Graphs & Biological Networks

Maribel Hernández Rosales

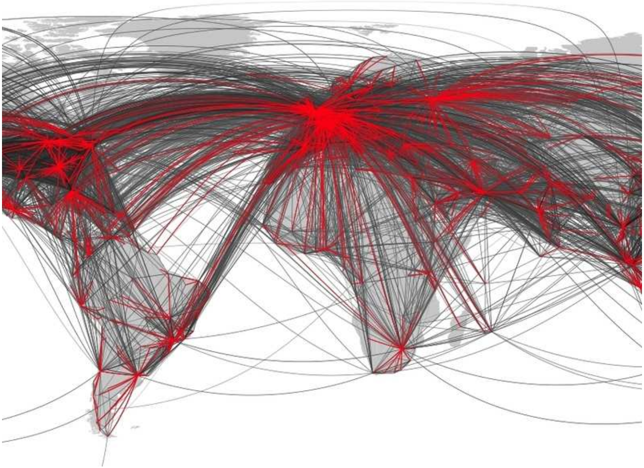
Institute of Mathematics, UNAM, Queretaro, Mexico

23. February 2016, EVOP, Leipzig, Germany

Outline

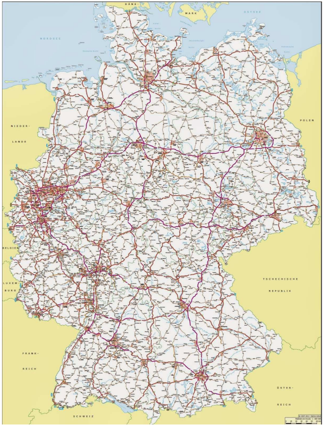
1. Introduction
2. Graph Theory
   - Non-directed Networks
   - Directed Graphs
   - Random Networks
3. Network Motifs

Networks everywhere!



International flight network.

Networks everywhere!



German road map.

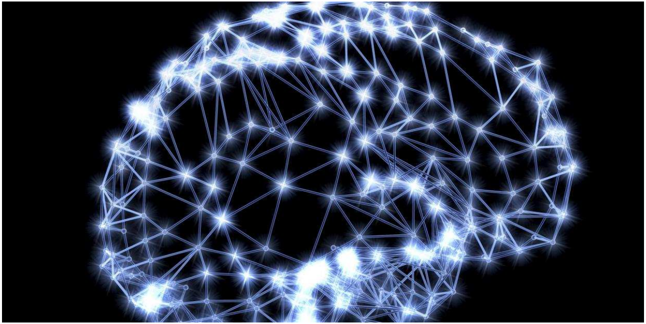
Networks everywhere!

German bike road map.

Networks everywhere!

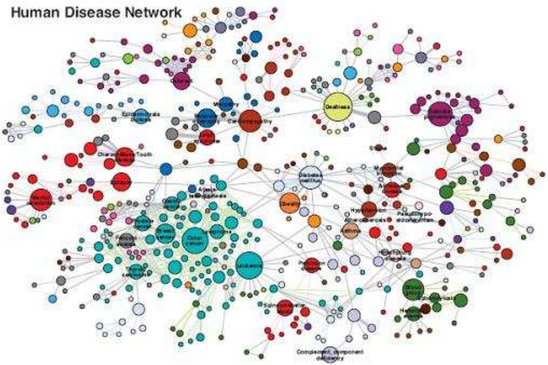
Facebook.

Networks everywhere!



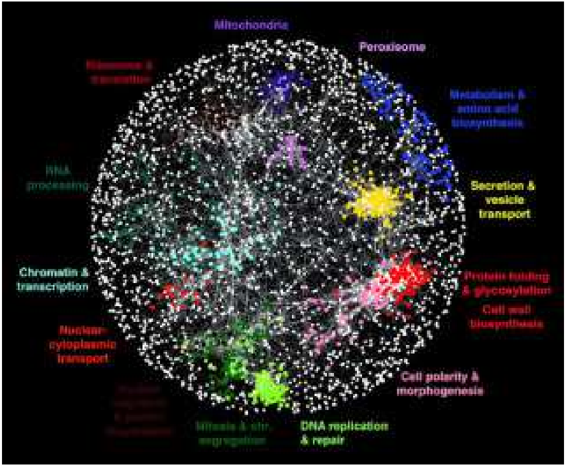
Neural network.

Networks everywhere!



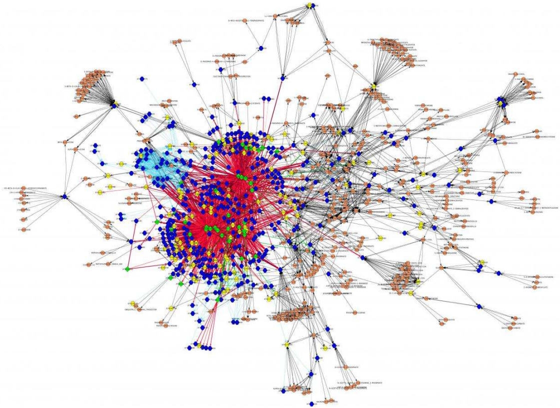
Drugs-proteins interaction network.

Networks everywhere!



Cell landscape.

Networks everywhere!



Regulatory Network.

How can we study networks?

Graph Theory

Network “defintion”?

A group or system of interconnected elements.

- Elements: are they unique? similar? belong to a class?
- Interconnected: physical connections? how are they defined?

Graph Definition

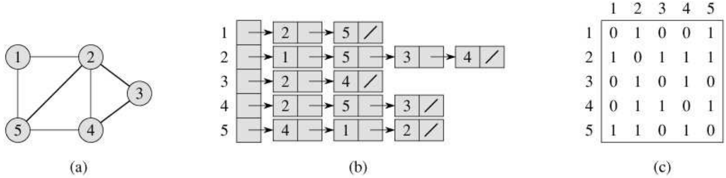
A graph is defined as $G(V, E)$, where V is a set of nodes or vertices, and E is a set of edges. Each edge connects two nodes in V.

Graphs

- undirected and simple graph
- directed or digraph
- multigraph
- multidigraph
- weighted graph

Undirected graphs

Matrix and list of adjacencies

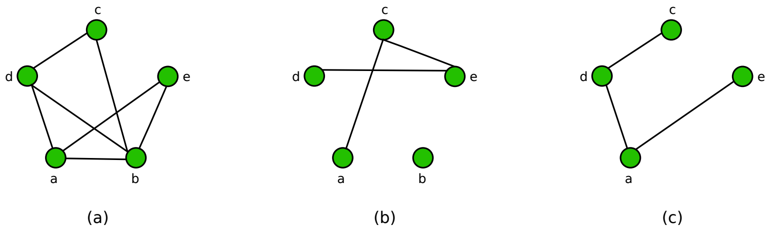


(a) A graph. (b) List of adjacency. (c) Matrix of adjacency.

Order and size of a graph

The **order** of a graph is the number of its nodes.
The **size** of a graph is the number of its edges.

Complement and Induced subgraphs



- (a) a graph
- (b) the complement of the graph
- (c) an induced subgraph

Ex1: Let's create a graph in Python

```
import networkx as nx
G = nx.Graph() # simple graph
# Add some nodes to the graph
G.add_node(1)
G.add_node(2)
G.add_node(3)
# Add edges
G.add_edge(1,2)
# Add weights to the edges
G.add_edge(2,3,weight = 15)
```

Ex2: Display properties of the graph

- Nodes
- Edges
- Neighbors
- Information per edge
- Complement
- An induced subgraph
- Plot the graph and the induced subgraph

Adjacency Matrix and Adjacency list in Python

Ex3: Show the adjacency matrix and the adjacency list of the previous example.

Degree Sequence

Definition: an ordered sequence of the vertex degrees of a graph.

Ex4:

- Create an increased list of the node degrees.
- Sum up the degrees to obtain the average degree of the graph.
- Maximum degree is the the number of nodes-1.

Graph Density

The graph density shows how much its nodes are connected

Ex5:

- $\frac{1}{2}\Sigma deg(v)$ = number of edges in the graph
- $\frac{average\_degree}{max\_degree}$
- $\frac{number\_of\_edges}{max\_number\_of\_edges}$

Degree Frequency

Ex.6.1: plot the degree distribution
X = vertex degrees
Y = number of nodes of degree(X)

Degree Distribution

Ex.6.2: plot the degree frecuency

X = degree

$$Y = \frac{num\_vertices\_of\_degree(X)}{num\_total\_of\_vertices(order)}$$

Distance between two nodes

Number of edges in the shortest path between two nodes.
Ex.7.1:
For each pair of vertices in the graph, calculate their distance.

Diameter of a graph

Maximum distance in the graph.

Ex.7.2:

Calculate the diameter of the graph.

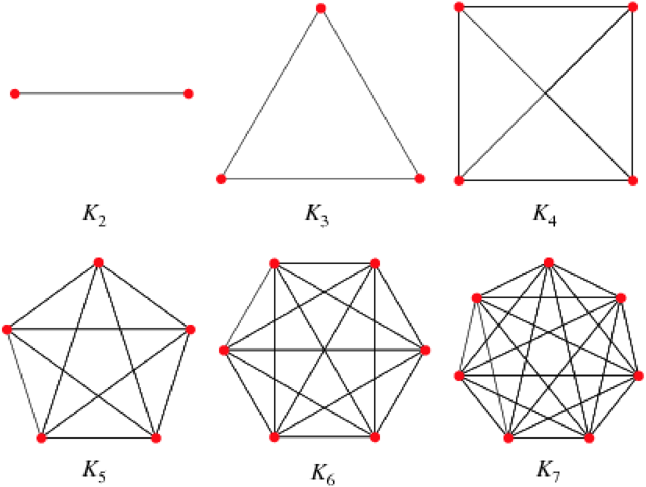
$d-$regular graphs

if the maximum degree is equal to the minimum degree d, then the graph is d-regular.

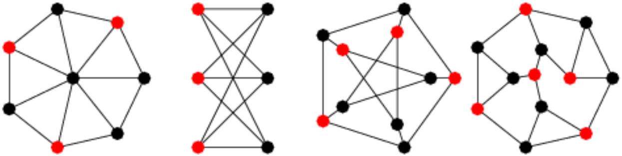
Ex.8:

Check if the graph is $d-$regular.

Complete Graph or Clique (K_n)



Independent set (I_n)

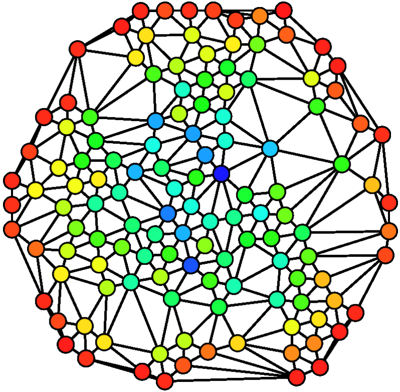


Path and Connected graph

Path: it is a finite (or infinite) sequence of edges which connect a sequence of different vertices.
Connected graph: There is a path between any two vertices.
Ex.9: Obtain the shortest path between each pair of vertices, if there is one.

Degree Centrality & Betweenness

- The degree centrality helps to identify the most important vertices within a graph.
- The betweenness quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.
- **Ex.10**: identify the most central nodes.



Node betweenness (from red = 0 to blue = max)

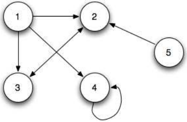
Directed graphs

Properties

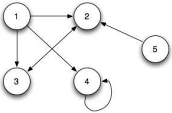
Ex.11:

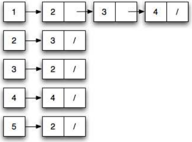
1. Define out-degre and in-degree.
2. Define the number of connections for undirected graphs.
3. Define neighbors.
4. Plot the distribution of the in-degrees and out-degrees.

Adjacency Matrix and Adjacency list



	1	2	3	4	5
1	0	1	1	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	0	0	1	0
5	0	1	0	0	0





Digraphs in Python

Ex.12: Let's create a digraph in Python!

Questions?

- if you have n nodes, how many edges do you need to construct a connected graph?
- if you have n singletons, we would like to create a connected graph, for each edge, you receive 1 euro, how much can you make maximum?
- Given n how many edges are necessary to obtain a complete graph.

Random Networks

$G(n, m)$, n nodes and m connections. $G(n, p)$, n nodes and probability p of a conection between each pair of nodes. Erdos y Renyi 1959.

Let's practice!

Ex.13: For a random network of n nodes:

1. Calculate the distance between each pair of nodes
2. calculate the diameter, for $n = 10, 100, 200, 300, 500, 1000$
   $erdos\_renyi\_graph(n, p)$
3. Check the degree distribution
4. Delete $r = 1, 2, 3, .., 10$ and repeat the calculations.

Complex Networks vs. Random Networks

- The nodes in random networks have similar degree, there are no nodes with high degree.
- Most of the nodes have few connections, and there are few nodes with many connections.

Complex Networks vs. Random Networks

Let's check it!

Compare a complex network with a random network

Ex. 14 Let's compare a random network with the $E.coli$ regulatory network.

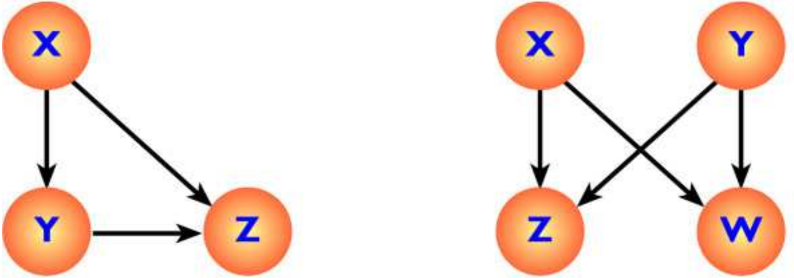
1. Read the $E.coli$ regulatory network
2. Calculate the number of nodes
3. Generate a random network with the same number of nodes
4. Calculate distances, average distance, and diameter, centrality and betweness
5. Obtain the distribution of the degrees of the nodes

Network Motifs

Network motifs

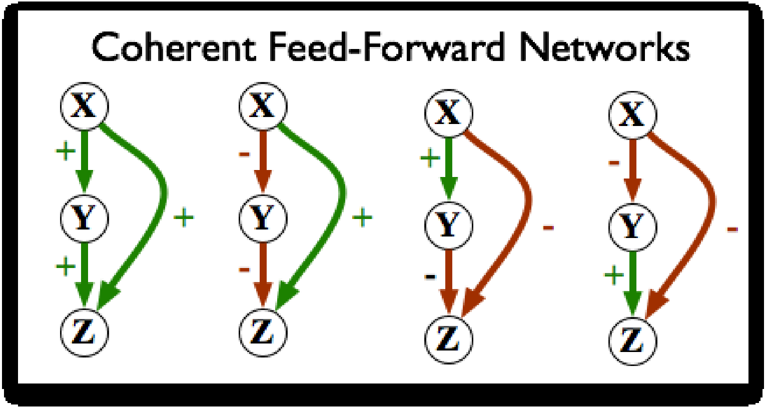
Network motifs are recurrent and statistically significant subgraphs (patterns). They repeat themselves in complex networks.

Feed Forward Loop (FFL) & Bi-fan

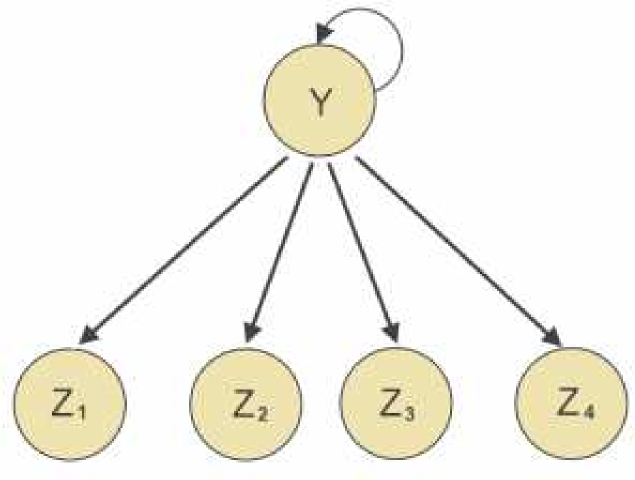


a b

Coherent FFL



Single Input Model (SIM)



Compare again!

Ex.15:

- Test MFinder
- Find the FFL, Bi-fan and SIM for the E.coli regulatory network.
- Do the same for the random network.
- Compare the distribution and frequency of the motifs in both networks.
- Are the autoregulatory genes in SIM central nodes?
- What is the difference in betweenness of the regulated genes in the FFL and Bi-fan and the regulator ones?

Acknowledgements

- The organizors: Katja Nowick, Rui Faria, TAs
- Collaborators: Marcus Lechner, Peter Stadler, Roberto Alvarez
- Students: Marcos Gonzalez, Andres Garcia, Edgar Chavez, Alitzel Lopez, Nancy Gonzalez

Thank you all for your attention!