes les *N* espèces de la communauté bleue sont affectées par des transferts de gènes des espèces de la communauté jaune.

L'indice de Transfert – une version alternative: Une version alternative de l'indice de *Transfert* peut être définie comme suit:

$$Tr_{alt}(Y \to X) = \frac{\sum_{i,j \in X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}}{\frac{1}{2}(N_X(N_X - 1))}, \tag{9}$$

où la variable binaire $\sigma_{ij}^{k} = 1$ si le $k^{\text{ème}}$ plus court chemin entre *i* et *j* est monochrome, et $\sigma_{ij}^{k} = 0$, sinon. Un avantage de cet indice est qu'il n'y a pas de paramètres de seuil à choisir (i.e., il ne dépend pas de α). Toutefois, cet indice ne peut être directement utilisé pour estimer le nombre/proportion d'espèces individuelles d'une communauté d'espèces affecté par les THG provenant d'une autre communauté.

À partir de l'indice de *Transfert* alternatif, nous pouvons déterminer le nombre de transferts et le pourcentage sur le nombre total de nœuds. À partir de l'Équation 9, nous pouvons reformuler la distance comme étant un ratio entre les plus courts chemins monochromes dans le réseau et le nombre maximal de plus courts chemins possibles.

Cette formule peut être écrite comme suit:

$$Tr_{alt} = 1 - \frac{k(N_X - 1) - \frac{k(k-1)}{2}}{\frac{1}{2}(N_X(N_X - 1))}, \tag{10}$$

où *k* est le nombre de transferts. Le numérateur ici représente le nombre de chemins monochromes retrouvés dans le réseau.

Lorsque *k* est égale à 0 (aucun transfert), le ratio est 0 et la distance est égale à 1. Lorsque *k* est égale à N_X (tous les nœuds), le ratio est 1 et la distance devient 0. Avec la

distance calculée en utilisant l'Équation 9, le nombre de transfert *k* peut facilement être inféré en insérant la distance dans l'Équation 10. En isolant *k*, l'Équation 10 devient alors une équation quadratique:

$$Tr_{alt} - 1 = -\frac{k(N_X - 1) - \frac{k(k-1)}{2}}{\frac{1}{2}(N_X(N_X - 1))},$$

$$(Tr_{alt} - 1)(\frac{1}{2}(N_X(N_X - 1))) = -(k(N_X - 1) - \frac{k(k-1)}{2}),$$

$$\frac{Tr_{alt} \cdot {N_X}^2}{2} - \frac{{N_X}^2}{2} - \frac{Tr_{alt} \cdot N_X}{2} + \frac{N_X}{2} = -k \cdot N_X + k + \frac{k^2}{2} - \frac{k}{2},$$

$$k \cdot N_X - \frac{k}{2} - \frac{k^2}{2} + \frac{Tr_{alt} \cdot {N_X}^2}{2} - \frac{{N_X}^2}{2} - \frac{Tr_{alt} \cdot N_X}{2} + \frac{N_X}{2} = 0,$$

À l'aide de la formule quadratique : $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$,

nous avons : $a = 1$, $b = 1 - 2N_X$ et $c = {N_X}^2 - N_X - Tr_{alt} \cdot {N_X}^2 + Tr_{alt} \cdot N_X$

$$k_{1,2} = \frac{-1 + 2N_X \pm \sqrt{(1 - 2N_X)^2 - 4({N_X}^2 - N_X - Tr_{alt} \cdot {N_X}^2 + Tr_{alt} \cdot N_X)}}{2},$$

$$k_{1,2} = \frac{-1 + 2N_X \pm \sqrt{4Tr_{alt} \cdot {N_X}^2 - 4Tr_{alt} \cdot N_X + 1}}{2}, \qquad (11)$$

où Tr_{alt} est la valeur de l'indice de *Transfert* alternatif.

Pour comparer le nombre de transferts, la valeur de *k* est normalisée en la divisant par le nombre total de nœuds dans cette communauté. Cette formule considère seulement les THG où un gène a été transféré à la fois.

Les mesures définies dans les Équations (1) et (3-9) assument que n'importe quelle communauté d'espèces est composée d'au moins deux espèces. Si une communauté est représentée par une seule espèce, sa relation aux autres communautés est regardée comme inconnue. Les valeurs de toutes ces distances et indices varient entre 0 et 1. Pour rendre la valeur de $d(X, X)$ égale à 0 (ici d remplace n'importe quelle distance ou indice défini ci-haut), qui est une condition nécessaire pour une distance, nous pourrions changer le critère monochromatique par un critère bichromatique lorsque nous travaillons avec des réseaux qui contiennent des nœuds qui appartiennent à plusieurs communautés à la fois. Par exemple, si le nœud *i* correspond à une espèce qui appartient à deux communautés, alors toutes les arêtes (i,j) sont bichromatiques, peu importe le nœud adjacent *j*.

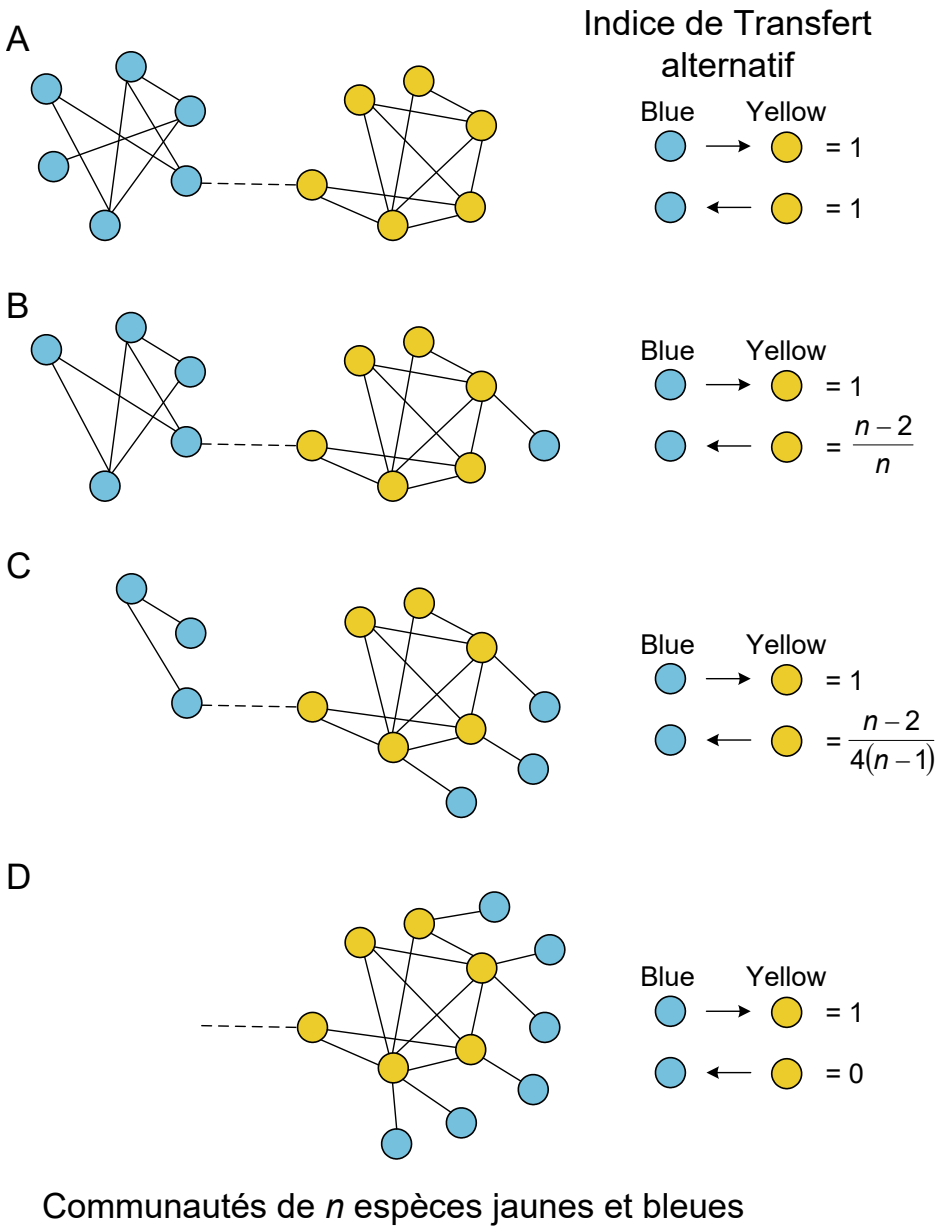


Figure 2.2: Exemple de calcul de l'indice de *Transfert alternatif* pour un réseau de similarité de séquences contenant *N* espèces de la communauté bleue et *N* espèces de la communauté jaune. Cas A: les communautés bleue et jaune sont complètement séparées (e.g., les sous-graphes sont connectés par une seule arête, représentée par la ligne pointillée, ou ne sont pas connectés du tout); Cas B: une espèce de la communauté bleue est affectée par un transfert de gènes d'une espèce de la communauté jaune; Cas C: *n*/2 espèces de la communauté bleue sont affectées par des transferts de gènes de espèces de la communauté jaune. Cas D: toutes les *N* espèces de la communauté bleue sont affectées par des transferts de gènes des espèces de la communauté jaune

2.2 Exemple des calculs de distances

Dans cette section, nous présentons un exemple de RSS simple pour illustrer le calcul des nouvelles distances définies dans les Équations (1) et (3-9). Le RSS présenté dans la Figure 2.3 est composé des nœuds de deux communautés (2 espèces jaunes et 3 espèces bleues). Il y a deux plus courts chemins, un qui est monochrome et un non-monochrome (passant par l'espèce 4), entre les espèces 2 et 3 appartenant à la communauté bleue. Tous les trois plus courts chemins entre les espèces de la communauté jaune sont uniques, mais deux d'entre eux (entre 1 et 4, et 1 et 5) sont non-monochromes. La Table 2.1 montre les résultats détaillés des distances pour le RSS de la Figure 2.3.

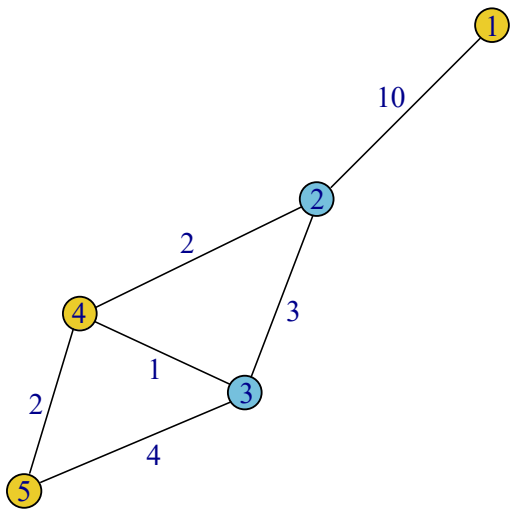


Figure 2.3 : RSS avec cinq nœuds (deux nœuds bleus et trois nœuds jaunes, représentant deux communautés d'espèces différentes) et six arêtes. Les longueurs d'arêtes sont indiquées à côté de chaque arête.

Table 2.1 : Numérateur, dénominateur et résultat de chaque nouvelle distance et nouvel indice pour le RSS présenté dans la Fig. 2.3.

Distance	*Numérateur*	*Dénominateur*	*Résultat*
NetUniFrac	2+3	1+2+2+3+4+10	0.23
Spp	0+0+0.5×(1+0)+1	0.5×(2+6)	0.375
Spep	0+0.33+0.5×(1+0)+1	0.5×(2+6)	0.46
Spelp	0+0.14+0.5×(1+0)+1	0.5×(2+6)	0.41
Spinp	0+0.5+0.5×(1+0)+1	0.5×(2+6)	0.50
$Tr(B \rightarrow Y, \alpha=0.25)$	0+1+1	3	0.67
$Tr(Y \rightarrow B, \alpha=0.5)$	2	2	1.0
$Tr_{alt}(B \rightarrow Y)$	0+0+1	0.5×6	0.33
$Tr_{alt}(Y \rightarrow B)$	0.5(1+0)	0.5×2	0.5

Les valeurs des distances de la Table 2.1 varient de 0.23 pour la distance *NetUniFrac* jusqu'à 0.5 pour la distance *Spinp* (nœuds internes), et les valeurs des indices varient de 0.33 pour l'indice de *Transfert* alternative jusqu'à 1.0 pour l'indice de *Transfert*. Les valeurs proches de 0 signifient que les communautés considérées interagissent beaucoup entres elles, alors qu'une valeur plus élevée (proche de 1) signifie que les communautés sont plus isolées, avec une basse fréquence d'interaction génétique.

2.3 Exemple de calculs des distances et indices sur des arbres phylogénétiques et RSS simples

Dans la Figure 2.4, 5 petits réseaux sont illustrés, avec toutes les informations nécessaires pour calculer les distances. Dans chaque cas, les nœuds sont partagés entre deux communautés arbitraires, soit les communautés jaune et bleu. Dans les graphes 1 et 2, les arêtes ont toutes le même poids, alors que dans les graphes 3, 4, 5, un poids différent leur a été associé.

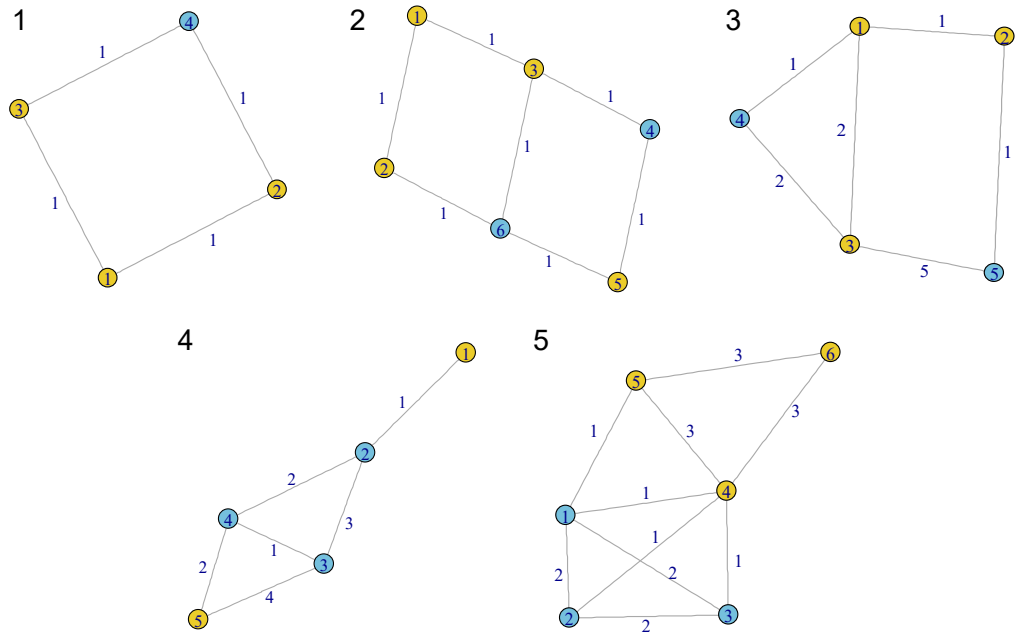


Figure 2.4 : Cinq RSS simples contenant des espèces des communautés jaune (Y) et bleu (B). Le nombre à côté des arêtes sont leur poids, et les nœuds sont numérotés.

Table 2.2 : Valeurs des distances et indices obtenus pour les RSS représentés dans la Figure 2.4.

Mesure/ RSS	*NetUnifrac*	*Motif_3*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer* ($Y \to B$)	*Transfer* ($B \to Y$)	*Alt. Trans* ($Y \to B$)	*Alt. Trans* ($B \to Y$)
1	0.5	0.25	0.75	0.667	0.667	0.25	0	1	0	0.833
2	0.286	0.22	0.25	0.28	0.28	0.6	0	0.75	0	0.5
3	0.33	0.1	0.75	0.571	0.667	0.4	0	1	0	1
4	0.5	0.2	0.8	0.625	0.643	0.33	1	0	1	0
5	0.6	0.4	0.556	0.385	0.6	0.444	0.5	0.66	0.5	0.66

Il est important de noter que toutes les distances et indices définis par les RSS peuvent aussi être calculés pour les arbres phylogénétiques. Ceci nous donne des nouvelles mesures pour estimer la distance entre les communautés d'espèces dans un arbre, par rapport à la distance *UniFrac* classique. Un chemin entre deux feuilles dans un arbre peut aussi être monochrome ou non, dépendamment du sous-arbre représentant le plus court chemin unique entre deux feuilles. La Figure 2.5 contient quelques exemples d'arbres à deux communautés, et la Table 2.3 présente les distances et indices calculés pour ces arbres.

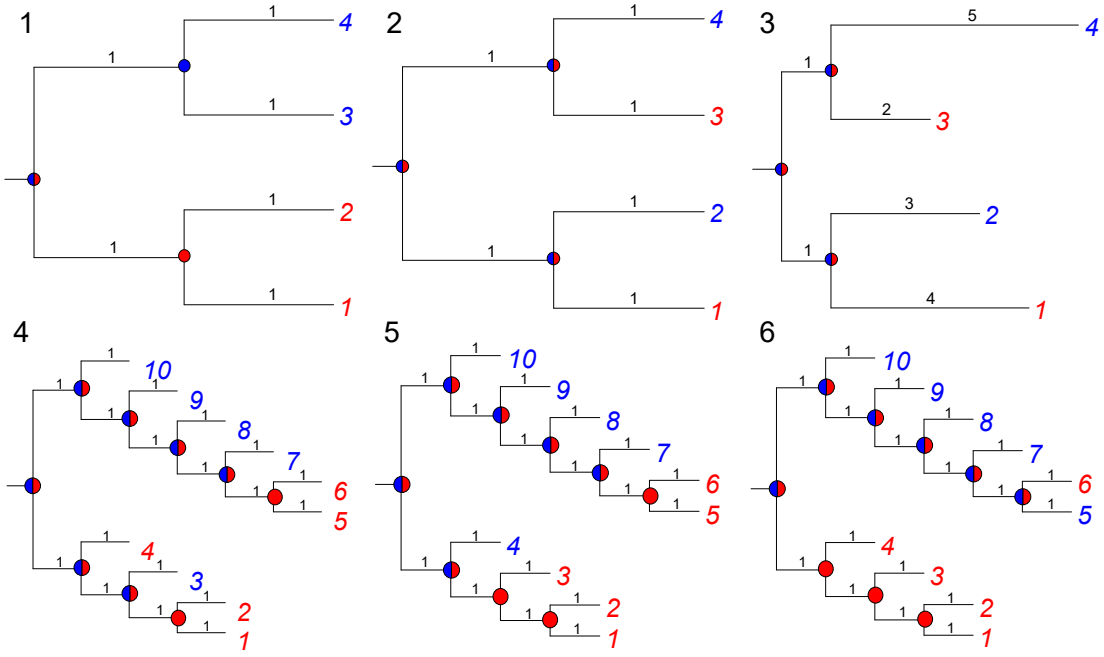


Figure 2.5 : Six arbres phylogénétiques enracinés utilisés comme exemples pour les calculs de distances et indices définis dans ce mémoire. Les poids des arêtes sont indiqués en noir. Les espèces appartenant aux communautés d'espèces bleues (B) et rouges (R) sont initialement associées aux feuilles des arbres. Par la suite, chaque nœud interne est coloré soit en en bleu, en rouge, ou en bleu et rouge, dépendamment des feuilles qui descendent de ce nœud. Nous remontons ainsi jusqu'au premier branchement en bas de la racine.

Table 2.3 : Valeurs des distances et indices obtenues pour les arbres de la Figure 2.5.

Mesure/ Arbre	*UniFrac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer* $(R \to B)$	*Transfer* $(B \to R)$	*Alt Trans* $(R \to B)$	*Alt Trans* $(B \to R)$
1	1	1	1	1	1	1	1	1	1
2	0.667	0	0	0	0	0.5	0.5	0	0
3	0.875	0	0	0	0	0.5	0.5	0	0
4	0.667	0.1	0.138	0.138	0.1	0.2	0.2	0	0.2
5	0.72	0.2	0.232	0.232	0.2	0	0.4	0	0.4
6	0.72	0.3	0.292	0.292	0.3	0.2	0.2	0	0.6

2.4 Algorithme pour un calcul rapide de l'indice de *Transfert*

Dans cette section, nous présentons un algorithme qui peut être utilisé pour calculer l'indice de *Transfert* dans les situations pratiques avec un grand RSS contenant plusieurs communautés d'espèces. Nous le montrons ici dans le contexte de l'indice de *Transfert*, mais cet algorithme peut facilement être adapté pour calculer toutes les distances introduites dans ce mémoire.

Les Équations (3) à (9) assument que le réseau est un graphe connexe dans lequel il y a au moins un plus court chemin entre chaque paire de nœuds d'une même communauté. Si le réseau original est déconnecté alors les différentes composantes connexes devraient être reliées en utilisant, par exemple, un critère de similarité (i.e., deux espèces dans deux composantes connexes ayant la plus grande similarité seront connectées par une nouvelle arête) ou la centralité d'intermédiarité, qui est une mesure de centralité dans un graphe basée sur les plus courts chemins, utilisée en théorie de graphe (i.e., deux espèces des deux composantes connexes ayant les plus grandes valeurs de centralité d'intermédiarité seront connectées par une nouvelle arête). Il est à noter que la complexité du calcul de l'indice de *Transfert* des Équations (7-8) est $O(n \times \sum_{i=1}^{C} C_i^2)$, où n est le nombre total de nœuds dans le réseau, C est le nombre de communautés et C_i est le nombre d'espèces dans la communauté i ($i = 1, \ldots, C$). Même si cette complexité algorithmique est polynomiale, le temps de calcul devient rapidement long pour des grands RSS.

Algorithme 1

Calculer l'indice de Transfert selon la proportion des espèces affectées par des transferts de gènes provenant d'une communauté différente.

Entrée: Le RSS G = (N, E), où N est l'ensemble des nœuds (espèces) et E est l'ensemble des arêtes. G peut être connexe ou déconnecté. Chaque espèce dans N appartient à une certaine communauté X.

Sortie: Matrice des distances par paires de toutes les communautés de G.

Étape 1. Identifier toutes les composantes connexes de chaque communité (i.e., sous-graphes de la plus grande taille contenant seulement les espèces d'une communauté; les espèces des autres communautés sont retirées pour déterminer la taille des sous-graphes).

Étape 2. Estimer l'indice de *Transfert* (de Y à X) pour chaque paire de communautés (X,Y) avec la formule suivante :

$$\hat{t}(Y \to X, \alpha_X) = 1 - \frac{\sum_{X_i \in X} \sigma_i x_i}{N_X}, \qquad (12)$$

où x_i est le nombre d'espèces dans la composante connexe X_i de la communauté X, et σ_i = 1 lorsque les trois conditions suivantes sont satisfaites:

(a) $x_i - 1 \le \alpha_X (N_X - 1)$, où α_X $(0 \le \alpha_X \le 1)$ est le seuil choisi pour décider si une espèce de la communauté X a été affectée par un *transfert* ou non (il est raisonnable d'assumer qu'une très grande composante connexe n'est pas affectée par des transferts; dans notre programme la valeur de α_X est initialisée avec le milieu du plus grand intervalle entre deux valeurs consécutives de $\frac{x_i - 1}{N_X - 1}$ des composantes connexes);

(b) Au moins un nœud de X_i est connecté par une arête à un nœud de la communauté Y dans le réseau original G;

(c) La taille de la composante connexe X_i de X est plus petite que la taille d'au moins une composante connexe Y_j de Y pour décider qu'un nœud de X_i est connecté par une arête dans le réseau original G (i.e., $x_i < y_j$).

Sinon, $\alpha_i = 0.5$ si les conditions (a) et (b) sont satisfaites et la taille de la composante connexe X_i de X est égale à la taille de la plus petite composante connexe Y_j de Y à laquelle elle est connectée par une arête dans le réseau original G (i.e., $x_i = y_j$).

Sinon, $\alpha_i = 0$.

Fin de l'algorithme

L'Algorithme 1 est seulement une heuristique qui propose un moyen d'estimer l'indice de *Transfert*, mais il s'agit d'un moyen très rapide de le faire. En effet, la complexité de l'étape 1 de l'Algorithme 1 est $O(m)$ et celui de l'étapes 2 est $O(nC)$, où m est le nombre d'arêtes, N est le nombre de nœuds, et C est le nombre de communautés d'espèces dans le réseau. Comme la valeur de C est petite, comparée à m et N, le temps d'exécution de l'Algorithme 1 est $O(m)$. De ce fait, il est linéaire par rapport au nombre d'arêtes, laissant cet algorithme applicable à des RSS très grands. De plus, comme les résultats des études de simulations suggèrent (voir la Section 3.1), l'Algorithme 1 est un bon moyen pour trouver les regroupements d'espèces affectés par des THG et identifier les communautés donneuses de gènes d'où les transferts proviennent.

La condition (a) de l'Algorithme 1 peut sembler être trop restrictive. Cependant, si une grande composante connexe est affectée par un THG, cela correspond généralement à un transfert qui est beaucoup plus ancien que les transferts que nous essayons de

découvrir. Ces anciens transferts sont plus difficiles à détecter que les transferts récents car ils sont ombragés par plusieurs événements d'évolution réticulée. Les THG récents sont observables et détectables, du moins en ce qui concerne la phylogénie des procaryotes (Koonin *et al.*, 2001).

L'Équation (12) ne considère les plus courts chemins, mais approxime plutôt l'indice de *Transfert* en se basant sur les composantes connexes, un certain seuil a_X, et le nombre de nœuds dans chaque composante connexe. Cette approximation convient bien au problème d'identification des THG récents dans un RSS, où plusieurs espèces individuelles ou petits regroupements d'espèces receveurs de matériel génétique sont connectés avec de plus grands regroupements donneurs de ce matériel. Certaines restrictions pour l'utilisation de l'Algorithme 1 peuvent aussi s'appliquer. Par exemple, quand une communauté d'espèces est sous-représentée, comparée aux autres communautés d'espèces, et le niveau de connectivité est élevé (i.e., le RSS inclut beaucoup d'arêtes par rapport au nombre de nœuds), il est possible que l'algorithme détecte certains faux positifs, soit des transferts d'une grande vers une petite communauté. De plus, lorsque le seuil de similarité h du RSS est trop grand, le réseau sera plutôt constitué de plusieurs regroupements déconnectés, ce qui mène à une surestimation du nombre de transferts, ou d'une mauvaise identification des donneurs et receveurs. L'Équation (12) dépend certainement des paramètres qui ont mené à la création du RSS qui est analysé.

CHAPITRE III

UNE ÉTUDE DE SIMULATIONS ET APPLICATIONS DES NOUVELLES DISTANCES ET INDICES À DES DONNÉES RÉELLES

Dans cette section, nous montrons comment les distances et indices définis dans ce mémoire peuvent être appliqués à des arbres phylogénétiques et RSS simples. De plus, nous montrons l'efficacité de notre Algorithme I sur la précision et le temps d'exécution pour retrouver les HGT dan les RSS.

3.1 Études de simulations

Dans cette étude de simulations, nous comparons tout d'abord les performances de différentes méthodes de détection des THG, soit RIATA-HGT (un algorithme dans le package PhyloNet, Wen *et al.*, 2018) et HGT-Detection (algorithme du serveur web T-Rex, Boc *et al.*, 2012) avec notre Algorithme 1. Les deux algorithmes mentionnés permettent d'inférer les THG en donnant en entrée une paire d'arbres phylogénétiques de gènes et d'espèces. De ce fait, les entrées utilisées sont des arbres phylogénétiques pour RIATA-HGT et HGT-Detection (qui a été paramétré avec l'option BD-optimization) et non des RSS. Les arbres utilisés sont tirés du jeu de données de Boc *et al.* (2010), qui contient des arbres aléatoires binaires et non-binaires. Le jeu de données est disponible à l'adresse suivante : http://www.labunix.uqam.ca/~makarenkov_v/Simulation_trees.zip. Le jeu de données contient 100 arbres d'espèces enracinés binaires et non-binaires pour chacune des combinaisons des paramètres suivants: Nombre de THG qui varie de 1 à 10; Nombre

de feuilles, qui varie de 10 à 100, avec un pas de 10). La racine a été utilisée comme le point de repère pour séparer les feuilles de l'arbre en deux communautés (ce qui fait aussi du sens biologiquement parlant puisque la distance d'évolution sera la plus grande à partir de la racine). Pour appliquer l'Algorithme 1, ces arbres sont ensuite transformés en des RSS en utilisant la procédure décrit par Atkinson *et al.* (2009). Les valeurs de α pour chaque communauté d'espèces ont été déterminées à l'Étape 2a de l'algorithme. L'erreur de détection moyen de THG consiste en une moyenne des différences absolues entre les nombres totaux de transferts générés et retrouvés (aussi utilisé dans Boc *et al.*, 2010), et la moyenne des temps d'exécution permettant de comparer la vitesse des algorithmes (voir Figure 3.1). La Figure 3.1a résume le résultat des erreurs de détection par chaque méthode analysée; notre Algorithme 1 a été généralement plus efficace que les algorithmes RIATA-HGT et HGT-Detection, surtout pour des arbres plus grands. La performance des méthodes s'améliore avec la taille des arbres. De plus, l'Algorithme 1 était de loin la méthode la plus rapide, la différence étant déjà très marquée à partir des arbres de taille 500 (voir Figure 3.1b). Pour cette analyse de vitesse, nous avons utilisé des arbres de taille plus variable, soit 10, 100, 500, 1000, 5000 et 10 000 feuilles), avec dix exécutions par méthode par taille. Par exemple, l'Algorithme 1 n'a nécessité que 11.4 secondes pour analyser un RSS avec 10 000 nœuds, alors que les deux autres méthodes n'ont pas été capables de compléter les analyses des arbres de tailles 5000 et 10 000 dans un délai raisonnable. Nos simulations ont été effectuées sur un PC équipé d'un processeur Intel Pentium IV dual-core 3.2 GHz et 4 Go de mémoire vive.

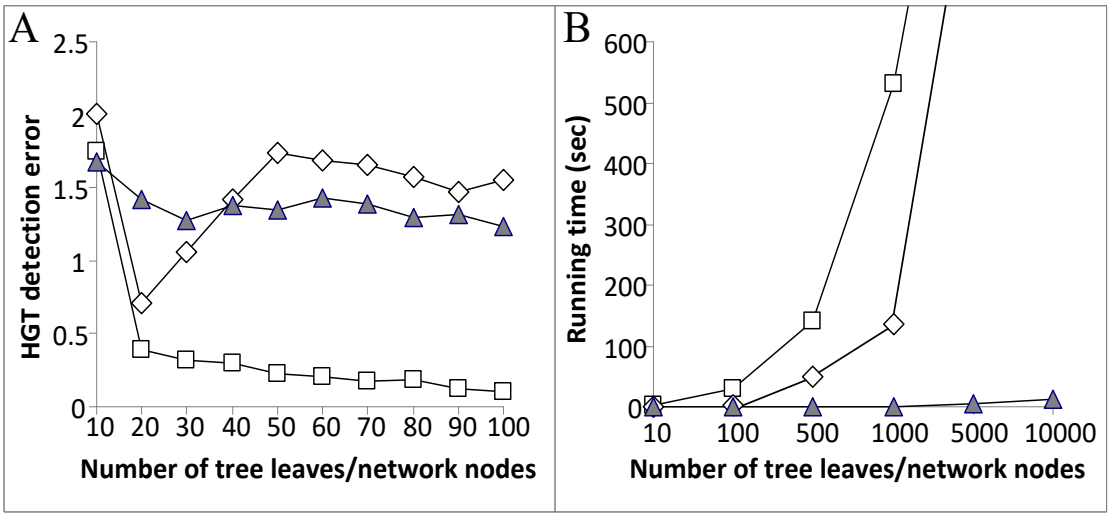


Figure 3.1: (a) erreur de détection de THG, qui est la différence absolue entre le nombre de transferts générés et retrouvés, pour RIATA-HGT (losanges blancs), HGT-Detection (carrés blancs) et l'Algorithme 1 (triangles gris) par rapport au nombre de feuilles de l'arbre ou de nœuds du réseau. Chaque valeur représente une moyenne obtenue sur 100 arbres pour chaque taille considérée; (b) temps d'exécution moyen (en secondes) pour chaque algorithme par rapport au nombre de feuilles ou de nœuds.

Les diagrammes en boîtes des mesures F1 et de rappel sont présentés dans la Figure 3.2. En reprenant les réseaux qui ont été utilisés précédemment dans la Figure 3.1a, les valeurs de ces mesures ont été calculées sur R pour confirmer la capacité de l'Algorithme 1 à identifier correctement les THG.

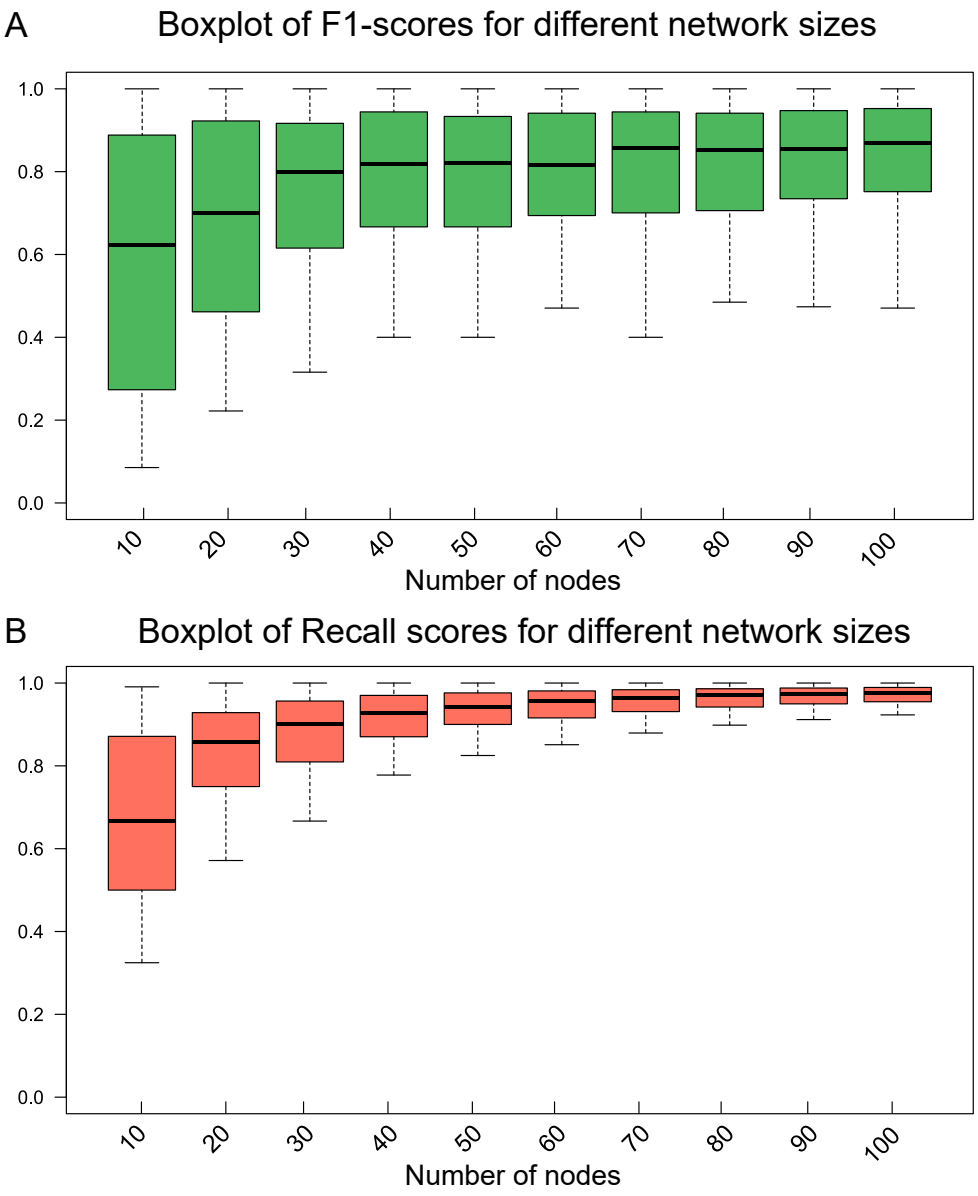


Figure 3.2: Diagrammes en boîte du score F1 (a) et du rappel (b) obtenus pour l'Algorithme 1 sur des RSS aléatoires (de 10 à 100 nœuds). L'axe des abscisses indiquent le nombre de nœud dans le réseau.

3.2 Études de simulations sur de larges données synthétiques

Pour mieux évaluer la performance de l'Algorithme 1 sur la détection des THG dans des réseaux plus larges, nous avons effectué une autre série de simulations. Tout d'abord, des RSS avec 1000 nœuds appartenant à deux communautés d'espèces différentes (communautés *X* et *Y*) ont été générés avec des arêtes aléatoires, et une connectivité prédéterminée. Ici, nous nous sommes spécifiquement concentrés sur notre algorithme, et donc nous n'avons pas effectué de comparaisons avec les méthodes de détection de THG utilisant les arbres). Dans les RSS originaux, la connectivité des espèces de la même communauté variait de 0.1 à 0.3 (ce paramètre a été sélectionné en utilisant une distribution uniforme) et les deux communautés d'espèces sont ensuite reliées par une seule arête (nécessaire pour avoir un RSS connexe). Les RSS avec les ratios d'espèces suivants ont été générés: 100 espèces de *X* / 900 espèces de *Y*; 200 espèces de *X* / 800 espèces de *Y*; 300 espèces de *X* / 700 espèces de *Y*; 400 espèces de *X* / 600 espèces de *Y*; et 500 espèces de *X* et *Y*). Les THG ont ensuite été simulés. Le nombre de nœuds transférés dans chaque transfert variait de 1 à 25. Le nœud donneur de gène et sous-graphes des nœuds transférés ont été sélectionnés aléatoirement. Des pourcentages d'espèces affectées différents par les TGH en tant que receveurs sont utilisés: 0%, 10%, 25% et 50% dans *X* et *Y* (voir l'axe des abscisses dans la Figure 3.3). L'Algorithme 1 a ensuite été exécuté sur ces réseaux artificiels et la valeur de α pour chaque communauté d'espèces a été déterminée tel que spécifié dans l'Étape2a de l'algorithme. Les mesures de F1 et Rappel ont été calculées par notre programme R pour évaluer la proportion des THG détectées (voir Figure 3.3). Les diagrammes en boîte ont été construits sur 1000 valeurs de chacune des mesures pour les différentes combinaisons de paramètres mentionnées ci-haut. La tendance générale suivante est observée pour F1 et le rappel: avec un plus grand nombre de transferts générés, l'Algorithme 1 a plus de difficultés à identifier les transferts. Toutefois, quand le nombre de receveurs de THG de communauté d'espèces la moins affectée était au plus

10% (i.e., les six premières boîtes dans la Figure 3.3a et 3.3b), la médiane était de 0.87 pour F1 et 0.81 pour le rappel, dans le pire des cas.

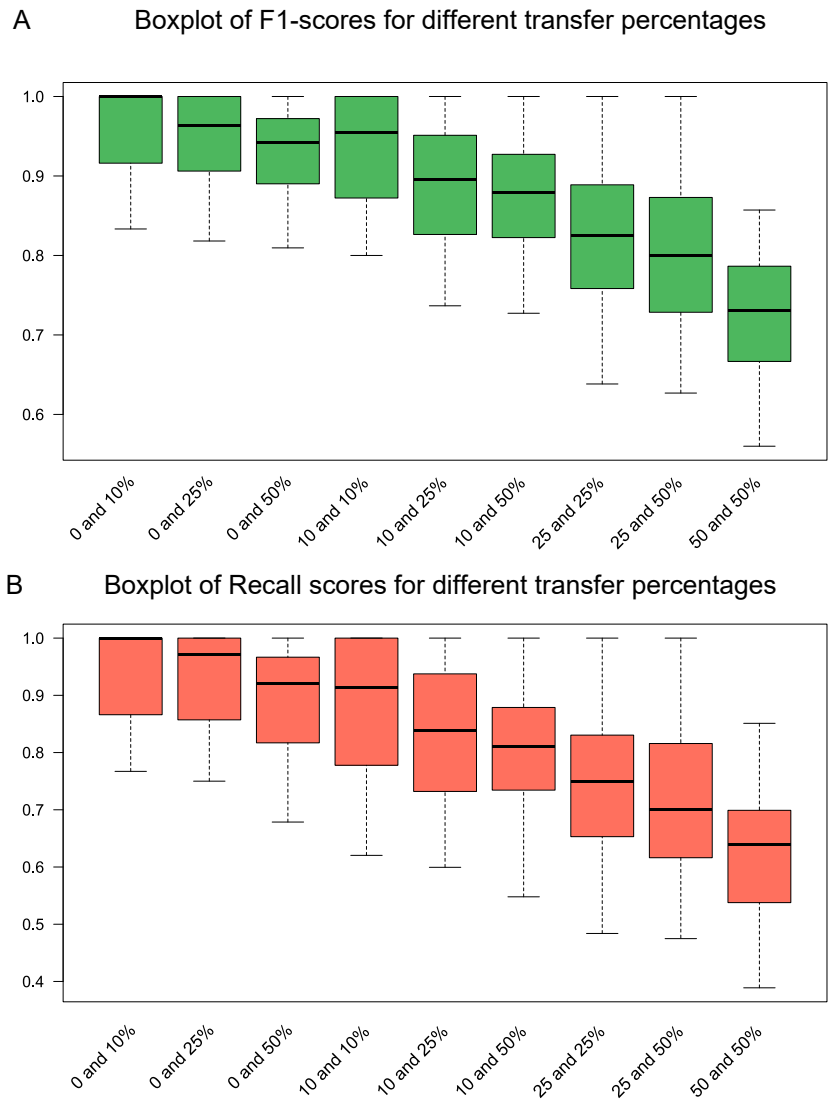


Figure 3.3 : Diagrammes en boîte du score F1 (a) et du rappel (b) obtenus dans la deuxième simulation quand l'Algorithme 1, calculant l'indice de *Transfert*, a été utilisé sur des RSS de 1000 nœuds. L'axe des abscisses indique les différents pourcentages de séquences transférées (i.e., nœuds du réseau) entre deux communautés d'espèces (i.e., 0% et 10% de transferts; 0% d'espèces de la première communauté étaient affectées par les THG des espèces provenant de la deuxième communauté, et 10% de la deuxième communauté étaient affectées par THG des espèces de la première communauté).

CHAPITRE IV

RÉSULTATS

Dans ce chapitre, quelques jeux de données réelles sont utilisés pour mieux illustrer les différentes distances et indices définis qui sont décrites dans le chapitre II et appliqués dans le chapitre III de ce mémoire sur des réseaux artificiels. Les résultats de l'application des distances sont ensuite interprétés selon le type de réseaux utilisés. Les jeux de données ont été tirés de deux articles différents, le premier étant un article traitant de la flore microbienne intestinale et la deuxième traitant des gènes de résistance aux antibiotiques.

4.1 Le réseau d'Esophagus

Les distances définies dans ce mémoire ont été appliquées sur le jeu de données réelles Esophagus (Pei *et al.*, 2004), disponible sur le site internet de Mothur (Schloss *et al.*, 2009, http://www.Mothur.org/wiki/Analysis_exemples), qui est un ensemble d'outils d'analyse métagénomique reconnus dans le domaine de l'écologie. Esophagus est un jeu de données constitué des échantillons de la flore microbienne provenant de l'œsophage distal de patients en santé. Le jeu de données contient les séquences d'ARN 16S de 684 bactéries trouvées pour 3 patients. Chaque patient est considéré comme une communauté dans notre réseau (les patients sont symbolisés par les lettres B, C et D). Il s'agit donc de séquences génétiques, qui peuvent être caractérisées par un réseau. Pour transformer le jeu de données en un RSS, nous avons appliqué la méthode décrite

dans Bapteste *et al.* (2012), en utilisant un BLAST local sur les séquences pour les comparer entres elles. Les liens entre les séquences (nœuds dans le réseau) où la similarité était au-dessus d'un certain seuil (97% dans ce cas) ont été conservés dans un fichier pour devenir les arêtes du RSS. Parce que les comparaisons ont été faites par paire, nous avons obtenu des sous-graphes contenant deux et seulement deux communautés. Les distances *NetUniFrac*, *Spp*, *Spep*, *Spelp* et *Spinp* ont été calculées sur ces graphes. L'indice de *Transfert* n'a pas été calculé sur ce jeu de données car les patients B, C et D n'ont pas eu d'interactions entre eux. En plus de nos distances, nous avons calculé la distance *UniFrac* classique sur un arbre phylogénétique. L'arbre a été créé à partir des mêmes séquences à l'aide d'un des outils disponibles dans Mothur. La Table 4.1 montre les métadonnées des RSS construits. Le nombre de nœuds dans les trois sous-graphes était semblable, mais le nombre d'arêtes était très différent d'un sous-graphe à l'autre.

Table 4.1 : Nombre de nœuds et d'arêtes pour chaque comparaison d'une paire de communautés, ainsi que le nombre total de nœuds et d'arêtes (construits en utilisant un BLAST local sur un seuil de 97%).

Sous-graphe	*Nœuds*	*Arêtes*
B-C	454	8443
C-D	486	12604
B-D	428	16310
Réseau complet	684	25061

Table 4.2 : Les valeurs des distances entre les communautés bactériennes B-C, C-D et B-D pour le jeu de données d'Esophagus pour notre RSS.

	Spp	*Spep*[1]	*Spinp*	*NetUnifrac*	*Unifrac*[2]	*p-valeur*[3]
B-C	0.52	0.65	0.71	0.61	0.64	<0.001
C-D	0.47	0.58	0.62	0.78	0.68	<0.001
B-D	0.50	0.62	0.67	0.59	0.62	0.129

En se basant sur les résultats des distances *Spp*, *Spep*, *Spelp* et *Spinp* présentées dans la Table 4.2, les microbiomes des patients C et D sont les plus ressemblants, et ceux des patients B et C les plus distincts. En général, il n'y a pas une grande différence entre les distances obtenues pour les paires de ces patients, ce qui veut dire qu'il y a à la fois une certaine cohérence dans l'interaction des microbiomes, et une différence de base entre ces différents patients sains. Bien sûr, l'échantillon présent est beaucoup trop petit pour faire une analyse approfondie de la flore microbienne de ces patients. Par la suite, il serait intéressant de continuer une analyse avec des échantillons de personnes malades.

Dans nos calculs, les valeurs de *Spep* et *Spelp* sont les mêmes, puisque les arêtes n'ont pas de longueurs ou poids spécifiques qui leur est rattachés. La différence entre les

[1] Les valeurs de *Spep* et *Spelp* sont identiques
[2] La distance UniFrac non-pondérée a été calculée sur l'arbre phylogénétique construit avec Mothur.
[3] La p-valeur pour le test de significativité sur Unifrac non-pondéré avec 1000 permutations. La p-valeur pour les 3 communautés est <0.001, ce qui veut dire qu'au moins un des patients a une différente structure. Le seuil de la p-valeur est $0.05/3 \approx 0.0166$ à cause des analyses multiples.

valeurs est beaucoup plus grande pour la distance *NetUnifrac*, mais les valeurs obtenues pour ces distances ne corrèlent pas avec les résultats obtenus pour les distances de plus courts chemins : *Spp*, *Spep*, *Spelp* et *Spinp*. Ceci montre que les distances basées sur les plus courts chemins soutirent des informations qu'il ne serait pas possible d'obtenir avec *NetUniFrac*. Basé sur *NetUniFrac*, nous aurions dit que les patients ont des microbiomes différents dans leur œsophage, avec des valeurs des distances qui varient entre 0.59 et 0.78, et qui sont plus élevées par rapport aux distances des plus courts chemins et l'*UniFrac* classique. Les résultats obtenus pour la paire de patients C-D démontrent le mieux cette disparité entre les valeurs des distances avec les plus courts chemins et celles obtenues avec *NetUniFrac*, alors que nous observons respectivement les plus petites et les plus grandes valeurs. Les valeurs d'*UniFrac*, nécessairement obtenues sur des arbres, ont été calculées à partir d'un arbre construit avec l'outil Mothur. Comme mentionné plus haut, il n'y a pas de THG possibles entre les communautés, car les communautés représentent les patients qui n'ont pas eu d'interactions. Cependant, si l'analyse était portée sur des communautés du même microbiome, il serait beaucoup plus pertinent de calculer l'indice de *Transfert* pour ces données, car les THG sont très fréquents dans le microbiome d'une personne (Martiny *et al.*, 2015).

4.2 Analyse des regroupements TetA et CAT dans un réseau de gènes de résistance aux antibiotiques

Une application de l'indice de *Transfert* dans un RSS concerne la dispersion des GRA. La problématique des superbactéries provient de la mauvaise utilisation des antibiotiques prescrits aux patients, mais aussi de leur utilisation dans la nature et l'agriculture avec la surutilisation des pesticides et autres produits chimiques sur les plantes. En effet, les barrières taxonomiques et géographiques sont souvent franchies puisque les bactéries qui sont normalement phylogénétiquement distinctes (e.g. des représentants de Corynebacterium et Entérobactérie) et de différents habitats (e.g.

organisme hôte ou milieu aquatique) peuvent partager la même résistance à un antibiotique due au TGH (Bengtsson-Palme *et al.*, 2018). L'indice de *Transfert* pourrait permettre de visualiser la dispersion de ces GRA qui se sont répandus à travers des THG. Nous utilisons les données publiées dans une étude de Fondi et Fani (2010) sur les GRA de bactéries de différents environnements, soit le sol, les organismes hôtes (Hôte), les milieux aquatiques (Aquatique) et celles qui sont présentes dans plusieurs/tous ces habitats (Omniprésent). Cette information est annotée automatiquement dans la base de données GOLD. Cependant, certaines séquences n'ont pas été retrouvées dans la base de données, l'environnement a été considéré comme inconnu. Fondi et Fani (2010) ont construit un grand RSS en utilisant les séquences de GRA disponibles sur ARDB (*Antibiotic Resistance Genes Database*). Leur RSS complet contient 5030 nœuds et 259 726 arêtes.

Pour enlever les transferts verticaux et transferts horizontaux intra-genre, les auteurs ont utilisé la valeur d'identité pondérée (WIV), calculée à partir de deux réseaux. Le premier réseau était le RSS construit pour les séquences protéiques de GRA mentionnés plus haut, et le deuxième RSS a été basé sur les gènes ARN 16S des mêmes espèces, construit avec un seuil de BLAST de 97%. Ainsi, en comparant les deux RSS, il était possible de calculer une distance taxonomique sur le premier RSS des GRA. Les arêtes sur ce réseau ont été pondérées en utilisant la distance entre les ARN 16S. Plus la distance taxonomique entre deux espèces est grande, plus le WIV sera élevé. Le réseau final a été obtenu avec un WIV de 9, qui était le seuil qui optimisait la détection des transferts horizontaux et excluait au mieux les transferts horizontaux intra-genre. Les regroupements TetA et CAT ont ensuite été soutirés pour en créer des réseaux indépendants qui sont illustrés dans la Figure 4.1. Les valeurs des distances et indices de *Transfert* pour ces deux réseaux sont montrées dans les tables Table 4.3.

Il est connu que les transferts de la résistance à la tétracycline apparaissent fréquemment entre différentes espèces de bactéries dans le corps humain, provenant

soit d'autres espèces dans le microbiome ou de pathogènes externes (Speer *et al.*, 1992). Par exemple, les transferts passant par les plasmides ont été le moyen principal pour l'acquisition et la dissémination de la résistance à la tétracycline dans le *Laribacter hongkongensis* (Lau, Wong *et al.*, 2008). De plus, les incidences de multirésistance parmi les isolats de *Aeromonas spp.*, qui sont à la fois des pathogènes chez les poissons et les humains, proviennent d'éléments génétiques mobiles (Jacobs et Chenia, 2007). La résistance au chloramphénicol suit un motif similaire dans les bactéries des genres Staphylococcus (Bhakta, Arora *et al.*, 2003), Neisseria (Galimand, Gerbaud *et al.*, 1998), Enterococcus (Gould, Fishman *et al.*, 2004) et Salmonella (Karunaratne, Wickremesinghe *et al.*, 2000). D'autres études ont révélé que les bactéries dans l'eau de la mariculture sont souvent résistantes au chloramphénicol (Dang, Zhao *et al.*, 2009). Notamment, la présence de liens entre ces espèces, sans égard à leur habitat et/ou classification taxonomique, suggère fortement que cette résistance aux antibiotiques provient des THG.

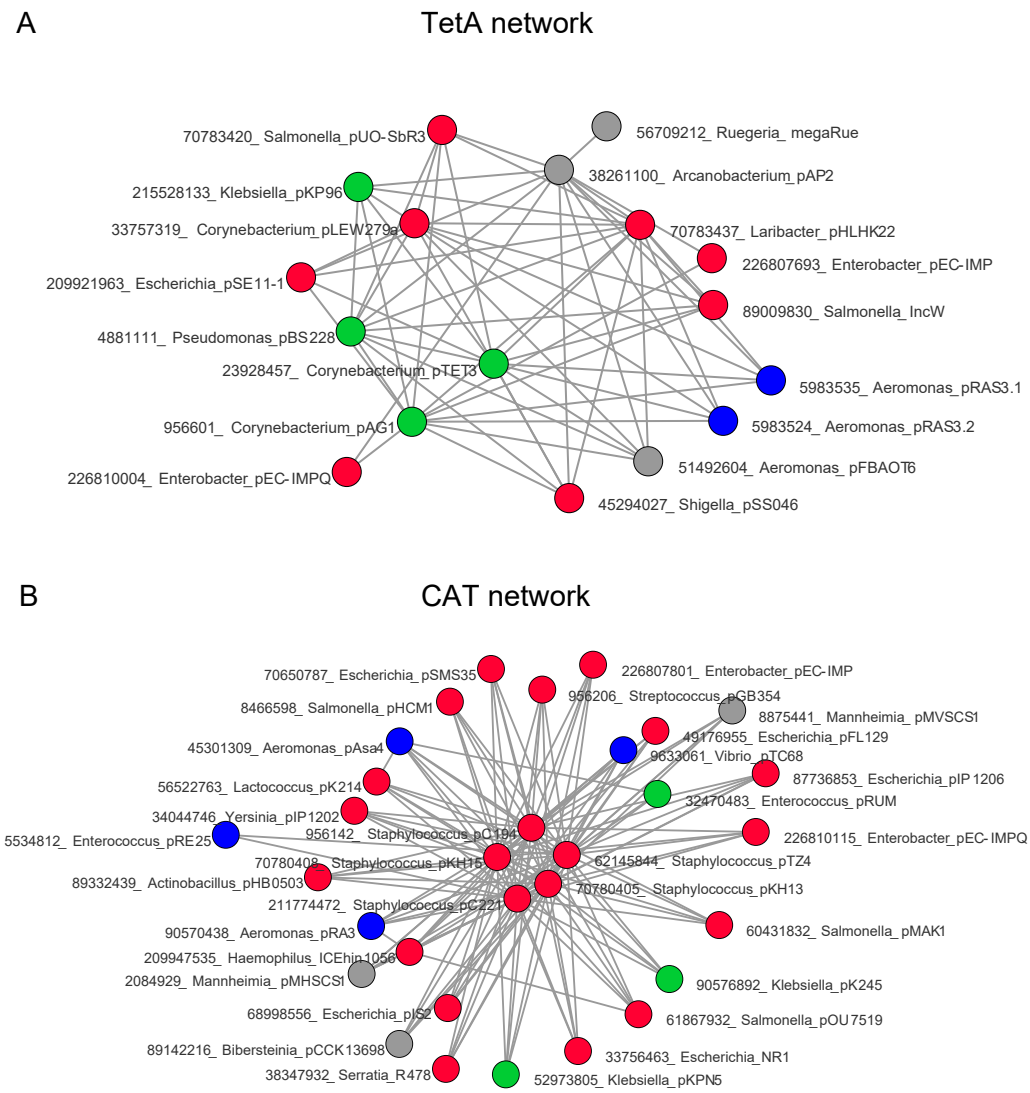


Figure 4.1 : Les RSS pour les gènes de résistance aux antibiotiques de TetA et CAT de la section *Supporting information* S4 dans Fondi et Fani (2010). Les nœuds représentent des protéines (GRA) d'organismes bactériens et les arêtes proviennent de la mesure de WIV. Les nœuds sont colorés selon l'habitat assigné par la base de données GOLD: les nœuds rouges pour les organismes dans les hôtes, les nœuds bleus pour les organismes aquatiques, les nœuds verts pour les organismes dans plusieurs ou tous les habitats (omniprésent) et les nœuds gris pour les organismes manquant dans la base de données GOLD. Les nœuds ou arêtes redondants ont été enlevés du réseau.

Table 4.3 : Valeurs des distances et des indices obtenues pour le réseau TetA de la Figure 4.1. La direction des mesures de transferts est donnée par *reverse* et *direct*.

Mesure/ Habitats	*NetUnifrac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer-direct*	*Transfer-reverse*	*Alt Trans direct*	*Alt Trans reverse*
Hôte & Omniprésent	0.38	0.41	0.43	0.43	0.97	0	0.75	0.32	0.83
Hôte & Inconnu	0.56	1.00	1.00	1.00	1.00	0	0.75	1	1
Hôte & Aquatique	0.54	0.86	0.86	0.86	0.86	0	1	0	1
Omniprésent & Inconnu	0.43	0.58	0.60	0.60	0.97	0	1	1	0.54
Omniprésent & Aquatique	0.69	0.97	0.97	0.97	0.90	0	1	0	1
Inconnu & Aquatique	0.43	0.75	0.75	0.75	0.75	0	1	0	1

Table 4.4 : Valeurs des distances et indices obtenues pour le réseau CAT de la Figure 4.1.

Distance/ Habitats	*NetUnifrac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer-direct*	*Transfer-reverse*	*Alt Trans direct*	*Alt Trans reverse*
Hôte & Omniprésent	0.87	0.99	0.99	0.99	0.99	0	1	0	1
Hôte & Inconnu	0.84	0.99	0.99	0.99	0.99	0	1	0	1
Hôte & Aquatique	0.80	0.97	0.97	0.97	0.97	0	1	0	1
Omniprésent & Inconnu	0.40	1.00	1.00	1.00	1.00	1	1	1	1
Omniprésent & Aquatique	0.38	0.78	0.78	0.78	1.00	0.88	0.83	1	0.33
Inconnu & Aquatique	0.43	1.00	1.00	1.00	1.00	1	1	1	1

4.3 Reconstruction des réseaux TetA et CAT

Parce que les RSS de Fondi et Fani (2010) étaient basés sur le WIV, ils ne contenaient pas ou très peu d'arêtes intra-genres, ce qui biaise énormément le calcul des distances des plus courts de chemins et de *NetUniFrac*. En effet, la séparation des communautés taxonomiques signifie qu'il y a beaucoup moins d'arêtes monochromes, ce qui réduit la distance entre les communautés. À cet effet, nous avons reconstruit les RSS de TetA (47 espèces) et CAT (38 espèces) à partir des séquences d'acides aminés provenant du jeu de données de Fondi et Fani (2010) dans leur section *Supporting information*. L'alignement de séquences multiples de Muscle avec les options par défaut (Edgar, 2004) et le calcul de distances utilisant le modèle de substitution protéine JTT (Jones *et al.*, 1992) ont été faits sur le serveur web T-Rex (Boc *et al.*, 2012). Les séquences protéiques alignées de TetA et CAT sont disponibles à l'adresse URL suivante: https://www.labunix.uqam.ca/~makarenkov_v/TetA_CAT_séquences.zip.

Deux classifications des espèces ont été considérées dans notre étude – la classification taxonomique et la classification environnementale. Selon la classification taxonomique (correspondant aux annotations taxonomiques sur NCBI), les regroupements d'espèces étaient divisés en communautés d'Actinobactérie, Bacilli, γ-protéobactérie et Autres bactéries pour les séquences dans le réseau TetA, et en Bacilli, γ-protéobactérie et Autre bactéries pour les séquences dans le réseau CAT. Selon la classification environnementale (voir le fichier en format LEDA dans la section Supporting information de Fondi et Fani, 2010), les regroupements d'espèces étaient divisés en communauté d'Hôte, Omniprésent, Inconnu et Aquatique pour les séquences dans les deux réseaux TetA et CAT.

Tout d'abord, le seuil de similarité pour connecter les nœuds a été choisi en comparant un certain nombre de possibilités avec la relation entre le seuil et le nombre d'arêtes

qui sont inclues dans les réseaux. Ces informations sont résumées dans les Figures 4.2a et 4.2b.

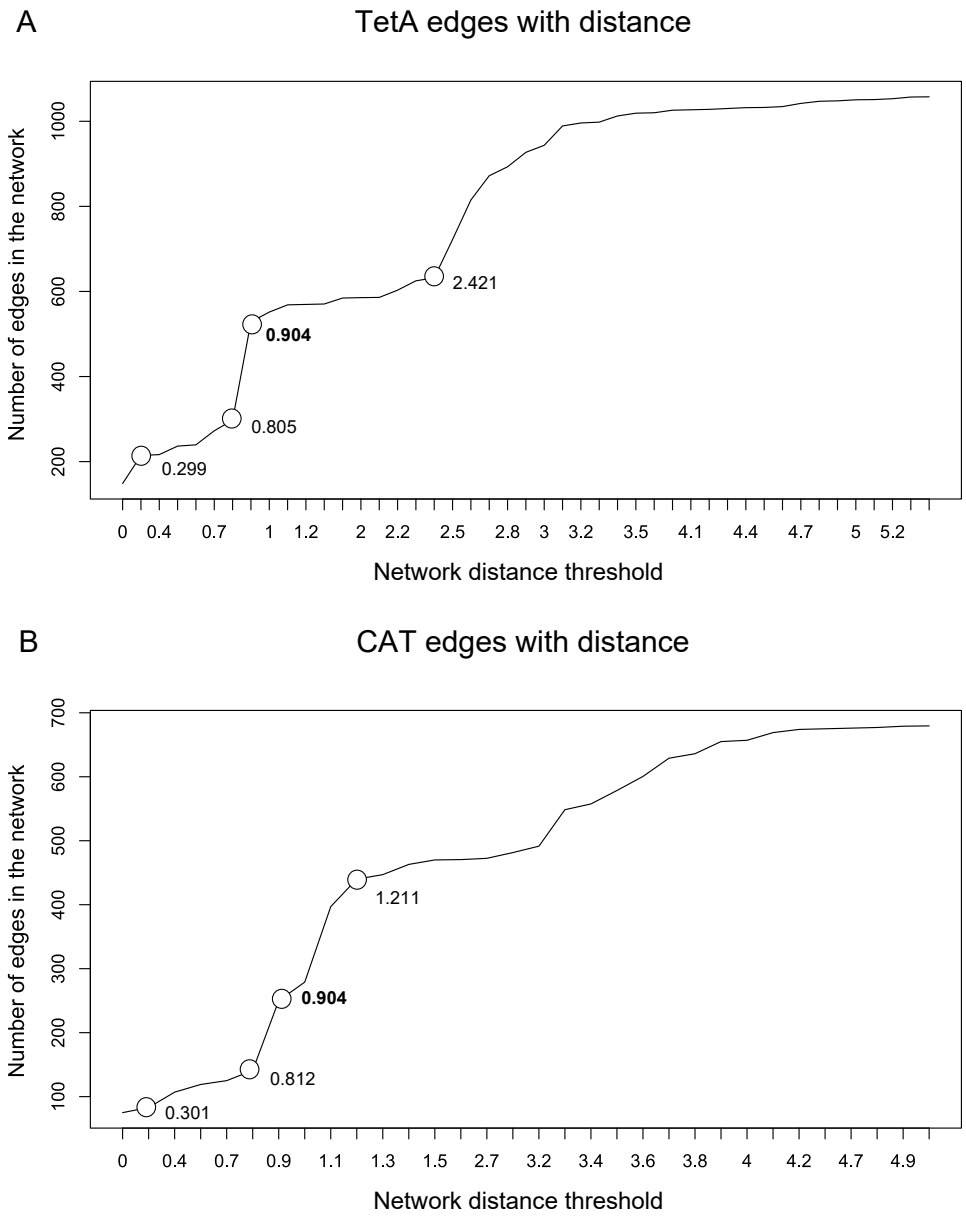


Figure 4.2: Chaque graphique montre la relation entre le nombre d'arêtes dans le RSS et le seuil choisi pour les réseaux de TetA (A) et CAT (B). Pour les deux réseaux 4 points pivots sont présentés comme des candidats possibles pour choisir le seuil des réseaux. La valeur de seuil de 0.904 a été choisie pour les deux réseaux TetA et CAT.

Nous avons utilisé le même seuil pour choisir les arêtes dans les réseaux TetA et CAT. Plus précisément, le seuil de 0.904 pour les réseaux TetA (taxonomique et environnemental, voir les Figures 4.3a et 4.4a), et pour les réseaux CAT (taxonomique et environnemental, voir les Figures 4.3b et 4.4b). Ce seuil a été appliqué avec les distances produites par le modèle JTT. Bien sûr, nos réseaux sont différents de ceux inférés par Fondi et Fani (2010).

Comme nous pouvons l'observer dans la Figure 4.3, les deux communautés d'espèces les plus larges, i.e., γ-protéobactéries et Bacilli sont bien séparées et forment des regroupements dans les réseaux taxonomiques de TetA et CAT (Figures 4.3a et b). Cependant, les communautés d'Actinobactéries et d'Autres bactéries sont trop petites et ne sont pas séparées en regroupements clairs dans ces réseaux. Les valeurs de l'indice de *Transfert*, calculées par l'Algorithme 1, ont été obtenues pour les réseaux taxonomiques TetA et CAT (voir les Tables 4.5 et 4.6). Ici, la séparation des communautés de γ-protéobactéries et Bacilli dans les réseaux taxonomiques TetA et CAT se traduit par des valeurs élevées de l'indice de *Transfert* pour les transferts dans les deux directions (pour les transferts: γ-protéobactérie → Bacilli, l'indice est de 1 dans les deux réseaux, et pour les transferts Bacilli → γ-protéobactérie, l'indice est de 0.97 dans le réseau TetA et de 0.96 dans le réseau CAT, représentant un événement de THG dans les deux cas).

Quand il s'agissait de la classification environnementale des habitats d'espèces ont été considérées (voir Figures les 4.4a et 4.4b), Les communautés sont plus mélangées entres elles par rapport aux réseaux taxonomiques. Il est raisonnable de considérer une division des espèces en communautés environnementales en plus de taxonomiques, car la propagation des antibiotiques dépend aussi des contraintes géographiques. Dans la classification environnementale, la plupart des séquences appartiennent à des organismes hôtes, c'est-à-dire que la bactérie vit principalement dans d'autres organismes (tel que l'humain). Ces séquences Hôtes sont bien mélangées avec des

espèces des communautés Omniprésent, Aquatique et Inconnu. Aucune espèce provient strictement du sol. Puisque la communauté Hôte contenait la majeure partie des espèces, la plupart des valeurs de l'indice de *Transfert* à partir des trois autres environnements vers l'Hôte ont été égales à 1 (voir les Tables 4.7 et 4.8). Les transferts détectés proviennent majoritairement de la communauté Hôte vers les autres environnements. Parfois, la direction du transfert est difficile à identifier, et dans ce cas notre Algorithme 1 indique plutôt la plus grande communauté comme étant la communauté donnant le transfert.

Table 4.5 : Indice de *Transfert* dans les communautés de TetA selon la classification taxonomique. La valeur de 1 indique qu'aucun transfert n'a eu lieu de la communauté de la ligne vers la communauté de la colonne. L'écart-type des valeurs de l'indice obtenu pour les 4 points pivots présentés sur la Figure 4.2 est indiqué entre parenthèses après chaque valeur.

Communauté	**Actino-bactéries**	**Bacilli**	**γ-protéo-bactéries**	**Autres bactéries**
Actino-bactéries	-	1 (0)	1 (0)	1 (0.14)
Bacilli	1 (0)	-	0.97 (0.013)	1 (0)
γ-protéo-bactéries	0.5 (0.41)	0.95 (0)	-	0.67 (0.17)
Autres bactéries	1 (0.22)	1 (0)	1 (0)	-

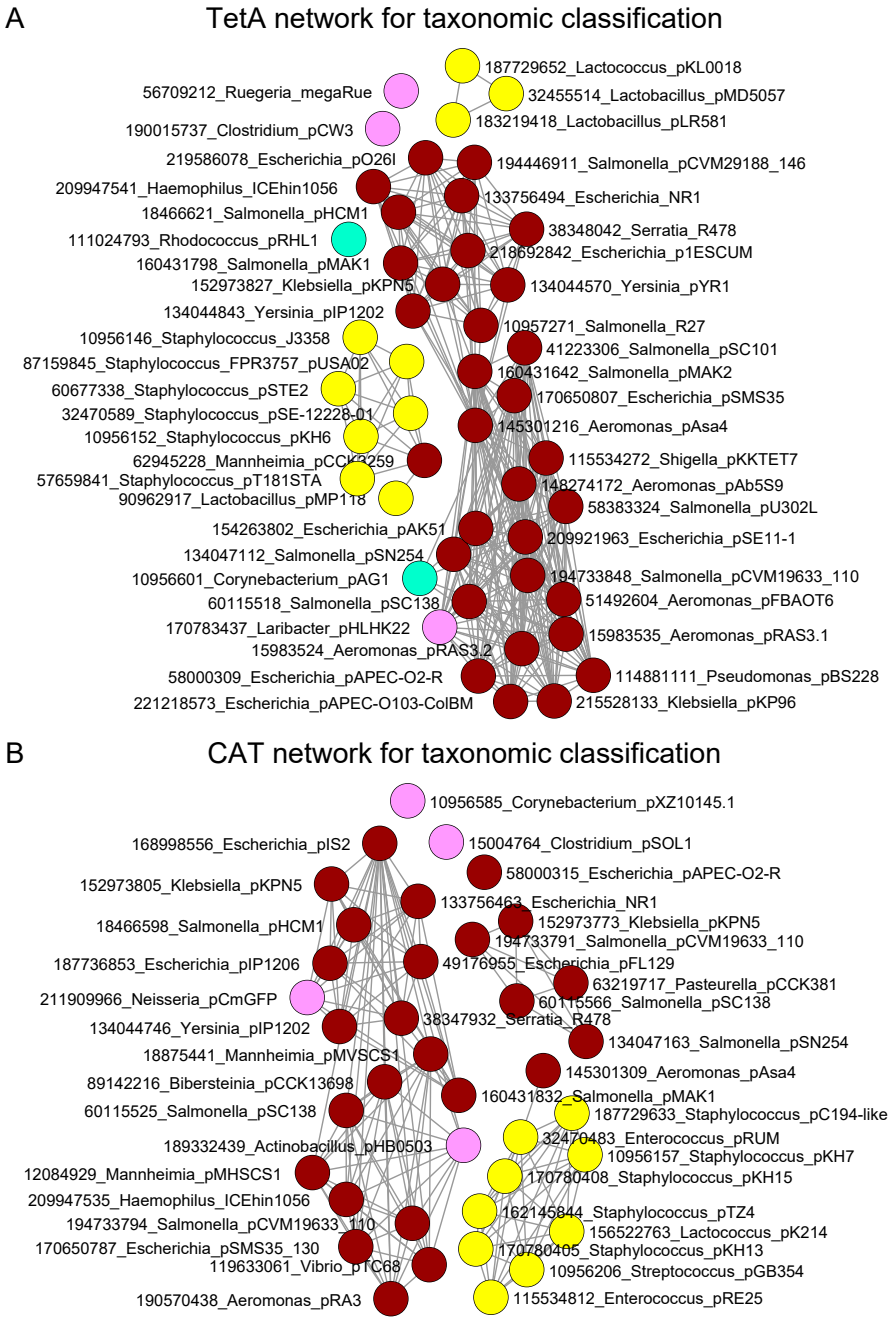


Figure 4.3 : Les réseaux taxonomiques TetA et CAT pour les communautés d'espèces Actinobactéries (turquoise), Bacilli (jaune), γ-protéobactéries (marron), et Autres bactéries (rose) construits en utilisant de la matrice de distances JTT, qui a été calculée à partir de séquences alignées avec *Muscle*. Le seuil de distance de 0.904 pour les réseaux TetA et CAT a été utilisé pour déterminer les arêtes du réseau (i.e., une arête entre deux nœuds a été ajoutée si la distance est plus petite que ce seuil).

Parce que les deux nœuds de la communauté Actinobactéries sont séparés dans le réseau TetA, l'indice de *Transfert* est de 0. Les nœuds dans les communautés Bacilli et γ-protéobactéries possèdent beaucoup d'arêtes les reliant à d'autres nœuds de la même communauté, et donc la valeur de l'indice de *Transfert* est plus grande puisque les possibilités de transfert sont moins nombreuses.

Les réseaux dans l'article original de Fondi et Fani (2010) montre plutôt les espèces séparées par l'environnement, parce que la propagation des antibiotiques dépend aussi des barrières géographiques, ce qui était leur sujet d'étude. C'est aussi un bon moyen pour montrer l'importance de choisir les bonnes communautés pour mettre en évidence les attributs désirés. Le réseau est bien sûr différent selon qu'il est séparé en communautés taxonomiques ou environnementales, ce qui apportera une interprétation différente aux distances que nous avons obtenues.

Table 4.6 : Indice de *Transfert* pour les communautés taxonomiques du réseau CAT.

Communauté	**Bacilli**	**γ-protéobactéries**	**Autres bactéries**
Bacilli	-	0.96 (0.017)	1 (0.21)
γ-protéo-bactéries	1 (0)	-	0.5 (0)
Autres bactéries	1 (0)	1 (0)	-

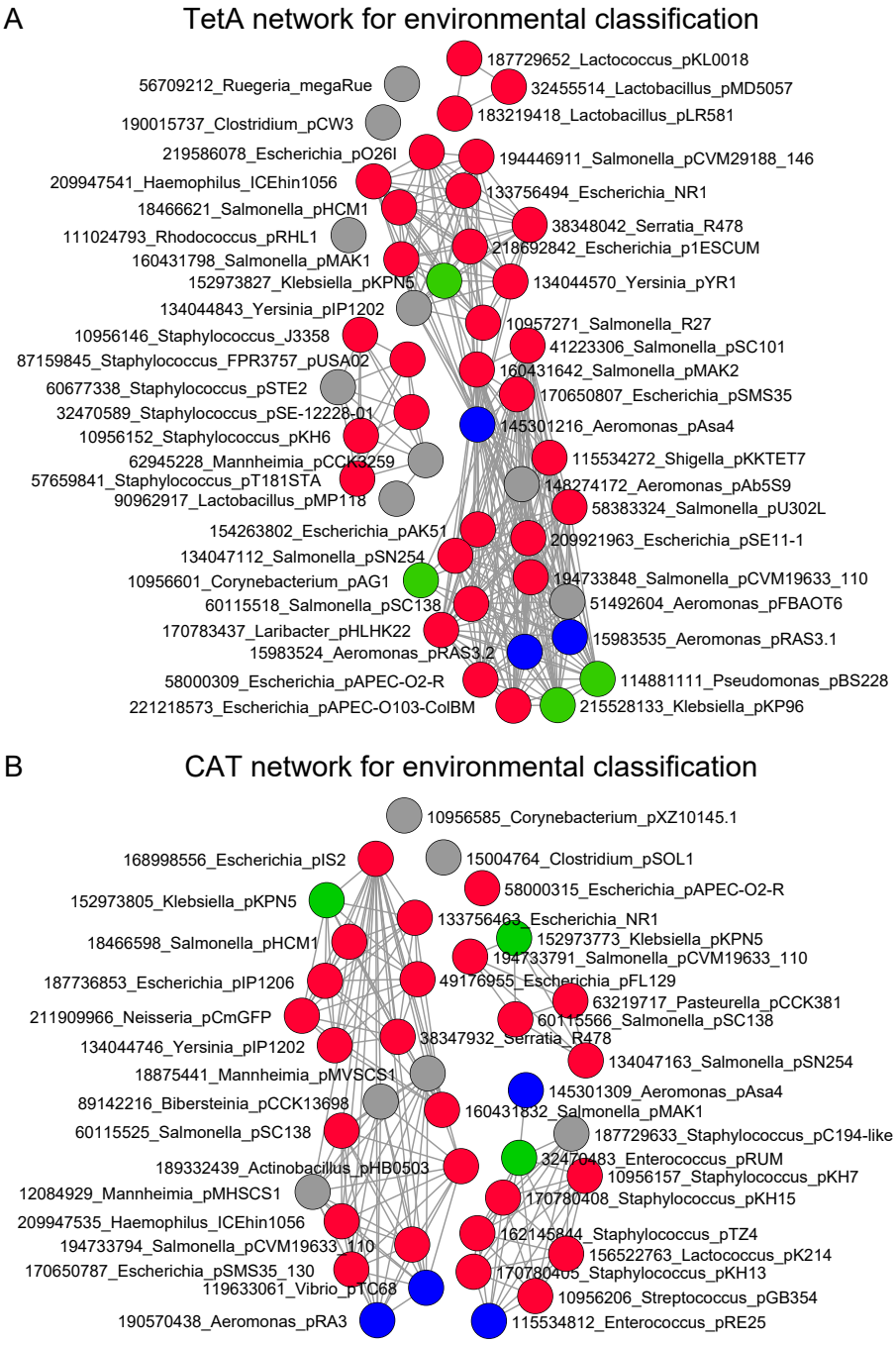


Figure 4.4 : Les réseaux environnementaux TetA et CAT pour les communautés d'espèces Hôte (rouge), Omniprésent (vert), Aquatique (bleu), et Inconnu (gris) construit en utilisant la matrice de distances JTT, qui a été calculée à partir de séquences alignées avec *Muscle*. Le seuil de distance de 0.904 pour les réseaux TetA et CAT a été utilisé pour déterminer les arêtes du réseau.

Table 4.7 : Indice de *Transfert* pour les communautés environnementales du réseau TetA.

Communauté	Hôte	Omniprésent	Inconnu	Aquatique
Hôte	-	0 (0.18)	0.33 (0.17)	0 (0.14)
Omniprésent	1 (0)	-	1 (0.073)	1 (0.36)
Inconnu	1 (0)	0.25 (0.36)	-	0.5 (0.43)
Aquatique	1 (0)	0 (0.27)	0.83 (0.050)	-

Table 4.8 : Indice de *Transfert* pour les communautés environnementales du réseau CAT.

Communauté	Hôte	Omniprésent	Inconnu	Aquatique
Hôte	-	0 (0.074)	0.33 (0.25)	0.25 (0.13)
Omniprésent	1 (0.017)	-	0.92 (0.052)	0.75 (0.12)
Inconnu	1 (0.017)	0.83 (0.26)	-	0.38 (0.26)
Aquatique	1 (0)	0.83 (0.22)	0.92 (0.041)	-

Les distances *NetUniFrac* globales, calculées pour les 4 réseaux dans les Figures 4.3 et 4.4, et leurs écart-types calculés pour les 4 points pivots sont comme suit :

TetA classification taxonomique 0.897 (0.067), CAT classification taxonomique 0.864 (0.164), TetA classification environnementale 0.480 (0.075) et CAT classification environnementale 0.517 (0.059). Dans la plupart des cas, ces écarts-types sont petits, ce qui montre une stabilité des valeurs de *NetUniFrac* et de l'indice de *Transfert* (Tables 4.5 à 4.8) pour ces données, et surtout pour les communautés d'espèces plus larges, telles que les γ-protéobactéries dans les réseaux taxonomiques et les organismes hôtes dans les réseaux environnementaux.

CONCLUSION

Dans un arbre phylogénétique, la distance des plus courts chemins est utilisée pour modéliser la distance d'évolution entre les espèces. Dans ce travail, nous nous servons de ce concept pour montrer comment il peut être étendu à l'analyse des réseaux de similarité de séquences. Nous avons défini cinq nouvelles distances entre communautés dans un réseau de similarité de séquences, incluant *NetUniFrac* (une généralisation de la distance *UniFrac* traditionnelle pour les réseaux), *Spp*, *Spep*, *Spelp* et *Spinp* (basées sur les plus courts chemins), et l'indice de *Transfert* (les versions originale et alternative). *NetUniFrac* peut être calculée dans un temps linéaire par rapport au nombre d'arêtes dans le réseau. L'indice de *Transfert* permet d'estimer le taux et la direction des transferts horizontaux de gènes entre des communautés différentes. Ces mesures peuvent être utiles dans l'analyse de microbiomes de différentes personnes et dans l'analyse des gènes de résistance aux antibiotiques. En général, une valeur qui se rapproche de 0 pour toutes ces mesures suggère que les communautés qui sont comparées interagissent beaucoup, alors que leurs valeurs proches de 1 indiquent peu d'interactions entre elles.

Tel qu'observé dans l'analyse du jeu de données d'Esophagus, les valeurs de la distance *NetUniFrac* calculées pour un réseau de similarité de séquences corrèlent bien avec les valeurs d'*UniFrac* dans un arbre phylogénétique correspondant. Dans ce même exemple, *NetUniFrac* montre mieux les différences entre les microbiomes des patients qu'*UniFrac* classique. De plus, la calcul de *NetUniFrac* peut être complété beaucoup plus rapidement que celui d'*UniFrac* puisque l'inférence d'un réseau s'effectue habituellement en $O(n^2)$ opérations (pour calculer toutes les paires de distances entre les nœuds du réseau), alors que l'inférence d'un arbre phylogénétique s'effectue plutôt

en $O(n^3)$ opérations élémentaires (comme dans le cas du populaire algorithme de *Neighbor-Joining* (Saitou et Nei, 1987)), où N est le nombre de nœuds dans un réseau ou le nombre de feuilles dans un arbre.

De plus, l'indice de *Transfert* peut être calculé en temps linéaire par rapport au nombre d'arêtes dans le réseau en utilisant notre Algorithme 1. Les résultats des études de simulations indiquent que l'Algorithme 1 retrouve les regroupements d'espèces affectées par des transferts horizontaux de gènes et détermine bien la communauté donneuse d'où ces transferts proviennent. L'Algorithme 1 a été implémenté en R et en C et peut être retrouvé dans notre package *NetFrac* disponible à: https://github.com/XPHenry/Netfrac. Au futur, il serait intéressant d'adapter l'Algorithme 1 pour la détection des gènes mosaïques et des gènes composites pour comparer sa performance à celles des programmes *CompositeSearch* (Pathmanathan *et al.* 2018), *FusedTriplets* et *MosaicFinder* (Jachiet *et al.* 2013). De plus, l'interprétation des distances reste un point à préciser, car elles peuvent être calculées non seulement entre les communautés différentes, mais aussi pour des réseaux qui contiennent d'autres données que les séquences biologiques (par exemple, la dispersion des espèces en écologie où un nœud représente un OTU (*Operational taxonomic unit*) recueilli dans un certain échantillon). Dépendamment du réseau et de la provenance des données, il est nécessaire d'avoir plus de flexibilité sur l'information que les valeurs des distances nous apportent dans l'analyse de réseaux de différents domaines.

L'intégration des données phylogéniques et écologiques est aussi une piste à explorer. L'utilisation des données phylogéniques dans l'analyse des communautés écologiques a permis de révéler une multitude de processus qui mènent à l'assemblage des communautés, ainsi que les conséquences de processus, telles que la spéciation, l'adaptation et l'extinction (Cavender-Bares *et al.*, 2009). De la même manière, il est possible de se pencher sur la question des microbiomes et leur analyse avec une approche phylogénétique, soit en analysant l'interaction et les communautés

microbiennes avec des données phylogéniques pour donner une nouvelle perspective entourant la composition, la compétition et l'interaction entre les espèces de notre flore microbienne (Martiny *et al.*, 2015). Dans tous ces cas, l'utilisation des réseaux de similarité de séquences ou d'autres types de réseaux avec une analyse des plus courts chemins peut être bénéfique à la compréhension et la transformation de ces données en information pertinente.

APPENDICE A

CODE DU PACKAGE R NETFRAC

Dans cet appendice nous présentons les principales fonctions du package NETFRAC que nous avons implémenté en langage R et rendu disponible sur : https://github.com/XPHenry/Netfrac.

Main.R

```
#=========== Function multicore (taken from BRIDES.r)

multicore<- function(nc=0) {
  cores <- if (.Platform$OS.type == "windows")
    1
  else
    min(8L, ceiling(detectCores()/2))

  getOption("mc.cores", cores)
  if (nc!=0) return (nc)
  return (cores)
}

#============ Main function

#=====================================================================

#= Main function that returns distance measures for a tree or a
graph
#=
#= Input: x, an igraph object (with $weight attribute and $tax
attribute for colors, see SDDE::load_network()) or a phylo-class
object(see ape::read_tree)
#=        type, "graph" or "tree"
#=        distance, "all" #TODO rajouter les autres options pour
s?lectionner une seule distance
```

```
#=        col1 and col2, colors to be analyzed (correspond to the
factor levels of V(g)$tax)
#=        info (for tree only), dataframe (of characters) that
indicates the color of each tips
#=                                  VertexName       Group
#=                                       1             A
#=                                       2             B
#=                                       3             B
#=                                       4             A
#=
#= Output: numeric vector with all results
#=
#= All distances are made with vertices of same colors (graph) or
external nodes (tree) of same colors
#= and take into account the shortest path
#= Need igraph, SDDE, ape, foreach and doParallel packages
#= Only works with graph and tree with two colors #TODO Works with
two colors at a time Ex: for ABC colors, do A and B, B and C, A and
C
#==========================================================================

#' Main function that computes all the distances and indices
#'
#' @param x The network (or tree) to be analyzed. The network must
be in the igraph format, the edges
#' being accessible with E(x) and vertices with V(x). The
communities are under V(x)$tax and the branch weights
#' are under E(x)$weight. These functions are applicable with igraph
objects only.
#' @param distances Distances or indices to calculate: Transfer,
Transfer2, Spaths, UniFrac and Motifs. The Transfer index,
#' Spaths and Unifrac distances can also be calculated for trees.
#' @param paths This parameter is used to decide whether all the
shortest paths between network nodes should be
#' calculated ("all"), or only one of them ("single"). The last
option can significantly reduce the time of
#' computation. For Motifs distance, decide the size of the motifs
(i.e. 2 or 3).
#' @param mats The similarity matrix used to reconnect the network.
See also the function \code{\link{reconnect}}.
#' @param maxcores Uses the multicore function to set up the number
of cores to be used.
#' @param share_weight Weight if there are mixed communities (when
some species belong to different commmunities).
#'
#' @return
#' $`Spp`\cr
#'A           B\cr
#'A 0.0000000 0.5555556\cr
#'B 0.5555556 0.0000000\cr
#'
#'$Spep\cr
```

```
#'A           B\cr
#'A 0.0000000 0.6296296\cr
#'B 0.6296296 0.0000000\cr
#'
#'$Spelp\cr
#'A           B\cr
#'A 0.0000000 0.6296296\cr
#'B 0.6296296 0.0000000\cr
#'
#'$Spinp\cr
#'A           B\cr
#'A 0.0000000 0.9907407\cr
#'B 0.9907407 0.0000000\cr
#
#'$Transfer\cr
#'A           B\cr
#'A 0.0000000 0.6666667\cr
#'B 0.3333333 0.0000000\cr
#' @export
#'
#' @examples NetFrac(net_a)
#'NetFrac(net_a,"Spaths","all")

NetFrac <- function(x, distances = "UniFrac",paths="single",
mats="", maxcores=1, share_weight =0){
  #create the different combinations using combn on the taxa levels
  taxlvl <- levels(as.factor(V(x)$tax))
  taxlvl <-unique(unlist(strsplit(taxlvl,"-")))
  taxlevel <- combn(taxlvl,2)

  #create an empty matrix with the size of the taxa levels, that we
will fill
  mat <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))
  mat2 <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))
  mat3 <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))
  mat4 <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))
  mat5 <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))

  #use a matrix to show the results between pairs of communities
  if (distances == "Transfer"){
    mat <- Transfer(x)
    return(mat)
  }
  for (i in 1:(length(taxlevel)/2)){
    if(distances == "Transfer2"){
      #recuperate the return of the dist_path for Transfer distance
(asymmetric)
```

```
      transf = dist_paths(x,taxlevel[1,i],taxlevel[2,i], matr=mats,
distance = distances, paths, maxcores=maxcores,share_w=share_weight)
      mat[taxlevel[1,i],taxlevel[2,i]] <- transf[[1]]
      mat[taxlevel[2,i],taxlevel[1,i]] <- transf[[2]]
      mat2[taxlevel[1,i],taxlevel[2,i]] <- transf[[3]]
      mat2[taxlevel[2,i],taxlevel[1,i]] <- transf[[4]]

    # }else if (distances == "Transfer"){
    #   transf = Transfer(x)
    }else if(distances == "Spaths"){
    #if distance chosen are the paths, then there are multiple
values
      transf = dist_paths(x,taxlevel[1,i],taxlevel[2,i], matr=mats,
distance = distances, paths, maxcores=maxcores,share_w=share_weight)
      mat[taxlevel[1,i],taxlevel[2,i]] <- transf[[1]]
      mat2[taxlevel[1,i],taxlevel[2,i]] <- transf[[2]]
      mat3[taxlevel[1,i],taxlevel[2,i]] <- transf[[3]]
      mat4[taxlevel[1,i],taxlevel[2,i]] <- transf[[4]]
      mat5[taxlevel[1,i],taxlevel[2,i]] <- transf[[5]]
      mat5[taxlevel[2,i],taxlevel[1,i]] <- transf[[6]]
      mat[taxlevel[2,i],taxlevel[1,i]] =
mat[taxlevel[1,i],taxlevel[2,i]]
      mat2[taxlevel[2,i],taxlevel[1,i]] =
mat2[taxlevel[1,i],taxlevel[2,i]]
      mat3[taxlevel[2,i],taxlevel[1,i]] =
mat3[taxlevel[1,i],taxlevel[2,i]]
      mat4[taxlevel[2,i],taxlevel[1,i]] =
mat4[taxlevel[1,i],taxlevel[2,i]]

    }else if(distances == "Spp" || distances == "Spep" || distances
== "Spelp" || distances == "Spinp"){
      transf = dist_paths(x,taxlevel[1,i],taxlevel[2,i], matr=mat,
distance = distances, paths, maxcores=maxcores,share_w=share_weight)
      mat[taxlevel[1,i],taxlevel[2,i]] <- transf[[1]]
      mat[taxlevel[2,i],taxlevel[1,i]] =
mat[taxlevel[1,i],taxlevel[2,i]]
      # mat[is.na(mat)] <- 0
    }else if(distances == "UniFrac"){
      mat[taxlevel[1,i],taxlevel[2,i]] <-
NetUnifrac(x,taxlevel[1,i],taxlevel[2,i])
      mat[taxlevel[2,i],taxlevel[1,i]] =
mat[taxlevel[1,i],taxlevel[2,i]]
    }else if(distances == "Motifs"){
      mat[taxlevel[1,i],taxlevel[2,i]] <-
dist_paths(x,taxlevel[1,i],taxlevel[2,i],distance = distances,
paths, maxcores=maxcores,share_w=share_weight)

    # #take care of NAs
    # mat[is.na(mat)] <- 0
    # mat2[is.na(mat)] <- 0
    }
  }
```

```
  if (distances == "Spaths"){
    mat = list(Spp = mat,Spep = mat2,Spelp = mat3,Spinp =
mat4,Transfer = mat5)
  }else if (distances == "Transfer" || distances == "Transfer2"){
    mat = list(Transfer = mat, Transfer2 = mat2)
  }
  return(mat)

}

dist_paths<-function(x, col1, col2, matr="", distance="paths",
paths="single", info=NULL, type="graph", maxcores, share_w){
  #cat("distance between ", col1, " and ", col2, "\n")
  #Register the doParallel parallel background
  if(type == "graph"){
    if(distance == "Spaths" || distance =="Transfer2" || distance ==
"Spp"){

      #Call the function that does the distances with foreach loops
      res_dist =dist_par(x, col1, col2, mat="",
distance,paths,maxcores,share_w)
      #res_dist =dist_par(x, col1, col2,distance,paths,maxcores)
      return(unlist(res_dist))
    }
    else if(distance == "Motifs"){
      if(paths==2){

      # NetFrac for motifs of size 2: number of monocolor
edges/total number of edges
      motif_dist = get_global_nb_motifs_by_colors(x, 2, col1, col2)
      }else if(paths==3){

      # NetFrac for motifs of size 3: number of monocolor size 3
motifs/total number of size 3 motifs
      motif_dist = get_global_nb_motifs_by_colors(x, 3, col1, col2)
      }
      return(motif_dist$ratio)
      #return(unlist(c(NetFrac_2=NetFrac_2$ratio, NetFrac_3=
NetFrac_3$ratio)))
    }

    #res = c(res_dist,list(NetFrac_2=NetFrac_2$ratio, NetFrac_3=
NetFrac_3$ratio))
    #return(unlist(res))

    else if(distance == "UniFrac"){
      return(NetUnifrac(x,col1,col2))
    }
  }else if(type == "tree"){
    if(!is.null(info)){

      ## convert tree to the corresponding network
```

```
      if(distance == "UniFrac"){
        return("ok")
      }
      else if(distance == "Spaths"){
        # reorder info according to tip labels order (phylo class
has tips order from 1 to N and internal nodes n+1 to m)
        info <-info[match(x$tip.label, info[,1]),]

        # convert to character if necessary
        if(class(info[1,1]) != "character"){
          info[1,1] = as.character(info[1,1])
        }

        # remove tips with colors other than color 1 and 2
        rtips_pos <-which( (info[,2]!= col1) & (info[,2]!= col2) )
        rtips <-as.vector(info[,1][rtips_pos])

        if(length(rtips)!=0){ # if there are tips to remove
          x <-drop.tip(x, rtips)
          info <-info[-rtips_pos,]
        }

        tips_number <-c(1:length(x$tip.label))
        tips_name <-x$tip.label
        color_t <-as.vector(info[,2])

        # create graph from edges list
        gtree <-graph_from_edgelist(x$edge, directed = FALSE)
#attention à la racine?

        # set colors for the vertices that correspond to the tips
        colored_tips <-intersect(V(gtree), which(tips_name %in%
(info[,1]))) #la position des tips_name correspond à son index dans
la séquence de vertices
        V(gtree)[colored_tips]$tax = color_t

        # set colors for the vertices that correspont to internal
nodes
        tax <-V(gtree)$tax
        vvector <-as_ids(V(gtree))
        vlistna <-vvector[which(is.na(tax))]
        vlist <-vvector[which(!is.na(tax))]

        vtax<-set_node(x$edge, vvector, vlistna, vlist, tax, col1,
col2)
        V(gtree)$tax = vtax

        # set weights for nodes
        E(gtree)$weight = x$edge.length

        # Get the vertices that correspond to the tips of the tree
        tips = V(gtree)[tips_number]
```

```
        # All paths are calculated from all pairs of same color
external nodes in the tree

        #Register the doParallel parallel background

        #Call the function that does the distances with foreach
loops
        res_dist_tree = dist_tree_par(gtree, col1, col2, tips)
      }

      return(unlist(res_dist_tree))
    }
    else{
      warning("Requires info argument")
      return(NULL)
    }
  } else {
    warning("Type not recognized")
    return(NULL)
  }
}
```

Dist_par.R

```
#=====================================================================
#= Function to calculate dist_paths with doParallel
#=
#= Input: graph, an undirected igraph with color levels
(V(graph)$tax)
#=        nom_col1, character string that indicates the color 1
attribute
#=        nom_col2, character string that indicates the color 2
attribute
#=
#= Output: list of distances values in the following order :
dpr,dl,dlpr,dnpr
#=
#= All shortest paths are calculated between vertices of same colors
#= Requires SDDE, igraph, foreach and doParallel packages
#= Multi-core backend has to be set
#=
#= Ex:
#= Works only for two colors graph
#=====================================================================

number_transfer_gen <- function(d, n){
  a = 1
```

```
  b = 2-n
  c = n*n -n - d*n*n + d*n
  delta = b*b - 4*a*c
  k = (-b - sqrt(delta))/2
  return(k)
}
number_transfer_1 <- function(d=0, n){
  if (d == 0){
    k = n
  } else {
    k = (-1 + 2*n - sqrt(4*d*n*n - 4*d*n +1))/2
  }
  return(k/n)
}
estimation_transfer <- function (x, comm){
  x_A = subgroup_graph(x, comm)
  x_comp = components(x_A)

  num_T = 0
  denom_T = 0

  for (i in 1:x_comp$no){
    size = x_comp$csize[i]
    denom_T = denom_T + size
    if (size > 1){
      num = (size*(size-1))/2
      num_T = num_T + num
    }
  }
  dist_T = num_T/((denom_T*(denom_T-1))/2)
}

shortest_paths_graph <-
function(graph,v1,v2,v.mix,col1,distance,opti,mean_w,max_w){

  x.col1_list <- c()
  #====== Calculation of all the shortest paths in one community
  x.col1 <- lapply(v1,function(x) shortest_paths(graph,x,v1[which(x
== v1):length(v1)],"all")$vpath)

  for(node in x.col1){
    node = node[-1]
    x.col1_list <- append(x.col1_list,node)
  }
  x.col1 <- x.col1_list

  #====== Parameters initialization

  #dpr
  dpr_mix_color_paths=0
  dpr_share_paths = 0
  dpr_monocolor_paths=0
```

```
  #dnpr
  dnpr_no_v=0
  dnpr_mono_v=0
  dnpr_mix_v=0
  dnpr_mix = 0
  dnpr_prop = 0
  #dl
  dl_mix_edges=0
  dl_mono_edges=0
  dl_prop = 0

  #dlpr
  dlpr_mono_edges=0
  dlpr_mix_edges=0
  dlpr_prop = 0

  #=========== Distance Calculation
  for(path in x.col1)
    if (length(path) < 2){
      next
    }
    v_path_col1 <- c()
    v_path_col2 = c()
    v_path_mix = c()
    #===== Associate the path vertices with their colors
    if (distance == "Transfer2"){
      for (node in path){
        if (node %in% v1){
          v_path_col1 <- append(v_path_col1,node)
          }
        else{
          dpr_share_paths = dpr_share_paths + 1.0
          break
          }
        }
      }
    else{
      v_path_col1 = intersect(path, v1)
      v_path_col2 = intersect(path, v2)
      v_path_mix = intersect(path, v.mix)
      }
    #=========== Spp distance

    # Check if there is at least one vertex of a different color as
the start and end vertex

    if (length(v_path_col1) == length(path) && length(path)>1){
      dpr_monocolor_paths = dpr_monocolor_paths + 1.0
    } else if(length(v_path_mix) != 0) {
      dpr_mix_color_paths = dpr_mix_color_paths + 1.0
    } else if(length(v_path_col2) != 0){
```

```
      dpr_share_paths = dpr_share_paths + 1.0
    }
    if (distance != "Transfer2"){
      #=========== Spinp distance
      #===== Remove start vertex and end vertex
      path_i = path[-length(path)]
      path_i = path_i[-1]

      #===== Associate the path vertices with their colors

      v_path_i_col1 = intersect(path_i, v1)
      v_path_i_col2 = intersect(path_i, v2)

      v_path_i_mix = intersect(path_i, v.mix)

      if(length(path_i) == 0){ # If there is no vertex in the path
besides the start and end vertices
        dnpr_no_v = 1.0
      }else if(length(v_path_i_col2) != 0){
        dnpr_prop =
1.0*length(v_path_i_col1)/(length(v_path_i_col1)+length(v_path_i_col
2))
      }else if(length(v_path_i_mix)!= 0){
        # Number of not col1 vertices in shortest path
        dnpr_mix = dnpr_mix + (1.0*length(v_path_i_col2) +
length(v_path_i_mix))

      }
      #calculate the proportion of the nodes same color over all
nodes
      dnpr_mix_v = dnpr_mix_v + (1.0*dnpr_prop/length(path))+
dnpr_no_v

      #=========== Spep and Spelp distance

      if(length(path) >1){
        dl_mix_edges=0
        dl_mono_edges=0

        #dlpr
        dlpr_mono_edges=0
        dlpr_mix_edges=0
        for(l in 1:length(path)){
          if((l+1) <= length(path)){
            # check the end vertices (path[l] and path[l+1]) of each
edge of the path
            if( (V(graph)[path[l]]$tax == col1) &
(V(graph)[path[l+1]]$tax == col1) ){

              dl_mono_edges = dl_mono_edges+ 1
              if (E(graph)[ V(graph)[path[l]] %--%
V(graph)[path[l+1]] ]$weight != max_w){
```

```
                dlpr_mono_edges = dlpr_mono_edges +
(E(graph)[ V(graph)[path[l]] %--% V(graph)[path[l+1]] ]$weight)
              }else{
                dlpr_mono_edges = dlpr_mono_edges + mean_w
              }
            }else{
              dl_mix_edges = dl_mix_edges+ 1
              if (E(graph)[ V(graph)[path[l]] %--%
V(graph)[path[l+1]] ]$weight != max_w){
                dlpr_mix_edges = dlpr_mix_edges +
(E(graph)[ V(graph)[path[l]] %--% V(graph)[path[l+1]] ]$weight)
              }else{
                dlpr_mix_edges = dlpr_mix_edges + mean_w
              }
            }
          }
        }
      #calculate the proportion of monochrome edges
            dl_prop = dl_prop +
(1.0*dl_mono_edges/(dl_mono_edges+dl_mix_edges))
      dlpr_prop = dlpr_prop +
(1.0*dlpr_mono_edges/(dlpr_mono_edges+dlpr_mix_edges))
      }
    }
  }

    if(distance == "Spp" || distance == "Transfer2"){
      print(dpr_monocolor_paths)
      print(dpr_share_paths)
      return(c(dpr_share_paths, dpr_monocolor_paths,
dpr_mix_color_paths))
    }
    res = c(dpr_share_paths, dpr_monocolor_paths,
dpr_mix_color_paths, dnpr_mix_v, dl_mix_edges, dl_prop, dlpr_prop,
dlpr_mix_edges)
    return(res)
}

dist_par<-function(igraph, nom_col1, nom_col2, mat,
distance,opti="single",maxcores=1,share_w){
  cl <- makeCluster(multicore(maxcores))
  registerDoParallel(cl=cl)

  #find the nodes with mixed communities
  col1_mix <- paste(append(nom_col1,"-"),collapse = "")
  col1_mix2 <- paste(append("-",nom_col1),collapse = "")

  #find the ones that contain the communities in this iteration
  v.mix <- grep(col1_mix,V(igraph)$tax)
  v.mix <- append(v.mix,grep(col1_mix2,V(igraph)$tax))

  #same for second community
```

```
  col2_mix <- paste(append(nom_col2,"-"),collapse = "")
  col2_mix2 <- paste(append("-",nom_col2),collapse = "")
  v.mix2 <- grep(col1_mix,V(igraph)$tax)
  v.mix2 <- append(v.mix2,grep(col1_mix2,V(igraph)$tax))

  graph = subgroup_graph(igraph,c(nom_col1,nom_col2))
  m_weight = mean(E(graph)$weight)
  max_weight = sum(E(graph)$weight)
  if (distance!= "Transfer2"){
    if (components(graph)$no > 1){
      if (mat == ""){
        graph = reconnect_btw(graph)
      }
      else{
        graph = reconnect(graph,matrice = mat)
      }
    }
  }

  else if (distance == "Transfer2"){
    if (components(graph)$no > 1){
      comp <- components(graph)
      clusters <- c()
      for (i in 1:comp$no){
        fact_comp <-
factor(V(graph)$tax[which(components(graph)$membership == i)])
        #=== counter for the clusters that will compose the distance
calculation graph
        if (comp$csize[i] > 5){
          clusters <- append(clusters,i)
        }else if (length(levels(fact_comp)) == 2){
          if (count(fact_comp)[which(count(fact_comp)$x ==
nom_col1),2] > count(fact_comp)[which(count(fact_comp)$x ==
nom_col2),2]){
            clusters <- append(clusters,i)
          }
        }
      }
      select_clust <- induced.subgraph(graph,comp$membership %in%
clusters)
    }
    return(c(estimation_transfer(graph,nom_col1),
estimation_transfer(graph, nom_col2), 2, 2))
  }


  ## Get vertex by colors
  col1 <- nom_col1
  col2 <- nom_col2
  mask.col1 <-which(V(graph)$tax == col1)
  mask.col2 <-which(V(graph)$tax == col2)
  v.mix <- grep(col1_mix,V(graph)$tax)
```

```
  v.col1 <-V(graph)[mask.col1] #all vertices of color 1 and mix
  v.col1 <- append(v.col1,V(graph)[v.mix])
  v.col2 <-V(graph)[mask.col2] #all vertices of color 2


  ## Shortest path search for all pair of same color vertex

  path=distance
  # Color 1
  if(length(v.col1) >=2){
    x.col1 =
shortest_paths_graph(graph,v.col1,v.col2,v.mix,col1,path,opti,m_weig
ht,max_weight)
  }

  v.col1 <-V(graph)[mask.col1]
  v.col2 <- append(v.col2,V(graph)[v.mix2])
  # Color 2
  if(length(v.col2) >=2){
    x.col2 =
shortest_paths_graph(graph,v.col2,v.col1,v.mix2,col2,path,opti,m_wei
ght,max_weight)
  }

  # For when col1 or col2 <= 1
  if(!exists("x.col1")){
    x.col1=c(rep(0,8))
  }
  if(!exists("x.col2")){
    x.col2=c(rep(0,8))
  }

  #the denominator is the same for every measure, because it becomes
a proportion

  #this is because not every node is connected (there might be many
islands in the graph), so we use the number of paths that Spp found
  num_paths =
x.col1[1]+x.col1[2]+x.col1[3]+x.col2[1]+x.col2[2]+x.col2[3]

  if(share_w == 1){
    dpr = (1.0*x.col1[2]+x.col2[2]+x.col1[3]+x.col2[3])/num_paths
  }else if (share_w == 0){
    dpr = (1.0*x.col1[2]+x.col2[2])/num_paths
  }
  dnpr = (1.0*x.col1[4]+x.col2[4])/num_paths
  dl = (1.0*x.col1[6]+x.col2[6])/num_paths
  dlpr = 1.0*(x.col1[7]+x.col2[7])/num_paths

  #Calculation for the transfer distance

  transfer_dir = 1.0*x.col1[2]/(x.col1[1]+x.col1[2])
```

```
  transfer_rev = 1.0*x.col2[2]/(x.col2[1]+x.col2[2])
  stopCluster(cl)

  if(path == "Spp"){
    return(c(Spp=dpr))
  } else if(distance == "Spaths"){
    return(c(Spp=dpr,Spep=dl,Spelp=dlpr,Spinp=dnpr, direct =
transfer_dir, reverse = transfer_rev))
  } else if(distance == "Transfer2"){
    return(c(direct = transfer_dir, reverse = transfer_rev,
abs_transfer1 = number_transfer_1(transfer_dir,length(v.col1)),
abs_transfer2 = number_transfer_1(transfer_rev,length(v.col2))))
  }
}
```

NetUniFrac.R

```
#=========================================================
#= Unifrac distance for similarity sequence networks
#=========================================================

#going on the same basis as the UniFrac definition, where only
unique branches are counted in the distance
#compare the nodes from an edge to determine if they are different
(if so, the branch(edge) is shared)
#' NetUniFrac based on UniFrac for trees
#'
#' Calculates the network version of the UniFrac distance.
#'
#' @param igraph The igraph object
#' @param tax1 The first community
#' @param tax2 The second community
#' @param weight Use weights of the edges if existing
#'
#' @return
#' @export
#'
#' @examples
#' NetUnifrac(net_a,"A","B")

NetUnifrac <- function(igraph, tax1 ="", tax2 ="",weight =TRUE){

  #find the nodes with mixed communities
  col1_mix <- paste(append(tax1,"-"),collapse = "")
  col1_mix2 <- paste(append("-",tax1),collapse = "")

  #find the ones that contain the communities in this iteration
  v.mix <- grep(col1_mix,V(igraph)$tax)
```

```
  v.mix <- append(v.mix,grep(col1_mix2,V(igraph)$tax))
  #change to tax1, so it is not mixed anymore
  V(igraph)$tax[v.mix] = tax1

  #create subgraph that contains the two communities
  igraph2 <- igraph
  if (tax1 != ""){
    igraph2<-subgroup_graph(igraph,c(tax1,tax2))
  }
  igraph_edge <- get.edgelist(igraph2)
  igraph_V <- V(igraph2)
  #which pair of nodes that make an edge have same tax, select only
number of pairs
  selection <- which(igraph_V[igraph_edge[,1]]$tax ==
igraph_V[igraph_edge[,2]]$tax)

  #same for second community
  col2_mix <- paste(append(tax2,"-"),collapse = "")
  col2_mix2 <- paste(append("-",tax2),collapse = "")
  v.mix2 <- grep(col2_mix,V(igraph)$tax)
  v.mix2 <- append(v.mix2,grep(col2_mix2,V(igraph)$tax))
  V(igraph)$tax[v.mix2] = tax2

  if (tax2 != ""){
    igraph2=subgroup_graph(igraph,c(tax1,tax2))
  }
  igraph_edge <- get.edgelist(igraph2)
  igraph_V <- V(igraph2)

  #which pair of nodes that make an edge have same tax, select only
number of pairs
  selection <- append(selection,which(igraph_V[igraph_edge[,1]]$tax
== igraph_V[igraph_edge[,2]]$tax))
  selection = unique(selection)

  if(weight){
    w_selection <- sum(E(igraph2)[selection]$weight)
    distance <- w_selection/sum(E(igraph2)$weight)
  }
  else{
    selection <- length(selection)
    distance <- selection/(length(igraph_edge)/2)
  }
  return (distance)
}
```

Transfer_dist.R

```r
#===== Using alpha that will determine the proportion of shortest
paths as a threshold for transfer
#==== optimized Transfer index with components

#' Algorithm for a fast calculation of the Transfer index (for large
networks)
#'
#' The Transfer index can be used to calculate the proportion of
species/sequences of community X that have been
#' affected by horizontal gene transfers from species/sequences of
community Y. The Transfer index is assymetric and directional.
#' The result is a pairwise matrix reporting the Transfer index for
each pair of species communities.
#'
#' @param graph The igraph object
#'
#' @return
#' @export
#' @examples
#' Transfer(net_a)

Transfer <- function(graph){
  graph <- simplify(graph)
  V(graph)$pred <- "no"
  #prepare all the possible pairs of commmunity distance
  taxlvl <- levels(factor(V(graph)$tax))
  taxlevels <- combn(taxlvl,2)
  all_comp <- c()
  all_csize <- c()
  all_alpha <- rep(0, length(taxlvl))
  names(all_alpha) <- taxlvl
  list_all <- list()
  denom_all <- count(V(graph)$tax)

  #=== Add component in vertice, initialize matrix
  for (i in taxlvl){
    #== create the components
    group = subgroup_graph(graph,i)
    comp = components(group)

    #== calculate the size of alpha, first subtract, then find min()
and the size of comm
    if(comp$no == 1){
      alpha = length(V(graph))/2
    }else{
      comp$diff <- ave(comp$csize[order(-comp$csize)],
FUN=function(x) c(0, diff(x)))
      min_comp <- which(comp$diff == min(comp$diff))
      alpha <- (comp$csize[order(-comp$csize)][min_comp-1] +
comp$csize[order(-comp$csize)][min_comp])/2
      if (length(alpha) > 1){
        all_alpha[i] <- alpha[1]
```

```
      }else{
        all_alpha[i] <- alpha
      }
    }

    #== create a unique name for each component
    name_comp <- names(comp$membership)
    comp$membership = paste0(i,comp$membership)

    #== store the information in lists to make it faster
    V(graph)[name_comp]$comp <- comp$membership
    all_comp <- append(all_comp,paste0(i,1:length(comp$csize)))
    list_all <-
append(list_all,list(paste0(i,1:length(comp$csize))))

    #== save the size for further use
    names(comp$csize) <- paste0(i,1:length(comp$csize))
    all_csize <- append(all_csize,comp$csize)
  }
  names(list_all) <- taxlvl
  mat1 <- matrix(0, length(all_comp),length(all_comp))
  dimnames(mat1) <- list(all_comp,all_comp)
  edges <- as_edgelist(graph)

  #=== each edge determines if components are connected
  nodes <- V(graph)$name
  compon <- V(graph)$comp
  for (i in 1:length(edges[,1])){
    comp1 <- compon[which(nodes == edges[i,1])]
    comp2 <- compon[which(nodes == edges[i,2])]

    if (all_csize[comp1] < all_csize[comp2]){
      mat1[comp1,comp2] <- 1
    }else if (all_csize[comp2] < all_csize[comp1]){
      mat1[comp2,comp1] <- 1
    }else{
      mat1[comp1,comp2] <- 0.5
      mat1[comp2,comp1] <- 0.5
    }
  }
  result <- matrix("-",length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))
  result2 <- matrix(0,length(taxlvl),length(taxlvl),dimnames =
list(taxlvl,taxlvl))

  for (i in 1:(length(taxlevels)/2)){
    col1 = noquote(taxlevels[1,i])
    col2 = noquote(taxlevels[2,i])
    list_col1 <- unlist(list_all[col1])
    list_col2 <- unlist(list_all[col2])
    mat1_col1 <- mat1[list_col1,list_col2,drop=FALSE]
    mat1_col2 <- mat1[list_col2,list_col1,drop=FALSE]
```

```
    denom_T = denom_all[which(denom_all$x==col1),2]
    # count the number of transferred nodes
    a = 0
    for(j in 1:length(mat1_col1[,1])){
      if(sum(mat1_col1[j,]) > 0){
        size = all_csize[row.names(mat1_col1)[j]]

        #alpha is the number of edges that are not connected to a
monochrome path
        #any community smaller than 5 ( or another number) will
automatically considered to be a transfer
        if (size <= all_alpha[col1] | denom_T <= 5){
          a = a + (size*max(mat1_col1[j,]))
          positive = which(V(graph)$comp == row.names(mat1_col1)[j])
          V(graph)$pred[positive] = "yes"
        }
      }
    }

    result[taxlevels[2,i],taxlevels[1,i]] <- (round((1-
a/denom_T),3))

    #=== same operation for the other side
    denom_T2 = denom_all[which(denom_all$x==col2),2]
    a = 0
    for(j in 1:length(mat1_col2[,1])){
      if(sum(mat1_col2[j,]) > 0){
        size2 = all_csize[row.names(mat1_col2)[j]]

        if(size2 <= all_alpha[col2] | denom_T2 <= 5){
          a = a + (size2*max(mat1_col2[j,]))
          positive = which(V(graph)$comp == row.names(mat1_col2)[j])
          V(graph)$pred[positive] = "yes"
        }
      }
    }

    result[taxlevels[1,i],taxlevels[2,i]] <- (round((1-
a/denom_T2),3))
  }

  #return(result)
  return(noquote(V(graph)$pred))
}
```

Reconnect.R

```
#=================================================
#Find clusters, betweenness, and connect the clusters with the
highest betweeness centrality node
#For unconnected subgraphs, since it will alter the shortest path
possibilities

#Requires the igraph package
#=================================================


#' Connects the network using betweenness centrality
#'
#' Creates a connected network using the betweenness centrality
principle.
#'
#' @param graph The disconnected igraph object
#'
#' @return
#' @export
#' @seealso \code{\link{reconnect}}
#' @examples

reconnect_btw <- function(graph){
  #find the clusters in the graph
  cl <- components(graph)
  #associate the nodes to the cluster
  all_cluster = lapply(seq_along(cl$csize), function(x)
V(graph)$name[cl$membership %in% x])

  #store all the maximum betweenness nodes
  max_btw = c()
  #find the nodes with the highest bet. centrality - we will use
those ones to make new edges
  for(clique in all_cluster){
    clust = induced.subgraph(graph,clique)

    V(clust)$btw = betweenness(clust)
    max_node = which(V(clust)$btw == max(V(clust)$btw))
    if(length(max_node >1)){
      max_btw <- append(max_btw,V(clust)[max_node[1]]$name)
    }else{max_btw = append(max_btw,V(clust)[max_node]$name)}
  }

  #also consider the lonely nodes
  deg = which(igraph::degree(graph,V(graph),mode = "all")==0)
  max_btw <- append(max_btw,V(graph)[deg]$name)
  #pair up all the possible combinations of nodes
  pairs_btw <- c()
  if(length(max_btw > 1)){
    pairs_btw <- combn(max_btw,2)
  }
```

```
  #set the length of the edge (will be equal to the sum of the graph
edges weights)
  e_weight = sum(E(graph)$weight)

  for (i in 1:length(pairs_btw)/2){
    graph = add.edges(graph,c(pairs_btw[1,i],pairs_btw[2,i]),weight
= e_weight)
  }
  return(graph)
}

#version de la fonction s'il y a une matrice
#' Connects the network using a similarity matrix
#'
#' Creates a connected network using a similarity matrix between
nodes.
#'
#' @param graph The disconnected igraph object
#' @param matrice The similarity matrix containing pairwise
distances between the nodes
#'
#' @return
#' @export
#' @seealso \code{\link{reconnect_btw}}
#' @examples

reconnect <- function(graph,matrice = ""){
  #find the clusters in the graph
  cl <- components(graph)
  #associate the nodes to the cluster
  all_cluster = lapply(seq_along(cl$csize), function(x)
V(graph)$name[cl$membership %in% x])

  #stocker les paires de cluster
  pairs_mat <- combn(seq(1:length(all_cluster)),2)

  #all_pairs va aller chercher les paires de noeuds par nom, puis
stocker les noeuds de valuer max dans max_mat
  #dans une matrice de similarité
  max_mat <- c()
  for (i in 1:(length(pairs_mat)/2)){
    all_pairs <-
expand.grid(all_cluster[[pairs_mat[1,i]]],all_cluster[[pairs_mat[2,i
]]])
    for(j in 1:length(all_pairs$Var1)){
      all_pairs$Val[j] = mat1[all_pairs[j,1],all_pairs[j,2]]
    }
    N = which(all_pairs$Val== max(all_pairs$Val))[1]
    max_mat <- base::append(max_mat,
as.character(all_pairs$Var1[n]))
    max_mat <- base::append(max_mat,
as.character(all_pairs$Var2[n]))
```

```
  }
  if (sum(E(graph)$weight) == 0){
    e_weight = 1
  }else{
    e_weight = sum(E(graph)$weight)
  }
  graph = add.edges(graph,max_mat,weight = e_weight)
  return(graph)
}
```

Sub_groups.R

```
#=======================================
#= creates subgraph or 2 groups from a complex graph
#=======================================
#assuming that we already have an igragh object with group
information
#' Creates a subgraph based on chosen species communities
#'
#' Creates another igraph object containing only the nodes from some
chosen species
#' communities (two or more), and the edges connecting them.
#'
#' @param graph The igraph object
#' @param groups The communities that should be isolated, indicated
as a list of two items
#'
#' @return
#' @export
#'
#' @examples
#' CAT_env1 <- set_color(subgroup_graph(CAT_env, c("host",
"ubiquitous")))
#' plot(CAT_env1)

subgroup_graph <- function(graph, groups){
  subvertice <- V(graph)[V(graph)$tax %in% groups]
  graph <- induced.subgraph(graph, subvertice)
  return (graph)
}
```

Set_colors.R

```
#=====================================================================
#= Function to set the vertex colors of a graph for plotting a graph
```

```
#=
#= Input: an igraph with colors as tax attribute (V(graph)$tax)
#= Output: a modified igraph object
#=
#= Requires SDDE and igraph packages
#=
#= Ex: graph<-set_color(graph)
#=     plot(graph)
#=========================================================================

#' Sets colors for the communities in the network
#'
#' Visualizes the network by adding different colors to different
species communities, when plotting
#' either with the R color list or a provided color list.
#'
#' @param graph The igraph object used to define colors. The nodes
accessed with V() should have
#' a $tax attribute which links them to a communit, so a different
color can be assigned to each community.
#' @param colors_list Optional: the colors can be defined in a list
with the community names defined with names().
#' Otherwise, the function will return the numbers already
associated to R colors.
#'
#' @return
#' @export
#'
#' @examples
#' foo_CAT_env <- c("red","green","blue","grey")
#' names(foo_CAT_env) = c("host","ubiquitous","water","unknown")
#' CAT_env <- set_color(CAT_env, foo_CAT_env)
#' plot(CAT_env)

set_color<- function(graph, colors_list=""){
  if (colors_list != ""){
    for (i in 1:length(V(graph))){
      V(graph)[i]$color = colors_list[V(graph)[i]$tax]
    }
  }
  else{
    V(graph)$color = V(graph)$tax
    color <-levels(as.factor(V(graph)$tax))

    for (i in 1:length(color)){
      V(graph)$color = replace(V(graph)$color,
which(V(graph)$color==color[i]), i)
    }
  }
  return(graph)
}
```

APPENDICE B

LES ARTICLES PUBLIÉS LORS DE CE PROJET DE MAÎTRISE

1. Xing, H., Hour, T., Kembel, S. W. and Makarenkov, V. (2018) "Quelques mesures pertinentes pour calculer la distance entre les communautés d'espèces dans des réseaux de similarité de séquences". Société francophone de la classification, Paris, France, pp 21-26
2. Xing, H., Kembel, S. W. and Makarenkov, V. (2020) "Transfer index, NetUniFrac and some useful shortest path-based distances for community analysis in sequence similarity networks". Bioinformatics, btaa043

Quelques mesures pertinentes pour calculer la distance entre les communautés d'espèces dans des réseaux de similarité de séquences

Henry Xing*, Tissicca Hour*, Steven W. Kembel** et Vladimir Makarenkov*

*Département d'informatique
**Département des sciences biologiques
Université du Québec à Montréal, Case postale 8888, Succursale Centre-ville
Montréal (Québec) H3C 3P8 Canada

xing.henry@courrier.uqam.ca, hour.tiss@gmail.com, kembel.steven_w@uqam.ca,
makarenkov.vladimir@uqam.ca

Résumé. L'utilisation de réseaux de similarité de séquences pour analyser des communautés d'espèces est souvent préférable à l'utilisation d'arbres phylogénétiques (arbres additifs ou *X*-arbres). À la différence des arbres qui montrent un lien de parenté unique, passant par l'ancêtre commun le plus proche, entre les espèces, les réseaux de similarité de séquences peuvent représenter des mécanismes évolutifs complexes, tels que le transfert horizontal de gènes et la recombinaison homologue au niveau génomique ou génique. Dans cet article, nous proposons cinq nouvelles mesures de distances entre les différentes communautés d'espèces présentes dans un réseau de similarité donné, dont une adaptation de la distance *UniFrac*, originellement définie pour les arbres phylogénétiques. Les quatre autres distances sont basées sur le calcul des plus courts chemins entres les nœuds du réseau. Leur capacité de discrimination est étudiée.

1. Introduction

Les arbres phylogénétiques (arbres additifs ou *X*-arbres) trouvent de nombreuses applications dans les domaines de la biologie et de l'écologie. Différentes mesures de distance entre les arbres, telles que la distance de Robinson et Foulds ou la distance des quartets (Felsenstein, 2004), et de biodiversité entre les communautés d'espèces présentes dans les arbres, telles que la distance *UniFrac* et ses variantes (Lozupone et al., 2011), ont été proposées et amplement utilisées. Cependant, il est bien connu que les arbres ont leurs limites quant à la représentation de l'évolution ainsi que de la biodiversité des espèces (Huson et Bryant, 2005). Les réseaux phylogénétiques et les réseaux de similarité de séquences sont souvent mieux adaptés que les arbres pour représenter des phénomènes évolutifs complexes, tels que l'hybridation, l'endosymbiose, la recombinaison ou le transfert horizontal de gènes.

Dans un réseau de similarité de séquences, un nœud représente une séquence d'un gène ou d'un génome d'intérêt. Deux nœuds sont connectés par une arête s'ils montrent une similarité supérieure à un seuil donné pour la totalité ou pour une partie de leur séquence. Les réseaux de similarité sont complémentaires aux arbres et réseaux phylogénétiques. Ils n'offrent pas seulement un moyen différent de représenter des similarités entre les séquences, mais proposent également un cadre de recherche souple d'analyse de données métagénomiques (Bapteste et al., 2013).

Des mesures de biodiversité équivalentes à la métrique *UniFrac*, qui permet d'estimer la distance entre les communautés d'espèces dans un arbre phylogénétique, ont récemment été proposées pour des split graphes (Parks et Beiko, 2012) et des réseaux phylogénétiques généraux (Wicke et Fischer, 2018). Dans cet article, nous proposons des nouvelles mesures de distances, dont l'extension directe de la distance *UniFrac*, qui peuvent être utilisées pour estimer les distances entre différentes communautés d'espèces (de gènes, d'individus ou d'objets) associées à

des nœuds d'un réseau de similarité. À la différence de la distance *NetFrac*, qui est basée sur le calcul des proportions des motifs spécifiques impliquant les espèces de la même communauté (Bapteste et al., 2012), la plupart de nos distances utilisent le concept du plus court chemin entre les espèces de la même communauté.

2. Méthodes

Dans cette section nous donnons les définitions des cinq nouvelles distances entre les communautés d'espèces présentes dans un réseau de similarité de séquences. Dans les arbres phylogénétiques, il existe un chemin évolutif unique entre deux noeuds donnés. Les réseaux de similarité montrent plus de flexibilité quant au nombre de chemins évolutifs possibles, c'est pourquoi nous utilisons les plus courts chemins multiples pour définir nos distances.

2.1 Définitions de nouvelles distances dans un réseau de similarité de séquences

Une *arête d'un réseau de similarité est dite monochrome* si les espèces associées à ses deux extrémités appartiennent à la même communauté. Un *chemin d'un réseau de similarité est dit monochrome* si toutes ses arêtes sont monochromes.

Distance *SsnUniFrac* (*Sequence similarity network UniFrac*): Il s'agit de la généralisation directe de la distance *UniFrac* (Lozupone et al., 2011) définie pour les arbres phylogénétiques :

$$SsnUniFrac = \frac{\text{Longueur totale d'arêtes monochromes du réseau}}{\text{Longueur totale d'arêtes du réseau}}. \quad (1)$$

Un des avantages de cette distance est qu'elle peut être calculée dans un temps linéaire par rapport au nombre d'arêtes du réseau. Quand toutes les longueurs d'arêtes du réseau sont égales, cette distance est équivalente à la distance *NetFrac* basée sur les motifs de longueur 2 (Bapteste et al., 2012).

Distance *Spp*: La distance *Spp* (*Shortest path proportion*) est la proportion des plus courts chemins qui sont monochromes dans un réseau de similarité de séquences :

$$Spp = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \quad (2)$$

où N_X est le nombre de nœuds dans la communauté X, N_Y est le nombre de nœuds dans la communauté Y, $K(ij)$ est le nombre de plus courts chemins entre les nœuds i et j appartenant à la même communauté. La variable binaire $\sigma_{ij}^{k} = 1$ si le chemin k entre les nœuds i et j est monochrome, et $\sigma_{ij}^{k} = 0$ sinon. Cette distance peut être utile pour mettre en évidence des liens entre deux communautés d'espèces qui sont dus au transfert horizontal de gènes.

Distance *Spep*: La distance *Spep* (*Shortest path edge proportion*) est basée sur le calcul du nombre d'arêtes monochromes dans tous les plus courts chemins entre les espèces d'une même communauté :

$$Spep = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \quad (3)$$

où E_{ij}^{k} est la proportion d'arêtes monochromes dans le chemin *k* entre les nœuds *i* et *j* appartenant à la même communauté.

Distance *Spelp*: La distance *Spelp* (*Shortest path edge length proportion*) est basée sur le calcul des longueurs d'arêtes dans tous les plus courts chemins monochromes :

$$Spelp = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \quad (4)$$

où EL_{ij}^{k} est la proportion des longueurs d'arêtes monochromes dans le chemin *k* entre les nœuds *i* et *j* appartenant à la même communauté.

Distance *Spinp*: La distance *Spinp* (*Shortest path internal node proportion*) est la proportion des noeuds internes d'un plus court chemin appartenant à la même communauté d'espèces que ses deux extrémités, qui est calculée en considérant tous les chemins les plus courts du réseau de similarité de séquences donné. Dans le cas où le plus court chemin ne contient pas de noeuds internes (il est donc de longueur 2), on ajoute 1 au dénominateur et au numérateur de la fraction qui définit cette distance :

$$Spinp = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \quad (5)$$

où N_{ij}^{k} est la proportion des nœuds internes de la même communauté (que les nœuds *i* et *j*) dans le chemin *k* entre *i* et *j* (qui appartiennent à la même communauté d'espèces).

Cette distance peut être rendue sensible aux poids associés aux noeuds du réseau et prendre ainsi en compte l'abondance ou la pertinence statistique des espèces associées à ces noeuds.

2.2 Un exemple de calcul des nouvelles distances

Pour illustrer le calcul des distances définies par les Équations (1-5), nous proposons un exemple d'un réseau de similarité simple présenté sur la Figure 1.

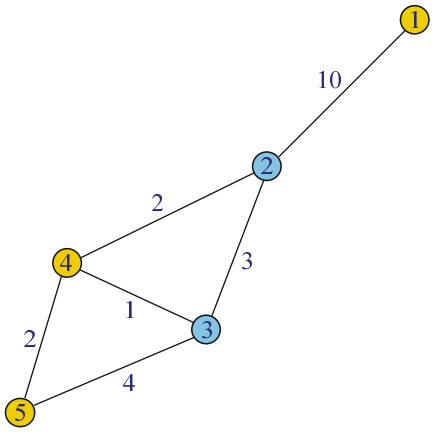


Fig. 1 : *Exemple d'un réseau de similarité à cinq nœuds (trois nœuds jaunes et deux nœuds bleus, représentant deux communautés d'espèces différentes) et à six arêtes dont la longueur varie entre 1 et 10.*

Dans ce réseau, contenant les représentants de deux communautés différentes (3 nœuds jaunes et 2 nœuds bleus), on retrouve 2 chemins les plus courts qui sont monochromes: un chemin entre 2 et 3 et un autre chemin entre 4 et 5. Il y a deux chemins les plus courts chemins (de longueur 3) entre les nœuds 2 et 3, dont seulement un est monochrome. La Table 1 ci-dessous indique les résultats des calculs des distances définies dans notre étude.

Distance	Numérateur	Dénominateur	Valeur
SsnUniFrac	2+3	1+2+2+3+4+10	0,23
Spp	0+0+0,5×(1+0)+1	0,5×(2+6)	0,375
Spep	0+0,33+0,5×(1+0)+1	0,5×(2+6)	0,46
Spelp	0+0,14+0,5×(1+0)+1	0,5×(2+6)	0,41
Spinp	0+0,5+0,5×(1+0)+1	0,5×(2+6)	0,50

Tab. 1 : *Le calcul des cinq distances définies par les Équations* (*1-5*) *pour le réseau de similarité de séquences présenté sur la Figure 1.*

Dans le cas du réseau de la Figure 1, les valeurs des distances *Spp*, *Spep*, *Spelp* et *Spinp* sont assez proches, bien que ce ne soit pas toujours le cas dans tous les réseaux.

3. Résultats

Des simulations ont été effectuées sur les données d'*Esophagus* (Pei et al., 2004), qui est un jeu de données réelles disponible sur le site du logiciel *mothur* (Schloss et al., 2009, http://www.mothur.org/wiki/Analysis_examples), un logiciel d'analyses des communautés microbiennes bien connu en écologie. *Esophagus* est un jeu de données du microbiome de l'œsophage distal chez l'humain. La stratégie pour créer les réseaux de similarité de séquences, tel que proposé par Bapteste et al. (2012), consiste à effectuer un BLAST local des séquences entre elles. On génère par la suite la liste d'arêtes du réseau, en créant des liens entre les séquences (nœuds du réseau) dont le degré de similarité atteint un seuil désiré (97% pour ce jeu de données). Les données disponibles contiennent les échantillons de microbiomes prélevés chez 3 patients (indiqués ici par B, C et D). En plus de nos cinq distances, nous avons calculé les valeurs de la distance *UniFrac* classique en utilisant le logiciel *mothur*. La Table 2 montre les principales caractéristiques du réseau de similarité de séquences construit. Le nombre de nœuds dans les 3 sous-graphes analysés est similaire, mais le nombre d'arêtes (ou de liens entre les communautés) diffère largement.

Sous-graphe	Nœuds	Arêtes	Total de nœuds	Total d'arêtes
B-C	454	8443	684	25061
C-D	486	12604		
B-D	428	16310		

Tab. 2 : *Les principales caractéristiques du réseau de similarité de séquences construit pour les données d'Esophagus (construit en utilisant le seuil de similarité BLAST de 97%).*

Sous-graphe	*SsnUniFrac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Unweighted UniFrac*[1]	*p*-value *UniFrac*[2]
B-C	0,61	0,52	0,65	0,65	0,71	0,64	<0,001
C-D	0,78	0,47	0,58	0,58	0,62	0,68	<0,001
B-D	0,59	0,50	0,62	0,62	0,67	0,62	0,129

Tab. 3 : *Les valeurs des distances entre les communautés bactériennes B-C, C-D et B-D, retrouvées pour le réseau d'Esophagus.*

Les résultats présentés dans la Table 3 indiquent que selon les distances *Spp*, *Spep*, *Spelp* et *Spinp*, les microbiomes des patients C et D sont les plus similaires entre eux, et ceux des patients B et C sont les plus distincts. Les valeurs des distances *Spep* et *Spelp* sont les mêmes, car le réseau n'a pas de différents poids sur les arêtes. La différence entre les valeurs est beaucoup plus marquée pour la distance *SsnUniFrac*. Selon les distances *UniFrac* et *SsnUniFrac*, dont les valeurs corrèlent entre elles, les microbiomes les plus distincts sont ceux des patients C et D. On peut aussi constater que la nouvelle distance *SsnUniFrac* discrimine mieux les microbiomes que la distance *UniFrac* classique.

4. Discussion

L'utilisation des plus courts chemins dans des réseaux de similarité de séquences semble être pertinente pour mesurer la distance entre les différentes communautés d'espèces. Cependant, le calcul des distances basées sur les plus courts chemins est beaucoup plus lent que le calcul des distances *UniFrac* et *SsnUniFrac*. Par exemple, le calcul des distances *Spp*, *Spep*, *Spelp* et *Spinp* pour le réseau de similarité d'*Esophagus* (Pei et al., 2004) a nécessité 3 heures de calcul sur 7 cœurs parallèles. Notre programme écrit en langage R n'a pas été encore optimisé. Nous sommes en train de développer une version optimisée du programme en R comprenant des appels des fonctions de calcul des plus courts chemins écrites en langage C. D'autre part, nous ajouterons à notre logiciel les options d'analyse supplémentaires, telles que la possibilité d'appartenance d'un nœud du réseau à plusieurs communautés d'espèces et l'ajout des poids sur les nœuds du réseau.

La question qui reste ouverte et de savoir si les plus courts chemins discriminent mieux les populations denses ou concentrées que les autres mesures. Nous proposons d'investiguer dans le futur s'il existe un effet de biais sphérique, *i.e.*, des populations avec des graphes denses par rapport à des populations avec des graphes en peigne, dans ces mesures. De plus, une comparaison avec les mesures de centralité classiques, utilisées souvent pour l'analyse de réseaux sociaux, sera effectuée.

Références

Bapteste, E., Lopez, P., Bouchard, F., Baquero, F., McInerney, J. O., & Burian, R. M. (2012). Evolutionary analyses of non-genealogical bonds produced by introgressive descent. *Proceedings of the National Academy of Sciences*, 109, 18266-18272.

Bapteste, E. et al. (2013). Networks: expanding evolutionary thinking. *Trends in Genetics*, 29, 439-441.

Felsenstein, J.(2004). *Inferring Phylogenies*, Vol. 2, Sunderland, MA: Sinauer associates.

[1] La distance *Unweighted UniFrac* a été calculée sur l'arbre phylogénétique fourni par *mothur*.

[2] La *p*-value obtenue pour le test de significativité de *Unweighted UniFrac*. 1000 permutations ont été effectuées. La *p*-value globale pour les 3 communautés est <0.001, ce qui indique qu'au moins un des 3 patients possède une communauté avec une structure différente. Cela est nécessaire pour que le score de *p*-value des communautés prises deux à deux soit pertinent. Pour tenir compte des analyses multiples, le seuil de significativité pour la *p*-value a été fixé à 0.05/3 □0.0166.

Huson, D. H., & Bryant, D. (2005). Application of phylogenetic networks in evolutionary studies. *Molecular Biology and Evolution*, 23, 254-267.

Lozupone, C., Lladser, M. E., Knights, D., Stombaugh, J., & Knight, R. (2011). *UniFrac*: an effective distance metric for microbial community comparison. *The ISME Journal*, 5, 169–172.

Newman, M. (2010). *Networks: an introduction*. Oxford University Press.

Parks, D. H., & Beiko, R. G. (2012). Measuring community similarity with phylogenetic networks. *Molecular Biology and Evolution*, 29, 3947-3958.

Pei, Z., Bini, E. J., Yang, L., Zhou, M., Francois, F., & Blaser, M. J. (2004). Bacterial biota in the human distal esophagus. *Proceedings of the National Academy of Sciences*, 101, 4250-4255.

Schloss, Patrick D., Westcott, Sarah L., Ryabin, Thomas, *et al* (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75, 23, 7537-7541.

Wicke, K., & Fischer, M. (2018). Phylogenetic diversity and biodiversity indices on phylogenetic networks. *Mathematical Biosciences*, 298, 80-90.

Summary

The use of sequence similarity networks to analyze species communities is often more relevant than the use of traditional phylogenetic trees (additive trees or *X*-trees). In this paper, we propose several new distances between different species communities present in a given similarity network, including an adaptation of the *UniFrac* distance, originally defined for phylogenetic trees only. The other four distances are based on the calculation of the shortest paths between the nodes of a given similarity network.

Bioinformatics, 2020, btaa043
doi: https://doi.org/10.1093/bioinformatics/btaa043
Advance Access Publication Date: January 23, 2020
Original Paper

Phylogenetics

Transfer index, NetUniFrac and some useful shortest path-based distances for community analysis in sequence similarity networks

Xing, Henry[1], Kembel, Steven W.[2] and Makarenkov, Vladimir[1*]

[1]Department of Computer Sciences, Université du Québec à Montréal, Montreal, Canada.

[2]Department of Biology, Université du Québec à Montréal, Montreal, Canada.

*To whom correspondence should be addressed.

Abstract

Motivation: Phylogenetic trees and the methods for their analysis have played a key role in many evolutionary, ecological and bioinformatics studies. Alternatively, phylogenetic networks have been widely used to analyze and represent complex reticulate evolutionary processes that cannot be adequately studied using traditional phylogenetic methods. These processes include, among others, hybridization, horizontal gene transfer and genetic recombination. Nowadays, sequence-similarity and genome-similarity networks have become an efficient tool for community analysis of large molecular datasets in comparative studies. These networks can be used for tackling a variety of complex evolutionary problems such as the identification of horizontal gene transfer events, the recovery of mosaic genes and genomes, and the study of holobionts.

Results: The shortest path in a phylogenetic tree is used to estimate evolutionary distances between species. We show how the shortest path concept can be extended to sequence similarity networks by defining five new distances, NetUniFrac, Spp, Spep, Spelp and Spinp, and the Transfer index, between species communities present in the network. These new distances can be seen as network analogues of the traditional UniFrac distance used to assess dissimilarity between species communities in a phylogenetic tree, while the Transfer index is intended for estimating the rate and direction of gene transfers, or species dispersal, between different phylogenetic, or ecological, species communities. Moreover, NetUniFrac and the Transfer index can be computed in linear time with respect to the number of edges in the network. We show how these new measures can be used to analyze microbiota and antibiotic resistance gene similarity networks.

Availability and implementation: Our NetFrac program, implemented in R and C, along with its source code, is freely available on Github at the following URL address: https://github.com/XPHenry/Netfrac.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Phylogenetic trees have numerous applications in molecular biology, ecology and bioinformatics. Different distances between phylogenies, such as the Robinson and Foulds distance (Robinson and Foulds, 1981) or the quartet distance (Felsenstein, 2004), and of biodiversity between species communities, such as the *UniFrac* distance and some of its variants (Lozupone *et al*., 2011), have been proposed and widely used. However, it is also well known that trees have their limitations when reticulate evolutionary processes, including horizontal gene transfer, recombination, endosymbiosis and hybridization, need to be modeled and visualized (Huson and Bryant, 2005). Network structures, including sequence similarity networks, are much better adapted than trees to represent these complex evolution phenomena (Bapteste *et al*., 2012).

In a sequence similarity network, a node represents a certain gene or a genome sequence. Sequence similarity can be defined in various ways such as BLAST sequence similarity or different evolutionary models of mutations (e.g., JTT or GTR model). Two network nodes are connected by an edge if they show a similarity higher than a certain threshold for the whole sequence or for some of its parts. Sequence similarity networks can be viewed as complementary to phylogenetic trees and phylogenetic networks (e.g., split graphs or horizontal gene transfer networks). Not only they offer a different way of representing similarities between species/sequences, but also provide a number of tools for the analysis of metagenomic data (Bapteste *et al*., 2013, Jachiet *et al*., 2013, Pathmanathan *et al.,* 2018). Biodiversity measures equivalent to the *UniFrac* distance (Lozupone *et al*., 2011), which allows estimating a phylogenetic distance between different species communities, have been proposed for split graphs (Parks and Beiko, 2012), and more recently, for other types of phylogenetic networks (Wicke and Fischer, 2018).

When a similarity network is analyzed, the concepts of betweenness, centrality, connectivity, modularity and shortest path are often used to

assess the importance or compare some of its parts (Girvan and Newman, 2002). In this paper, we introduce six new network measures, including a direct extension of *UniFrac* to sequence similarity networks, which can be used to assess dissimilarity between different species communities present in a given network. Unlike the *Motif* distance (Bapteste *et al*., 2012), which is based on the computation of the proportion of specific motifs (patterns or sub-graphs) in the network, all of our distances and indices, except *NetUniFrac*, use the concept of the shortest path between the species of the same community. Although we are using gene networks as examples, our new measures may have different possible meanings in different similarity networks. For example, in the case of an ecological species network our distances could account for some ecological phenomena such as dispersal of extant species between communities. Thus, the interpretation of the proposed distances and indices depends on the type of network being analyzed.

2 Methods

In this section, we define new distances and indices between the species communities present in a given SSN (sequence similarity network). The first of these distances, *NetUniFrac*, is a direct extension to SSNs of the traditional *Unifrac* distance (Lozupone *et al*., 2011) defined in the context of phylogenetic trees. The four next distances, *Spp*, *Spep*, *Spelp* and *Spinp*, are symmetric pairwise community distances based on the computation of the shortest paths between the nodes of the network. Finally, we also define two versions of the *Transfer index*, which is an asymmetric directional measure that can be used to estimate the direction and rate of the horizontal transfer of genetic material between the species of different communities present in a given SSN. Traditionally, a unique shortest path between two nodes in a phylogenetic tree (i.e., tree leaves are usually labeled by species names, and internal tree nodes are associated with their ancestors) is used to assess the evolutionary distance between them. SSNs may contain several shortest paths between two given nodes (i.e., these nodes are usually labeled by the corresponding species names). In both, a phylogenetic tree and an SSN, the similarity/dissimilarity between gene or genome sequences of related species is used to define their structure. While the shortest path concept in an SSN does not have the same evolutionary meaning as in a phylogenetic tree, it can still be exploited to discover evolutionary relationships between species communities using the new measures described in this work.

2.1 Definition of the new distances and the transfer index

Sequence similarity network is a graph whose nodes, *N*, represent nucleotide or amino acid sequences of certain species, and edges, *E*, are used to connect the nodes corresponding to the sequences with the highest similarities. We assume that each node of the network belongs to one species communities, but the case when a node belong to several species communities can also be considered. An edge connects two nodes of the networks if similarity between the sequences associated with them is higher than or equal to a predefined threshold *h*. If the network edges have weights, which are usually proportional to the distances (differences or dissimilarities) between the species connected, then the network is called a *weighted SSN*, otherwise the SSN is *unweighted*. An *edge* of an SSN is said to be *monochromatic* if the species associated to its extremities belong to the same community. Similarly, a *path* of an SSN is said to be *monochromatic* if all of its edges are monochromatic. Phylogenetic trees and SSNs bear a certain relationship since they deal with the same objects, i.e., homologous sequences, and since they are both graphs. However, phylogenetic trees are used to represent the evolution of these sequences, whereas SSNs are used to depict similarities/dissimilarities between them. In the opposite of trees, SSNs are unrooted graphs which can be unconnected. Moreover, multiple shortest paths connecting two given nodes may exist in an SSN. Turning a tree into an SSN is straightforward, but the opposite is not true (Atkinson *et al*., 2009).

Below we define some new network distances and indices between species communities. A community can consist of a subset of network nodes representing organisms belonging to a given clade (e.g., see Taxonomic network in Section 3.3), or nodes representing organisms sharing the same habitat (e.g., see Environmental network in Section 3.3), or any other subset of nodes representing organisms with a certain common property. It is important to note that our community distances and indices are calculated pairwise, i.e., when the network contains more than two communities, the distances are calculated pairwise by considering a sub-graph that includes only the nodes associated with the species of the two communities of interest. The general idea behind our shortest path-based SSN distances and indices, defined in Equations 3 to 9, is as follows: If a node representing a species of community *Y* is located on the shortest path, or one of the shortest paths, between two nodes representing species of community *X*, it can means that:

1) *In a gene SSN*: The gene evolution of one or of both of these species of *X*, or of some of their close ancestors, has been affected by horizontal gene transfer coming from this species of *Y*, or from some of its close ancestors (in this case, the network cluster, i.e., connected component, to which belongs this species of *Y* is usually much larger than at least one of the two separated clusters to which belong these two species of *X*). This corresponds to the case of traditional gene transfer that assumes that the transferred gene either supplants the orthologous gene in the host genome or is added to it (when it was originally absent in it);

2) *In a gene SSN*: The gene sequence of this species of *Y*, or of some of its close ancestors, is a mosaic gene that was created through intragenic recombination of the gene sequences of these two species of *X*, or of some of their close ancestors (in this case, the network cluster to which belongs this species of *Y* usually consists of either a single species or a small number of species);

3) *In a genome SSN*: This species of *Y*, or some of its close ancestors, is a hybrid of these two species of *X*, or of some of their close ancestors (in this case, the network cluster to which belongs this species of *Y* usually consists of either a single species or a small number of species).

***NetUniFrac* distance** is a direct analogue of the classical *UniFrac* distance (Lozupone *et al*., 2011) defined for phylogenetic trees:

$$NetUniFrac = \frac{\text{Total length (number) of monochromatic edges in SSN}}{\text{Total length (number) of all edges in SSN}} \quad (1)$$

One of the advantages of this distance is that it can be calculated in linear time with respect to the number of the edges in the network. It is worth noting that *NetUniFrac* can be computed either for the whole SSN or for a given pair of species communities present in the network (as the classical *UniFrac* distance). This distance reflects the homogeneity of the whole SSN and of its pairwise community sub-networks. The *NetUniFrac* distance can be seen as a simpler variant of the *Assortativity* measure (Newman, 2003), which also represents a fraction of edges that connect the nodes belonging to the same species community. For an undirected similarity network:

$Assortativity = (f_M \bullet \sum_{C_i} f^2_{NM}(C_i))/(1 \bullet \sum_{C_i} f^2_{NM}(C_i))$, where f_M is the fraction of monochromatic edges in the network and $f_{NM}(C_i)$ is the fraction of non monochromatic edges in the network that connect a node belonging to community C_i to a node belonging to a different community. For instance, when the number of non monochromatic network edges tends to 0, both *Assortativity* and *NetUniFrac* tend to 1.

When all edge lengths are equal or when the network is unweighted, *NetUniFrac* becomes equivalent to the *Motif* distance, based on the motifs of size 2 (i.e., edges) defined in Bapteste *et al.* (2012). The *Motif* distance is defined as a proportion of specific sub-graph patterns that belong to a given species community:

$$Motif = \frac{\text{Number of specific motifs belonging to the community}}{\text{Total number of specific motifs}}. \qquad (2)$$

Usually, the motif size is limited to 2 or 3 because of the computational time required to identify all the motifs of a higher size.

Below, we present the complete versions of the distance formulas, assuming that several shortest paths between a given pair of nodes (*i*,*j*) may exist in a given SSN.

***Spp* distance**: The *Shortest path proportion distance* is the weighted proportion of the monochromatic shortest paths in the network:

$$Spp(X,Y) = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \qquad (3)$$

where N_X is the number of nodes representing species that belong to community *X*, N_Y is the number of nodes representing species that belong to community *Y*, $K(ij)$ is the number of shortest paths between the nodes *i* and *j* belonging to the same community. The binary variable $\sigma_{ij}^{k} = 1$ if the *k*th shortest path between the nodes *i* and *j* is monochromatic; otherwise $\sigma_{ij}^{k} = 0$. The quantity $\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))$ appearing in the denominator of (3) is the maximum number of the unique monochromatic shortest paths in an SSN including the nodes of *X* and *Y*. The use of σ_{ij}^{k} in the numerator of (3) allows us to weight the contribution of the shortest path *k* between the nodes *i* and *j* to the total count of monochromatic shortest paths in the network. In a horizontal gene transfer SSN, this distance can be used to estimate the proportion of evolutionary connections between species communities that are due to horizontal gene transfer events, while in a hybridization SSN, this distance can be used to identify potential hybrids and their parents.

***Spep* distance**: The *Shortest path edge proportion distance* is based on the number of monochromatic edges in all shortest paths between the nodes representing species of the same community:

$$Spep(X,Y) = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} E_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \qquad (4)$$

where E_{ij}^{k} is the proportion of monochromatic edges in the *k*th shortest path between the nodes *i* and *j* (belonging to the same community). In the case of gene transfer or hybridization SSNs, when the number of recipients of genetic material is known, this distance can be used to assess whether the recipient species have been affected by recent transfers of genetic material, as shown in Figs. 1b and d, when all of the transfer recipients are well-separated from each other (see single blue nodes on the right part of the figure), or by more ancient transfers, when the transfer recipients form large interconnected clusters.

***Spelp* distance**: The *Shortest path edge length proportion distance* is based on the total length of the monochromatic edges in all shortest paths between the nodes representing species of the same community:

$$Spelp(X,Y) = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} EL_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \qquad (5)$$

where EL_{ij}^{k} is the proportion of the length of the monochromatic edges in the *k*th shortest path between the nodes *i* and *j*. This distance is very close to *Spep*. The main difference is that *Spelp* gives a larger weight to longer network edges, i.e., edges accounting for longer evolutionary time between the species they connect.

***Spinp* distance**: The *Shortest path internal node proportion distance* is calculated as a weighted proportion of internal nodes representing species that belong to the same community that their extremities, found in all shortest paths between them:

$$Spinp(X,Y) = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k} + \sum_{i,j \bullet Y} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} N_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1) + N_Y(N_Y \bullet 1))}, \qquad (6)$$

where N_{ij}^{k} is the proportion of internal nodes representing species of the same community to which belong the species corresponding to the nodes *i* and *j*, in the *k*th shortest path between *i* and *j*. This distance is sensitive to the weights of the network nodes (if any), taking into account the abundance or statistical significance of the associated species.

Transfer index: This index, which is also based on a shortest path computation, is an asymmetric and directional dissimilarity measure. It can be used to estimate the number of species of community *X* that have been affected by horizontal gene transfers from species of community *Y*, or to account for dispersal of species between those communities. It is worth noting that this estimation is particularly well adapted to the case of recent gene transfers (see Fig. 1). Let us first define the variable $p_i(\bullet)$ that equals 0 if the node *i* representing a species of *X* have received genetic material from some nodes representing species of *Y*, and equals 1, otherwise:

$$p_i(\bullet(Y,X)) = \begin{cases} 1, \text{ if } \sum_{j \bullet X(j \bullet i)} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k} \bullet \bullet (N_X - 1) > 0, \\ 0, \text{ otherwise,} \end{cases} \qquad (7)$$

where the binary variable $\sigma_{ij}^{k} = 1$ if the *k*th shortest path between *i* and *j* is monochromatic, and $\sigma_{ij}^{k} = 0$, otherwise, $K(ij)$ is the number of shortest paths between the nodes *i* and *j*, and $\bullet(Y,X)$ ($0 \bullet \bullet \bullet 1$) is the selected threshold, accounting for the proportion of monochromatic paths, used to decide whether a node representing a species of *X* has been affected by a transfer from *Y* or not. Then, the *transfer index* from community *Y* to community *X*, depending on the threshold $\bullet$, can be defined as follows:

$$Tr(Y \bullet X, \bullet(Y,X)) = \frac{\sum_{i \bullet X} p_i(\bullet(Y,X))}{N_X}. \qquad (8)$$

The transfer index from *X* to *Y*, $Tr(X \bullet Y, \bullet)$, is defined in a similar manner. Obviously, $Tr(Y \bullet X, \bullet)$ is not necessarily equal to $Tr(X \bullet Y, \bullet)$. The key question here is how to select an appropriate value of the transfer threshold $\bullet(Y,X)$. Let us consider the sorted array of the following quantities: $\frac{1}{N_X \bullet 1} \sum_{j \bullet X(j \bullet i)} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}$, computed for each node $i \bullet X$, where $i = 1, \ldots, N_X$. An intuitive choice here consists in locating $\bullet(Y,X)$ in the middle of the largest interval between two consecutive values of this sorted array. As was confirmed by our simulations (see Section 3.1), this is usually a good choice for the selection of the $\bullet$ threshold. Clearly, the value of $\bullet$ should vary depending on both, the pair of species communities, (*X*,*Y*), being analyzed and the transfer direction.

Figure 1 presents an example of calculation of the transfer index. The sequence similarity network shown here includes *n* nodes representing species of the blue (*B*) community and *n* nodes representing species of

the yellow (Y) community. The value of $\bullet$ is selected as explained above. In the case A, the blue and yellow species sub-networks are connected by a unique edge which is not related to horizontal gene transfer. Thus, in this case, $Tr(B \bullet \ \ Y, \bullet = 0.5) = Tr(B \bullet \ \ Y, \bullet = 0.5) = 1$. In the case B, one species of the blue community is affected by transfer from the yellow community and the index values are respectively as follows: $Tr(B \bullet \ \ Y, \bullet = 0.5) = 1$ and $Tr(Y \bullet \ \ B, \bullet = \frac{n \bullet 2}{2(n \bullet 1)}) = \frac{n\text{-}1}{n}$. When half of the species of B are affected by transfers from Y, the index values will be as follows: $Tr(B \bullet \ \ Y, \bullet = 0.5) = 1$ and $Tr(Y \bullet \ \ B, \bullet = \frac{n \bullet 2}{4(n \bullet 1)}) = 1/2$. When all species of B are affected by transfers from Y, the index values will be as follows: $Tr(B \bullet \ \ Y, \bullet = 0.5) = 1$ and $Tr(Y \bullet \ \ B, \bullet = 0) = 0$.

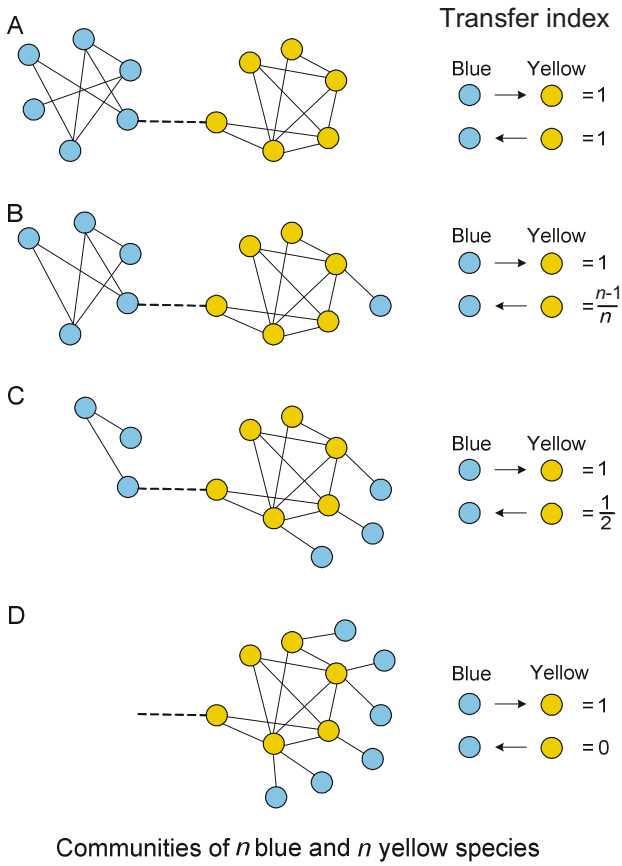


Fig. 1. Example of calculation of the transfer index for an SSN including n nodes representing species of the blue community and n nodes representing species of the yellow community. Case A: blue and yellow species communities are completely separated (e.g., their respective subgraphs are connected and linked to each other by a unique similarity edge, shown by a dashed line here); Case B: one species of the blue community is affected by gene transfer from a species of the yellow community; Case C: $n/2$ species of the blue community are affected by gene transfers from species of the yellow community; Case D: all n species of the blue community are affected by gene transfers from species of the yellow community.

Transfer index – an Alternative version: It is defined as follows:

$$Tr_{alt}(Y \bullet \ \ X) = \frac{\sum_{i,j \bullet X} \frac{1}{K(ij)} \sum_{k=1}^{K(ij)} \sigma_{ij}^{k}}{\frac{1}{2}(N_X(N_X \bullet 1))}, \qquad (9)$$

where the binary variable $\sigma_{ij}^{k} = 1$ if the kth shortest path between i and j is monochromatic, and $\sigma_{ij}^{k} = 0$, otherwise. An advantage of such an alternative index is that it is not parametric (i.e., it does not depend on the value of $\bullet$). However, this index cannot be directly used to estimate the number/proportion of individual species of a species community affected by horizontal gene transfers from species of other communities.

It is worth noting that the *Transfer* indices defined in Equations (7-9) are asymmetric. Thus, they cannot be called distances. The measures defined in Equations (1) and (3-9) assume that any species community includes at least two species. If a community is represented by a single species, its relation to the rest of the species of this community is supposed to be unknown. The values of all new distances and indices are located in the range [0,1]. In order to make the value of $d(X,X)$ equal to 0 (here d replaces any distance defined in this work), which is a necessary condition for a distance, we can prioritize bichromatic over monochromatic when dealing with network edges that contain nodes that correspond to species belonging to both considered species communities. For instance, if the node i corresponds to a species that belongs to two species communities, then all edges (i,j) could be considered bichromatic, whatever the adjacent vertex j.

2.2 Distance calculation example

In this section, we present an example of a simple SSN to illustrate the calculation of the new community dissimilarity measures defined in Equations (1) and (3-9). The sequence similarity network presented in Figure 2 includes the nodes representing species of two (yellow and blue) communities (2 yellow and 3 blue species). There are two shortest paths, one monochromatic and one non monochromatic (passing by the node 4), between the nodes 2 and 3 representing species of the blue community. All the three shortest paths between the nodes representing species of the yellow community are unique, but two of them (between 1 and 4, and 1 and 5) are not monochromatic. Table 1 reports the detailed calculation results for this simple SSN.

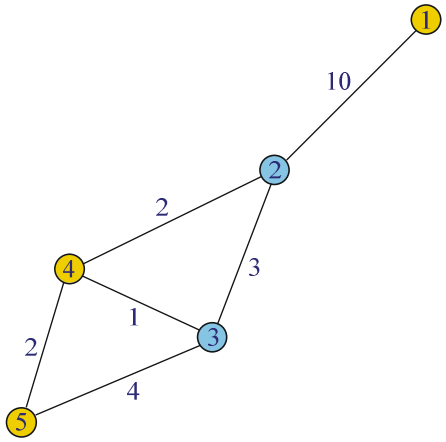


Fig. 2. Sequence similarity network with five nodes (two yellow nodes and three blue nodes, representing two different species communities) and six edges. Edge length is indicated on each edge.

For the SSN shown in Figure 2, the distance values vary from 0.23 for the *NetUniFrac* distance to 0.5 for the *pinp* distance, and the index values vary from 0.33 for the *alternative Transfer* index to 1.0 for the *Transfer* index. Distance or index values close to 0 suggest a high level of interaction between the related communities, while values close to 1 indicate the absence of interaction between them. Some additional examples of computation of the new distances and indices for simple

SSNs and phylogenetic trees can be found in Supplementary Materials (see Supplementary Figure/Table 1 and Figure/Table 1A, respectively).

Table 1. Detailed calculation results for the *NetUniFrac, Spp, Spep, Spelp* and *Spinp* distances, and the *Transfer* and *alternative Transfer* indices defined by Equations (1) and (3-9) for the sequence similarity network in Fig. 2.

Distance	Numerator	Denominator	Value
NetUniFrac	2+3	1+2+2+3+4+10	0.23
Spp	0+0+0.5×(1+0)+1	0.5×(2+6)	0.375
Spep	0+0.33+0.5×(1+0)+1	0.5×(2+6)	0.46
Spelp	0+0.14+0.5×(1+0)+1	0.5×(2+6)	0.41
Spinp	0+0.5+0.5×(1+0)+1	0.5×(2+6)	0.5
$Tr(B\rightarrow Y, \bullet=0.25)$	0+1+1	3	0.67
$Tr(Y\rightarrow B, \bullet=0.5)$	2	2	1.0
$Tr_{alt}(B \bullet Y)$	0+0+1	0.5×6	0.33
$Tr_{alt}(Y \bullet B)$	0.5×(1+0)	0.5×2	0.5

2.3 Algorithm

In this section, we present an algorithm which can be used to calculate the *Transfer* index in a practical situation when a large sequence similarity network including nodes that represent species of several communities should be explored. We show it here in the context of the *Transfer* index, but the algorithm can be easily adapted to calculate the network distances introduced in this paper. In fact, Equations (3) to (9) assume that the input network is a connected graph in which there is at least one path between each pair of nodes. If the original network is unconnected, then its different connected components could be connected pairwise using, for example, the similarity criterion (i.e., when two nodes with the highest similarity in two unconnected components are connected by a new edge) or the betweenness centrality criterion, which is a measure of centrality in a graph based on shortest paths largely used in network theory (i.e., when two nodes with the highest betweenness centrality in two unconnected components are connected by a new edge). It is worth noting that the time complexity of the *Transfer* index calculation using Equations (7-8) is $O(n \bullet \sum_{i=1}^{C} C_i^2)$, where n is the total number of nodes in the network, C is the number of different species communities, and C_i is the number of species in the community i (i = 1,…, C). Even though this time complexity is polynomial, the computation can be long for large SSNs.

Algorithm 1

This algorithm computes an approximate value of the Transfer index accounting for the proportion of species of a given community affected by gene transfer stemming from species of different communities

Input: Sequence similarity network $S = (N, E)$, where N is the set of nodes (each node represents a species) and E is the set of edges. S can be connected or not. Each species in S belongs to one species community.

Output: Matrix of pairwise transfer indices between different species communities.

Step 1. Identify all connected components of each species community (i.e., connected sub-graphs of maximum size composed only of nodes corresponding to species of a given species community; nodes corresponding to species of the other communities are removed from the graph when determining these sub-graphs).

Step 2. Use the following formula to estimate the *Transfer* index (from Y to X) for each pair of species communities (X,Y):

$$\hat{Tr}(Y \bullet X, \bullet_X) = 1 \bullet \frac{\sum_{X_i \bullet X} \bullet_i x_i}{N_X}, \qquad (10)$$

where x_i is the number of species in the connected component X_i of community X, and $\bullet_i = 1$ if all the three conditions below hold:

(a) $x_i \bullet 1 \bullet \bullet_X (N_X \bullet 1)$, where $\bullet_X$ $(0 \bullet \bullet_X \bullet 1)$ is the selected connected component threshold used to decide whether a species of X has been affected by a transfer or not (it is reasonable to assume that large connected components have not been affected by gene transfers; in our program the default value of $\bullet_X$ is set to the middle of the largest interval between pairs of consecutive sorted values of the quantity $\frac{x_i \bullet 1}{N_X \bullet 1}$);

(b) At least one node representing a species of X_i is connected by an edge to at least one node representing a species of community Y in the original network S;

(c) The size of the connected component X_i of X is smaller than the size of at least one connected component Y_j of Y to which a node representing a species of X_i is connected by an edge in the original network S (i.e., $x_i < y_j$);

Otherwise, $\bullet_i = 0.5$ if the conditions (a) and (b) above hold and the size of the connected component X_i of X is equal to the size of the smallest connected component Y_j of Y to which it is connected by an edge in the original network S (i.e., $x_i = y_j$);

Otherwise, $\bullet_i = 0$.

End of algorithm

Clearly, Algorithm 1 is only a heuristic that proposes a way of an approximate estimation the *Transfer* index, but it also provides a fast way of estimating this index. Indeed, the time complexity of Step 1 in Algorithm 1 is $O(m)$ and that of Steps 2 is $O(nC)$, where m is the number of edges, n is the number of nodes, and C is the number of species communities in the network. As the value of C is usually small, compared to m and n, the running time of Algorithm 1 is $O(m)$. Thus, it is linear with respect to the number of edges, making this algorithm applicable to large sequence similarity networks. Moreover, as the results of our simulation study suggest (see Section 3.1), Algorithm 1 provides a good way of finding species clusters affected by horizontal gene transfers and identifying the donor community in which these transfers have originated.

The connected component size assumption that appears in the condition (a) of Algorithm 1 may seem to be too restrictive. However, if a large connected species component is affected by horizontal gene transfer, it usually corresponds to an ancient gene transfer event. Such ancient transfers are typically much harder to detect than the recent ones because they are often obscured by further reticulate evolutionary events. Fortunately, a much higher number of recent gene transfers than of ancient ones are usually observed, and thus can be eventually detected, in prokaryotic phylogenies (Koonin et *al*., 2001).

It is worth noting that Equation (10) does not consider the shortest paths and the approximated value of the transfer index depends only on the selected connected component threshold, $\bullet_X$, and the number of nodes in every connected component. This approximation suits well the problem of identification of recent horizontal gene transfers resulting in an SSN in which individual species or small clusters of species being recipients of genetic material are connected to larger clusters of donors of this material. Some restrictions for the use of Algorithm 1 may also

apply. For example, when one species community is underrepresented compared to the other species community and the level of network connectivity is very high (i.e., the SSN includes too many edges), some false positive transfers from a larger to a smaller species community may be detected by our algorithm. Moreover, when the selected network similarity threshold h (i.e., the threshold used to decide whether two given nodes of the network should be connected or not by an edge) is too high, the network will consist of a large number of unconnected clusters which may lead to overestimating the real number of horizontal gene transfer events associated with this network or to misidentifying the true donors and recipients of genetic material.

3 Results

3.1 Simulation study

We carried out a simulation study to assess the performance of Algorithm 1 in recovery of horizontal gene transfers (HGTs). In our first simulation we compared its performance to those of the popular RIATA-HGT (being part of the PhyloNet package, Wen *et al*., 2018) and HGT-Detection (being part of the T-Rex phylogenetic web server; Boc *et al*., 2012) algorithms. These algorithms infer horizontal gene transfers reconciling a given pair of species and gene phylogenies. The comparison with RIATA-HGT and HGT-Detection (used with BD-optimization option) was carried out on tree-like data in terms of HGT detection accuracy and running time. In this simulation, we used random binary and non-binary species trees that were originally used in the simulations described in Boc *et al.* (2010), where the HGT-Detection algorithm was introduced. These benchmark trees are available at: http://www.labunix.uqam.ca/~makarenkov_v/Simulation_trees.zip (100 binary and non-binary rooted species trees were generated for each pair of parameters: (Number of gene transfers that varied from 1 to 10; Number of tree leaves that varied from 10 to 100, with a step of 10). The gene trees were rooted and binary (see Boc *et al.*, 2010 for more details on the generation of these phylogenies and gene transfers). The root in the species trees was used to separate two species communities (X and Y) in this simulation. In order to apply Algorithm 1, the gene trees were transformed into the corresponding SSNs using the procedure described in Atkinson *et al*. (2009). Algorithm 1 was applied and the value of $\bullet$ for each species community was determined as indicated in Step2a of the algorithm. The average HGT detection error consisting of an average absolute difference between the total number of generated and recovered transfers (this measure was also used in Boc *et al.* 2010) and the average running time were used to compare the competing algorithms (see Fig. 3). Figure 3a shows that HGT-Detection outperformed our Algorithm 1 and RIATA-HGT in terms of the transfer recovery, and that Algorithm 1 was generally more accurate than RIATA-HGT, especially for larger trees. The performance of Algorithm 1 improves as the tree size grows. However, in terms of the running time (see Fig. 3b), the proposed Algorithm 1 was by far the best performer among these methods (here we used trees and SSNs of the following sizes: 10, 100, 500, 1000, 5,000 and 10,000 leaves/nodes). For example, Algorithm 1 took only 11.4 sec. on average to process an SSN with 10,000 nodes, while both RIATA-HGT and HGT-Detection were unable to complete the computations for trees with 5,000 and 10,000 leaves. Our simulations were carried out on a PC computer equipped with an Intel Pentium IV dual-core 3.2 GHz processor and 4 GB of RAM.

Boxplots of the F1-score and Recall measures for Algorithm 1 in the first simulation, calculated using our R program, are presented in Figure 4. The results depicted in this figure confirm that the performance of our algorithm improves as the number of network nodes grows.

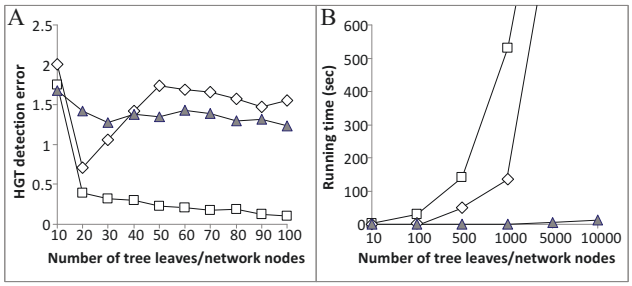


Fig. 3 (a) HGT detection error, consisting of an average absolute difference between the total number of generated and recovered transfers, for RIATA-HGT (white diamonds), HGT-Detection (white squares) and Algorithm 1 (grey triangles) with respect to the number of tree leaves or network nodes. Each reported value represents an average result obtained over 100 random trees generated for each considered tree size and number of transfers; (b) Average running time (in seconds) taken by each algorithm with respect to the number of tree leaves/network nodes.

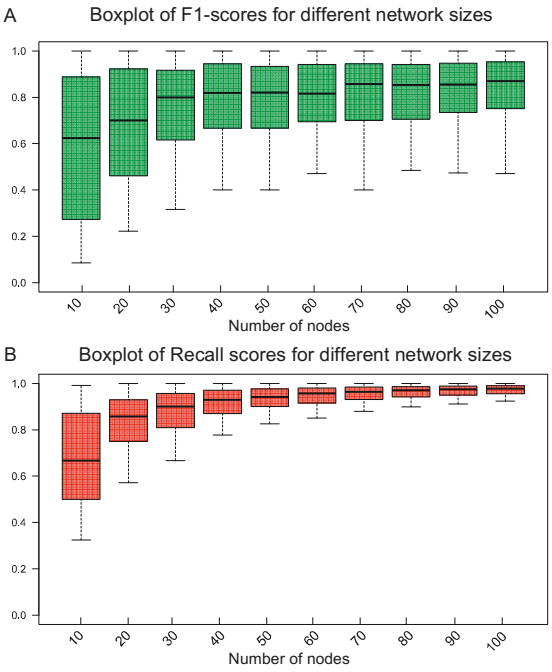


Fig. 4 Boxplots of F1-score (a) and Recall (b) obtained in our first simulation when Algorithm 1 was applied to small SSNs (with 10 to 100 nodes). The x-axis indicates the number of network nodes.

In our second simulation, we investigated the performance of Algorithm 1 on large SSNs including 1000 nodes (here we were unable to compare it to tree-based HGT detection algorithms which were very slow on these data). Random connected SSNs with 1000 nodes representing species belonging to two different species communities (X and Y) were generated. In the original SSNs, the connectivity of nodes corresponding to species of the same species community varied from 0.1 to 0.3 (this parameter was selected randomly using a uniform distribution) and the two species communities were linked by a single edge (necessary to get a connected SSN). SSNs with the following ratios of species belonging to each community were generated: 100 species of X / 900 species of Y; 200 species of X / 800 species of Y; 300 species of X/ 700 species of Y; 400 species of X / 600 species of Y; and 500 species of both X and Y). Gene transfers were simulated in the following way. The number of transferred nodes within each transfer varied from one to

twenty five. The donor node and the sub-graph of transferred nodes were selected randomly. Different total percentages of species affected by HGT as recipients were considered: from 0% to 50% in both *X* and *Y* (see the *x*-axis in Fig. 5). The F1 and Recall measures were calculated to assess the HGT detection quality provided by Algorithm 1 (see Fig. 5). The presented boxplots were calculated over 1000 values of each of these measures computed for different parameter combinations. The following general trend can be observed for both F1 and Recall: The larger the number of generated transfers, the less accurate recovery yielded by Algorithm 1. However, when the number of HGT recipients of the less affected species community was at most 10% (i.e., the first six boxes in Figs 5a and b), the median HGT recovery rate was 0.87 for F1 and 0.81 for Recall, in the worst case.

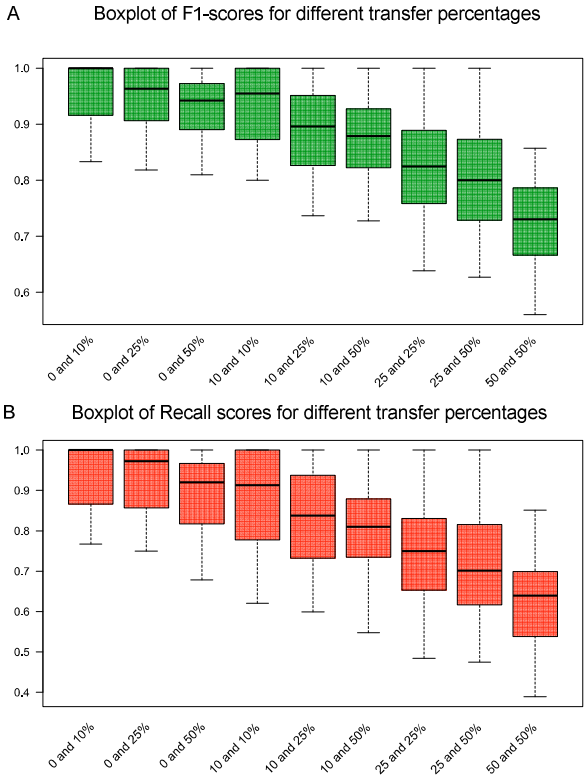


Fig. 5 Boxplots of F1-score (a) and Recall (b) obtained in our second simulation when Algorithm 1 was applied to large SSNs with 1000 nodes. The *x*-axis indicates different percentages of gene transfers between two species community (e.g., 0 and 10% means than 0% of species of the first community were affected by HGT coming from species of the second community, and 10% of species of the first community were affected by HGT coming from species of the second community).

3.2 Analysis of Esophagus data

To illustrate how the new distances can be used in practice we first considered the well-known *Esophagus* dataset (Pei *et al*., 2004). This dataset along with some useful tools for its analysis, including the traditional *UniFrac* distance, are available on the website of the *mothur* package used to fill the bioinformatics needs of the microbial ecology community (Schloss *et al*., 2009). The esophagus dataset contains 684 16S rRNA sequences coming from the distal esophagus of three healthy patients (B, C and D). Thus, one species community by patient was considered. In order to create a sequence similarity network, we used the procedure recommended by Bapteste *et al.* (2012) consisting in the application of the BLAST method on the local database of 684 esophagus sequences. Two nodes in the network were connected by an edge when the similarity between the corresponding 16S rRNA sequences was higher than or equal to 97%. Table 2 reports the main characteristics of the sequence similarity network constructed for the esophagus data. Pairwise species community sub-graphs were created (B-C, C-D and B-D), and the *NetUniFrac*, *Spp*, *Spep*, *Spelp* and *Spinp* distances were calculated along with the traditional *UniFrac* distance. The *Transfer* index was not computed here because the patients B, C and D did not have any interaction among them. The numbers of nodes in the three analyzed subgraphs were very similar, but the numbers of edges in each of them were quite different (see Table 2).

Table 2. Number of nodes and edges in each sub-network of the Esophagus sequence similarity network (built using the BLAST threshold of 97%).

Subnetwork	*Number of nodes*	*Number of edges*
B-C	454	8443
C-D	486	12604
B-D	428	16310
Whole network	684	25061

Table 3. Distance values between bacterial communities B-C, C-D and B-D for the Esophagus sequence similarity network.

	Spp	*Spep/ Spelp*	*Spinp*	*NetUni-Frac*	*Unifrac*[1]	*p-value Unifrac*[2]
B-C	0.52	0.65	0.71	0.61	0.64	<0.001
C-D	0.47	0.58	0.62	0.78	0.68	<0.001
B-D	0.50	0.62	0.67	0.59	0.62	<0.001

According to the *Spp*, *Spep* (or *Spelp*; here, the values of *Spep* and *Spelp* were the same because we did not assign specific weights to the network edges) and *Spinp* distances, the esophagus microbiomes of the patients C and D are the most similar, and the microbiomes of the patients B and C are the most distinct. Overall, the distance values between the patients microbiomes provided by the *Spp*, *Spep* (*Spelp*), *Spinp* and classical *UniFrac* distances are very close to each other, meaning that there is a certain consistency in the microbiome species communities of these healthy patients. However, the structures of their communities are different. The variation between the distance values is much higher for the *NetUniFrac* distance, whose values correlate well with the values of the classical *UniFrac* test, but not with the results yielded by the shortest path-based distances. In this example, the *NetUniFrac* distance better highlights the difference between the three distal esophagus microbiota (with the distance values ranging from 0.59 to 0.78) than the traditional *UniFrac* distance. Interestingly, the smallest values of the shortest path-based distances and the largest values of the *UniFrac* and *NetUniFrac* distances were obtained for the C-D sub-network. It is worth noting that the values of *Spp*, *Spep* (*Spelp*) and *Spinp* correlate well with the parsimony scores obtained for these communities using the pairwise parsimony test. The score of 66 for the B-C sub-network, 56 for the B-D sub-network and 53 for the C-D sub-network, with the value of the *ParsSig* significance parameter < 0.001 for all the three neighbor-joining trees,

[1] *Traditional unweighted UniFrac distance was calculated for the phylogenetic trees given by mothur.*

[2] *The p-values obtained during the significance test for the traditional unweighted UniFrac distance; 1000 permutations were carried out. The p-values for the three trees were <0.001, meaning that the three patients have significantly different community structures.*

were obtained using the *parsimony* command of the *mothur* package (see https://www.mothur.org/wiki/Esophageal_community_analysis). Furthermore, according to Table 1 in Pei *et al*. (2004), the patients C and D have the most similar representation of bacterial phyla in the distal esophagus and subgingival crevice.

3.3 Analysis of TetA and CAT antibiotic resistance gene clusters

A natural application of the *Transfer* index is to estimate the rate of gene transfers of antibiotic resistance genes among different bacteria. Nowadays, the misuse of antibiotics prescribed to patients helps create new drug-resistant "superbugs" that are bacterial strains resistant to several types of antibiotics. This problem is also characteristic for nature and agriculture. Most of these superbugs are created through horizontal gene transfer (HGT) (Koonin *et al.*, 2001; Boc *et al*., 2010). Geographical and taxonomical barriers have been broken since bacteria that are phylogenetically unrelated (e.g. Corynebacterium representatives and Enterobacteria) and live in distinct habitats (e.g. aquatic and host bacteria) can share the same antibiotic resistance determinants due to HGT (Bengtsson-Palme *et al*., 2018). In this paper, we explore the data originally considered by Fondi and Fani (2010) concerning antibiotic resistance genes present in bacteria from different environments, namely: soil, hosts, water, or in any of these three environments (ubiquitous). In some rather rare cases, the species environment was considered as unknown according to the annotation of the GOLD database. Fondi and Fani (2010) built a large antibiotic resistance gene network using the antibiotic resistance gene sequences available in ARDB (Antibiotic Resistance Genes Database). Apart from this comprehensive gene resistance network, they have also conducted a deeper analysis of its TetA and CAT sub-networks representing the interactions between the tetracycline and chloramphenicol resistance proteins, respectively (see Supporting information S4 in Fondi and Fani, 2010, and Supplementary Fig. 2 and Data description in Supplementary Materials).

Because the similarity networks of Fondi and Fani (2010) were based on weighted identity value and did not contain many plausible intra-phylum edges, the community distance calculation applied to them could be biased. Indeed, as many internal links within each taxonomic community were removed, this would artificially reduce the distances between communities. Thus, we reconstructed the TetA (47 species) and CAT (38 species) networks from scratch using the amino acid sequence data provided by Fondi and Fani (2010) in their Supporting information. The multiple sequence alignment using the default *Muscle* options (Edgar, 2004) and the distance calculation using the JTT protein substitution model (Jones *et al.,* 1992) were performed on the T-Rex webserver (Boc *et al*., 2012). The aligned TetA and CAT sequences are available at: https://www.labunix.uqam.ca/~makarenkov_v/TetA_CAT_sequences.zip.

Two species classifications were considered in our study - *Taxonomic classification* and *Environmental classification*. According to the Taxonomic classification (corresponding to NCBI Taxonomy) the given set of organisms was subdivided into: Actinobacteria, Bacilli, -proteobacteria and Other bacteria communities for the TetA gene sequences, and into: Bacilli, -proteobacteria and Other bacteria communities for the CAT gene sequences. According to the Environmental classification (see the LEDA format file in Supporting information of Fondi and Fani, 2010) the given set of species was subdivided into Host, Ubiquitous, Unknown and Water communities for both the TetA and CAT gene sequences. We used the same distance threshold to connect nodes in the TetA and CAT networks. Precisely, the distance threshold of 0.904 was used in both TetA networks (Taxonomic and Environmental, see Figs. 6a and 7a), and both CAT networks (Taxonomic and Environmental, see Figs 6b and 7b). The threshold was applied to the distances generated using the JTT model. Obviously, our networks are different from those inferred by Fondi and Fani (2010).

As we can observe, the two largest species communities, i.e., -proteobacteria and Bacilli are well separated and form community clusters in both the TetA and CAT Taxonomic networks (Figs. 6a and b). However, much smaller Actinobacteria and other bacteria communities don't form any well separated clusters in these networks. The following values of the *Transfer* index, computed using Algorithm 1, were obtained for the TetA and CAT Taxonomic networks (see Tables 4 and 5). Here, a good separation of the -proteobacteria and Bacilli community clusters in both the TetA and CAT Taxonomic networks (Figs. 6a and b) is reflected by high values of the *Transfer* index for the transfer in both directions (for the transfer: -proteobacteria Bacilli, the *Transfer* index is 1 in both networks, and for the transfer Bacilli -proteobacteria, the *Transfer* index is 0.97 in the TetA network and 0.96 in the CAT network, accounting for one HGT event in both cases).

When the environmental species classification was considered (see Figs. 7a and b), the species communities seemed to be much more mixed. It is reasonable to consider the environmental species classification in this case because the propagation of antibiotics also depends on the geographical constraints. In the environmental classification, most of the sequences belong to the Host community and these sequences are well mixed with the representatives of the Ubiquitous, Water and Unknown environments. As the Host community is a dominating one in the number of species, most of the values of the *Transfer* index from the three other environments to host are equal to 1 (see Tables 6 and 7). Most of the transfers detected, resulting in smaller values of the *Transfer* index, go from the Host community to the three other environments. Sometimes, the direction of transfer is difficult to identify, and in this case Algorithm 1 tends to indicate as the transfer donor community, the community with a larger number of species present in a given connected component of the network. The distances calculated for the TetA and CAT habitat networks (see Supplementary Fig. 2) built by Fondi and Fani (2010), can be found in Supplementary Tables 2 and 3.

Table 4. Transfer indices for the TetA taxonomic network. The value of 1 on the intersection of the row of Actinobacteria and the column of Bacilli means that no gene transfers from Actinobacteria to Bacilli has occurred according to this sequence similarity network. The standard deviations calculated over the four elbow points, which are good candidates for the network distance threshold (see Supplementary Fig. 3), are indicated between parentheses.

Community	*Actino-bacteria*	*Bacilli*	*-proteo-bacteria*	*Other bacteria*
Actino-bacteria	-	1 (0)	1 (0)	1 (0.14)
Bacilli	1 (0)	-	0.97 (0.01)	1 (0)
-proteo-bacteria	0.5 (0.41)	0.95 (0)	-	0.67 (0.17)
Other bacteria	1 (0.22)	1 (0)	1 (0)	-

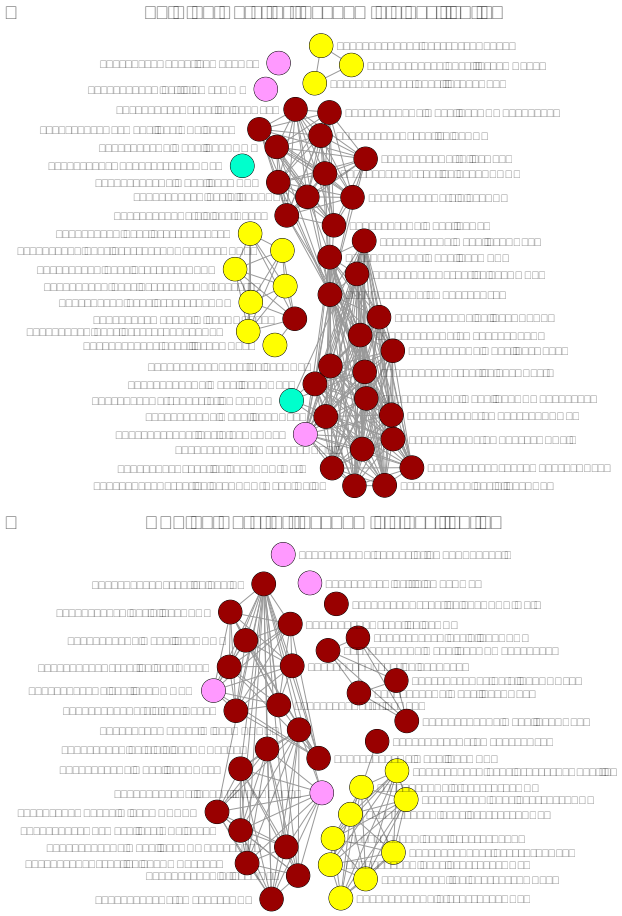


Fig. 6. TetA and CAT Taxonomic networks built for the Actinobacteria (teal nodes), Bacilli (yellow nodes), -proteobacteria (brown nodes) and Other bacteria (pink nodes) species communities from the JTT distance matrices computed from the amino acid sequences aligned using *Muscle*. The distance threshold of 0.904 for both the TetA and CAT networks was applied to determine whether a given pair of nodes should be connected by an edge or not (i.e., the edge between two nodes was added if the corresponding distance was smaller than the selected threshold).

Table 5. Transfer indices for the CAT taxonomic network.

Community	*Bacilli*	*-proteobacteria*	*Other bacteria*
Bacilli	-	0.96 (0.02)	1 (0.21)
-proteobacteria	1 (0)	-	0.5 (0)
Other bacteria	1 (0)	1 (0)	-

Table 6. Transfer indices for the TetA environmental network.

Community	*host*	*ubiquitous*	*unknown*	*water*
host	-	0 (0.18)	0.33 (0.17)	0 (0.14)
ubiquitous	1 (0)	-	1 (0.07)	1 (0.36)
unknown	1 (0)	0.25 (0.36)	-	0.5 (0.43)
water	1 (0)	0 (0.27)	0.83 (0.05)	-

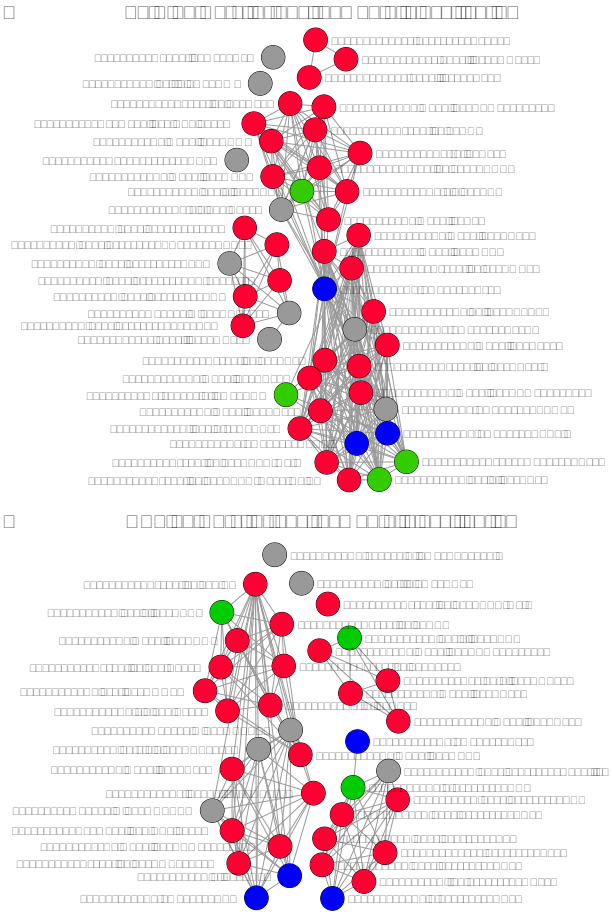


Fig. 7. TetA and CAT Environmental networks built for the Host (red nodes), Ubiquitous (green nodes), Water (blue nodes) and Unknown (grey nodes) communities from the JTT distance matrices computed from the amino acid sequences aligned using *Muscle*. The distance threshold of 0.904 for both the TetA and CAT networks was applied to determine whether a given pair of nodes should be connected or not.

Table 7. Transfer indices for the CAT environmental network.

Community	*host*	*ubiquitous*	*unknown*	*water*
host	-	0 (0.07)	0.33 (0.25)	0.25 (0.13)
ubiquitous	1 (0.02)	-	0.92 (0.05)	0.75 (0.12)
unknown	1 (0.02)	0.83 (0.26)	-	0.38 (0.26)
water	1 (0)	0.83 (0.22)	0.92 (0.04)	-

General *NetUniFrac* distances, calculated for the four whole networks under study, and their standard deviations (indicated between parentheses), calculated over the four elbow points which are good candidates for the network distance threshold (Supplementary Fig. 3), are as follows: TetA Taxonomic network 0.897 (0.067), CAT Taxonomic network 0.864 (0.164), TetA Environmental network 0.480 (0.075) and CAT Environmental network 0.517 (0.059). The standard deviations of the *Transfer* index computed over the same elbow points are indicated between parentheses in Tables 4 to 7. In most cases, these standard deviations are low, showing a relative stability of *NetUniFrac* and the *Transfer* index for these data, and especially for larger species communities, such as -proteobacteria in the Taxonomic networks (Fig. 6) and Host organisms in the Environmental networks (Fig. 7).

4 Conclusion

In a phylogenetic tree the length of a unique shortest path between two nodes representing certain species is used to estimate evolutionary distance between them. In this work, we showed how the shortest path concept can be extended to sequence similarity networks. We defined five new network distances between species communities, including *NetUniFrac*, which is a direct generalization of the traditional *UniFrac* distance to sequence similarity networks. *NetUniFrac* can be calculated in linear time with respect to the number of network edges. The four other distances, *Spp*, *Spep*, *Spelp* and *Spinp*, are shortest path-based distances. Moreover, we introduced the *Transfer* index which can be used to estimate the rate and direction of horizontal gene transfers between different species communities. We showed how these novel distances and indices can be used for the analysis of microbiota and antibiotic resistance gene similarity networks. Generally, distance or index values close to 0 suggest that the related communities interact a lot, while their higher values indicate a low level of interaction between them.

As we could see during the analysis of the *Esophagus* dataset, the values of the *NetUniFrac* distance calculated for an SSN correlate well with those of the classical *UniFrac* distance calculated for a phylogenetic tree. In this example, *NetUniFrac* better highlighted the differences between the distal esophagus microbiota than classical *UniFrac*. Furthermore, the *NetUniFrac* computation can be completed much faster than that of classical *UniFrac* since the inference of an SSN usually requires $O(n^2)$ operations (to calculate all pairwise distances between the nodes of the network), while the inference of a phylogenetic tree usually requires at least $O(n^3)$ operations (as in the case of the popular Neighbor-Joining algorithm (Saitou and Nei, 1987)), where n is the number of network nodes or tree leaves. We need to emphasize the fact that using Algorithm 1, we can calculate the pairwise matrix of the *Transfer* indices between different species communities in linear time with respect to the number of edges in the network. The results of our simulation study indicate that Algorithm 1 provides a fast and accurate way of identifying species clusters affected by horizontal gene transfer and determining the donor community where these transfers have originated. Algorithm 1, along with the new shortest-path based network distances, was implemented in R and C, and included in our *NetFrac* package available at: https://github.com/XPHenry/Netfrac. In the future, it would be interesting to adapt Algorithm 1 to the detection of mosaic genes and composite gene families (Boc and Makarenkov, 2011) in order to compare its performance to that of the CompositeSearch (Pathmanathan *et al.* 2018), FusedTriplets, and MosaicFinder (Jachiet *et al*. 2013) programs.

Acknowledgements

We are grateful to Dr. Pier Luigi Martelli and three anonymous reviewers for their helpful comments and discussions.

Funding

This work was supported by Le Fonds Québécois de la Recherche sur la Nature et les Technologies (grant no. 173878) and Natural Sciences and Engineering Research Council of Canada (grant no. 249644).

Conflict of Interest: none declared.

References

Atkinson,HJ. *et al.* (2009) Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PLOS ONE,* **4(2)**, e4345.

Bapteste,E. *et al.* (2013) Networks: expanding evolutionary thinking. *Trends Genet.*, **29**, 439-441.

Bapteste,E. *et al.* (2012) Evolutionary analyses of non-genealogical bonds produced by introgressive descent. *Proc. Natl. Acad. Sci. USA*, **109**, 18266-18272.

Bengtsson-Palme,J. *et al.* (2018) Environmental factors influencing the development and spread of antibiotic resistance. *FEMS Microbiol. Rev*. **42**, fux053.

Boc,A. and Makarenkov,V. (2011) Towards an accurate identification of mosaic genes and partial horizontal gene transfers. *Nucleic Acids Res.*, **39**, e144-e144.

Boc,A. *et al.* (2012) T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks, *Nucleic Acids Res.*, **40**, W573-W579.

Boc,A. *et al.* (2010) Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. biol.*, **59**, 195-211.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*., **32**, 1792-97.

Felsenstein,J. (2004) *Inferring Phylogenies*, Vol. 2, Sunderland, MA: Sinauer associates.

Fondi,M. and Fani,R. (2010) The horizontal flow of the plasmid resistome: clues from inter-generic similarity networks. *Environ. microbiol.*, **12**, 3228-3242.

Girvan,M. and Newman,M.E. (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, **99**, 7821-7826.

Huson,D.H. and Bryant,D. (2005) Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol.*, **23**, 254-267.

Jachiet,P.A. *et al.* (2013) MosaicFinder: identification of fused gene families in sequence similarity networks. *Bioinformatics*, **29**, 837–844.

Jones,D.T. *et al.* (1992) The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*., **8**, 275-82.

Koonin,E.V. *et al.* (2001) *Horizontal gene transfer in prokaryotes: quantification and classification*. *Annu. Rev. Microbiol*., **55**, 709-742.

Lozupone,C. *et al.* (2011) *UniFrac*: an effective distance metric for microbial community comparison. *ISME J.*, **5**, 169-172.

Newman,M.E. (2003). Mixing patterns in networks. *Phys. Rev.* E, 67, 026126.

Newman,M.E (2010) *Networks: an introduction*. Oxford University Press.

Parks,D.H. and Beiko,R.G. (2012) Measuring community similarity with phylogenetic networks. *Mol. Biol. Evol.*, **29**, 3947-3958.

Pathmanathan,J.S. *et al.* (2018) CompositeSearch: A generalized network approach for composite gene families detection, *Mol. Biol. Evol*., **35**, 252–255.

Pei,Z. *et al.* (2004) Bacterial biota in the human distal esophagus. *Proc. Natl. Acad. Sci. USA*, **101**, 4250-4255.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*., **4**, 406-425.

Schloss,P.D. *et al.* (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microb.*, **75**, 7537-7541.

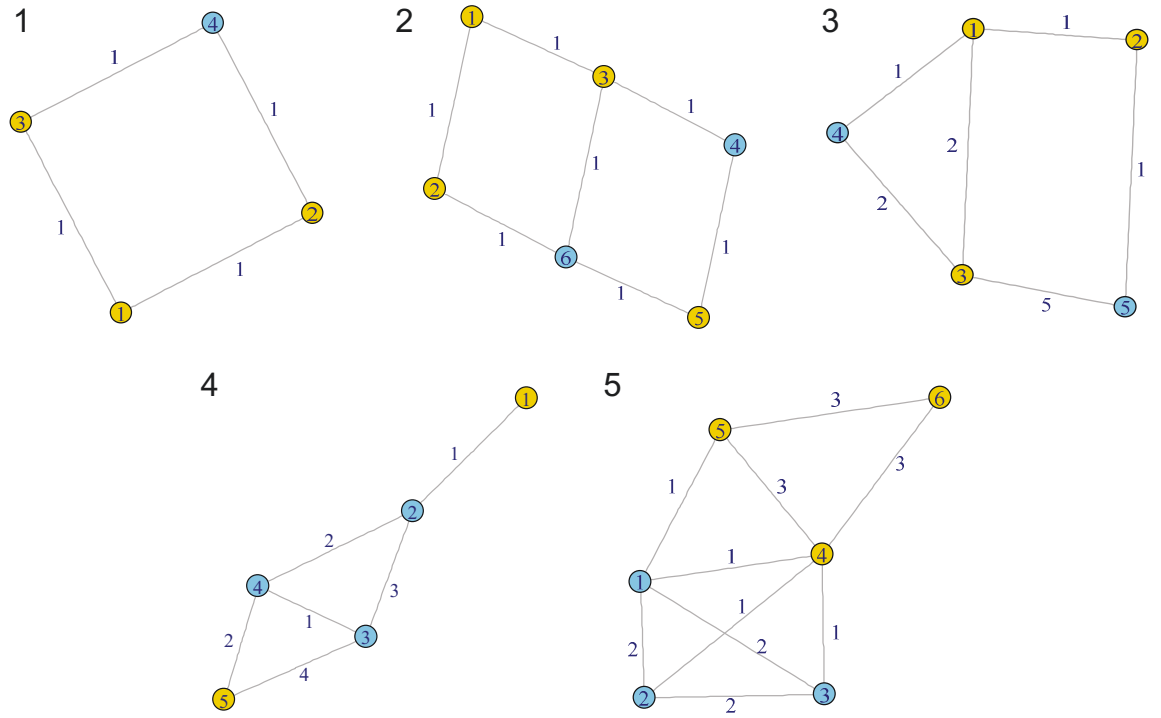
Wen,D *et al.* (2018). Inferring phylogenetic networks using PhyloNet. *Syst. Biol*., **67**, 735-740.

Wicke,K. and Fischer,M. (2018) Phylogenetic diversity and biodiversity indices on phylogenetic networks. *Math. Biosci.*, **298**, 80-90.

Supplementary materials

Distance calculations for simple Sequence Similarity Networks (SSNs)

Supplementary Fig. 1 depicts five simple SSN (Sequence Similarity Network) configurations. The *NetUnifrac, Motif, Spp, Spep, Spelp,* and *Spinp,* and *Transfer* and *Alternative Transfer* indices were calculated for these networks. The obtained results are reported in Supplementary Table 1.

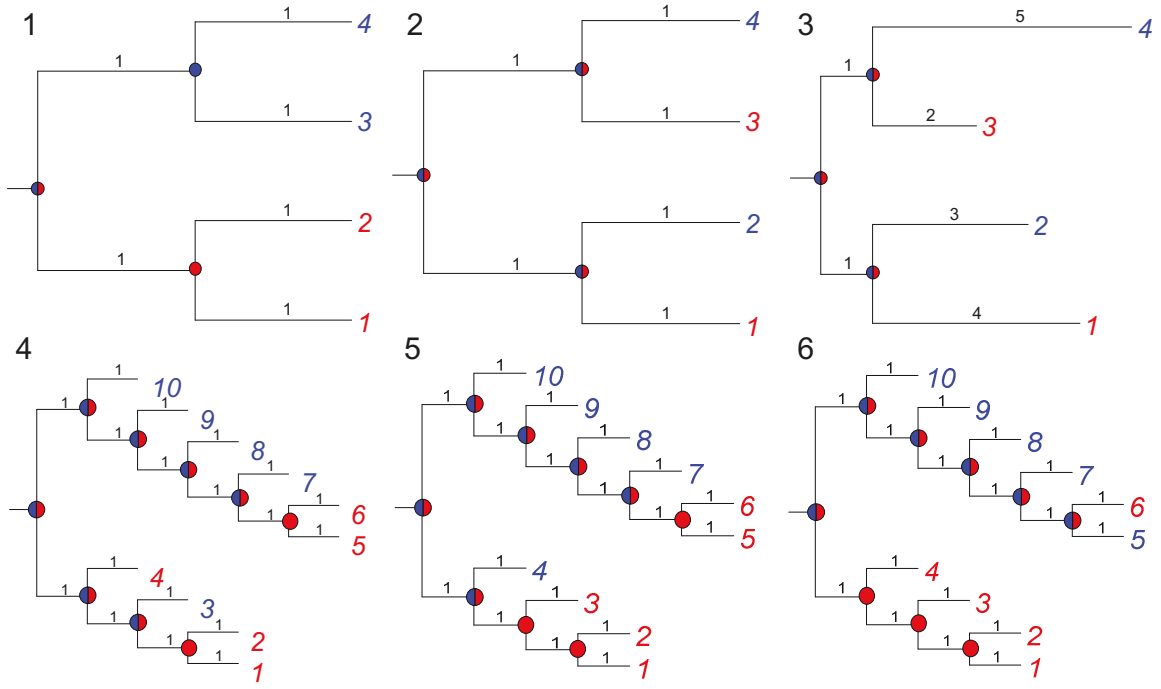


Supplementary Figure 1. Five simple SSNs including nodes representing species of the yellow (Y) and blue (B) communities. The numbers indicated on the edges represent their length.

Supplementary Table 1. Distance and index values obtained for the SSNs represented in Supplementary Fig. 1.

Distance/ SSN	*NetUnifrac*	*Motif_3*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer* (*Y* • *B*)	*Transfer* (*B* • *Y*)	*Alt. Trans* (*Y* • *B*)	*Alt. Trans* (*B* • *Y*)
1	0.5	0.25	0.75	0.667	0.667	0.25	0	1	0	0.833
2	0.286	0.22	0.25	0.28	0.28	0.6	0	0.75	0	0.5
3	0.33	0.1	0.75	0.571	0.667	0.4	0	1	0	1
4	0.5	0.2	0.8	0.625	0.643	0.33	1	0	1	0
5	0.6	0.4	0.556	0.385	0.6	0.444	0.5	0.66	0.5	0.66

Distance calculations for simple phylogenetic trees



Supplementary Figure 1A. Six phylogenetic rooted trees used to demonstrate the distance and index calculations. The weights of edges are indicated in black. Species belonging to two species communities blue (B) and red (R) are initially associated to the tree leaves. Then each internal node of the tree is colored in blue only, or in red only, or in blue/red, depending on the leaves located in the sub-tree rooted by this node.

Six simple phylogenetic tree configurations are presented in Supplementary Fig. 1A. The following pairwise community distances: *UniFrac* (traditional), *Spp*, *Spep*, *Spelp*, and indices: *Transfer* and *Alternative Transfer* can be also calculated for these trees.

Supplementary Table 1A. Distance values obtained for the rooted trees represented in Supplementary Fig. 2.

Distance/ Tree	*UniFrac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer* ($R \bullet B$)	*Transfer* ($B \bullet R$)	*Alt Trans* ($R \bullet B$)	*Alt Trans* ($B \bullet R$)
1	1	1	1	1	1	1	1	1	1
2	0.667	0	0	0	0	0.5	0.5	0	0
3	0.875	0	0	0	0	0.5	0.5	0	0
4	0.667	0.1	0.138	0.138	0.1	0.2	0.2	0	0.2
5	0.72	0.2	0.232	0.232	0.2	0	0.4	0	0.4
6	0.72	0.3	0.292	0.292	0.3	0.2	0.2	0	0.6

TetA and CAT sub-networks description

Fondi and Fani (2010) built a large antibiotic resistance gene network using the antibiotic resistance gene sequences available in ARDB (Antibiotic Resistance Genes Database). The 97% similarity threshold in BLAST was used to construct an antibiotic resistance gene network with 5030 nodes and 259 726 edges. To remove the links representing vertical inheritance (intra-phylum) genus connections, another network with the same species was created using 16S RNA sequences (Fondi and Fani, 2010). The weighted identity value (WIV) was calculated for each edge. The WIV threshold of 9 was then used to reduce the number of intra-phylum network edges in the network, and thus highlight the edges connecting species from different taxonomic communities. Thus, the reduced network mainly contained the edges representing HGT events. Apart from this comprehensive gene resistance network, these authors have also conducted a deeper analysis its TetA and CAT sub-networks, representing the interactions between the tetracycline and chloramphenicol resistance proteins, respectively (see Supporting information S4 in Fondi and Fani, 2010, and Supplementary Fig. 2). It is known that the transfer of tetracycline resistance genes has occurred frequently between different species of bacteria present in human body, either from the microbiota or from external pathogens, because these bacteria interact within the same host (Speer *et al*., 1992). For example, plasmid mediated transfers have been the main mean of acquisition and dissemination of tetracycline resistance in *Laribacter hongkongensis* (Lau, Wong *et al*., 2008). Also, the increasing incidence of multidrug resistance to tetracycline amongst *Aeromonas spp.* isolates, which are both fish pathogens and human pathogens, was shown to stem from mobile genetic elements (Jacobs and Chenia, 2007). The chloramphenicol resistance follows a similar pattern in bacteria of the Staphylococcus (Bhakta, Arora *et al*., 2003), Neisseria (Galimand, Gerbaud *et al*., 1998), Enterococcus (Gould, Fishman *et al*., 2004) and Salmonella (Karunaratne, Wickremesinghe *et al*., 2000) genera. Further investigations have revealed that maricultural water bacteria are often chloramphenicol resistant (Dang, Zhao *et al*., 2009). Notably, the presence of links among these species, regardless their habitat and/or taxonomical classification, strongly suggests that the occurrence of such an antibiotic resistance comes from HGT events.

References

Bhakta,M. *et al*. (2003) Intraspecies transfer of a chloramphenicol-resistance plasmid of staphylococcal origin. *Indian J. Med. Res.*, **117**, 146-151.

Dang,H. *et al.* (2009) Molecular characterizations of chloramphenicol and oxytetracycline-resistant bacteria and resistance genes in mariculture waters of China. *Mar. Pollut. Bull.*, **58**, 987-994.

Galimand,M., *et al.* (1998) High-level chloramphenicol resistance in Neisseria meningitidis. New England Journal of Medicine, 339(13), 868-874.

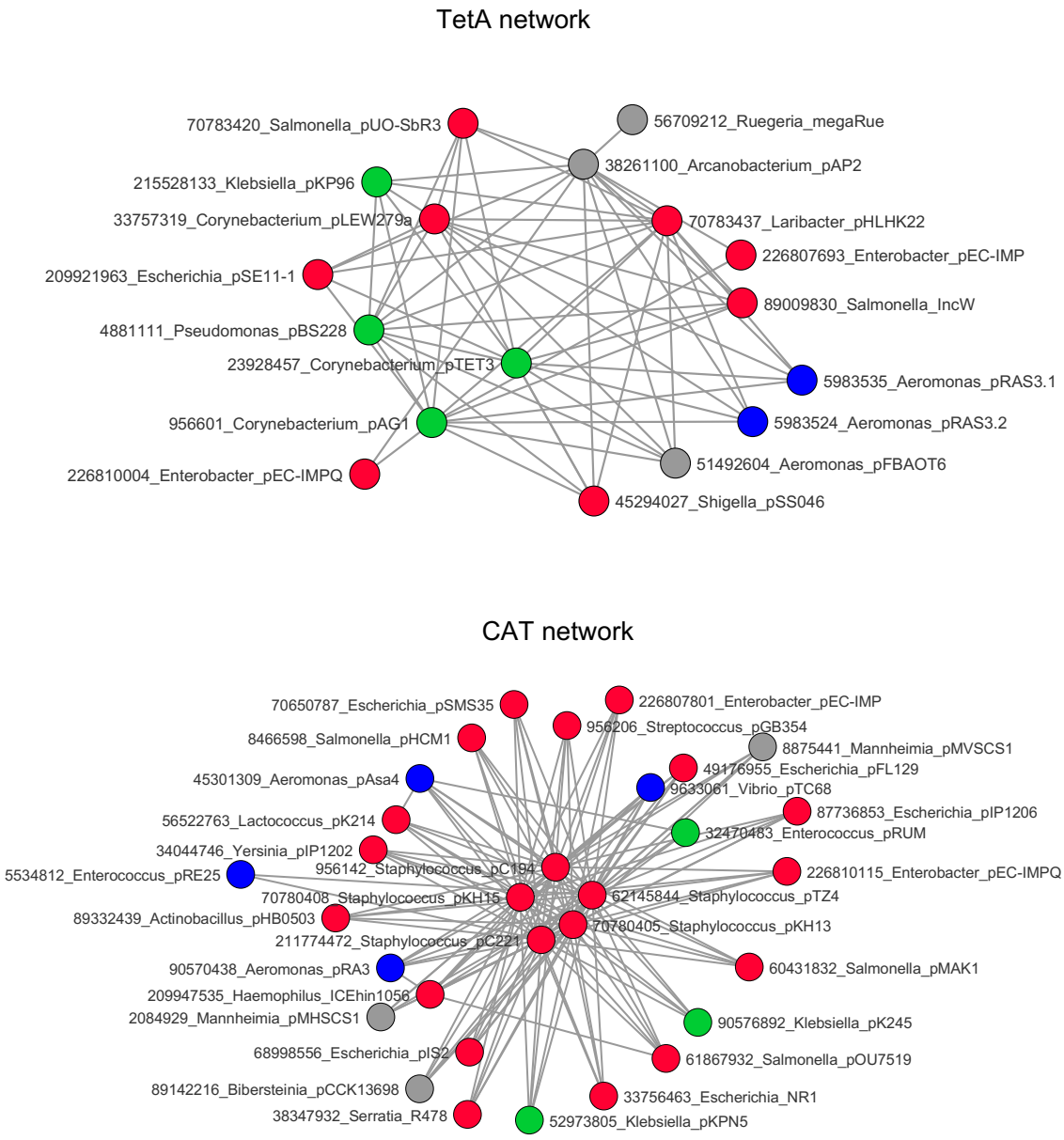
Gould,C.V. *et al.* (2004) Chloramphenicol resistance in vancomycin-resistant enterococcal bacteremia: impact of prior fluoroquinolone use? *Infect. Cont. Hosp. Ep.*, **25**, 138-145.

Jacobs,L. and Chenia,H.Y. (2007) Characterization of integrons and tetracycline resistance determinants in Aeromonas spp. isolated from South African aquaculture systems. *Int. J. Food Microbiol.*, **114**, 295-306.

Karunaratne,G.K. *et al.* (2000) Salmonella typhi and chloramphenicol resistance. *Ceylon Med. J.*, **45**, 136-137.

Lau,S.K. *et al.* (2008) "Distribution and molecular characterization of tetracycline resistance in Laribacter hongkongensis." *J. Antimicrob. Chemoth.*, **61**, 488-497.

Speer,B. *et al*. (1992) Bacterial resistance to tetracycline: mechanisms, transfer, and clinical significance. *Clin. Microbiol. Rev.*, **5**, 387-399.



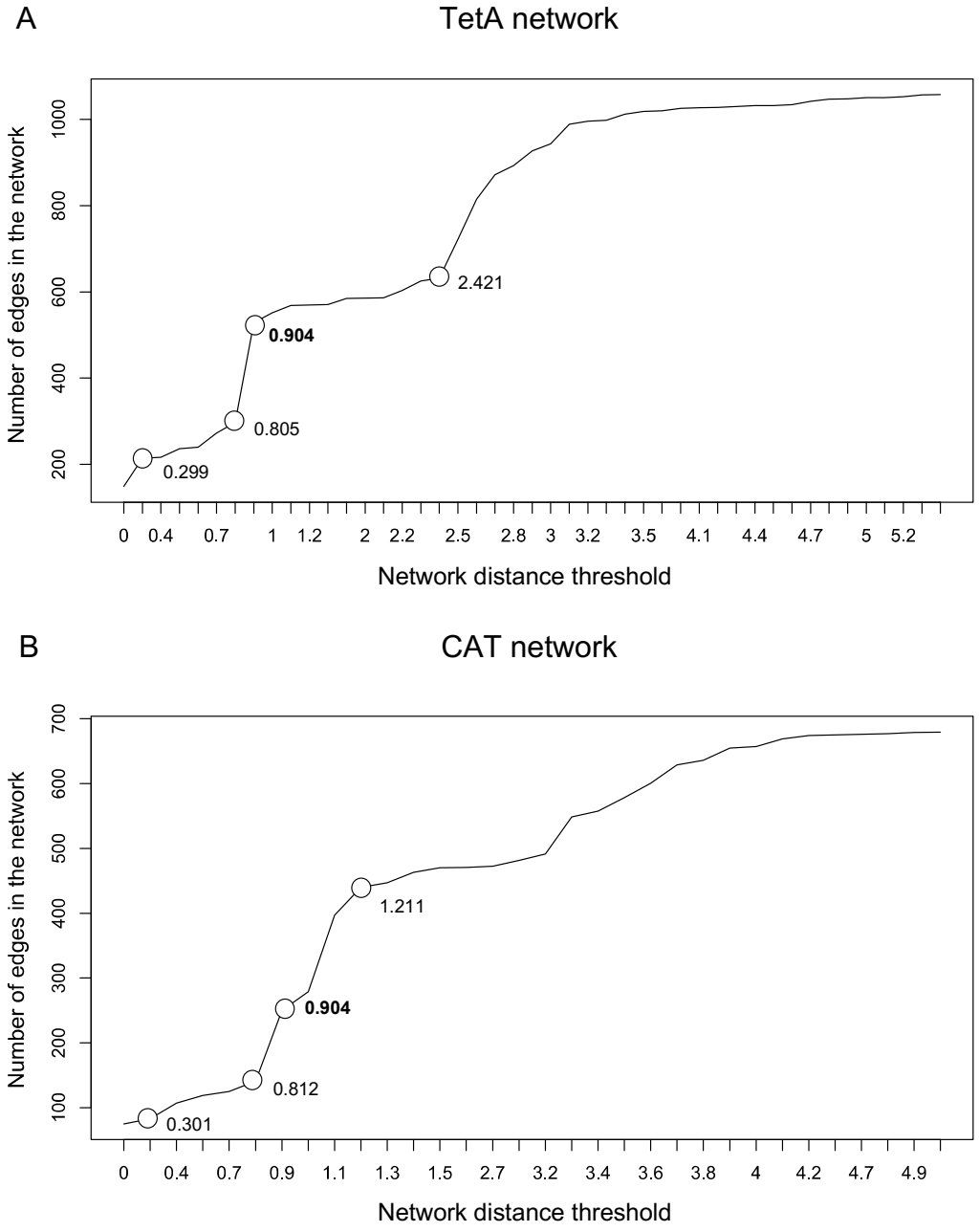
Supplementary Figure 2. TetA and CAT sequence similarity networks from Supporting information S4 in Fondi and Fani (2010). Nodes represent proteins of bacterial organisms and edges represent the WIV they share (see the main text). Nodes have been colored according to the habitat assigned to each organism: red nodes stand for host organisms, blue nodes for organisms living in water, green nodes for organisms found in all habitats (ubiquitous), and grey nodes for organisms lacking habitat assignation in the GOLD database. Redundant nodes or edges were removed from the network.

Supplementary Table 2. Distance and index values obtained for the TetA network represented in Supplementary Fig. 2.

Distance/Habitats	*Netuni Frac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer-direct*	*Transfer-reverse*	*Alt Trans direct*	*Alt Trans reverse*
Host & ubiquitous	0.38	0.41	0.43	0.43	0.97	0	0.75	0.32	0.83
Host & unknown	0.56	1.00	1.00	1.00	1.00	0	0.75	1	1
Host & water	0.54	0.86	0.86	0.86	0.86	0	1	0	1
Ubiquitous & unknown	0.43	0.58	0.60	0.60	0.97	0	1	1	0.54
Ubiquitous & water	0.69	0.97	0.97	0.97	0.90	0	1	0	1
Unknown & water	0.43	0.75	0.75	0.75	0.75	0	1	0	1

Supplementary Table 3. Distance and index values obtained for the CAT network represented in Supplementary Fig. 2.

Distance/Habitats	*Netuni Frac*	*Spp*	*Spep*	*Spelp*	*Spinp*	*Transfer-direct*	*Transfer-reverse*	*Alt Trans direct*	*Alt Trans reverse*
Host & ubiquitous	0.87	0.99	0.99	0.99	0.99	0	1	0	1
Host & unknown	0.84	0.99	0.99	0.99	0.99	0	1	0	1
Host & water	0.80	0.97	0.97	0.97	0.97	0	1	0	1
Ubiquitous & unknown	0.40	1.00	1.00	1.00	1.00	1	1	1	1
Ubiquitous & water	0.38	0.78	0.78	0.78	1.00	0.88	0.83	1	0.33
Unknown & water	0.43	1.00	1.00	1.00	1.00	1	1	1	1



Supplementary Figure 3. The presented graphs show how the number of network edges depends on the network distance threshold for the TetA (case a) and CAT (case b) sequence similarity networks discussed in the article. For both networks 4 elbow points are presented. These elbow points are good candidates for selecting the network distance threshold. In our study (see the main text), the value of 0.904 was selected as the network distance threshold for both the TetA and CAT networks.

RÉFÉRENCES

Atkinson, H. J., Morris, J. H., Ferrin, T. E., et Babbitt, P. C. (2009). Using sequence similarity networks for visualization of relationships across diverse protein superfamilies. *PloS one*, *4*(2).

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., et Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403-410.

Bapteste, E., van Iersel, L., Janke, A., Kelchner, S., Kelk, S., McInerney, J. O., ... Whitfield, J. (2013). Networks: expanding evolutionary thinking. *Trends in Genetics*, *29*(8), 439-441.

Bollobás, B. (2013). *Modern graph theory* (Vol. 184). Springer Science et Business Media.

Bapteste, E., Lopez, P., Bouchard, F., Baquero, F., McInerney, J. O., et Burian, R. M. (2012). Evolutionary analyses of non-genealogical bonds produced by introgressive descent. *Proceedings of the National Academy of Sciences*, *109*(45), 18266-18272.

Ben-Gal, I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., ... Grosse, I. (2005). Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, *21*(11), 2657-2666.

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., et Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS computational biology*, *4*(10).

Bengtsson-Palme, J., Kristiansson, E., et Larsson, D. J. (2018). Environmental factors influencing the development and spread of antibiotic resistance. *FEMS microbiology reviews*, *42*(1), fux053.

Boc, A., et Makarenkov, V. (2011). Towards an accurate identification of mosaic genes and partial horizontal gene transfers. *Nucleic acids research*, *39*(21), e144-e144.

Boc, A., Diallo, A. B., et Makarenkov, V. (2012). T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks. *Nucleic acids research*, *40*(W1), W573-W579.

Boc, A., Philippe, H., et Makarenkov, V. (2010). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Systematic biology*, *59*(2), 195-211.

Cavender-Bares, J., Kozak, K. H., Fine, P. V., et Kembel, S. W. (2009). The merging of community ecology and phylogenetic biology. *Ecology letters*, *12*(7), 693-715.

Chatzou, M., Magis, C., Chang, J. M., Kemena, C., Bussotti, G., Erb, I., et Notredame, C. (2016). Multiple sequence alignment modeling: methods and applications. *Briefings in bioinformatics*, *17*(6), 1009-1023.

Deng, Y., Chen, Y., Zhang, Y., et Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, *12*(3), 1231-1237.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269-271.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, *32*(5), 1792-1797.

Edgar, R. C., et Batzoglou, S. (2006). Multiple sequence alignment. *Current opinion in structural biology*, *16*(3), 368-373.

Felsenstein, J. (2004) *Inferring Phylogenies*, Vol. 2, Sunderland, MA: Sinauer associates.

Fondi, M., et Fani, R. (2010). The horizontal flow of the plasmid resistome: clues from inter-generic similarity networks. *Environmental microbiology*, *12*(12), 3228-3242.

Forster, D., Bittner, L., Karkar, S., Dunthorn, M., Romac, S., Audic, S., ... Bapteste, E. (2015). Testing ecological theories with sequence similarity networks: marine ciliates exhibit similar geographic dispersal patterns as multicellular organisms. *BMC biology*, *13*(1), 16.

Girvan, M., et Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, *99*(12), 7821-7826.

Gligorijević, V., et Pržulj, N. (2015). Methods for biological data integration: perspectives and challenges. *Journal of the Royal Society Interface*, *12*(112), 20150571.

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., ... Tegnér, J. (2014). Data integration in the era of omics: current and future challenges.

Hawkins, R. D., Hon, G. C., et Ren, B. (2010). Next-generation genomics: an integrative approach. *Nature Reviews Genetics*, *11*(7), 476-486.

Huson, D. H., et Bryant, D. (2006). Application of phylogenetic networks in evolutionary studies. *Molecular biology and evolution*, *23*(2), 254-267.

Jachiet, P. A., Pogorelcnik, R., Berry, A., Lopez, P., et Bapteste, E. (2013). MosaicFinder: identification of fused gene families in sequence similarity networks. *Bioinformatics*, *29*(7), 837-844.

Jones, D. T., Taylor, W. R., et Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, *8*(3), 275-282.

Joyce, A. R., et Palsson, B. Ø. (2006). The model organism as a system: integrating ‘omics' data sets. *Nature reviews Molecular cell biology*, *7*(3), 198-210.

Koonin, E. V., Makarova, K. S., et Aravind, L. (2001). Horizontal gene transfer in prokaryotes: quantification and classification. *Annual Reviews in Microbiology*, *55*(1), 709-742.

Korf, I., Yandell, M., et Bedell, J. (2003). *Blast*. " O'Reilly Media, Inc.".

Lozupone, C., Lladser, M. E., Knights, D., Stombaugh, J., et Knight, R. (2011). UniFrac: an effective distance metric for microbial community comparison. *The ISME journal*, *5*(2), 169-172.

Martiny, J. B., Jones, S. E., Lennon, J. T., et Martiny, A. C. (2015). Microbiomes in light of traits: a phylogenetic perspective. *Science*, *350*(6261), aac9323.

Newman, M.E. (2003). Mixing patterns in networks. *Phys. Rev. E,* 67, 026126.

Newman, M.E (2010) *Networks: an introduction*. Oxford University Press.

Parks, D. H., et Beiko, R. G. (2012). Measuring community similarity with phylogenetic networks. *Molecular biology and evolution*, *29*(12), 3947-3958.

Pathmanathan, J. S., Lopez, P., Lapointe, F. J., et Bapteste, E. (2018). CompositeSearch: a generalized network approach for composite gene families detection. *Molecular biology and evolution*, *35*(1), 252-255.

Pei, Z., Bini, E. J., Yang, L., Zhou, M., Francois, F., et Blaser, M. J. (2004). Bacterial biota in the human distal esophagus. *Proceedings of the National Academy of Sciences*, *101*(12), 4250-4255.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., et Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*(5721), 523-529.

Saitou, N., et Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, *4*(4), 406-425.

Schloss, P. D., Westcott, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., ... Sahl, J. W. (2009). Introducing Mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.*, *75*(23), 7537-7541.

Semple, C., et Steel, M. (2003). *Phylogenetics* (Vol. 24). Oxford University Press on Demand.

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F. H., Goehler, H., ... Timm, J. (2005). A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, *122*(6), 957-968.

Walter, J., et Ley, R. (2011). The human gut microbiome: ecology and recent evolutionary changes. *Annual review of microbiology*, *65*, 411-429.

Wang, Y., Chen, S., Deng, N., et Wang, Y. (2013). Drug repositioning by kernel-based integration of molecular structure, molecular activity, and phenotype data. *PloS one*, *8*(11).

Wen, D., Yu, Y., Zhu, J., et Nakhleh, L. (2018). Inferring phylogenetic networks using PhyloNet. *Systematic biology*, *67*(4), 735-740.

Wicke, K., et Fischer, M. (2018). Phylogenetic diversity and biodiversity indices on phylogenetic networks. *Mathematical biosciences*, *298*, 80-90.

Winterbach, W., Van Mieghem, P., Reinders, M., Wang, H., et de Ridder, D. (2013). Topology of molecular interaction networks. *BMC systems biology*, *7*(1), 90.

Xiong, J. (2006). *Essential bioinformatics*. Cambridge University Press.

Yang, Z., et Rannala, B. (2012). Molecular phylogenetics: principles and practice. *Nature reviews genetics*, *13*(5), 303-314.

Zhan, F. B. (1997). Three fastest shortest path algorithms on real road networks: Data structures and procedures. *Journal of geographic information and decision analysis*, *1*(1), 69-82.