Network k-test Tutorial

Created by: Kim VanderWaal, January 24, 2017

Please cite the following when using th network k-test for research or educational purposes: VanderWaal, K. L., Enns, E. A., Picasso, C., Packer, C. & Craft, M. E. 2016. Evaluating empirical contact networks as potential transmission pathways for infectious diseases. Journal of the Royal Society Interface, 13, 20160166.

We will be working with a set of data related to social interactions among ground squirresl and infection data for Cryptosporidium.

First, make sure that you have the proper packages installed and loaded. In R, anything that is writted after a # symbol are considered comments (not code).

It also is a good idea to set your working directory to a known location using the “session” drop down menu. Make sure the files downloaded as part of the tutorial are in thew orking directory

```
#install.packages("igraph")

library(igraph)
```

Read in your source file. This should be located in your working directory. This will read in all the functions associated with this “package”

```
source("Cluster.test.source.parallel.R")
```

Making a igraph object for analysis.

An attribute file and an edgelist file are required for running the k-test. Both should be saved in .csv format. See side panel for details on format.

Edgelist file: a .csv file listing all edges/links in the network. One column should be titled ‘v1’ and another column should be titled ‘v2’. Names used in the edgelist should match the nodes listed in the attribute file. For directed networks, the sender should be in the first column, and the reciever in the second. Any additional columns will be interpreted as edge weights or edge attributes by igraph.

Read in the edgelist file (edges.A.csv):

```
e.A <- read.csv(file.choose(),stringsAsFactors=F)  #read edges.A.csv or edges.B.csv
```

Attribute file: a .csv file listing all nodes in the network (inlcuding isolates). One column should be titled ‘name’ and another column should be titled ‘state’. The names of columns should have no capitalization. Names can be either written as characters or numbers. The state column represents infection status: 0 = Non-infected; 1 = Infected

```
attr1 <- read.csv(file.choose(),stringsAsFactors=F) #read in attributes
```

Ultimately, this function requires your data to be in the iGraph format with a a vertex attribute called “name” (as a character), and “state” (as an integer). All non-infected nodes should be zeros, and infected nodes should be 1.

Now, make an igraph object and add vertex attribute “state.” Here, we assume that the network in undirected. For directed networks, use “directed = T”.

Note that making the network from the edgelist in this way will eliminate any isolates. To include isolates, the vertices= argument can be used to specify a data.frame (i.e., such as the attribute dataframe) that includes all vertices including isolates.

```
netA <- graph.data.frame(e.A,directed=F) #directed=F if this is an undirected network
```

Add attribute for state as an integer.Non-infecteds should be 0, infecteds should be >0

```
V(netA)$state <- as.integer(attr1$state[match(V(netA)$name,attr1$name)] )

plot(netA,layout=layout.fruchterman.reingold,vertex.label=NA,vertex.color=V(netA)$sta
te)
```

Running the analysis

Now, run the k-test using the following code. You will get a few warning messages, but those do not affect the running of the analysis.

```
test2 <- k.test(netA,k=1,type="both",bin="full",iterations=20)

test2
```

The output is a list of three objects: test2[[1]] is a dataframe with the output of the permutations test2[[2]] is the observed mean and median k-statistics and the p-values. The mean should be used for interpretation of results test2[[3]] is a density plot of the null distribution of the k-statistic with the observed k-statistic in red

Your test can be saved as an .Rdata file

```
save(test2,file="test2.Rdata")

#to re-load results
load("test2.Rdata")
```

A few notes about the options availabe

k=

Refers to how large of a neighborhood is considered for the k-statistic. k=1 refers to only direct connections of the infected node, where as k=2 refers to all connections within two steps. It is not advisable to do k=2 for very large networks

iterations=

How many permutations to perform. Default should be 1000

type=

The type option should be set to ‘all’ if using an undirected network. For directed networks, the k-statistic can be calculated for incoming paths, outgoing paths, or both. Using the ‘all’ command with a directed network will result in the edge directions being ignored

bin=

The bin option is set to ‘full’ by default, indicated a complete randomization of the pattern of infected nodes. For large networks, node infection statuses are randomized within each quartile of either degree or neighborhood size (all nodes with two steps) to preserve the overall connectivty levels of infected nodes while randomizing who is connected with whom

node.type=

The k-statistic can be calculated for nodes only of a given type if you have a vertex attribute called “node.type.” Node.type attributes (e.g, Male or Female) can be added from an attribute table in the same way that we added the “state” attribute. If node.type = “Male”, then all infected nodes (male or female) within k steps will be calculated for the infected male nodes (females are not considered focal).

par=

This indicates whether the user desires the permutations to be calculated via parallel processing. Default is true.

Degree-based approaches for assessing network structure and infection (Kruskal-Wallis test)

```
library(agricolae)
deg <- degree(simplify(netA))

inf.status <- V(netA)$state
#replace NA with 0
inf.status <- ifelse(is.na(inf.status)==T,0,inf.status)

data <- data.frame(
  degree = deg,
  state = inf.status
  )

kw <- kruskal(data$deg,data$state,p.adj="bonferroni")
kw
```

Extensions of the k-test

ks-test

When spatial coordinate data is available, we can evaluate both the role of proximity in space and proximity in the network in determining patterns of infection. Here, additional vertex attributes should be added to the network, entitled “lat” and “long.”

IMPORTANT: The geographic coordinates (lat and long) should be given in UTMs.

The threshold argument refers to a threshold distance around each node in which to look for other infected nodes. This generates a value similar to the k-statistic in terms of how many infected nodes are within this threshold distance from the infected node.

For this analysis, we will use a different data that includes geogrpahic information. Load the Rdata files “edges_farms.Rdata” and “attr_farms.Rdata” using load(file.choose()). This dataset includes weighted edges (i.e., the # or “batch.size” of cattle moved between farms), and the GPS coordinates of farms. The attribute R object is named attr1 and the edgelist is named edges

First, we will have to again go through the steps to format this data for analysis as an igraph object. Note that this time, this is a directed network.

```
#first, ensure that state is an integer in the attribute data frame (This time, we wi
ll add all attributes using the vertices argument in graph.data.frame)
class(attr1$state)
attr1$state <- as.integer(attr1$state)

net2 <- graph.data.frame(edges,directed=T,vertices=attr1)
#note that the state variable was added automatically because we used the vertices ar
gument in graph.data.frame

#Double-check to see if state is an integer
class(V(net2)$state)

plot(net2,layout=layout.fruchterman.reingold,vertex.label=NA,vertex.color=V(net2)$sta
te,vertex.size=10,edge.arrow.size=.3)

test <- ks.test(net2,k=1,type="both",bin="full",iterations=15,threshold=5,plot.ks=T)

test
```

For comparison, compare results of the k-test for this network with the Kruskal-Wallis test

Path-based test for weighted networks

If the network is weighted, it may be preferable to use the weighted data (which is ignored in the k-test). Two ways exist to do this: 1. Threshold the network to filter out edges below a certain weight, leaving only the stronger edges within the network 2. Use the path-based test

The path-based test is similar to the network-test in that we are comparing an observed test statistic to a permuted distribution where the location of infected nodes is randomized. Here, the test statistic is the average weighted inverse path-length between each infected node and the nearest other infected node. In this case, a SMALL observed value is indicative of tranmission through the network, given that we are looking at the inverse of weights.

```
test.path <- path.test(net2,weight=E(net2)$batch.size,bin="full",iterations=10,type="
in")

test.path
```