1 The Data

We are using six datasets collected by English, French and Italian researchers. The data are available at the SocioPatterns website. These datasets are well known and are used to benchmark algorithms in network science and graph theory.

The data provide temporal information about face-to-face contacts between individuals in several environments. Each individual was wearing a sensor capable to detect face-to-face close range proximity (1.5 m) between the sensors. In addition to this dynamic face-to-face contact network, the researchers also recorded the location − albeit with a much coarser resolution − of the participants using RFID readers located at various locations inside the environment wherein the participants interacted. There are therefore two datasets, sampled at the same exact instants,

name	location	number of nodes n
InVS13	French Institute for Public Health Surveillance	92
InVS15	French Institute for Public Health Surveillance	232
LH10	Hospital ward (Lyon, France)	81
LyonSchool	Primary school (Lyon, France)	242
SFHH	2009 French Society for Hospital Hygiene Conference	403
Thiers13	High School (Marseilles, France)	326

Table 1: Name, origin, and size (number of nodes) of the datasets

1. a time series of face-to-face contact events between individuals,
2. a time series of co-presence events between individuals.

Table 1 provides the name, origin, and size (number of nodes) of the datasets. In each case, we have access to two matrices that encode face-to-face contacts and co-presence. Each matrix has three columns and a large number of rows. Each row contains a time index t and two node indices i and j. The presence of the row indicates that nodes i and j were in face-to-face (co-presence) contact at time t. The data are sampled with a temporal resolution of 20s. Time is measured in seconds and expressed in UNIX `ctime`.

A ZIP archive of the six datasets can be retrieved here: click to download. You simply need to load the dataset in MATLAB. For instance,

```
>> load tij_LyonSchool.dat
>> whos
  Name                   Size                Bytes  Class     Attributes

  tij_LyonSchool      125773x3             3018552  double
```

The first contact happens between node $i = 1558$ and node $j = 1567$ at time $t = 31220$. The last contact happens between node $i = 1913$ and node $j = 1922$ at time $t = 148120$.

```
>> tij_LyonSchool(1:1,:)

ans = 31220        1558        1567
>> tij_LyonSchool(end:end,:)

ans = 148120        1913        1922
```

Assignment

1. For each of the six datasets, and for each type of contact (face-to-face and co-presence), build the static, temporally aggregated network, where each entry of the adjacency matrix of the corresponding network is given by

$$A(i,j) = \text{total number of contact events between nodes } i \text{ and } j. \tag{1}$$

We can represent each participant as a node of a graph. If two individuals i and j are in contact (face-to-face or co-presence), then we connect i and j with an edge. The weight $A(i,j)$ along the edge provides a natural mechanism to quantify the strength of the face-to-face or co-presence contact between the individuals i and j. If $A(i,j) = 0$, then nodes i and j were never in contact, and there is no edge between i and j.

In this document, all graphs are defined with respect to the temporally aggregated matrices, $\boldsymbol{A}$, defined by (1).

Definition 1 *For each of the six datasets, $i = 1, \ldots, 6$, defined in Table 1, we denote by $\boldsymbol{A}_f^{(i)}$ and by $\boldsymbol{A}_p^{(i)}$ the i^{th} face-to-face and co-presence contact networks respectively.*

2. You should observe that the co-presence graph has many more edges, with larger weights, than the face-to-face contact graph. Please explain this phenomenon.

In the rest of the project, we will sample the denser co-presence graph and compare it to the thinner face-to-face contact graph.

2 How do we compare the original graph to the reduced graph?

Throughout this project, you will evaluate the performance of several algorithms to reduce the size of a large graph $G = (V, E)$. When sampling very large graphs, one is confronted with the following fundamental question: can the large scale structural properties of the graph G be recovered from the sampled subgraph $G_s = (V_s, E_s)$?

The evaluation of the performance of the various sampling methods has focused so far on the ability to recover from the subgraph G_s the local statistics of the original graph, such as vertex degree distribution, average or global clustering coefficient, etc. We describe in the following these statistics.

2.1 Step 1: Global scale graph properties

The first step is used to check whether the large scale properties of the graph have changed.

Given a weighted graph $G = (V, E, \boldsymbol{w})$, you will compute the following statistics.

1. size = number of vertices = $n \overset{\text{def}}{=} |V|$
2. $m \overset{\text{def}}{=}$ number of edges
3. volume = sum of the weights along all the edges $\overset{\text{def}}{=} \sum_{e \in E} w(e)$
4. density = ratio of the number of edges over the maximum possible number of edges (given the number of nodes)

$$\text{density} \overset{\text{def}}{=} \frac{2m}{n(n-1)}. \tag{2}$$

2.2 Step 2: Local scale graph properties

The second step is used to check whether the local scale properties of the graph have changed.

Given a weighted graph $G = (V, E, \boldsymbol{w})$, you will compute the following statistics.

4. degree distribution

$$\deg(v) \overset{\text{def}}{=} \sum_{u:(u,v) \in E} w_{uv}. \tag{3}$$

   Because many degree distribution decay as $1/k^\gamma$, one should consider displaying the histogram using a log-log plot: the logarithm of the number of nodes with degree k should be displayed as a function of the logarithm of the degree, $\log k$.

5. average degree

$$\overline{d} \overset{\text{def}}{=} \frac{1}{n} \sum_{v \in V} \deg(v). \tag{4}$$

We define the neighborhood of a vertex v as the set of vertices that are directly connected to v by and edge. Please note that v is not in the neighborhood of v.

Definition 2 *Let $v \in V$. The neighborhood of v is defined by*

$$N(v) \overset{\text{def}}{=} \{w \in V \backslash \{v\} : (v, w) \in E\}. \tag{5}$$

6. clustering coefficient. This is the ratio of the actual number of triangles that include v as a node and for which the two other nodes are in the neighborhood of v, over the maximal possible number of such triangles (given the size of $N(v)$). The clustering coefficient at v can be computed as follows,

$$\operatorname{cc}(v) \stackrel{\text{def}}{=} \frac{2|\Delta(v)|}{\deg(v)\,(\deg(v)-1)} \tag{6}$$

where

$$\Delta(v) \stackrel{\text{def}}{=} \{(u,w) | u \in N(v), w \in N(v), (u,w) \in E\} \tag{7}$$

is the set of edges in the neighborhood of v, $N(v)$.

Because nodes tends to cluster in social networks (the friend of my friend is also my friend), the clustering coefficient can be used to quantify the "transitivity" of the connectivity.

7. average clustering coefficient

$$\overline{\operatorname{cc}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{v \in V} \operatorname{cc}(v). \tag{8}$$

2.3 Step 3: Advanced structural graph properties

The final step involves quantifying structural changes that happen at multiple scales. In Peter Wills' thesis you will find sophisticated distances that can be used to capture these multiscale changes created by sampling.

Assignment

3. For the 12 contact networks $\left(\boldsymbol{A}_f^{(i)}, \boldsymbol{A}_p^{(i)}\right), i = 1, \ldots, 6$, evaluate the 7 statistics described in sections 2.1 and 2.2.

When you need to compute a probability distribution (e.g., degree, clustering coefficient), please display an histogram. If the histogram (probability distribution) of a statistic (e.g., degree) appears to decay as a rational function of k, please try a log-log plot.

You can use a table to display all the other statistics, where you only need to compute a scalar.

4. For each of the six datasets, compare the statistics of the face-to-face networks, with those of the co-presence networks. Explain your findings.

3 Sampling of Large Graphs

3.1 The Naive Approaches

We first describe naive approaches that either lead to unreasonable models of computation (e.g., the adjacency matrix will never fit in memory), or have very poor properties (e.g., the sampled graph is disconnected).

3.1.1 Node sampling: induced subgraph sampling

The node sampling algorithm is described in Algorithm 2. A subset $V_s \subset V$ of vertices are selected independently according to a probability distribution $\boldsymbol{p}$: a node v is selected with a probability $p(v)$. Once k vertices have been selected, the edges of the sampled graph, E_s, are added by considering the induced subgraph: $e = (v, w) \in E_s$ if and only if $v \in V_s$ and $w \in V_s$ (see Fig. 1). The probability distribution $\boldsymbol{p}$ can be

- uniform: all vertices are equally likely;
- proportional to the degree distribution: nodes with high degree are more likely to be selected

A variation of the node sampling involves selecting all the neighbors of the subset V_s. The subset of edges remain the subgraph induced by the subset of nodes and their neighbors.

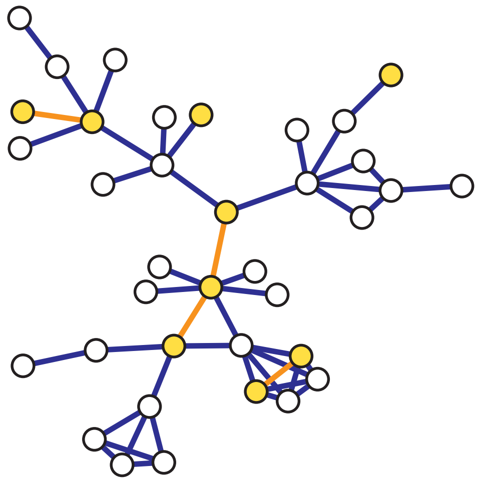

Figure 1: Sampled nodes are shown in yellow. Edges connecting the sampled nodes are in orange. Several nodes are isolated.

Algorithm 1 Induced Graph Sampling

```
1: procedure nodeSampling($G = (E, V), \boldsymbol{p}, k$)
   // Input: $E$, $k$, $\boldsymbol{p}$
   // Output: $V_s$ the reduced vertex set; $A_s$ the adjacency matrix of the reduced edge set
   // Return a subset of $k$ vertices $V_s$ chosen with probability distribution $\boldsymbol{p}$ from $V$,
   // and the induced set of edges

   // Selection of the vertices
2:     $V_s \leftarrow \emptyset$
3:     for $j \leftarrow 1, k$ do
4:         Select a vertex $v \in V$ at random according to the probability distribution $p(v)$
5:         $V_s \leftarrow V_s \cup \{v\}$
6:     end for

   // Selection of the edges
7:     for $i \leftarrow 1, k$ do
8:         for $j \leftarrow 1, k$ do
9:             $v \leftarrow V_s(i)$          // indices of the vertices in the original graph
10:            $w \leftarrow V_s(j)$
11:            $A_s(i, j) \leftarrow A(v, w)$ // the reduced adjacency matrix
12:        end for
13:    end for

14: end procedure
```

3.1.2 Edge sampling: incident subgraph sampling

The edge sampling strategy involves selecting a subset of edges E_s at random from the set of edges according to a probability distribution $\boldsymbol{p}$. The subset of vertices is formed by collecting all the endpoints of the subset of edges E_s (see Fig. 2). Algorithm 2 describes the edge sampling algorithm.

Edge sampling and node sampling can be combined in a manner whereby a vertex v is selected at random, and an edge $e = (v, w)$ incident to v is then selected at random.

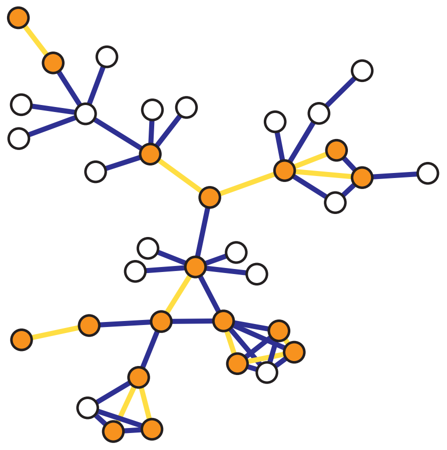

Figure 2: Sampled edges are shown in yellow. Vertices incident to the sampled edges are in orange. Several edges are isolated.

Algorithm 2 Edge Sampling

```
1: procedure edgeSampling($G = (E, V), \boldsymbol{p}, k$)
   // Input: $E$, $k$, $\boldsymbol{p}$
   // Output: $E_s$ the reduced edge set, $V_s$: the reduced vertex set
   // return a subset of $k$ edges from $E$ chosen with probability distribution $\boldsymbol{p}$
2:     $E_s \leftarrow \varnothing$
3:     $V_s \leftarrow \varnothing$
4:     for $j \leftarrow 1, k$ do
5:         Select an edge $e \in E$ at random according to the probability distribution $p(e)$
6:         $E_s \leftarrow E_s \cup \{e\}$
7:         $V_s \leftarrow V_s \cup \{e^+, e^-\}$
8:     end for

9: end procedure
```

3.2 Sampling by random exploration

In contrast to the previous naive approaches, these sampling strategies can yield more faithful replica of the original graph, using computationally efficient algorithms. We note that many of the algorithms require that the neighborhood of a given node be readily accessible (fast random access in the main memory, as opposed to random access on disk, which is much slower).

The central idea of these algorithms is to construct the reduced graph by assembling several connected paths. Each path is initiated at some random vertices in the graph, and the algorithm explores the graph using a combination of deterministic and stochastic strategies. We describe in the following sections several sampling strategies based on the random exploration of the graph.

3.2.1 Metropolis-Hastings Random Walk

The random walk algorithm starts at a randomly chosen vertex v_0. The next vertex v_1 is chosen at random according the probability

$$p(v_1) = \frac{A_{v_0 v_1}}{\deg(v_0)} \tag{9}$$

where the degree of v_0 is computed using the formula for weighted graphs (3),

$$\deg(v_0) \overset{\text{def}}{=} \sum_{v \in N(v_0)} A_{v_0 v}. \tag{10}$$

The simple random walk is inherently biased: nodes with high degrees will be visited more often. The following sampling strategy, called the Metropolis-Hastings Random Walk corrects this bias. The idea is to always accept the move towards a node of smaller degree, and reject some of the moves towards higher degree nodes. This eliminates the bias towards high degree nodes.

Algorithm 3 Metropolis-Hastings Random Walk

```
1:  procedure metropolisHastingsRW($G = (E, V), m$)
    // Input: $V, E, m_s$
    // Output: the reduced edge set $E_s$ of size $m_s$
2:      $S \leftarrow \varnothing$
3:      $v \leftarrow$ random $(V))$
4:      while $|S| \leqslant m_s$ do
5:          Select new node $w \in N(v)$ uniformly at random
            // choose p according to a uniform distribution on [0,1]
6:          $p \leftarrow U[0, 1]$
7:          if $p \leqslant \deg(v) / \deg(w)$ then
8:              $v \leftarrow w$
9:          else
10:             Stay at $v$
11:         end if
12:     end while
13: end procedure
```

3.2.2 Frontier Sampling

Frontier sampling randomly selects a list of m seed vertices. The vertices are selected by uniformly sampling the set of vertices V. The list of seed vertices is visited randomly, with a probability proportional to the degree of each vertex. The seed vertex is replaced by one of its neighbors at random, and the exploration continues with the new vertex. The algorithm stops the exploration once its has sampled m_s edges. Algorithms 4 describes the Frontier Sampling procedure.

Algorithm 4 Frontier Sampling

```
1: procedure frontierSampling($G = (E, V), m_s$)
       // Input: $E$, $m_s$
       // Output: $E_s$ the reduced edge set
       // return a subset of $m_s$ edges from $E$ chosen with probability distribution $\boldsymbol{p}$
2:     $k \leftarrow 0$          // $k$ is the number of iterations
3:     Initialize $L = \{v_1, \ldots, v_m\}$ with $m$ randomly chosen vertices (uniformly)
4:     repeat
5:         Select $u \in L$ with probability $p(u) = \deg(u) / \sum_{v \in L} \deg(v)$
6:         Select an outgoing edge of $(u, v)$ uniformly at random
7:         Replace $u$ with $v$ in $L$
8:         $E_s \leftarrow E_s \cup \{(u, v)\}$
9:         $V_s \leftarrow V_s \cup \{u, v\}$
10:        $k \leftarrow k + 1$
11:    until $k \leqslant m_s$
12: end procedure
```

3.2.3 Snowball Sampling

Snowball sampling randomly selects a set V_0 of vertices. The vertices are selected by uniformly sampling the set of vertices V. For each vertex v in V_0, we consider the neighborhood of v formed by the vertices connected to v. We extend this definition to any subset S of vertices.

Definition 3 *Let $S \subset V$. The neighborhood of S is defined by*

$$N(S) \overset{def}{=} \{w \in V \backslash S : \exists v \in S, (v, w) \in E\} . \tag{11}$$

The snowball sampling creates a series of waves. During the first wave, V_0 is extended to $V_1 \overset{\text{def}}{=} N(V_0) \cap (V \backslash V_0)$. During the second wave, V_1 is extended to $V_2 \overset{\text{def}}{=} N(V_1) \cap (V \backslash (V_0 \cup V_1))$ (see Fig. 3). The reduced graph is formed by collecting the vertices sampled during each wave, $V_s \overset{\text{def}}{=} V_0 \cup V_1 \cup \ldots \cup V_k$. The edges are the incident edges, created by the connection between the successive waves.

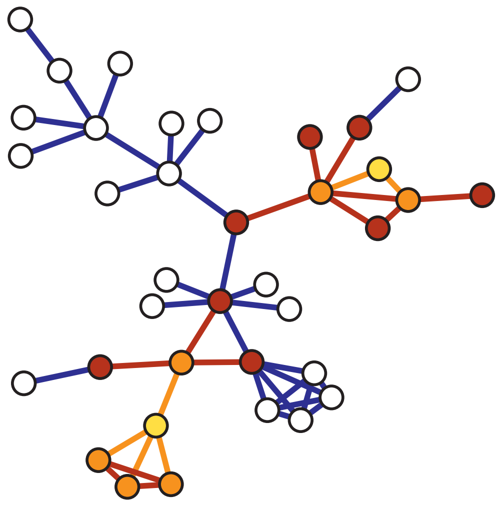
Figure 3: Snowball sampling. The initial set of vertices V_0 is shown in yellow. Vertices reaching to the second wave, V_1, are shown in orange. The second wave of nodes, V_1, is shown in orange as well. The third waves of nodes is shown in maroon.

3.2.4 Snowball Expansion Sampling

The Snowball Expansion sampler is a variation of the snowball sampler that maximizes the expansion of the graph between the successive waves.

Algorithm 5 Snowball Expansion Sampling

```
1: procedure snowBallExpansion($G = (E, V), m_s$)
   // Input: $V, E, m_s$
   // Output: $V_s$ the reduced vertex set
2:     $S \leftarrow \varnothing$
3:     $v = \text{random } (V))$
4:     $S \leftarrow S \cup \{v\}$
5:     while $|S| \leqslant m_s$ do
6:         Select new node $v \in N(S)$ based on maximization of $|N(\{v\}) \backslash (N(S) \cup S)|$
7:         $S \leftarrow S \cup \{v\}$
8:     end while
9: end procedure
```

Assignment
Implement the seven sampling algorithms: Algorithms 1-7.

3. For each sampling algorithm, and for the six contact networks $\boldsymbol{A}_p^{(i)}$, create five subsampled networks, using five sampling ratios, $f = 0.9, 0.8, 0.7, 0.6, 0.5$. The sampling ratio f measures the proportion of edges that remain in the sampled network, irrespective of their locations,

   $$f \overset{\text{def}}{=} \frac{|E_s|}{|E|}. \tag{12}$$

   For each sampled network, you will evaluate the 7 statistics described in sections 2.1 and 2.2.

   When you need to compute a probability distribution (e.g., degree, clustering coefficient), please display an histogram. If the histogram (probability distribution) of a statistic (e.g., degree) appears to decay as a rational function of k, please try a log-log plot.

   You can use a table to display all the other statistics, where you only need to compute a scalar.

4. For each sampling strategy, compare the seven statistics computed from the original face-to-face contact network $\boldsymbol{A}_f^{(i)}$ with those computed from the sampled co-presence network $\boldsymbol{A}_p^{(i)}$.

   Propose some simple strategies to remedy some of the biases introduced by the sampling algorithms. For instance, you may assume that you have access to global statistics: size, volume of the graph, etc. to correct for a bias in the degree distribution.

4 Spread of Epidemics

We consider the temporal network forming the substrate of the spreading process to be a sequence of undirected and unweighted static networks. This section needs to be written.