Supplementary Materials

S1. Comparison DyNetViewer with other tools

Table S1 displays the comparison between DyNetViewer and several other dynamic network analysis toolkits including TVNViewer (Curtis et al., 2011), KDDN(Tian et al.,2015) and DyNet (Goenawan et al., 2016). The outstanding contributions of DyNetViewer, compared with other tools, include: (i) DyNetViewer offers four different methods for constructing dynamic protein interaction networks. (ii) DyNetViewer provides more centrality analysis algorithms for dynamic networks. Moreover, users can calculate the standard deviation for the centrality values across all sub-networks and the resulting scores are listed in order. Again, the analysis results over time can be updated in a table and can be visualized in a chart. (iii) DyNetViewer offers the cluster analysis of dynamic networks. The results of cluster analysis are intuitively presented in the form of a thumbnail list and the list can be updated over time. Similarly, the attributes of clusters over time can also be visualized in a chart.

Table S1 Several typical toolkits of dynamic network analysis

<table>
<thead>
<tr>
<th>Features</th>
<th>DyNetViewer</th>
<th>TVNViewer</th>
<th>KDDN</th>
<th>DyNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic network import</td>
<td>Supporting to import txt file of dynamic network</td>
<td>Accepting a series of networks to represent a dynamic network in text or xml format</td>
<td>-</td>
<td>Importing multiple graph files that represent a dynamic network</td>
</tr>
<tr>
<td>Dynamic network construction</td>
<td>For dynamic network construction: TC-PIN, DPIN,</td>
<td></td>
<td>Constructing differential dependency networks</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NF-APIN, ST-APIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrality analysis on the dynamic network</td>
<td>For centrality analysis: BC, CC, DC, EC, LC, NC, SC, IC, RC, SC-1, CC-1, EC-1</td>
<td>Analyzing how the degree of different nodes in the network changes over time.</td>
<td>Detecting significant rewiring</td>
<td>Identifying and analyzing the most 'rewired' nodes in a dynamic network</td>
</tr>
<tr>
<td>Cluster analysis on the dynamic network</td>
<td>For cluster analysis: MCODE, EAGLE, ClusterONE, TSN-PCD</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Visualization</td>
<td>Visualization of dynamic network, centrality analysis result visualization, cluster attributes visualization</td>
<td>Providing circle and force-directed layouts to explore the change of network structure</td>
<td>Visualization of constructed networks</td>
<td>High scoring rewired nodes are highlighted via node color, the visualization of node/edge changes across networks</td>
</tr>
</tbody>
</table>

Annotation: TC-PIN(Tang et al., 2011), DPIN(Wang et al., 2013), NF-APIN(Xiao et al., 2013), ST-APIN(Meng et al., 2016), Betweenness Centrality (BC) (Anthonisse, 1971, Newman et al.,2005), Closeness Centrality (CC) (Bavelas, 1950), Degree Centrality (DC) (Jeong et al., 2001), Eigenvector Centrality (EC) (Bonacich, 1987), Local average connectivity based method (LAC) (Li et al., 2011), Network Centrality (NC) (Wang et al., 2012), Subgraph Centrality (SC) (Estrada and Rodriguez-Velazquez, 2005), Information Centrality (IC) (Stephenson and Zelen,1969), Radiality Centrality (RC) (Valent et al.,1998), Stress Centrality (SC-1) (Shimbel,1953), Centroid Centrality (CC-1) (Harary,1969), Eccentricity Centrality (EC-1) (Hage et al.,1995), MCODE (Bader et al., 2003), EAGLE (Shen et al., 2009), ClusterONE (Nepusz et al., 2012), TSN-PCD (Li et al., 2012).

S2. Dynamic network construction algorithms integrated in DyNetViewer

In DyNetViewer, four different methods for constructing dynamic protein interaction networks are: TC-PIN (Tang et al., 2011), DPIN (Wang et al., 2013), NF-APIN (Xiao et al., 2013), and ST-APIN (Meng et al., 2016). The following is a brief introduction of four algorithms and the difference on these methods.

TC-PIN

The TC-PIN algorithm incorporates gene expression profiles into static PPI networks and constitutes time course protein interaction networks (TC-PINs). The TC-PIN algorithm mainly consists of two stages:

(i) Selecting expressed proteins. The expression values of each gene at different time points are compared with a potential threshold. When the expression level of a protein at a time point is higher than the threshold, the protein is regarded as being expressed at that time point.

(ii) Constructing sub-networks. An interaction is considered to be expressed at a time point when its two interacting proteins express simultaneously. All of expressed proteins and interactions at each time point are extracted from a static network until a TC-PIN is created. The TC-PIN consists of a series of sub-networks at different time points.

DPIN

The DPIN algorithm uses an active threshold to identify active time points of proteins based on gene expression
data and constructs a dynamic protein interaction network (DPIN). The DPIN algorithm contains two steps.

(i) Proposing a formula based on the three-sigma principle to calculate an active threshold for each protein and identifying the active time points of proteins according to the characteristics of its expression curve. A harmonic threshold of a gene is computed in terms of its algorithmic mean and variance as follows.

\[
u(p) = \frac{\sum_{i=1}^{n} EV_i(p)}{n} \quad (1)
\]

\[
\sigma^2(p) = \frac{\sum_{i=1}^{n} (EV_i(p) - u(p))^2}{n-1} \quad (2)
\]

\[
F(p) = \frac{1}{1 + \sigma^2(p)} \quad (3)
\]

\[
S_1(p) = u(p) \quad (4)
\]

\[
S_2(p) = u(p) + 3\sigma(p) \quad (5)
\]

\[
Active_{Th}(p) = S_1(p) \times F(p) + S_2(p) \times (1 - F(p))
\]

\[
= \mu(p) + 3\sigma(p)(1 - F(p)) \quad (6)
\]

For each gene \( p \), \( EV_i(p) \) is the expression value of \( p \) at time point \( i \), \( u(p) \) and \( \sigma(p) \) are the mean and the standard deviation of its expression values, respectively. \( F \) reflects the fluctuation of its expression curve, \( Active_{Th}(p) \) is the active threshold of gene \( p \). If the gene expression value of a protein is above its threshold at a given time, the protein is considered to be active at that time point.

(ii) Extracting a dynamic protein interaction network (DPIN) by combining the activity information with a protein interaction network. A DPIN is constituted by active proteins and the efficacious interactions. A DPIN is defined as a graph \( G_A = (V_A, E_A) \), where \( V_A \) is the set of all active proteins and \( E_A \) is the set of all active interactions.

**NF-APIN**

To construct a noise-filtered active protein interaction network, the NF-APIN algorithm is structured as a sequence of the following steps:

(i) Filtering noisy genes. Firstly, NF-APIN employs a dynamic model-based method to screen contaminated data from original expression array. By using time-dependent model and time-independent model (Xiao et al., 2013), gene expression profiles are divided into two categories: time-dependent data and time-independent data. If the mean of time-independent gene expression data is very small, this gene is regarded as being noise.

(ii) Filtering gene expression data point. To confirm the active time points of each noise-filtered gene, a threshold for each gene is computed by a threshold function which is described as follows:

\[
F = \frac{1}{1 + \sigma^2}
\]

\[
\sum Active_{threshold} = u + k\sigma \times (1 - F) \quad (8)
\]

For each gene, \( u \) and \( \sigma \) are the mean and standard deviation of its expression values, respectively. The value of coefficient \( k \) is selected as 2.5. If the expression level of a protein is higher than its corresponding threshold at a time point. A gene (protein) is considered to be active at a time point while its expression level is over its corresponding threshold.

(iii) Constructing a noise-filtered active protein interaction network (NF-APIN). The active information can be obtained from gene expression profiles through the above two steps. A noise filter active protein interaction network is constituted based on the active information and the static network.

**ST-APIN**

The ST-APIN algorithm reconstructs a Spatial and Temporal Active Protein Interaction Network (ST-APIN) by integrating gene expression profiles and sub-cellular location data with the static protein interaction network. It includes four steps as follows:
Filtering noisy genes by using time-dependent model and time-independent model.

(ii) Filtering gene expression data time points based on the 3-sigma principle.

(iii) Filtering edges of protein interactions by using sub-cellular localization information. If two interacting proteins are not in the same sub-cellular localization, this interaction is removed.

(iv) Constructing the spatial and temporal active protein interaction network.

The difference on these methods

There are some differences between these four algorithms. TC-PIN, DPIN and NF-APIN construct a dynamic network by integrating time-course gene expression data and a static PPI network. However, ST-APIN integrates not only time-course data but also tissue-specific or location data into PPI networks to construct dynamic biological networks. TC-PIN compares gene expression levels at different time points for all genes with a fixed threshold to determine the dynamic of protein expression. DPIN uses an active threshold to determine the active time points of proteins based on the three-sigma principle, rather than a fixed threshold. In the construction of NF-APIN, a dynamic model-based approach is presented to distinguish time-dependent gene expression data from time-independent gene expression data to filter out contaminated data. In consequence, this algorithm is suitable for situations when background noise exists in the gene expression array. ST-APIN consists of a series of spatial and temporal active protein interaction sub-networks. Therefore, sub-cellular location information should be prepared to construct ST-APIN. Users can select an appropriate method according to data characteristics and their own needs for a real application.

The modular way to construct dynamic network

DyNetViewer can implement dynamic network construction methods in a modular way and allow users to freely select and combine these modules to obtain their own network construction method. There are four modules can be selected: Step 1. Filtering noisy genes (using time-dependent model and time-independent model); Step 2. Filtering gene expression data (using threshold or 3-sigma principle); Step 3. Using sub-cellular localization information; Step 4. Constructing dynamic networks.

S3. Centrality analysis algorithms integrated in DyNetViewer

DyNetViewer supports node centrality analysis of dynamic networks and integrates twelve typical centrality measures including Betweenness Centrality (BC), Closeness Centrality (CC), Degree Centrality (DC), Eigenvector Centrality (EC), Local Average Connectivity-based method (LAC), Network Centrality (NC), Subgraph Centrality (SC), Information Centrality (IC), Stress Centrality (SC-1), Radiality Centrality (RC), Eccentricity Centrality (EC-1), and Centroid Centrality (CC-1). The description of twelve algorithms is listed in Table S2.

S4. Clustering algorithms integrated in DyNetViewer

In DyNetViewer, four different graph clustering algorithms for analyzing clusters of dynamic networks are: MCODE (Bader et al., 2003), EAGLE (Shen et al., 2009), ClusterONE (Nepusz et al., 2012) and TSN-PCD (Li et al., 2012). The following is a brief description of these four algorithms and the difference on these methods.

MCODE (Molecular COMplex Detection) is a density-based algorithm of clustering. TSN-PCD (Time-sequenced network-based protein complex discovery) algorithm is developed from our previous algorithm HC-PIN. HC-PIN (fast Hierarchical Clustering algorithm in PPI Networks) is a fast, hierarchical, non-overlapping clustering algorithm. EAGLE (agglomerative hierarchicAI clusteringG based on maximal clique) is a hierarchical-based, overlapping algorithm to identify complex community structure in a given network. ClusterONE(Clustering with Overlapping Neighborhood Expansion) is a density-based clustering algorithm to identify protein complexes by recognizing dense subgraphs from biological networks.

S5. CASE STUDY

S5.1 Experimental data

In DyNetViewer, a dynamic network can be constructed by integrating a static network with time-course data. For example, one can construct a dynamic protein interaction network by integrating time-course gene expression data. Of
course, tissue-specific or location data can also be used for constructing dynamic biological networks, such as sub-cellular localizations data. In this case, we obtain H5N1 normal network and H5N1 inflammatory network and their corresponding gene expression data from Jin’s article (Jin et al., 2014). Then we apply DyNetViewer to construct and analyze the normal and inflammatory dynamic networks for H5N1 infections. We also provide the sample files on the web at https://drive.google.com/open?id=0B2Xs3J1bezYMDY5cHE4UZjRWM.

### Table S2 Centrality Analysis Algorithms

<table>
<thead>
<tr>
<th>Centrality name</th>
<th>Description</th>
<th>Refs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betweenness Centrality (BC)</td>
<td>$C_B(u) = \sum_{u \neq t \neq v} \frac{\rho(s,u,t)}{\rho(s,t)}$ where $\rho(s,t)$ is the total number of shortest paths from node $s$ to node $t$, $\rho(x,y)$ is the number of those paths that pass through $x$</td>
<td>Newman et al., 2005</td>
</tr>
<tr>
<td>Closeness Centrality (CC)</td>
<td>$C_C(u) = \frac{1}{\sum_{v \neq u} dist(u,v)}$ is the distance of the shortest path from node $u$ to node $v$.</td>
<td>Bavelas, 1950</td>
</tr>
<tr>
<td>Degree Centrality (DC)</td>
<td>$C_D(u) =</td>
<td>N_u</td>
</tr>
<tr>
<td>Eigenvector Centrality (EC)</td>
<td>$C_E(u) = a_{max}(u), a_{max}$ is the eigenvector corresponding to the largest eigenvalue of the adjacency matrix $A$.</td>
<td>Bonacich, 1987</td>
</tr>
<tr>
<td>Local average connectivity based method (LAC)</td>
<td>$LAC(u) = \frac{\sum_{v \neq u}</td>
<td>deg_G^0(v)\cap {u,v}</td>
</tr>
<tr>
<td>Network Centrality (NC)</td>
<td>$C_N(u) = \sum_{v \neq u} ECC(u,v)\sum_{v \neq u} \frac{</td>
<td>N_u</td>
</tr>
<tr>
<td>Subgraph Centrality (SC)</td>
<td>$C_S(u) = \sum_{v \neq u} f(V^k)\sum_{v \neq u} \mu_k(u)$ is the $k$th diagonal entry of the $k$th power of the adjacency matrix of network. $v_1, v_2, \ldots, v_k$ is $R^k$ be an orthonormal basis of composed by eigenvectors of $A$ associated to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$. $V_k$ is the $k$th component of $V_k$.</td>
<td>Estrada and RodriguezVelazquez, 2005</td>
</tr>
<tr>
<td>Information Centrality (IC)</td>
<td>$I_{uv} = (C_{uu} + C_{uv} - 2 \times C_{uw})^{-1}1_{uv} = (C_{uu} + C_{uv} - 2 \times C_{uw})^{-1}e = (C_{uw}) = (D - A + f)^{-1}e = A$ is the adjacency matrix of network, $D$ is the diagonal matrix of all nodes’ degree, $f$ is a matrix whose all elements are 1.</td>
<td>Stephenson and Zelen, 1989</td>
</tr>
<tr>
<td>Radiality Centrality (RC)</td>
<td>$C_{rad}(u) = (\sum_{v \neq u} RD_{uv})/(n - 1)$, $RD$ is the reverse distance matrix which is defined as $RD_{uv} = diameter(G) + 1 - D_{uv}$, $diameter(G)$ is the diameter, distance matrix $D = (dist(u,v))$.</td>
<td>Valente et al., 1998</td>
</tr>
<tr>
<td>Stress Centrality (SC-1)</td>
<td>$C_{str}(v) = \sum_{w \in V(u,v)} \sigma_{uv}(v)$ Stress is calculated by measuring the number of shortest paths passing through a node.</td>
<td>Shimbel, 1953</td>
</tr>
<tr>
<td>Centroid Centrality (CC-1)</td>
<td>$C_{cen}(v) = \min {f(v,w) : w \in V(v)}$, $f(v,w) = \rho_e(\omega) - \rho_e(\omega')$, $\rho_e(\omega)$ is the number of vertex closer to $v$ than to $\omega$.</td>
<td>Harary, 1969</td>
</tr>
<tr>
<td>Eccentricity Centrality (EC-1)</td>
<td>$C_{ecc}(v) = \frac{1}{\max{\rho_e(v,w) : w \in V(v)}}$, the eccentricity of a node $v$ is calculated by compute the shortest path between the node $v$ and all other nodes in the graph.</td>
<td>Hage et al., 1995</td>
</tr>
</tbody>
</table>

### S5.2 Construction and visualization of the normal and inflammatory dynamic networks for H5N1 infections

We import the static network and gene expression data into DyNetViewer firstly. There are four algorithms which can be selected to construct a dynamic network. Here, we select TC-PIN algorithm and set the threshold to 6.5. The operation steps follow as:

- To open DyNetViewer Panel click on Apps -> DyNetViewer -> Construct Dynamic Network-> open.
- To import the static network into Cytoscape click on File -> Import -> Network ->File “network_H5N1_inflammation.txt” (“network_H5N1_normal.txt”).
To import gene expression data:
  a. Click on Upload Bioinformatics Button.
  b. Click on Upload from file Button.
  c. Select txt file of gene expression data “expression_H5N1_inflammation.txt” (“expression_H5N1_normal.txt”).

To construct dynamic network:
  a. Select algorithm TC-PIN and set the threshold to 6.5.
  b. Click on Construct Dynamic Network Button.

After constructing the dynamic network, the network view can be displayed in Cytoscape (Fig. S1).

S5.3 Node centrality analysis and visualization of the normal and inflammatory dynamic networks for H5N1 infections

There are 12 algorithms which can be selected to implement centrality analysis. Here, we select all algorithms to analyze the centralities of the normal and inflammatory dynamic networks for H5N1 infections. The steps of centrality analysis follow as:

- To open Dynamic Node Analysis Panel click on Apps->DyNetViewer -> Dynamic Node Analysis -> open.
- To compute centralities:
  a. Select one or multiple algorithms from the twelve centrality analysis algorithms.
  b. Click on Dynamic Centrality Analysis Button.
- To calculate the standard deviation for the centrality values across all sub-networks and sort the resulting score click on Ranking by Standard Deviation Button.
- To visualize centralities of nodes of a single network:
  a. Select one or more algorithms from Dynamic Node Analysis Panel.
  b. Select one or more nodes from the Standard Deviation Result Panel or from network view.
  c. Select “from one Network” RatioButton.
  d. Click on Centrality Visualization Button.
- To visualize centralities of nodes of multiple networks:
  a. Select one or more nodes from H5N1 normal network view and H5N1 inflammatory network view.
  b. Select H5N1 normal network and H5N1 inflammatory network at the same time.
  c. Select one or more algorithms from Dynamic Node Analysis Panel.
  d. Select “from multiple Networks” RatioButton.
  e. Click on Centrality Visualization Button.

The centrality analysis result is added to Node Table (Fig. S2). The centrality values of nodes can be updated with the movement of the time slider of the DyNetViewer Panel. Then we plot charts of different centrality values of certain node over time and centrality values of different nodes over time. We can also visualize a specific node overtime for normal and inflammatory network (Fig S3).

S5.4 Cluster analysis and visualization of the normal and inflammatory dynamic networks for H5N1 infections

Four graph clustering algorithms were implemented in DyNetViewer for analyzing clusters of dynamic networks: MCODE (Bader et al., 2003), EAGLE (Shen et al., 2009), ClusterONE (Nepusz et al., 2012) and TSN-PCD (Li et al., 2012). Here, we select TSN-PCD to analyze the clusters of the normal and inflammatory dynamic networks for H5N1 infections. Cluster analysis of normal and inflammatory dynamic networks for H5N1 is described as follows.

- To open Dynamic Clusters Panel click on Apps->DyNetViewer -> Dynamic Clusters -> Open.
- To analyze clusters:
  a. Select H5N1 normal dynamic network (or H5N1 inflammatory dynamic network).
b. Select “TSN-PCD” RadioButton.
c. Select “Strong” RadioButton.
d. Click on Dynamic Cluster Analyze Button.

- To visualize cluster attributes of a single network:
  a. Select a cluster from Cluster Result Panel.
  b. Select one or more cluster attributes.
  c. Select “from one Network”RatioButton.
  d. Click on Cluster Attributes Visualization Button.

- To visualize cluster attributes of multiple networks:
  a. Select a cluster from Cluster Result Panel of H5N1 normal network and select another cluster from Cluster Result Panel of H5N1 inflammatory network.
  b. Select H5N1 normal network and H5N1 inflammatory network at the same time.
  c. Select one or more cluster attributes.
  d. Select “from multiple Networks”RatioButton.
  e. Click on Dynamic Cluster Analyze Button.

After finishing the cluster analysis of the dynamic network with algorithm TSN-PCD, the result is displayed in the right panel. In the option of “list all clusters”, all clusters from each sub-network are merged and listed in the panel. In the option of “list the clusters at this time”, the clusters obtained at current time are listed. In this situation, the clusters list can be updated with the movement of the time slider (Fig S4). Moreover, the charts of cluster attributes over time can be plotted. In particular, we plot different attributes of the complex TNFSF10/HDAC4/HDAC5 from normal and inflammation network and observed how the attributes of the complex TNFSF10/HDAC4/HDAC5 change over time (Fig. S5).

Fig. S1. The construction of inflammatory dynamic networks for H5N1 infections with TC-PIN algorithm. This is a sub-network at time 4. The network contains 44 nodes and 95 edges at current moment.
Fig. S2. The centrality analysis results of inflammatory dynamic networks for H5N1 infections. Centrality analysis results can get update over time. (a) is the result at time 0. (b) is the result at time 4.

Fig. S3. The visualization of node centrality analysis results of inflammatory dynamic networks for H5N1 infections. (a) The visualization of LAC, CC, NC values of node MAPK14. (b) The visualization of DC values of node IL8, TNF, CCL5. (c) The visualization of BC of node HDAC5 for normal and inflammatory network.

Fig. S4. The list of cluster analysis results from dynamic inflammation network during H5N1 infections. (a) In the “list all clusters” mode, 7 clusters in total. (b) In the “list the clusters at this time” mode, there are 2 clusters at time 2. (c) In the “list the clusters at this time” mode, there is one cluster at time 4.
References


