Toolbox Readme

March 28, 2017

Toolbox accompanying Dynamic Graph Metrics: Tutorial, Toolbox, and Tale,

Visualization tools

plotDNline Plots the dynamic network given by *contacSequence* with *nNodes* nodes and exists within the interval *timeInterval*. Example shown in Fig. 1. Node colors chosen randomly. Best for networks where only one contact occurs at each timepoint.

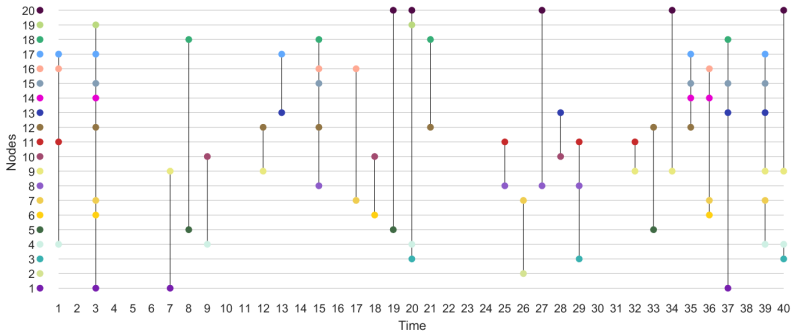

Figure 1: Plot of dynamic network using plotDNline.

plotDNarc Plots the dynamic network given by *contacSequence* with *nNodes* nodes and exists within the interval *timeInterval*. Example shown in Fig. 2. Node colors chosen randomly. Recommended for generally sparse networks with fewer than 50 nodes.

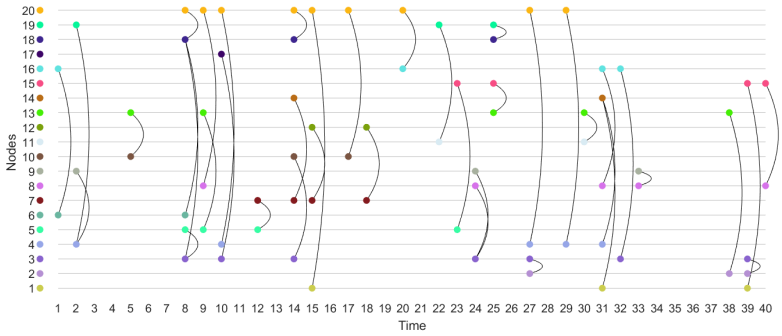

Figure 2: Plot of dynamic network using plotDNarc.

plotDNarc_dir Plots the directed dynamic network given by *contacSequence* with *nNodes* nodes and exists within the interval *timeInterval*. Example shown in Fig. 3. Node colors chosen randomly. Recommended for generally sparse networks with fewer than 50 nodes.

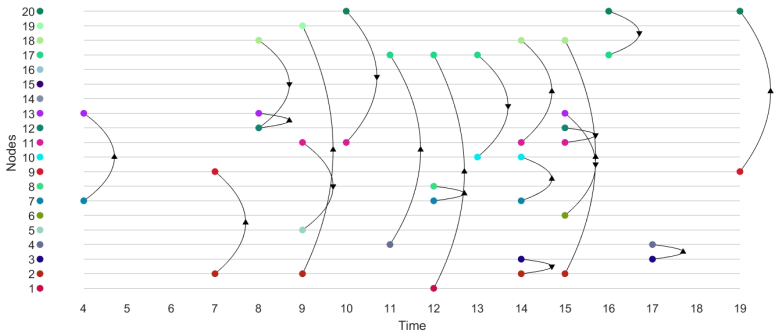

Figure 3: Plot of directed dynamic network using plotDNarc_dir.

plotArcNetwork Plots a weighted, undirected network along a circle with edges shown as arcs.

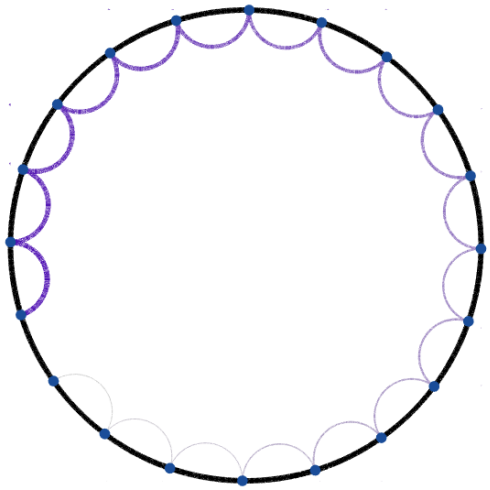

Figure 4: Plot of ring dynamic network time-aggregate graph using plotArcNetwork.

Models

randomDN Creates a random dynamic network on *nNodes* nodes with *nEdges* total connections. Nodes are randomly connected and contact times chosen randomly from *edgeTimes*. Self-connections are not allowed.

ringDN Creates a dynamic network in a ring structure on *nNodes* nodes. Edges connect adjacent nodes in ascending order at linearly spaced times in the interval *timeInterval*.

randomizedEdges Given a dynamic network as *contactSequence*, visit each edge and rewire it to a randomly chosen node while keeping the time of contact the same.

randomPermutedTimes Given a dynamic network as *contactSequence*, permute all contact times.
shuffledTimeSteps Thinking of the dynamic network as a sequence of graphs $G_1, G_2, \ldots, G_T$, shuffle the order of the graphs in this sequence to return a randomized dynamic network.

Dynamic graph metrics

betweennessCentrality Given a *contactSequence*, calculate the betweenness centrality $C_B(i,t) = \sum_{i \neq j \neq k} \frac{\sigma_{j,k}(i,t)}{\sigma_{j,k}(t)}$ for $\sigma_{j,k}(i,t)$ the number of shortest paths from j to k which pass through node i beginning no earlier than time t.
burstiness Given input vector, calculate the burstiness. As defined in Holme, Samaraki 2011, the burstiness of a vector v is $B = \frac{\sigma_v - m_v}{\sigma_v + m_v}$ with σ_v and m_v the standard deviation and mean, respectively, of v. This code also outputs the coefficient of variation.

closenessCentrality Given an initial node i and time, compute the closeness centrality $C_C(i,t) = \frac{1}{N-1} \sum_{i \neq j} \frac{1}{\tau(i,j,t)}$. This function also returns the vector *tau_vec* recording $\tau(i,j,t)$ for each node j.

forwardLatency Given an initial and final node (*node_i* and *node_j*), along with a starting *time*, return the next time at which node j could be contacted by node i ($\tau_{i,t}(j)$).

informationLatency Given a particular moment in time, this function returns the most recent point at which *node_i* received information from *node_j* ($\phi_{i,t}(j)$) in addition to the information latency $\lambda_{i,t}(j) = t - \phi_{i,t}(j)$.

informationView Calculate node i's view of node j's information at a particular time t.

isStrongly(Weakly)Connected Determine if nodes *node_i*, *node_j* are strongly (weakly) connected.

latency Returns the latency for node pair *node_i*, *node_j*, calculated using *latencyComputations*. At the first moment the latency is minimal, this function also returns the path length (number of edges in this minimal time path) and last contact time (time at which this path was completed).

listAllMotifs Return all unique paths within the dynamic network. Very memory intensive, not recommended for large/dense dynamic networks.

reachabilityAtTimeT Creates a binary network describing which nodes are connected at time t in the dynamic network encoded by *contactSequence*.

setOfInfluence Determine the nodes reachable from node *node_i* via time-respecting paths beginning at time *t_i* or later.

sourceSet Determine set of nodes from which node *node_i* can be reached via time-respecting paths beginning at time *t_i* or later.

temporalCorrelation For a *contactSequence*, calculate the temporal correlation coefficient for each node, defined $C_i = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{\sum_j A_{i,j}(t) A_{i,j}(t+1)}{\sqrt{[\sum_j A_{i,j}(t)][\sum_j A_{i,j}(t+1)]}}$.

temporalSmallWorldness Given a *contactSequence*, calculate the temporal characteristic path length and temporal correlation coefficient.

Other helpful functions

arrayToContactSeq For a network stored in an *nNodes x nNodes x timePoints* array, convert to contact sequence. Takes binary or weighted networks.
forwardLatencyComputaitons Output additional arrays recording latency, path length, and last contact information given a *contactSequence* and pair of nodes.
latencyComputations Calculate latency, path length, and time since last contact for all node pairs at each possible contact time. Returns three *nNodes* x *nNodes* x nContactTimes arrays with this information recorded. Optionally input a vector *interval* with start and end times for a windowed calculation. Currently the memory is infinite, so a node can hold information for any amount of time.

makeReachabilityArray Construct a *nNodes* x *nNodes* x —*contactTimes*— array recording if node j can be reached from node i via time-respecting paths by time t. Can use either directed or undirected dynamic networks.

reachabilityGraph Construct a *nNodes* x *nNodes* graph recording if node j can be reached from node i via time-respecting paths at anytime. Can use either directed or undirected dynamic networks.

networksFromContacts Takes a contact sequence and creates sequence of binary or weighted networks.

timeAggregate_bin Creates a binary undirected network from a dynamic network of *nNodes* nodes and encoded by *contactSequence*. Output network has an edge between node pairs which are connected at any time.

thresholdMatDensity Threshold a matrix by edge density.