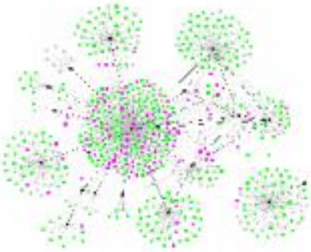
Βιοπληροφορική ΙΙ

Παντελής Μπάγκος
Αναπληρωτής Καθηγητής

Πανεπιστήμιο Θεσσαλίας

Λαμία, 2015

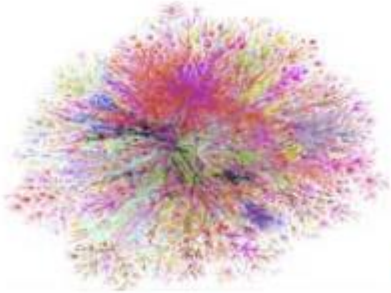
Network Data



Disease Spread

Electronic Circuit

Food Web



Internet

Social Network

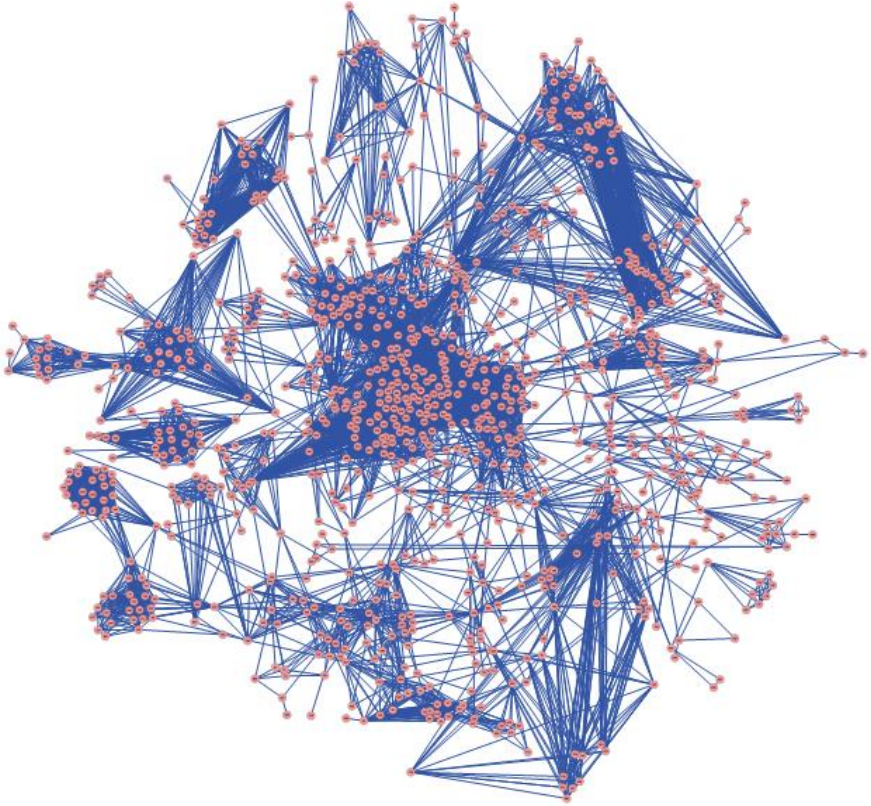
Τύποι βιολογικών δικτύων

1. Δίκτυα Πρωτεινικών αλληλεπιδράσεων

(Protein-protein interactions PPIs)

•Οι πρωτεΐνες είναι οι κόμβοι του δικτύου και οι αλληλεπιδράσεις τους οι ακμές

Βάσεις δεδομένων:
Yeast Proteome Database (YPD), Munich Information Center for Protein Sequences (MIPS), Molecular Interaction (MINT), Databae of Interacting Proteins (DIP), BioGRID, Human Protein Reference Database (HPRD)

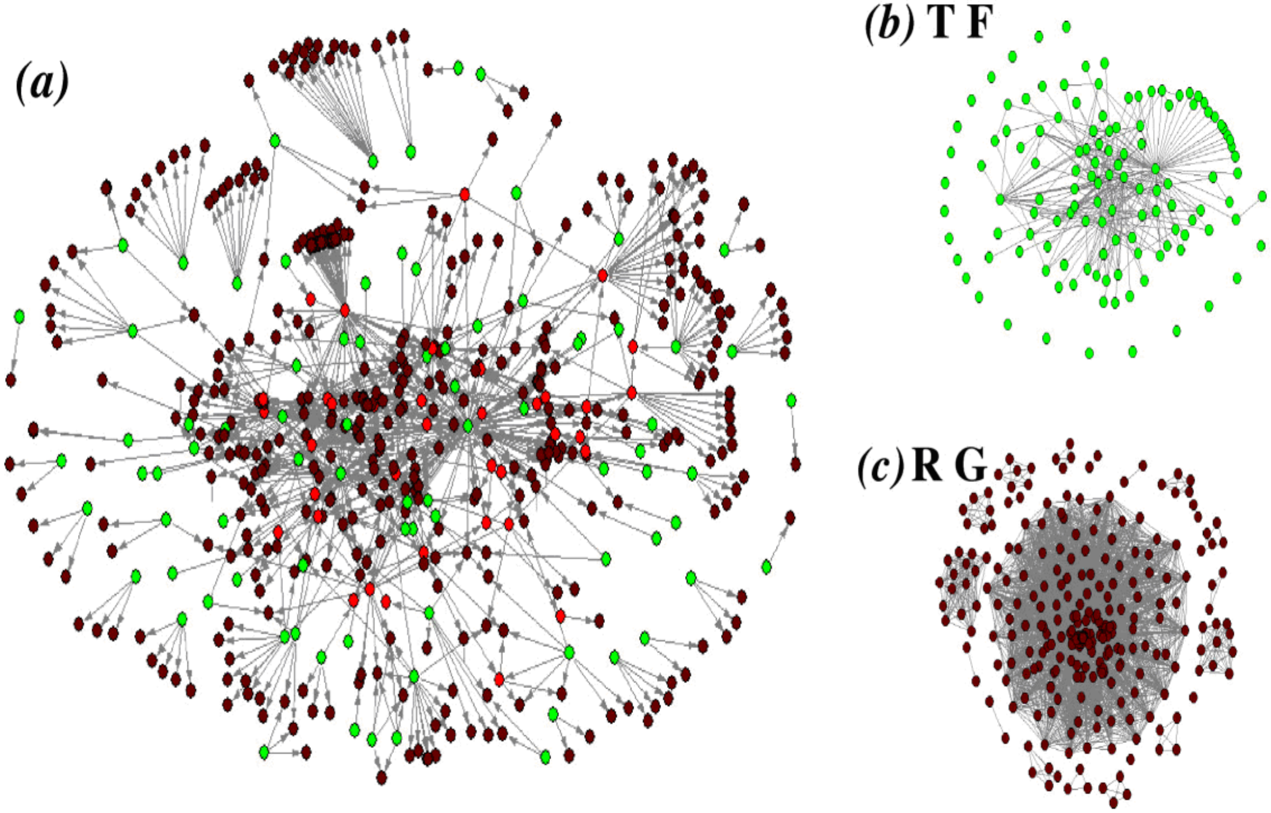


Saccharomyces cerevisiae (yeast) PPI network
1,004 nodes and 8,323 edges (17)

2.Μεταγραφικά ρυθμιστικά δίκτυα (Transcriptional- Regulatory networks (GRNs)

- Μοντελοποιείται ο τρόπος που οι πρωτεΐνες και άλλα βιομόρια εμπλέκονται στην διαδικασία της έκφρασης των γονιδίων

Βάσεις δεδομένων:
JASPAR, TRANSFAC, B-cell interactome (BCI) , Phospho.ELM
NetPhorest, PHOSIDA



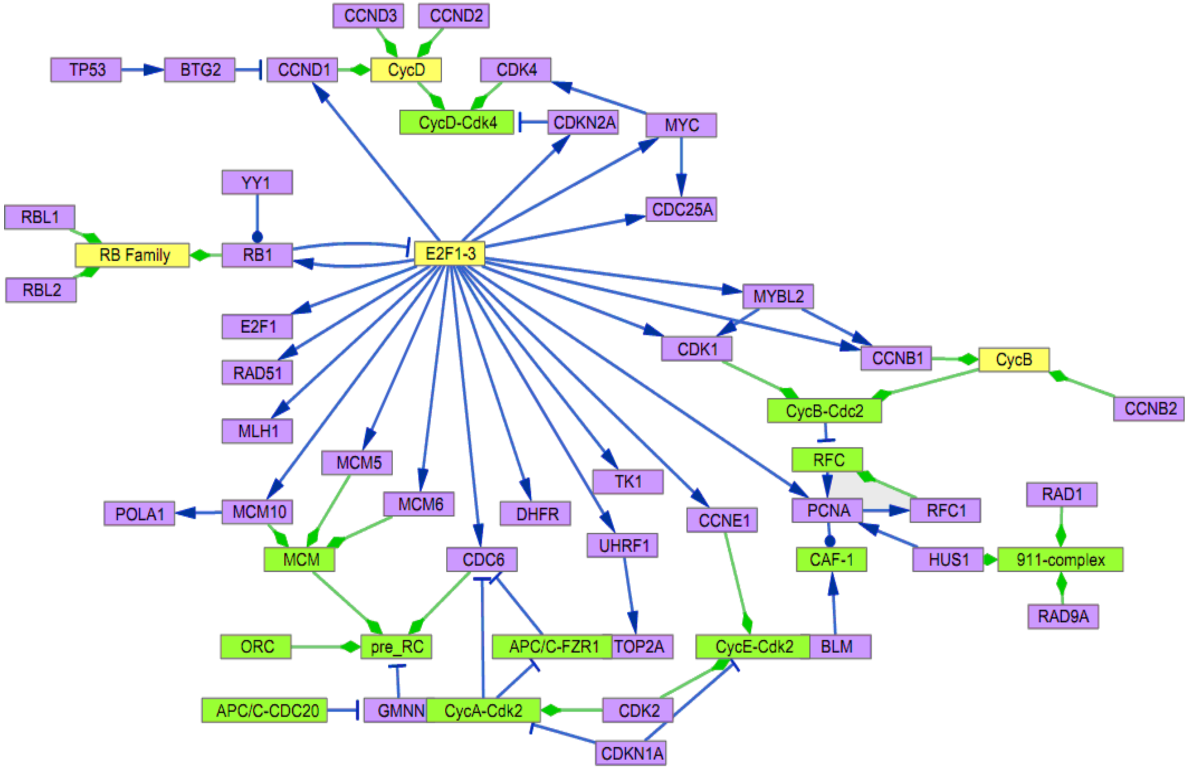
E. coli. Guzmán-Vargas and Santillán *BMC Systems Biology* 2008 **2**:13

3. Δίκτυα Μεταγωγής Σήματος

(Signal trasduction networks)

- Αναπαριστάται ο τρόπος μετάδοσης του σήματος από τον έξωκυττάριο στον ενδοκυττάριο χώρο, είτε στο εσωτερικό του κυττάρου.

Βάσεις δεδομένων:
MiST, TRANSPATH, Spike, Kegg



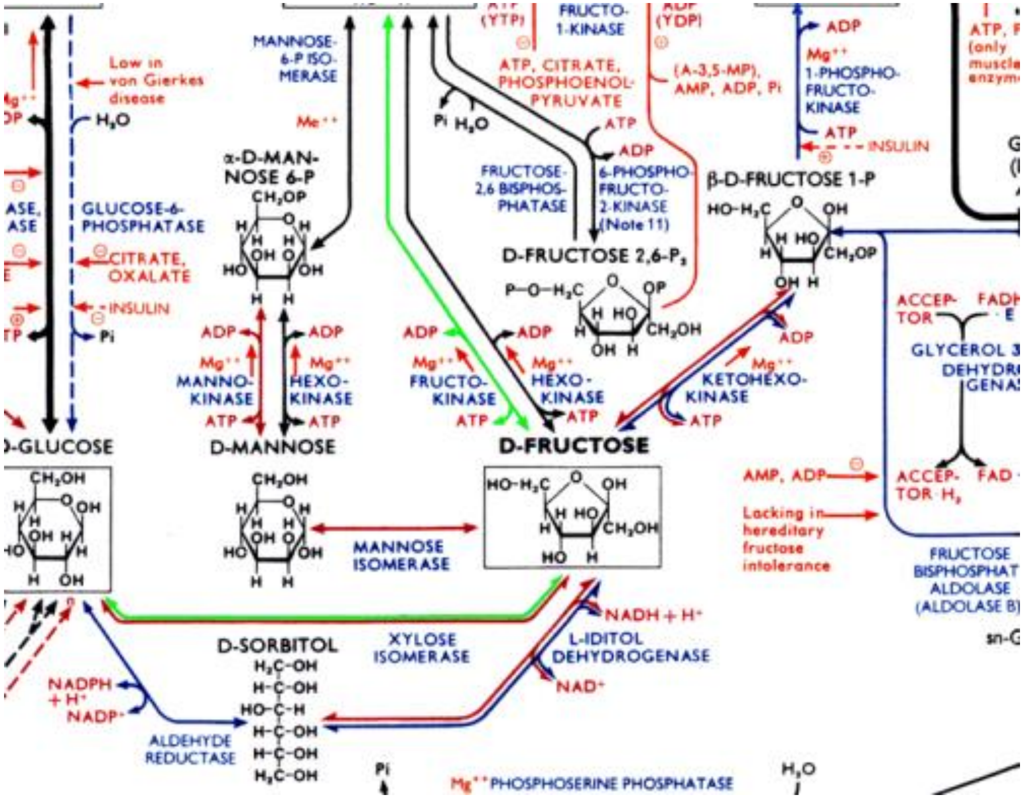
G1-S Phase of the Cell from Spike DataBase (PMID: 18289391)

4. Μεταβολικά – Βιοχημικά δίκτυα

(Metabolic and Boichemical networks)

- Μεταβολικό μονοπάτι θεωρείται μια σειρά από χημικές αντιδράσεις μέσα στο κύτταρο σε διαφορετικές χρονικές καταστάσεις

Βάσεις δεδομένων:
Kyoto Encyclopedia of Genes and Genomes (KEGG), TRANSPATH, EcoCyc, metaTIGER



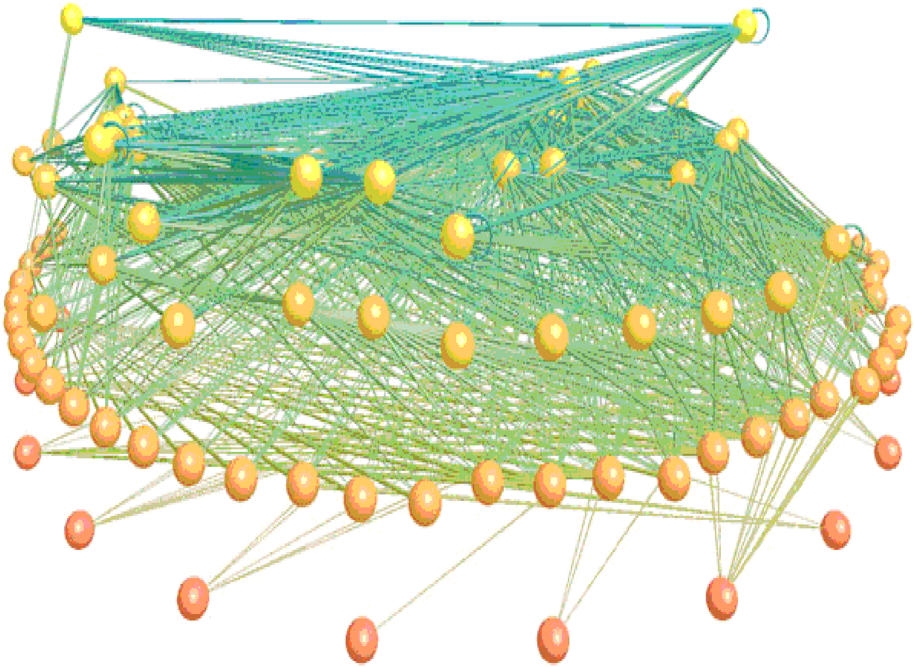
A portion of a metabolic network. (From Biochemical Pathways, Roche Applied Science, http://www.expasy.org/tools/pathways/)

5. Οικολογικά δίκτυα-δίκτυα διατροφικών αλυσίδων (Food Webs)

- Αναπαριστώνται οι βιοτικές αλληλεπιδράσεις σε ένα οικοσύστημα.

Τα είδη των οργανισμών που βρίσκονται σε ένα οικοσύστημα συνδέονται με αλληλεπιδράσεις κατά ζεύγη και μπορεί να είναι είτε τροφικές είτε συμβιωτικές

Βάσεις δεδομένων:
Iwdb interaction Web DataBase, Food Web Bank

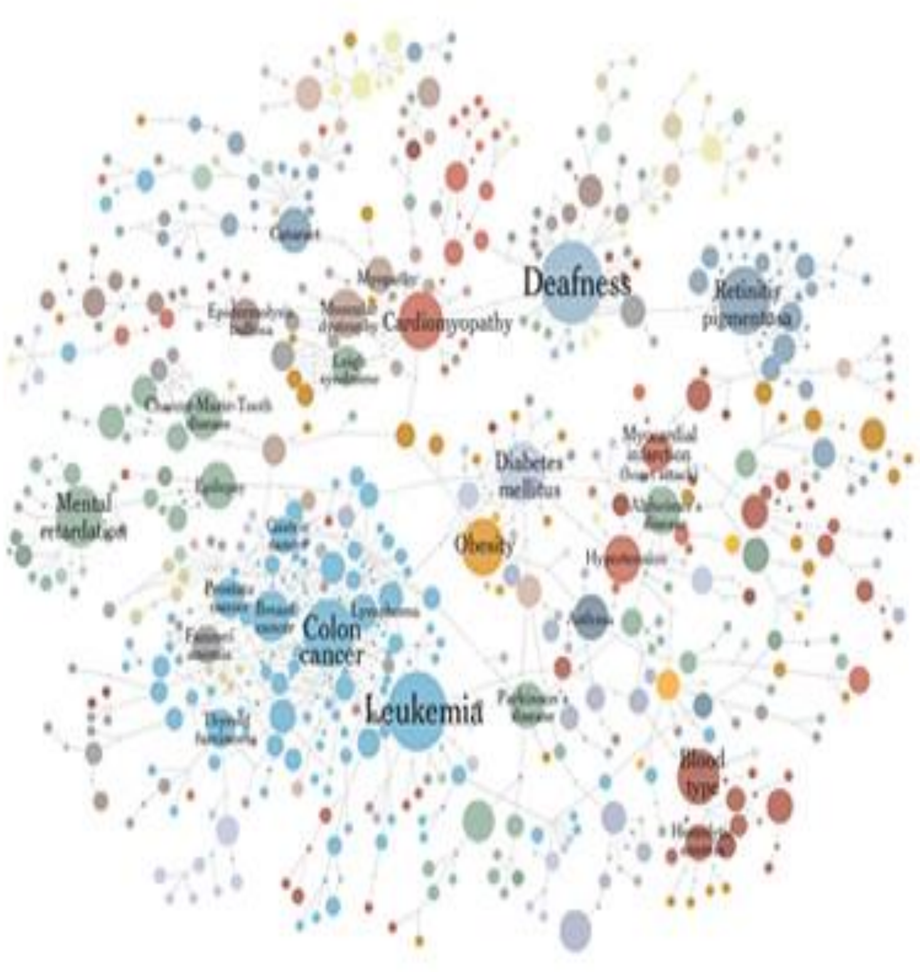


Food Web, el Verde (www.foodwebs.org, Yoon et al. 2004)

6. Δίκτυα ασθενειών (Diseases networks)

- Δίνουν πληροφορίες για την προέλευση ασθενειών

Βάσεις δεδομένων: Online Mendelian Inheritance in Man (OMIM)



Human disease network (Matthew Bloch, Jonathan Corum) PMID:17502.601

Άλλα βιολογικά δίκτυα

7. Δίκτυα θεραπείας ασθενειών (treatment networks)

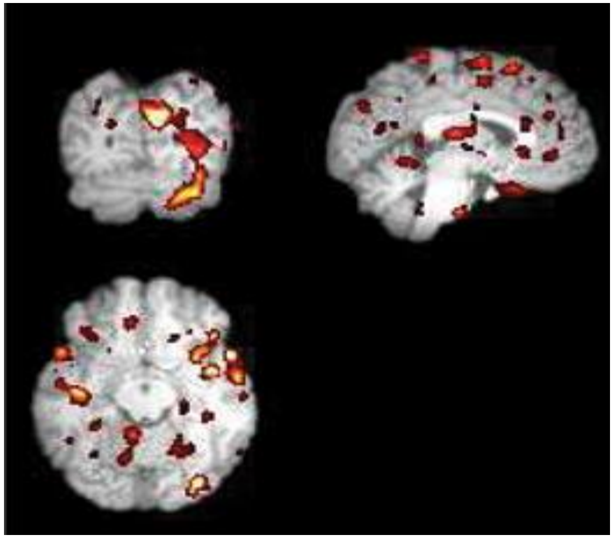
- Πληροφορίες για την επικοινωνία φαρμάκων και θεραπειών κάποιας ασθένειας

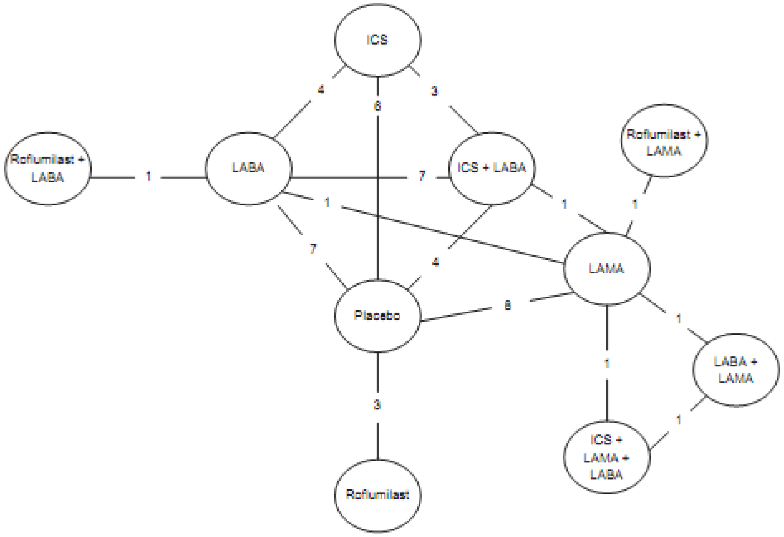
8. Νευρωνικά δίκτυα (neural networks)

- Πληροφορίες για τον τρόπο μετάδοσης σημάτων στο νευρικό σύστημα

9. Δίκτυα εγκεφάλου (Brain networks)

- Ο τρόπος που τα διάφορα σημεία του εγκεφάλου ενεργοποιούνται- επικοινωνούν





Brain network: directed links in large scale functional networks
G.A. Cecchi, A.R. Rao, M.V. Centeno, M. Baliki, A.V. Apkarian
& D.R. Chialvo, BMC Cell Biology 8(Suppl 1):S5 (2007)

network of 10 treatments involved in the MTC analyses of the COPD data: chronic obstructive pulmonary disease;

Παραδείγματα

- Steven H. Strogatz Exploring complex networks Nature 410, 268-276(8 March 2001)

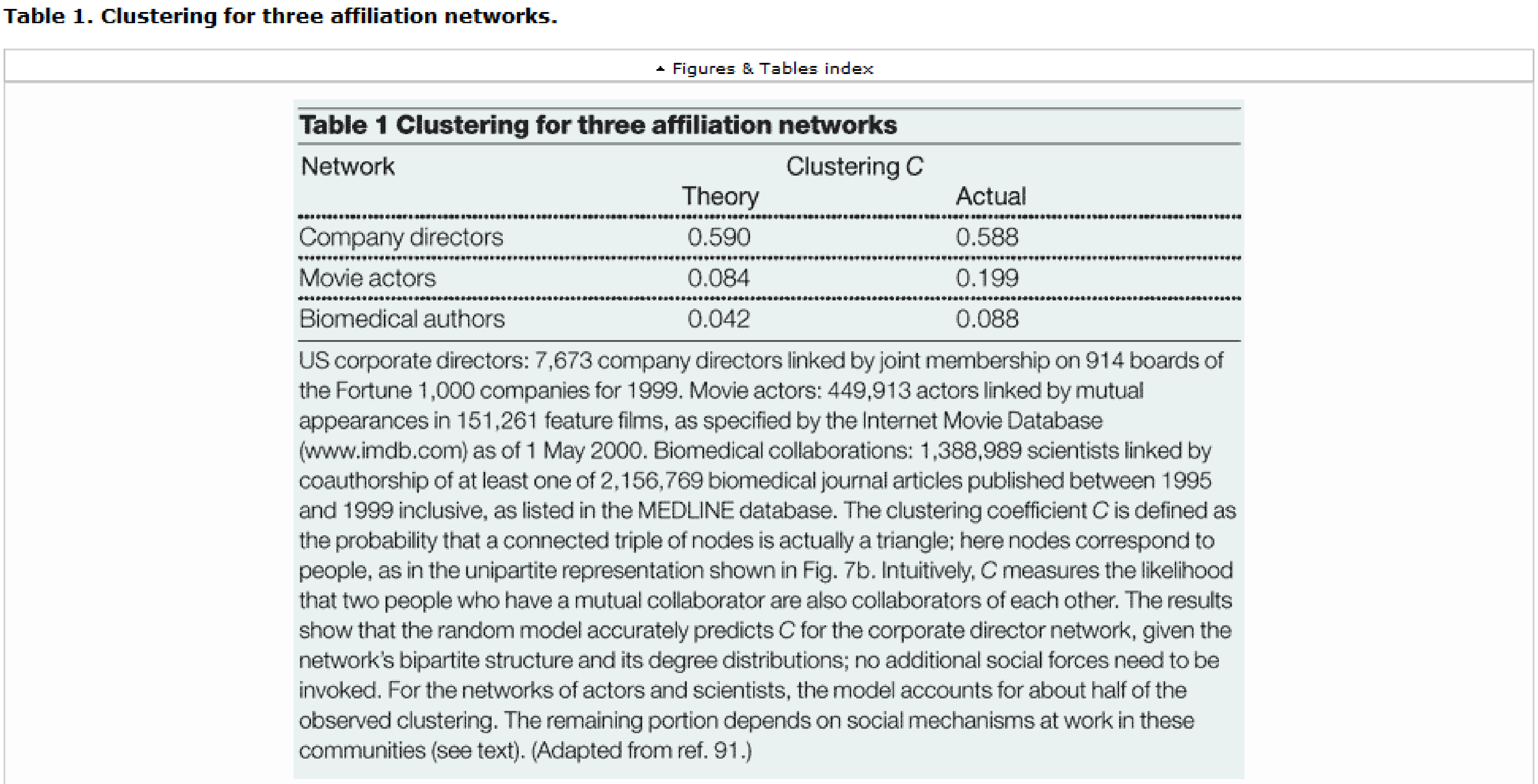


Table 1. Clustering for three affiliation networks.

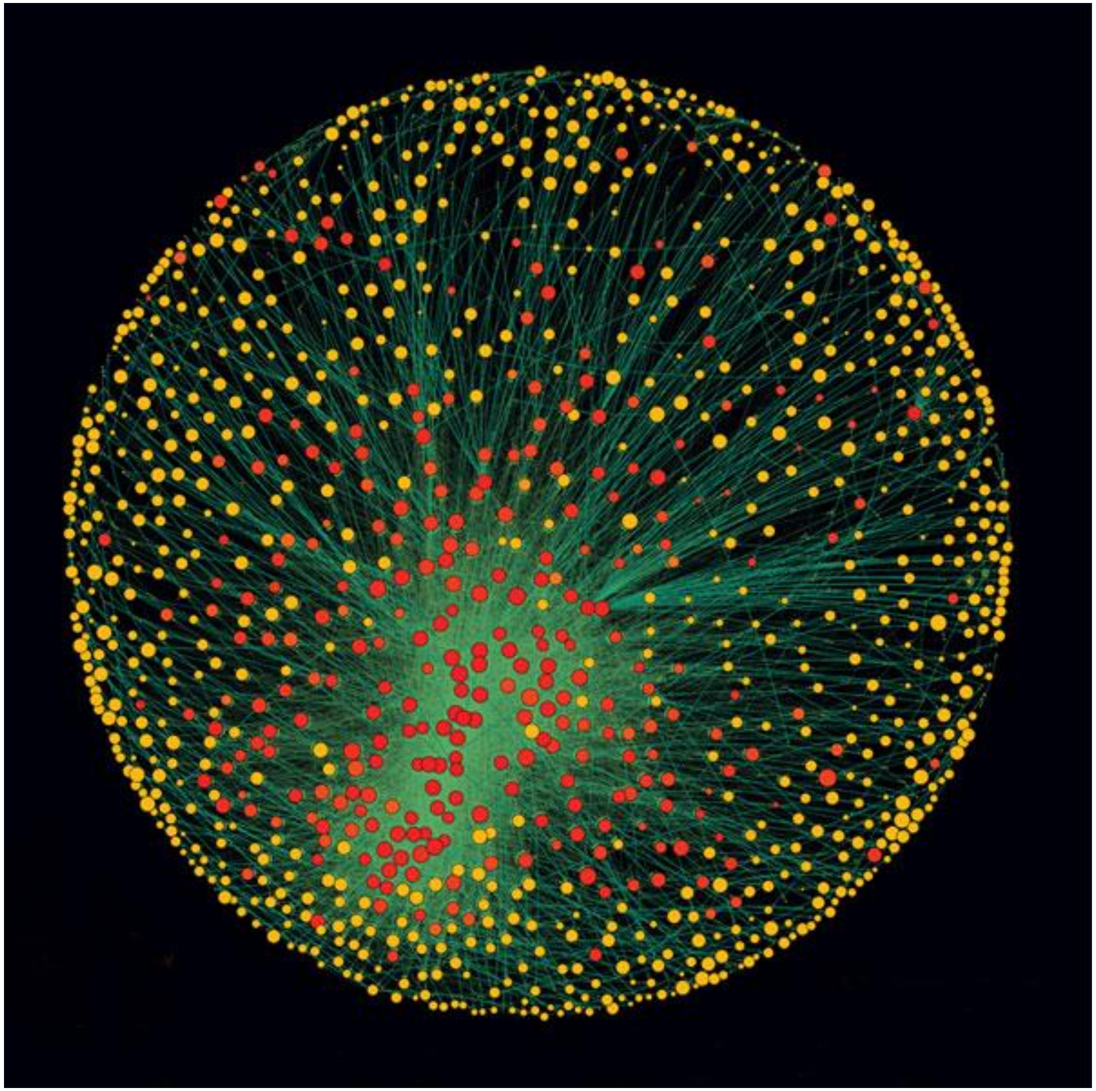
▲ Figures & Tables index

Table 1 Clustering for three affiliation networks

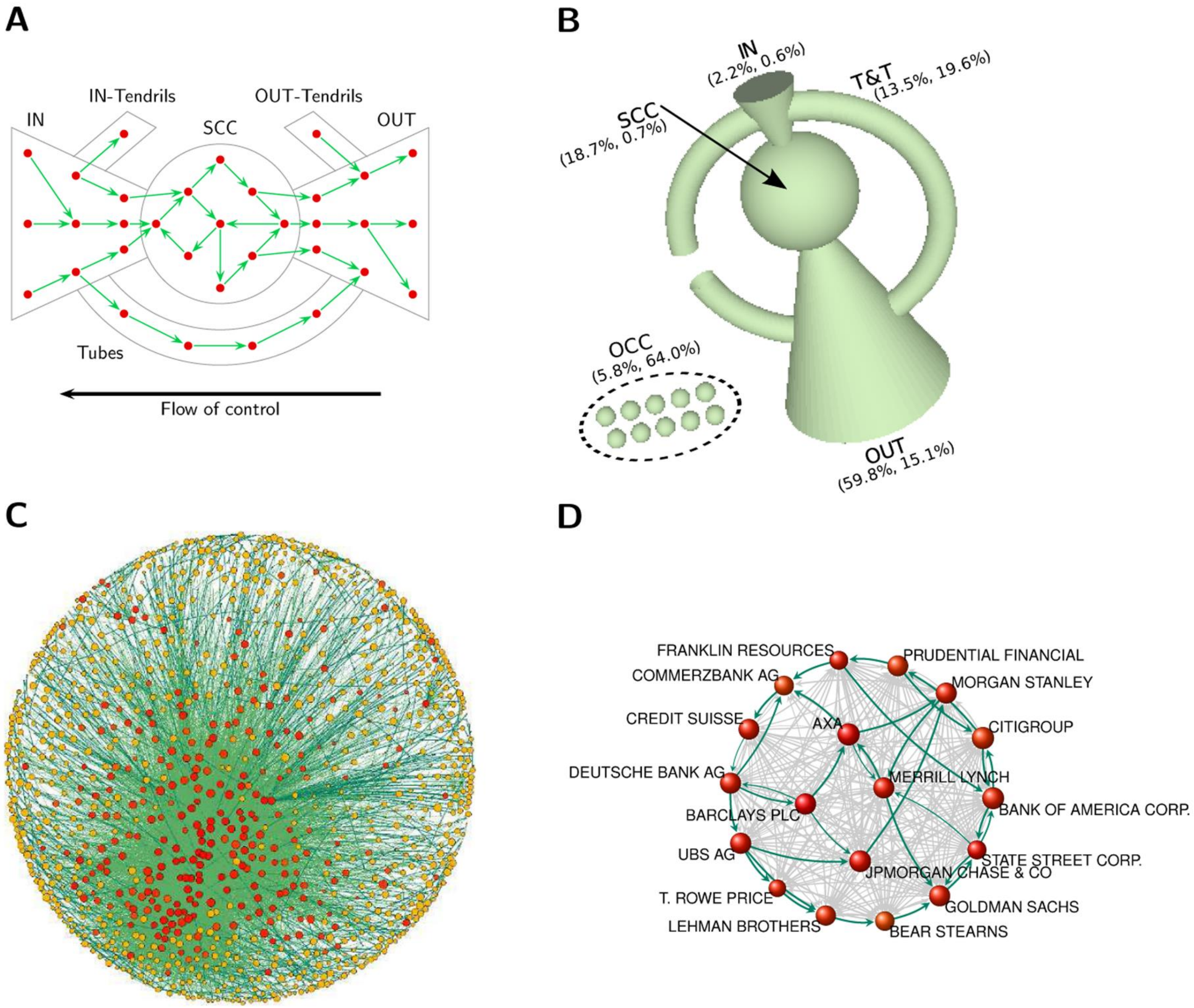
Network	Clustering *C*	
	Theory	Actual
Company directors	0.590	0.588
Movie actors	0.084	0.199
Biomedical authors	0.042	0.088

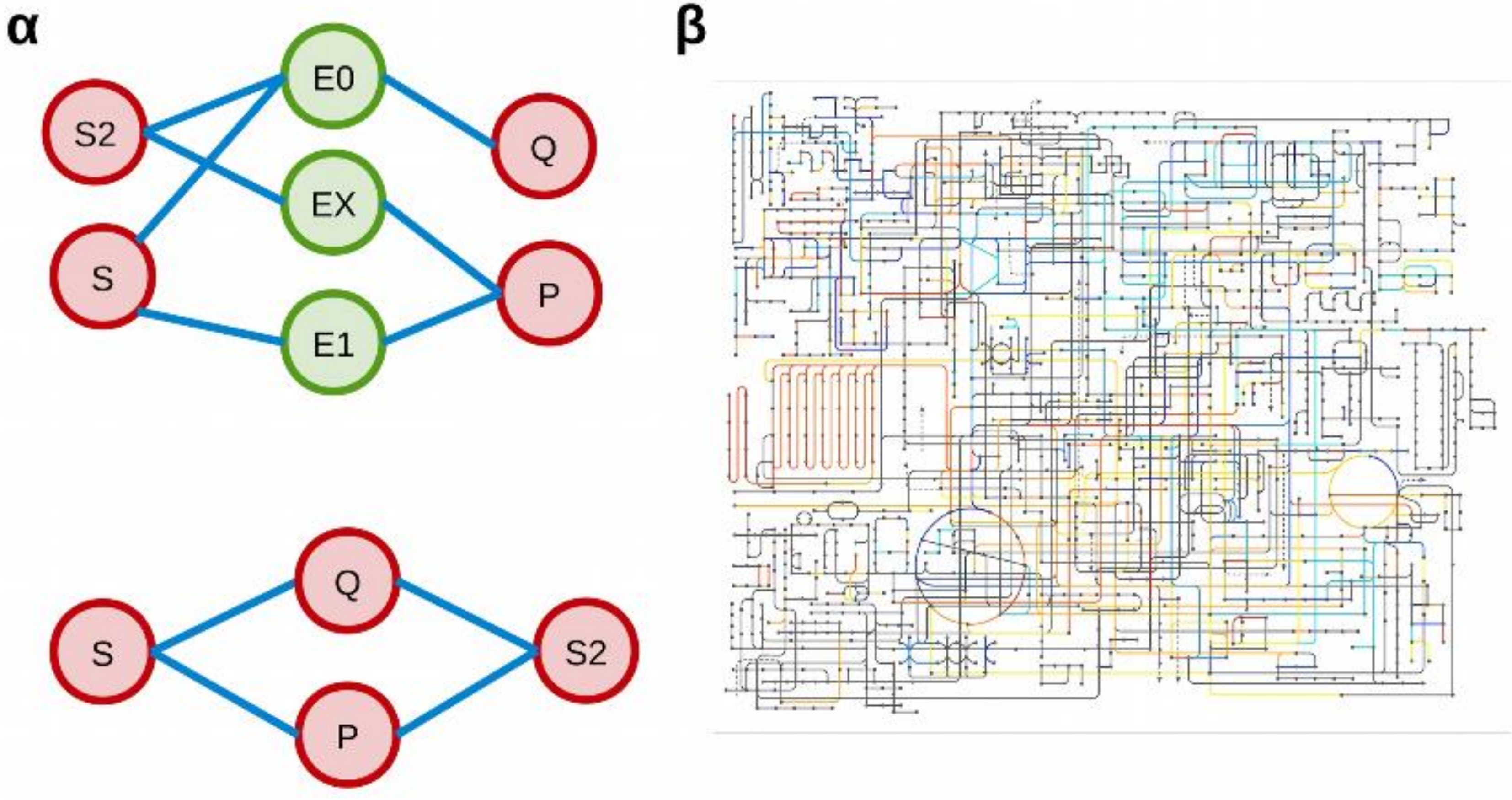
US corporate directors: 7,673 company directors linked by joint membership on 914 boards of the Fortune 1,000 companies for 1999. Movie actors: 449,913 actors linked by mutual appearances in 151,261 feature films, as specified by the Internet Movie Database (www.imdb.com) as of 1 May 2000. Biomedical collaborations: 1,388,989 scientists linked by coauthorship of at least one of 2,156,769 biomedical journal articles published between 1995 and 1999 inclusive, as listed in the MEDLINE database. The clustering coefficient *C* is defined as the probability that a connected triple of nodes is actually a triangle; here nodes correspond to people, as in the unipartite representation shown in Fig. 7b. Intuitively, *C* measures the likelihood that two people who have a mutual collaborator are also collaborators of each other. The results show that the random model accurately predicts *C* for the corporate director network, given the network's bipartite structure and its degree distributions; no additional social forces need to be invoked. For the networks of actors and scientists, the model accounts for about half of the observed clustering. The remaining portion depends on social mechanisms at work in these communities (see text). (Adapted from ref. 91.)

the capitalist network that runs the world

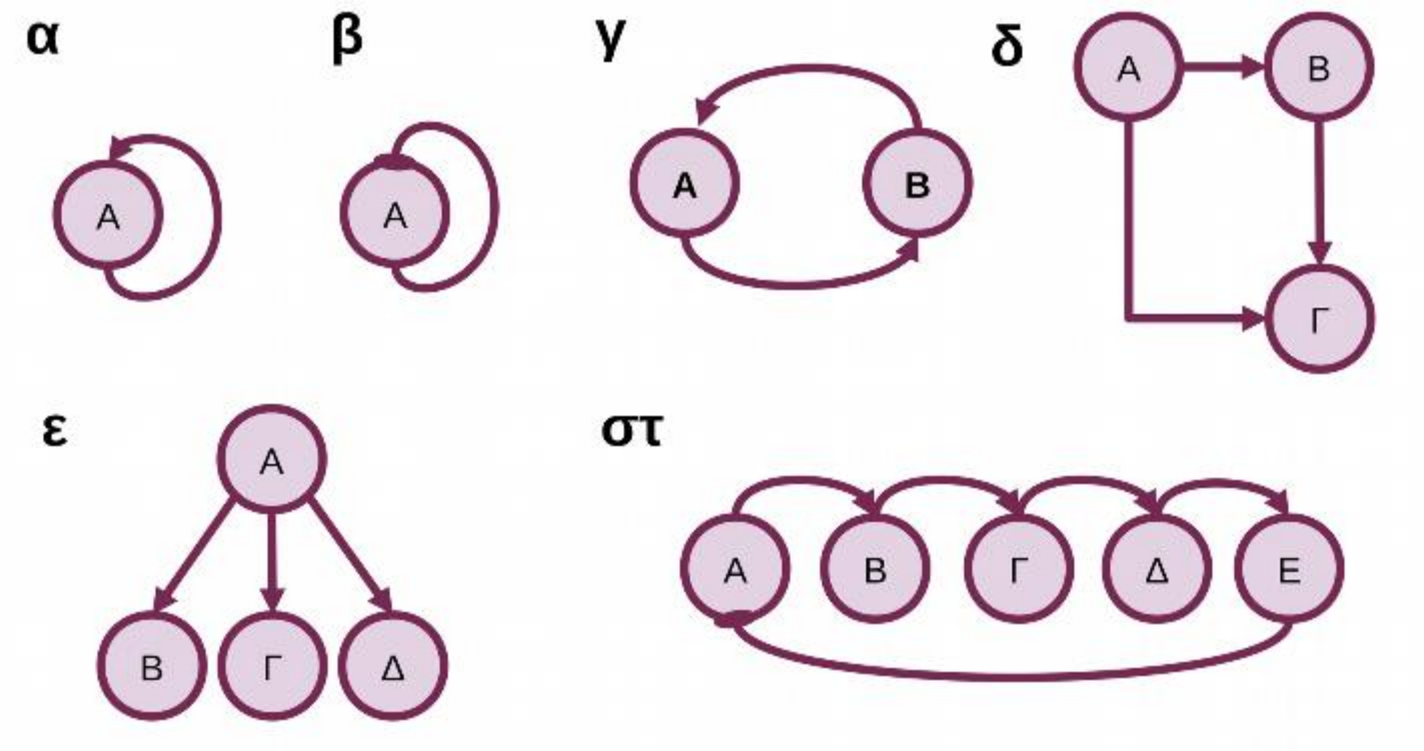


Vitali S, Glattfelder JB, Battiston S (2011) The Network of Global Corporate Control. PLoS ONE 6(10): e25995. doi:10.1371/journal.pone.0025995


A
IN-Tendrils
OUT-Tendrils
IN
SCC
OUT
Tubes
Flow of control
B
IN
(2.2%, 0.6%)
T&T
(13.5%, 19.6%)
SCC
(18.7%, 0.7%)
OCC
(5.8%, 64.0%)
OUT
(59.8%, 15.1%)
C
D
FRANKLIN RESOURCES
PRUDENTIAL FINANCIAL
COMMERZBANK AG
MORGAN STANLEY
CREDIT SUISSE
AXA
CITIGROUP
DEUTSCHE BANK AG
MERRILL LYNCH
BANK OF AMERICA CORP.
BARCLAYS PLC
UBS AG
STATE STREET CORP.
JPMORGAN CHASE & CO
T. ROWE PRICE
GOLDMAN SACHS
LEHMAN BROTHERS
BEAR STEARNS



Εικόνα 9.3: *α) Παράδειγμα διμερούς δικτύου μεταβολικής ρύθμισης όπου με κόκκινο εμφανίζονται οι μεταβολίτες (S, S2, P, Q) και με πράσινο (E0,E1,EX) τα ένζυμα που καταλύουν τις μεταξύ τους αντιδράσεις. Πάνω: το πλήρες δίκτυο. Κάτω: το δίκτυο που περιλαμβάνει μόνο τους μεταβολίτες και έχει προκύψει από συνένωση των ακμών του πάνω δικτύου. β) Ένα μεταβολικό δίκτυο που αντιστοιχεί σε μέρος των αντιδράσεων του μεταβολισμού ενός ανθρώπινου κυττάρου. Σε αυτόν τον πολύπλοκο "μεταβολικό χάρτη" διακρίνονται οι κύκλοι του κιτρικού οξέος και της ουρίας. (Εικόνα από J3D3, CC BY-SA 4.0, από Wikimedia Commons).*



Εικόνα 9.5: *Βασικά μοτίβα μεταγραφικής ρύθμισης. α) Αυτο-ρύθμιση με ενεργοποίηση β) Αυτο-ρύθμιση με καταστολή (βρόχος ανάδρασης) γ) Βρόχος δύο συνιστωσών δ) Πρόδρομη ρύθμιση ενεργοποίησης (feed-forward stimulation) ε) Μοτίβο μοναδικής εισόδου στ) Αλυσίδα ρύθμισης με ανάδραση (Lee et al., 2002).*

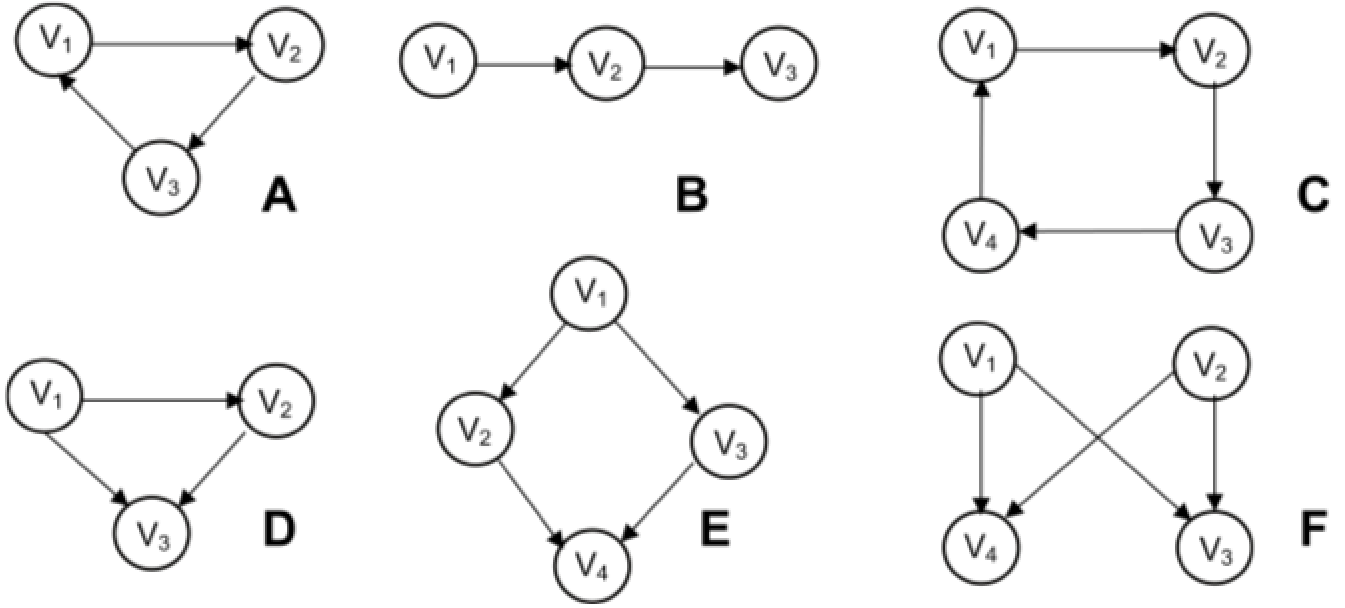
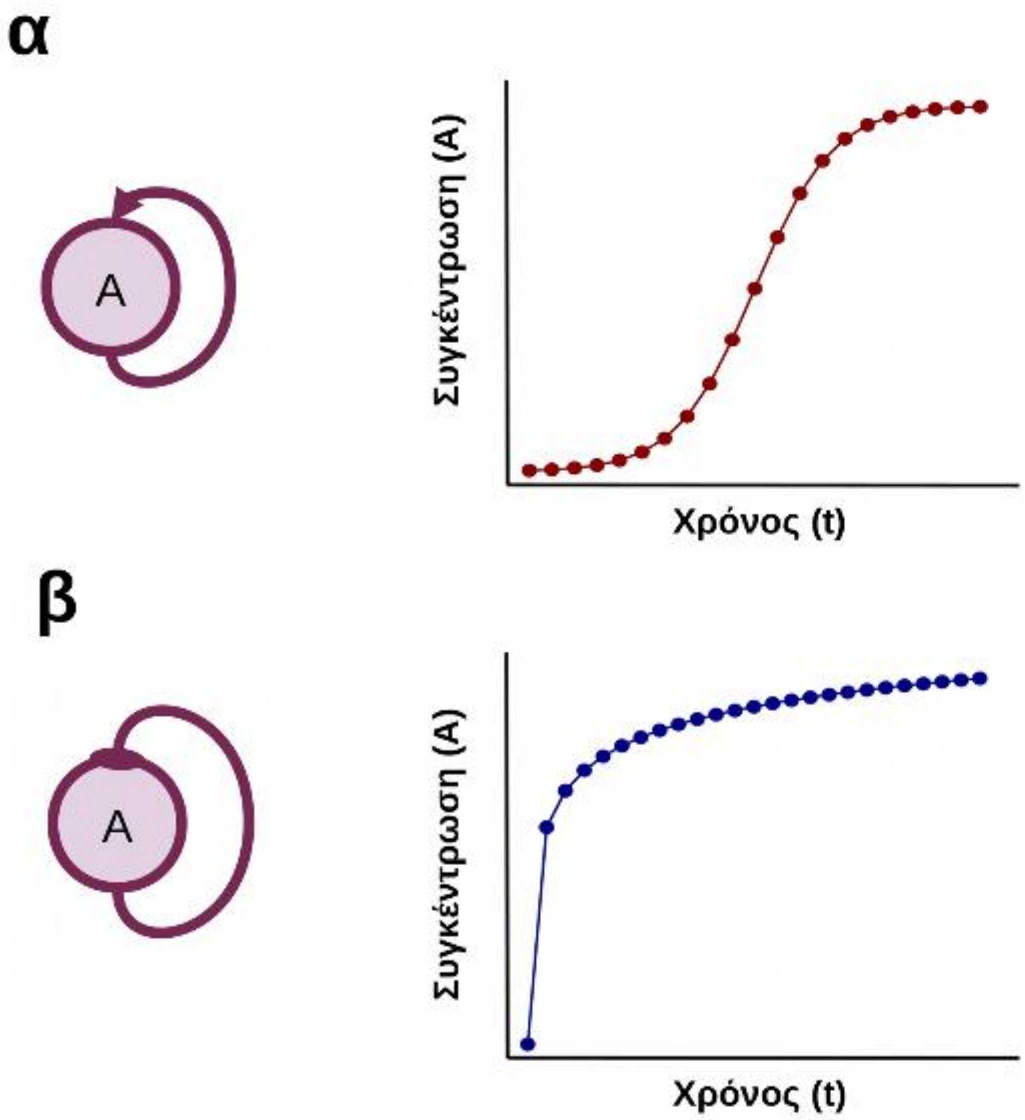
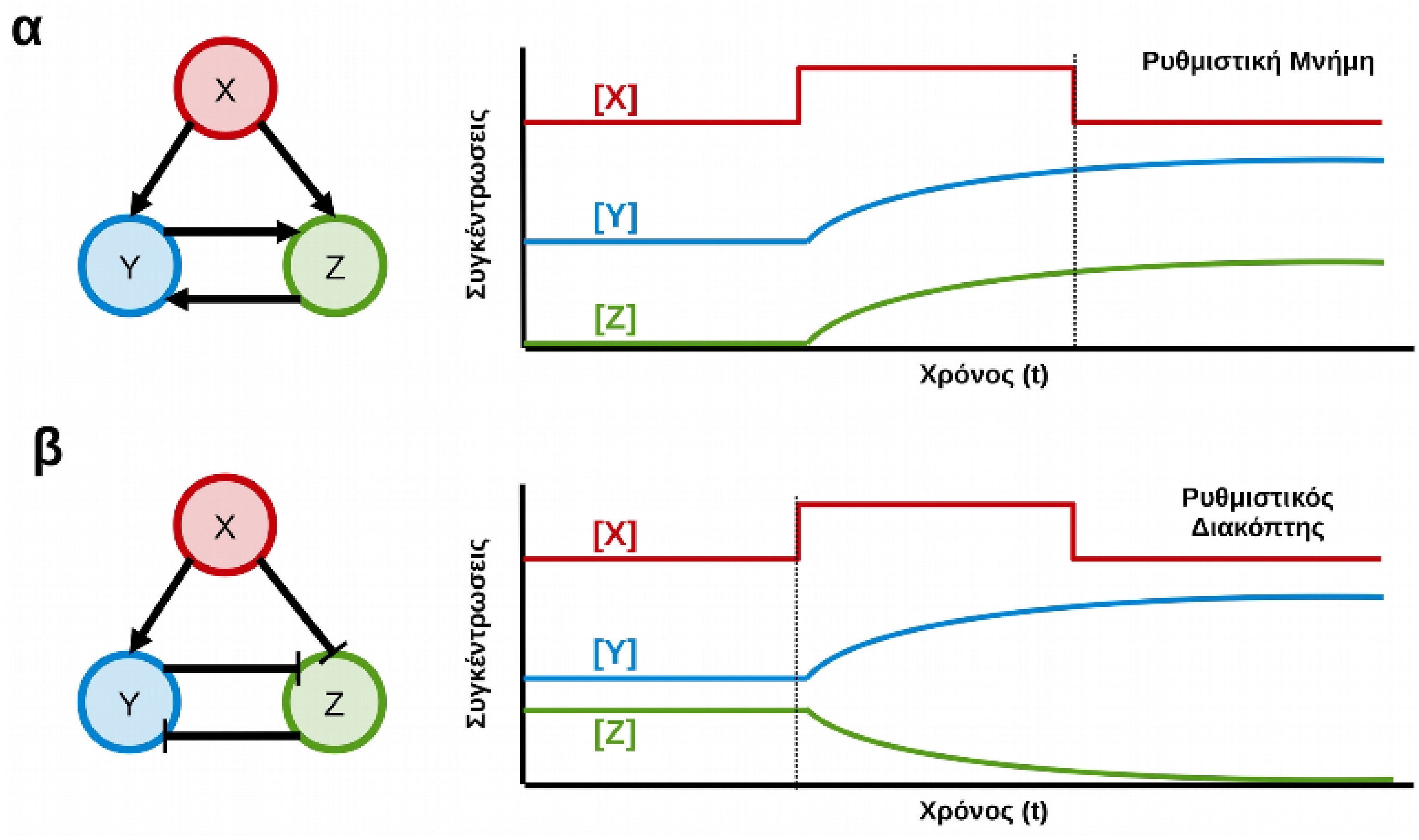


Figure 6 Network Motifs. Some common network motifs. A) *Feed-forward loop*. Type of networks: protein, neuron, electronic. B) *Three chain*. Type of network: food webs. C) *Four node feedback*. Type of network: gene regulatory, electronic. D) *Three node feedback*. Type of network: gene regulatory, electronic. E) *Bi-parallel*. Type of network: gene regulatory, biochemical. F) *Bi-Fan*. Type of networks: protein, neuron, electronic [74].



Εικόνα 9.6: *Μοτίβο ανατροφοδότησης α) Θετική ανατροφοδότηση β) Αρνητική ανατροφοδότηση και οι αντίστοιχες διαφορετικές δυναμικές συμπεριφορές του μοτίβου. Στην πρώτη περίπτωση η θετική ανατροφοδότηση οδηγεί σε μια σιγμοειδή αύξηση της συγκέντρωσης, ενώ στη δεύτερη τα αρχικά ποσοστά αυξάνονται με όλο και μικρότερο ρυθμό λόγω της καταστολής από την αρνητική ανατροφοδότηση.*



Εικόνα 9.7: *Δύο μοτίβα πρόδρομης ρύθμισης με πολύ διαφορετικά λειτουργικά χαρακτηριστικά. α) Πρόδρομη συνεκτική επαγωγή σε μια πλήρως συνδεδεμένη τριάδα (X,Y,Z). Το X ενεργοποιεί οποιοδήποτε από τα X, Y τα οποία λόγω της μεταξύ τους αλληλο-επαγωγικής σχέση παραμένουν ενεργά ακόμα και μετά την απόσυρση του X από το σύστημα (Ρυθμιστική Μνήμη). β) Σε μια μη-συνεκτική επαγωγή τα Y, Z έχουν αλληλο-κατασταλτική σχέση που οδηγεί στην εναλλαγή της ενεργότητάς τους. Το Z που αρχικά ήταν ενεργό καταστέλλεται τόσο από το X όσο και από το Y που το X έχει ενεργοποιήσει. Απλή στιγμιαία ενεργοποίηση του X αντιστρέφει αυτόματα την ισορροπία Y, Z (Ρυθμιστικός Διακόπτης). Η Εικόνα βασίζεται στις αντίστοιχες απεικονίσεις από το (Alon, 2007).*

ΘΕΩΡΙΑ ΓΡΑΦΩΝ

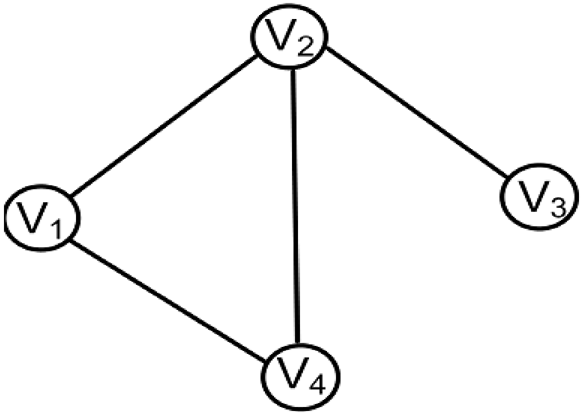
- *Γράφος*: μια δομή που αποτελείται από ένα διατεταγμένο ζεύγος G =(V, E)

G: ο Γράφος

V: σύνολο κόμβων (vertices)

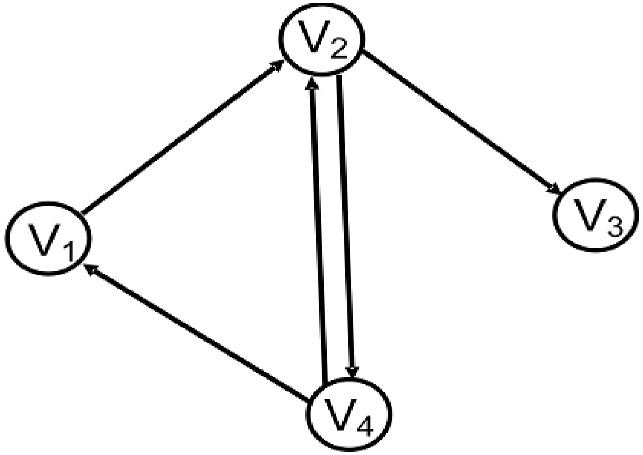
E: σύνολο ακμών (edges), οι οποίες ενώνουν τους κόμβους μεταξύ τους

- Κατευθυνόμενοι (directed), μη κατευθυνόμενοι (undirected)



V={V1,V2,V3,V4}, |V|=4,

E={(V1,V2), (V2,V3), (V2,V4), (V4,V1)}, |E|=4.



V={V1,V2,V3,V4}, |V|=4, E={(V1,V2), (V2,V3), (V2,V4), (V4,V1), (V4,V2)}, |E|=5

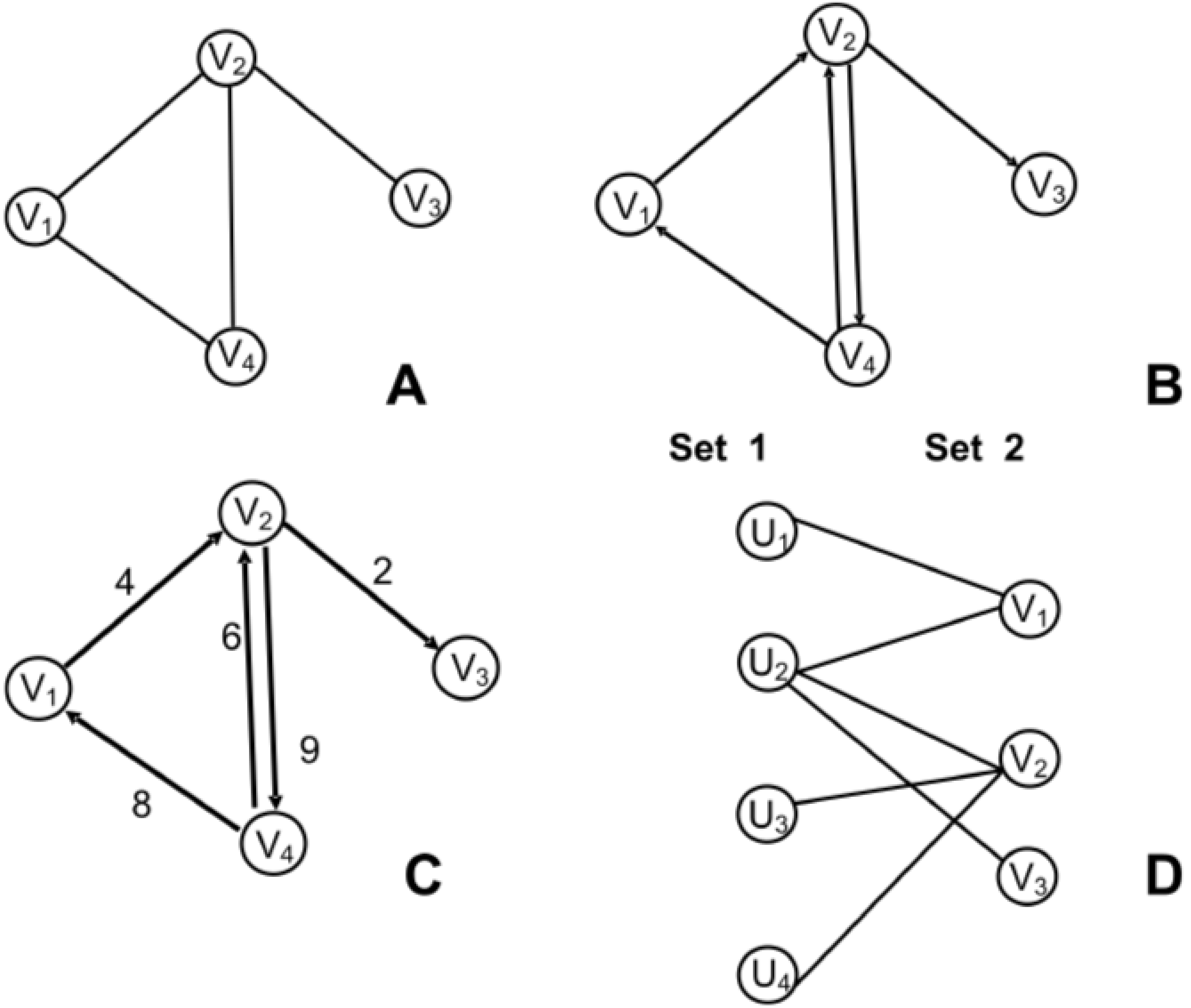
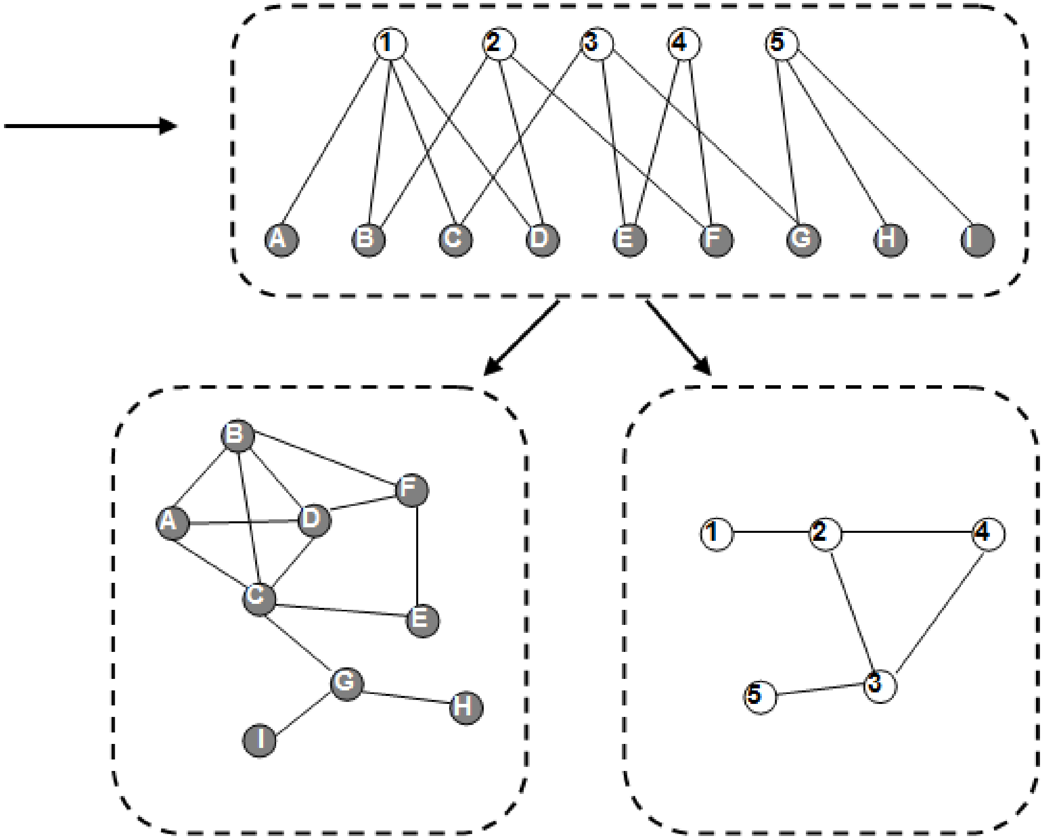


Figure 1 Undirected, Directed, Weighted, Bipartite graphs. A. Undirected Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_4, V_1)\}$, $|E| = 4$. B. Directed Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_4, V_1), (V_4, V_2)\}$, $|E| = 5$. C. Weighted Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2, V_4), (V_2, V_3, V_2), (V_2, V_4, V_9), (V_4, V_1, V_8), (V_4, V_2, V_6)\}$, $|E| = 5$. D. Bipartite graph: $V = \{U_1, U_2, U_3, U_4, V_1, V_2, V_3\}$, $|V| = 7$, $E = \{(U_1, V_1), (U_2, V_1), (U_2, V_2), (U_2, V_3), (U_3, V_2), (U_4, V_2)\}$, $|E| = 6$.

Bipartite graphs

A	1
B	1
B	2
C	1
C	2
C	3
D	1
D	2
E	3
E	4
F	2
F	4
G	3
G	5
H	5
I	5



Παραδείγματα διμερών γράφων

- Γονίδια/Ασθένειες
- Ασθένειες/Συμπτώματα
- Ασθένειες/Φάρμακα
- Αλλά και άλλα:
  - Επιστημονικές εργασίες/Συγγραφείς
  - Ταινίες/Ηθοποιοί
  - Πολυεθνικές/μέλη ΔΣ

Δομή δεδομένων

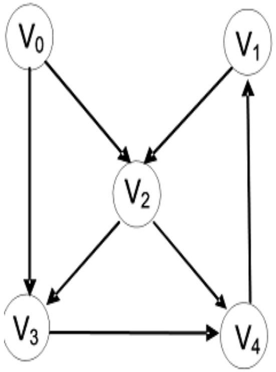
Αναπαράσταση :

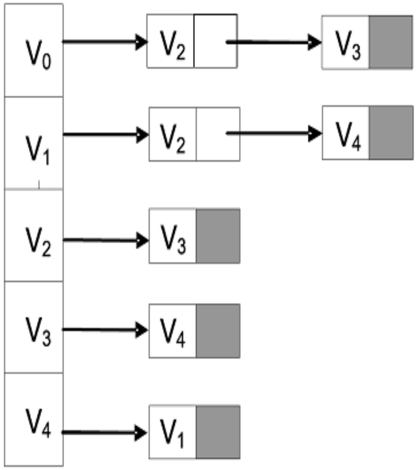
- Πίνακας γειτνίασης (adjacency matrix)

Για έναν γράφο $G = (V, E)$, ο πίνακας γειτνίασης αποτελείται από έναν $|V|x|V| = nxn$ πίνακα, $A=(aij)$ έτσι ώστε $aij=1$ *αν* $(i,j)\in V$ ή $aij=0$

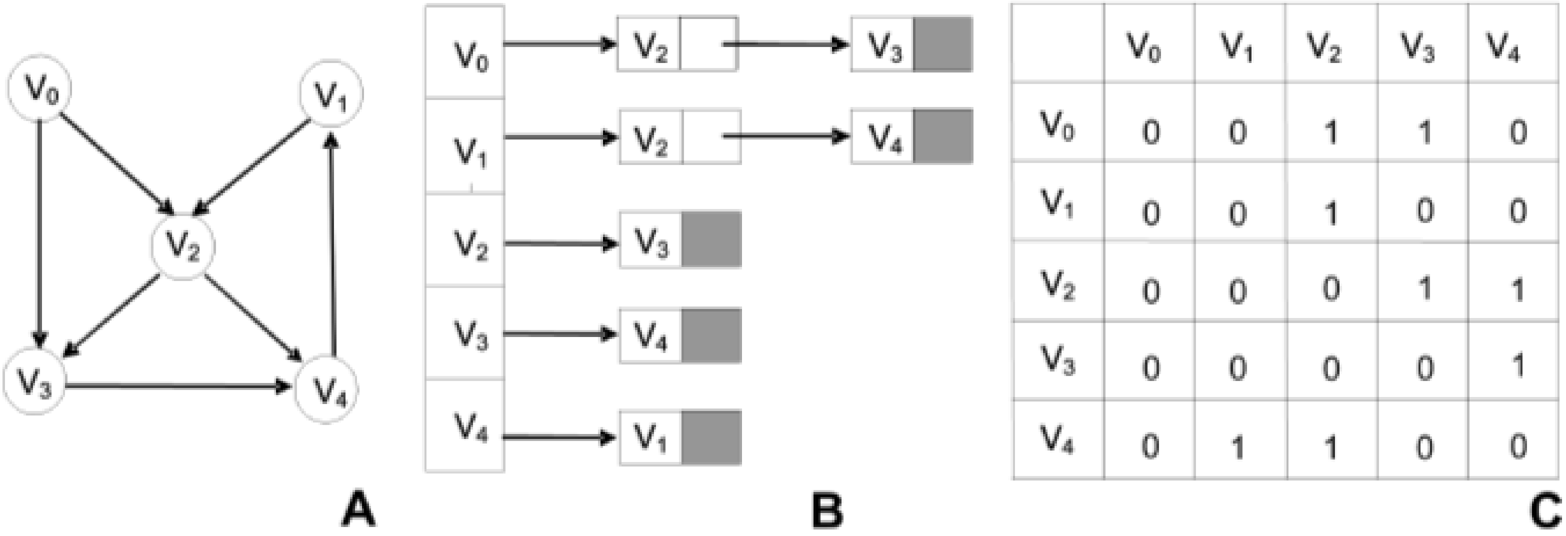
- Λίστα γειτνίασης (adjacency list)

Ένας γράφος G=(V,E), αναπαριστάται ως ένας μονοδιάστατος πίνακας, όπου κάθε κόμβος *i*, είναι δείκτης σε μία συνδεδεμένη λίστα, στην οποία αποθηκεύονται οι κόμβοι που γειτνιάζουν με τον *i* κόμβο





	V_0	V_1	V_2	V_3	V_4
V_0	0	0	1	1	0
V_1	0	0	1	0	0
V_2	0	0	0	1	1
V_3	0	0	0	0	1
V_4	0	1	1	0	0



	V_0	V_1	V_2	V_3	V_4
V_0	0	0	1	1	0
V_1	0	0	1	0	0
V_2	0	0	0	1	1
V_3	0	0	0	0	1
V_4	0	1	1	0	0

Figure 2 Data structures. A. A Directed Graph: A random graph consisting of five nodes and six directed edges. B. Adjacency List: The data structure which represents the directed graph using lists. C. Adjacency Matrix: The data structure which represents the directed graph using a 2D matrix. The zeros represent the absence of the connection whereas the ones represent the existence of the connection between two nodes. The matrix is not symmetric since the graph is directed.

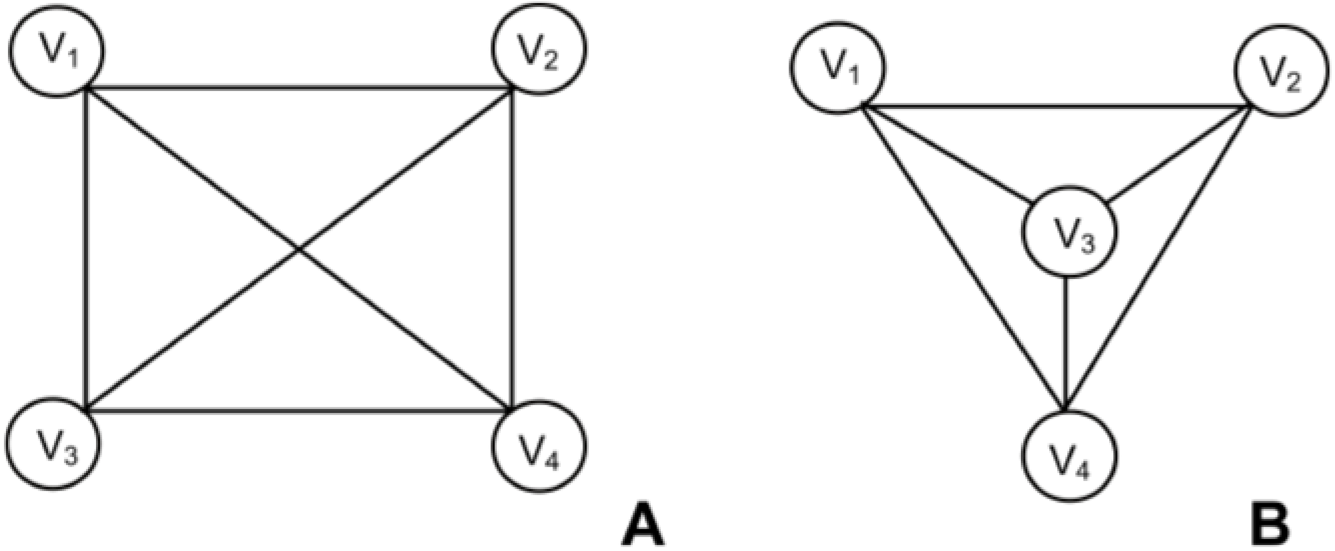


Figure 3 Graph Isomorphism. *$V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_1, V_3), (V_1, V_4), (V_2, V_3), (V_2, V_4), (V_3, V_4)\}$, $|E| = 6$*. Graphs *A* and *B* have different topology but they are isomorphs. The graph is fully connected and every node is connected to any other so that it forms a fully connected clique.

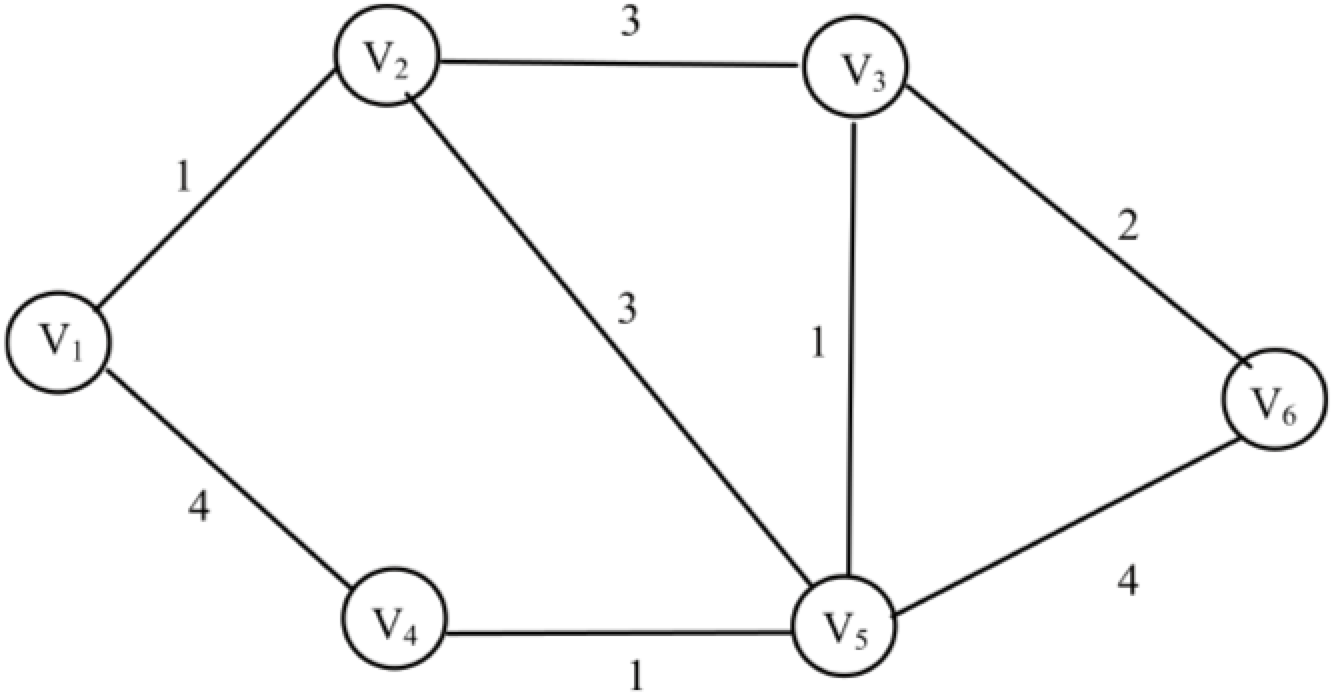


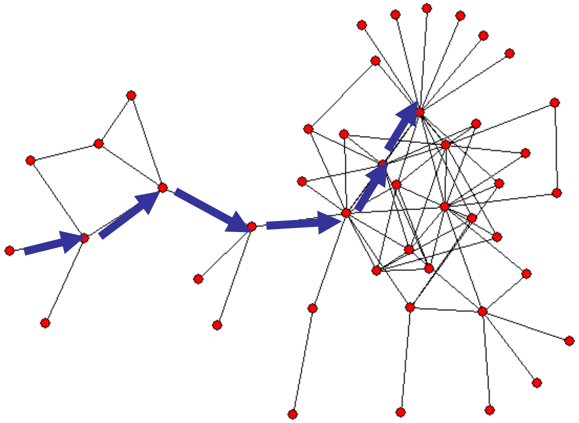
Figure 4 Walks, simple paths trails and cycles in graphs. A *walk* is a sequence of nodes *e.g.* (V_2, V_3, V_6, V_5, V_3). A *simple path* is a walk with no repeated nodes, e.g. (V_1, V_4, V_5, V_2, V_3). A trail is a walk where no edges are repeated e.g. (V_1, V_2, V_3 V_6). A *cycle* is a walk (V_1, V_2,..., V_L) where V_1 = V_L with no other nodes repeated and L>3, e.g. (V_1, V_2, V_5, V_4, V_1).

Απλές μετρήσεις

- Network density → Πόσο αραιός ή πυκνός είναι ένας γράφος

  $\frac{2|E|}{|V|(|V|-1)}$ |E| ≈ |V| |E| ≈ |V|2

- Shortest paths → ονομάζεται απόσταση *δ*(*i* ,*j*) από τον *i* κόμβο στον *j*



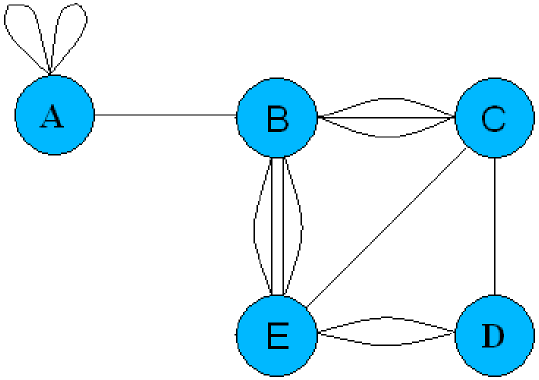
- Network diameter → η μέγιστη τιμή της απόστασης *D=max* $\delta_{\min}(i,j)$

- Network radius → η ελάχιστη τιμή της απόστασης *D=min* $\delta_{\min}(i,j)$

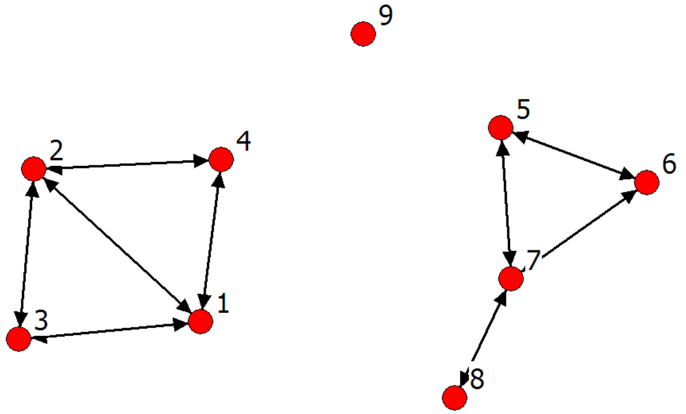
- Characteristic path length → η μέση τιμή της απόστασης *D=avg* $\delta_{\min}(i,j)$

Απλές μετρήσεις

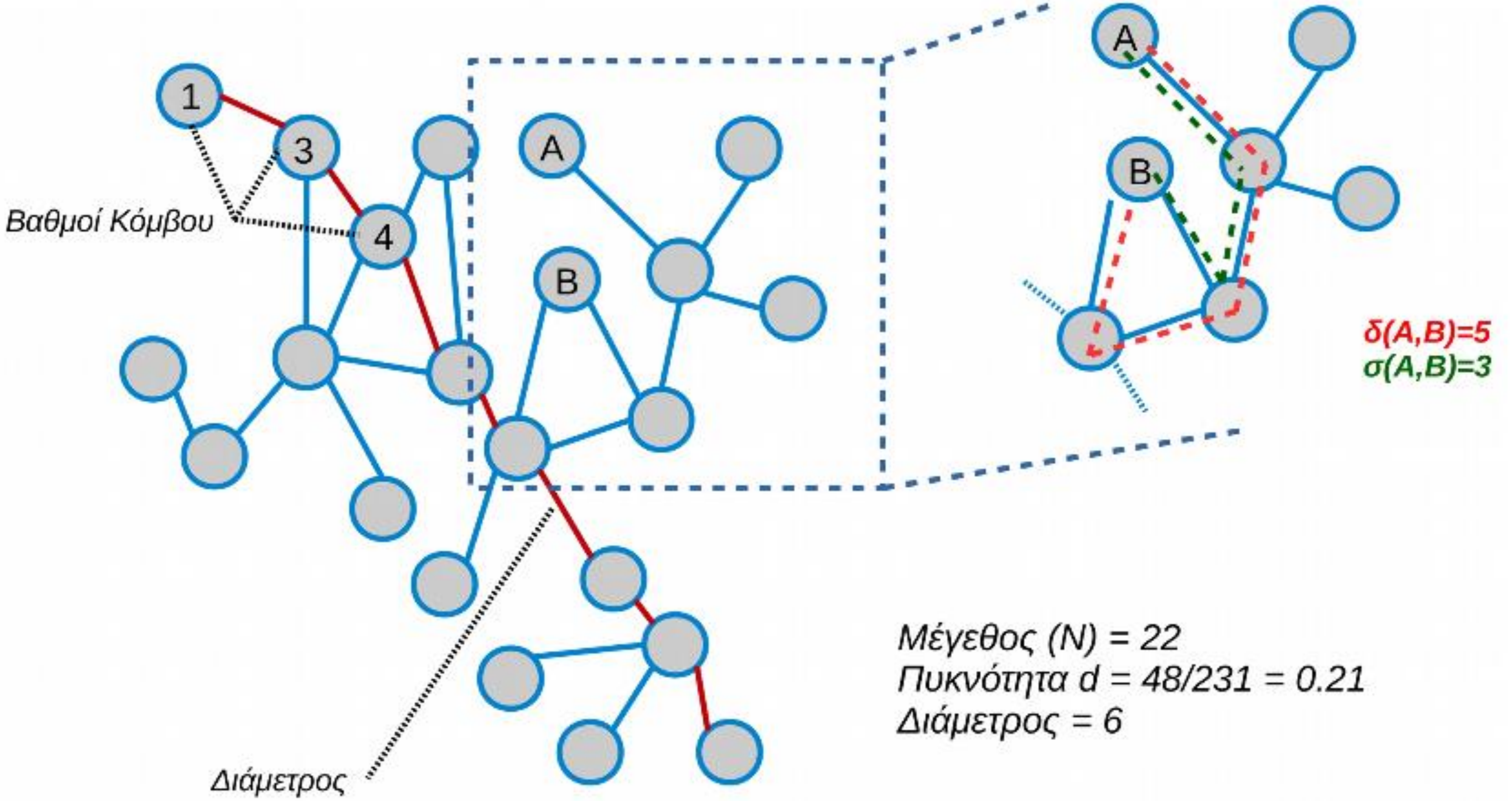
- Connected components → όλοι οι κόμβοι που ενώνονται ανά ζεύγη
- Isolated nodes → κόμβοι απομονωμένοι χωρίς σύνδεση
- Number of self loops → αριθμός βρόγχων
- Multi-edge node pairs → παραπάνω από μία σύνδεση σε δύο κόμβους
- Avg. number of neighbors → μέσος όρος σύνδεσης του κόμβου στο δίκτυο



Multi-graph

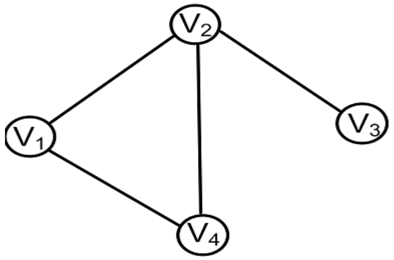


3 components (1,2,3,4), (5,6,7,8),(9) 1 isolate (9).

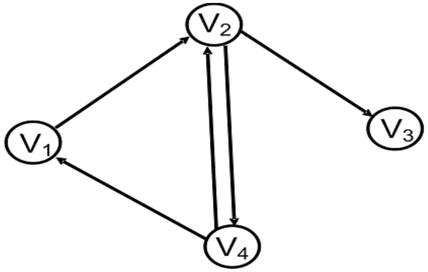


Εικόνα 9.8: *Χαρακτηριστικά μεγέθη και ιδιότητες δικτύων. Το δίκτυο της εικόνας αποτελείται από 22 κόμβους των οποίων οι βαθμοί κυμαίνονται μεταξύ 1 και 5. Το σύνολο των ακμών είναι 48 με το μέγιστο δυνατό να είναι (22x21)/2=231 κι έτσι η πυκνότητα είναι ίση με 0.21. Στην ένθετη εικόνα φαίνεται η ελάχιστη απόσταση δύο κόμβων A, B. Από τις δύο δυνατές διαδρομές (κόκκινη, πράσινη) μεταξύ A και B η ελάχιστη είναι η πράσινη (σ(A,B)=3). Η μεγαλύτερη ελάχιστη διαδρομή στο δίκτυο, η οποία αντιστοιχεί στη διάμετρό του, φαίνεται στη μεγαλύτερη εικόνα σκιασμένη με βαθύ κόκκινο και είναι ίση με 6.*

Βαθμός-Node degree



Degree (v2): 3

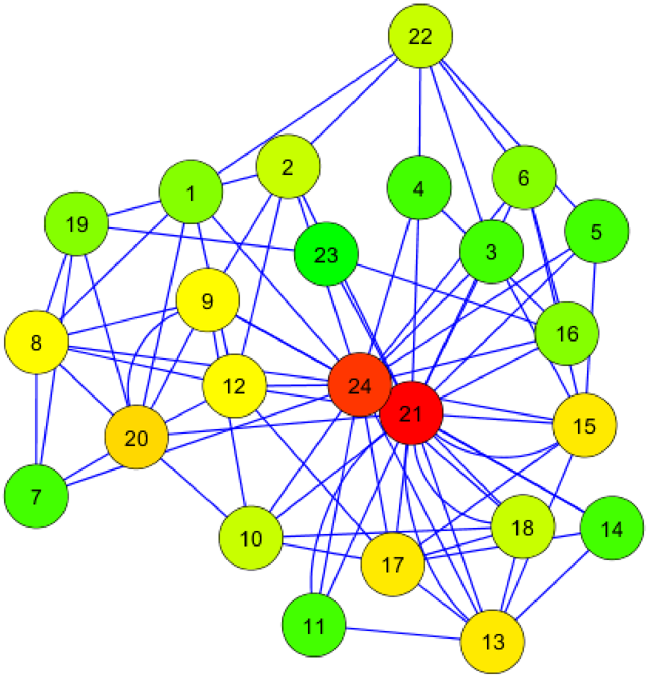


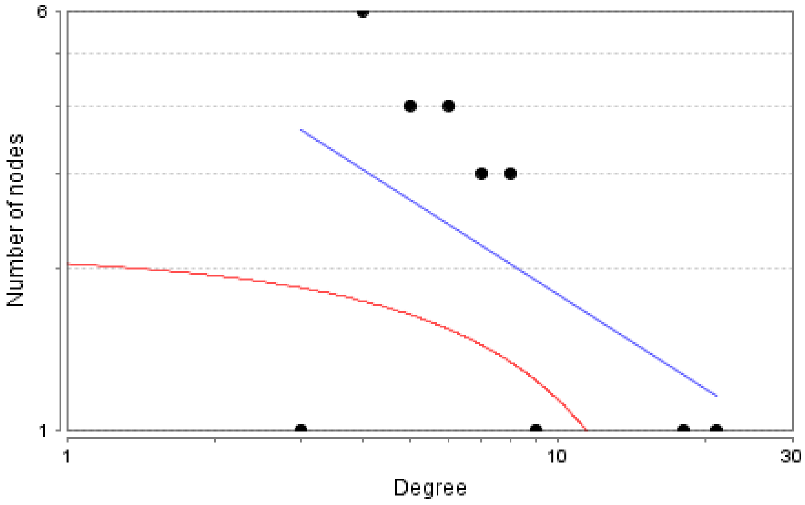
Indegree (v2):2 Outdegree(v2):2

- Ο συνολικός αριθμός των ακμών που προσπίπτουν σε έναν κόμβο
- Κόμβοι με ισχυρή συνδεσιμότητα (high degree) ονομάζονται “hubs”

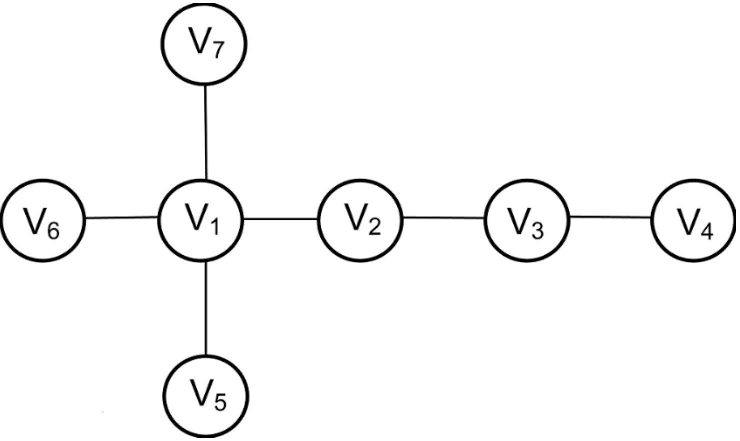
•Undirected: $C_d(i) = \deg(i)$

•Directed: $C_{d\,in}(i) = \deg_{in}(i)$ $C_{d\,out}(i) = \deg_{out}(i)$





Betweenness centrality



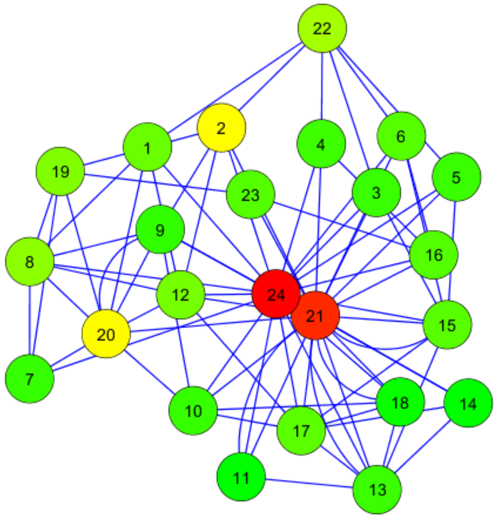
- Δείχνει τους σημαντικούς κόμβους που βρίσκονται σε υψηλό ποσοστό στα μονοπάτια άλλων κόμβων σε ένα δίκτυο

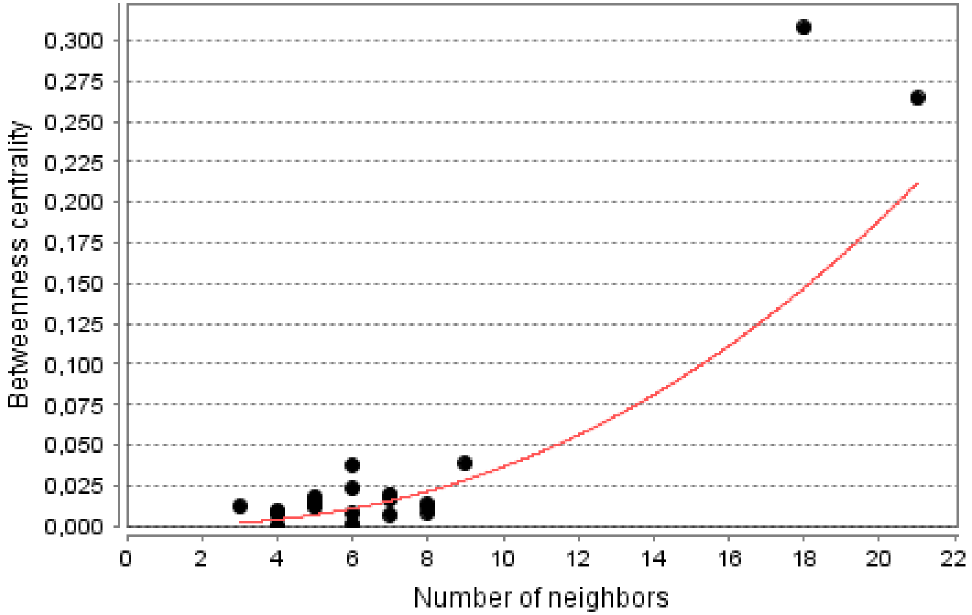
$$C_b(w) = \sum_{(i,j)\in V(w)} \frac{\sigma_{ij}(w)}{\sigma_{ij}}$$

Np(1)= 12 , Np(2)=8 , Np(3) = 5 , Np(4)=Np(5)=Np(6)=Np(7)=0

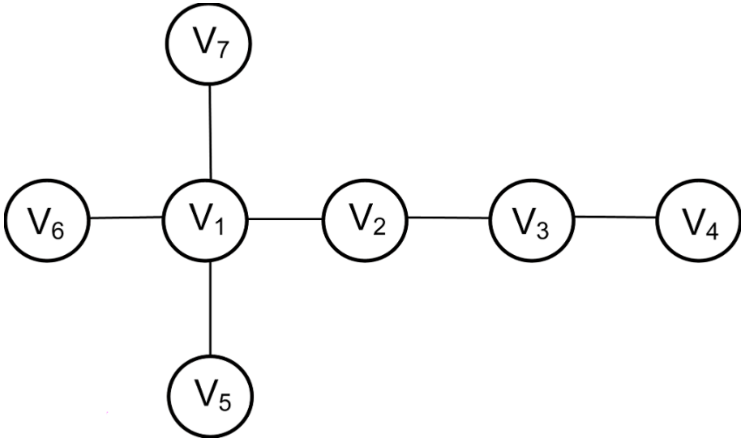
Np= Np(1)+Np(2)+Np(3)+Np(4)+Np(5)+Np(6)+Np(7) =25

Cb (1) = 12/25 ,Cb (2) = 8/25 ,Cb (3) = 5/25 ,Cb (4) = Cb (5) = Cb (6) = Cb (7) = 0





Closeness centrality

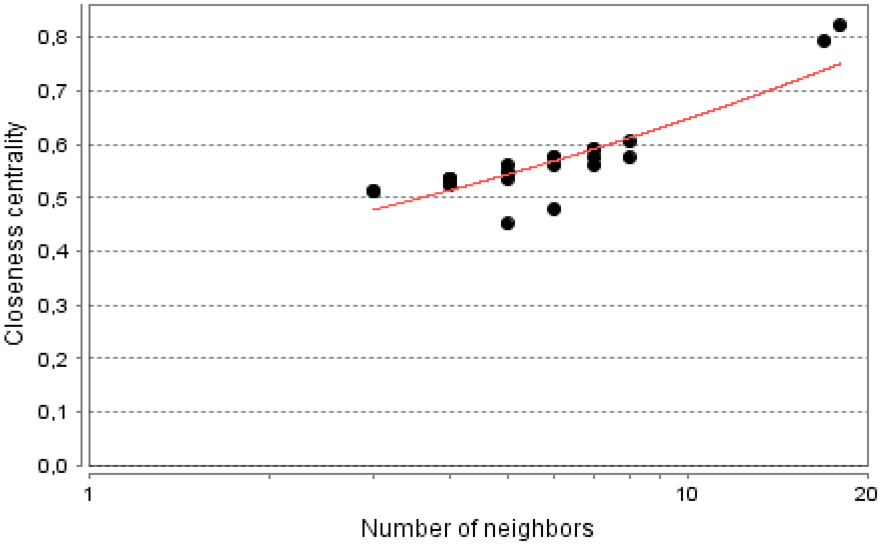


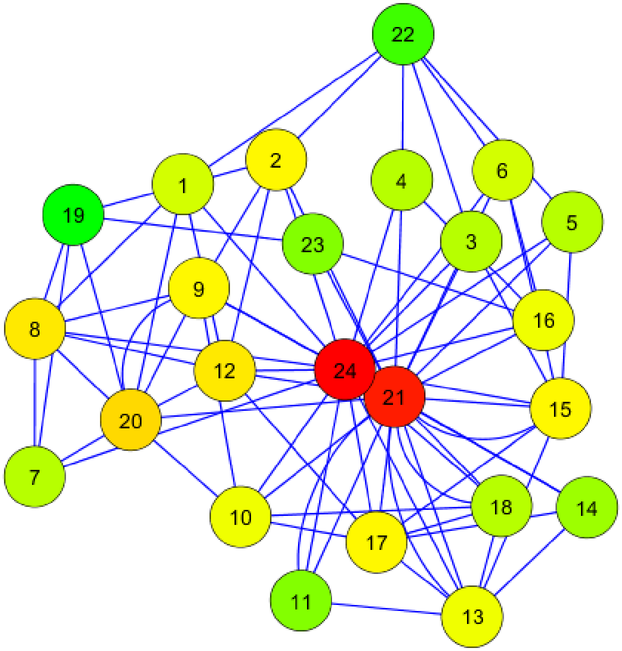
- Η μέτρηση αυτή υποδεικνύει τους σημαντικούς κόμβους οι οποίοι μπορούν να επικοινωνήσουν γρήγορα με άλλους κόμβους του δικτύου

$$C_{clo}(i) = \frac{1}{\sum_{t\in V}^{|V|} dist(i,j)}$$

- d1 = 4x1 + 1x2 + 1x3 = 9 Cclo(1) = 6/9
- d2 = 2x1 + 4x2 = 10 > d1 και Cclo(2) = 6/10

ο V1 είναι περισσότερο σημαντικός και κεντρικός από τον V2 διότι d1>d2





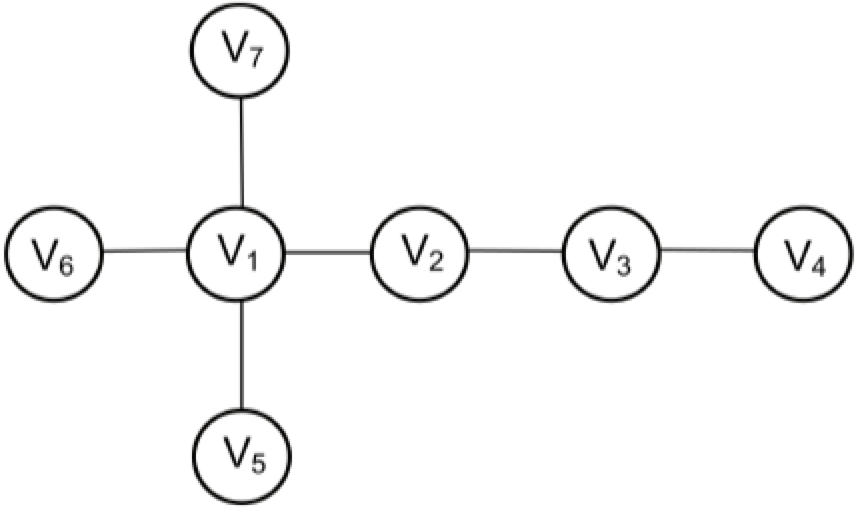


Figure 7 Closeness and Betweeness centralities. **Closeness centrality.** ***V_1***: $d_1 = 4 \times 1 + 1 \times 2 + 1 \times 3 = 9$, $C_{clo}(1) = 6/9$. V_1 accesses 4 nodes (V_2, V_5, V_6, V_7) with step 1, 1 node (V_3) with step 2 and 1 node (V_4) with step 3. 6 nodes can be accessed in total by V_1. ***V_2***: $d_2 = 2 \times 1 + 4 \times 2 = 10 > d_1$, $C_{clo}(2) = 6/10$. V_2 accesses 2 nodes (V_1, V_3) with step 1 and 4 nodes (V_4, V_5, V_6, V_7) with step 2. 6 nodes can also be accessed in total by V_2. As a result, V_1 is more central than node V_2 since $d1>d_2$. Betweenness centrality. $N_p(1) = 12$ shortest paths that pass through node V_1. The paths from the starting to the ending node are $\{V_2\text{-}V_5, V_2\text{-}V_6, V_2\text{-}V_7, V_3\text{-}V_5, V_3\text{-}V_6, V_3\text{-}V_7, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7, V_5\text{-}V_6, V_5\text{-}V_7, V_6\text{-}V_7\}$. $N_p(2) = 8$ shortest paths that pass through node V_2. The paths are $\{V_1\text{-}V_3, V_1\text{-}V_4, V_3\text{-}V_5, V_3\text{-}V_6, V_3\text{-}V_7, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7\}$. $N_p(3) = 5$ $\{V_1\text{-}V_4, V_2\text{-}V_4, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7\}$. $N_p(4) = N_p(5) = N_p(6) = N_p(7) = 0$. $N_p = 25$ the total sum of shortest paths that pass through the nodes, thus $N_p = N_p(1)+N_p(2)+N_p(3)+N_p(4)+N_p(5)+N_p(6)+N_p(7)$. The centralities are $C_b\ (1) = 12/25 = 0.48$, $C_b\ (2) = 8/25 = 0.32$, $C_b\ (3) = 5/25 = 0.20$, $C_b\ (4) = C_b\ (5) = C_b\ (6) = C_b\ (7) = 0$, thus node V_1 is more central.

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. There are two different strategies to organize data. These are the *agglomerative* and the *divisive*: ***Agglomerative***: It is a “bottom-up” approach. Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. ***Divisive***: This is a “top-down” approach. In this case, all of the observations start by forming one cluster, and then split recursively as one moves down the hierarchy. Some of the most common tree based clustering algorithms that organize data in hierarchies are the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [117,118], Neighbor Joining [112,119] and Hierarchical Clustering [120,121], all of which represent their clusters as tree structures. The results of hierarchical clustering are usually presented in a dendrogram. Figure 10 shows an example of how genes can be clustered.

Let n_r be the number of clusters and x_{ri} is the *i*th object in cluster *r* and cluster r is formed from clusters *p* and *q*. In the following, we describe the different methods used to calculate distances between clusters in hierarchical clustering.

Single linkage calculates the smallest distance between objects in the two clusters to merge them: $d(r, s) = \min(dist(x_{ri}, x_{sj})), i \in (i,..., n_r), j \in (1,....n_s)$.

Complete linkage calculates the largest distance between objects in the two clusters to merge them: $d(r, s) = \max(dist(x_{ri}, x_{sj})), i \in (i,..., n_r), j \in (1,....n_s)$.

Average linkage uses the average distance between all pairs of objects in any two clusters: $d(r,s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$. This algorithm is also known as *Unweighted Pair Group Method with Arithmetic Mean (**UPGMA**)* [117,118].

Centroid linkage finds the Euclidean distance between the centroids of the two clusters: $d(r,s) = ||\overline{x_r} - \overline{x_s}||_2, \overline{x_r} = \frac{1}{n_r}\sum_{i=1}^{n_r} x_{ri} \cdot || \, ||_2$ is the Euclidean distance.

Median linkage uses the Euclidean distance between weighted centroids of the two clusters, $d(r,s) = ||x_r - x_s||_2, x_r, x_s$ are weighted centroids for the clusters r and s. If cluster r was created by combining clusters p and q, $\mathrm{x_r}$ is defined recursively as $x_r = \frac{1}{2}(x_p + x_q)x_r$.

Single or complete linkages are the fastest of the linkage methods. However, single linkage tends to produce stringy clusters, which is not always preferable. The centroid or average linkage produce better results regarding the accordance between the produced clusters and the structure present in the data. These methods require much more computations. Average linkage and complete linkage may be the preferred methods for microarray data analysis [115].

Ward's linkage finds the incremental sum of squares; that is, the increase in the total within-cluster sum of squares as a result of joining two clusters. The within-cluster sum of squares is defined as the sum of the squares of the distances between all objects in the cluster and the centroid of the cluster. The sum of squares measure is equivalent to the following distance measure $d(r,s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}}||\overline{x_r} - \overline{x_s}||_2$,

where $|| \, ||_2$ is the Euclidean distance and $\overline{x_r}, \overline{x_s}$ are the centroids of clusters r and s and n_r and n_s are the number of elements in clusters r and s.

Weighted average linkage uses a recursive definition for the distance between two clusters. If cluster r was created by combining clusters p and q, the distance between r and another cluster s is defined as the average of the distance between p and s and the distance between q and s: $d(r,s) = \frac{(d(p,s) + d(q,s))}{2}$.

Neighbor Joining [112,119] was initially proposed for finding pairs of operational taxonomic units (OTUs) that minimize the total branch length at each stage of clustering of OTUs starting with a star-like tree. The branch lengths as well as the topology of a parsimonious tree can quickly be obtained by using this method [112].

Known platforms that already share the tree-based algorithms described above are the Hierarchical Clustering Explorer (HCE) [122,123], MEGA [124-127] or TM4 [128].

A recent review article shows which file formats, visualization techniques and algorithms can be used for tree analysis [129].

Another category of clustering algorithms tries to cluster data in separate groups by identifying common properties that the nodes of a network share. Different strategies exist, like for example trying to find dense areas in a graph or areas where message exchange between nodes is easier or to identify strongly connected components or clique-like areas etc. Many of such algorithms have been used in different case studies like for example to identify protein families [130], to detect protein complexes in PPI networks [131,132], or for finding patterns and motifs in a sequence [133]. Even though many more exist, some of the most famous algorithms are given below.

Markov Clustering [134] (MCL) algorithm is a fast and scalable unsupervised clustering algorithm based on simulation of stochastic flow in graphs. The MCL algorithm can detect cluster structures in graphs by a mathematical bootstrapping procedure which takes into account the connectivity properties of the underlying network. The process deterministically computes the probabilities of random walks through a graph by alternating two operations: expansion, and inflation of the underlying matrix. The principle behind it is that random walks on a graph are likely to get locked within dense subgraphs rather than move between dense subgraphs via sparse connections. In other words, higher length paths are more often encountered between nodes in the same cluster than between nodes within different clusters, such that the probabilities between nodes in the same complex will typically be higher in expanded matrices. Clusters are identified by alternating expansion and inflation until the graph is partitioned into subsets so that there are no longer paths between these subsets [135,136].

***k*-Means** [137] is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. K-means and its modifications are widely used for gene expression data analysis [138]. It is a supervised method and users need to predefine the number of clusters. Its complexity is *O(nlk)* where *k* is the number of clusters, *n* the size of the dataset and *l* the loops of the algorithm. The k-means algorithm is one of the simplest and fastest clustering algorithms. However, it has a major drawback: the results of the k-means algorithm may change in successive runs because the initial clusters are chosen randomly.

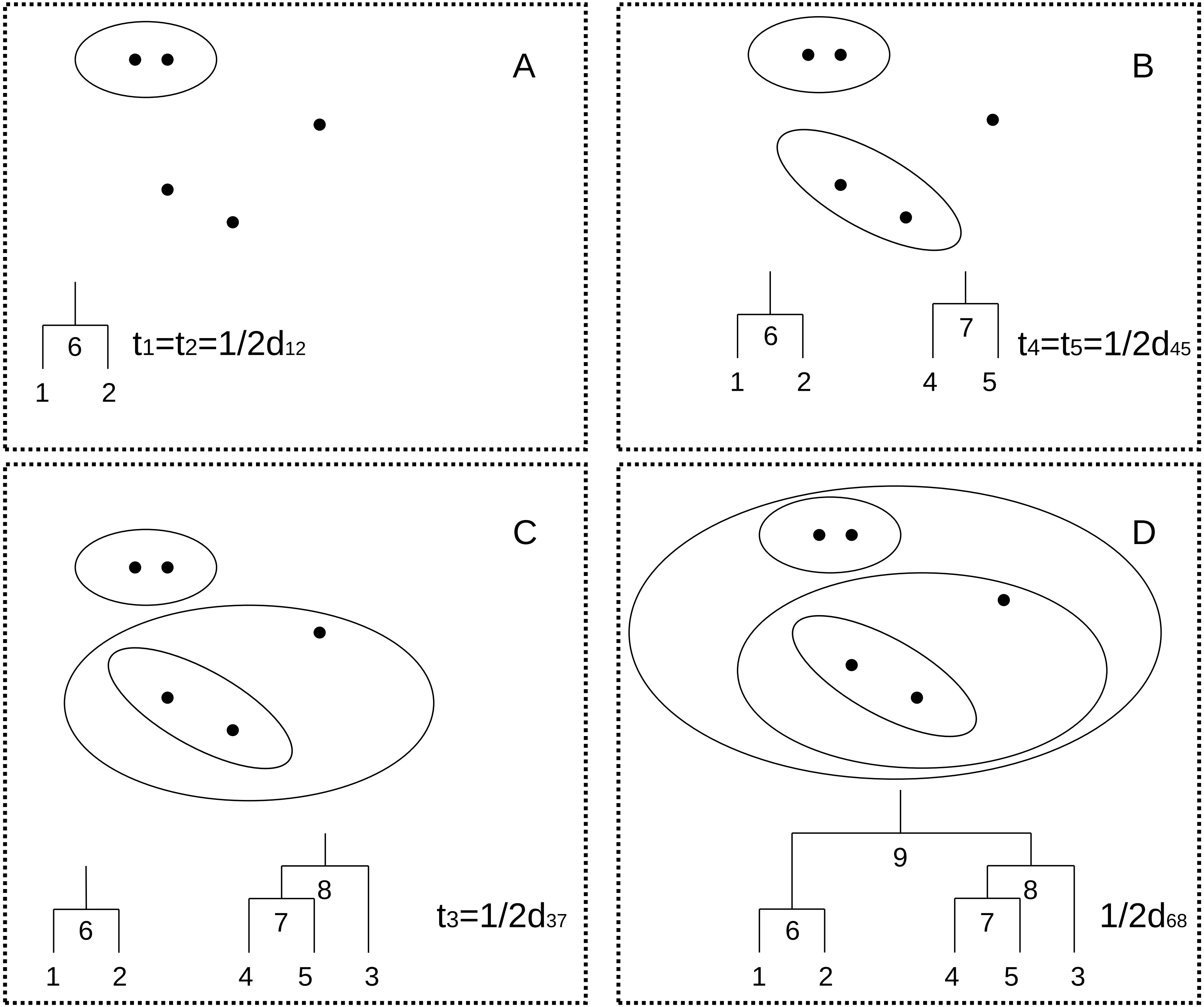
Affinity Propagation [139] takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential candidates. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges.

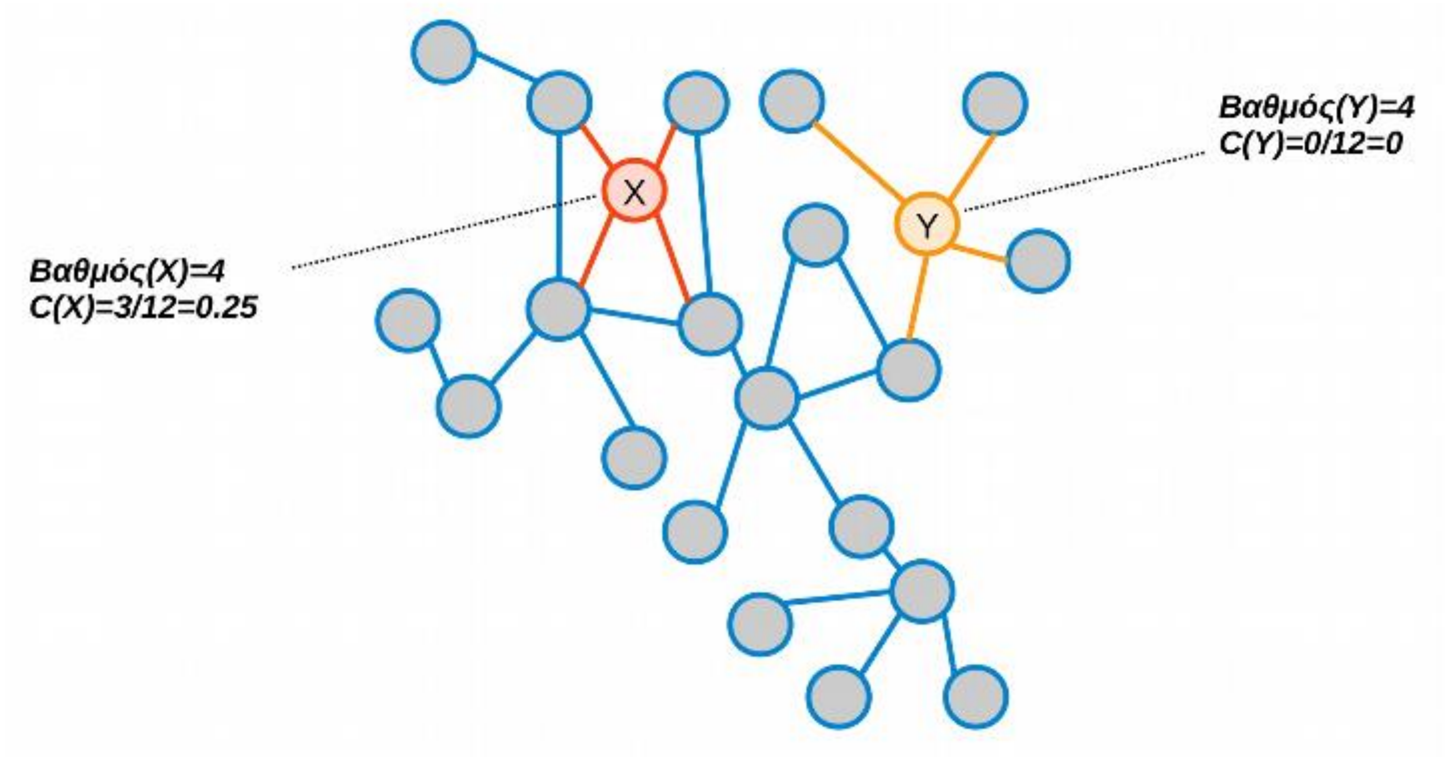
Restricted Neighborhood Search Cluster Algorithm [140]: It tries to find low cost clustering by composing first an initial random clustering. Later it iteratively moves one node from one cluster to another in a random way trying to improve the clustering cost.

Spectral clustering [141]: This algorithm tries to find clusters in the graph such that the nodes within a cluster are connected with highly-similar edges and the connections between such areas are weak, constituted by edges with low similarity. The aim is to identify these tightly coupled clusters, and cut the inter-cluster edges. Figure 11 shows an example of protein complex prediction from PPI yeast dataset [12].

Despite the great variety of clustering techniques, many articles directly compare the various clustering methodologies like [135] and [142]. Very often we encounter articles that compare similar algorithms using different datasets and come to very diverse conclusions and results i.e [142,143].

Concerning the visualization of networks, the availability of clustering techniques and their complex configuration/combination, today to a large extent, there is a lack of visualization platforms or tools that are able to integrate a variety of more advanced algorithms and the development implementation of such implementations emerges [144]. Platforms that share clustering algorithms are the Network Analysis Tool (NEAT) [145] and jClust [146] but they are still poor in the variety of methods they offer. Software like ArrayCluster [147] and MCODE [60] is often used in analysis of gene expression profiles and coexpression detection. Many visualization tools [144] such as Medusa [148], Cytoscape [149], Pajek [98] and many others [144] visualize networks in both 2D and 3D, but very few of them like Arena3D [150] try to bridge the gap between clustering analysis and visualization.


A
6
1
2
$t_1=t_2=1/2d_{12}$
B
6
7
1
2
4
5
$t_4=t_5=1/2d_{45}$
C
6
8
7
1
2
4
5
3
$t_3=1/2d_{37}$
D
9
8
7
6
1
2
4
5
3
$1/2d_{68}$



Εικόνα 9.9: *Συντελεστής συσσωμάτωσης κόμβων. Οι δύο κόμβοι Χ, Υ του σχήματος έχουν τον ίδιο βαθμό (4) όμως διαφέρουν πολύ σε ότι αφορά το συντελεστή συσσωμάτωσης. Τρεις από τους τέσσερις γείτονες του Χ συνδέονται μεταξύ τους κι έτσι με βάση το μέγιστο αριθμό συνδέσεων k(k-1)=12 ο συντελεστής συσσωμάτωσης για το Χ είναι ίσος με 3/12=0.25. Η αντίστοιχη τιμή για τον Υ είναι 0 καθώς μεταξύ των γειτόνων του δεν υπάρχει καμία σύνδεση.*

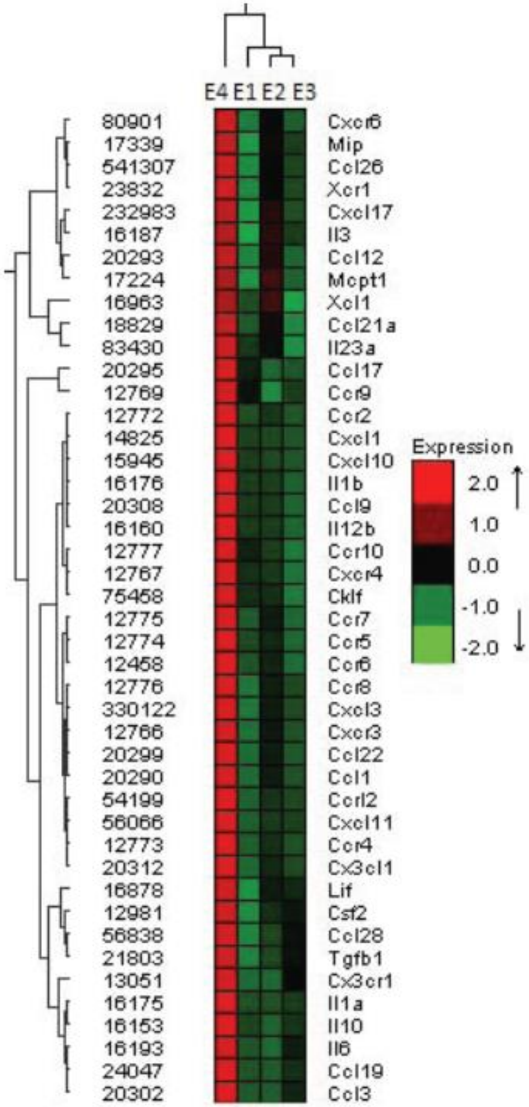


Figure 10 Average linkage hierarchical clustering example. The expression of 44 genes was measured in 4 experiments (E_1, E_2, E_3, E_4). The genes were classified according to their coexpression levels. The Pearson Correlation Coefficient was used (*r*-value) to analyze gene set signal values. Genes were clustered according to the *r*-value correlation matrix using the Average Linkage Hierarchical clustering method. The tree on the left clusters the expressions of the genes whereas the tree on top of the figure clusters the profiles of the experiments. Thus experiments E_2 and E_3 are similar and closely related.

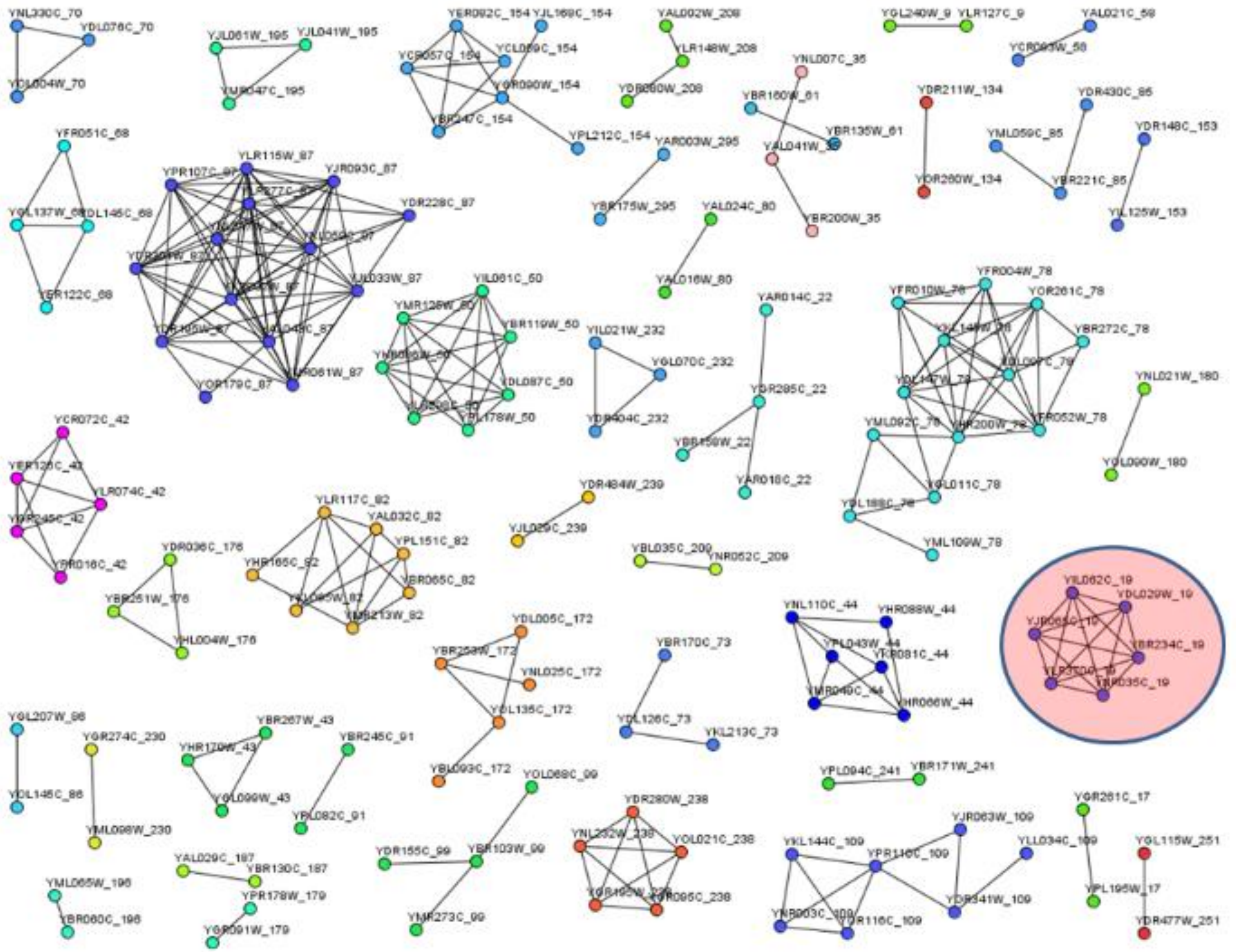


Figure 11 Predicting protein complexes from PPI networks. Protein complexes predicted after applying Spectral clustering algorithm and filtering the results in a yeast protein-protein dataset [12] using the jClust application [146]. The budding yeast Arp2/3 complex that is highlighted was successfully predicted.

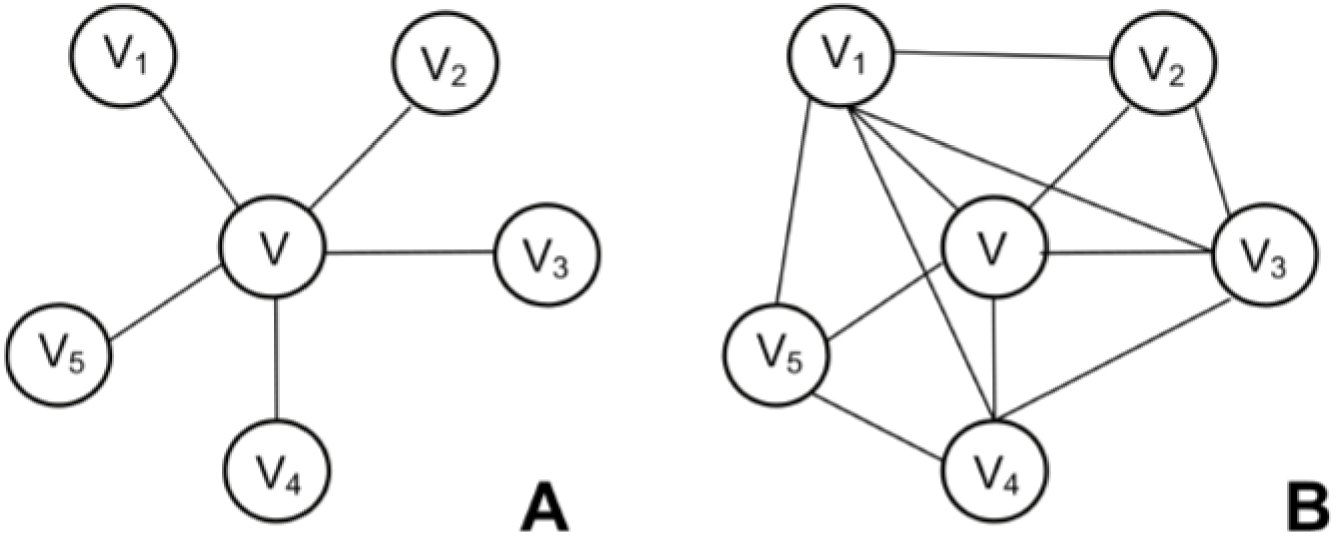


Figure 5 Clustering Coefficient. A) Node *V* behaves like a hub but it has clustering coefficient $C = 0$. B) Node *V* comes with a high clustering coefficient. The maximum number of potential connection is given by $E_{max}=|V|(|V|-1)/2$ where $|V| = 5$ is the number of the neighbors of node V, thus $E_{max} = 10$. The neighbors of node V are connected with 7 edges between each other, $E = \{(V_1, V_2), (V_2, V_3), (V_3, V_4), (V_4, V_5), (V_5, V_1), (V_1, V_3), (V_1, V_4)\}$. The clustering coefficient of node V is $C = E_V/E_{max} = 7/10 = 0.7$.

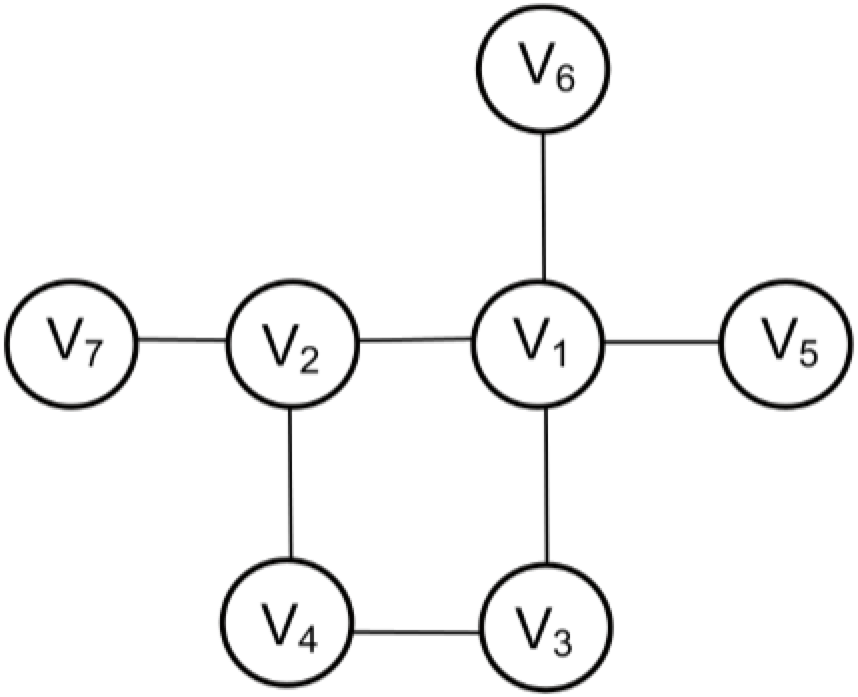
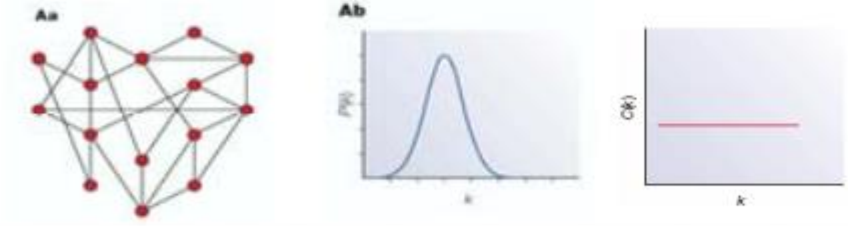


Figure 8 Eccentricity Centrality. $\boldsymbol{V_1}$: 4 × 1, 2 × 2; V_1 accesses 4 nodes (V_2, V_3, V_5, V_6) with step 1 and 2 nodes (V_4, V_7) with step 2. The step represents the shortest path. The maximum shortest path $d_{max} = 2$. $\boldsymbol{V_2}$: 3 × 1, 3 × 2; Similarly V_2 accesses 3 nodes (V_4, V_7, V_1) with step 1 and 3 nodes (V_3, V_5, V_6) with step 2. The maximum shortest path $d_{max} = 2$. $\boldsymbol{V_3}$: 2 × 1, 3 × 2, 1 × 3; Similarly V_3 accesses 2 nodes (V_1, V_4) with step 1, 3 nodes (V_2, V_5, V_6) and one node (V_7) with step 3. The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_4}$: 2 × 1, 2 × 2, 2 × 3; The maximum shortest path $d_{max}=3$. $\boldsymbol{V_5}$: 1 × 1, 3 × 2, 2 × 3; The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_6}$: 1 × 1, 3 × 2, 2 × 3; The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_7}$: 1 × 1, 2 × 2, 3 × 3; The maximum shortest path $d_{max} = 3$. As a result, the ordering of the nodes according to C_{ecc} : (V_1,V_2), (V_3,V_4,V_5,V_6,V_7).

A. Random Networks [Erdos and Rényi (1959, 1960)]



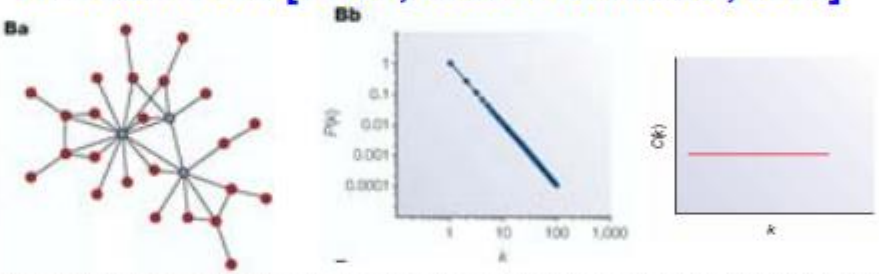
$$P(k) = \frac{e^{-\bar{k}}\,\bar{k}^{k}}{k!}$$

Mean path length ~ ln(k)

Phase transition:

Connected if: $p \ge \ln(k)/k$

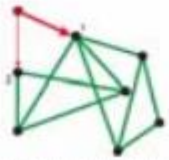
B. Scale Free [Price,1965 & Barabasi,1999]



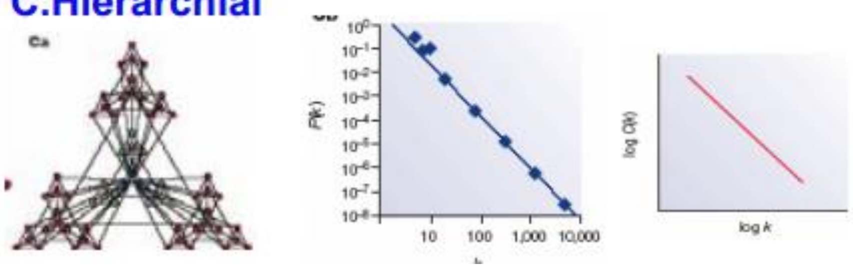
$$P(k) \sim k^{-\gamma},\ k >> 1,\ \ 2 < \gamma$$

Mean path length ~ lnln(k)

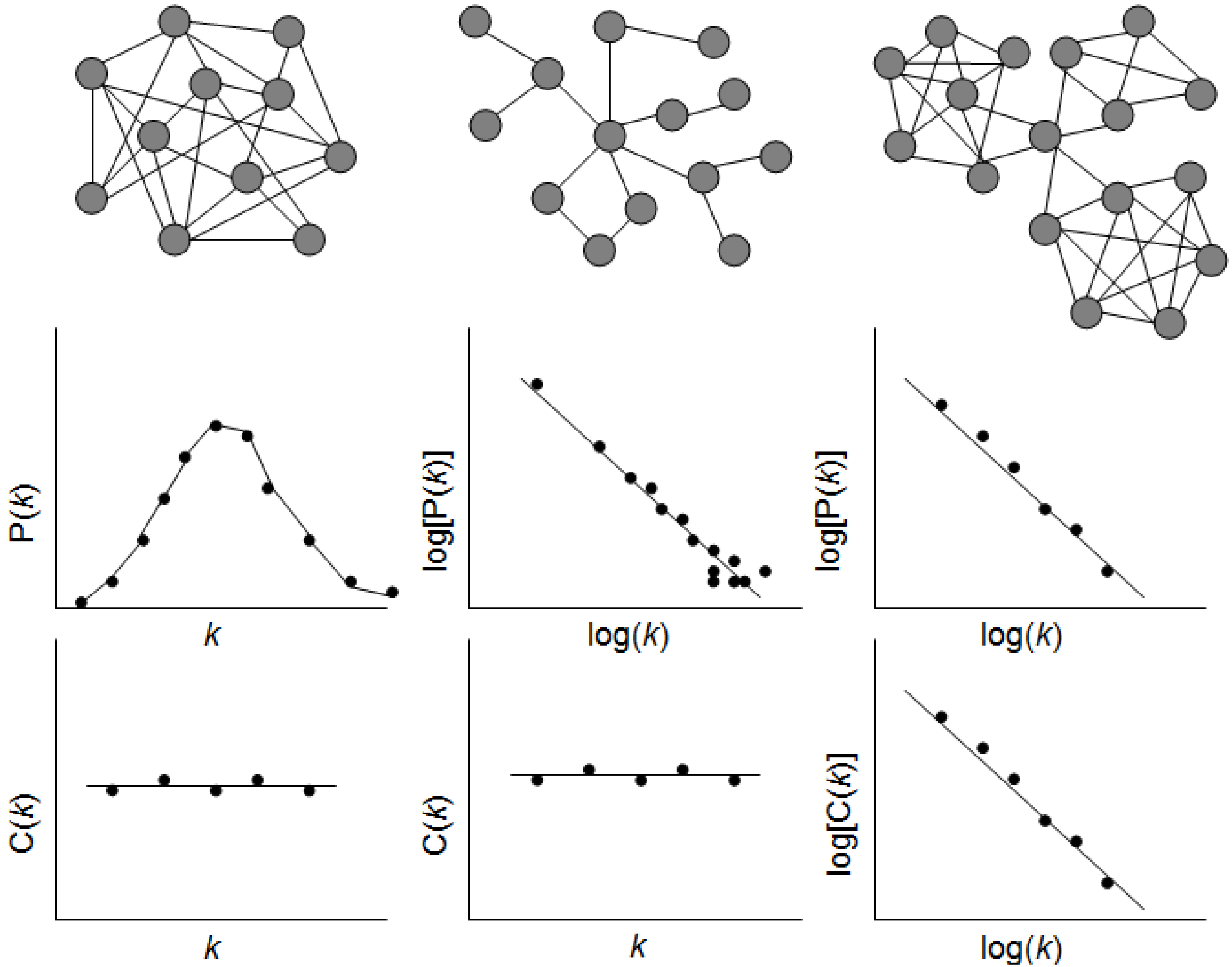
Preferential attachment. Add proportionally to connectedness

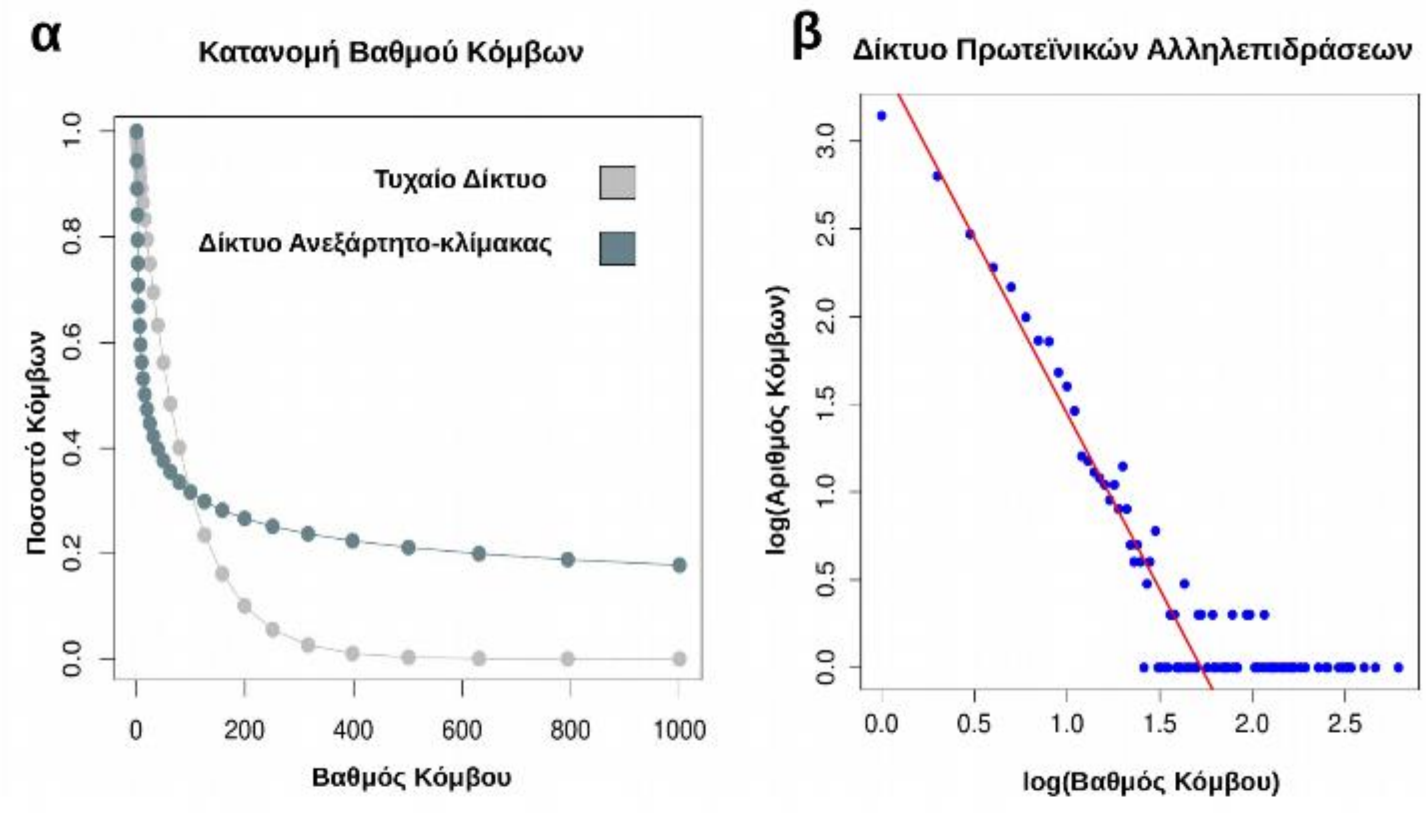


C.Hierarchial

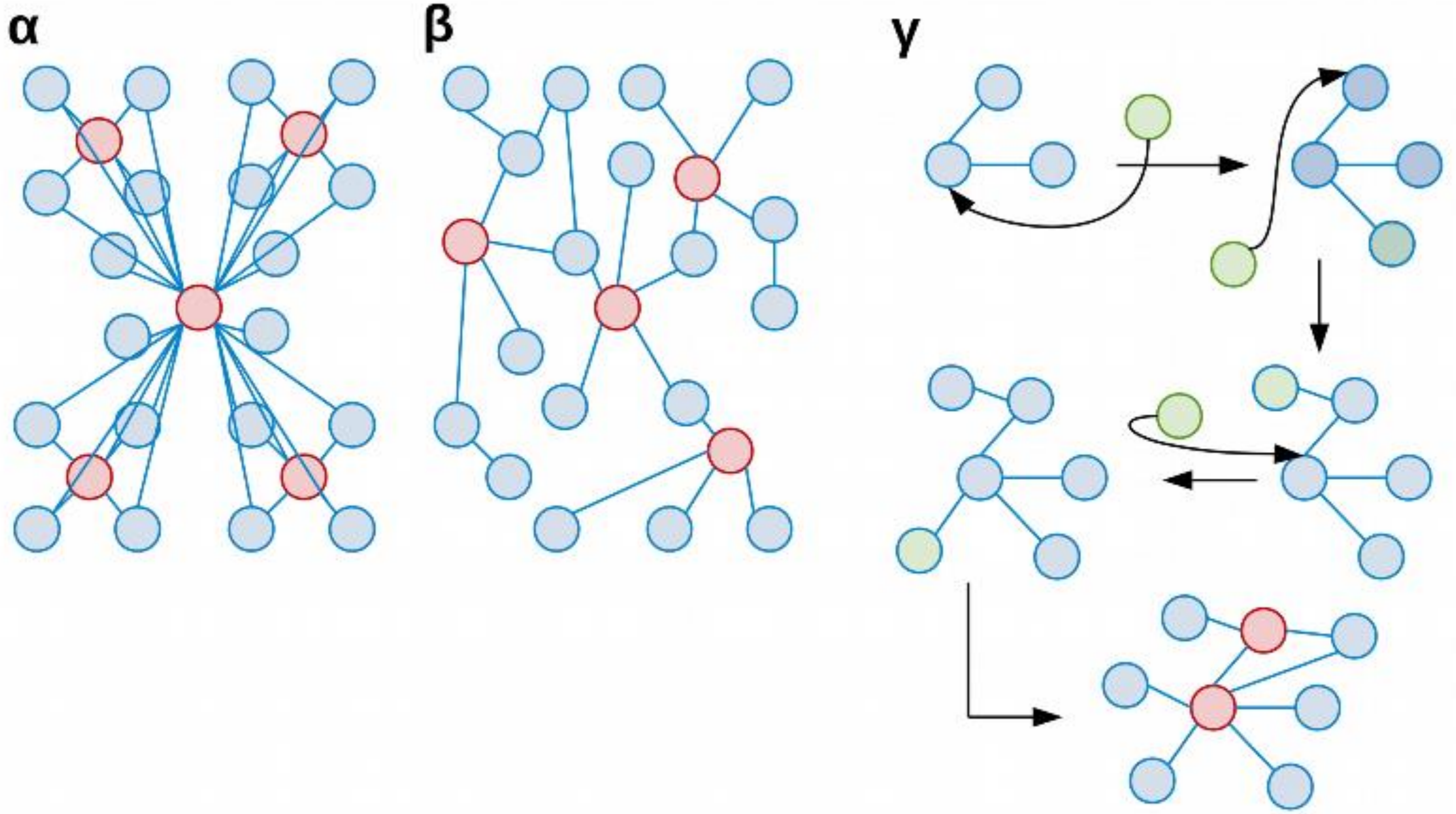


Copy smaller graphs and let them keep their connections.

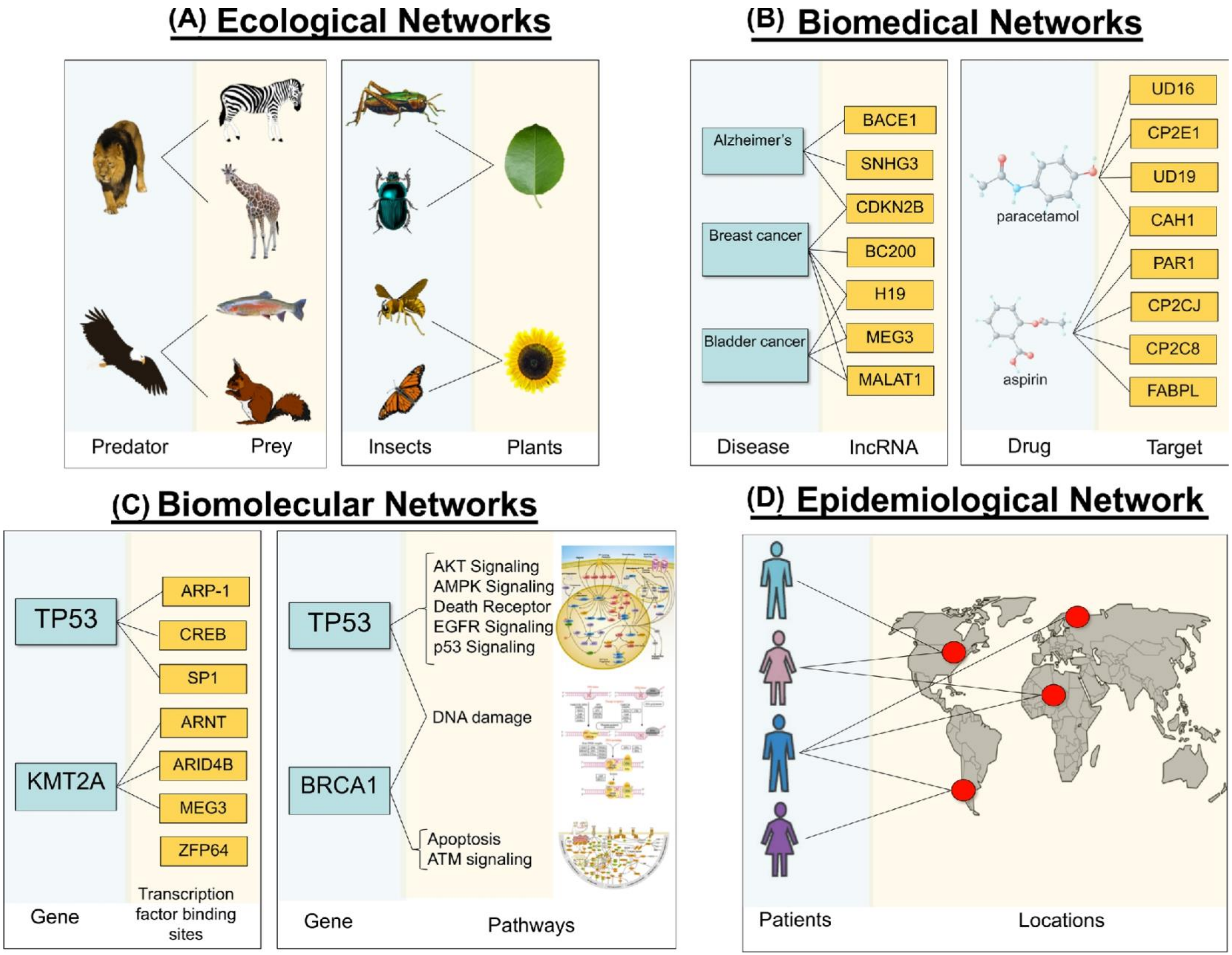

P(k)
k
log[P(k)]
log(k)
log[P(k)]
log(k)
C(k)
k
C(k)
k
log[C(k)]
log(k)



Εικόνα 9.10: *Τυχαία δίκτυα και Δίκτυα ανεξάρτητα-κλιμακας. α) Κατανομή Βαθμού Κόμβων για ένα τυχαίο δίκτυο (γκρι) κι ένα ανεξάρτητο-κλίμακας δίκτυο (γαλάζιο). Η χαρακτηριστική "μακριά ουρά" του δεύτερου είναι ενδεικτική μιας κατανομής νόμου δύναμης, η οποία δίνει ευθεία γραμμή σε διπλή λογαριθμική κλίμακα, όπως φαίνεται στο β) όπου αναπαρίσταται γραφικά η κατανομή βαθμού κόμβων για ένα δίκτυο πρωτεϊνικών αλληλεπιδράσεων (όπως αυτό της Εικόνας 9.1).*



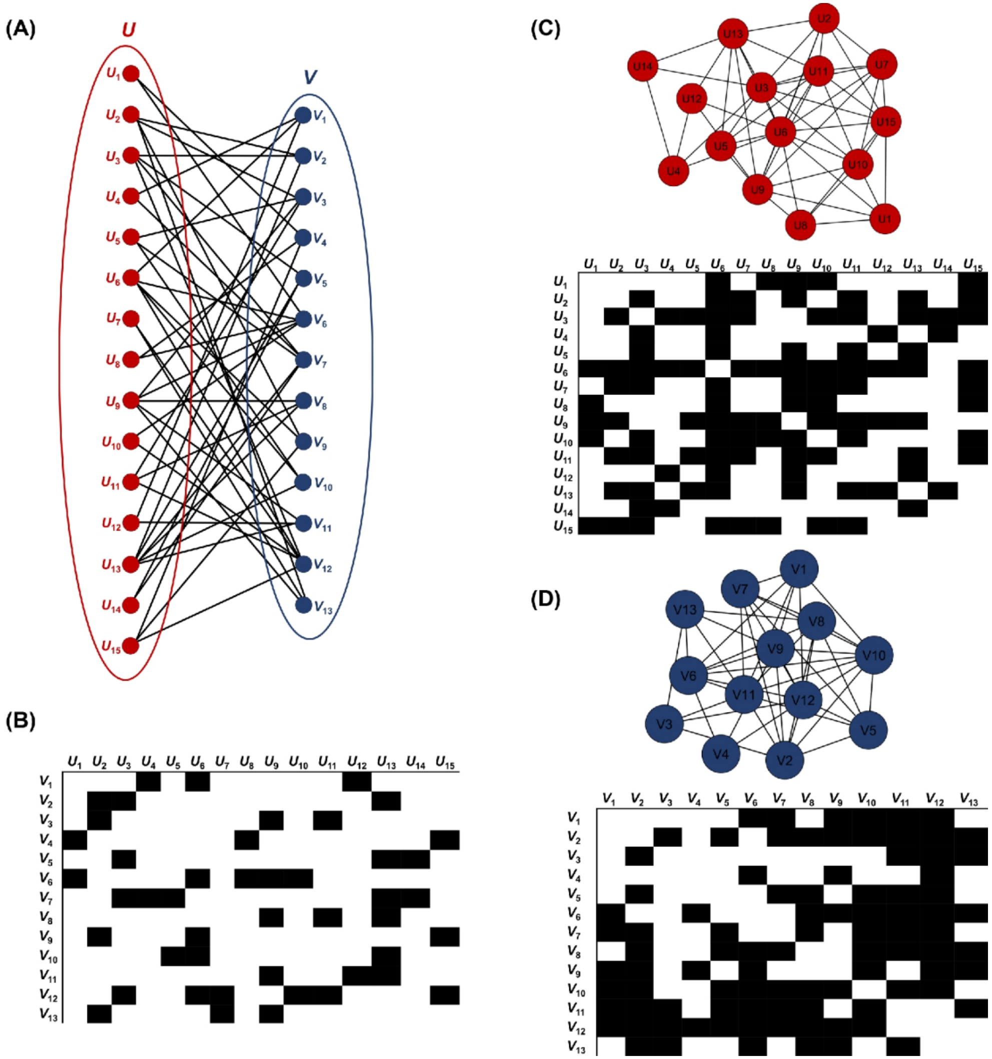
Εικόνα 9.12: *α) Ιεραρχικό Δίκτυο. β) Δίκτυο ανεξάρτητο κλίμακας γ) Το μοντέλο προτιμησιακής σύνδεσης.*



From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.


(A)
U
V
U1 U2 U3 U4 U5 U6 U7 U8 U9 U10 U11 U12 U13 U14 U15
V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13
(B)
(C)
U13 U2 U14 U11 U7 U12 U3 U15 U6 U5 U10 U4 U9 U1 U8
(D)
V1 V7 V13 V8 V9 V10 V6 V11 V12 V3 V5 V4 V2

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Properties

- **Degree**

$$\sum_{v\in V}\deg(v)=\sum_{u\in U}\deg(u)=|E|$$

- **Closeness centrality**
- **Betweenness centrality**
- **Eigenvector centrality**
- **Clustering coefficient**
- **Nestedness**

$$Q(\mathbf{A})=\frac{1}{W}\sum_{C\in P}\sum_{i,j\in C}\left[A_{ij}-\frac{k_ik_j}{W}\right]$$

- **Modularity**
- **Bipartivity**

Nestedness

- If *B* is a perfectly nested binary matrix, then there exists a permutation of rows and columns such that the set of edges in each row *i* contains the edges in row *i*+1, while the set of edges in each column *j* contains those in column *j*+1. In particular, the rows and columns of *B* can be sorted (with $B_{1,j} > 0\ \forall j$ and $B_{i,1} > 0\ \forall i$) such that $B_{i,j} \leq \min(B_{i,j-1}, B_{i-1,j})$, a property that can be extended to quantitative matrices as well

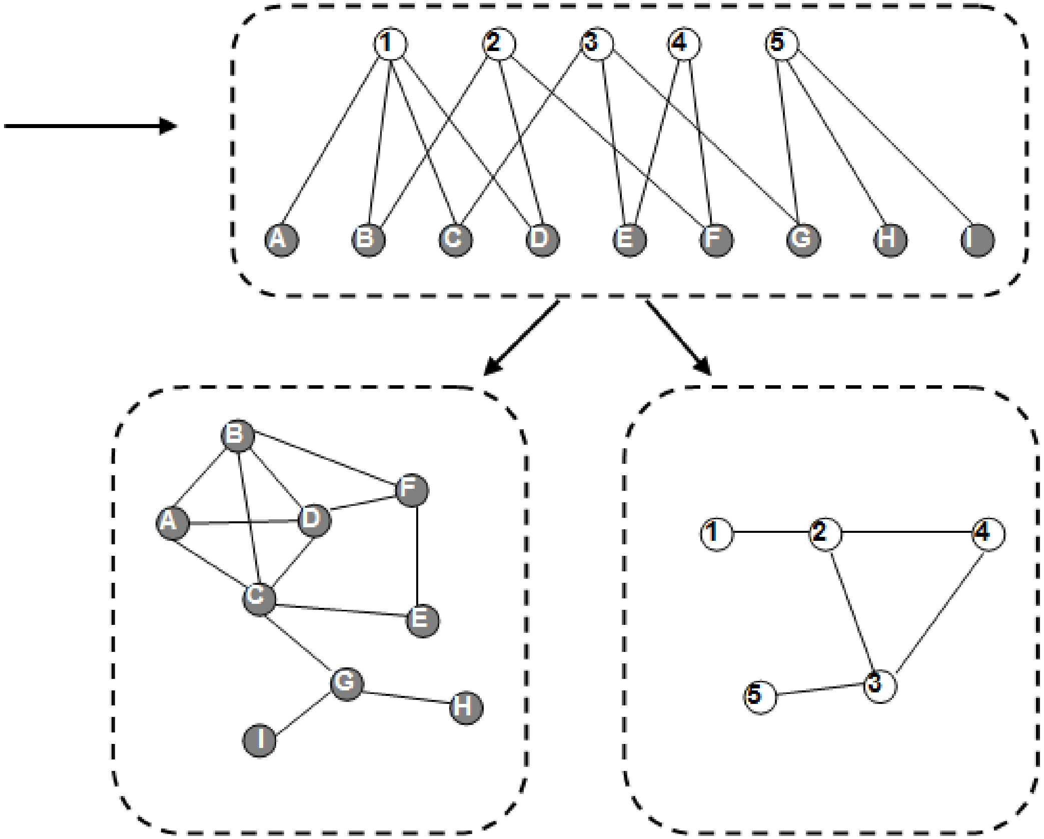
Ecological indices

- The symbol L indicates the number of realized links, whereas $|U|$and$|V|$ denote the number of species of each party in bipartite networks (e.g., hosts [U] vs. parasites [V]).
- Connectance (C) is the fraction of all possible links that are realized,$C=L/(|U|\ast|V|))$, which represents a standard measure of food web complexity.
- The related linkage density is defined as $D= L/(|U|+|V|)$.
- In a food web of $|U|$ consumers and $|V|$ prey species, the mean number of prey species (links) per consumer is termed generality, given by $G =L/|U|$, and the mean links per prey vulnerability, given by $V=L/V$.

- The web-asymmetry defines the balance between numbers in the 2 levels and it is given by $W = (|V| - |U|)/(|U| + |V|)$, where positive numbers indicate more low-trophic level species and negative more high-trophic level species.
- Most of these metrics also have a weighted counterpart, whereas there are also several other metrics designed for quantitative interactions, such as Shannon's evenness (for measuring interactions), H2 (a network-level measure of specialization based on the deviation of a species’ realized number of interactions and that expected from each species’ total number of interactions), and niche overlap (the mean similarity in the interaction patterns between species of the same trophic level).

Bipartite graphs

A	1
B	1
B	2
C	1
C	2
C	3
D	1
D	2
E	3
E	4
F	2
F	4
G	3
G	5
H	5
I	5

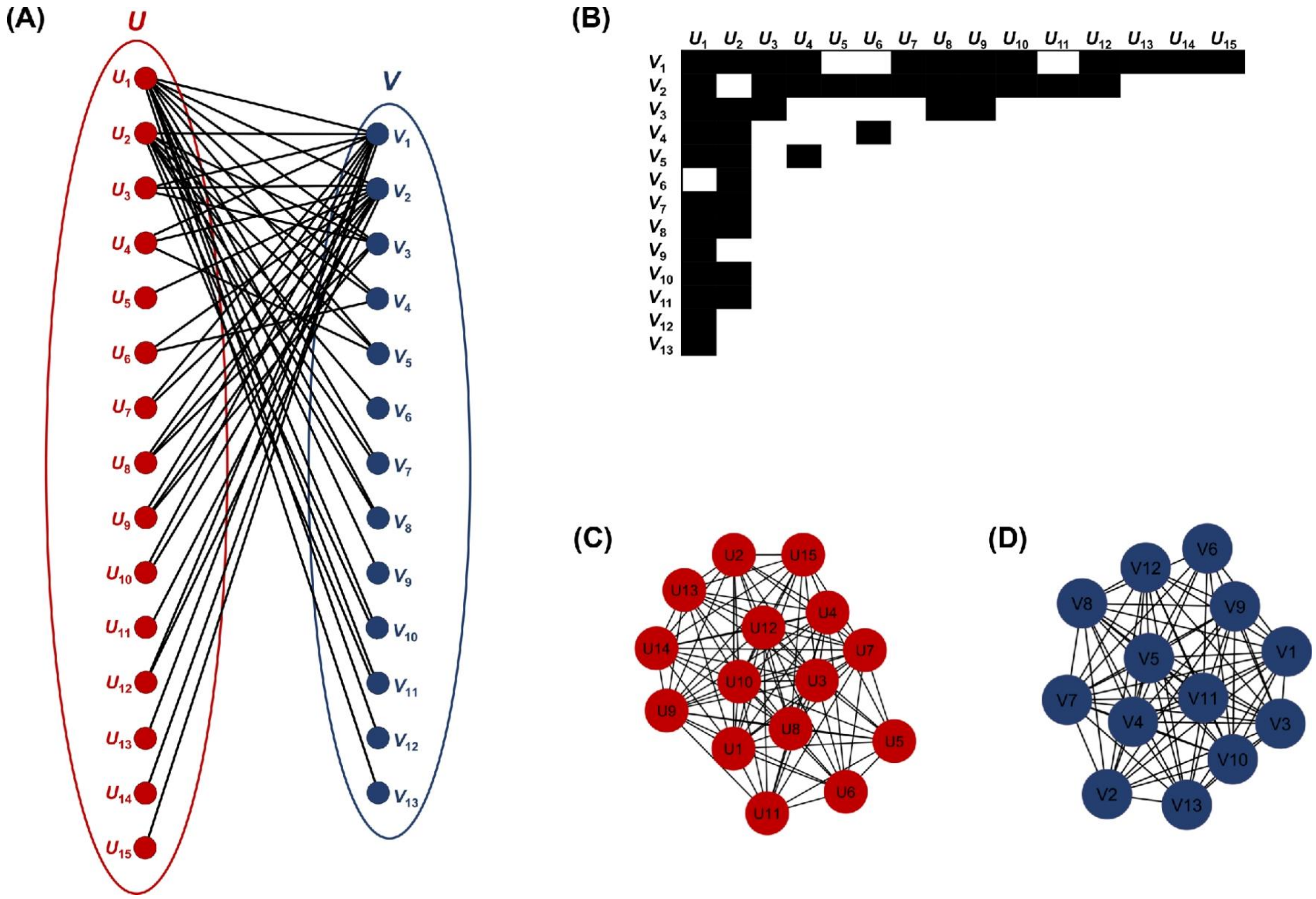


Projection

- The most complete treatment was given by Nacher and Akutsu [64] who studied the case of scale-free distributions for both sets of nodes (denoted by S-S) and that of scale-free and exponential degree distribution (denoted by S-E) for the 2 sets of nodes. They presented a mathematical analysis demonstrating that it is possible to infer the degree distributions of projected networks given the information contained in the original bipartite network, thereby deriving some simple relationships. For instance, a bipartite network with 2 sets of nodes with degree distributions $P_{\mathrm{U}}(k) \propto k^{-\gamma 1}$ and $P_{\mathrm{V}}(k) \propto k^{-\gamma 2}$ exhibits a V-projection that follows a power-law $k^{\max(-\gamma 1+1,-\gamma 2)}$ for node degree, where $\gamma 1$ and $\gamma 2$ indicate the power law exponents of the distribution of U and V nodes, respectively, in the bipartite network. On the other hand, a bipartite network with 2 sets of nodes with degree distributions $P_{\mathrm{U}}(k) \propto k^{-\gamma 1}$ and $P_{\mathrm{V}}(k) \propto \exp(-\lambda k)$ leads to a V-projection, defined by a power-law $k^{-\gamma 1+1}$ node degree distribution. The analytical results were confirmed by computer simulations performed using artificially constructed networks [64].

Projection

- Various methods of bipartite network projection have been proposed in the literature, and they all involve the use of a threshold, and, in most cases, they yield weighted unipartite networks. Usually, edges, the weights of which exceed the threshold value, are retained, while those with weights that are below the threshold value are omitted. The methods greatly vary, however, on the way threshold values are identified. The simplest and most widespread approach for extracting the backbone of bipartite projections is through the application of an unconditional (or global) threshold. In particular, a single weight threshold is selected and applied to all edges in the bipartite projection, and edges are retained in the backbone network only if their weight in the bipartite projection exceeds this predefined threshold. The most commonly used weight threshold of zero preserves all edges with a non-zero weight, whereas others have used different thresholds, including these sets at the percentage of the maximum observed edge weight or at the mean observed edge weight. The unconditional threshold approach, although widely used, suffers from several shortcomings. In general, if the presence of any shared connections to V-nodes is considered adequate for inferring that an edge exists between 2 U-nodes, then an unconditional threshold should be used for backbone network extraction. If, however, an instance of shared V-nodes is not sufficient to infer that an edge exists between 2 U-nodes, then unconditional threshold backbones may be problematic. The structure of a backbone extracted by using an unconditional threshold depends heavily on the selected threshold value; moreover, certain structural features of unconditional threshold backbones of bipartite networks are systematically biased. Thus, this approach in which a universal threshold is applied indiscriminately to all edge weights can yield a 1-mode projection with several undesirable properties.

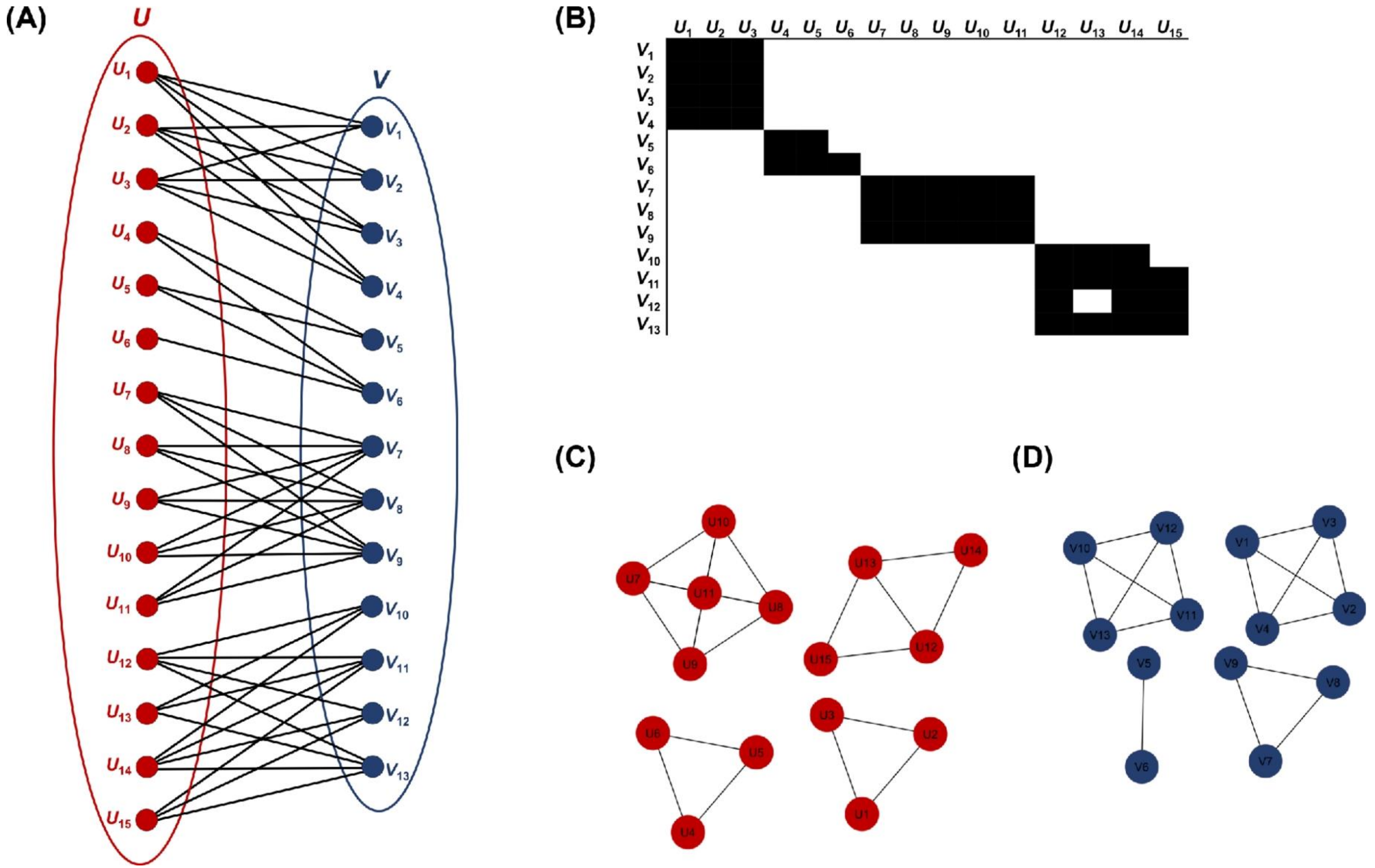


	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}	U_{13}	U_{14}	U_{15}
V_1	■	■	■	■			■	■	■	■		■	■	■	■
V_2	■		■	■	■	■	■	■	■	■	■	■			
V_3	■	■	■					■	■						
V_4	■	■				■									
V_5	■	■		■											
V_6		■													
V_7	■	■													
V_8	■	■													
V_9	■														
V_{10}	■	■													
V_{11}	■	■													
V_{12}	■														
V_{13}	■														

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

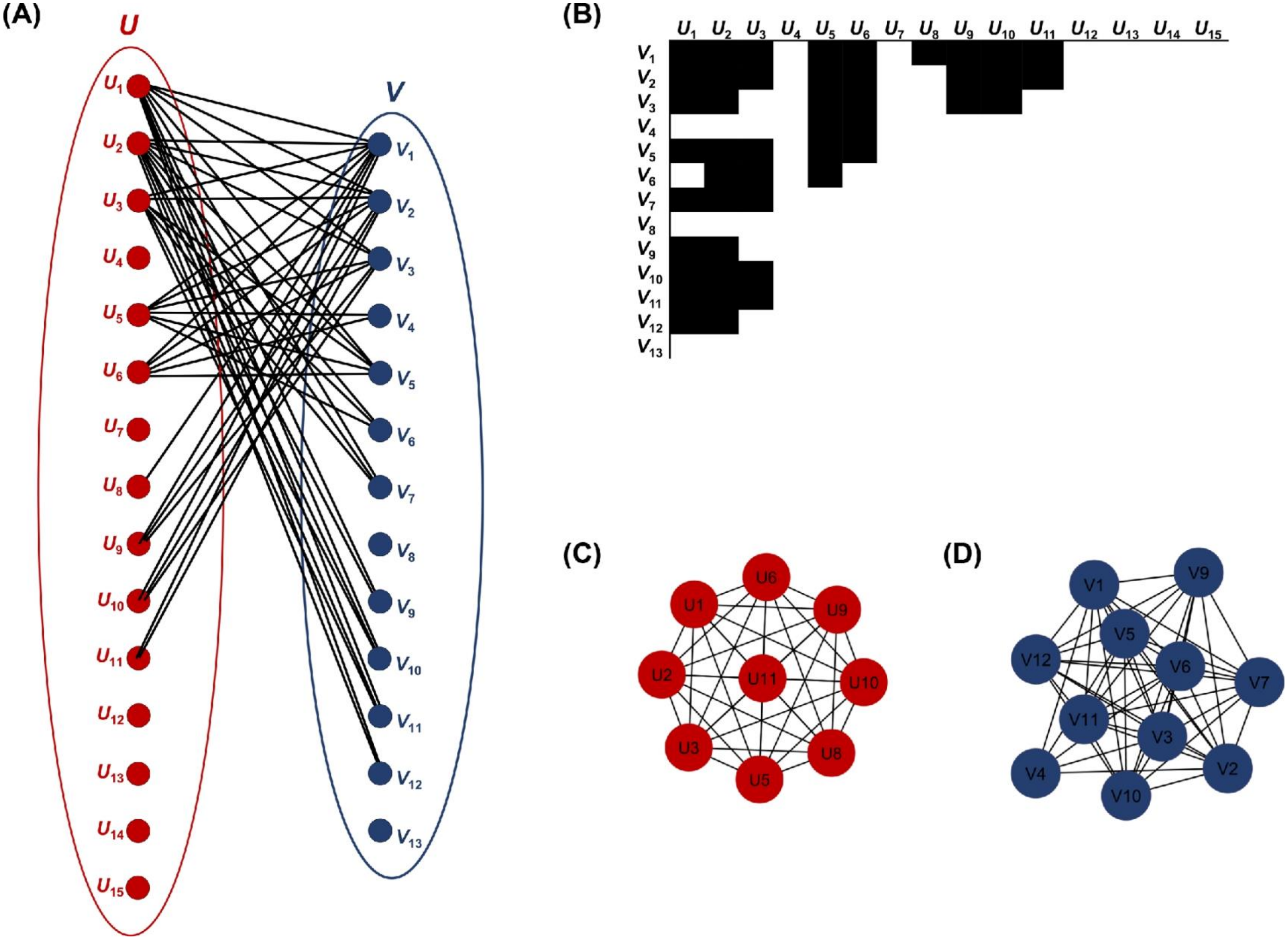


	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}	U_{13}	U_{14}	U_{15}
V_1	■	■	■												
V_2	■	■	■												
V_3	■	■	■												
V_4	■	■	■												
V_5				■	■										
V_6				■	■	■									
V_7							■	■	■	■	■				
V_8							■	■	■	■	■				
V_9							■	■	■	■	■				
V_{10}												■	■	■	
V_{11}												■	■	■	■
V_{12}												■		■	■
V_{13}												■	■	■	■

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

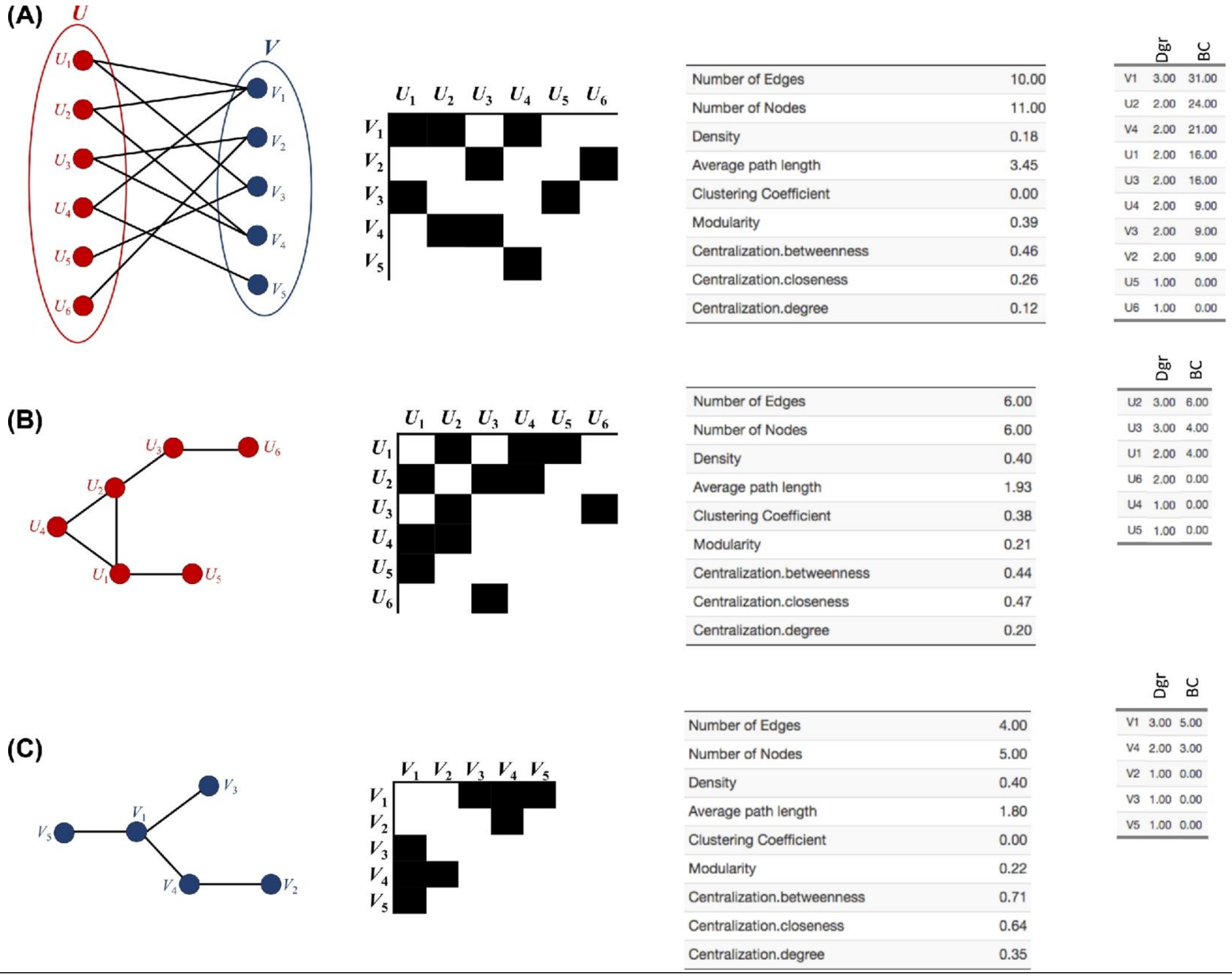


	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}	U_{13}	U_{14}	U_{15}
V_1	■	■	■		■	■		■	■	■	■				
V_2	■	■	■		■	■			■	■	■				
V_3	■	■			■	■			■	■					
V_4					■	■									
V_5	■	■	■		■	■									
V_6		■	■		■										
V_7	■	■	■												
V_8															
V_9	■	■													
V_{10}	■	■	■												
V_{11}	■	■	■												
V_{12}	■	■													
V_{13}															

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



	U_1	U_2	U_3	U_4	U_5	U_6
V_1	■	■		■		
V_2			■			■
V_3	■				■	
V_4		■	■			
V_5				■		

Number of Edges	10.00
Number of Nodes	11.00
Density	0.18
Average path length	3.45
Clustering Coefficient	0.00
Modularity	0.39
Centralization.betweenness	0.46
Centralization.closeness	0.26
Centralization.degree	0.12

	Dgr	BC
V1	3.00	31.00
U2	2.00	24.00
V4	2.00	21.00
U1	2.00	16.00
U3	2.00	16.00
U4	2.00	9.00
V3	2.00	9.00
V2	2.00	9.00
U5	1.00	0.00
U6	1.00	0.00

	U_1	U_2	U_3	U_4	U_5	U_6
U_1		■		■	■	
U_2	■		■	■		
U_3		■				■
U_4	■	■				
U_5	■					
U_6			■			

Number of Edges	6.00
Number of Nodes	6.00
Density	0.40
Average path length	1.93
Clustering Coefficient	0.38
Modularity	0.21
Centralization.betweenness	0.44
Centralization.closeness	0.47
Centralization.degree	0.20

	Dgr	BC
U2	3.00	6.00
U3	3.00	4.00
U1	2.00	4.00
U6	2.00	0.00
U4	1.00	0.00
U5	1.00	0.00

	V_1	V_2	V_3	V_4	V_5
V_1			■	■	■
V_2				■	
V_3	■				
V_4	■	■			
V_5	■				

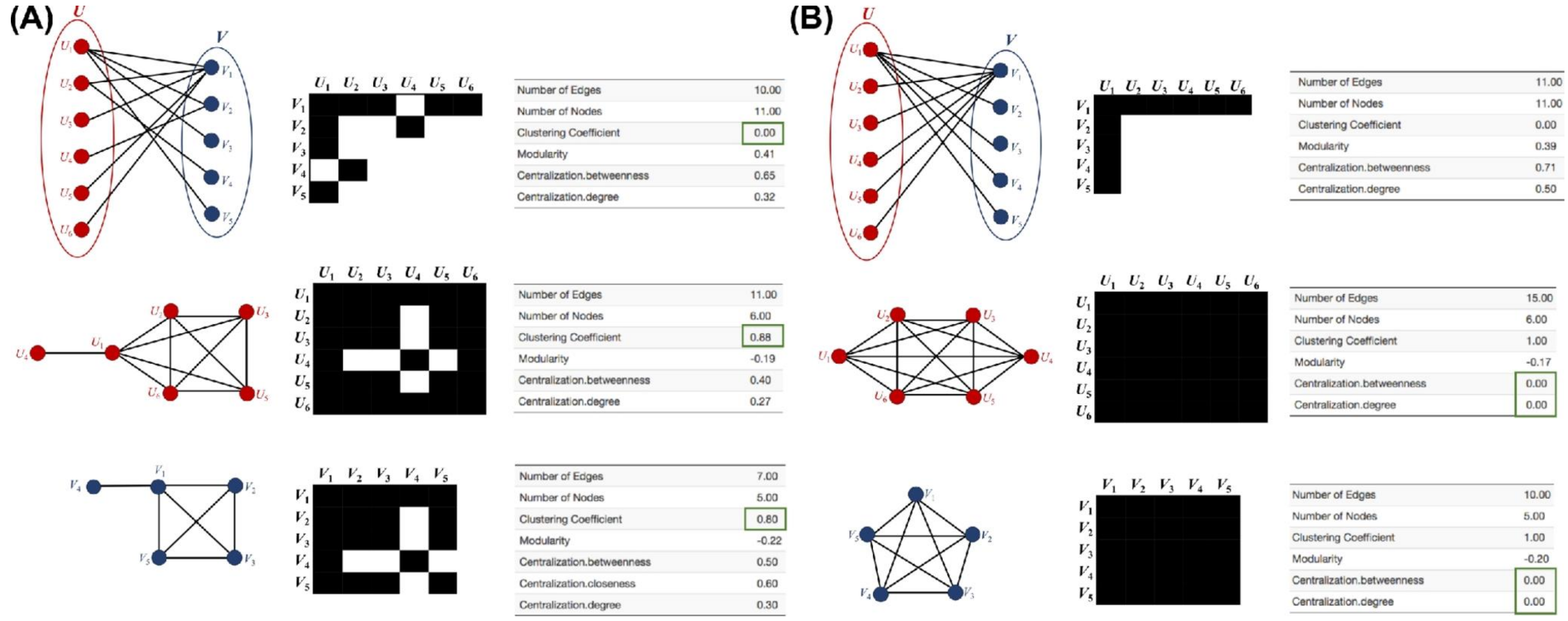
Number of Edges	4.00
Number of Nodes	5.00
Density	0.40
Average path length	1.80
Clustering Coefficient	0.00
Modularity	0.22
Centralization.betweenness	0.71
Centralization.closeness	0.64
Centralization.degree	0.35

	Dgr	BC
V1	3.00	5.00
V4	2.00	3.00
V2	1.00	0.00
V3	1.00	0.00
V5	1.00	0.00

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.


(A)
U
V
U_1 U_2 U_3 U_4 U_5 U_6
V_1 V_2 V_3 V_4 V_5
(B)

	U_1	U_2	U_3	U_4	U_5	U_6
V_1	■	■	■	□	■	■
V_2	■	□	□	■	□	□
V_3	■	□	□	□	□	□
V_4	□	■	□	□	□	□
V_5	■	□	□	□	□	□

Number of Edges	10.00
Number of Nodes	11.00
Clustering Coefficient	0.00
Modularity	0.41
Centralization.betweenness	0.65
Centralization.degree	0.32

	U_1	U_2	U_3	U_4	U_5	U_6
U_1	■	■	■	■	■	■
U_2	■	■	■	□	■	■
U_3	■	■	■	□	■	■
U_4	■	□	□	■	□	■
U_5	■	■	■	□	■	■
U_6	■	■	■	■	■	■

Number of Edges	11.00
Number of Nodes	6.00
Clustering Coefficient	0.88
Modularity	-0.19
Centralization.betweenness	0.40
Centralization.degree	0.27

	V_1	V_2	V_3	V_4	V_5
V_1	■	■	■	■	■
V_2	■	■	■	□	■
V_3	■	■	■	□	■
V_4	■	□	□	■	□
V_5	■	■	■	□	■

Number of Edges	7.00
Number of Nodes	5.00
Clustering Coefficient	0.80
Modularity	-0.22
Centralization.betweenness	0.50
Centralization.closeness	0.60
Centralization.degree	0.30

	U_1	U_2	U_3	U_4	U_5	U_6
V_1	■	■	■	■	■	■
V_2	■	□	□	□	□	□
V_3	■	□	□	□	□	□
V_4	■	□	□	□	□	□
V_5	■	□	□	□	□	□

Number of Edges	11.00
Number of Nodes	11.00
Clustering Coefficient	0.00
Modularity	0.39
Centralization.betweenness	0.71
Centralization.degree	0.50

	U_1	U_2	U_3	U_4	U_5	U_6
U_1	■	■	■	■	■	■
U_2	■	■	■	■	■	■
U_3	■	■	■	■	■	■
U_4	■	■	■	■	■	■
U_5	■	■	■	■	■	■
U_6	■	■	■	■	■	■

Number of Edges	15.00
Number of Nodes	6.00
Clustering Coefficient	1.00
Modularity	-0.17
Centralization.betweenness	0.00
Centralization.degree	0.00

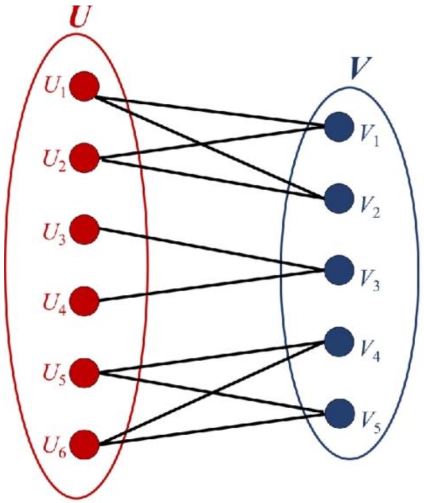
	V_1	V_2	V_3	V_4	V_5
V_1	■	■	■	■	■
V_2	■	■	■	■	■
V_3	■	■	■	■	■
V_4	■	■	■	■	■
V_5	■	■	■	■	■

Number of Edges	10.00
Number of Nodes	5.00
Clustering Coefficient	1.00
Modularity	-0.20
Centralization.betweenness	0.00
Centralization.degree	0.00

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

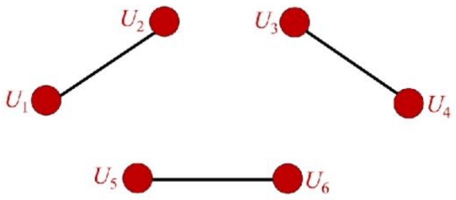
Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



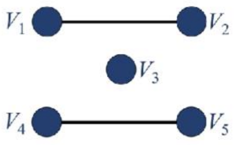
	U_1	U_2	U_3	U_4	U_5	U_6
V_1	■	■				
V_2	■	■				
V_3			■	■		
V_4					■	■
V_5					■	■

Number of Edges	10.00
Number of Nodes	11.00
Centralization.betweenness	0.01
Centralization.degree	0.02



	U_1	U_2	U_3	U_4	U_5	U_6
U_1	■	■				
U_2	■	■				
U_3			■	■		
U_4			■	■		
U_5					■	■
U_6					■	■

Number of Edges	3.00
Number of Nodes	6.00
Centralization.betweenness	0.00
Centralization.degree	0.00



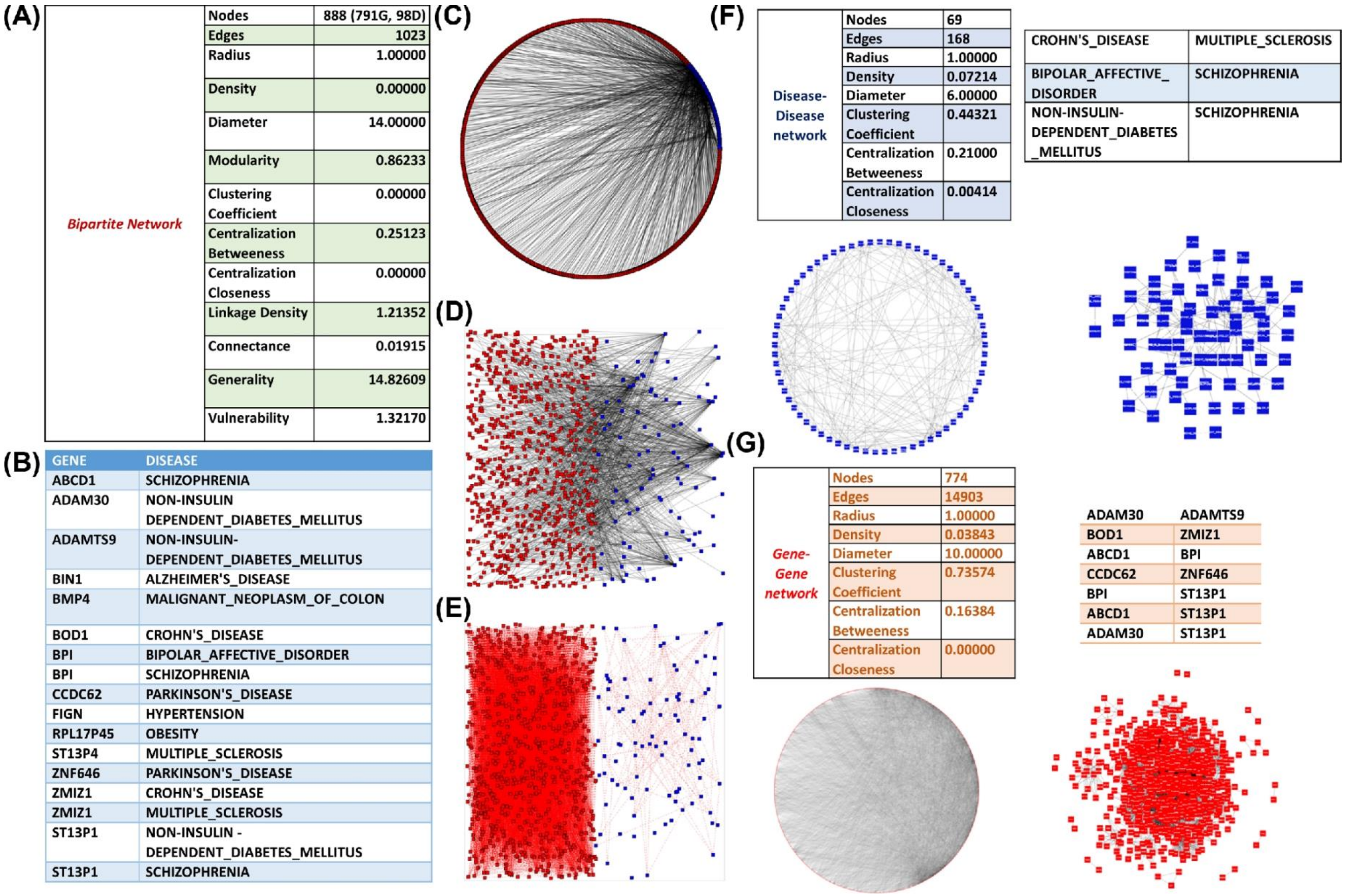
	V_1	V_2	V_3	V_4	V_5
V_1	■	■			
V_2	■	■			
V_3			■		
V_4				■	■
V_5				■	■

Number of Edges	2.00
Number of Nodes	4.00
Centralization.betweenness	0.00
Centralization.degree	0.00

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



Bipartite Network		
	Nodes	888 (791G, 98D)
	Edges	1023
	Radius	1.00000
	Density	0.00000
	Diameter	14.00000
	Modularity	0.86233
	Clustering Coefficient	0.00000
	Centralization Betweeness	0.25123
	Centralization Closeness	0.00000
	Linkage Density	1.21352
	Connectance	0.01915
	Generality	14.82609
	Vulnerability	1.32170

GENE	DISEASE
ABCD1	SCHIZOPHRENIA
ADAM30	NON-INSULIN DEPENDENT_DIABETES_MELLITUS
ADAMTS9	NON-INSULIN-DEPENDENT_DIABETES_MELLITUS
BIN1	ALZHEIMER'S_DISEASE
BMP4	MALIGNANT_NEOPLASM_OF_COLON
BOD1	CROHN'S_DISEASE
BPI	BIPOLAR_AFFECTIVE_DISORDER
BPI	SCHIZOPHRENIA
CCDC62	PARKINSON'S_DISEASE
FIGN	HYPERTENSION
RPL17P45	OBESITY
ST13P4	MULTIPLE_SCLEROSIS
ZNF646	PARKINSON'S_DISEASE
ZMIZ1	CROHN'S_DISEASE
ZMIZ1	MULTIPLE_SCLEROSIS
ST13P1	NON-INSULIN - DEPENDENT_DIABETES_MELLITUS
ST13P1	SCHIZOPHRENIA

Disease-Disease network		
	Nodes	69
	Edges	168
	Radius	1.00000
	Density	0.07214
	Diameter	6.00000
	Clustering Coefficient	0.44321
	Centralization Betweeness	0.21000
	Centralization Closeness	0.00414

CROHN'S_DISEASE	MULTIPLE_SCLEROSIS
BIPOLAR_AFFECTIVE_DISORDER	SCHIZOPHRENIA
NON-INSULIN-DEPENDENT_DIABETES_MELLITUS	SCHIZOPHRENIA

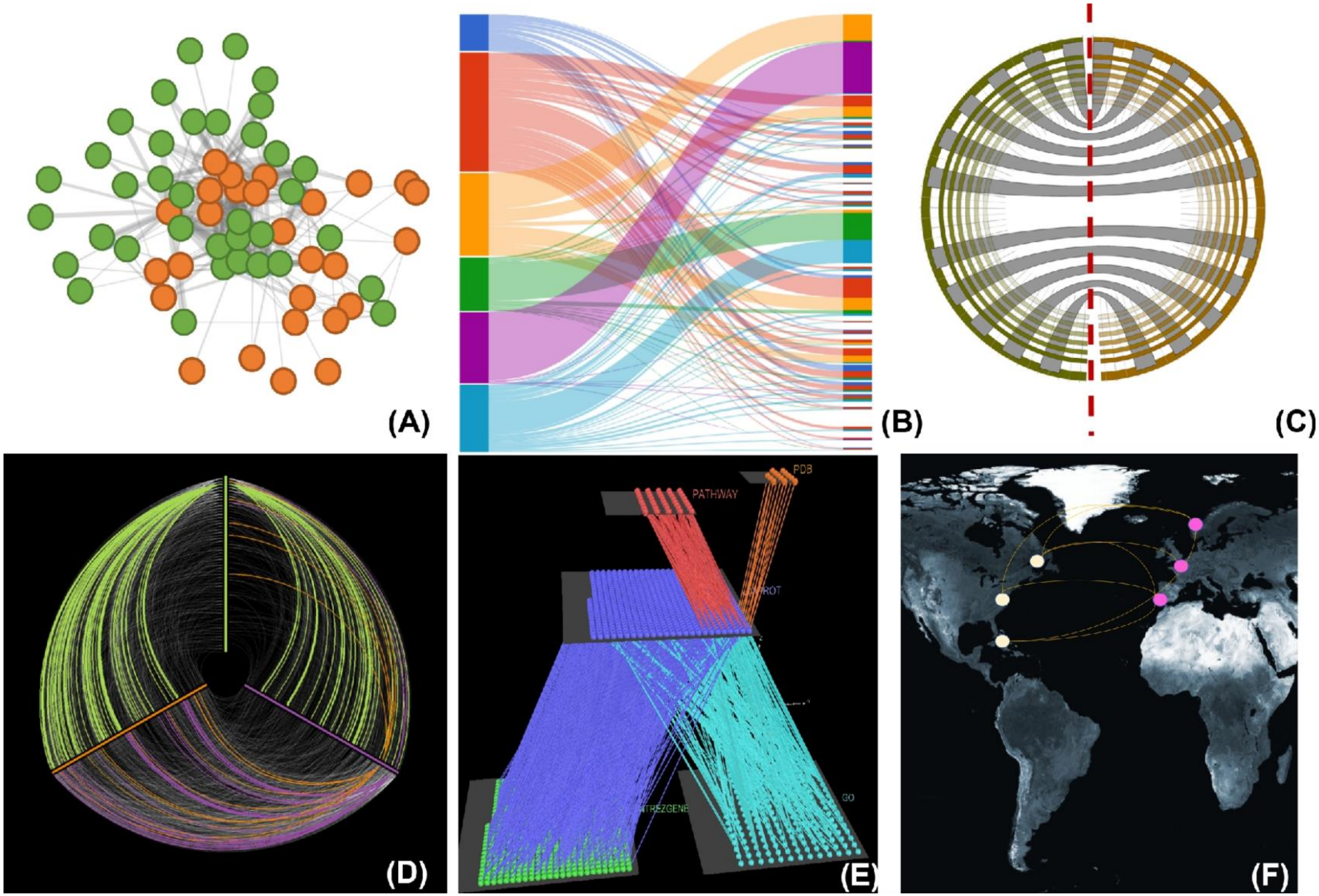
Gene-Gene network		
	Nodes	774
	Edges	14903
	Radius	1.00000
	Density	0.03843
	Diameter	10.00000
	Clustering Coefficient	0.73574
	Centralization Betweeness	0.16384
	Centralization Closeness	0.00000

ADAM30	ADAMTS9
BOD1	ZMIZ1
ABCD1	BPI
CCDC62	ZNF646
BPI	ST13P1
ABCD1	ST13P1
ADAM30	ST13P1

From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



From: Bipartite graphs in systems biology and medicine: a survey of methods and applications

Gigascience. 2018;7(4). doi:10.1093/gigascience/giy014

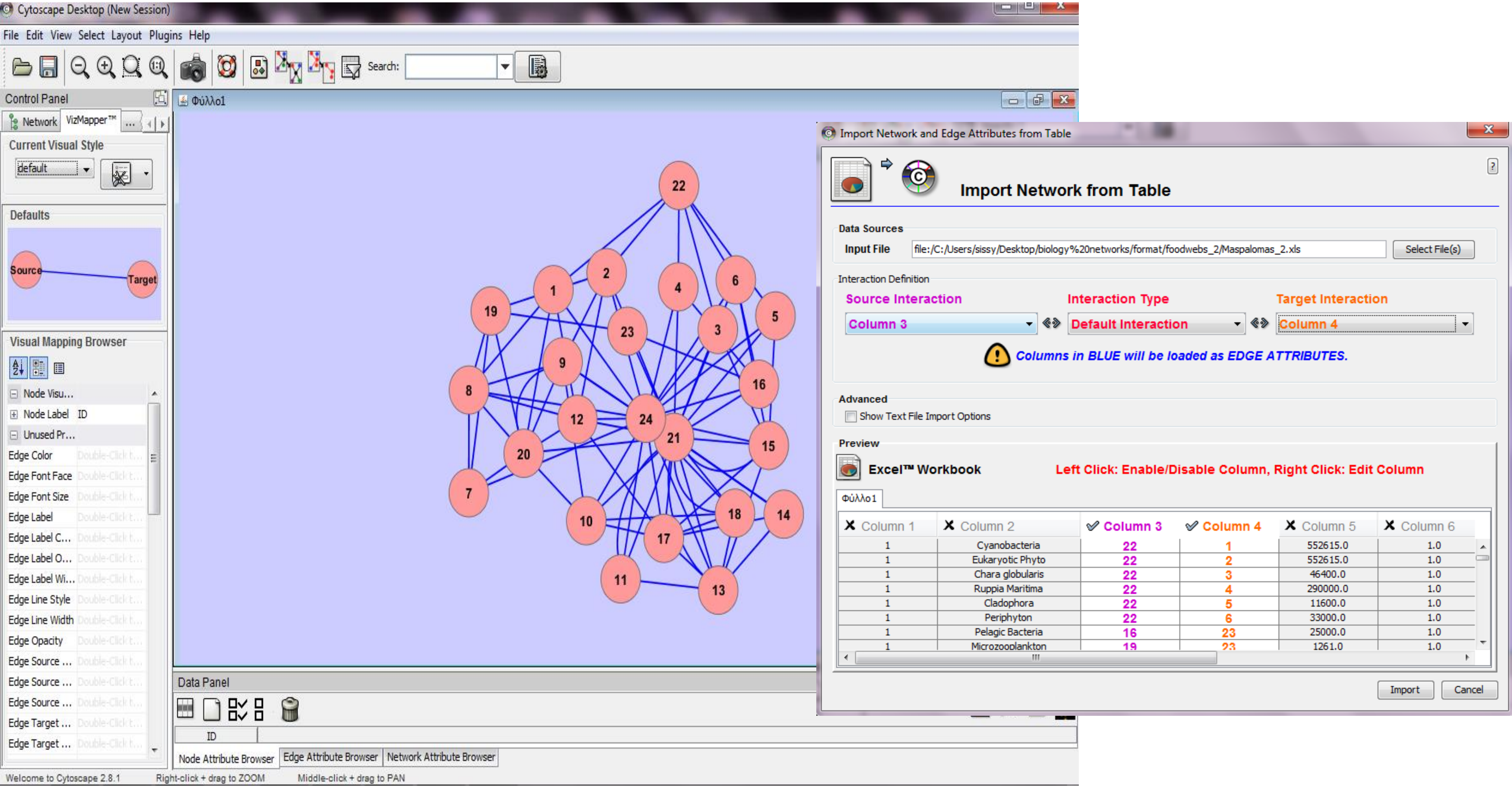
Gigascience | © The Author(s) 2018. Published by Oxford University Press.This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Cytoscape

- Cytoscape is an open source bioinformatics software platform for visualizing molecular interaction networks and integrating with gene expression profiles and other state data. Additional features are available as plugins. Plugins are available for network and molecular profiling analyses, new layouts, additional file format support and connection with databases and searching in large networks. Plugins may be developed using the Cytoscape open Java software architecture by anyone and plugin community development is encouraged.Cytoscape also has a JavaScript-centric sister project named Cytoscape.js that can be used to analyse and visualise graphs in JavaScript environments, like a browser.

Cytoscape

- Λογισμικό ανοιχτού πηγαίου κώδικα
- Πλατφόρμα σχεδίασης, σύνθετης ανάλυσης και παρουσίασης διαφόρων δικτύων



Input File: file:/C:/Users/sissy/Desktop/biology%20networks/format/foodwebs_2/Maspalomas_2.xls

Source Interaction	Interaction Type	Target Interaction
Column 3	Default Interaction	Column 4

Columns in BLUE will be loaded as EDGE ATTRIBUTES.

☐ Show Text File Import Options

Left Click: Enable/Disable Column, Right Click: Edit Column

Φύλλο1

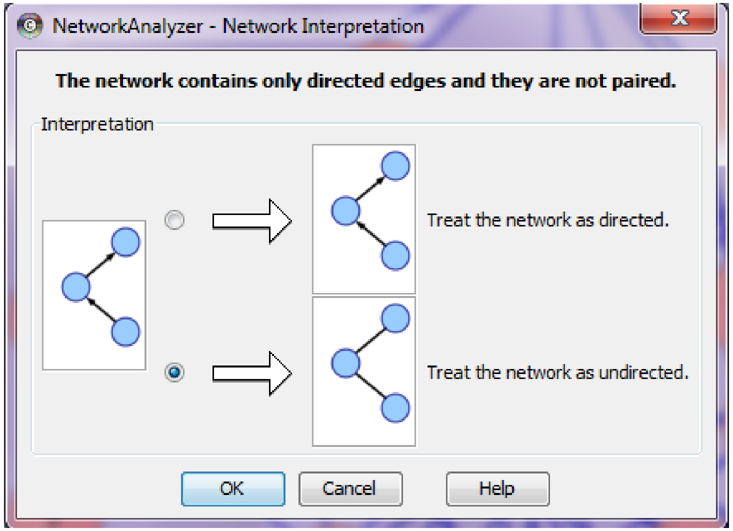
✗ Column 1	✗ Column 2	✓ Column 3	✓ Column 4	✗ Column 5	✗ Column 6
1	Cyanobacteria	22	1	552615.0	1.0
1	Eukaryotic Phyto	22	2	552615.0	1.0
1	Chara globularis	22	3	46400.0	1.0
1	Ruppia Maritima	22	4	290000.0	1.0
1	Cladophora	22	5	11600.0	1.0
1	Periphyton	22	6	33000.0	1.0
1	Pelagic Bacteria	16	23	25000.0	1.0
1	Microzooplankton	19	23	1261.0	1.0

Schwikowski B, Ideker T: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 2003, 13(11):2498-2504

Network-Analyzer

Εργαλείο ανάλυσης που υπολογίζει σε ένα δίκτυο τις βασικές και τις σύνθετες τοπολογικές παραμέτρους

- Εισαγωγή δικτύου
- Επιλογή τύπου δικτύου (κατευθυνόμενος- μη κατευθυνόμενος γράφος)
- Υπολογισμός βασικών (μία τιμή)
- Υπολογισμός σύνθετων (κατανομή)



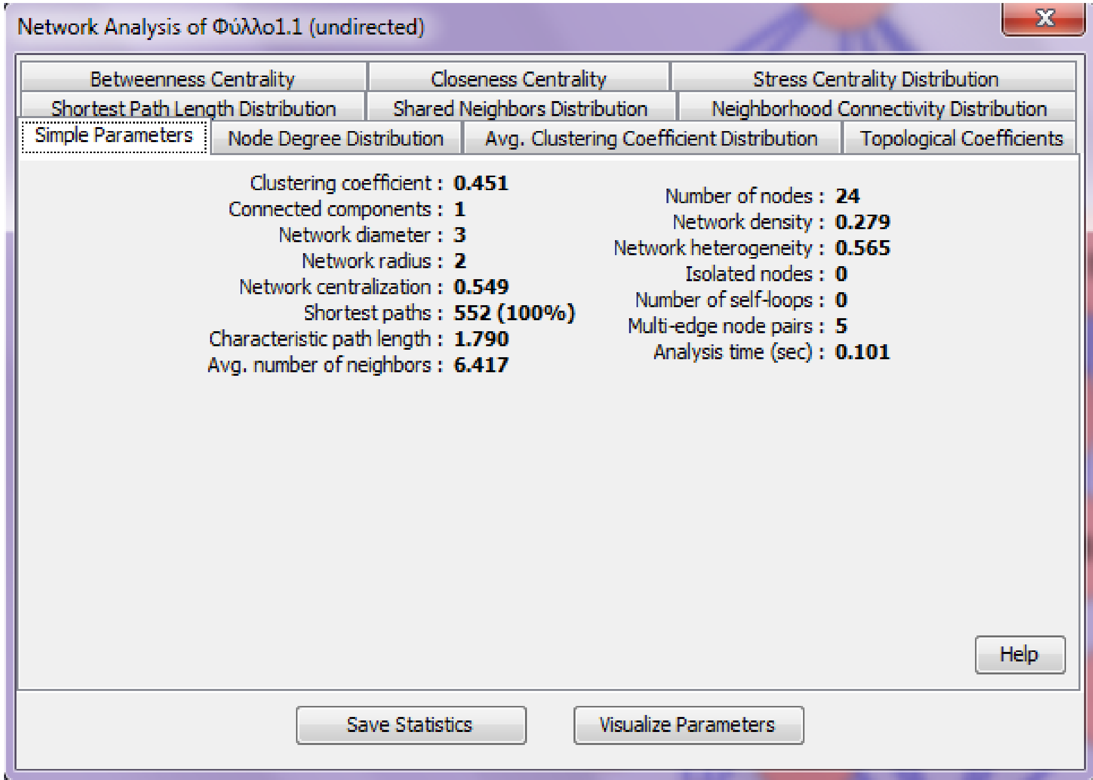
NetworkAnalyzer - Network Interpretation

The network contains only directed edges and they are not paired.

Interpretation

○ Treat the network as directed.

● Treat the network as undirected.



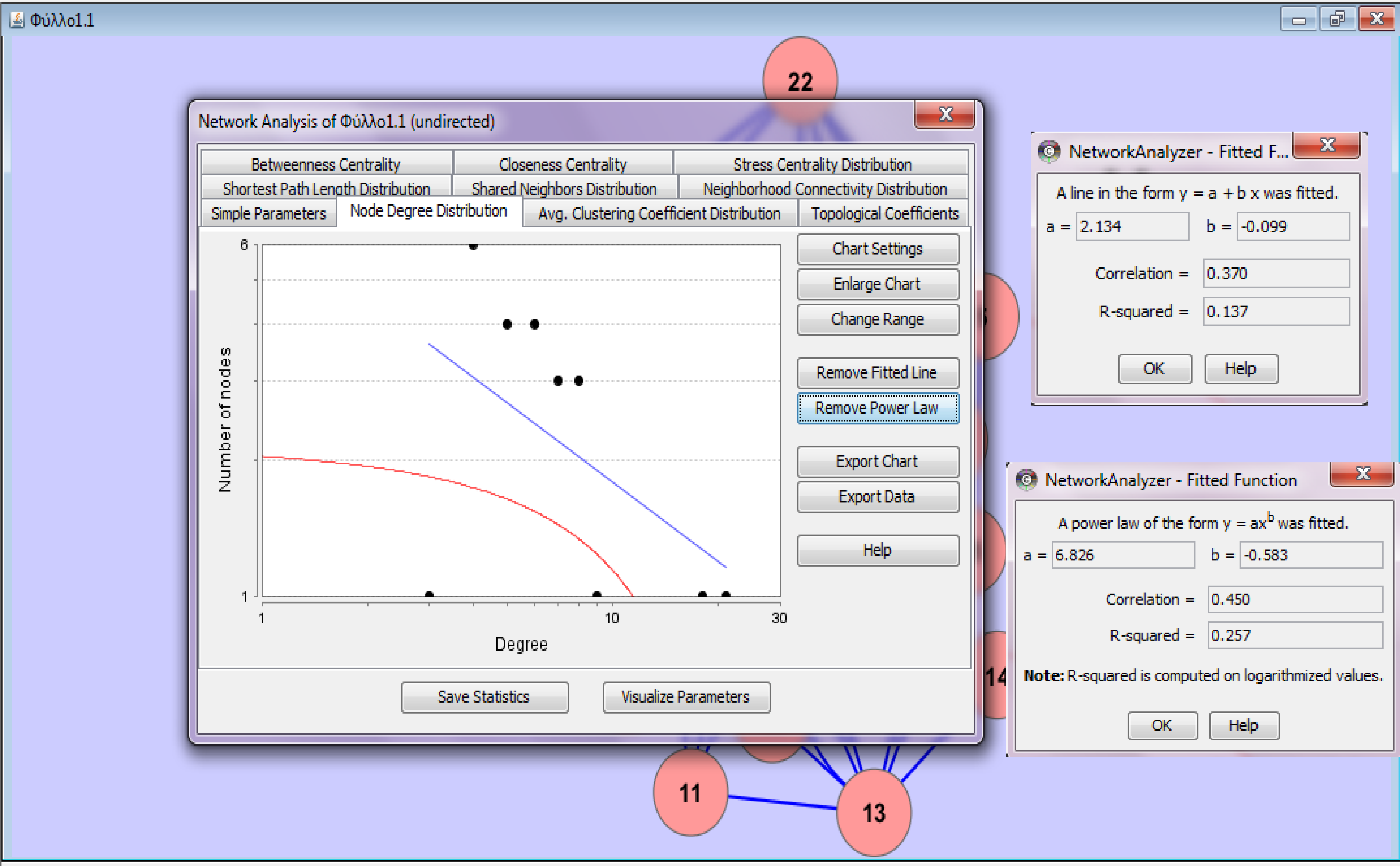
Network Analysis of Φύλλο1.1 (undirected)

Betweenness Centrality
Closeness Centrality
Stress Centrality Distribution
Shortest Path Length Distribution
Shared Neighbors Distribution
Neighborhood Connectivity Distribution
Simple Parameters
Node Degree Distribution
Avg. Clustering Coefficient Distribution
Topological Coefficients

```
      Clustering coefficient :  0.451
        Connected components :  1
            Network diameter :  3
              Network radius :  2
      Network centralization :  0.549
              Shortest paths :  552 (100%)
  Characteristic path length :  1.790
   Avg. number of neighbors :  6.417

             Number of nodes :  24
             Network density :  0.279
       Network heterogeneity :  0.565
              Isolated nodes :  0
       Number of self-loops :  0
       Multi-edge node pairs :  5
         Analysis time (sec) :  0.101
```

Network-Analyzer



A line in the form y = a + b x was fitted.

a = 2.134 b = -0.099

Correlation = 0.370

R-squared = 0.137

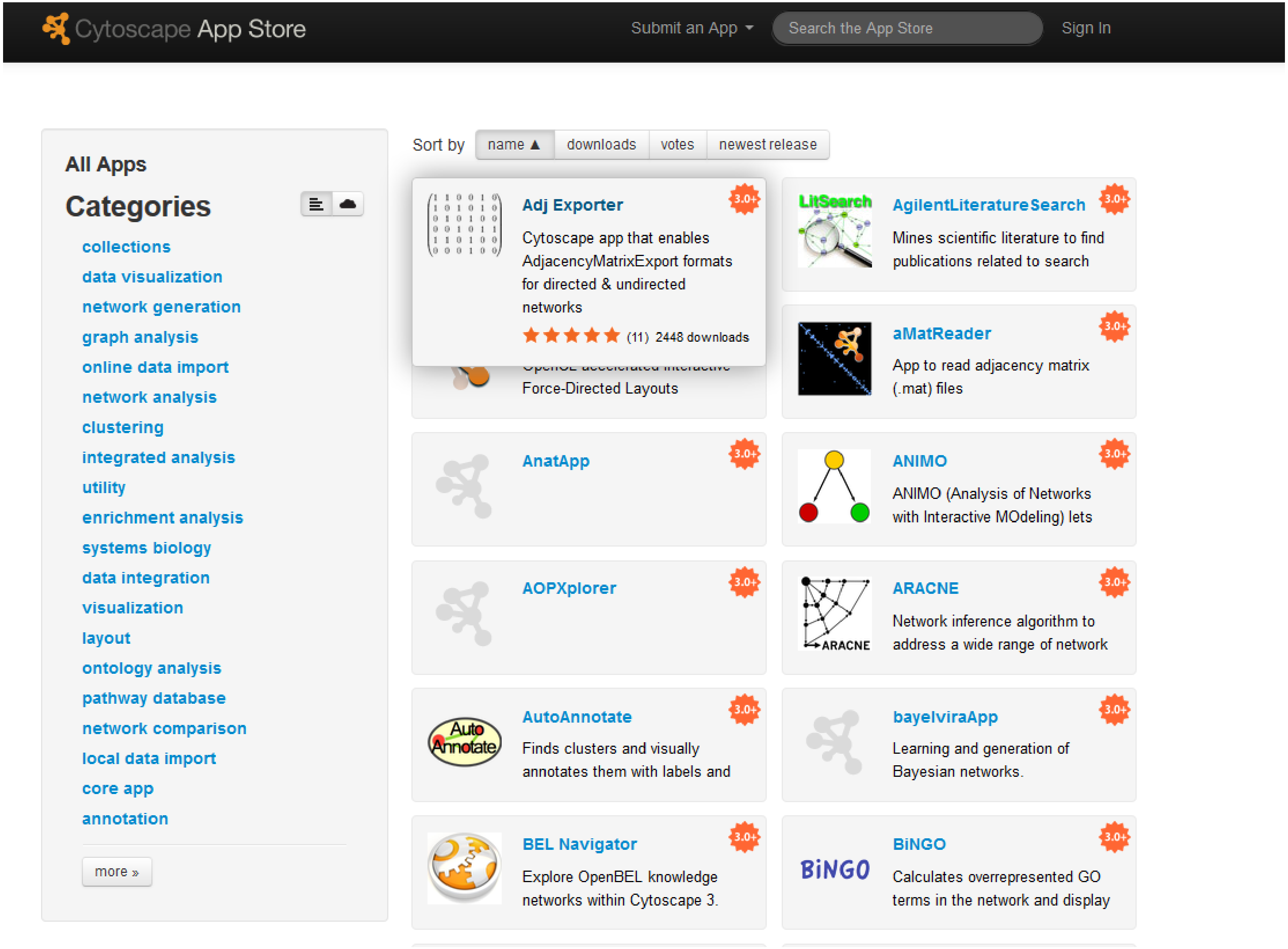
A power law of the form $y = ax^b$ was fitted.

a = 6.826 b = -0.583

Correlation = 0.450

R-squared = 0.257

Note: R-squared is computed on logarithmized values.



Cytoscape App Store

Submit an App ▾

Search the App Store

Sign In

All Apps

Categories

- collections
- data visualization
- network generation
- graph analysis
- online data import
- network analysis
- clustering
- integrated analysis
- utility
- enrichment analysis
- systems biology
- data integration
- visualization
- layout
- ontology analysis
- pathway database
- network comparison
- local data import
- core app
- annotation

more »

Sort by: name ▲ | downloads | votes | newest release

Adj Exporter 3.0+
Cytoscape app that enables AdjacencyMatrixExport formats for directed & undirected networks
★★★★★ (11) 2448 downloads

OpenCL accelerated interactive Force-Directed Layouts

AgilentLiteratureSearch 3.0+
Mines scientific literature to find publications related to search

aMatReader 3.0+
App to read adjacency matrix (.mat) files

AnatApp 3.0+

ANIMO 3.0+
ANIMO (Analysis of Networks with Interactive MOdeling) lets

AOPXplorer 3.0+

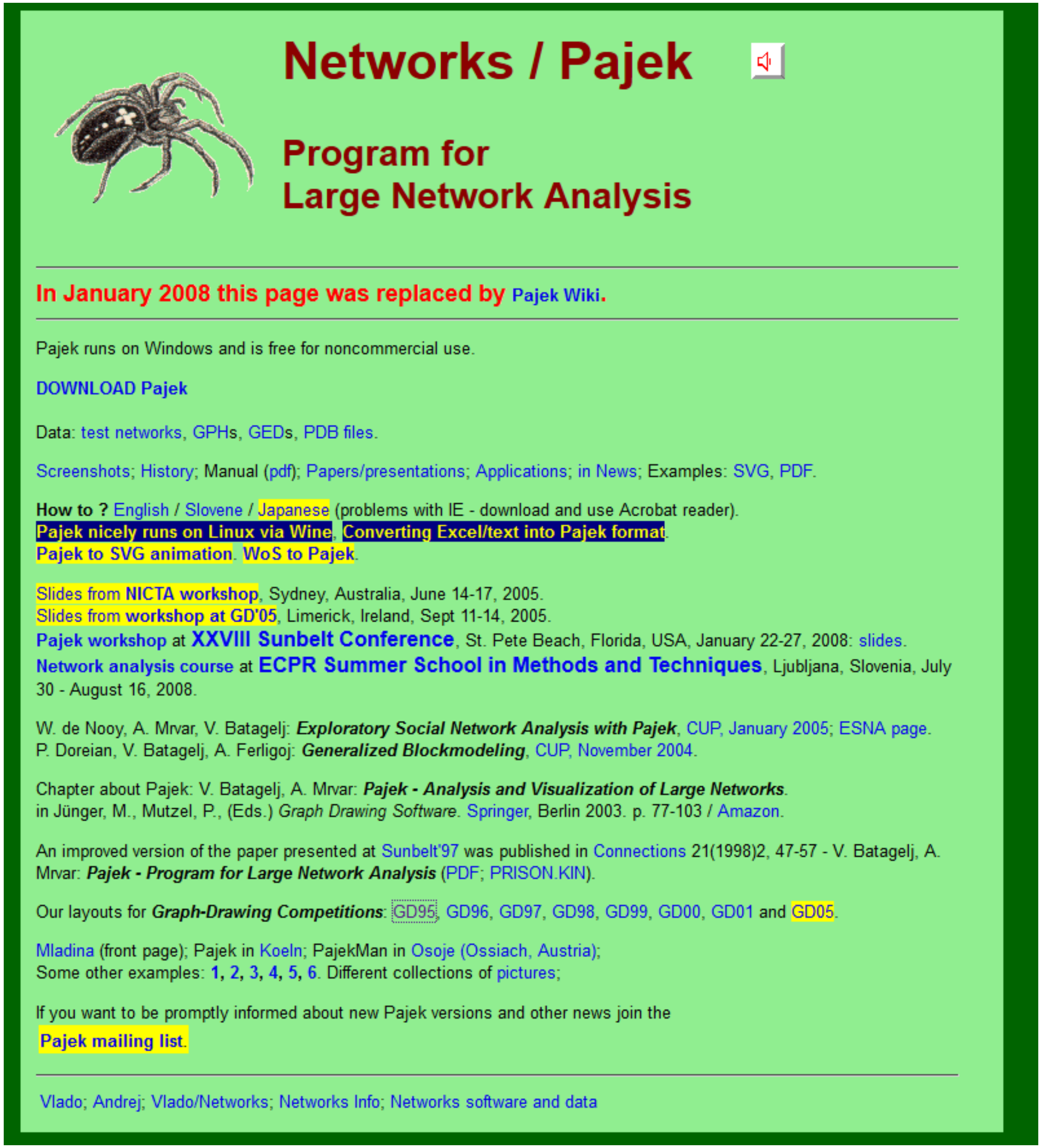
ARACNE 3.0+
Network inference algorithm to address a wide range of network

AutoAnnotate 3.0+
Finds clusters and visually annotates them with labels and

bayelviraApp 3.0+
Learning and generation of Bayesian networks.

BEL Navigator 3.0+
Explore OpenBEL knowledge networks within Cytoscape 3.

BiNGO 3.0+
Calculates overrepresented GO terms in the network and display



Networks / Pajek

Program for
Large Network Analysis

In January 2008 this page was replaced by Pajek Wiki.

Pajek runs on Windows and is free for noncommercial use.

DOWNLOAD Pajek

Data: test networks, GPHs, GEDs, PDB files.

Screenshots; History; Manual (pdf); Papers/presentations; Applications; in News; Examples: SVG, PDF.

How to ? English / Slovene / Japanese (problems with IE - download and use Acrobat reader).
Pajek nicely runs on Linux via Wine, **Converting Excel/text into Pajek format**.
Pajek to SVG animation. **WoS to Pajek**.

Slides from **NICTA workshop**, Sydney, Australia, June 14-17, 2005.
Slides from **workshop at GD'05**, Limerick, Ireland, Sept 11-14, 2005.
Pajek workshop at **XXVIII Sunbelt Conference**, St. Pete Beach, Florida, USA, January 22-27, 2008: slides.
Network analysis course at **ECPR Summer School in Methods and Techniques**, Ljubljana, Slovenia, July 30 - August 16, 2008.

W. de Nooy, A. Mrvar, V. Batagelj: ***Exploratory Social Network Analysis with Pajek***, CUP, January 2005; ESNA page.
P. Doreian, V. Batagelj, A. Ferligoj: ***Generalized Blockmodeling***, CUP, November 2004.

Chapter about Pajek: V. Batagelj, A. Mrvar: ***Pajek - Analysis and Visualization of Large Networks***.
in Jünger, M., Mutzel, P., (Eds.) *Graph Drawing Software*. Springer, Berlin 2003. p. 77-103 / Amazon.

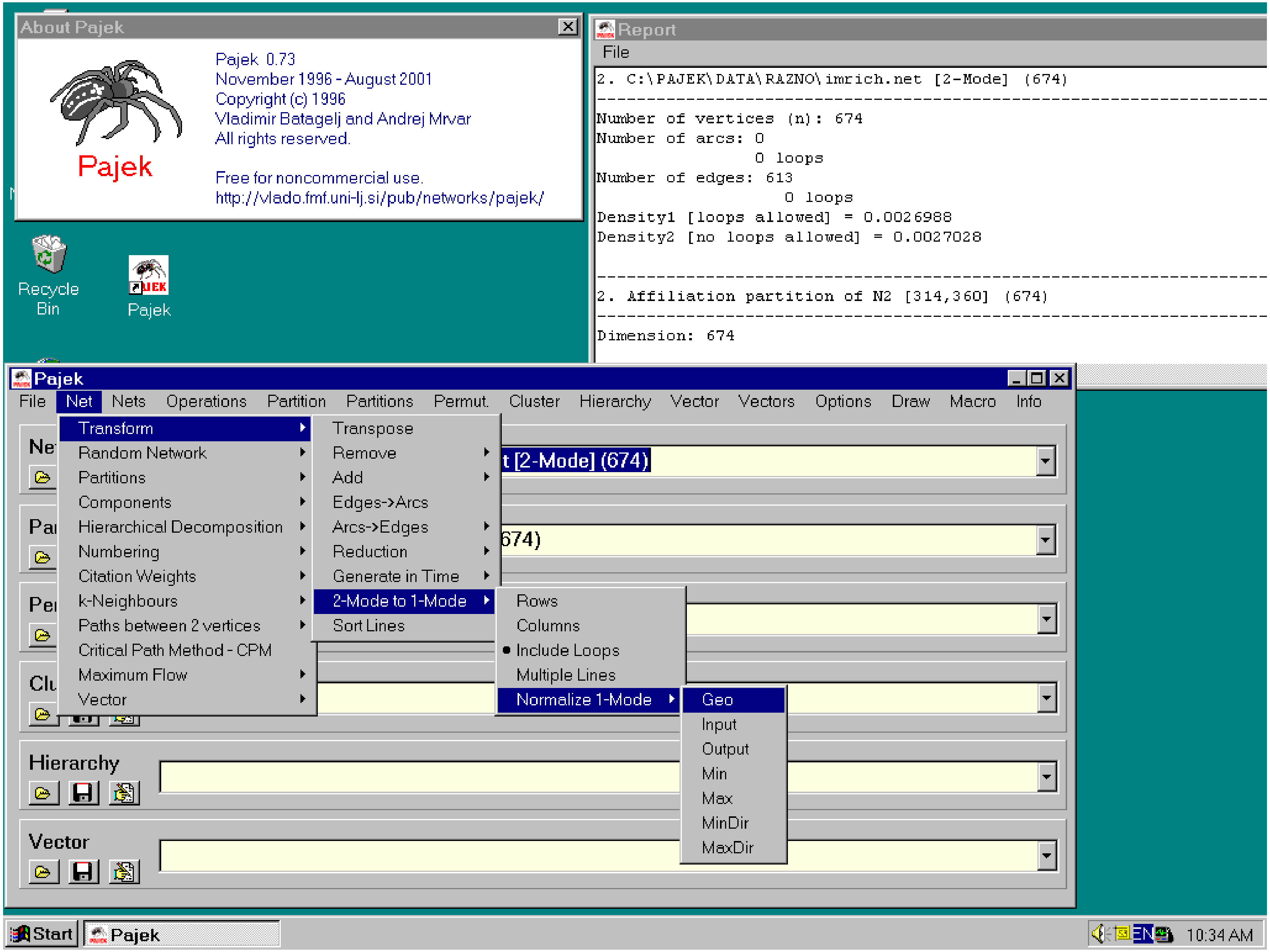
An improved version of the paper presented at Sunbelt'97 was published in Connections 21(1998)2, 47-57 - V. Batagelj, A. Mrvar: ***Pajek - Program for Large Network Analysis*** (PDF; PRISON.KIN).

Our layouts for ***Graph-Drawing Competitions***: GD95, GD96, GD97, GD98, GD99, GD00, GD01 and GD05.

Mladina (front page); Pajek in Koeln; PajekMan in Osoje (Ossiach, Austria);
Some other examples: **1**, **2**, **3**, **4**, **5**, **6**. Different collections of pictures;

If you want to be promptly informed about new Pajek versions and other news join the
Pajek mailing list.

Vlado; Andrej; Vlado/Networks; Networks Info; Networks software and data



About Pajek

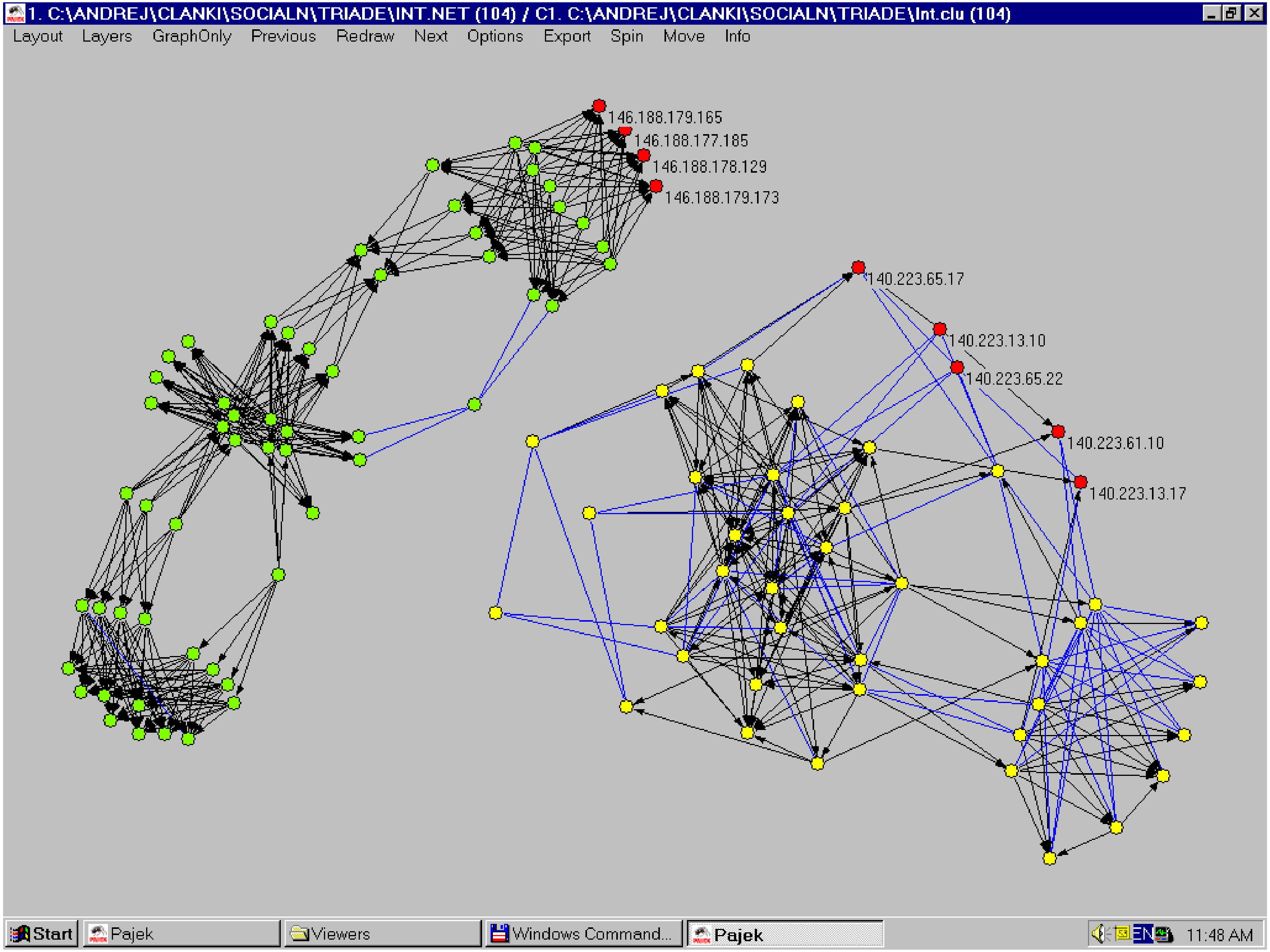
Pajek 0.73
November 1996 - August 2001
Copyright (c) 1996
Vladimir Batagelj and Andrej Mrvar
All rights reserved.

Free for noncommercial use.
http://vlado.fmf.uni-lj.si/pub/networks/pajek/

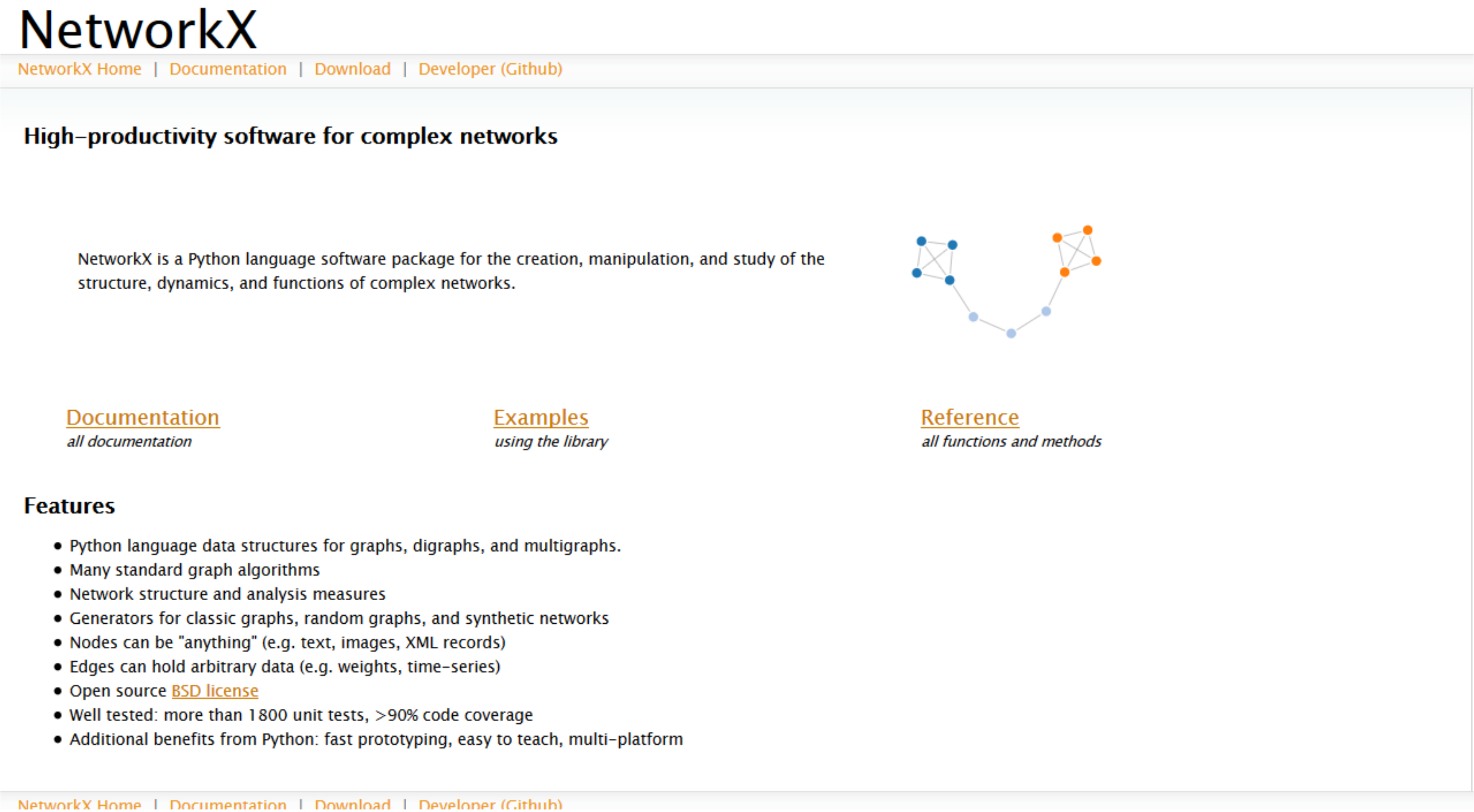
Report

```
2. C:\PAJEK\DATA\RAZNO\imrich.net [2-Mode] (674)
--------------------------------------------------------------------
Number of vertices (n): 674
Number of arcs: 0
                0 loops
Number of edges: 613
                   0 loops
Density1 [loops allowed] = 0.0026988
Density2 [no loops allowed] = 0.0027028

--------------------------------------------------------------------
2. Affiliation partition of N2 [314,360] (674)
--------------------------------------------------------------------
Dimension: 674
```


1. C:\ANDREJ\CLANKI\SOCIALN\TRIADE\INT.NET (104) / C1. C:\ANDREJ\CLANKI\SOCIALN\TRIADE\Int.clu (104)
Layout
Layers
GraphOnly
Previous
Redraw
Next
Options
Export
Spin
Move
Info
146.188.179.165
146.188.177.185
146.188.178.129
146.188.179.173
140.223.65.17
140.223.13.10
140.223.65.22
140.223.61.10
140.223.13.17
Start
Pajek
Viewers
Windows Command...
Pajek
EN
11:48 AM

- JUNG (Java Universal Network/Graph) (O'Madadhain, Fisher, White, & Boey, 2003) is a free, open-source software for the manipulation, analysis, and visualization of network data. JUNG can handle various types of networks, including bipartite and multipartite graphs, multigraphs, and hypergraphs directed and undirected graphs. The tool offers the ability to annotate graphs, entities, and relations with metadata. Additionally, contains Implementations of a number of algorithms from graph theory, social network analysis and machine learning. These include routines for clustering, random graph generation, statistical analysis, decomposition, optimization, and calculation of network distances and ranking measures (centrality etc). Finally, JUNG provides also visualization tools for the interactive exploration of network data. Users can choose among the provided layout and rendering algorithms, or use the software to create their own custom algorithms



NetworkX

NetworkX Home | Documentation | Download | Developer (Github)

High-productivity software for complex networks

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Documentation
all documentation

Examples
using the library

Reference
all functions and methods

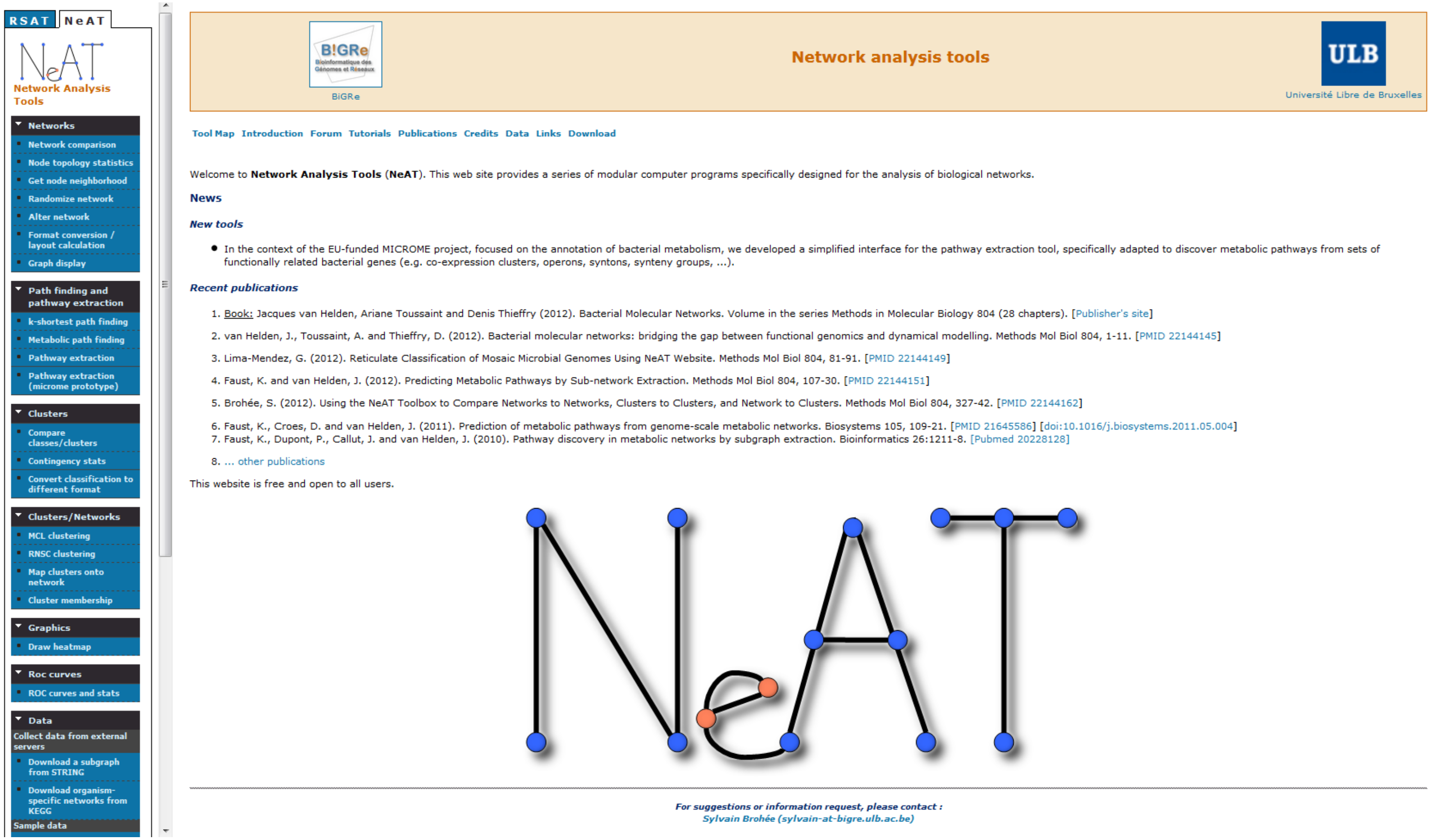
Features

- Python language data structures for graphs, digraphs, and multigraphs.
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks
- Nodes can be "anything" (e.g. text, images, XML records)
- Edges can hold arbitrary data (e.g. weights, time-series)
- Open source BSD license
- Well tested: more than 1800 unit tests, >90% code coverage
- Additional benefits from Python: fast prototyping, easy to teach, multi-platform

NetworkX Home | Documentation | Download | Developer (Github)

NeAT

- http://rsat.bigre.ulb.ac.be/rsat/index_neat.html



Welcome to **Network Analysis Tools** (**NeAT**). This web site provides a series of modular computer programs specifically designed for the analysis of biological networks.

News

New tools

- In the context of the EU-funded MICROME project, focused on the annotation of bacterial metabolism, we developed a simplified interface for the pathway extraction tool, specifically adapted to discover metabolic pathways from sets of functionally related bacterial genes (e.g. co-expression clusters, operons, syntons, synteny groups, ...).

Recent publications

1. Book: Jacques van Helden, Ariane Toussaint and Denis Thieffry (2012). Bacterial Molecular Networks. Volume in the series Methods in Molecular Biology 804 (28 chapters). [Publisher's site]
2. van Helden, J., Toussaint, A. and Thieffry, D. (2012). Bacterial molecular networks: bridging the gap between functional genomics and dynamical modelling. Methods Mol Biol 804, 1-11. [PMID 22144145]
3. Lima-Mendez, G. (2012). Reticulate Classification of Mosaic Microbial Genomes Using NeAT Website. Methods Mol Biol 804, 81-91. [PMID 22144149]
4. Faust, K. and van Helden, J. (2012). Predicting Metabolic Pathways by Sub-network Extraction. Methods Mol Biol 804, 107-30. [PMID 22144151]
5. Brohée, S. (2012). Using the NeAT Toolbox to Compare Networks to Networks, Clusters to Clusters, and Network to Clusters. Methods Mol Biol 804, 327-42. [PMID 22144162]
6. Faust, K., Croes, D. and van Helden, J. (2011). Prediction of metabolic pathways from genome-scale metabolic networks. Biosystems 105, 109-21. [PMID 21645586] [doi:10.1016/j.biosystems.2011.05.004]
7. Faust, K., Dupont, P., Callut, J. and van Helden, J. (2010). Pathway discovery in metabolic networks by subgraph extraction. Bioinformatics 26:1211-8. [Pubmed 20228128]
8. ... other publications

This website is free and open to all users.

For suggestions or information request, please contact :
Sylvain Brohée (sylvain-at-bigre.ulb.ac.be)

graph-tool

- Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (a.k.a. networks). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in C++, making extensive use of template metaprogramming, based heavily on the Boost Graph Library. This confers it a level of performance that is comparable (both in memory usage and computation time) to that of a pure C/C++ library.
- http://graph-tool.skewed.de/

Network Workbench

- Network Workbench: A Large-Scale Network Analysis, Modeling and Visualization Toolkit for Biomedical, Social Science and Physics Research.This project will design, evaluate, and operate a unique distributed, shared resources environment for large-scale network analysis, modeling, and visualization, named Network Workbench (NWB).
- http://nwb.cns.iu.edu/doc.html

igraph

- igraph is a collection of network analysis tools with the emphasis on **efficiency**, **portability** and ease of use. igraph is **open source** and free. igraph can be programmed in **R**, **Python** and **C/C++**.
- http://igraph.org/redirect.html

BioPerl (**Network package)**

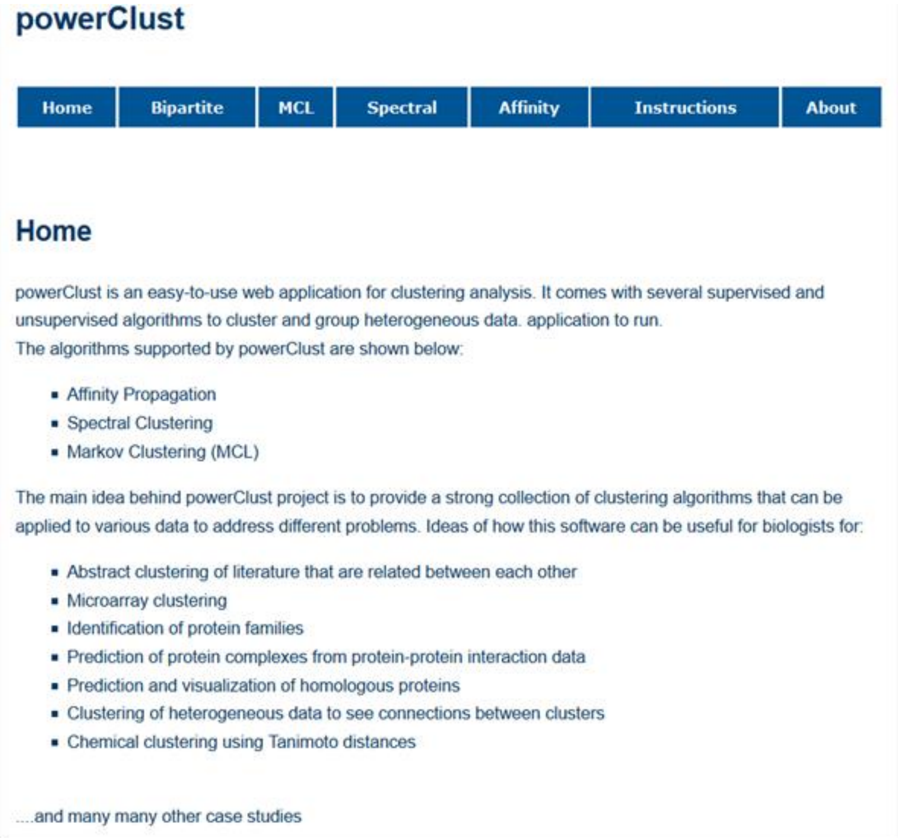
- The *bioperl-network* or Bio::Network package parses and analyzes protein-protein interaction data provided by databases such as DIP, BIND, IntAct, HPRD, and MINT . It replaces the Bio::Graph* modules written by Nat Goodman and Richard Adams.
- This package is based on Perl's Graph module and uses it to supply an underlying suite of graph algorithms. In theory any Graph method can be used to query or analyze a Bio::Network::ProteinNet object. The Bio::Network package is currently maintained by Brian Osborne.
- The IO module is used to read a file and create a network. The formats that are currently supported are DIP and PSI MI 2.5. If you are parsing PSI MI read Module:Bio::Graph::IO::psi_xml first for notes on various databases and their PSI MI.
- http://www.bioperl.org/wiki/Network_package

Graph (Perl)

- This module is for creating *abstract data structures* called graphs, and for doing various operations on those. The implementation depends on a Perl feature called "weak references" and Perl 5.6.0 was the first to have those
- http://search.cpan.org/~jhi/Graph-0.9704/lib/Graph.pod

PowerClust

- http://www.compgen.org/tools/powerclust/



Home

powerClust is an easy-to-use web application for clustering analysis. It comes with several supervised and unsupervised algorithms to cluster and group heterogeneous data. application to run.
The algorithms supported by powerClust are shown below:

- Affinity Propagation
- Spectral Clustering
- Markov Clustering (MCL)

The main idea behind powerClust project is to provide a strong collection of clustering algorithms that can be applied to various data to address different problems. Ideas of how this software can be useful for biologists for:

- Abstract clustering of literature that are related between each other
- Microarray clustering
- Identification of protein families
- Prediction of protein complexes from protein-protein interaction data
- Prediction and visualization of homologous proteins
- Clustering of heterogeneous data to see connections between clusters
- Chemical clustering using Tanimoto distances

....and many many other case studies

Ανάπτυξη πλατφόρμας οπτικοποίησης δεδομένων powerClust

Ανάπτυξη πλατφόρμας οπτικοποίησης δεδομένων **powerClust**

http://www.compgen.org/tools/powerclust/

Το εργαλείο δίνει τη δυνατότητα στους χρήστες να το χρησιμοποιήσουν για οπτικοποίηση είτε σε **απλά** είτε σε **διμερή** δίκτυα.

Το powerClust δέχεται ως είσοδο ένα **αρχείο κειμένου** με τις συνδέσεις του δικτύου είτε **με βάρη είτε χωρίς βάρη**.

Το εργαλείο παρέχει οπτικοποίηση σε διάφορες διατάξεις όπως: **διάταξη Fruchterman-Reingold, κυκλική διάταξη, τυχαία διάταξη, διάταξη σε πλέγμα**.

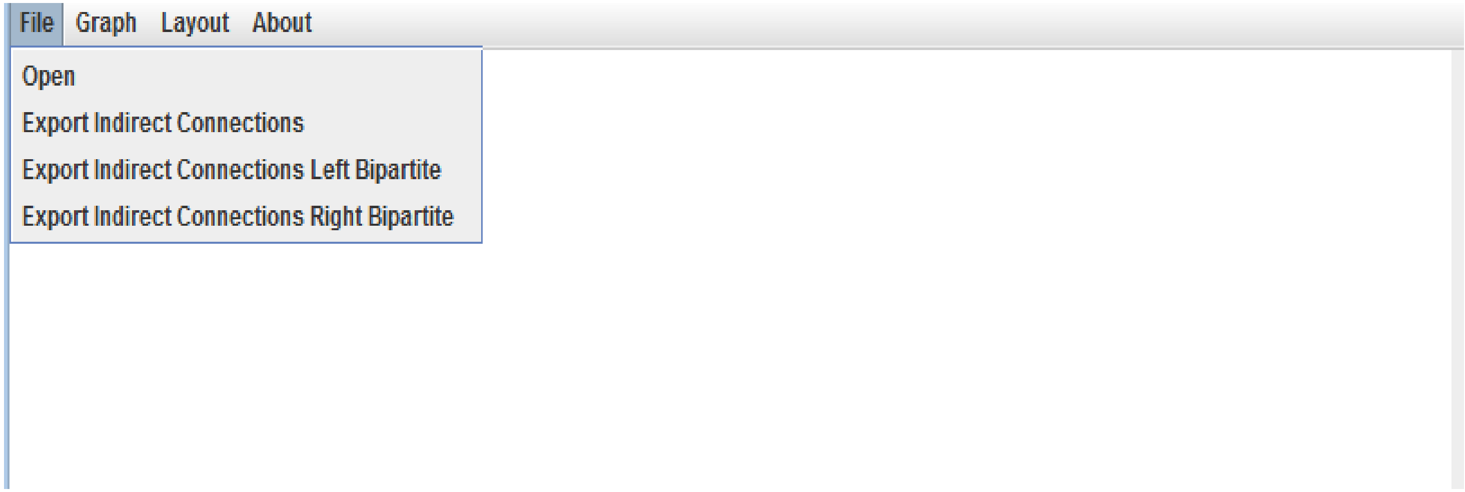
Προσφέρει επίσης επιπλέον επιλογές για να κάνει την οπτικοποίηση πιο κατατοπιστική όπως: **απόκρυψη/εμφάνιση ετικετών, απόκρυψη/εμφάνιση κόμβων, απόκρυψη/εμφάνιση απευθείας ή/και έμμεσων συνδέσεων**.

Ο χρήστης μπορεί να εισάγει στο εργαλείο ένα **διμερές** γράφο και να εξάγει **δυο απλούς** γράφους.

Το **αρχείο εξόδου** μπορεί να χρησιμοποιηθεί ως είσοδο σε εργαλεία **τρισδιάστατης** (**3D**) απεικόνισης δικτύων.

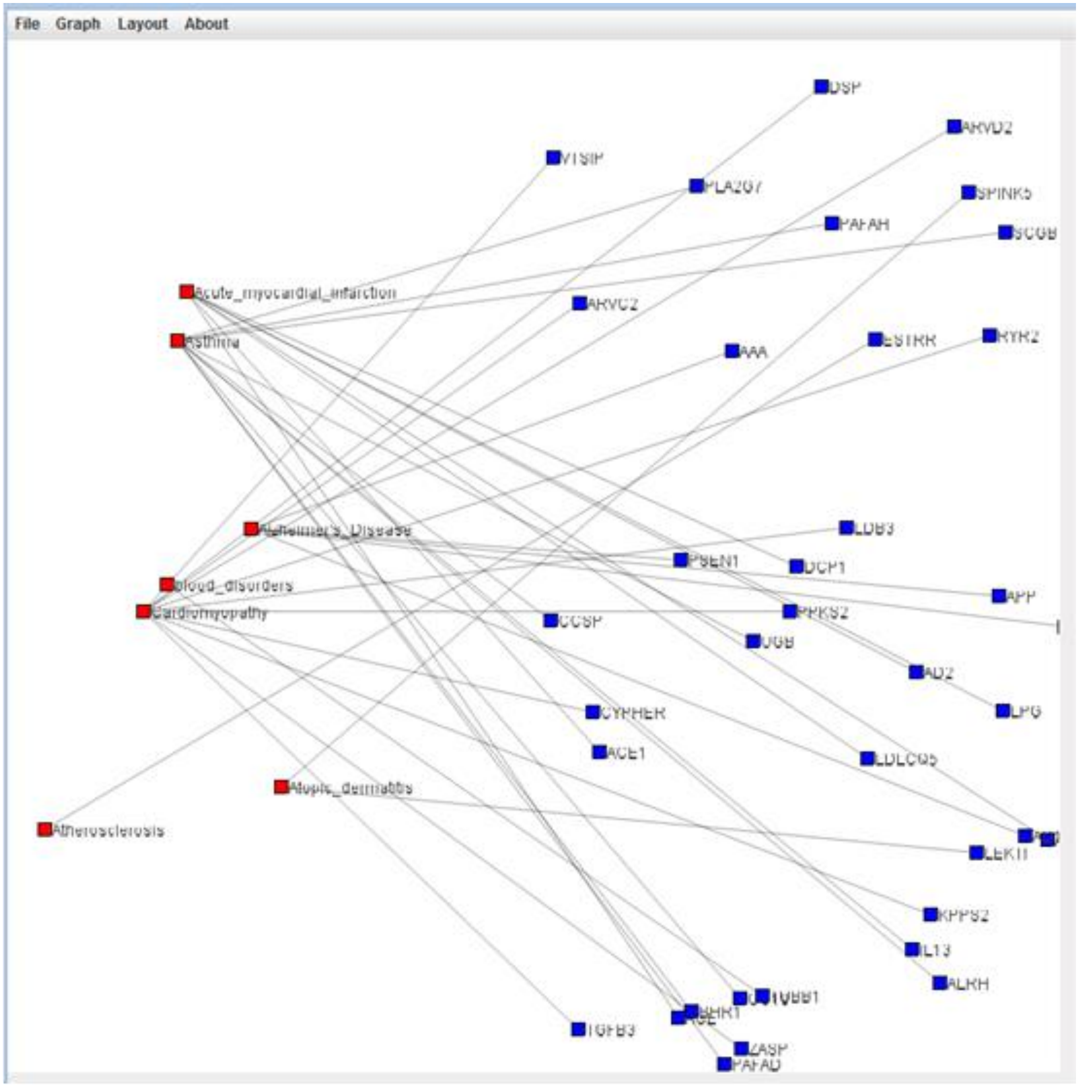
powerClust

Αρχική σελίδα προγράμματος οπτικοποίησης βιολογικών δικτύων


File
Graph
Layout
About
Open
Export Indirect Connections
Export Indirect Connections Left Bipartite
Export Indirect Connections Right Bipartite

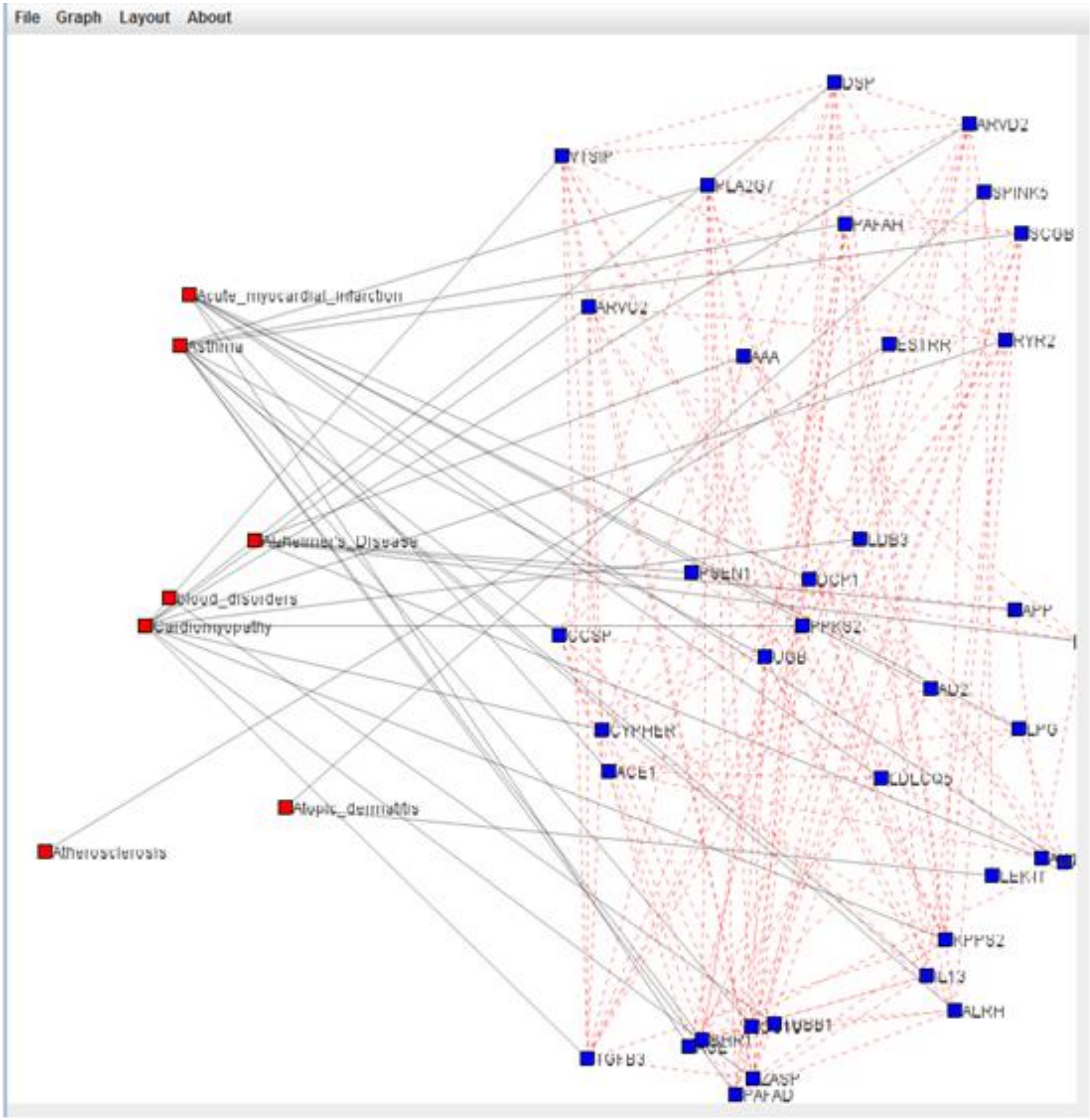
powerClust

Παράδειγμα οπτικοποίησης ενός διμερούς βιολογικού δικτύου (γονίδιο-ασθένεια) σε τυχαία διάταξη (**Random**)



powerClust

Παράδειγμα οπτικοποίησης ενός διμερούς βιολογικού δικτύου σε τυχαία διάταξη (Random) και επιπλέον εμφάνιση των **έμμεσων συνδέσεων** (κόκκινη διακεκομμένη γραμμή)



Clustering

Ανάπτυξη πλατφόρμας ομαδοποίησης δεδομένων powerClust

http://www.compgen.org/tools/powerclust/

Περιλαμβάνει 3 αλγόριθμους ομαδοποίησης:

- **MCL (Markov Clustering), Affinity Propagation:** μη επιβλεπόμενοι αλγόριθμοι ομαδοποίησης δηλαδή ο αλγόριθμος υπολογίζει των αριθμό των ομάδων στις οποίες ταξινομηθεί τα δεδομένα.
- **Spectral Clustering:** επιβλεπόμενος αλγόριθμος ομαδοποίησης δηλαδή ο χρήστης θα πρέπει να ορίσει των αριθμό των ομάδων στις οποίες επιθυμεί να ταξινομηθούν τα δεδομένα του.

Ως είσοδο στους αλγόριθμους δίνετε ένα σετ δεδομένων το οποίο περιλαμβάνει ένα δίκτυο με ή χωρίς βάρη στις συνδέσεις του.

Ο χρήστης εισάγει τα δεδομένα στον αλγόριθμο που θέλει να χρησιμοποιήσει και αυτός του επιστρέφει τα ομαδοποιημένα δεδομένα του.

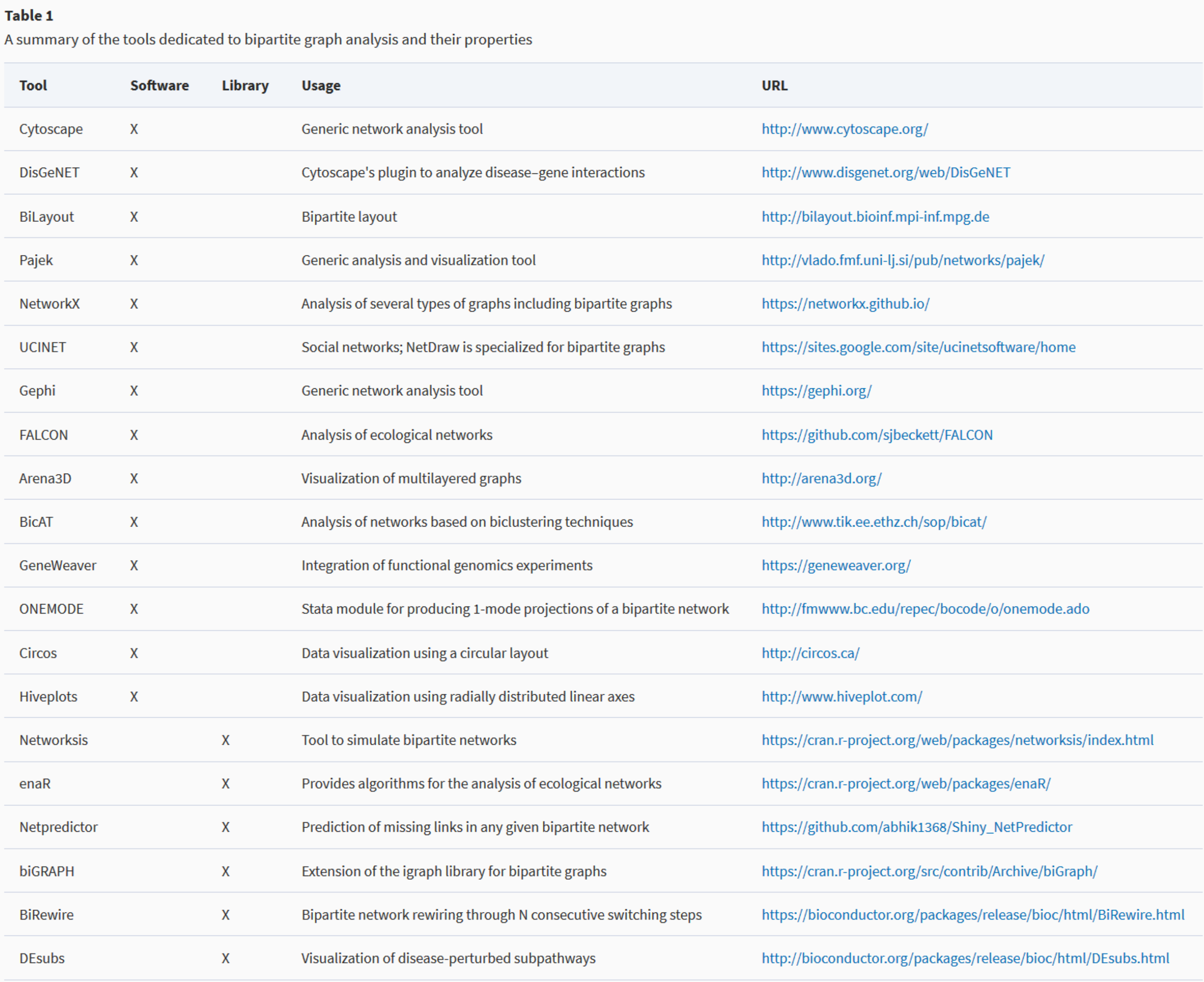


Table 1

A summary of the tools dedicated to bipartite graph analysis and their properties

Tool	Software	Library	Usage	URL
Cytoscape	X		Generic network analysis tool	http://www.cytoscape.org/
DisGeNET	X		Cytoscape's plugin to analyze disease–gene interactions	http://www.disgenet.org/web/DisGeNET
BiLayout	X		Bipartite layout	http://bilayout.bioinf.mpi-inf.mpg.de
Pajek	X		Generic analysis and visualization tool	http://vlado.fmf.uni-lj.si/pub/networks/pajek/
NetworkX	X		Analysis of several types of graphs including bipartite graphs	https://networkx.github.io/
UCINET	X		Social networks; NetDraw is specialized for bipartite graphs	https://sites.google.com/site/ucinetsoftware/home
Gephi	X		Generic network analysis tool	https://gephi.org/
FALCON	X		Analysis of ecological networks	https://github.com/sjbeckett/FALCON
Arena3D	X		Visualization of multilayered graphs	http://arena3d.org/
BicAT	X		Analysis of networks based on biclustering techniques	http://www.tik.ee.ethz.ch/sop/bicat/
GeneWeaver	X		Integration of functional genomics experiments	https://geneweaver.org/
ONEMODE	X		Stata module for producing 1-mode projections of a bipartite network	http://fmwww.bc.edu/repec/bocode/o/onemode.ado
Circos	X		Data visualization using a circular layout	http://circos.ca/
Hiveplots	X		Data visualization using radially distributed linear axes	http://www.hiveplot.com/
Networksis		X	Tool to simulate bipartite networks	https://cran.r-project.org/web/packages/networksis/index.html
enaR		X	Provides algorithms for the analysis of ecological networks	https://cran.r-project.org/web/packages/enaR/
Netpredictor		X	Prediction of missing links in any given bipartite network	https://github.com/abhik1368/Shiny_NetPredictor
biGRAPH		X	Extension of the igraph library for bipartite graphs	https://cran.r-project.org/src/contrib/Archive/biGraph/
BiRewire		X	Bipartite network rewiring through N consecutive switching steps	https://bioconductor.org/packages/release/bioc/html/BiRewire.html
DEsubs		X	Visualization of disease-perturbed subpathways	http://bioconductor.org/packages/release/bioc/html/DEsubs.html

Βιβλιογραφία

- A.-L. Barabási, Z. N. Oltvai **Network biology: understanding the cell's functional organization** *Nature Reviews Genetics* 5, 101-113 (2004)
- Steven H. Strogatz **Exploring complex networks** *Nature* 410, 268-276(8 March 2001)
- Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, Aerts J, Schneider R, Bagos PG. **Using graph theory to analyze biological networks.** *BioData Mining*. 2011, **4**: 10 https://biodatamining.biomedcentral.com/articles/10.1186/1756-0381-4-10
- Georgios A Pavlopoulos, Panagiota I Kontou, Athanasia Pavlopoulou, Costas Bouyioukos, Evripides Markou, Pantelis G Bagos; **Bipartite graphs in systems biology and medicine: a survey of methods and applications**, *GigaScience*, Volume 7, Issue 4, 1 April 2018, giy014, https://doi.org/10.1093/gigascience/giy014
- Νικολάου, Χ., Χουβαρδάς, Π. 2015. **Βιολογικά Δίκτυα**. [Κεφάλαιο Συγγράμματος]. Στο Νικολάου, Χ., Χουβαρδάς, Π. 2015. *Υπολογιστική βιολογία*. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. κεφ 10. Διαθέσιμο στο: http://hdl.handle.net/11419/1587