Source Identification in H1N1 Flu Infection

Bin Deng
Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bdeng@andrew.cmu.edu

Abstract

Background. H1N1 is a worldwide epidemic, and targeting early stage of human proteins involved will be effective in treatments due to high mutability of H1N1.
Aim. We aim to identify source proteins in H1N1 infection.
Data. Human protein-gene network containing protein-protein, post-transnational modification and protein-DNA interactions forms a weighed directed graph given transmission time of each edge. At different time points of H1N1 infection, expressions of part of gene were recorded and activation time of these gene can be obtained based on these records.
Method. Activation starts from source protein nodes and diffuses toward neighbors. Availability of activation time from part of gene nodes makes the problem become diffusion network with partially observed nodes. Starting with the shortest path tree from the original graph, we have developed two algorithms for source identification. The MLE with edge optimization performs iterative update of likelihood for each potential source node and identifies source based on likelihood ranking. The fastest diffusion method identifies source nodes which overall propagate activations towards observed nodes faster.
Results. Both methods are better than random guess, and the fastest diffusion method gains about 12 folds of source nodes hits compared to random guess.
Conclusions. Source proteins in H1N1 infection can be traced in the methods here with much better chance than random guess. The methods derived in the study can be applied not only to source prediction in H1N1 infection but also in other diseases.

1 Introduction

1.1 H1N1 infection pathways

H1N1 is a subtype of influenza A virus. Its name comes from the type of surface glycoproteins. H1N1 can infect many animal species besides human. The outbreak of H1N1 in 2009 caused about $17,000$ deaths within one year. Designing therapeutics against this disease become important in treatment and prevention. To achieve that, we need to understand biological pathways involved in the infection and identify some early proteins as target to design drugs. In this study, we aim to identified source proteins in H1N1 infection based on human protein-gene network and partially observe gene activation during the infection.

Before going through the biological pathways of H1N1 infection, first we will go over some important biological terms:

- Gene: a functional unit or a locus resides in DNA which encodes RNA and/or protein. There are about $20,000$ protein-coding genes in human [1, 2]. On the basis of genes, human system is constructed and various functions have been accomplished.
- Transcription: genes can not take action directly. Instead they exert their functions through mediator called RNA. Transcription is to synthesize RNA using gene in DNA as template under RNA polymerase and other proteins so that the information in gene can be transmitted.
- Protein-DNA interaction: transcription is usually strictly regulated. Certain family of proteins are able to bind to DNA and alter transcription rate of downstream genes.
- Translation: protein is synthesized by using RNA as template so the genetic information in gene is represented in protein. In most of cases, proteins are the major components to execute functions.
- Post-transnational modification: certain classes of proteins can be modified by other proteins so the activity or performance of these classes of proteins changes.

Hereby from gene which stores genetic information to protein which execute functions, there exist multiple cascades and each of them is regulated. Accordingly from a virus infection to cell response, multiple signal pathway activations take place. Several interactions in the cell as shown in Figure 1 are key for this signal passing. In protein-DNA (PD) interaction, a protein binds to a DNA segment, increasing or decreasing the downstream gene transcription level. This interaction can be mapped with a directed edge, meaning this protein targets this gene. In protein-protein (PP) interaction, two proteins bind together and alter performance of each other. This interaction can be mapped with bidirectional edges to indicate the change is mutual. In post-transnational modification (PTM), a protein modifies another protein and change its performance. The interaction can be mapped with a directed edge meaning the former targets the latter.

The knowledge above can be applied to model H1N1 flu infection. H1N1 is composed of viral proteins and RNA. Once H1N1 attaches a host cell, the viral protein will trigger cell response through a series of PP and PTM interactions. The effect will be eventually transduced to gene expressions level through PD interaction and induce cell stress and cell death pathways. Meanwhile viral RNA is injected into the host cell and is vital for virus proliferation. The host cell is also able to sense viral RNA and activate inflammation and immune response pathways as defense through PD, PP, and PTM interactions. Consequently components, proteins and genes, in these multiple pathways can be linked with directed edges, which depends on their relations to form a network.

1.2 Diffusion network and previous work

Networks are universal in modern society which enable information propagation and node communications. They are generally composed of nodes, which take responsibility of receiving, processing and sending messages, and edges which communicate nodes with each other. In such an network, an infection can start from one node and rapidly diffuse across the whole network. This arises a term called diffusion network.

Diffusion network deal with the problem of identifying the source nodes based on the infection pattern. This is important in many areas. For example, finding the source in a virus infected network will be helpful to reduce the influence and prevent infections in future. In addition, we often face challenges to find rumor or disease source from a social or human network. In this study, we will focus on gene and protein regulation network. This can be also modeled under the framework of diffusion network, in which a gene or a protein is a node and their interactions can be represented with edges. With regulations, a gene or a protein is either activated or inhibited by upstream protein.

Figure 2 is an simple example of diffusion network. The infection time for each node is indicated in parenthesis. Each number besides edge is transmission time from one parent node to its child. At time 0, infection starts from node A, diffuses across the network and reaches node E at time 8. This diffusion follows a shortest path property. For example, from node A to node D, there are three paths: $A-B-D$, $A-D$ and $A-C-D$. Apparently $A-B-D$ takes the shortest path so the actual infection path, shaded in red, follow this path. Similarly from node A to node E, $A-E$ is the shortest path.

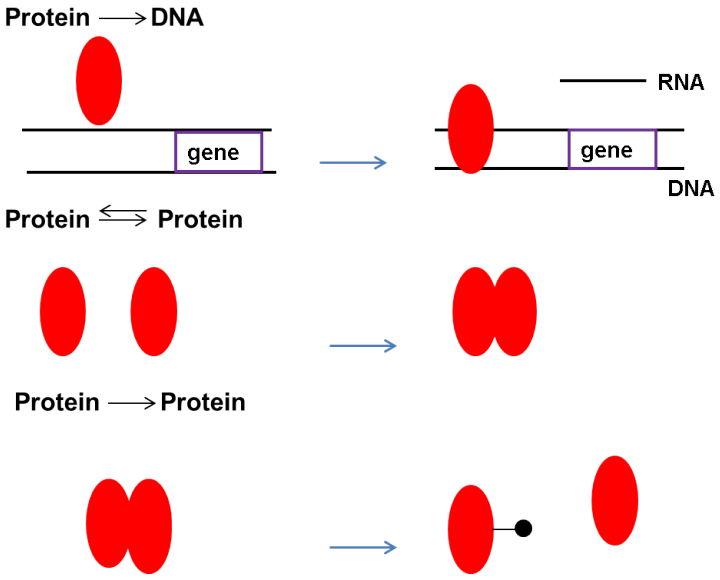

Figure 1: Interactions in biological pathways. Protein-DNA: protein binds to DNA and affects the downstream gene expression. In this example, more RNA is produced. This interaction can be modeled as a directed edge with protein pointing to gene. Protein-Protein: two proteins binds together to activate/inhibit each other. This interaction can be modeled as bi-directional edges. Protein-Protein or post-transnational modification: one protein modified another protein and affect its function. This interaction can be modeled as a directed edge with protein (exert modification) pointing to protein (modified).

Regarding models in diffusion networks, there are Susceptible-Infected(SI) and Independent-Cascade (IC) models. SI is the most popular model. Under SI, a node stays infected and is able to infect multiple nodes. The infection transmission time between nodes are independent with each other. In contrast, under IC, a node will recover to normal state once it infects its child and one node can infect only one child node once. In this study, we will focus on H1N1 infection pathway. Since one gene or protein get activated, it will affect multiple downstream genes or proteins and the effect will last for sometime. Therefore SI model will fit the case better and all future discussions will be under this model.

Multiple efforts have been made to study the problem of diffusion network. Kempe et al. used the IC model to find the source node [3]. However, due to the limitation of IC model, it is hard to evaluate performance in the real world and apply the method to the real data. Lappas et al.[4] and Du et al. [5] employed self-defined influence functions in searching source based on node influence. Feizi et al. propose a diffusion kernel method which essentially converts sum of exponential distributions to gamma distribution to obtain maximal likelihood estimate (MLE) [6]. This method also allows them to incorporate observed infection time into calculation. However, in these studies, they all assumed that infection states for all nodes were known, which may not be applied to some circumstances. Zhang et al. used a reverse propagation and a clustering technique to identified source nodes with partially observed infection states [7]. The method is good at dealing with data in which lots of hidden nodes exist, but it required predetermined infection probability and can not incorporate information from infection time.

Recently Pinto et al. assumed that only small portion of nodes were observed and their infection time is known when searching the source [8]. This is close to lots of real-world data, but the method requires large distance between these observed nodes. More recently Farajtabar et al relied on importance sampling to get rid of hidden nodes and find source by ranking likelihood under each potential source nodes [9]. This is so far the optimal method to deal with data containing large percentages of hidden nodes and a small fraction of observed nodes with infection time known.

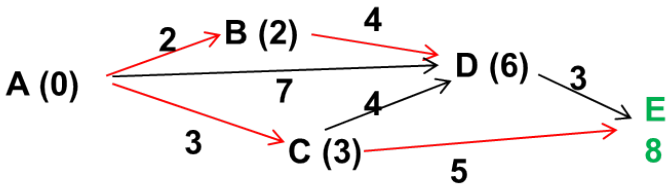

Figure 2: Infection is propagated in a diffusion network. Each number beside an edge is transmission time took from a parent to its child. Nodes are labeled with letters and numbers in parenthesis are infection time. Node E in green is an observed node with infection time known, where other nodes in black are hidden node with unknown infection time (we added time here to make it clearer, but in real data, this information is missing). The actual infection diffusion path is represented with red arrows. The diffusion pattern follows the shortest path property which allows the infection propagated rapidly.

Although big progress has been made in searching source from a diffusion network, finding exact diffusion path or source is a NP-hard problem [10]. There are several challenges:

- Large number of nodes and edges: in a real graph, there may exist many nodes which have multiple children. As an infection diffuses, the possible paths may increase exponentially.
- Data with partially observed node: in a dataset we do not have any information (infected or not; infection time) for majority of nodes.
- Infection transmission time from one node to another is unknown: in the case of [9], edge distributions were know which allows them to model transmission time. Under some cases, we do not know the edge distributions.

So far we summarized the previous work in diffusion network as well as challenge in identifying the source. When it moves to an biological network such as pathways in H1N1 infection, the edge (transmission time) distributions is generally unknown and hard to determine, which makes the problem even challenging.

2 Problem and approach

In this study, we will identify source nodes in H1N1 infection. Concretely we aim to find source proteins that activate immune response pathways in human. While this can be done experimentally for some viruses, it is hard to do for several others. If we can recover these in a computational manner, it would be very useful. During H1N1 infection, the components and relations involved in the infection can be considered as a diffusion network with activation propagated from a parent protein to its children. In this network, each protein or gene can be considered as a node, while interaction between them can be modeled by edges as shown in Figure 1. On the basis of biological experiments, expression levels of part of genes can be measured in multiple time points so we will know when each gene get activated. The model here is analogous to diffusion network with partial observed nodes, which can be fit into the frame of maximal likelihood estimate with important sampling [9].

Hereby, we will first introduce the relevant method of MLE with importance sampling by Farajtabar et al.[9]. Then we will derive and implement edge optimization to obtain MLE and to find source nodes. It represents the algorithm of coordinate accent. We next will implement a simple algorithm which relies on shortest path property. This method more emphasizes the structure of network when we can not obtain the accurate edge distributions.

Component 1	Interacton	Component 2	Confidence
MDM2	PTM	TP53	1.0
TP53	PP	UBC	1.0
SP4	PD	FGFRL1	0.3

Table 1: Example of interactions from dataset

14451 (TLK1)	0.25 h	0.5 h	1 h	1.5 h	2 h	4 h	6 h	8 h	12 h	18 h
Expression	0.70	0.16	-0.13	0.10	-0.03	0.17	0.20	-0.16	-0.07	-0.24

Table 2: Example of human gene expression during H1N1 infection

3 Dataset

There are two parts in the whole dataset. The first part contains interaction information which allow us to construct the weighted graph. It has PP, PTM and PD interactions, which corresponds bidirectional, directed and directed edges, respectively. The information was collected from multiple data sources including BioGRID (PP interactions) [11], Protein Reference Database (PP and PTM interactions) [12] and TF-gene binding predictions (PD interactions) [13]. Interactions in each database are identified with one or more biological techniques such as binding assays, mass spectra, crystallization. The summarized interaction data are available at `https://docs.google.com/file/d/0BwvqEovxDE4gS2ZBaU1wNUJnS00/edit?usp=drive_web`. Table 1 shows the format of the dataset. Column 1 and 3 are protein or gene in interactions. Column 2 are interaction type. Column 4 is the confidence for this interaction. For example, in the second row, TP53 has protein-protein iteration with UBC. So we can put bidirectional edges between TB53 and UBC. Since some PP interaction is not very confident, in data process, we removed PP interactions with confidence less than 0.5. All the genes and proteins which participate the network will be included in the dataset. However, lots of them have aliases and different data source may use different names. We got all the possible aliases from NCBI (`http://www.ncbi.nlm.nih.gov/gene`) and indexed proteins or genes so different name under the same protein or gene have one index.

The second part of dataset contains 10 time points (0.25h, 0.5h, 1h, 1.5h, 2h, 4h, 6h, 8h, 12h, 18h) of gene expressions for more than $13,000$ genes during H1N1 infection [14]. The data are available at `https://docs.google.com/file/d/0BwvqEovxDE4gS2ZBaU1wNUJnS00/edit?usp=drive_web`. They are essentially relative RNA level of each gene compared to 0h (right before the H1N1 infection). We consider activation of a gene when its expression level increase or decrease by 0.5. Table 2 shows an example of TLK1 (14451) expression. $0.25h$ is considered as time of activation since the change is above 0.5. It is worth noting that we define activation by changing not just increasing. We check each gene to record the earliest time to get activated. Thus after this process, we will get nodes (gene) and their activation time. Again all the genes are indexed to remove ambiguity from aliases.

4 Methods

In this part, we will first illustrate how to build the weighted graph. Then we will explore the method of MLE with importance sampling by Farajtabar et al.[9]. Next we will introduce our algorithm which employs edge-wise optimization. Finally we will introduce a simple algorithm which is based on the shortest path ranking.

4.1 Weighted graph building

To incorporate observed activation time, we need to obtain the transmission time between nodes. Transmission time or edge weight τ_{ij} is defined as difference between activation time of node i and activation time of node j, where node i has a directed edge to node j. τ_{ij} is positive and allows us

Index 1	Interacton	Index 2	Transmission time (h)
8262	PTM	14831	0.029
14831	PP	15312	0.015
13510	PD	5075	0.074

Table 3: Example of weighted edges for building graph

Node number	15440
Edge number	273786
PP edges	215464
PTM edges	2458
PD edges	55864

Table 4: Statistics of weighted graph built from human protein-gene network

to model transmission delay. There are three types of interactions. The transmission time of them can be modeled as:

- PP and PTM: These types of interactions are usually short and transient. The transmission time are generally not available for these interactions. We put exponential distribution for transmission time $\tau_{ij} \sim exp(\lambda)$. This implies once parent node gets activated, child node will be activated within short time with high probability. We further assume the transmission time for each edge is independently identically distributed. In other word, transmission time for one edge does not affect the time for other edges. Given the distribution, we can obtain transmission time or edge weight with sampling.
- PD: For a protein-gene interaction, the target gene is considered to be activated when its RNA is transcripted. Human gene lengths are available in NCBI (`http://www.ncbi.nlm.nih.gov/gene`). In addition, gene transcription rate almost remains constant in human and rate has been determined as $3.8kb/min$ [15, 16]. So we can obtain transmission time by simple divisions.

We now able to constructed graph with weighted edges. Table 3 gives some examples. For example, in the second row, 14831 interacts 15312 with protein-protein interaction. We added bidirectional edges between them and put weight 0.015 as transmission time. Consequently we built a directed graph with 15440 nodes and 273786 edges (bidirectional edges are counted as two). The statistics of graphs are shown in Table 4. For each node, we store both its children and parents to accelerate search.

4.2 Shortest path property

Activation from source to downstream nodes follows the shortest path property. The path from the source to one node will take the smallest sum of edge weights. In other word, source tends to activate the downstream as soon as possible. In Figure 1, the actual diffusion path shaded in red is the shortest path. Not only simplifies the diffusion path, the shortest path property also allow us to assign activation time for those hidden nodes. In the graph, one node may have multiple parent nodes. The shortest path property considers the earliest parent node as true parent and assigns the smallest activation time for it. It is worth noting that activation time of observed nodes are known from the dataset so we does not need to assign time for them. One important outcome for this time assigning is that it allow to calculate likelihood function given edge distributions.

The shortest path tree can be obtained with Dijkstra' algorithm in $O(|E| + |V|log|V|)$ [17], where $|E|$ and $|V|$ denote edge number and node number, respectively. Briefly, the algorithm starts from source node and update distances of its children based on edge weights. Next the closest node will be check and distance of its children will be updated if the sum of weights now is smaller than their previous values. Then the these nodes will add to the previous nodes for distance comparison and the second closest node will be checked to update its child nodes. As the algorithm goes on, dis-

tance of all nodes will be updated with the smallest sum of edge weights. Typically a min priority queue is generated to obtain the node with the smallest distance every time. At the same time, the parent leading to this shortest distance will be maintained. The final outcome of this algorithm is a tree structure with the shortest distance from source to every nodes. Notably, the original graph may contains many cycles which exclude factorization of joint distribution. After running Dijkstra's algorithm, the final tree structure will allow us to factorize the joint likelihood to a series of multiplications of conditional probabilities so that we can plug in each edge distribution.

4.3 MLE with Importance sampling

The method proposed by Farajtabar et al.[9] represents the idea of removing hidden nodes with sampling. Let O denote a set of observed activated nodes and t_i denote the activation time of observed nodes; let S denote a set of source nodes; let H denote a set of unobserved nodes and t_j denote the activation time of unobserved nodes. The likelihood function can be written as:

$$L(S) = p(\{t_i\}_{i\in O}|S) = \int_H p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S)d\{t_j\}_{j\in H} \tag{1}$$

Since H contains large numbers of nodes (more than $12,000$ of proteins and genes lack information on whether they are activated or not during H1N1 infection), the integration is in high dimension and is intractable for this large graph. For this reason, they introduced auxiliary variable $a_i, i \in O$ and a proposal distribution $q(\{a_i\}_{i\in O}, \{t_j\}_{j\in H})$. $a_i, i \in O$ represents activation time of observed nodes from sampling. Thus for observed nodes, there are two kinds of time: $a_i, i \in O$ is assigned activation time from sampling; $t_i, i \in O$ is true activation time from experimental records. Therefore:

$$\begin{aligned}
p(\{t_i\}_{i\in O}|S) &= \int_H p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S)d\{t_j\}_{j\in H} \\
&= \int p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S)q(\{a_i\}_{i\in O})d\{a_i\}_{i\in O}d\{t_j\}_{j\in H} \\
&= \int \frac{p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S)q(\{a_i\}_{i\in O})}{q(\{a_i\}_{i\in O}, \{t_j\}_{j\in H})} q(\{a_i\}_{i\in O}, \{t_j\}_{j\in H})d\{a_i\}_{i\in O}d\{t_j\}_{j\in H} \\
&\approx \frac{1}{L}\sum_l^L \frac{p(\{t_i\}_{i\in O}, \{t_j^l\}_{j\in H}|S)q(\{a_i^l\}_{i\in O})}{q(\{a_i^l\}_{i\in O}, \{t_j^l\}_{j\in H})}
\end{aligned} \tag{2}$$

where L denotes the number of sampling time and l is the $l\ th$ sampling. In the previous step, they converted a high dimension integration to sum of a series of multiplications. Choose $q(\{a_i^l\}_{i\in O}) = p(\{a_i^l\}_{i\in O}|\{t_j^l\}_{j\in H})$ and factorize the joint distributions:

$$\begin{aligned}
&p(\{t_i\}_{i\in O}|S) \\
&= \frac{1}{L}\sum \frac{p(\{t_i\}_{i\in O}, \{t_j^l\}_{j\in H})p(\{a_i^l\}_{i\in O}|\{t_j^l\}_{j\in H})}{q(\{a_i^l\}_{i\in O}, \{t_j^l\}_{j\in H}))} \\
&= \frac{1}{L}\sum \frac{\prod_{i\in O} p(t_i|par(i)) \prod_{j\in H} p(t_j^l|par(j))p(\{a_i^l\}_{i\in O}|\{t_j^l\}_{j\in H})}{\prod_{i\in O} p(a_i^l|par(i)) \prod_{j\in H} p(t_j^l|par(j))} \\
&= \frac{1}{L}\sum \frac{\prod_{i\in O} p(t_i|par(i)) \prod_{j\in H} p(t_j^l|par(j)\in O) \prod_{j\in H} p(t_j^l|par(j)\in H)p(\{a_i^l\}_{i\in O}|\{t_j^l\}_{j\in H})}{\prod_{i\in O} p(a_i^l|par(i)) \prod_{j\in H} p(t_j^l|par(j)\in O) \prod_{j\in H} p(t_j^l|par(j)\in H)} \\
&= \frac{1}{L}\sum \frac{\prod_{i\in O} p(t_i|par(i)) \prod_{j\in H} p(t_j^l|par(j)\in O)p(\{a_i^l\}_{i\in O}|\{t_j^l\}_{j\in H})}{\prod_{i\in O} p(a_i^l|par(i)) \prod_{j\in H} p(t_j^l|par(j)\in O)} \\
&= \frac{1}{L}\sum \prod_{i\in O} p(t_i|par(i)) \frac{\prod_{j\in H} p(t_j^l|t_i, i\in par(j), i\in O)}{\prod_{j\in H} p(t_j^l|a_i^l, i\in par(j), i\in O)}
\end{aligned} \tag{3}$$

where $par(i)$ denotes the parent of the node i. In the last step, $p(t_j^l|t_i^l, i \in par(j), i \in H)$ is equal in both nominator and denominator, and get canceled. Thus only observed nodes as parents are left.

1.Input: $\{t_i\}_{i\in O}$ and a graph with distribution of each edge known 2.Randomly sample transmission time of each edge for L times 3. Under each source node, for the $l\ th$ sample, start t_S with 0 and assigned $t_j^l, j \in par(i)$ for each $i \in O$ with shortest path tree; also assign a_i^l 4. Under each source node, compute the likelihood function with equation 3; choose t_S to maximize the likelihood function 5. Search source node with equation 5

Table 5: Algorithm of MLE with importance sampling

Next each conditional probability can be plugged in with exponential distribution $\lambda exp\{-\lambda(t_i - t_{par(i)})\}$, where $t_i - t_{par(i)}$ is transmission time between node i and its parent $par(i)$:

$$\begin{aligned} &p(\{t_i\}_{i\in O}|S) \\ &\quad= \frac{1}{L}\sum\prod_{i\in O} p(t_i|par(i))\frac{\prod_{j\in H} p(t_j^l|t_i, i\in par(j), i\in O)}{\prod_{j\in H} p(t_j^l|a_i^l, i\in par(j), i\in O)} \\ &\quad= \frac{1}{L}\sum\prod_{i\in O} \lambda exp\{-\lambda(t_i - t_{par(i)})\}\frac{\prod_{j\in H}\lambda exp\{-\lambda(t_j^l - t_i)\}_{i\in par(j), i\in O}}{\prod_{j\in H}\lambda exp\{-\lambda(t_j^l - a_i)\}_{i\in par(j), i\in O}} \end{aligned} \tag{4}$$

The source node can be found with the following equation:

$$S^* = \underset{S\in H}{argmax}\ \underset{t_S}{argmax}\ p(\{t_i\}_{i\in O}|t_S) \tag{5}$$

With the importance sampling, they take sampling of each edge, start t_S(source node activation time) with 0, and obtain parents activation time of these observed nodes with the shortest path tree. Then they used equation 3 to calculate the likelihood function. Next for a given potential source node they chose t_S to maximize the likelihood function for each potential source node. Finally they ranked likelihood values among all potential source nodes and identified the the source node, as demonstrated in 5. The algorithms is summarized in Table 5.

4.4 MLE with edge-wise optimization

Previous method introduced a proposal distribution and employed importance sampling to get rid of hidden variables during maximal likelihood estimation. Instead of removing hidden variables our method keeps them and employs coordinate ascent to reach maximal likelihood under each source. Under the frame, the joint distribution is:

$$\begin{aligned} p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S) &= \prod_{i\in O} p(\{t_i\}|par(i), S)\prod_{j\in H} p(\{t_j\}|par(j), S) \\ &= \prod_{i\in O}\lambda exp\{-\lambda(t_i - t_{par(i)})\}\prod_{j\in H}\lambda exp\{-\lambda(t_j - t_{par(j)})\} \end{aligned} \tag{6}$$

It is worthy noting that the original graph is not necessarily acyclic. However, in the shortest path tree we built with Dijkstra's algorithm, the joint probability can be factorized as shown in equation 6. Under each source node, we adjusted transmission time for each feasible edge to maximize the likelihood:

$$\begin{aligned} &\max_{\{\tau_i\}} p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S) \\ &\qquad= \max_{\{\tau_i\}} \prod_{i\in O}\lambda exp\{-\lambda(t_i - t_{par(i)})\}\prod_{j\in H}\lambda exp\{-\lambda(t_j - t_{par(j)})\} \end{aligned} \tag{7}$$

where τ_i denotes the transmission time or edge weight from the i th node's parent to the i th node. The edge number under the tree frame is equivalent to node number and is much smaller than in original graph. To find source node, we first maximize the likelihood value for each potential source

nodes and then rank these potential source nodes by likelihood value. The optimization problem can be written as:

$$\begin{aligned} S^* =& \underset{S\in H}{argmax}\ \underset{\tau\in E}{argmax}\ p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S) \\ =& \underset{S\in H}{argmax}\ \underset{\{\tau_i\}}{argmax} \prod_{i\in O} \lambda exp\{-\lambda(t_i - t_{par(i)})\} \prod_{j\in H} \lambda exp\{-\lambda(t_j - t_{par(j)})\} \end{aligned} \tag{8}$$

To perform edge-wise optimization efficiently, we can split likelihood function into two parts:

$$\begin{aligned} p(\{t_i\}_{i\in O}, \{t_j\}_{j\in H}|S) &= \prod_{i\in O} \lambda exp\{-\lambda(t_i - t_{par(i)})\} \prod_{j\in H} \lambda exp\{-\lambda(t_j - t_{par(j)})\} \\ &= K\lambda exp(-\lambda\tau_k) \prod_{i\in D(\tau_k)} \lambda exp\{-\lambda(t_i - (b_i + \tau_k))\} \end{aligned} \tag{9}$$

τ_k is the k th edge we are optimizing. b_i is the sum of time from source to parent of the i th observed node but without τ_k. Therefore $t_{par(i)} = b_i + \tau_k$. This expression allow us to write likelihood as function of τ_k. $D(\tau_k)$ denotes observed nodes have one-level downstream of edge τ_k. Here one level means first observed node in each branch when going downstream of edge τ_k. When optimizing edge τ_k, its downstream time assign of hidden nodes change until the first observed node appears. The edges between these downstream hidden nodes remain the same and thus do not affect likelihood. The edges between the one level observed nodes and their hidden parent nodes change because time assign for hidden nodes is shifted and time for observed nodes is always fixed. So in equation 9, we only consider two parts: one is the edge optimized; one is the edges between one-level downstream nodes and their hidden parents. K is a constant which include products from hidden nodes and non-downstream observed nodes. If transmission time between two nodes becomes 0, the exponential probability density function (PDF) will become maximal. When we push transmission time for one edge to zero, other edges may become negative. For that we include a penalty option for each exponential PDF:

$$p(t_i|par(i), S) = \begin{cases} \lambda exp\{-\lambda(t_i - t_{par(i)})\} & t_i \geq t_{par(i)} \\ \lambda exp\{-\lambda\epsilon\} & t_i < t_{par(i)}, \epsilon = \frac{3+log\lambda}{\lambda} \end{cases}$$

If transmission time is positive, we use the regular form in the first case. If transmission time is negative, we will use penalty term in the second case. ϵ is fixed so it can not be optimized to 0. Similarly if a potential source can not be connected to an observed node, we also use this penalty term. For each edge optimization, we need to record the critical points defined as the time of parent node equals time of child node. If passing the critical point, we switched single PDF to penalty term. As long as τ_k moves across all the critical points, we will able to find the arg max. When optimizing edges under one potential source node, we do not need to calculate likelihood repeatedly. Instead we record Δt. Based on equation 9, we can multiply the original likelihood with $exp(-\lambda\Delta t)exp(n\lambda\Delta t)$, where n is the size of set $D(\tau_k)$.

Under the algorithm of edge-wise optimization. First we randomly initialize PP and PTM edges with exponential distribution, and use determined time for PD edges. Next we select a hidden node as potential source to generate the shortest path tree. Then we perform edge-wise optimization as shown in Figure 3. When A-B edge is optimized, other edges between hidden nodes remain the same. The effect passes within one level range so E-F and G-H edges get shrink or stretched. We only need to modified the likelihood with the change of these three edges. For I-J edge, node F is observed and fixed, so there is no change. For A-C-D, they are not downstream of A-B, so there is no change. The edge-wise optimization will iteratively go from the closest feasible edges to further. Here feasible edges include:

- PP or PTM interaction between two hidden nodes
- PP or PTM interaction between an observed parent node and a hidden child node

For PD edge, transmission time is determined. For an edge formed by two observed nodes, it is also fixed. For an edge between a hidden parent node and an observed child node, we consider the time of the hidden node is already fixed during previous optimization so the edge between it and its observed child node is fixed. Once likelihood under one potential source node is optimized, we will move to next potential source node and repeat the procedure above. We rank the nodes based on their likelihood values and identified the source nodes. The algorithm is summarized in Table 6.

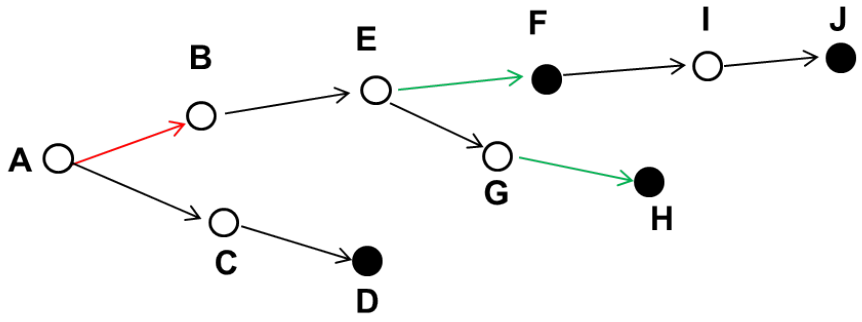

Figure 3: Edge-wise optimization. Hidden nodes are represented with circles and observed nodes are represented with spots. Red edge between A and B is the one we are optimizing. Green edges between E and F, and between G and H are within one level range and are affected during this optimization. Since time of E and H is fix, the effect will not be passed to downstream nodes.

1.Input: $\{t_i\}_{i\in O}$ and a graph with edge distributions
2.Randomly initialize transmission time of each edge based on its distribution
3. Under each source node, start t_S with 0;
assigned $t_j, j \in H$ for each hidden node with shortest path tree
4. Under each source node, compute the likelihood function with equation 6;
5. For the current tree, start with the feasible edge closest to source,
change the transmission time across critical points to maximize the likelihood
6. Proceed to next closest feasible edge and maximize the likelihood as 5;
repeat until all feasible edges or less are optimized
7. Back to 3 and start with a new potential source to calculate likelihood
8. Rank the potential source nodes based on their likelihood and find source nodes with equation 8

Table 6: Algorithm of MLE with edge-wise optimization

4.5 Fastest diffusion

Both of the previous two methods maximized likelihood (in different ways) to find source nodes. They highly relied on edge distributions. However, in current dataset, the distributions of PP edge and PTM edge are essentially unknown. Can we identify source node without MLE? In this part, we will investigate it.

We assume that a true source node will propagate the activation to majority of observed nodes overall in the shortest time. However, if time of each potential source node starts at 0, given fixed time of observed nodes, we can not differentiate each potential source. Instead we compare difference between an observed node and its parent. If a node has multiple parents, we consider the node with the earliest time as true parent. For that, we defined a score for each observed node i as:

$$score(i) = \begin{cases} \lambda(t_i - t_{par(i)}) & t_i \geq t_{par(i)} \\ 0 & t_i < t_{par(i)} \end{cases} \tag{10}$$

An observed node will have a high score if its true parent get activated early. An observed node will have 0 score if its parent is activated later or there no path from potential source to this node. This part is considered as penalty. The total score under a potential source node S is defined as:

$$score(S) = \sum_{i \in O} score(i) \tag{11}$$

This means summing scores over all observed nodes. Intuitively a potential source which delivers activation to all of the observed nodes' parents faster will have higher score.

Under the fastest diffusion algorithm, transmission time for PP and PTM edges will be randomly initialized with exponential distribution, while determined time is used for PD edges. Next under one potential source, shortest path tree will be generated to assign time for each observed node' parent. Then total score for this potential source can be calculated with equation 11. Then we move to next potential source node to calculate the total score. Next we go back to random generation transmission time and repeat the procedure multiple times to get accumulation of total score for each potential source. By ranking them, we are able to identified the source nodes. This method deemphasizes edge distributions and focus more on the shortest path property as well as allow the incorporation of activation time.

5 Results

One important question is how we verified the source nodes identified under each method. Fortunately the authentic source proteins have been identified based on multiple RNAi experiments [14, 18, 19, 20, 21]. RNAi is a useful techniques which knocks down RNA for a gene and decreases its protein level. This will block the pathway and allow us to evaluate the role of this protein in the pathway. Proteins located upstream tend to have more global effects. In this way, 204 source proteins, as summarized in [23], have been identified and can serve to verify our methods.

5.1 MLE with Edge-wise optimization

For each potential source node, we generated the shortest path tree. Start from the feasible edge closest to the potential source, we optimized the edge as described in Method. In Figure 4, we plotted log-likelihood increase for a true source node and a non-source node. Both optimizations are based on the same weighed graph. Figure 4 shows as the iterative steps go over all the feasible edges, the log-likelihood is non-decreasing. This suggests our algorithm is efficient in maximizing likelihood. Notably the number of edges optimized is much less than in the shortest path tree. This is because our algorithms will stop searching the branch if there is no observed node downstream. In addition, if repeating the optimization several times with random initialization of PP and PTM edges, the log-likelihood will generally converge to the values shown in Figure 4.

Next we investigate the accuracy of source node identification under different λ, the parameter in exponential distribution. Remember we sampled PP and PTM edges from exponential distribution

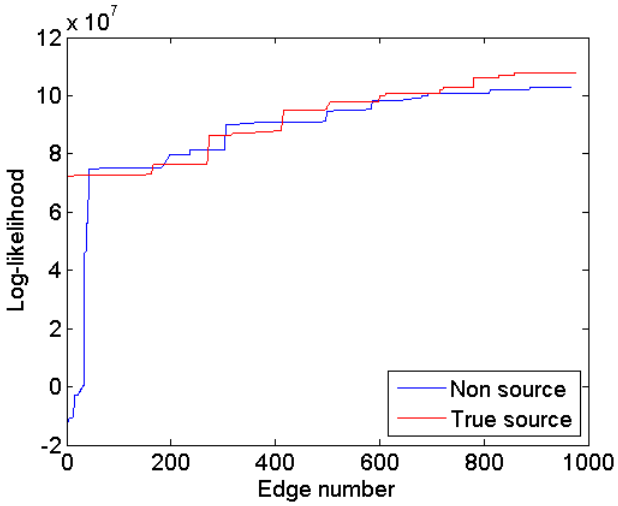

Figure 4: Edge-wise optimization. Edges in the shortest path tree by a potential source node are optimized iteratively. Here we include a non-source node and a true source node for comparison. Under both cases, log-likelihood increases as the optimization proceed to all the feasible edges.

λ	0.1	1	10	100	1000	10000
top 204 hits	0.0190	0.0190	0.0333	0.0095	0.0143	0.0095

Table 7: Percentage of hits with complete optimization of edges

and we plug in this distribution to calculate likelihood under each potential source node. Due to lacking exact information for transmission time, we used the same exponential distribution for every of these edges. We define accuracy by the percentage of hits compared with true source node data. That is, we extract the top 204 nodes from log-likelihood ranking and calculate what percentage of them match with true source nodes. Table 7 shows with all the feasible edges optimized, when $\lambda = 10$, we can get about 3.3% of hits in true source data, higher than the random guess of 204 nodes from 15440, which is about 1.3%.

Next we investigate the accuracy when we only optimize the first 10 feasible edges. This is to test if we are able to stop in the middle and save some iterations. Table 8 shows in term of top 204 source node matching, under $\lambda = 10$, we obtain 3.3% hits, which is equivalent to the case with complete optimization (Table 7). We also include top 100 hits and top 50 hits, which use only first 100 and 50 in the likelihood rank to match source nodes. For top 100 hits generally give higher percentage than top 204 hits, while top 50 hits have larger variations.

λ	0.1	1	10	100	1000	10000
top 204 hits	0.0143	0.0048	0.0333	0.0095	0.0238	0.0095
top 100 hits	0.0200	0.0100	0.0500	0.0100	0.0200	0.0200
top 50 hits	0	0	0.0600	0.0200	0	0

Table 8: Percentage of hits with first 10 edges optimized

Method	Top 204 hits	p value	Run time per iteration
Random guess	0.013 ± 0.006		
MLE with edge optimization	0.027 ± 0.008	0.018	622s
Fastest diffusion	0.163 ± 0.005	$1.32e^{-10}$	229s

Table 9: Summary of performance by different methods. For hits, each entry represents mean $\pm$ stdev from 5 independent experiments. One-way ANOVA was performed to calculate p values. All are compared with random guess.

5.2 Fastest diffusion

Next we investigate the method of fastest diffusion. This method put less emphasis on edge distribution which is essentially unknown. Under the frame of fastest diffusion, nodes will be ranked by a total score which is relevant to time differences between observed nodes and their parents. The algorithm accumulates the total score under each potential source node with multiple random initialization of edges (remember that we need to sample transmission time for PP and PTM edges).

Figure 5 shows percentages of hits under different λ. Here the percentage of hits is defined by matching top 204 nodes ranked by likelihood with the true source nodes. When $\lambda = 0.1, 1$, the percentage of hits stays low and does not respond to accumulation of total score. However, when $\lambda \geq 10$, percentage of hits increases as accumulation of total score from more random initializations. Under $\lambda = 10, 100, 1000, 10000$, percentage of hits generally reaches about 15%, much higher than the random guess (about 1.3%).

We summarize performance of source node prediction for each method in Table 9. Under random guess, we randomly drew 204 nodes from total 15440 nodes and match them with true source. The mean value is consistent with the expected 1.3%. For MLE with edge optimization, we report result based on $\lambda = 10$. For fastest diffusion, we report result based on $\lambda = 10000$. Compared to random result, edge optimization achieved 2 fold of hits with $p < 0.05$. Furthermore, fastest diffusion achieves 12 fold of hits compared to random with $p = 1.32e^{-10}$.

As for run time, the fastest diffusion method takes about $229s$ per iteration, faster than MLE with edge optimization, which is $662s$ per iteration. Here iteration means number of random initialization of edges. In an iteration, the fastest diffusion method only runs Dijkstra's algorithm for every potential source nodes while MLE with edge optimization will perform edge-wise optimization after Dijkstra. However, for fastest diffusion method, it will take about 20 iterations to get an optimal value while more iterations for MLE with edge optimization is not helpful. Thus total run time for the fastest diffusion is about $76min$.

6 Discussions

H1N1 is known to be highly infectious and highly mutable. H1N1 is able to escape from virus-target vaccine or drug once new generation come out [22]. In contrast, the pathways inside human cells are more conserved. Identification of mediators during infection will provide useful guidance for designing effective drugs. Gitter et al.[23] developed a model called SDREM to predict key mediators in H1N1 infection pathways. The protein-gene network constructed are originate from human.

In this study, we aim to find source proteins during H1N1 infection based on human protein-gene network and activation time of part of genes during infection. We first introduce a relevant method which employ importance sampling in MLE by [9]. Then we developed the method which performs edge-wise optimization and later a simple method which mostly relies on the shortest path property.

MLE with edge optimization is generally a sophisticated method and try to find source to lead certain node activation pattern with MLE. The method performs better than random guess but not as good as the simple fastest diffusion method. This may be because the current dataset is insufficient for this

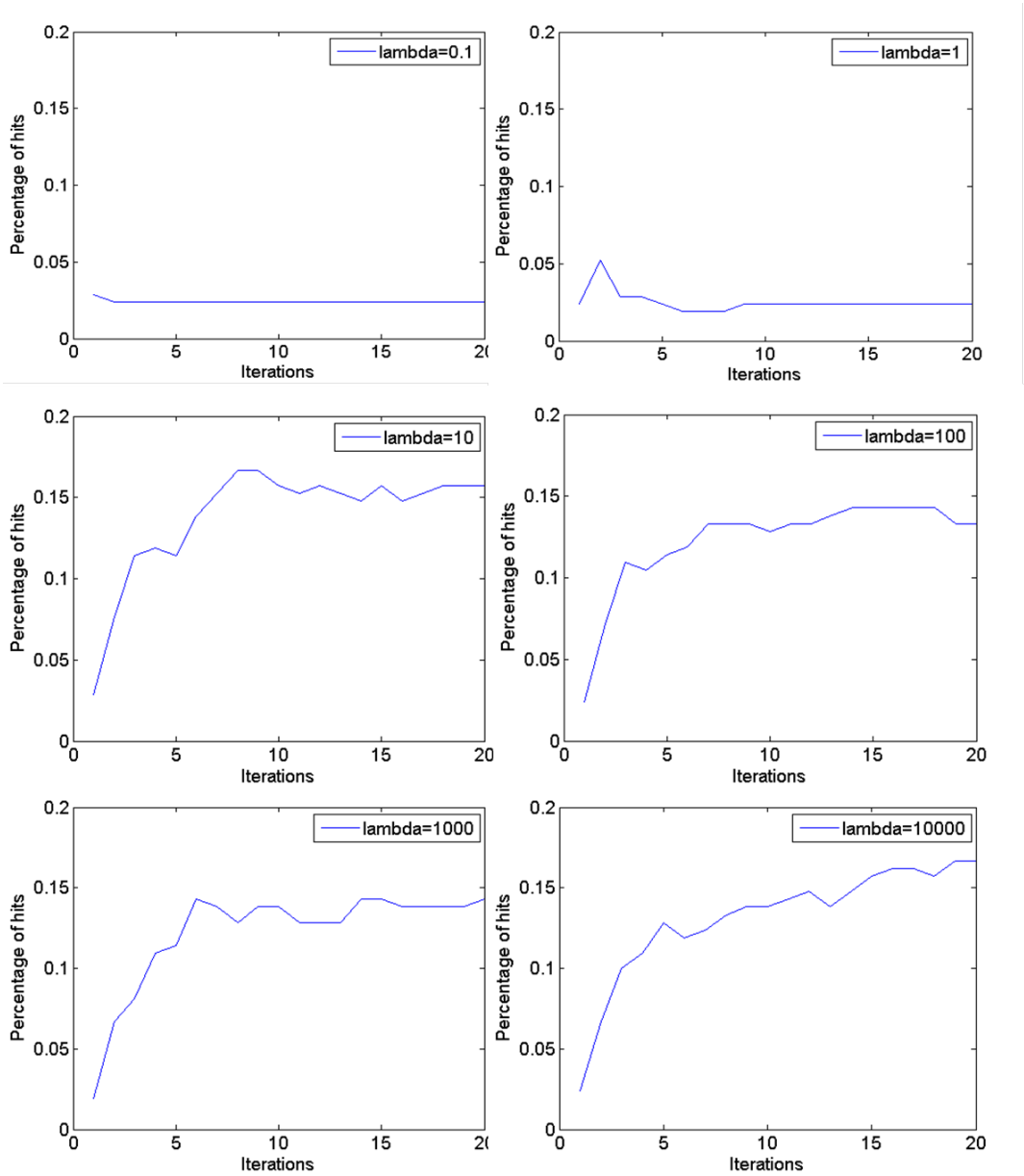

Figure 5: Percentage of hits under different λ. Top 204 nodes ranked by total scores were matched with the true source data to calculate percentage of hits. When $\lambda \leq 1$, hits remain low and do not respond to accumulation of total scores. When $\lambda \geq 10$, hits increase as total score accumulated from more iterations and reach plateau.

Source	Internal	Target
18	11	6

Table 10: Hits of H1N1 infection pathway components from top 100 nodes by the fastest diffusion method.

sophisticated method. The method relies on the likelihood to find the source. However, in current dataset, the edge distributions for PP and PTM are not available. We have to remedy by putting some empirical λ and assume equal λ for every of these edges. The assumption may not reflect the true distributions these edges. So when plugged in the likelihood function, there may exist some deviations. In addition, assumption of equal λ may lose some structural information and cause many nodes to have similar likelihood after optimization. For example, in Figure 3, when optimizing edge A-B, A-B, E-F and G-H will affect likelihood. If $\lambda_{AB} >> \lambda_{EF}, \lambda_{GH}$, shrinking AB will increase likelihood. However, when assuming $\lambda_{AB} = \lambda_{EF} = \lambda_{GH}$, for every such case, we will stretch edge A-B to increase likelihood. Thus availability of accurate edge distributions will allow this method to be improved and possibly perform better.

The fastest diffusion assumes source delivers activations to observed nodes' parents faster. It incorporates the shortest path information as well activation time of observed nodes, but put much less dependency on the edge distributions. When λ is small, it has poor performance. However, when $\lambda \geq 10$, the outcome is significantly improved. For an exponential distribution, $\lambda = 10$ corresponds to $mean = 6min$ and $\lambda = 10000$ corresponds to $mean = 0.36s$. These values are within the range of PP and PTM interactions which are generally transient. As long as λ approach this range, the accuracy of λ becomes not that importance. Thus we observed similar hits when λ increase from 10 to 10000. Therefore this method has better tolerance of edge distributions. It is worth noting that the percentage of hits increase with more iterations. This is because PP and PTM edges are initialized randomly. With less accumulations, the randomness may undermine the original structure information (one PP edge may be longer than three PP edge), while more iterations will alleviate this effect.

Based on the roles or locations of proteins n H1N1 infection pathway, proteins can be divided into three categories: source proteins are those located at very early stage of infection; internal proteins serve as mediators to transfer signal from source to downstream; target proteins located at bottom and may be responsible for gene activations. Using top 100 nodes from the fastest diffusion method, we can match 18 for source, 11 for internal and 6 for target (Table 10). Our method identified a set of source proteins as well as some mediators in H1N1 infection pathway.

7 Conclusions

H1N1 is a life-threaten and infectious disease, while understanding the key mediators in the infection pathways will provide insight into designing effective therapeutics against it. We have developed two algorithms to identify source proteins (nodes) in H1N1 infection. The first MLE with edge-wise optimization employs an iterative optimization on each feasible edge and find source nodes based on likelihood rank. This method achieve about 2 folds of hits compared to random guess. The fastest diffusion assigns scores based on the shortest path property and find source nodes based on scores. This method achieves 12 folds of hits compared to random with $p = 1.32e^{-10}$. Two methods depend differently on assumption of edge distributions and may exhibit different performance with various data availability. Given the conservation of human protein-gene network, the methods developed can be applied into predictions of source for other diseases and provide prophetic guidance for preventions and treatments.

Acknowledgments

The author thanks Professor Ziv Bar-Joseph for insightful guidance and effective discussions. The author thanks Dr.Emre Sefer and Chieh Lin for helpful communications. The author also thanks Siddhartha Jain for providing dataset.

References

[1] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[2] I. H. G. S. Consortium *et al.*, "Finishing the euchromatic sequence of the human genome," *Nature*, vol. 431, no. 7011, pp. 931–945, 2004.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, ACM, 2003.

[4] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1059–1068, ACM, 2010.

[5] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Advances in neural information processing systems*, pp. 3147–3155, 2013.

[6] S. Feizi, K. Duffy, M. Kellis, M. Medard, *et al.*, "Network infusion to infer information sources in networks," 2014.

[7] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Discovering multiple diffusion source nodes in social networks," *Procedia Computer Science*, vol. 29, pp. 443–452, 2014.

[8] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Physical review letters*, vol. 109, no. 6, p. 068702, 2012.

[9] M. Farajtabar, M. Gomez-Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song, "Back to the past: Source identification in diffusion networks from partially observed cascades," *arXiv preprint arXiv:1501.06582*, 2015.

[10] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 4, p. 21, 2012.

[11] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "Biogrid: a general repository for interaction datasets," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D535–D539, 2006.

[12] G. R. Mishra, M. Suresh, K. Kumaran, N. Kannabiran, S. Suresh, P. Bala, K. Shivakumar, N. Anuradha, R. Reddy, T. M. Raghavan, *et al.*, "Human protein reference database—2006 update," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D411–D414, 2006.

[13] J. Ernst, H. L. Plasterer, I. Simon, and Z. Bar-Joseph, "Integrating multiple evidence sources to predict transcription factor binding in the human genome," *Genome research*, vol. 20, no. 4, pp. 526–536, 2010.

[14] S. D. Shapira, I. Gat-Viks, B. O. Shum, A. Dricot, M. M. de Grace, L. Wu, P. B. Gupta, T. Hao, S. J. Silver, D. E. Root, *et al.*, "A physical and regulatory map of host-influenza interactions reveals pathways in h1n1 infection," *Cell*, vol. 139, no. 7, pp. 1255–1267, 2009.

[15] J. Singh and R. A. Padgett, "Rates of in situ transcription and splicing in large human genes," *Nature structural & molecular biology*, vol. 16, no. 11, pp. 1128–1133, 2009.

[16] S.-H. Chao and D. H. Price, "Flavopiridol inactivates p-tefb and blocks most rna polymerase ii transcription in vivo," *Journal of Biological Chemistry*, vol. 276, no. 34, pp. 31793–31799, 2001.

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] A. L. Brass, I.-C. Huang, Y. Benita, S. P. John, M. N. Krishnan, E. M. Feeley, B. J. Ryan, J. L. Weyer, L. Van Der Weyden, E. Fikrig, *et al.*, "The ifitm proteins mediate cellular resistance to influenza a h1n1 virus, west nile virus, and dengue virus," *Cell*, vol. 139, no. 7, pp. 1243–1254, 2009.

[19] A. Karlas, N. Machuy, Y. Shin, K.-P. Pleissner, A. Artarini, D. Heuer, D. Becker, H. Khalil, L. A. Ogilvie, S. Hess, *et al.*, "Genome-wide rnai screen identifies human host factors crucial for influenza virus replication," *Nature*, vol. 463, no. 7282, pp. 818–822, 2010.

[20] R. König, S. Stertz, Y. Zhou, A. Inoue, H.-H. Hoffmann, S. Bhattacharyya, J. G. Alamares, D. M. Tscherne, M. B. Ortigoza, Y. Liang, *et al.*, "Human host factors required for influenza virus replication," *Nature*, vol. 463, no. 7282, pp. 813–817, 2010.

[21] E. Bortz, L. Westera, J. Maamary, J. Steel, R. A. Albrecht, B. Manicassamy, G. Chase, L. Martínez-Sobrido, M. Schwemmle, and A. García-Sastre, "Host-and strain-specific regulation of influenza virus polymerase activity by interacting cellular proteins," *MBio*, vol. 2, no. 4, pp. e00151–11, 2011.

[22] G. Neumann, T. Noda, and Y. Kawaoka, "Emergence and pandemic potential of swine-origin h1n1 influenza virus," *Nature*, vol. 459, no. 7249, pp. 931–939, 2009.

[23] A. Gitter and Z. Bar-Joseph, "Identifying proteins controlling key disease signaling pathways," *Bioinformatics*, vol. 29, no. 13, pp. i227–i236, 2013.