

---

# EXTENDED BIOLOGICAL PROCESSES: THE ROLE OF NETWORKS

---

Modelization and simulation of biosystems



MASTER'S DEGREE IN COMPUTATIONAL BIOLOGY  
Sandra Alonso Paz

## Exercise 6.1 – Centrality and dynamical processes in networks

Carry out a numerical simulation of the following discrete-time spreading processes in a random Erdős-Rényi network of moderate size (around 100 nodes). Hint: make sure that all the nodes in the network form a single connected component.

### 1. Random walk:

Start in a random node and, at each time step, randomly move to one of its neighbors. Record the position of the walker after  $t$  timesteps, with  $t \gg d$  (with  $d$  the diameter of the network). Repeat the simulation a very large number of times and estimate the probability that the random walker is found in each particular node at the end of the process. Think of biological scenarios where this kind of dynamics could be relevant.

**Random walk** is a mathematical object, known as a stochastic or random process, that describes a path that consists of a succession of random steps on some mathematical space such as the integers. An elementary example of a random walk is the random walk on the integer number line, which starts at 0 and at each step moves +1 or -1 with equal probability.

For this task I have implemented an algorithm which iterates for 1000 times a random walker algorithm. Firstly, it will ask you for the number of nodes ( $N$ ) you want to take into account. Then inside the 100 steps loop, it will generate a random number between 1 and  $N$  (start node). Afterwards, it will decide to go right or left (up or down) with 50% of probability. Each node will be saved in an array and the whole process will be showed by a plot.

It will also store the last node of the simulation, in order to calculate the probability that the random walker is found in each particular node at the end of the process.

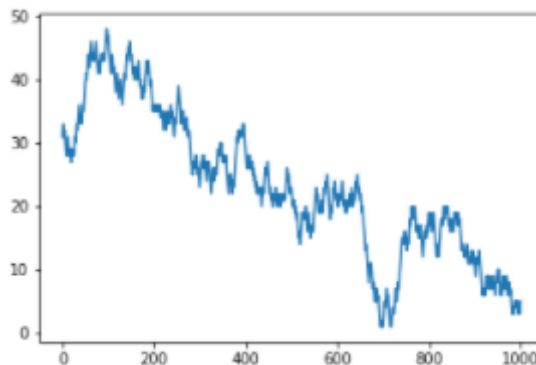


Figure 1: example of the output plot for initial node = 31 and last node = 5.

The result obtained by executing the algorithm for  $N = 100$  is the following one:

- During 1000 iterations, there are 100 nodes in which the random walker has finished.
- The most repeated one are node 3 and node 76 which were the last node on 19 iterations.
- Therefore, the probability to be the last node or the final position for these nodes is  $19/1000$  which means 1.9% each one. Otherwise, there are nodes which are not selected as final position any time and therefore their probability is 0%.
- In the light of this results and keeping in mind that random walker is a stochastic method, every node should have the same probability to be the last selected during the iteration. For instance, in this case is  $1/100$  which is 1%. Similar to the result obtained.

Random walkers have applications to many scientific fields including ecology, psychology, computer science, physics, chemistry, biology as well as economics. Some biological examples could be the search path of a foraging animal or the path traced by a molecule as it travels in a liquid or a gas.

## 2. Spreading of an infection across towns:

**Assume that each node represents a town and edges represent mobility patterns across towns. Start with a single town infected at very low prevalence. At each time step, the prevalence in each town increases by 1% (i.e., it is multiplied by 1.01). Moreover, the infection spreads to all neighboring towns at a rate that is proportional to the prevalence in the town of origin multiplied by a mobility factor  $m$  (set  $m = 1$  for simplicity). Simulate the process until the infection has reached appreciable prevalence in all towns (e.g.,  $10^4$  times the prevalence of the initial state). Compare the relative prevalence across towns.**

For this task I have developed an algorithm which simulates a country with several near cities' population exchanges. Firstly, the program will ask you for the number of near cities and the initial prevalence of the infection. Then a loop, whose end condition is reaching the initial prevalence multiply by 10000, will start. Inside of it, every current infected city will try to infect every other. I have set this "intention" proportional to its prevalence by a condition "random number < current prevalence". If this condition is reached, mobility and therefore infection will be produced (it is represented by an edge in the graph). During the simulation, in each step, prevalence will increase in 1%.

Finally, I showed a set with the id of each infected city and the mean of the current prevalence of all of them.

The result obtained by executing the algorithm for  $N = 50$  cities and initial prevalence = 0.01 (city 1) is the following one:

- Maximum prevalence to reach is 100 ( $0.01 \cdot 10000$ ).
- City with the highest prevalence is city 1 (127.42). This is reasonable because it was the first infected and so had more time to increase its prevalence.
- Mean prevalence at the end of the simulation is 127.12. This result shows that plenty of the cities were infected at the first days.
- Less prevalence is found in city 25 which was the last infected. Its final prevalence is 126.321, very close to the mean value. Therefore, as I have said, I can affirm that most of cities were infected at the first's days.

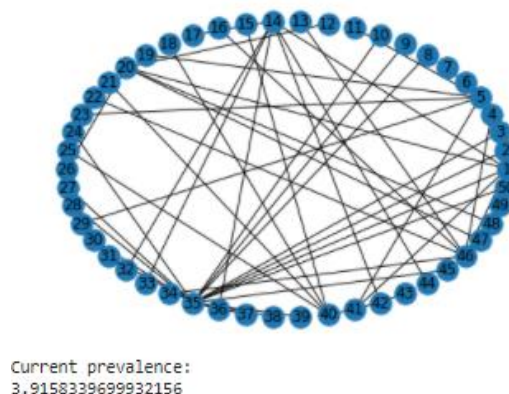


Figure 2: example of the mobility graph and the current maximum prevalence in a middle step.

**Which measure of network centrality (degree or eigenvector centrality) correlates the best with the distribution of probability (in the RW) or prevalence (in the disease spreading) across nodes at the end of the simulation? Are there differences between both scenarios?**

Prevalence distribution which is seen in disease spreading, correlates with the eigen vector centrality. Eigen vector centrality of a node  $k$  in a network is defined as the  $k$ -th component of the eigenvector of its adjacency matrix  $A$  corresponding to the largest eigenvalue  $\lambda$ . In our case, the epicenter of the disease will stay in the same place with the highest prevalence and the prevalence of surrounding nodes will be proportional to the distance to the epicenter.

On the other hand, distribution of probability shown in the random walk algorithm correlates with a degree centrality measure because it depends on the number of links and in this case the chosen path by the random walker. In our case, random walker can move around all the nodes and can decide to stay briefly or larger in a place.

The main difference in both scenarios is that disease spreading tends to stay in a place and spread to the nearest nodes. With some methods can be set in a limited place. Otherwise, random walker cannot be contented due its stochasticity, so it will tend to spread all around the net.

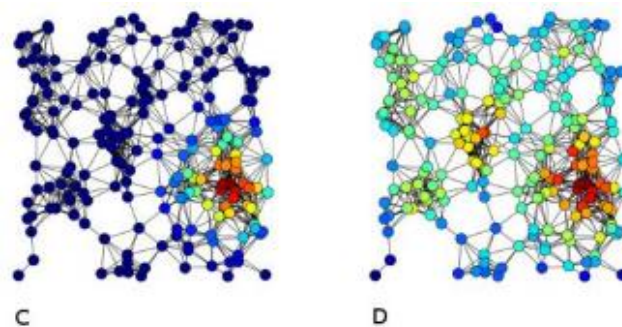


Figure 3: C: eigenvector centrality. D: degree centrality.

**Try reducing the mobility by a factor 10 and 100 ( $m = 0.1$ ,  $m = 0.01$ ). Does that change the general pattern of relative prevalence across towns? What are the practical effects of reducing mobility?**

In order to model this scenario, I have introduced in the last algorithm a new variable called 'm' which represent the factor mobility. Since m value was 1, there were no restriction for cities' population to travel around the country. By introducing this variable in the model, we reduce this mobility in a 90% or 99%.

The results for  $m = 0.1$ , were the next ones:

- Maximum prevalence to reach is 100 ( $0.01 \cdot 10000$ ).
- City with the highest prevalence is city 1 (127.43). This is reasonable because it was the first infected. In addition, it is the same maximum prevalence that in last scenario which makes sense.
- Mean prevalence at the end of the simulation is 92.85. This result shows that cities were infected much later than in last scenario because there is more variability in their prevalence.
- Less prevalence is found in city 50 which was the last infected. Its final prevalence is 1.26, very far from the mean value. This means that city 50 get infected near the end of the simulation. Therefore, as I have said, I can affirm that most of cities where infected later than with completely free mobility.

The results for  $m = 0.001$  were the next ones:

- Maximum prevalence to reach is 100 ( $0.01 \cdot 10000$ ).
- City with the highest prevalence is again city 1 (127.43).
- Mean prevalence at the end of the simulation is 67.68. This result shows that cities were infected much later than in two last scenarios.
- Less prevalence is found in city 50 which was the last infected. Its final prevalence is 1.26, very far from the mean value. Therefore, as I have said, I can affirm that most of cities where infected later than with completely free mobility.

#### Conclusion:

The created variable  $m$  could be understood as a quarantine method because it limits the mobility of the population of infected cities in order to not spread the disease. As  $m$  is introduced in the algorithm by a multiplicator of the prevalence it is proportional to it. This means that cities with higher prevalence will have highest mobility restrictions.

As the mean prevalence in the country was low its value as  $m$  was increased, it is sure that contentions in mobility and quarantine methods are effective.

## Exercise 6.2 – The effects of heterogeneity in the number of contacts

**Carry out a numerical simulation of a rumor propagation process that starts with a single individual carrying a hot new. At each time step, individuals can pass the rumor to their neighbors with probability  $\mu$  (per neighbor). With probability  $\beta$ , informed individuals stop propagating the rumor as it is not a hot new anymore (this would be equivalent to recovery in a SIR model). Follow the numbers of informed individuals at each time step and compare with the mean-field solution. (Hint: the mean-field in this case corresponds to the propagation at a constant speed given by the average number of contacts.)**

For this task I have developed a model which simulates our Modelization and simulation of biosystem class in the very beginning of the course. There are 30 students attending this subject and as we are in the first days, we only know 2 people in class each one.

Suppose now that student A said that student B has cheated in one of the assignments we had made. How long will it take for the whole class to know this rumor?

The mentioned algorithm will show the evolution of the rumor from the day that student A told to one of his friends to the day in which all the class knows it or in which the rumor stops due its antique.

The results of the execution of the algorithm with 30 students which have 2 close friends ( $\mu = 2/30 = 0.066$ ) each one and with  $\beta = 0.02$  are:

- All the class will be aware about the rumor of the cheater in day 26.

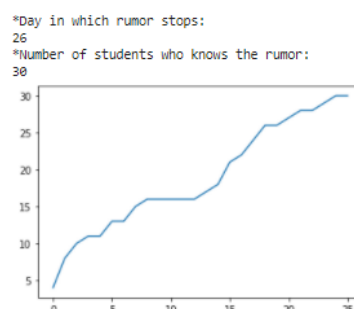


Figure 4: evolution of the rumor spreading.

If we execute the same model when students know more each other and therefore have a greater number of close friends, for instance 5, the rumor spread speed accelerates:

- In this case, all the class will be aware of the rumor earlier, at day 14.

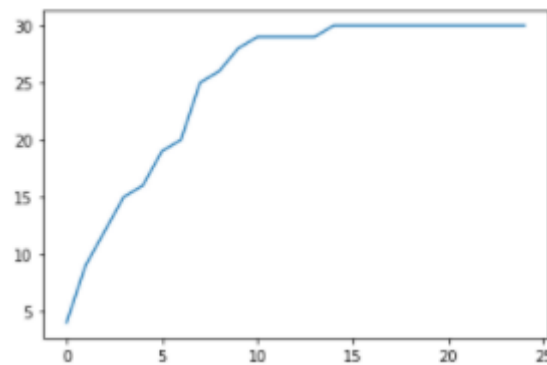


Figure 5: evolution of the rumor spreading with more close friends.

Let's play now with  $\beta$  parameter. This parameter could represent, as the exercise statement said, the antique of the rumor but it can also represent the confidentiality of the closest friends of each student. As  $\mu$  value increases, more discreet friends behave. In this sense, I have simulated the same class with the same number of close friends per student (5) but this time with a higher level of confidentiality ( $\beta = 0.12$ ).

- The result is of course, that less student will finally know the rumor, only 8. And in consequence the rumor will stop earlier, at day 2.

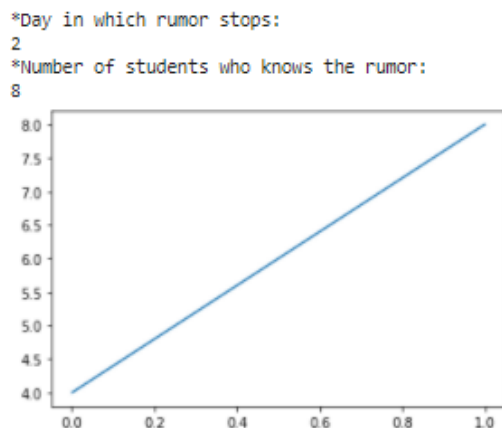


Figure 6: evolution of the rumor spread in days.

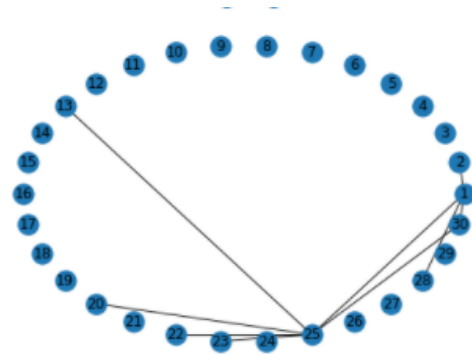


Figure 7: connections between students which have spread the rumor by the end of day 2.

## Compare the following situations in a large population ( $N \geq 1000$ ):

**Note:** I have tried the mentioned algorithms with this size. However, it has been impossible for my computer to execute it due to its high level of iterations. Therefore, I have model it with  $N = 500$

### 1. Individuals are randomly connected (Erdős-Rényi network);

Let's raise the number of students. Suppose that now we focus on the whole faculty, so there are about 500 students. Imagine each student has 15 close friends and suddenly student A told his friends that has the answers of all of the exams of the software engineer degree. As you can imagine this is a very juicy rumor so we will suppose  $\beta = 0.02$ . How long will it take to the whole faculty to know the rumor and ask student A for the answers?

- In this scenario and taken into account this parameter I have obtained the following results:
  - The rumor will be known by all the faculty in 39 days!

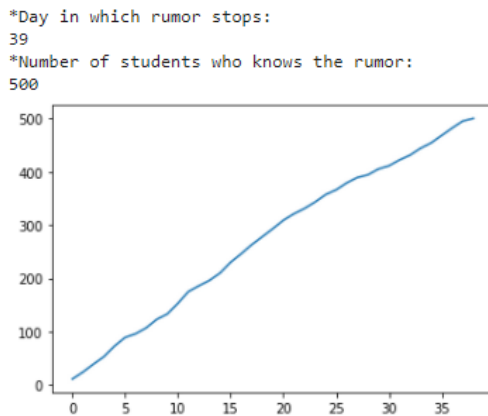


Figure 8: evolution of the rumor spread in days.

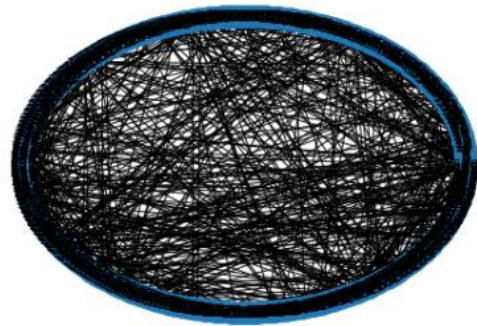


Figure 9: connections between students which have spread the rumor by the end of day 39.

### 2. Individuals are fixed in a 2D lattice;

**Small World Phenomenon** is the study and notion that we are all connected via a small number of edges. Watts and Strogatz came up with a model about how to construct Small World Networks. Let there be  $n$  nodes, where each node is connected to  $m$  nearest neighbors, this is known as *Regular Lattice*.

In Python, we can easily construct a Small World Network using Networkx library and `wattsstrogatzgraph()` function to create the graph. Using the same algorithm, I have obtained the following results:

- The rumor will be known by all the faculty in 402 days.

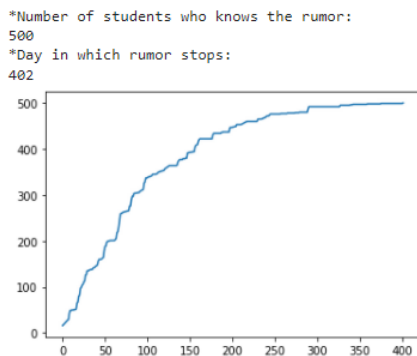


Figure 9: evolution of the rumor spread in days.

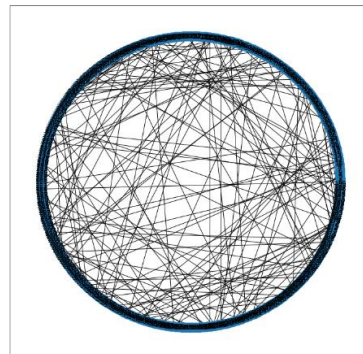


Figure 10: connections between students which have spread the rumor by the end of day 402.



### 3. Individuals are the nodes of a preferential attachment network.

The Barabási–Albert model is one of several proposed models that generate scale-free networks. It incorporates an important general concept: preferential attachment which widely exist in real networks. Preferential attachment means that the more connected a node is, the more likely it is to receive new links. Nodes with higher degree have stronger ability to grab links added to the network.

In order to visualize the structure of this kind of network I have used the python library networkx and its function *barabasi\_albert\_graph()*. This kind of networks are considered the best approach to social networks due its preferential attachment can be considered as popularity.

Using the same algorithm, I have obtained the following results:

- The rumor will be known by all the faculty in 58 days.

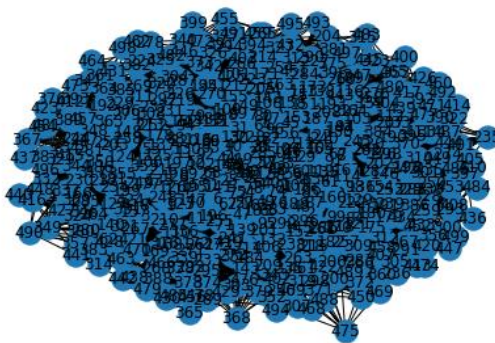


Figure 9: evolution of the rumor spread in days.

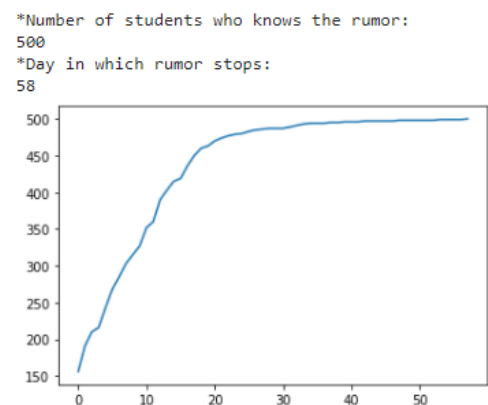


Figure 10: connections between students

which have spread the rumor by the end of day 58.

**Do the results depend on the initial individual? Can you find some general principle explaining when the rumor spreads to the whole population and when it dies out, leading to just a locally “informed” group?**

Time taken by the rumor to be known by all the class or faculty, depends on the initial student. If student A has lots of close friends (connections), the rumor will spread faster and therefore is more likely to be known by all the students. For instance, if in figure 9 we observe node 475 it is appreciable that is far from the center of the net and have less connection than the ones that are in the center. If student A were node 475, rumor will take much longer to spread.

If beta value is taken into account in the simulation, the highest value of the called “confidentiality”, the lowest probability for the rumor to be completely spread for all students even student A has plenty of connections.

Finally, it is necessary to highlight that rumor are more likely to be widely spread for smaller groups than for bigger in the same conditions of network type and beta value.



## Conclusions:

To conclude I will compare three plots bellow which correspond to Erdős-Rényi network, 2D lattice and preferential attachment network respectively. As we can observe, the fastest spread is produced in Erdős-Rényi network. However, this type of network is not much similar to reality because every node has the same chance to spread the rumor besides its number of connections (friends).

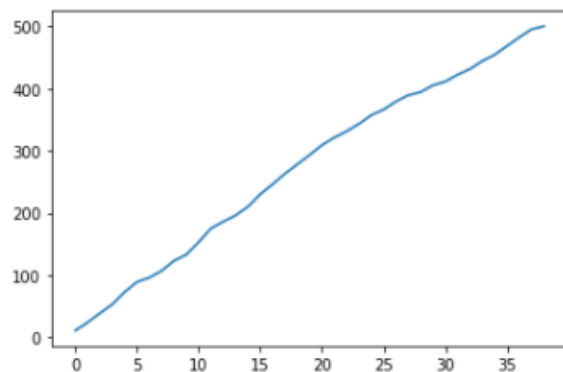
On the other hand, 2D lattice and preferential attachment graphs which are closer to reality because they include topics as popularity or the small world theory, obtains larger result talking about spreading rumors.

\*Day in which rumor stops:

39

\*Number of students who knows the rumor:

500

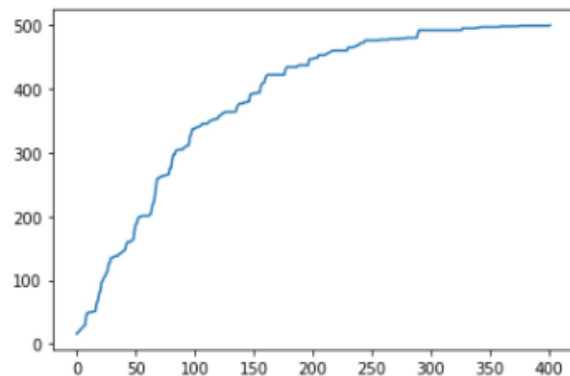


\*Number of students who knows the rumor:

500

\*Day in which rumor stops:

402

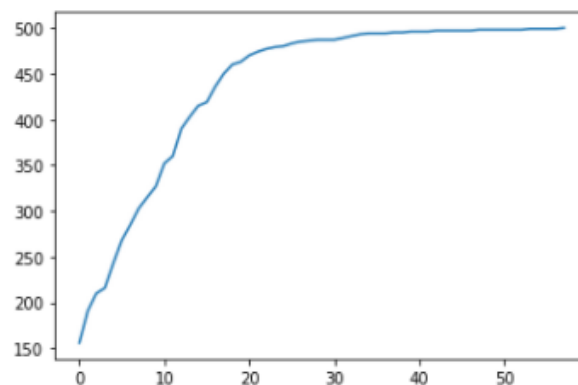


\*Number of students who knows the rumor:

500

\*Day in which rumor stops:

58



## Annex:

Code is accessible in:

Google colab:

[https://colab.research.google.com/drive/10RT0B7KYcV8mDGWcd3RtxQkKLuCpQnY\\_?usp=sharing](https://colab.research.google.com/drive/10RT0B7KYcV8mDGWcd3RtxQkKLuCpQnY_?usp=sharing)

GitHub:

<https://github.com/SandraAlonso/Modelization-and-simulation-of-biosystems/tree/main/Extended%20biological%20process/The%20role%20of%20networks>