



# Raport

## Studiu privind identificarea elementelor cu rol critic în cadrul unei rețele de interacțiuni

efectuat în cadrul proiectului *Abordarea bioeconomică a  
agenților antimicrobieni – utilizare și rezistență*

(cod - PN-III-P1-1.2-PCCDI-2017-0361).

Colectiv de redacție:

Alina Cărunta: noduri critice și comunități în rețele complexe, pachete R pentru analiza rețelelor, studiu de caz (sect. 2, 3, 4)

Daniela Zaharie: introducere (sect. 1), concluzii (sect. 5)

Coordonator: Daniela Zaharie

Membri: Alina Cărunta, Mihai Chiș

Data finalizării: 20.11.2020

Raport: R.5.3.3.

Versiunea: 1.0

### **Acknowledgements**

Activities under this work were carried out in the *West University of Timisoara* - financed by project "A bio-economical approach of the antimicrobial agents - use and resistance", in the frame of contract PCCDI 7/19.03.2018, code: PN-III P1-1.2-FPRD2017.

## 1. Introducere

Rețelele complexe sunt caracterizate de un număr mare de entități implicate, precum și de interacțiunile între acestea. În contextul analizei rezistenței antimicrobiene, entitățile corespunzătoare nodurilor unei rețele pot fi gene, proteine, specii sau medicamente, iar rețelele de interacțiuni între aceste noduri sunt construite pe baza informațiilor existente în diverse baze de date specifice (Comprehensive Antibiotic Research Database – CARD, National Antimicrobial Resistance Monitoring System for Enteric Bacteria - NARMS, etc.). În acest raport sunt prezentate rezultatele preliminare privind identificarea unor posibile noduri critice și comunități în cadrul rețelelor care descriu interacțiuni între entități implicate în analiza rezistenței antimicrobiene.

## 2. Noduri critice și comunități în rețele complexe

Rețelele complexe sunt utilizate pentru a descrie interacțiunile între entități, iar reprezentarea acestora este posibilă sub diverse forme, inclusiv cu grafuri orientate sau neorientate. Rețelele complexe modelate cu ajutorul grafurilor implică utilizarea noțiunilor de noduri și muchii. Nodurile conțin informații, iar muchiile reprezintă legăturile dintre aceste noduri. În cazul rețelelor biologice, nodurile sunt entitățile implicate, care pot fi gene, proteine, specii etc., iar muchiile sunt relațiile dintre ele.

Un nod critic (hub) este caracterizat prin faptul că ordinul său (numărul de conexiuni) este mai mare decât ordinul celorlalte noduri din cadrul rețelei.

Un alt element specific unei rețele complexe este noțiunea de *nod critic*, cunoscut și sub denumirea de *“hub”*. Acesta este caracterizat de un număr de legături care depășește cu mult media. În general, hub-urile au un număr semnificativ mai mare de legături în comparație cu alte noduri din rețea. Hub-urile au un impact semnificativ asupra topologiei rețelei. De asemenea, nodurile critice au un rol în distribuirea efectivă a informației în rețea. De exemplu, hub-urile sunt considerate super-răspânditori în contextul unei analize a răspândirii bolii sau a fluxului de informații.

*Subrețelele* sunt constituite din submulțimi de noduri împreună cu conexiunile corespondente și permit identificarea unor tipare de conexiune. *Comunitățile* sunt un caz particular de subrețea. O comunitate (Barabási, 2017) este un grup de noduri care au o probabilitate mai mare de conectare între ele decât la noduri din alte comunități. Comunitățile joacă un rol deosebit de important în înțelegerea bolilor umane (rețele biologice), dar și în cadrul rețelor sociale. În lucrarea (Chen et al., 2020) este descris MUMI (“Multitask Module Identification”), un algoritm propus pentru identificarea modulelor din rețele biologice prin detectarea suprapunerilor dintre module active (sau subrețele) și module topologice (sau comunități) simultan. Algoritmul MUMI este implementat în MATLAB, bazat pe un algoritm evolutiv de tip multifactorial.

### 3. Pachete R pentru analiza rețelelor

Limbajul R ([R tool, 2020](#)) furnizează librării sub forma de pachete pentru a prelucra, analiza și explora date ([Datar, 2019](#)), ([Dehmer et al., 2016](#)). Printre aceste pachete se numără și câteva care pot prelucra și analiza structuri de tipul rețelelor. De exemplu, pachetul “*igraph*” ([Csardi, 2006](#)) permite crearea și manipularea grafurilor și analiza rețelor. Acesta gestionează foarte bine grafuri de dimensiuni mari și oferă funcții pentru generarea de grafuri aleatoare, vizualizarea grafurilor, metode de centralizare și multe altele. În lucrarea ([Hartvigsen, 2011](#)) sunt analizate diferite rețele folosind pachetul “*igraph*” și este propusă o metodă pentru a studia modul în care vaccinarea poate modifica structura unei rețele de transmitere a bolii.

Pachetul “*Corbi*” ([Wu et al., 2020](#)) conține o colecție de instrumente specifice pentru bioinformatică. În cadrul acestui pachet sunt implementate funcții pentru *interogarea rețelelor* (localizarea unor subrețele în cadrul unei rețele), aliniere (identificarea celei mai mari subrețele pentru care există matching în două rețele). Funcțiile principale utilizate în interogare și identificarea de subrețele sunt: *net\_query()*, *net\_query\_batch()*, *net\_align()*, *get\_subnets()*, *extend\_subnets()*, *best\_subnets()*. Pachetul este specializat pentru analiza expresiei genice, astfel că include funcții pentru identificarea de biomarkeri (*markrank()*), funcții pentru analiza expresiilor diferentiale (*netDEG()*) dar și funcții pentru normalizarea datelor (*URG\_getFactor()*).

Pachetul “*netbioV*” ([Tripathi, 2014, 2020](#)) oferă funcții pentru vizualizarea de rețele biologice complexe permițând punerea în evidență a caracteristicilor structurale (asociate cu module din cadrul rețelelor). Pachetul permite specificarea explicită a modulelor dar și identificarea automată a acestora. Vizualizarea rețelelor poate fi controlată pe diferite nivele: global, modular, flux de informație, structura ierarhică. Este construit pornind de la “*igraph*”, prin urmare formatul datelor de intrare este similar celui din “*igraph*”. Funcția *plot.NetworkHubView()* permite vizualizarea nodurilor de tip hub realizând o plasare a nodurilor într-o structură circulară în funcție de gradul pe care îl au. Funcția *tkplot.netbioV()* din pachetul “*netbioV*” permite o reprezentare interactivă a rețelei.

Pachetul “*PCSF*” ([Akhmedov et al., 2017](#)) oferă funcții pentru interpretarea volumelor mari de date prin maparea acestora în structuri de tip rețea (corespunzătoare interacțiunilor de tip proteină-proteină sau genă-genă). În astfel de rețele nodurile corespund proteinelor sau genelor, iar muchiile corespund interacțiunilor dintre aceste entități. Muchiile pot avea asociate ponderi corespunzătoare gradului de interacțiune dintre noduri, iar nodurile au asociate valori de tip recompensă (prize) corelate cu gradul de exprimare a genei corespunzătoare, numărul de mutații implicate etc. Acest pachet este implementat pornind de la o metodă eficientă de identificare a subrețelelor care corespund unor interacțiuni relevante și se bazează pe un algoritm de rezolvare a problemei PCSF (“Prize-collecting Steiner Forest”) ([Akhmedov et al., 2016](#)). Metoda PCSF() din pachetul “*PCSF*” are ca parametri de intrare: o rețea de interacțiuni (obiect de tip “*igraph*”),

o listă de gene terminale, mai exact un vector numeric numit ca în rețeaua de interacțiuni, cu valori de tip recompensă care urmează să fie analizate în contextul PCSF, un număr de arbori care urmează să fie generați, o valoare de penalizare pentru nodurile de tip hub. Această funcție returnează o subrețea obținută prin aplicarea algoritmului PCSF asupra rețelei de interacțiuni dată ca parametru, rețeaua rezultată fiind un obiect "igraph". Alte funcții implementate în pachetul "PCSF" sunt: `construct_interactome()` pentru construirea unei rețele de interacțiuni, `enrichment_analysis()` pentru completarea rețelei cu informații de natură funcțională, `PCSF_rand()` pentru a genera (sub)rețele aleatoare utile în analiza robusteții soluției obținute, `plot.PCSF()` pentru vizualizarea interactivă a subrețelei rezultate și `plot.PCSFe()` pentru vizualizarea interactivă a subrețelei rezultate prin aplicarea scheme de îmbogățire cu informații funcționale.

Pachetul "visNetwork" (Almende et al., 2019) oferă funcții pentru a vizualiza rețele folosind "vis.js" (<http://visjs.org/>), o bibliotecă pentru javascript. Funcția `visNetwork()` implementată în pachetul "visNetwork" are ca parametri: o listă de noduri cu informațiile aferente, o listă de muchii cu informațiile aferente (coloanele "from" și "to" sunt necesare), precum și alți parametri. Această funcție returnează o rețea conform parametrilor selectați. Alte funcții auxiliare implementate în pachet sunt: `visNodes()` - opțiuni pentru noduri, `visEdges()` - opțiuni pentru muchii, `visGroups()` - opțiuni pentru grup, `visLegend()` - legendă, `visOptions()` - pentru specificarea unor opțiuni specifice, `visLayout()` și `visHierarchicalLayout()` pentru specificarea cadrului de vizualizare etc.

Pachetul "BioNet" (Beisser et al., 2010) permite pentru analiza funcțională a rețelelor biologice pornind de la date de tip microarray. În stabilirea scorurilor corespunzătoare nodurilor sunt folosite p-valori corespunzătoare unor teste statistice sau modele de regresie aplicate pentru analiza datelor. Pentru identificarea subrețelelor de scor maxim sunt utilizate metode de optimizare liniară în domeniul numerelor întregi (linear integer programming).

## 4. Studiu de caz

### 4.1 Set de date

Seturile de date au fost preluate din baza de date NARMS și conțin informații despre 329 izolate (în setul I), respectiv 3316 izolate (în setul II), de *Escherichia Coli*. În ambele seturi de date, fiecare izolat a fost clasificat ca rezistent sau susceptibil, pe baza pragurilor de concentrație minimă, la următoarele antibiotice: ampicilină - AMP, amoxicilină - acid clavulanic - AUG, ceftriaxonă - AXO, cloramfenicol - CHL, ciprofloxacina - CIP, trimetoprim - sulfametoxazol - COT, gentamicină - GEN, acid nalidixic - NAL, streptomycină - STR, tetraciclină - TET. Astfel, fiecare antibiotic este reprezentat ca o variabilă Bernoulli, având valorile 0 (susceptibil) sau 1 (rezistent). Aceste antibiotice se regăsesc abreviate (AMP, AUG, AXO, CHL, CIP, COT, GEN, NAL, STR, TET) în rețele prezentate mai jos.

Studiu privind identificarea elementelor cu rol critic în cadrul unei rețele de interacțiuni

Primul set de date conține cele zece antibiotice menționate anterior cu 329 observații aferente conform NARMS, iar rețeaua acestora este reprezentată sub forma de graf orientat în Figura 1, interacțiunile fiind stabilite pe baza regulilor de asociere extrase din date (cf. R 5.1.2 Modele și tehnici computaționale selectate pentru analiza datelor corelate cu rezistența antimicrobiană.)

**Set1:** 10 antibiotice, 329 observații – reguli de asociere extrase

AMP <- AXO, COT	COT <- AUG
AXO <- ---	GEN <- ---
AUG <- AXO	NAL <- ---
CHL <- STR	STR <- NAL, COT
CIP <- AMP, NAL	TET <- AMP, CHL, COT, NAL

**Set2:** 10 antibiotice, 3316 observații – reguli de asociere extrase

AMP -> AUG, CIP, COT, STR	AXO <- ----
AUG -> COT	AMP <- AXO, GEN
AXO -> AMP, AUG, CHL, CIP	AUG <- AMP, AXO
CHL -> CIP, TET	CHL <- AXO, STR
CIP -> ----	CIP <- AMP, AXO, CHL, GEN, NAL
COT -> NAL	COT <- AMP, AUG, STR, TET
GEN -> AMP, CIP	GEN <- ----
NAL -> CIP	NAL <- COT
STR -> CHL, COT, TET	STR <- AMP
TET -> COT	TET <- CHL, STR

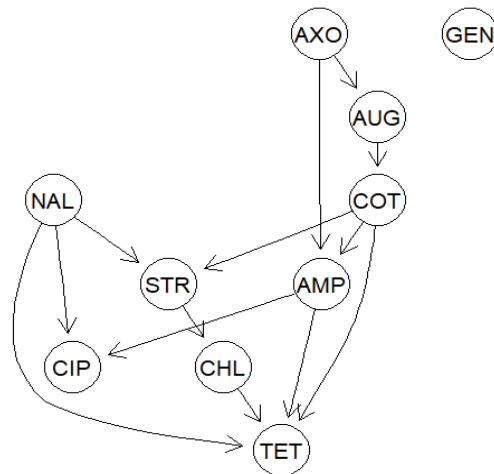


Figura1: Rețea de interacțiuni între 10 antibiotice extrase din 329 de observații aferente pentru setul I

Celălalt set de date conține cele zece antibiotice cu 3316 observații aferente conform NARMS, iar rețeaua de antibiotice este reprezentată sub forma de graf orientat în Figura 2, interacțiunile fiind reprezentate pe baza observațiilor.

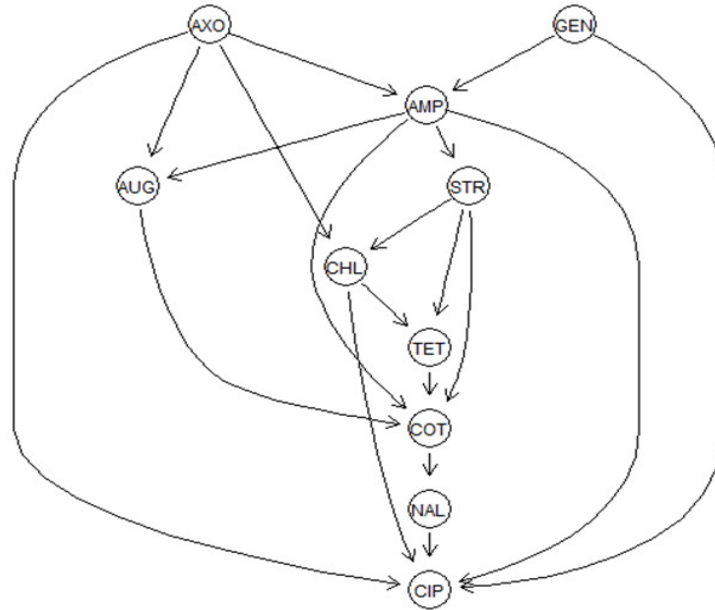


Figura2: Rețeaua de interacțiuni între 10 antibiotice extrase din 3316 de observații aferente pentru setul II

## 4.2 Metodologie de analiză

După generarea rețelelor de interacțiuni între antibiotice acestea au fost analizate cu scopul identificării de noduri critice (sau „hub-uri”), respectiv a comunităților. Prelucrările au fost realizate folosind pachetul R *igraph*. Pornind de la descrierea rețelei prin matrice de adiacență, obiectul de tip rețea din *igraph* a fost creat folosind funcția *graph.adjacency()*. Pentru identificarea nodurilor critice au fost calculate gradele nodurilor (numărul de noduri vecine) precum și distribuția acestora folosind funcția *degree()*, respectiv funcția *degree\_distribution()*.

Funcția *hub\_score()* din pachetul „*igraph*” calculează scorul de centralitate (Kleinberg, 1999), iar funcția *authority\_score()* calculează scorul nodurilor de tip autoritate. În cazul rețelelor reprezentate de grafuri neorientate, scorul utilizat în identificarea nodurilor de tip hub coincide cu cel corespunzător nodurilor de tip autoritate. Pentru detectarea autorităților a fost utilizată funcția *cluster\_edge\_betweenness()*. Această funcție primește ca parametru un graf și returnează comunități bazate pe calculul indicatorilor de tip „edge

betweenness”. Comunitățile pot fi detectate și folosind o strategie de tip greedy (Clauset et al., 2004), implementată în funcția *cluster\_fast\_greedy()*.

Altă metodă de detectare a comunităților este folosirea funcției *cluster\_label\_prop()*, care folosește un algoritm eficient bazat pe propagarea etichetelor (Raghavan et al., 2007). Ideea algoritmului este de a eticheta nodurile rețelei și de actualiza acest etichete printr-un proces de propaagare în rețea bazat pe un mecanism de votare care ia în considerare etichetele nodurilor din vecinătate.

### 4.3 Rezultate

În continuare, sunt prezentate câteva rezultate preliminare obținute în urma aplicării metodelor aferente pachetului „*igraph*” asupra celor două rețele de interacțiuni între antibiotice prezentate în secțiunea 4.1. (Figura 1 și Figura 2).

În cazul rețelei din Figura 1, au fost identificate ca noduri critice („hub-uri”): AMP, COT și NAL după cum este ilustrat și în Figura3 (gradul fiecărui nod este corelat cu dimensiunea discului care reprezintă nodul). În cadrul aceleiași rețele a fost identifica un nod de tip “autoritate” corespunzător antibioticului TET (Figura 4).

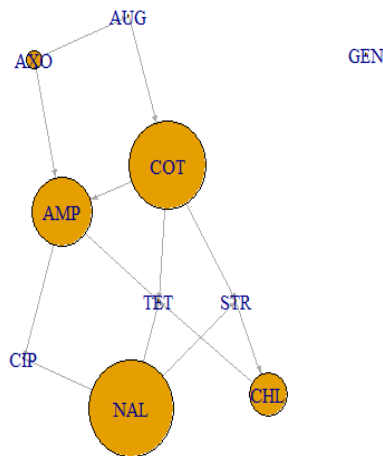


Figura 3: Noduri critice identificate în rețeaua de interacțiuni de antibiotice I

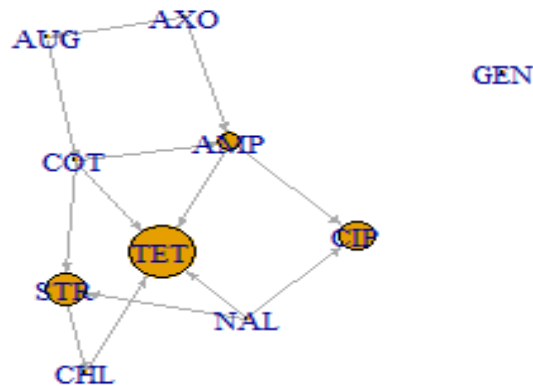


Figura 4: Autorități identificate în rețeaua de interacțiuni de antibiotice 1

Folosind diferitele variante de algoritmi de identificare a comunităților (Edge Betweenness, Greedy, Propagating Labels) au fost identificate câte 3 comunități în rețeaua corespunzătoare din Figura 1:

- *Edge Betweenness*: comunitate cu 7 noduri ("AMP", "CHL", "COT", "NAL", "STR", "CIP", "TET"), comunitate cu 2 noduri ("AUG" "AXO"), respectiv 1 nod ("GEN")
- *Greedy*: comunitate cu 4 noduri ("CHL", "NAL", "STR", "CIP"), comunitate cu 5 noduri ("AMP", "AUG", "AXO", "COT", "TET"), comunitate cu 1 nod ("GEN")
- *Propagating Labels*: comunitate cu 6 noduri ("AMP", "AUG", "AXO", "COT", "CIP", "TET"), comunitate cu 3 noduri ("CHL", "NAL", "STR"), comunitate cu 1 nod ("GEN")

Comunitățile identificate sunt prezentate în Figura 5.

Studiu privind identificarea elementelor cu rol critic în cadrul unei rețele de interacțiuni

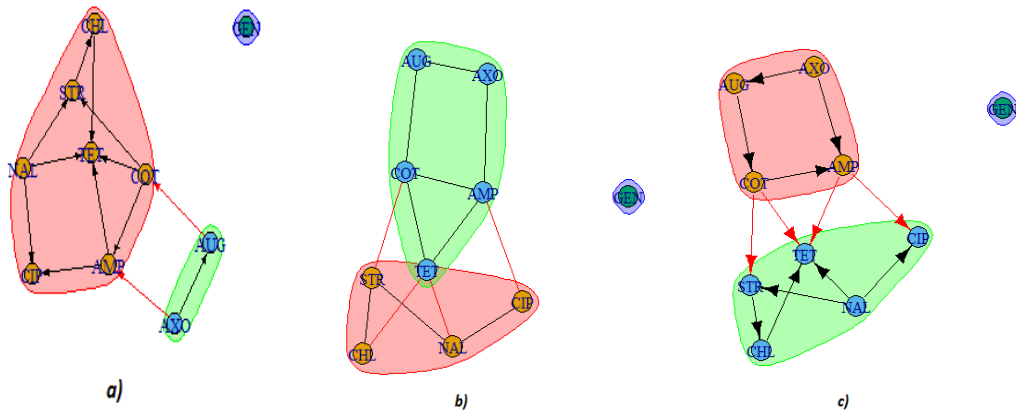


Figure 5: Comunități detectate în rețeaua de interacțiuni de antibiotice I: a) comunități detectate folosind algoritmul Edge Betweenness; b) comunități detectate folosind algoritmul Greedy; c) comunități detectate folosind algoritmul Propagating Labels.

Rețeaua de interacțiuni din Figura 2 este mai complexă decât cea din Figura 1 astfel este posibil ca numărul nodurilor critice și cel al comunităților să fie diferit. Figura 6 prezintă nodurile critice identificate, iar autoritățile identificate sunt prezentate în Figura 7.

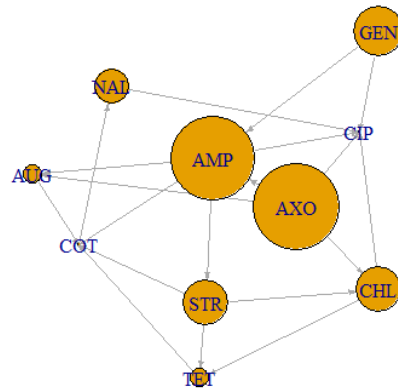


Figura 6: Noduri critice identificate în rețeaua de interacțiuni de antibiotice II

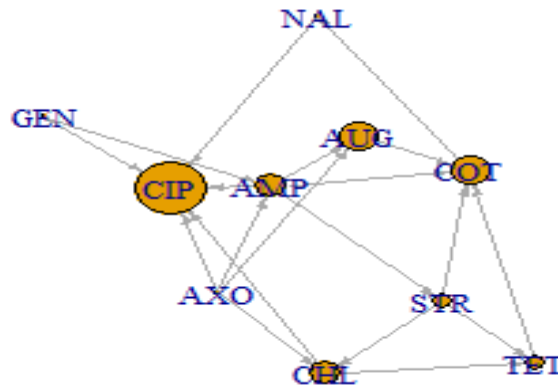


Figura 7: Autorități identificate în rețeaua de interacțiuni de antibiotice II

Comunitățile identificate în cadrul rețelei II variază de la 2 la 4 comunități în funcție de algoritmul folosit:

- *Edge Betweenness* (Figura 8.a): 4 comunități dintre două cu câte un nod ("AMP", respectiv "AUG"), și două cu câte 4 noduri ("AXO", "GEN", "NAL", "CIP"), respectiv ("CHL", "COT", "STR", "TET").
- *Greedy* (Figura 8.b): 3 comunități, dintre care una cu 5 noduri ("AMP", "AUG", "AXO", "GEN", "CIP"), una cu 3 noduri ("CHL", "STR", "TET"), și una cu 2 noduri ("COT", "NAL").
- *Propagating Labels* (Figura 8.c): 2 comunități, dintre care o comunitate având 2 noduri ("AMP", "GEN") și alta având 8 noduri ("AUG", "AXO", "CHL", "COT", "NAL", "STR", "CIP", "TET").

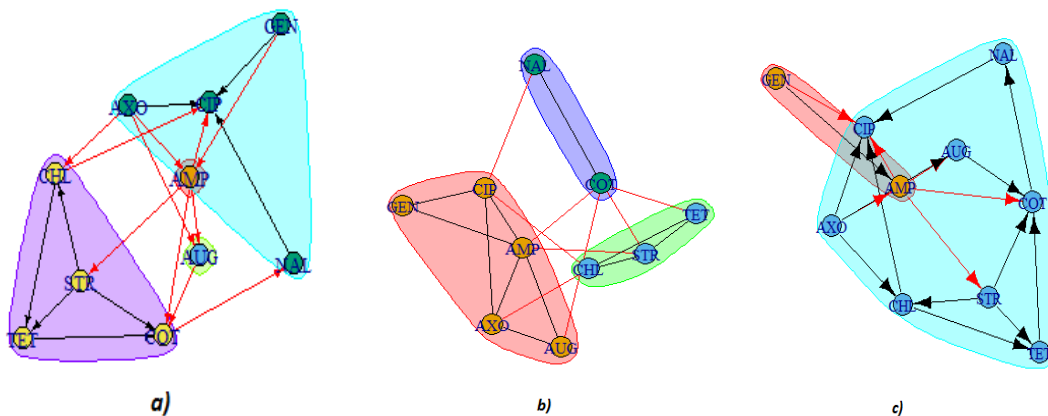


Figura 8: Comunități detectate în rețeaua de interacțiuni de antibiotice II: a) comunități detectate folosind algoritmul Edge Betweenness; b) comunități detectate folosind algoritmul Greedy; c) comunități detectate folosind algoritmul Propagating Labels.

#### 4.4 Discuție

Limbajul R oferă librării care facilitează analiza grafurilor și a rețelor. Pachetul „*abn*” este pentru generarea de grafuri bazate pe modele bayesiene, iar pachetul „*igraph*” este pentru analiza rețelor. Grafurile generate cu ajutorul pachetului „*abn*” depind de volumul de date. În acest caz, graful din Figura 1 are 329 observații și graful din Figura 2 are 3316 observații, numărul de noduri cu informații despre antibiotice fiind identic. Graful din Figura 1 are 26 de muchii (13 muchii de tip „IN”, 13 muchii de tip „OUT”), pe când graful din Figura 2 are 38 de muchii (19 muchii de tip „IN”, 19 muchii de tip „OUT”).

Asemănările și diferențele dintre cele două rețele de interacțiuni referitoare la nodurile critice (sau hub-uri) identificate sunt prezentate în Figura 9, respectiv cele pentru autoritățile identificate în Figura 10.

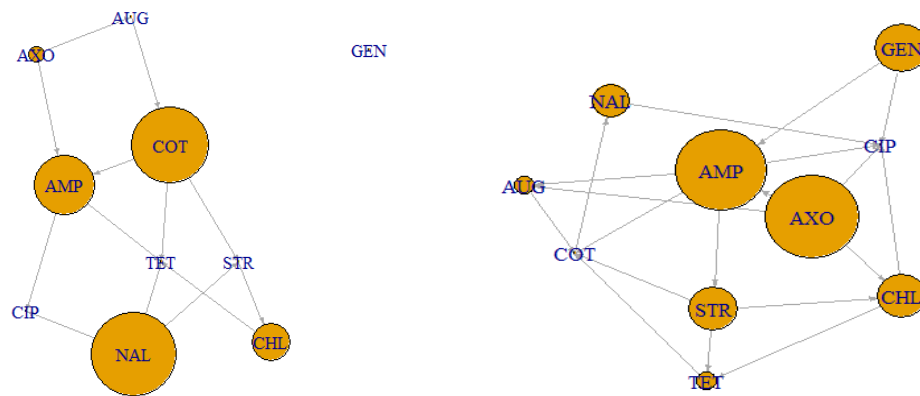


Figura 9: Noduri critice identificate în rețeaua de interacțiuni de antibiotice I (stânga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

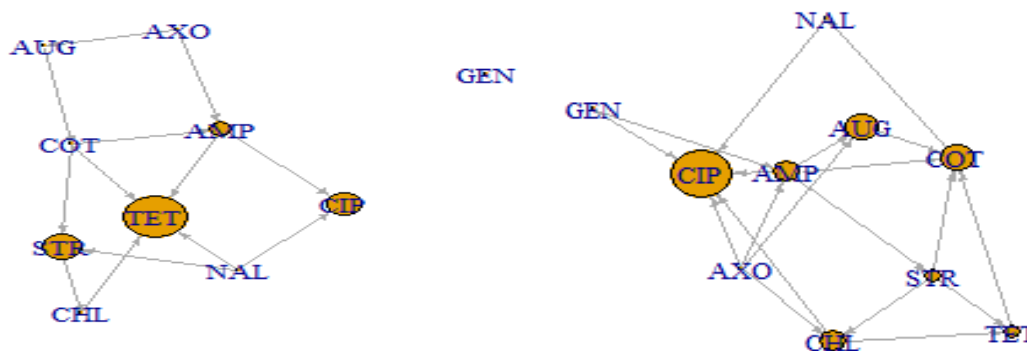


Figura 10: Autorități identificate în rețeaua de interacțiuni de antibiotice I (stânga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

În continuare, sunt prezentate comparații dintre cele două rețele din punctul de vedere al comunităților identificate. Numărul comunităților identificate în cadrul rețelei din Figura 1 variază de la 2 la 4 în funcție de algoritmul folosit, iar, în cazul rețelei din Figura 2, au fost identificate 3 comunități de către fiecare din cei trei algoritmi. O comparație privind comunitățile identificate prin aplicarea algoritmului Edge Betweenness asupra rețelelor de interacțiuni este prezentată în Figura 11 și Figura 12, unde atât numărul de comunități identificate cât și numărul membrilor aferent comunităților diferă de la o rețea la alta.

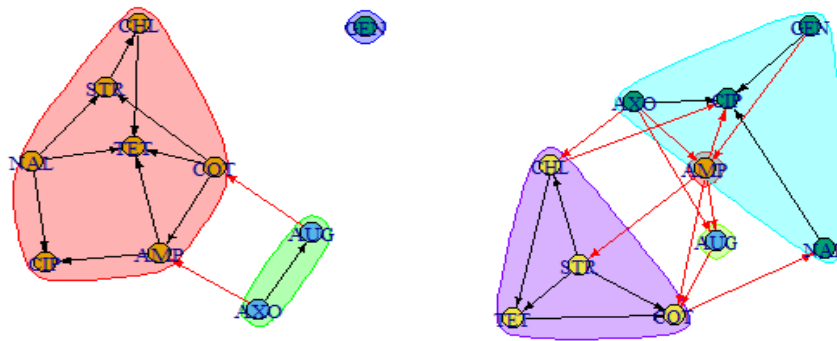


Figura 11: Comunități detectate folosind algoritmul Edge Betweenness în rețeaua de interacțiuni de antibiotice I (stânga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

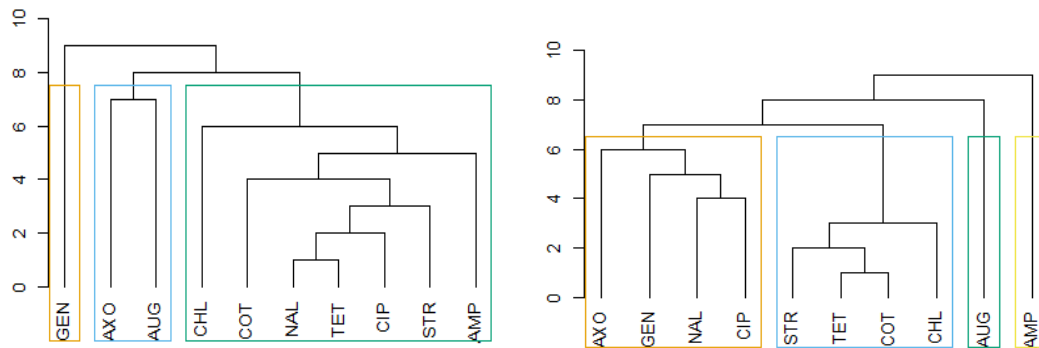


Figura 12: Dendrograma - comunități detectate folosind Edge Betweenness în rețeaua de interacțiuni de antibiotice I (stânga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

O comparație privind comunitățile identificate prin aplicarea algoritmului greedy asupra rețelelor de interacțiuni este prezentată în Figura 13, unde numărul de comunități identificate este 3 pentru ambele rețele, dar numărul membrilor aferent comunităților diferă de la o rețea la alta.

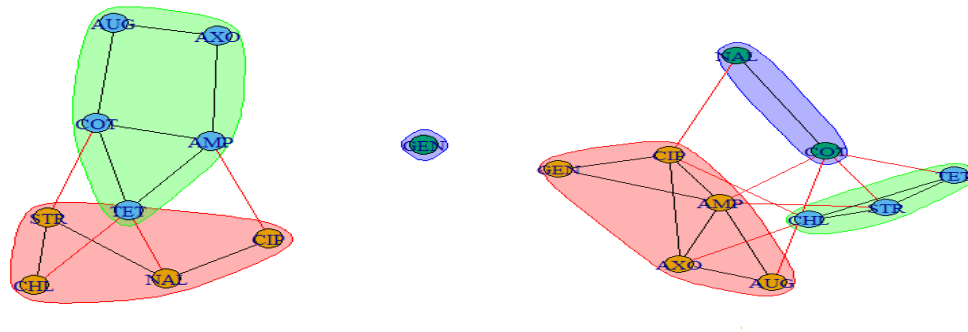


Figure 13: Comunități detectate folosind algoritmul greedy în rețeaua de interacțiuni de antibiotice I (stânga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

Altă comparație privind comunitățile identificate prin aplicarea **propagating labels** asupra rețelelor de interacțiuni este prezentată în Figura 14, unde atât numărul de comunități identificate cât și numărul membrilor aferent comunităților diferă de la o rețea la alta.

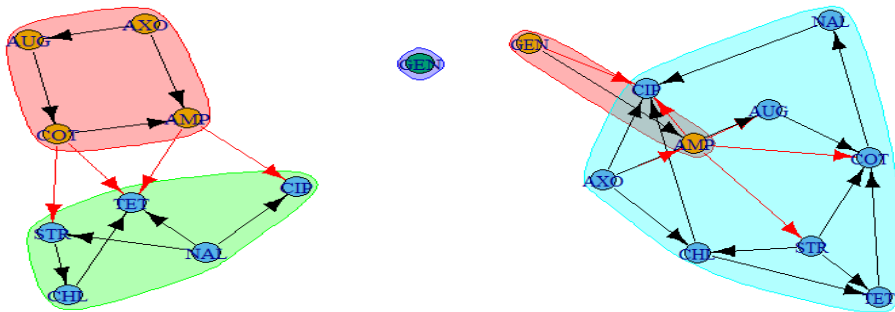


Figura 14: Comunități detectate folosind Propagating Labels în rețeaua de interacțiuni de antibiotice I (stanga), respectiv în rețeaua de interacțiuni de antibiotice II (dreapta)

## 5. Concluzii

Pachetul R *igraph* este un instrument flexibil pentru analiza rețelelor permițând atât identificarea cât și vizualizarea nodurilor critice (hub-uri și autorități) precum și a comunităților (sub-rețele puternic conectate) putând fi utilizat pentru rețele care modelează diferite tipuri de interacțiuni relevante în contextul analizei rezistenței la antibiotice.

## Bibliografie:

1. (Akhmedov et al., 2017) Murodzhon Akhmedov, Amanda Kedaigle, Renan Escalante, Roberto Montemanni, Francesco Bertoni, Ernest Fraenkel, Ivo Kwee, PCSF: Network-based interpretation of high throughput data, R package version 0.99.1, 2017.
2. (Akhmedov et al., 2016) Murodzhon Akhmedov, Ivo Kwee, Roberto Montemanni, A divide and conquer heuristic algorithm for the Prize-collecting Steiner Tree Problem, Computers and Operations Research, Vol. 70, 2016, pag. 18–25, DOI: 10.1016/j.cor.2015.12.015.
3. (Almende et al., 2019) B.V. Almende, Thieurmel Benoit, Robert Titouan, visNetwork: Network Visualization using 'vis.js' Library, R package version 2.0.9, 2019, URL: <https://CRAN.R-project.org/package=visNetwork>.
4. (Barabási, 2017) Albert-László Barabási, Network Science, 2017.
5. (Beisser et al., 2010) Daniela Beisser, Gunnar W. Klau, Thomas Dandekar, Tobias Mueller, Marcus Dittrich, BioNet: an R-package for the Functional Analysis of Biological Networks, Bioinformatics, Vol. 26, No. 8, 2010, pag. 1129-1130.
6. (Chen et al., 2020) Weiqi Chen, Zexuan Zhu, Shan He, MUMI: Multitask Module Identification for Biological Networks, IEEE Transactions On Evolutionary Computation, Vol. 24, No. 4, 2020.
7. (Clauset et al., 2004) A Clauset, MEJ Newman, C Moore, Finding community structure in very large networks, <http://www.arxiv.org/abs/cond-mat/0408187>.
8. (Csardi, 2006) Gabor Csardi, Tamas Nepusz: The igraph software package for complex network research, InterJournal, Complex Systems 1695, 2006, URL: <https://igraph.org>.
9. (Datar, 2019) Radhika Datar, Harish Garg, Hands-On Exploratory Data Analysis with R: Become an expert in exploratory data analysis using R packages, 2019.
10. (Dehmer et al., 2016) Matthias Dehmer, Yongtang Shi, Frank Emmert-Streib, Computational Network Analysis with R Applications in Biology, Medicine and Chemistry, Wiley-VCH Verlag GmbH & Co. KGaA, Vol. 7, 2016, ISBN: 978-3-527-33958-7.
11. (Hartvigsen, 2011) Gregg Hartvigsen, Using R to Build and Assess Network Models in Biology, Mathematical Modelling of Natural Phenomena, Vol. 6, No. 6, 2011, pag. 61 - 75, DOI: 10.1051/mmnp/20116604.
12. (Kleinberg, 1999) J. Kleinberg, Authoritative sources in a hyperlinked environment, Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997.
13. (Kratzer et al., 2019) Gilles Kratzer, Marta Pittavino, Fraser Iain Lewis, Reinhard Furre, abn: an R package for modelling multivariate data using additive Bayesian networks, R package version 2.2, 2019, URL: <https://CRAN.R-project.org/package=abn>.
14. (Raghavan et al., 2007) U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys Rev E 76, 036106, 2007.
15. (R tool, 2020) R Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2020, URL: <https://www.R-project.org/>.
16. (Tripathi, 2014, 2020) Shailesh Tripathi, Frank Emmert-Streib, NetBioV: an R package for visualizing large network data in biology and medicine; Bioinformatics doi:10.1093/bioinformatics/btu384, 2014; NetBioV: A package for visualizing complex biological network, R package version 1.22.0, 2020, URL: <http://www.bio-complexity.com>.
17. (Wu et al., 2020) Ling-Yun Wu, Qiang Huang, Duanchen Sun, Corbi: Collection of Rudimentary Bioinformatics Tools, R package version 0.6-0, 2020, URL: <https://CRAN.R-project.org/package=Corbi>.