**Supplementary Material - The application of statistical network models in disease research**

**Example 1. Using a network autocorrelation model to test hypotheses related to individual bovine tuberculosis infection status in European badgers**

A network autocorrelation model (NAM) was applied to a weighted contact network constructed using proximity logging data collected in a population of badgers in Southwest England in 2009/2010 (see Weber et al 2013a). For the purposes of this example an association matrix based on the log of the duration of social contacts over the whole study period was used (June 2009 – May 2010), and all collared badgers (n=51) were included regardless of the duration of their datastream. To make the analysis similar to that used in Weber *et al.* (2013a) we also split the full association matrix into a network of within-group interactions and between-group interactions using six social groups determined from the full network using a multi-level community detection algorithm in R package igraph (Csardi and Nepusz 2006).

The model fitted was used to illustrate some of the key options available when using this approach. Bovine tuberculosis (bTB) infection status (infected individuals being defined as individuals that were test positive by either Stat-Pak serological assay or Interferon Gamma Release Assay: Weber *et al.* 2013a) of the 51 individuals in the network was modelled as a binary response using a binomial family model. Sex (female/male) and age (adult/yearling) were fitted as fixed effects. Between-group flow was fitted as a centrality effect – we used this term as it of intuitive interest from a disease perspective and has been shown to be related to bTB infection status by previous work (Weber et al 2013a). We also fitted centrality terms for within-group centrality – both “eigenvector” and “outdegree” effects. Each of these network centrality terms was centred using the “center=TRUE” argument. Finally we used the weightlag() argument to account for non-independence of neighbouring individuals in our weighted network. Results of the model are provided in Table S1.

*Table S1. Results of a Network Autocorrelation Model**of bTB infection status as an outcome of individual sex, age and network position, while controlling for network autocorrelation.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model term | Estimate | Standard Error | Z value | P value |
| Intercept | -0.005 | 1.70 | -0.003 | 0.042 |
| Sex | 2.70 | 1.17 | 2.31 | 0.021 |
| Age | -2.57 | 1.09 | -2.35 | 0.019 |
| Between-group  Flow | 0.009 | 0.003 | 2.89 | 0.004 |
| Within-group  Degree | 0.07 | 0.04 | 2.00 | 0.046 |
| Within-group  Eigenvector | 18.84 | 6.83 | 2.76 | 0.006 |
| Weightlag() term | -0.37 | 0.12 | -3.22 | 0.001 |

The model shows a significant positive relationship between the probability of bTB infection and between-group flow, as would be expected. Individuals with high between-group flow centrality will occupy “spread-capacitor” positions in a social network, as identified by Weber et al (2013a). The within-group centrality results are also of interest. In particular, the highly significant positive effect of within-group eigenvector centrality is of note as it is not a metric considered by the original study. The slight positive effect of within-group degree (while marginal) is also different to the original study (Weber *et al.* 2013a), albeit with the caveats that a) we have not made any effort to control for the length of time individuals were collared (and collecting data) for this example analysis, b) our results are not split by season as in the original work, and c) we have defined social groups in a different way to the original study. Together, these centrality results suggest that the main results of Weber *et al.* (2013a), in particular for between-group centrality, are robust, but that there may be some additional nuance that using multiple explanatory variables together in the same model can uncover. There was also a significant negative effect of the weightlag() term, indicating that individuals are less likely to test bTB positive if they interact with other infected individuals. This reveals that there is no assortment of bTB infection in the social groups studied, a result that can be compared to the even distribution of bTB infection among groups depicted in figure one of Weber *et al.* (2013a). It should also be highlighted that individuals in this study were defined as infected when test positive (test positive by either Stat-Pak serological assay or Interferon Gamma Release Assay: Weber *et al.* 2013a) rather than necessarily being infectious (test positive by bacterial culture), which means positive covariance in infection may not be expected. The model also reveals that adults are more likely to be infected than yearlings (as would be expected as they have had more time to acquire infection), and that males are more likely to be infected than females, as would be expected from previous population-level results in badgers.

***R code:***

##load required R package

library(xergm)

##set directory - change as required

setwd("X")

##read in matrix and sort

X<-read.csv("overallnetwork.csv")

names<-X[,1]

X<-as.matrix(X[,2:ncol(X)])

colnames(X)<-rownames(X)<-names

diag(X)<-NA

#read in social group membership

groups<-read.csv("Complete Membership.csv")

#log edge weights

for(i in 1:nrow(X)){

for(j in 1:ncol(X)){

if(X[i,j]>0&is.na(X[i,j])==FALSE){

X[i,j]<-log(X[i,j])

}

}

}

#create network of within-group interactions

X2<-X

for(i in 1:51){

for(j in 1:51){

if (groups[i,2]!=groups[j,2]){

X2[i,j]<-0

}

}

}

#create network of between-group interactions

X3<-X

for(i in 1:51){

for(j in 1:51){

if (groups[i,2]==groups[j,2]){

X3[i,j]<-0

}

}

}

#read in bTB status

TB<-read.csv("TBstatsF.csv")

#This refers to TB infection at the end of the study

#Read in Sex

Sex<-read.csv("indivsexes.csv")

#read in Ages

Age<-read.csv("ages.csv")

#Change response and covariates into appropriate format

#this involves 1 column with values with individuals IDs as the rownames

TB2<-as.data.frame(as.numeric(TB[,2])-1)

names(TB2)<-"TB"

rownames(TB2)<-TB[,1]

Sex2<-as.data.frame(Sex[,2])

rownames(Sex2)<-Sex[,1]

Age2<-as.data.frame(Age[,2])

rownames(Age2)<-Age[,1]

###Run model

#binomial family to model infected/uninfected

#TB infection status as response

#sex and age as covariates (2 level factors)

#An effect of network centrality - we chose flow as it is of intuitive interest from a disease perspective and of interest from previous results

#A weighted network autocovariance term - weightlag

m1<-tnam(formula=TB2~

covariate(Sex2,coefname="sex")+

covariate(Age2,coefname="age")+

centrality(X3,type="flow",center=TRUE)+

centrality(X2,type="eigenvector",center=TRUE)+

centrality(X2,type="outdegree",center=TRUE)+

weightlag(TB2,X,center=TRUE,normalization="no"),

family=binomial)

###Summarise model output

summary(m1)

**Example 2. Using an exponential random graph model to study the link between bovine tuberculosis infection and social behaviour in European badgers *Meles meles*.**

A simple exponential random graph model (ERGM) was applied to a contact network constructed using proximity logging data collected in a population of badgers in Southwest England in 2009/2010 (see Weber et al 2013a). For the purposes of this example an association matrix based on the duration of social contacts over the whole study period was used (June 2009 – May 2010), and all collared badgers (n=51) were included regardless of the duration of their datastream (same data as previous example). No attempt was made to control for this, as these data were just used to provide an example of the use of the technique.

We fitted a model to the binary network to provide a simple example. The code used to run this example is provided below. After inputting the network data, and the individual-level data used in the model, we ran a model containing terms pertaining to the disease status of each individual (individuals were determined to be infected if they had previously tested positive by one of two diagnostic tests – see above), sex and the social group they belonged (as defined by a multi-level community detection algorithm applied to the full dataset – described more fully in the previous example) to as well as a structural term (kstar1) that modelled the expected number of interactions for each individual. The use of this term could be considered similar to that of an intercept in a conventional GLM/GLMM framework.

Terms involving social group were added to control for known effects of social group, and to control for any bias caused by variation in group size or the distribution of proximity loggers within the population. We predicted that there would be a highly significant effect of nodematch(“group”) as a consequence of individuals being more likely to interact with other individuals in their own social group (or network community), and additionally that nodefactor(“group”) would reveal (and control for) differences between groups in the typical number of interactions formed by individuals within them. We expected that bTB positive badgers would interact with more other badgers in line with previous results on their social interactions and ranging behaviour (Garnett et al 2005, Weber et al 2013a). Finally, we included terms to explore the influence of sex on the number of interactions, and the tendency of individuals of the same sex to interact together. We hypothesised that there would be little effect of the latter across the whole year, but that there may be differences between the sexes due to differences in social and ranging behaviour (Weber et al 2013a, 2013b). The results are in table S2 below.

***Table S2.*** *Results of an exponential random graph model showing the factors affecting the tendency for social contacts in a population of European Badgers. Significant terms are in bold.*

|  |  |  |  |
| --- | --- | --- | --- |
| Term | Estimate | Standard Error | P value |
| kstar(1) | **-2.09** | **0.26** | **<0.001** |
| TB status (P vs N) | 0.24 | 0.15 | 0.099 |
| Sex (M vs F) | **0.47** | **0.15** | **0.001** |
| Same sex vs Different sex | 0.07 | 0.19 | 0.701 |
| Same group vs. Different Group | **4.10** | **0.24** | **<0.001** |
| Group 2 vs Group 1 | **1.04** | **0.28** | **<0.001** |
| Group 3 vs Group 1 | -0.02 | 0.25 | 0.940 |
| Group 4 vs Group 1 | **0.70** | **0.31** | **0.024** |
| Group 5 vs Group 1 | 0.45 | 0.26 | 0.090 |
| Group 6 vs Group 1 | 0.45 | 0.28 | 0.103 |

As predicted, badgers were much more likely to form interactions with individuals in their own social group (network community), and males were more likely to interact with a greater number of other badgers than females. While the effect of bTB status was (marginally) non-significant the trend was in the expected direction with infected badgers interacting with more individuals than uninfected individuals. The detection of this trend is despite using only binary interaction data and analysing the entire dataset together rather than splitting it into seasons. The similarity of these results to previous findings from this study illustrate the potential of ERGMs as a tool to study wildlife disease.

We then simulated 1000 additional networks using parameters from our model (e.g. Fig. S1) to demonstrate the potential of using ERGM simulations. It can be seen that our model captures the modular structure of the network well but not overall population structure. This is highly likely to be caused by the lack of inclusion of a spatial constraint on interactions. This could be included as a dyadic covariate, with the probability of an edge occurring declining as the distance between two individual’s main setts increased.



***Figure S1.*** *A comparison of the observed badger social network (a) with four networks simulated using our simple exponential random graph model (b-e).*

***R code:***

##set working directory##

#load packages required

require(tnet)

require(network)

require(ergm)

#the following code reads in the adjacency matrix as a dataframe and

#converts it into an adjacency matrix containing only individuals that

#have interacted with other individuals during that period

FULL<-read.csv("overallnetwork.csv")

names<-FULL[,1]

FULL<-FULL[,2:ncol(FULL)]

colnames(FULL)<-rownames(FULL)<-names

diag(FULL)<-NA

#the following code turns the association matrix into a network that the

#**network** package is happy with

F.el<-as.tnet(FULL)

F<-network(F.el[,1:2],directed=FALSE)

#this then sets the weight attribute for the network (not used by us in

#this example)

set.edge.attribute(F,"weight",as.vector(F.el[,3]))

#read in covariates (Sex, TB infection status and group membership)

TB<-read.csv("TBstatsF.csv")

#This refers to TB infection at the end of the study

Sex<-read.csv("indivsexes.csv")

groups<-read.csv("Complete Membership.csv")

#and then we set the sex, bTB and group as attributes attached to the nodes

set.vertex.attribute(F,"sex",as.vector(Sex$Sex))

set.vertex.attribute(F,"TB",as.vector(TB[,2]))

set.vertex.attribute(F,"group",as.vector(groups[,2]))

###and now we run the model – in this example it is a model of the binary

network (i.e. edge weights are ignored)###

##We include kstar (equivalent to an intercept term) and group-level terms

#to control for network structure. Both the fact the individuals are more

#likely to interact with other individuals in their group (the nodematch

#term) and groups vary in size/number of collared individuals (nodefactor

#term) are included.

##We include other terms to match our hypotheses – the effect on sex on the

#number of interactions (nodefactor), the tendency for same sex individuals

#to associate (nodematch) and the tendency for bTB infection to affect the

#number of interactions (nodefactor)

Fm1<-ergm ( F ~ kstar(1) #number of interactions for each individual

+nodefactor("group")+nodematch("group") #group-related terms

+nodefactor("sex")+nodematch("sex") #sex-related terms

+nodefactor("TB")) #TB related term

#test convergence of the model – this is best achieved by visual inspection of the markov chains plotted and by using the p values (both the overall and for each term) in the function output

mcmc.diagnostics(Fm1,vars.per.page=10)

#and output the model summary

summary(Fm1)

#now simulate 1000 networks using the model parameters

simnets<-simulate(Fm1,1000,seed=set.seed(1))

**Example 3. Using a stochastic actor-oriented model to explore seasonal dynamics of badger social networks in relation to sex and bovine tuberculosis infection.**

For this illustration of the use of stochastic actor-oriented models (SAOM), we used binary networks constructed separately for summer (June-Aug 2009), autumn (Sep-Nov 2009) and winter (Dec 2009-Feb 2010). Networks contained only the 36 individuals collared during at least part of all three of these time periods. Data on social group membership (as defined previously; using membership of a network communities within the full dataset), sex and bTB infection status on each individual were included. bTB status was included as a dynamic covariate for which the most up to date information on each badger was used for each season. Sex was used as a fixed covariate. Group information was used in two ways: shared group membership was fitted as a dyadic covariate (a 1 for when both individuals were in the same group and a 0 otherwise), and the distance between the setts (or midpoint between main setts for social groups with multiple main setts) of two badgers was also fitted as a dyadic covariate. Social groups were defined using multilevel community detection algorithms as described for our previous example analyses.

We fitted models according to methods suggested previously for animal datasets (Fisher et al. 2017, Ilany et al 2015). Structural terms were fitted initially, and convergence and goodness of fit checked before the model was made complex. We added structural terms to the model as follows: 1) basic model (rate, degree and transitive triads only), 2) basic model + (shared) group membership, 3) basic model + group membership + distance (between setts). To this structural model we then added: 4) a sex effect on degree and assortativity, 5) a bTB effect on degree and assortativity and 6) bTB and sex effects on the rate of network change. We then removed both bTB and sex assortativity from the final model as their removal did not reduce goodness of fit considerably, and the effect size of both variables were small. The estimates from the final model are presented in Table S3 and the goodness of fit of the final model is in Fig. S2.

***Table S2.*** *Results of a stochastic actor-oriented model**of the network dynamics of badgers in relation to bovine tuberculosis infection status and sex. Terms in which the estimate is ~2x the standard error are highlighted in bold. The model enables the rate of network change to be modelled alongside the probabilities of interactions taking place (top four rows are rate terms and bottom six rows effects on contact probability).*

|  |  |  |
| --- | --- | --- |
| Term | Estimate | Standard Error |
| Rate (Summer-Autumn) | 1.40 | 0.34 |
| Rate (Autumn-Winter) | 3.79 | 1.05 |
| Effect of sex on rate (M vs F) | **1.72** | **0.78** |
| Effect of bTB status (Positive vs Negative) on rate | -1.00 | 0.69 |
| Degree | **-2.48** | **0.26** |
| Transitive Triads | **0.47** | **0.08** |
| Group Membership | **0.96** | **0.35** |
| Distance between setts | **-3.13** | **1.07** |
| Sex (M vs F) | 0.06 | 0.33 |
| TB status (Positive vs Negative) | 0.22 | 0.34 |

The results of the SAOM are in Table S3. All structural effects are highly significant (Degree, Transitive Triads, Group Membership and Distance). Of note is the fact that there is a positive effect of group in addition to a distance effect. This indicates that there is an effect of shared social group membership (as determined by network methods) on badger interactions over and above the effect of spatial proximity. Sex and bTB status have no significant relationship with the number of connections in the contact network (although the trend for bTB status is in the expected direction), but sex does have an effect on the rate of network change over time. Male badgers tend to change their social interactions more between seasons than females (potentially due to seasonal changes reproductive behaviour and territoriality having more of an effect). The reasons for this would require further models to explore, but are of potential interest in revealing sex differences in social behaviour and risk of acquiring infection.

***Figure S2.*** *Goodness of fit for the final stochastic actor-oriented model**fitted to the dataset. The observed values for a) the out degree distribution, b) the geodesic distribution and c) a triad census of the network (red points and line) and compared to the results obtained from simulations based on model parameters (density plots). P values are given below the plots. These plots reproduce the default output from RSiena.*

***R code:***

##set directory##

#load required R packages

require(RSiena)

#---------------------------------------

#Functions for geodesic distribution and triad census GoF

#available:

#http://finzi.psych.upenn.edu/library/RSiena/html/sienaGOF-auxiliary.html

GeodesicDistribution<-function(i,data,sims,period,groupName,varName,levls=c(1:5,Inf)

,cumulative=TRUE, ...) {

x<-networkExtraction(i,data,sims,period,groupName,varName)

require(sna)

a<-sna::geodist(x)$gdist

if(cumulative){

gdi <- sapply(levls, function(i){ sum(a<=i) })

}

else{

gdi <- sapply(levls, function(i){ sum(a==i) })

}

names(gdi) <- as.character(levls)

gdi

}

TriadCensus<-function(i,data,sims,wave,groupName,varName,levls=1:16){

unloadNamespace("igraph") # to avoid package clashes

require(sna)

require(network)

x<-networkExtraction(i,data,sims,wave,groupName,varName)

tc<-sna::triad.census(x)[1,levls]

#names are transferred automatically

tc

}

#read in network data

summer<-read.csv("summermatrix.csv")

names<-summer[,1]

summer<-as.matrix(summer[,2:ncol(summer)])

colnames(summer)<-rownames(summer)<-names

autumn<-read.csv("autumnmatrix.csv")

names<-autumn[,1]

autumn<-as.matrix(autumn[,2:ncol(autumn)])

colnames(autumn)<-rownames(autumn)<-names

winter<-read.csv("wintermatrix.csv")

names<-winter[,1]

winter<-as.matrix(winter[,2:ncol(winter)])

colnames(winter)<-rownames(winter)<-names

#turn network data into Siena format

nets<-sienaDependent(array(c(summer,autumn,winter),

dim=c(36,36,3)))

# set up to run most basic/default model

b.dat<-sienaDataCreate(nets)

b.eff<-getEffects(b.dat)

b.algm <- sienaAlgorithmCreate(cond=T,nsub=3,n3=1000)

#run most basic model

b.RS<-siena07(b.algm,data=b.dat,effects=b.eff,batch=F,returnDeps=T)

#summary of model to check convergence

summary(b.RS)

#goodness of fit plot – out degree distribution

s.gof1<-sienaGOF(b.RS,OutdegreeDistribution,varName="nets")

plot(s.gof1,scale=T,center=T)

#goodness of fit plot – geodesic distribution

s.gof1b<-sienaGOF(b.RS,GeodesicDistribution,varName="nets")

plot(s.gof1b,scale=T,center=T)

#goodness of fit plot – triad census

s.gof1c<-sienaGOF(b.RS,TriadCensus,varName="nets")

plot(s.gof1c,scale=T,center=T)

#----------------------------------------------------

#now add group membership to the model

#2 group membership

groups<-read.csv("MembershipSAOM.csv")

groupmem<-matrix(0,nc=36,nr=36)

colnames(groupmem)<-rownames(groupmem)<-colnames(summer)

#this creates a matrix of shared group membership from the

#group data imported

for(i in 1:36){

for(j in 1:36){

if(groups[groups[,1]%in%colnames(groupmem)[i],2]

==groups[groups[,1]%in%colnames(groupmem)[j],2]){

groupmem[i,j]<-1

}

if(groups[groups[,1]%in%colnames(groupmem)[i],2]

!=groups[groups[,1]%in%colnames(groupmem)[j],2]){

groupmem[i,j]<-0

}

}

}

# add group membership matrix as a dyadic covariate

group<-coDyadCovar(groupmem)

#set up to run more complex model

b.dat2<-sienaDataCreate(nets,group)

b.eff2<-getEffects(b.dat2)

#adding roup effect

b.eff2<-includeEffects(b.eff2,X,interaction1="group")

#run more complex model until summary says it has converged

b.RS2<-siena07(b.algm, data=b.dat2, effects=b.eff2, batch=F, prevAns = b.RS, returnDeps=T)

summary(b.RS2)

b.RS3<-siena07(b.algm, data=b.dat2, effects=b.eff2, batch=F, prevAns = b.RS2, returnDeps=T)

summary(b.RS3)

b.RS4<-siena07(b.algm, data=b.dat2, effects=b.eff2, batch=F, prevAns = b.RS3, returnDeps=T)

summary(b.RS4)

#and check the three goodness of fit plots

s.gof2<-sienaGOF(b.RS4,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof2,scale=T,center=T)

s.gof2b<-sienaGOF(b.RS4,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof1b,scale=T,center=T)

s.gof2c<-sienaGOF(b.RS4,TriadCensus,varName="nets")

dev.new()

plot(s.gof1c,scale=T,center=T)

#----------------------------------------------------

#now add the distance between groups to the model

locations<-read.csv("grouplocsSAOM.csv")

#turn information (at sett level) into distance matrix and

#divide through by the max value so that it runs from 0 to 1

dmat<-as.matrix(dist(locations[,3:4]))

dmat<-dmat/max(dmat)

#create and fill a matrix to use for this information in the model

dmat.ind<-matrix(0,nc=36,nr=36)

for(i in 1:36){

for(j in 1:36){

dmat.ind[i,j]<-dmat[groups[i,2],groups[j,2]]

}

}

#add distance matrix as a dyadic covariate

dists<-coDyadCovar(dmat.ind)

#set up to run more complex model

b.dat3<-sienaDataCreate(nets,group,dists)

b.eff3<-getEffects(b.dat3)

#add group and distance effects

b.eff3<-includeEffects(b.eff3,X,interaction1="group")

b.eff3<-includeEffects(b.eff3,X,interaction1="dists")

#run model until convergence

b.RS5<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS4, returnDeps=T)

summary(b.RS5)

b.RS6<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS5, returnDeps=T)

summary(b.RS6)

b.RS7<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS6, returnDeps=T)

summary(b.RS7)

b.RS8<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS7, returnDeps=T)

summary(b.RS8)

b.RS9<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS8, returnDeps=T)

summary(b.RS9)

b.RS10<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS9, returnDeps=T)

summary(b.RS10)

b.RS11<-siena07(b.algm, data=b.dat3, effects=b.eff3, batch=F, prevAns = b.RS10, returnDeps=T)

summary(b.RS11)

#and check goodness of fit

s.gof3<-sienaGOF(b.RS11,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof3,scale=T,center=T)

s.gof3b<-sienaGOF(b.RS11,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof3b,scale=T,center=T)

s.gof3c<-sienaGOF(b.RS11,TriadCensus,varName="nets")

dev.new()

plot(s.gof3c,scale=T,center=T)

#----------------------------------------------------

#now add sex to the model

#read in data

sex<-read.csv("SAOMsexes.csv")

#turn into 0s and 1s to input into model

sex[,2]<-as.numeric(sex[,2])

#add as a fixed covariate

sexes<-coCovar(sex[,2])

#set up to run model

b.dat4<-sienaDataCreate(nets,group,dists,sexes)

b.eff4<-getEffects(b.dat4)

b.eff4<-includeEffects(b.eff4,X,interaction1="group")

b.eff4<-includeEffects(b.eff4,X,interaction1="dists")

#add sex effects including assortativity (egoXaltX)

b.eff4<-includeEffects(b.eff4,altX,egoXaltX,interaction1= "sexes")

#run model into convergence

b.RS12<-siena07(b.algm, data=b.dat4, effects=b.eff4, batch=F, prevAns = b.RS11, returnDeps=T)

summary(b.RS12)

b.RS13<-siena07(b.algm, data=b.dat4, effects=b.eff4, batch=F, prevAns = b.RS12, returnDeps=T)

summary(b.RS13)

b.RS14<-siena07(b.algm, data=b.dat4, effects=b.eff4, batch=F, prevAns = b.RS13, returnDeps=T)

summary(b.RS14)

#check goodness of fit

s.gof4<-sienaGOF(b.RS14,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof4,scale=T,center=T)

s.gof4b<-sienaGOF(b.RS14,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof4b,scale=T,center=T)

s.gof4c<-sienaGOF(b.RS14,TriadCensus,varName="nets")

dev.new()

plot(s.gof4c,scale=T,center=T)

#----------------------------------------------------

#now add in bTB status to the model

#read in data and manipulate into correct format

TB2<-read.csv("SAOMTBstats.csv")

TB2[,2]<-as.numeric(TB2[,2])

TB2[,3]<-as.numeric(TB2[,3])

TB2[,4]<-as.numeric(TB2[,4])

#add as a dynamic covariate

TBs<-varCovar(as.matrix(TB2[,2:4]))

#set up to run model

b.dat5<-sienaDataCreate(nets,group,dists,sexes,TBs)

b.eff5<-getEffects(b.dat5)

b.eff5<-includeEffects(b.eff5,X,interaction1="group")

b.eff5<-includeEffects(b.eff5,X,interaction1="dists")

b.eff5<-includeEffects(b.eff5,altX,egoXaltX,interaction1= "sexes")

#add TB effects (including assortativity)

b.eff5<-includeEffects(b.eff5,altX,egoXaltX,interaction1= "TBs")

#run model until convergence

b.RS15<-siena07(b.algm, data=b.dat5, effects=b.eff5, batch=F, prevAns = b.RS14, returnDeps=T)

summary(b.RS15)

b.RS16<-siena07(b.algm, data=b.dat5, effects=b.eff5, batch=F, prevAns = b.RS15, returnDeps=T)

summary(b.RS16)

b.RS17<-siena07(b.algm, data=b.dat5, effects=b.eff5, batch=F, prevAns = b.RS16, returnDeps=T)

summary(b.RS17)

b.RS18<-siena07(b.algm, data=b.dat5, effects=b.eff5, batch=F, prevAns = b.RS17, returnDeps=T)

summary(b.RS18)

b.RS19<-siena07(b.algm, data=b.dat5, effects=b.eff5, batch=F, prevAns = b.RS18, returnDeps=T)

summary(b.RS19)

#and check goodness of fit plots

s.gof4<-sienaGOF(b.RS19,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof4,scale=T,center=T)

s.gof4b<-sienaGOF(b.RS19,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof4b,scale=T,center=T)

s.gof4c<-sienaGOF(b.RS19,TriadCensus,varName="nets")

dev.new()

plot(s.gof4c,scale=T,center=T)

#----------------------------------------------------

##now add rate effects for TB and sex

#set up to run model

b.dat6<-sienaDataCreate(nets,group,dists,sexes,TBs)

b.eff6<-getEffects(b.dat6)

b.eff6<-includeEffects(b.eff6,X,interaction1="group")

b.eff6<-includeEffects(b.eff6,X,interaction1="dists")

b.eff6<-includeEffects(b.eff6, altX,egoXaltX, interaction1 = "sexes")

b.eff6<-includeEffects(b.eff6, altX,egoXaltX, interaction1 = "TBs")

#add rate effect for sex

b.eff6<-includeEffects(b.eff6, RateX,type="rate", interaction1 = "sexes")

#add rate effect for TB

b.eff6<-includeEffects(b.eff6, RateX,type="rate", interaction1 = "TBs")

#run models until convergence

b.RS20<-siena07(b.algm, data=b.dat6, effects=b.eff6, batch=F, returnDeps=T)

summary(b.RS20)

b.RS21<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS20, batch=F, returnDeps=T)

summary(b.RS21)

b.RS22<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS21, batch=F, returnDeps=T)

summary(b.RS22)

#and check goodness of fit

s.gof6<-sienaGOF(b.RS21,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof6,scale=T,center=T)

s.gof6b<-sienaGOF(b.RS21,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof6b,scale=T,center=T)

s.gof6c<-sienaGOF(b.RS21,TriadCensus,varName="nets")

dev.new()

plot(s.gof6c,scale=T,center=T)

#----------------------------------------------------

#remove assortativity (by sex and TB) from final models

#ste up to run models

b.dat6<-sienaDataCreate(nets,group,dists,sexes,TBs)

b.eff6<-getEffects(b.dat6)

b.eff6<-includeEffects(b.eff6,X,interaction1="group")

b.eff6<-includeEffects(b.eff6,X,interaction1="dists")

#add sex effect (without assortativity)

b.eff6<-includeEffects(b.eff6, altX, interaction1 = "sexes")

#add TB effect (without assortativity)

b.eff6<-includeEffects(b.eff6, altX, interaction1 = "TBs")

b.eff6<-includeEffects(b.eff6, RateX,type="rate", interaction1 = "sexes")

b.eff6<-includeEffects(b.eff6, RateX,type="rate", interaction1 = "TBs")

#run model until convergence

b.RS23<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS22, batch=F, returnDeps=T)

summary(b.RS23)

b.RS24<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS23, batch=F, returnDeps=T)

summary(b.RS24)

b.RS25<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS24, batch=F, returnDeps=T)

summary(b.RS25)

b.RS26<-siena07(b.algm, data=b.dat6, effects=b.eff6,prevAns = b.RS25, batch=F, returnDeps=T)

summary(b.RS26)

#check goodness of fit

s.gof7<-sienaGOF(b.RS23,OutdegreeDistribution,varName="nets")

dev.new()

plot(s.gof7,scale=T,center=T)

s.gof7b<-sienaGOF(b.RS23,GeodesicDistribution,varName="nets")

dev.new()

plot(s.gof7b,scale=T,center=T)

s.gof7c<-sienaGOF(b.RS23,TriadCensus,varName="nets")

dev.new()

plot(s.gof7c,scale=T,center=T)

##-------------------------

##summary of final model

summary(b.RS26)

##-------------------------

**Example 4: Simple temporal exponential random graph models applied to badger contact network data**

We used the same dataset that we used to provide an example for stochastic actor-oriented models to demonstrate how temporal exponential random graph models (TERGM) might be applied to contact/interaction data, where binary networks were constructed separately for summer (June-Aug 2009), autumn (Sep-Nov 2009) and winter (Dec 2009-Feb 2010). Networks contained only the 36 individuals collared during at least part of all three of these time periods. Data on group membership (as defined by the community detection approach used previously), sex and bTB infection status on each individual were included. bTB status was included as a dynamic covariate (achieved by assigning vertex attributes for each network in the list independently), for which the most up to date information on each badger was used for each season. Only using 3 time windows allowed us to fit basic models only (more complicated temporal hypotheses would result in the model being over-parameterised with temporal terms), so we fitted multiple models to provide examples of how key model terms could be included, and how goodness-of-fit can be tested, the details of these are provided below. In our examples we fit models with function btergm(), which use maximum pseudo-likelihood estimation and bootstrapped confidence intervals. It is also possible to use the function mtergm() to use Markov chain Monte Carlo methods to estimate maximum likelihood. The latter is less likely to produce biased estimators (especially for short temporal series of networks), but is slower and more likely to result in degenerate models. For model 1 below we provide a comparison between both approaches in our provided R code. For model 2, the model fitted using mtergm() is degenerate so the output provided was from a model fitted using btergm().

*Model 1: Is there stability to badger social interactions between summer and winter?*

The model included the structural effects edges (see table 1 in the main text), nodematch() and nodefactor() terms for social group (network community) and sex and a nodefactor() term for bTB infection status (equivalent to the terms used in the ERGM fitted example two). Importantly it also included the term memory(type=”autoregression”). This modelled the tendency for interactions to stay the same over time to test the hypothesis that badger interactions would be expected to stay similar over time. Different arguments can be provided to the memory term to change what it models. For example, if type=stability is used the stability of all possible interactions is determined instead. This can be changed in the attached code without changing the qualitative conclusions of the model. We also demonstrate the development of an alternative way to construct model 1, in which the memory effect is coded manually (i.e. using the network from previous time-steps as a time-varying dyadic covariate).

*Model 2: Is there a decline in the likelihood of interactions as individuals become less active in the autumn and winter?*

The model also included the structural effect edges (see table 1 in the main text), nodematch() and nodefactor() terms for social group (network community) and sex, and a nodefactor() term for bTB infection status (same as above, and equivalent to the terms used in example two). However, this time the time-dependent effect included was timecov( transform=function(t) t), with the additional argument providing a function for the effect of time (for example, this could be made quadratic if there was expected to be a peak in interactions during a particular season). This enabled us to test the hypothesis that the probability of interactions occurring might decline as individuals became less active later in the year.

*Model 3: Do the interactions of bTB infected badgers change in a different way between summer and winter?*

Theoretically if a longer time series of networks were available it would be possible to include an interaction between a timecov() argument and a dyadic covariate, for example to test the hypotheses like the one above. We could not run this model in our dataset as we only used network data from three seasons (the spring network was smaller in size and not included). This meant that it was not possible to fit an interaction between the dyadic covariate and timecov() effect without over-parameterising the model (having more terms/hypotheses than observations). Nevertheless we provide the code to illustrate how this can be achieved. For example, one important thing to note is the requirement for a dyadic covariate – in this case we construct a matrix that is 1 if both individuals in a dyad are infected with bTB and 0 otherwise.

*Table S4: Model output from TERGM model 1 (estimated from the btergm() function). Confidence interval provides the lower and upper bounds of the 95% confidence interval around the model estimate. Terms for which the confidence interval does not cross zero are highlighted in bold.*

|  |  |  |
| --- | --- | --- |
| Model term | Estimate | Confidence Interval |
| Edges | **-4.03** | **-5.45, -2.00** |
| Memory | **2.97** | **2.73, 4.28** |
| Sex (M vs F) | 0.11 | -0.57, 0.48 |
| TB status (P vs N) | **-0.27** | **-0.46, -0.19** |
| Same Sex vs Different Sex | 0.04 | -0.32, 0.27 |
| Same Group vs Different Group | **3.09** | **3.09, 3.41** |
| Group 2 vs Group 1 | 0.62 | -0.91, 1.55 |
| Group 3 vs Group 1 | -0.28 | -1.34, 0.13 |
| Group 4 vs Group 1 | 0.40 | -0.50, 0.86 |
| Group 5 vs Group 1 | **-0.47** | **-0.79, -0.47** |
| Group 6 vs Group 1 | 0.67 | -0.43,1.36 |

*Table S5: Model output from TERGM model 2. Confidence interval provides the lower and upper bounds of the 95% confidence interval around the model estimate. Terms for which the confidence interval does not cross zero are highlighted in bold.*

|  |  |  |
| --- | --- | --- |
| Model term | Estimate | Confidence Interval |
| Edges | **-4.13** | **-4.83, -3.43** |
| Timecov(function(t) t) | **-0.17** | **-0.21, -0.15** |
| Sex (M vs F) | **0.39** | **0.22, 0.55** |
| TB status (P vs N) | 0.19 | -0.06, 0.42 |
| Same Sex vs Different Sex | -0.05 | -0.15, 0.02 |
| Same Group vs Different Group | **4.78** | **4.74, 4.87** |
| Group 2 vs Group 1 | **1.10** | **0.74, 1.43** |
| Group 3 vs Group 1 | **-0.29** | **-0.54, 0.07** |
| Group 4 vs Group 1 | **0.54** | **0.35, 0.72** |
| Group 5 vs Group 1 | -0.12 | -0.49, 0.26 |
| Group 6 vs Group 1 | 0.24 | -0.15, 0.61 |

Model 1 revealed that there is indeed greater stability in badger networks than expected by chance and model 2 revealed a significant decline in interactions as individuals become less active between summer and winter. Badgers were much more likely to form interactions with individuals from their own groups (as was found using other models), and some groups formed more interactions than others regardless of the model. However, the inclusion of particular time-dependent terms altered the effect of individual-level covariates. In model 1, bTB infection status had a negative effect while in model 2 males formed more interactions than females. The fact that the inclusion of a lagged stability term altered the effect of bTB infection on social interactions (this can be seen by removing the memory effect from model 1) is potentially of interest as it may suggest that infected and uninfected badgers display differences in interaction stability that result in the differences in these context dependent estimates. It is also perhaps reflective of the seasonal differences found by Weber *et al.* (2013), as the inclusion of a memory term means that the summer network is used as an explanatory term in the model. Weber *et al.* (2013) showed that infected badgers had lower within-group degree in autumn and winter making this result less surprising than it initially seems. The goodness-of-fit plots for models 1 and 2 are presented in Fig. S3, with the box plot representing the output of simulations using the model parameters and the black line the observed dataset. The models fit fairly well, although with room for improvement.

C:\Users\mjs245\Dropbox\BADGERS\Methods Submission\Methods resubmission\Methods Resubmission 2\tergm_gof.tif

***Figure S3.*** *Goodness of fit plots for a) model 1 and b) model 2 for three different network metrics. The black line shows values from the observed networks and the box plots the distributions from 1000 sets of networks simulated using the model parameters. This figure is directly reproduced R output.*

***R code:***

#load packages

require(xergm)

#set directory as required

setwd("X")

#-------------------------------------------------------------

#read in matrices

sum<-read.csv("summermatrix.csv")

names<-sum[,1]

sum<-sign(sum[,2:ncol(sum)])

rownames(sum)<-colnames(sum)<-names

aut<-read.csv("autumnmatrix.csv")

names<-aut[,1]

aut<-sign(aut[,2:ncol(aut)])

rownames(aut)<-colnames(aut)<-names

wint<-read.csv("wintermatrix.csv")

names<-wint[,1]

wint<-sign(wint[,2:ncol(wint)])

rownames(wint)<-colnames(wint)<-names

#-------------------------------------------------------------

#read in TB status

TB<-read.csv("TERGMTBstats.csv")

#and save to have separate info for each season

TBsum<-TB[,c(1,2)]

TBaut<-TB[,c(1,3)]

TBwint<-TB[,c(1,4)]

#Read in Sex

Sex<-read.csv("TERGMsexes.csv")

#read in group membership

groups<-read.csv("MembershipTERGM.csv")

#-------------------------------------------------------------

#Convert matrices into networks

s.net<-network(sum,directed=FALSE)

a.net<-network(aut,directed=FALSE)

w.net<-network(wint,directed=FALSE)

#Create a list of the seasonal networks to use in the models

networks<-list(s.net,a.net,w.net)

#And add sex and groups as fixed vertex attributes to these networks

for(i in 1:length(networks)){

set.vertex.attribute(networks[[i]],

"sex",

as.vector(as.numeric(Sex[,2])))

set.vertex.attribute(networks[[i]],

"group",

as.vector(as.numeric(groups[,2])))

}

#Now add TB infection status as a time-varying covariate to the three #networks in the list

set.vertex.attribute(networks[[1]],

"TB",

as.vector(as.numeric(TBsum[,2])-1))

set.vertex.attribute(networks[[2]],

"TB",

as.vector(as.numeric(TBaut[,2])-1))

set.vertex.attribute(networks[[3]],

"TB",

as.vector(as.numeric(TBwint[,2])-1))

#-------------------------------------------------------------

##############MODEL 1##############

#run model, R gives the number of bootstrap replications - #increasing R makes the model more accurate but slower to run

m1<-btergm(networks~edges+ #structural (like GLM intercept)

memory(type="autoregression")+ #memory term (time dependent)

nodematch("group")+ #same vs different group

nodefactor("sex")+ #sex effect on no. of edges

nodematch("sex")+ #sex assortment

nodefactor("TB")+ #TB effect on no. of edges

nodefactor("group"), #group effect on no. of edges

R=10000)

m1.mcmc<-mtergm(networks~edges+ #structural(like GLM intercept)

memory(type="autoregression")+ #memory term (time dependent)

nodematch("group")+ #same vs different group

nodefactor("sex")+ #sex effect on no. of edges

nodematch("sex")+ #sex assortment

nodefactor("TB")+ #TB effect on no. of edges

nodefactor("group"), #group effect on no. of edges

R=10000)

#summary of model

summary(m1)

#summary of MCMC estimated model

Summary(m1.mcmc)

#check goodness of fit

gof.m1 <- gof(m1,

nsim = 1000,

statistics = c(esp, geodesic, deg))

plot(gof.m1,cex.main=2,cex.lab=1.5)

#OPTIONAL - look at model without memory effect to see how TB effect changes

m1a<-btergm(networks~edges+

nodematch("group")+

nodefactor("sex")+

nodematch("sex")+

nodefactor("TB")+

nodefactor("group"),

R=10000)

######MODEL 1 ALTERNATIVE

#set up response network (AUT-WINT)

networks2<-list(networks[[2]],networks[[3]])

for(i in 1:length(networks2)){

set.vertex.attribute(networks2[[i]],

"sex",

as.vector(as.numeric(Sex[,2])))

set.vertex.attribute(networks2[[i]],

"group",

as.vector(as.numeric(groups[,2])))

}

set.vertex.attribute(networks2[[1]],

"TB",

as.vector(as.numeric(TBaut[,2])-1))

set.vertex.attribute(networks2[[2]],

"TB",

as.vector(as.numeric(TBwint[,2])-1))

#set up time-lagged explanatory network (the manual memory #effect; SUM-AUT)

networks3<-list(networks[[1]],networks[[2]])

for(i in 1:length(networks3)){

set.vertex.attribute(networks3[[i]],

"sex",

as.vector(as.numeric(Sex[,2])))

set.vertex.attribute(networks3[[i]],

"group",

as.vector(as.numeric(groups[,2])))

}

set.vertex.attribute(networks3[[1]],

"TB",

as.vector(as.numeric(TBsum[,2])-1))

set.vertex.attribute(networks3[[2]],

"TB",

as.vector(as.numeric(TBaut[,2])-1))

#run model - identical syntax apart from change memory() to

#edgecov()

m1.alt<-btergm(networks2~edges+

edgecov(networks3)+ #”manual” memory effect

nodematch("group")+

nodefactor("sex")+

nodematch("sex")+

nodefactor("TB")+

nodefactor("group"),

R=10000)

#summary of model to confirm that output is the same

summary(m1.alt)

#-------------------------------------------------------------

##############MODEL 2##############

#run model, R gives the number of bootstrap replications - #increasing R makes the model more accurate but slower to run

m2<-btergm(networks~edges+

nodematch("group")+

nodefactor("sex")+

nodematch("sex")+

nodefactor("TB")+

nodefactor("group")+

timecov(transform=function(t) t),#linear effect, time on edges

R=10000)

#summary of model

summary(m2)

#check goodness of fit

gof.m2 <- gof(m2,nsim=1000,statistics=c(esp, geodesic, deg))

plot(gof.m2,cex.lab=1.5,cex.main=2)

#-------------------------------------------------------------

##############MODEL 3##############

####N.B. this model will not run properly in the provided dataset####

#set up a list of matrices that are 1 when both individuals in

#a dyad are TB positive and 0 otherwise

TB.dy1<-matrix(0,nr=nrow(sum),nc=ncol(sum))

colnames(TB.dy1)<-rownames(TB.dy1)<-colnames(sum)

for(i in 1:nrow(TB.dy1)){

for(j in 1:nrow(TB.dy1)){

if(TBsum[TBsum[,1]==rownames(TB.dy1)[i],2]==1&

TBsum[TBsum[,1]==rownames(TB.dy1)[j],2]==1){

TB.dy1[i,j]<-1

}

}

}

TB.dy2<-matrix(0,nr=nrow(sum),nc=ncol(sum))

colnames(TB.dy2)<-rownames(TB.dy2)<-colnames(sum)

for(i in 1:nrow(TB.dy2)){

for(j in 1:nrow(TB.dy2)){

if(TBaut[TBaut[,1]==rownames(TB.dy2)[i],2]==1&

TBaut[TBaut[,1]==rownames(TB.dy2)[j],2]==1){

TB.dy2[i,j]<-1

}

}

}

TB.dy3<-matrix(0,nr=nrow(sum),nc=ncol(sum))

colnames(TB.dy3)<-rownames(TB.dy3)<-colnames(sum)

for(i in 1:nrow(TB.dy3)){

for(j in 1:nrow(TB.dy3)){

if(TBwint[TBwint[,1]==rownames(TB.dy3)[i],2]==1&

TBwint[TBwint[,1]==rownames(TB.dy3)[j],2]==1){

TB.dy3[i,j]<-1

}

}

}

TB.dy<-list(TB.dy1,TB.dy2,TB.dy3)

#run model - same model as m2 but with the addition of a new

#timecov() term that includes X=TB.dy - i.e. TB.dy is a dummy

#variable effectively here

m3<-btergm(networks~edges+

nodematch("group")+

nodefactor("sex")+

nodematch("sex")+

nodefactor("TB")+

nodefactor("group")+

timecov(transform=function(t) t)+

timecov(TB.dy,transform=function(t) t), #new interaction term

R=10000)

#summary of model - this model will not converge/run properly

#with this data

summary(m3)

#check goodness of fit

gof.m3<-gof(m3,nsim=50,statistics=c(esp,geodesic,deg))

plot(gof.m3)

**References**

Csardi, G. and Nepusz, T. 2006. The igraph software package for complex network research, InterJournal, Complex Systems **1695**. <http://igraph.org>

Fisher, D., Ilany, A., Silk, M. and Tregenza T. 2017. Analysing animal social network dynamics: the potential of stochastic actor-oriented models. *Journal of Animal Ecology*, **86**, 202-212.

Garnett B.T., Delahay R.J. and Roper T.J. 2005. Ranging behaviour of European badgers (*Meles meles*) in relation to bovine tuberculosis (*Mycobacterium bovis*) infection. *Appl. Anim Behav Sci*, **94**, 331-340.

Ilany A., Booms A.S. and Holekamp K.E. 2015. Topological effects of network structure on long-term social network dynamic in a wild mammal. *Ecology Letters*, **18**, 687-695.

Weber N., Carter S.P., Dall S.R., Delahay R.J., McDonald J.L., Bearhop S. and McDonald R.A. 2013a. Badger social networks correlate with tuberculosis infection. *Current Biology*, **23**, R915–R916

Weber N., Bearhop S., Dall S.R.X., Delahay R.J., McDonald R.A. and Carter S.P. 2013b. Denning behaviour of the European badger (*Meles meles*) correlates with bovine tuberculosis infection status. *Behav Ecol Sociobiol*, **67**, 471-479.