Discrete Applied Mathematics Final

Jason Chung and Jasdeep Singh

December 10, 2021

1 Introduction

This project involves data science, graph theory, and computer simulations: We predict the spread of epidemics using contact networks, which are networks that provide temporal data representative of the contacts between individuals in varying environments. We model the networks using graphs, which are each encoded with an adjacency matrix A. The set of nodes is V, and the set of edges is E. Thus, each graph G is described with the pair G = (V, E).

In order to make predictions, we run computer simulations on two types of contact networks: face-to-face contact networks and co-presence networks. Face-to-face networks consist of individuals who wear a sensor capable of detecting other individuals' sensors in a face-to-face close range proximity. They may be difficult to obtain because they require users to carry special sensors. Co-presence networks are constructed with RFID readers - albeit with a coarser resolution than the face-to-face networks - located at various locations within the space the individuals reside. Unlike face-to-face networks, their information is easily available since they only require that some RFID readers be installed in an area.

Part of our objective is to also compare these networks, so we compute and consider network statistics that provide a signature of their specific geometry: density, degree distribution, and clustering coefficient. Our main goal is to determine if one can exchange the precise yet expensive face-to-face contact data with the vague yet cheap co presence data, while considering our simulation of the spread of an epidemic on the networks.

2 Question One

2.1 Given Data Set

name	location	face-to-face			co-presence		
		n	m	$\overline{d}$	n	m	$\overline{d}$
InVS13	French Institute for Public Health Surveillance	92	755	16	95	3,915	82
InVS15	French Institute for Public Health Surveillance	217	4,274	39	219	16,725	153
LH10	Hospital ward (Lyon, France)	76	1,156	30	73	1,381	38
LyonSchool	Primary school (Lyon, France)	242	8,317	69	242	26,594	220
SFHH	2009 French Society for Hospital Hygiene Conference	403	9,565	48	403	73,557	365
Thiers13	High School (Marseilles, France)	327	5,818	35	328	43,496	265

Figure 1: Data from SocioPatterns website including name of dataset, location of data collection, number of nodes n, number of edges m, and average degree $\overline{d}$ of the 12 face-to-face contact (left) and co-presence(right) networks (6 of each type of network).

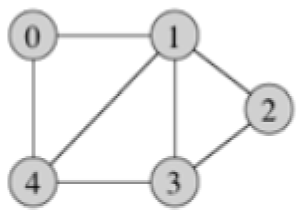
2.2 Loading in the Data

The data.mat file has all of the data for the face-to-face and co-presence networks above. In order to load this file into our matlab program, we can type the following line:

```
load('-mat', 'data.mat');
```

2.3 Computation of Edges

Each network can be represented with an adjacency matrix that looks like this:



$$\begin{array}{c|ccccc|} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

Figure 2: Example Adjacency Matrix of a Network

Our first task is to calculate the number of edges in each network. To do this, we add up all of the 1's in the adjacency matrix and divide by 2. The reason we divide by 2 is because each edge is counted twice since the matrix is symmetric along the diagonal. We use the `sum(A, 'all')` command on MATLAB to calculate the sum of `A` where `A` is a matrix and `'all'` is to specify that all of the values in the matrix should are added. We can also just calculate the number of 1's in the upper or lower triangle of the matrix. The code complied to complete this computation is below:

```
A_InVS13_SumOfEdges = sum(A_InVS13, 'all')/2;
A_InVS15_SumOfEdges = sum(A_InVS15, 'all')/2;
A_LH10_SumOfEdges = sum(A_LH10, 'all')/2;
A_LyonSchool_SumOfEdges = sum(A_LyonSchool, 'all')/2;
A_pres_InVS13_SumOfEdges = sum(A_pres_InVS13, 'all')/2;
A_pres_InVS15_SumOfEdges = sum(A_pres_InVS15, 'all')/2;
A_pres_LH10_SumOfEdges = sum(A_pres_LH10, 'all')/2;
A_pres_LyonSchool_SumOfEdges = sum(A_pres_LyonSchool, 'all')/2;
A_pres_SFHH_SumOfEdges = sum(A_pres_SFHH, 'all')/2;
A_pres_Thiers13_SumOfEdges = sum(A_pres_Thiers13, 'all')/2;
A_SFHH_SumOfEdges = sum(A_SFHH, 'all')/2;
A_Thiers13_SumOfEdges = sum(A_Thiers13, 'all')/2;
```

Executing these lines gave us the same number of edges for each network found in Figure 1.

2.4 Computation of Network Density

In order to calculate the density of each network, we must use the following equation:

$$\text{density} = \frac{(2m)}{n(n-1)}$$

Below are the lines of code compiled in order to compute network densities:

```
A_InVS13_Density = (2*A_InVS13_SumOfEdges)/(length(A_InVS13)*(length(A_InVS13) - 1));
A_InVS15_Density = (2*A_InVS15_SumOfEdges)/(length(A_InVS15)*(length(A_InVS15) - 1));
A_LH10_Density = (2*A_LH10_SumOfEdges)/(length(A_LH10)*(length(A_LH10) - 1));
A_LyonSchool_Density =
```

```
(2*A_LyonSchool_SumOfEdges)/(length(A_LyonSchool)*(length(A_LyonSchool) - 1));
A_pres_InVS13_Density =
(2*A_pres_InVS13_SumOfEdges)/(length(A_pres_InVS13)*(length(A_pres_InVS13) - 1));
A_pres_InVS15_Density =
(2*A_pres_InVS15_SumOfEdges)/(length(A_pres_InVS15)*(length(A_pres_InVS15) - 1));
A_pres_LH10_Density =
(2*A_pres_LH10_SumOfEdges)/(length(A_pres_LH10)*(length(A_pres_LH10) - 1));
A_pres_LyonSchool_Density =
(2*A_pres_LyonSchool_SumOfEdges)/(length(A_pres_LyonSchool)*(length(A_pres_LyonSchool) - 1));
A_pres_SFHH_Density =
(2*A_pres_SFHH_SumOfEdges)/(length(A_pres_SFHH)*(length(A_pres_SFHH) - 1));
A_pres_Thiers13_Density =
(2*A_pres_Thiers13_SumOfEdges)/(length(A_pres_Thiers13)*(length(A_pres_Thiers13) - 1));
A_SFHH_Density = (2*A_SFHH_SumOfEdges)/(length(A_SFHH)*(length(A_SFHH) - 1));
A_Thiers13_Density =
(2*A_Thiers13_SumOfEdges)/(length(A_Thiers13)*(length(A_Thiers13) - 1));
```

*Note that some lines were split into two lines, so the semicolons mark the end of each line.

The table below contains the computed values for density:

Name	Location	Density	
		Face-to-Face	Co-Presence
lnVS13	French Institute for Public Health Surveillance	0.1804	0.8768
lnVS15	French Institute for Public Health Surveillance	0.1824	0.7006
LH10	Hospital ward (Lyon, France)	0.4056	0.5255
LyonSchool	Primary school (Lyon, France)	0.2852	0.9120
SFHH	2009 French Society for Hospital Hygiene Conference	0.1181	0.9081
Thiers13	High School (Marseilles, France)	0.1092	0.8111

Figure 3: Density of Networks

2.5 Computation of Network Degree Distribution

The degree of a node in a network is the number of connections it has to other nodes, so the degree distribution is the probability distribution of these degrees in the entire network. It is given by the formula:

$$deg(v) = \sum_{u:(u,v)\epsilon E} w_{uv}$$

In order to calculate the degree distribution of a network, we must add up all of the 1's in each row or column for every node. Below are the lines of code to compute degree distribution of the networks:

```
A_InVS13_DegreeDistribution = sum(A_InVS13, 1);
A_InVS15_DegreeDistribution = sum(A_InVS15, 1);
A_LH10_DegreeDistribution = sum(A_LH10, 1);
A_LyonSchool_DegreeDistribution = sum(A_LyonSchool, 1);
A_pres_InVS13_DegreeDistribution = sum(A_pres_InVS13, 1);
A_pres_InVS15_DegreeDistribution = sum(A_pres_InVS15, 1);
A_pres_LH10_DegreeDistribution = sum(A_pres_LH10, 1);
```

```
A_pres_LyonSchool_DegreeDistribution = sum(A_pres_LyonSchool, 1);
A_pres_SFHH_DegreeDistribution = sum(A_pres_SFHH, 1);
A_pres_Thiers13_DegreeDistribution = sum(A_pres_Thiers13, 1);
A_SFHH_DegreeDistribution = sum(A_SFHH, 1);
A_Thiers13_DegreeDistribution = sum(A_Thiers13, 1);
```

In order to analyze additional statistics related to the degree distributions, we calculated the mean, standard deviation, and variance of each degree distribution as follows:

Name	Face to Face			Co-Presence		
	Mean	Standard Deviation	Variance	Mean	Standard Deviation	Variance
InVS13	16.413	7.6117	57.9374	82.4211	13.8898	192.9272
InVS15	39.3917	15.6133	243.7764	152.7397	37.1716	1381.7255
LH10	30.4211	16.085	258.727	37.8356	16.3622	267.7226
LyonSchool	68.7355	26.626	708.9422	219.7851	19.788	391.5636
SFHH	47.469	30.1809	910.8865	365.0471	50.196	2519.6421
Thiers13	35.5841	13.5182	182.7406	265.2195	33.6613	1133.0832

Figure 4: Mean, Standard Deviation, and Variance of the Degree Distribution of Networks

2.6 Plotting Degree Distribution

In order to compare the face-to-face and co-presence networks, we need to visualize the degree distributions than using histograms. Below are the lines of code we compiled to populate, label, and display the histograms of the degree distributions:

```
A_InVS13_DegreeDistributionHistogram = histogram(A_InVS13_DegreeDistribution);
A_InVS13_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Face-to-Face InVS13 Degree Distribution');
A_pres_InVS13_DegreeDistributionHistogram = histogram(A_pres_InVS13_DegreeDistribution);
A_pres_InVS13_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence InVS13 Degree Distribution');
```

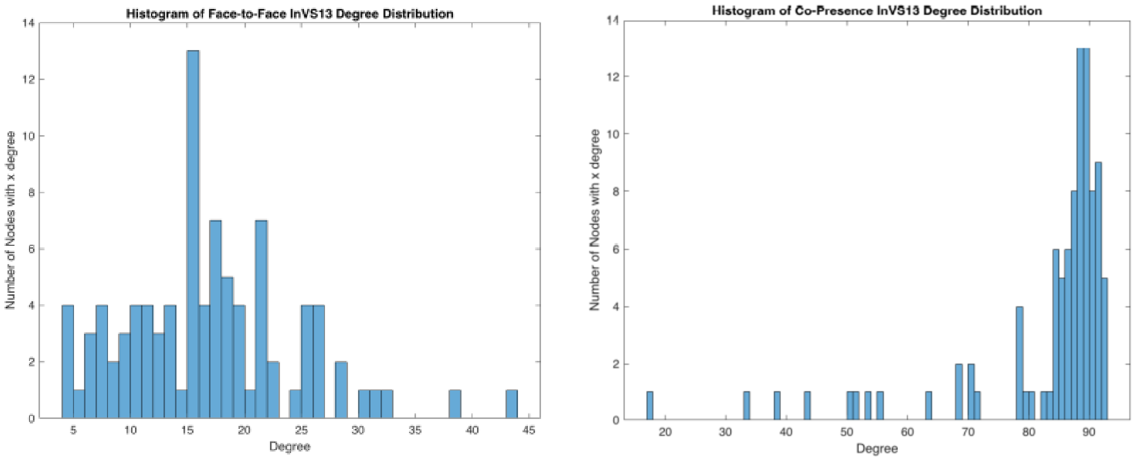


Figure 5: lnVS13 Degree Distribution

```
A_InVS15_DegreeDistributionHistogram = histogram(A_InVS15_DegreeDistribution);
A_InVS15_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Face-to-Face InVS15 Degree Distribution');
A_pres_InVS15_DegreeDistributionHistogram = histogram(A_pres_InVS15_DegreeDistribution);
A_pres_InVS15_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence InVS15 Degree Distribution');
```

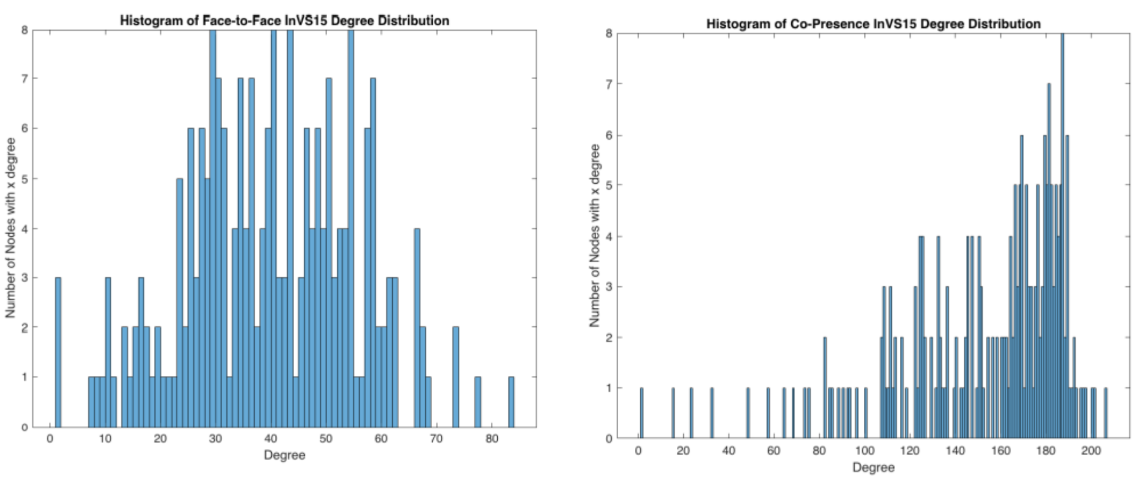


Figure 6: lnVS15 Degree Distribution

```
A_LH10_DegreeDistributionHistogram = histogram(A_LH10_DegreeDistribution);
A_LH10_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Face-to-Face LH10 Degree Distribution');
A_pres_LH10_DegreeDistributionHistogram = histogram(A_pres_LH10_DegreeDistribution);
A_pres_LH10_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
```

```
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence LH10 Degree Distribution');
```

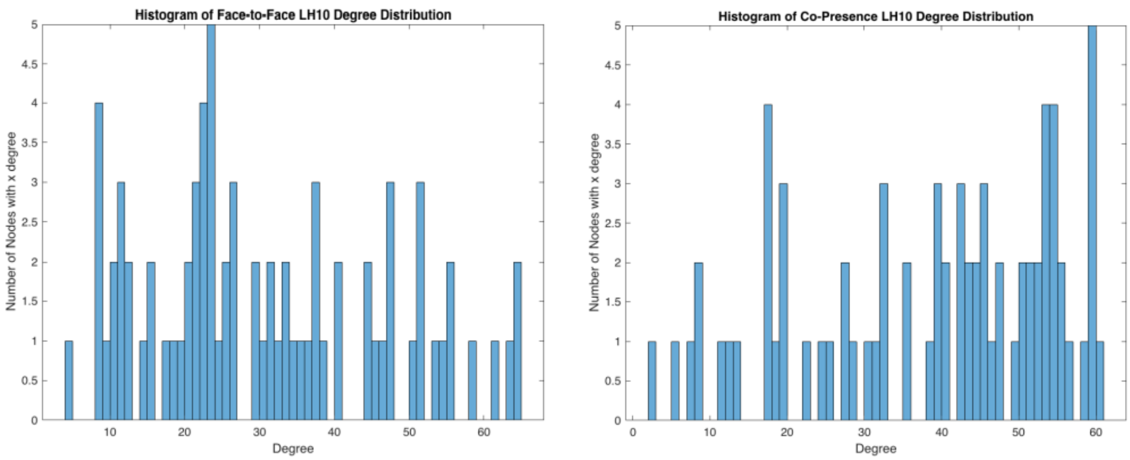


Figure 7: LH10 Degree Distribution

```
A_LyonSchool_DegreeDistributionHistogram = histogram(A_LyonSchool_DegreeDistribution);
A_LyonSchool_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Face-to-Face LyonSchool Degree Distribution');
A_pres_LyonSchool_DegreeDistributionHistogram =
histogram(A_pres_LyonSchool_DegreeDistribution);
A_pres_LyonSchool_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence LyonSchool Degree Distribution');
```

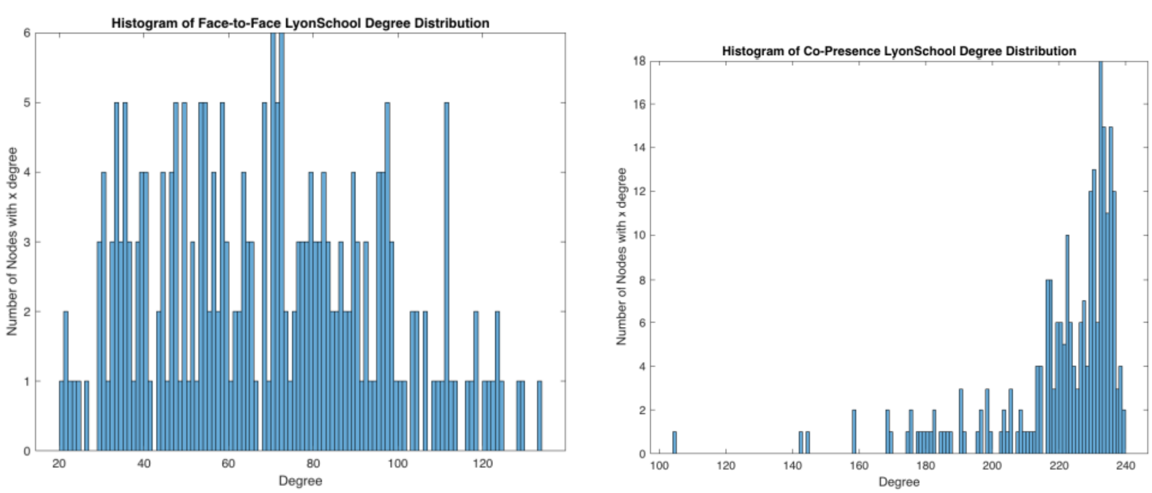


Figure 8: LyonSchool Degree Distribution

```
A_SFHH_DegreeDistributionHistogram = histogram(A_SFHH_DegreeDistribution);
A_SFHH_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
```

```
title('Histogram of Face-to-Face SFHH Degree Distribution');
A_pres_SFHH_DegreeDistributionHistogram = histogram(A_pres_SFHH_DegreeDistribution);
A_pres_SFHH_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence SFHH Degree Distribution');
```

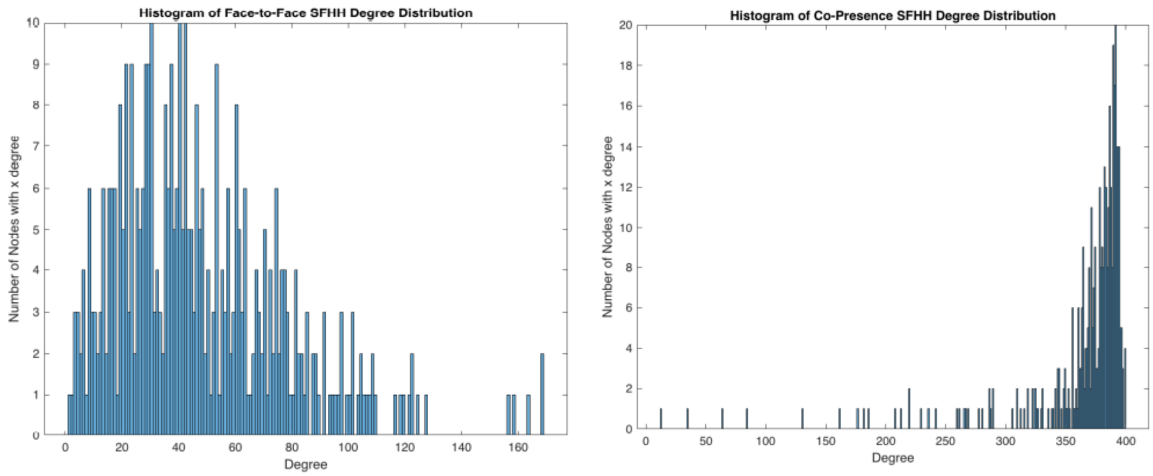


Figure 9: SFHH Degree Distribution

```
A_Thiers13_DegreeDistributionHistogram = histogram(A_Thiers13_DegreeDistribution);
A_Thiers13_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Face-to-Face Thiers13 Degree Distribution');
A_pres_Thiers13_DegreeDistributionHistogram = histogram(A_pres_Thiers13_DegreeDistribution);
A_pres_Thiers13_DegreeDistributionHistogram.BinWidth = 1;
xlabel('Degree');
ylabel('Number of Nodes with x degree');
title('Histogram of Co-Presence Thiers13 Degree Distribution');
```

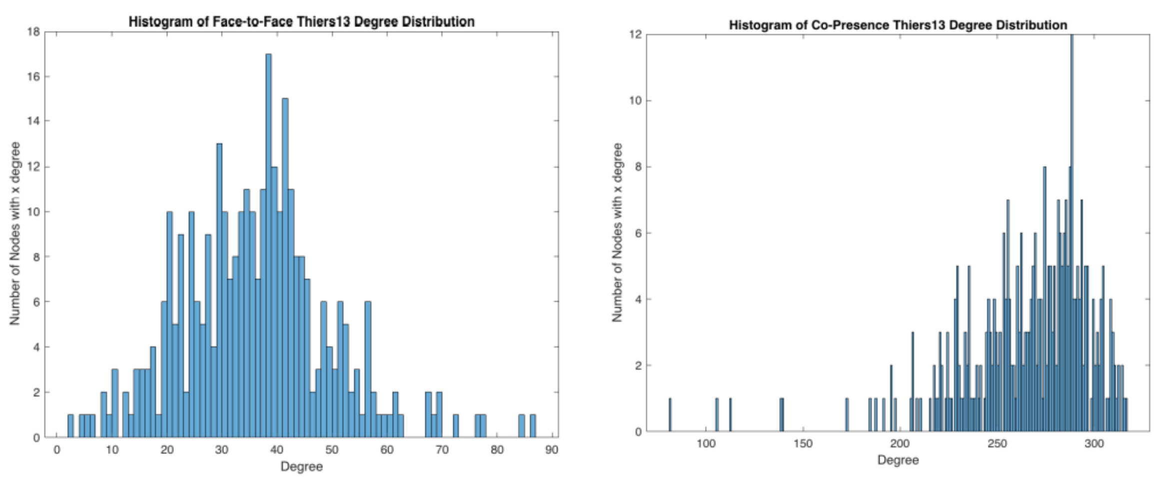


Figure 10: Thiers13 Degree Distribution

2.7 Computation of Network Clustering Coefficient

The clustering coefficient is a measure of how much nodes in a graph tend to cluster together. In our case, it is a measure of how people in the networks tend to cluster. The coefficient is given by:

cc(v) $= \frac{(2|\Delta v|)}{deg(v)(deg(v)-1)}$

In order to calculate the clustering coefficient we need to find Δv, which is the number of triangles, that is paths of length three to itself, a node has. Thus, we first have to cube the matrix of the network, so that it shows paths of length three. Then, we can take the diagonal of the matrix at a given node to get the paths of length 3 from and to that node, and multiply it by 2. Finally, we divide the value by the degree of the vertex/node and the degree of the vertex/node minus 1 in order to get the clustering coefficient for that node. Below are the lines of code we compiled to compute the clustering coefficients for each network:

```
A_InVS13_ClusteringCoefficient = rdivide(2 * transpose(diag(A_InVS13^3)),
times(A_InVS13_DegreeDistribution, A_InVS13_DegreeDistribution - 1));
A_InVS15_ClusteringCoefficient = rdivide(2 * transpose(diag(A_InVS15^3)),
times(A_InVS15_DegreeDistribution, A_InVS15_DegreeDistribution - 1));
A_LH10_ClusteringCoefficient = rdivide(2 * transpose(diag(A_LH10^3)),
times(A_LH10_DegreeDistribution, A_LH10_DegreeDistribution - 1));
A_LyonSchool_ClusteringCoefficient = rdivide(2 * transpose(diag(A_LyonSchool^3)),
times(A_LyonSchool_DegreeDistribution, A_LyonSchool_DegreeDistribution - 1));
A_pres_InVS13_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_InVS13^3)),
times(A_pres_InVS13_DegreeDistribution, A_pres_InVS13_DegreeDistribution - 1));
A_pres_InVS15_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_InVS15^3)),
times(A_pres_InVS15_DegreeDistribution, A_pres_InVS15_DegreeDistribution - 1));
A_pres_LH10_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_LH10^3)),
times(A_pres_LH10_DegreeDistribution, A_pres_LH10_DegreeDistribution - 1));
A_pres_LyonSchool_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_LyonSchool^3)),
times(A_pres_LyonSchool_DegreeDistribution, A_pres_LyonSchool_DegreeDistribution - 1));
A_pres_SFHH_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_SFHH^3)),
times(A_pres_SFHH_DegreeDistribution, A_pres_SFHH_DegreeDistribution - 1));
A_pres_Thiers13_ClusteringCoefficient = rdivide(2 * transpose(diag(A_pres_Thiers13^3)),
times(A_pres_Thiers13_DegreeDistribution, A_pres_Thiers13_DegreeDistribution - 1));
A_SFHH_ClusteringCoefficient = rdivide(2 * transpose(diag(A_SFHH^3)),
times(A_SFHH_DegreeDistribution, A_SFHH_DegreeDistribution - 1));
A_Thiers13_ClusteringCoefficient = rdivide(2 * transpose(diag(A_Thiers13^3)),
times(A_Thiers13_DegreeDistribution, A_Thiers13_DegreeDistribution - 1));
```

After executing the above lines of code, we realized that sometimes the degree of a node could be 0, in which case the clustering coefficient for that node was nonexistent (you cannot divide by 0). However, the clustering coefficient had to be 0 in these cases, so we could calculate and analyze the clustering coefficient average. Therefore, we replaced every NaN value with a 0 with the following lines of code:

```
A_InVS13_ClusteringCoefficient(isnan(A_InVS13_ClusteringCoefficient))=0
A_InVS15_ClusteringCoefficient(isnan(A_InVS15_ClusteringCoefficient))=0
A_LH10_ClusteringCoefficient(isnan(A_LH10_ClusteringCoefficient))=0
A_LyonSchool_ClusteringCoefficient(isnan(A_LyonSchool_ClusteringCoefficient))=0
A_pres_InVS13_ClusteringCoefficient(isnan(A_pres_InVS13_ClusteringCoefficient))=0
A_pres_InVS15_ClusteringCoefficient(isnan(A_pres_InVS15_ClusteringCoefficient))=0
A_pres_LH10_ClusteringCoefficient(isnan(A_pres_LH10_ClusteringCoefficient))=0
A_pres_LyonSchool_ClusteringCoefficient(isnan(A_pres_LyonSchool_ClusteringCoefficient))=0
A_pres_SFHH_ClusteringCoefficient(isnan(A_pres_SFHH_ClusteringCoefficient))=0
A_pres_Thiers13_ClusteringCoefficient(isnan(A_pres_Thiers13_ClusteringCoefficient))=0
A_SFHH_ClusteringCoefficient(isnan(A_SFHH_ClusteringCoefficient))=0
A_Thiers13_ClusteringCoefficient(isnan(A_Thiers13_ClusteringCoefficient))=0
```

In order better analyze the clustering coefficients, we calculated the clustering coefficient average for each network with the following lines of code:

```
A_InVS13_ClusteringCoefficient(isnan(A_InVS13_ClusteringCoefficient))=0
A_InVS15_ClusteringCoefficient(isnan(A_InVS15_ClusteringCoefficient))=0
A_LH10_ClusteringCoefficient(isnan(A_LH10_ClusteringCoefficient))=0
A_LyonSchool_ClusteringCoefficient(isnan(A_LyonSchool_ClusteringCoefficient))=0
A_pres_InVS13_ClusteringCoefficient(isnan(A_pres_InVS13_ClusteringCoefficient))=0
A_pres_InVS15_ClusteringCoefficient(isnan(A_pres_InVS15_ClusteringCoefficient))=0
A_pres_LH10_ClusteringCoefficient(isnan(A_pres_LH10_ClusteringCoefficient))=0
A_pres_LyonSchool_ClusteringCoefficient(isnan(A_pres_LyonSchool_ClusteringCoefficient))=0
A_pres_SFHH_ClusteringCoefficient(isnan(A_pres_SFHH_ClusteringCoefficient))=0
A_pres_Thiers13_ClusteringCoefficient(isnan(A_pres_Thiers13_ClusteringCoefficient))=0
A_SFHH_ClusteringCoefficient(isnan(A_SFHH_ClusteringCoefficient))=0
A_Thiers13_ClusteringCoefficient(isnan(A_Thiers13_ClusteringCoefficient))=0
```

Below are the clustering coefficient averages we computed for each network:

Name	Location	Clustering Coefficient Average	
		Face-to-Face	Co-Presence
lnVS13	French Institute for Public Health Surveillance	0.8521	1.8557
lnVS15	French Institute for Public Health Surveillance	0.7627	1.5976
LH10	Hospital ward (Lyon, France)	1.3573	1.5522
LyonSchool	Primary school (Lyon, France)	1.0511	1.8578
SFHH	2009 French Society for Hospital Hygiene Conference	0.5635	1.8877
Thiers13	High School (Marseilles, France)	1.0070	1.6853

Figure 11: Clustering Coefficient Average of Networks

2.8 Plotting Clustering Coefficient

Below are the lines of code we compiled to populate, label, and display the histograms of the network clustering coefficient distributions:

```
A_InVS13_ClusteringCoefficientHistogram = histogram(A_InVS13_ClusteringCoefficient);
A_InVS13_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face InVS13 Clustering Coefficient');
A_pres_InVS13_ClusteringCoefficientHistogram = histogram(A_pres_InVS13_ClusteringCoefficient);
A_pres_InVS13_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence InVS13 Clustering Coefficient');
```

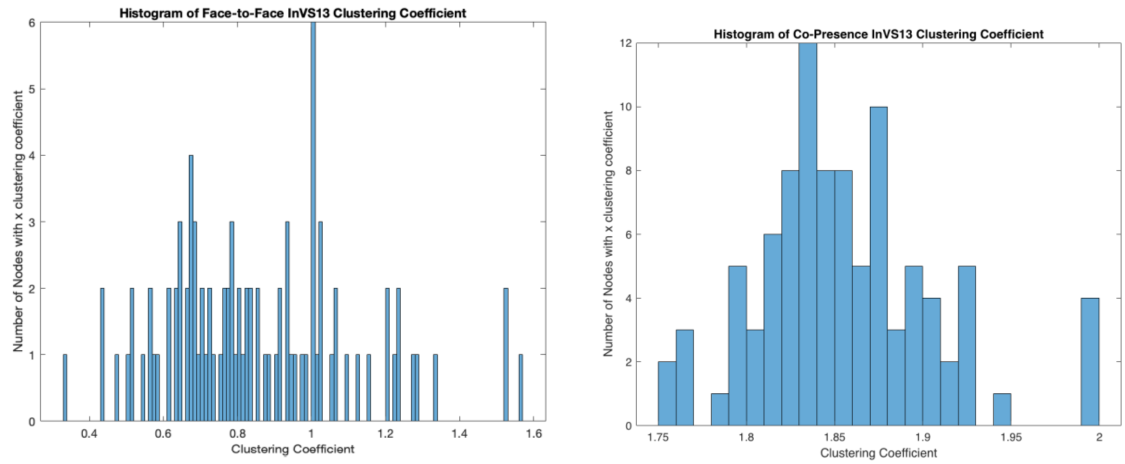


Figure 12: lnVS13 Clustering Coefficient Distribution

```
A_InVS15_ClusteringCoefficientHistogram = histogram(A_InVS15_ClusteringCoefficient);
A_InVS15_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face InVS15 Clustering Coefficient');
A_pres_InVS15_ClusteringCoefficientHistogram = histogram(A_pres_InVS15_ClusteringCoefficient);
A_pres_InVS15_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence InVS15 Clustering Coefficient');
```

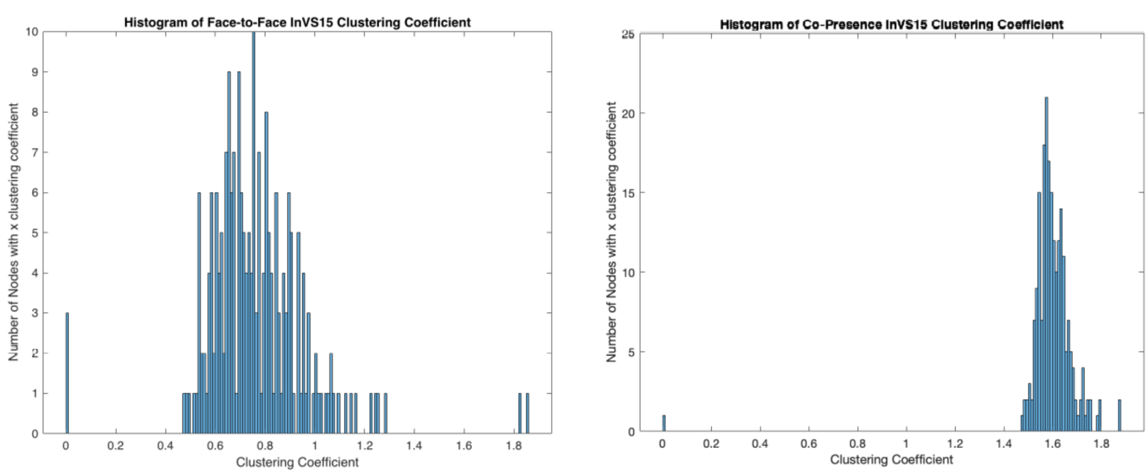


Figure 13: lnVS15 Clustering Coefficient Distribution

```
A_LH10_ClusteringCoefficientHistogram = histogram(A_LH10_ClusteringCoefficient);
A_LH10_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face LH10 Clustering Coefficient');
A_pres_LH10_ClusteringCoefficientHistogram = histogram(A_pres_LH10_ClusteringCoefficient);
A_pres_LH10_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
```

```
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence LH10 Clustering Coefficient');
```

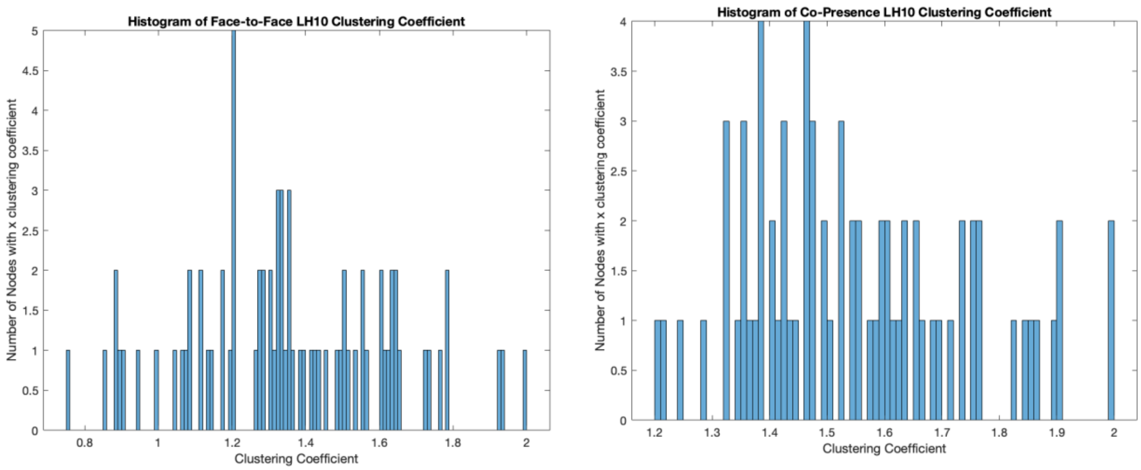


Figure 14: LH10 Clustering Coefficient Distribution

```
A_LyonSchool_ClusteringCoefficientHistogram = histogram(A_LyonSchool_ClusteringCoefficient);
A_LyonSchool_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face LyonSchool Clustering Coefficient');
A_pres_LyonSchool_ClusteringCoefficientHistogram = histogram(A_pres_LyonSchool_ClusteringCoefficient)
A_pres_LyonSchool_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence LyonSchool Clustering Coefficient');
```

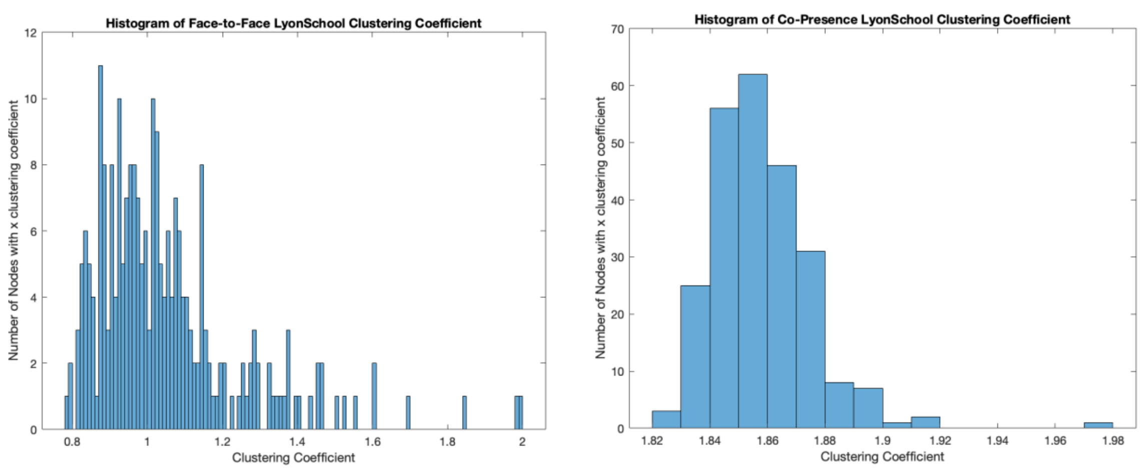


Figure 15: LyonSchool Clustering Coefficient Distribution

```
A_SFHH_ClusteringCoefficientHistogram = histogram(A_SFHH_ClusteringCoefficient);
A_SFHH_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
```

```
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face SFHH Clustering Coefficient');
A_pres_SFHH_ClusteringCoefficientHistogram = histogram(A_pres_SFHH_ClusteringCoefficient);
A_pres_SFHH_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence SFHH Clustering Coefficient');
```

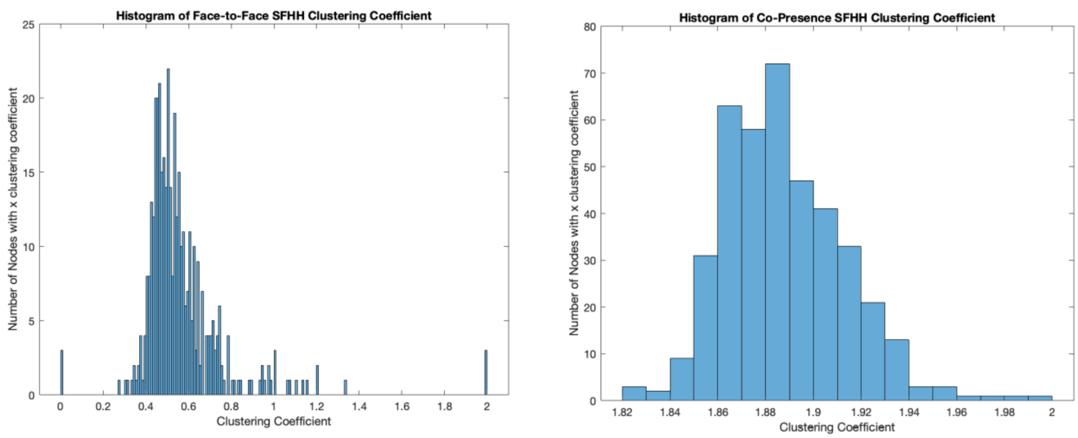


Figure 16: SFHH Clustering Coefficient Distribution

```
A_Thiers13_ClusteringCoefficientHistogram = histogram(A_Thiers13_ClusteringCoefficient);
A_Thiers13_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Face-to-Face Thiers13 Clustering Coefficient');
A_pres_Thiers13_ClusteringCoefficientHistogram = histogram(A_pres_Thiers13_ClusteringCoefficient);
A_pres_Thiers13_ClusteringCoefficientHistogram.BinWidth = 0.01;
xlabel('Clustering Coefficient')
ylabel('Number of Nodes with x clustering coefficient')
title('Histogram of Co-Presence Thiers13 Clustering Coefficient');
```

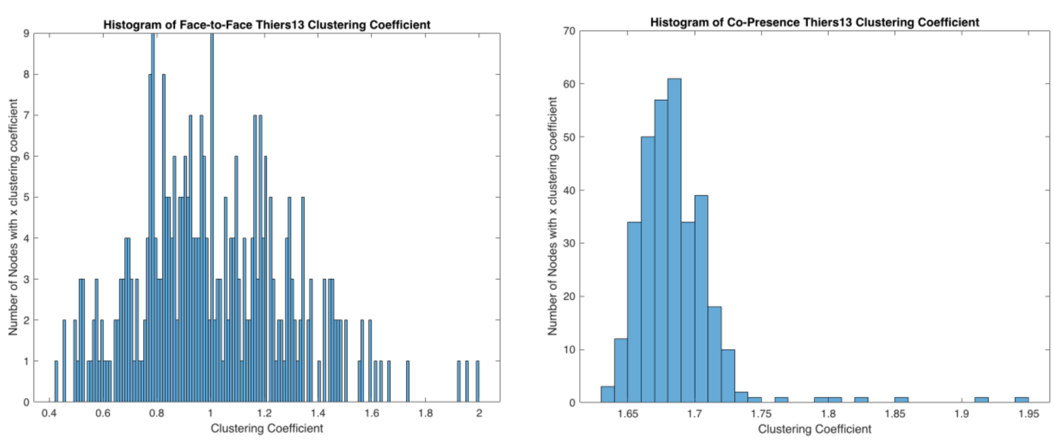


Figure 17: Thiers13 Clustering Coefficient Distribution

3 Question Two

When comparing each location's face-to-face and co-presence networks we notice that the density significantly changes for the co-presence networks. For InVS13, there is a 486% increase from face-to-face to the co-presence network. For InVS15, there is a 384% increase. For LH10, there is a 129% increase. For LyonSchool, there is a 319% increase. For SFHH, there is a 768% increase. And lastly there is a 742% increase for Thiers13. The higher density of co-presence networks shows how they link two people together just because they're in the same area, instead of linking them together because they directly interacted with one another. We can say that with face-to-face networks, the nodes aren't very compact or dense when compared to the co-presence networks. This is due to how the people, or nodes, of the face-to-face networks need to directly interact with one another in order to be connected. Considering this, simulating a virus over the co-presence network would mean that everyone in a given area is connected and susceptible to disease, whereas in the face-to-face networks the virus can only spread through the limited direct contacts.

The degree distribution shows that almost all nodes are connected with one another in the co-presence networks. Whereas the face-to-face networks show that all nodes may not be well connected with each other as the co-presence networks show. The mean for degree distributions for co-presence networks significantly increase (with the exception of LH10) as you can see in Figure 4. Additionally the standard deviation increases significantly (with the exception of LH10). We can also see in the histograms provided (Right side of Figures 5, 6, 8, 9, and 10) that a lot of the co-presence networks are uniform and left skewed. The co-presence networks create an upward trend towards the right of the graph that grows exponentially, hence a left skewed histogram. But for the face-to-face networks we notice that the histograms are uniform and symmetric. There is also no type of skew for the face-to-face networks. On the other hand, LH10 has very similar co-presence and face-to-face degree distribution networks. There are lots of gaps in the two histograms which means they are not uniform and they are both mostly symmetric. So far the data trends towards only using co-presence networks in hospital wards rather than other locations.

Lastly the clustering coefficient in Figure 5 shows a significant increase in co-presence networks when compared to face-to-face networks. The co-presence networks may associate many nodes with one another just because they are in the same room whereas face-to-face shows that just because they are in the same room doesn't mean they came in contact with each other. We can see this with the 0.67 or more increase in clustering coefficient in co-presence networks from face-to-face networks (with the exception of LH10).

After analyzing the data, we can now conclude that co-presence networks aren't as accurate as face-to-face networks when in certain environments. In almost every location that is provided, the co-presence networks overestimate the number of nodes that are connected. Now this isn't a bad thing in some situations. For situations where a disease/virus can be highly contagious, one may want to use co-presence networks to decide which individuals to quarantine. It may be beneficial to quarantine many individuals because of a highly contagious disease/virus. But in the case of a disease/virus that isn't very contagious, it may be a consequential to quarantine many individuals after falsely concluding that they may be exposed to a virus or disease. Although face-to-face networks are more expensive, they may save a country from catastrophic disasters such as quarantining many important figures or essential workers during a pandemic, and/or causing the economic market to crash. Although for hospital wards, it may be useful to save money and just use a co-presence network since there is not a significant difference in the density, degree distribution, and clustering coefficient when comparing the hospital ward's (LH10) face-to-face and co-presence network. In any case, simulating epidemics on these networks can also give us insight on which type of network gives us a better representation of spreading epidemics in various circumstances, and whether or not the networks are interchangeable.