HW 7 Report

Yongchao Qiao

Introduction

Social network, and many other types of networks, are dynamic over a period of time, where ERGMs are not suitable to apply. A network can change over time is called the dynamic network. For dynamic network, it can grow (more nodes) or shrink (less nodes), leading to network composition changes and the ties can also change among the network nodes. Social selection and influence are two crucial concepts of dynamic network. Social selection occurs when an actor (vertex) selects or forms a new social tie (edge) with another actor (vertex) who shares some similar or relevant characteristics. Social influence occurs when the behavior of one actor (vertex) is changed to become more similar (or dissimilar) to the behavior of one or more other actors (vertices). SIENA standing for Simulation Investigation for Empirical Network Analysis is special for modeling longitudinal network data which is a type of dynamic network data. Specifically, for longitudinal network data, the dataset has been collected for the same network at several time points and at each time point, there is a network dataset.

In an RSiena model, a network variable is the basic dependent variable with a behavior variable being another type of dependent variable which is a node characteristic that changes over time, a coCovar being a constant node attribute that does not change over time, and a varCovar, on the other hand, being an attribute that does change over time. For effects, "sameX" represents the hypothesis that the likelihood of forming a new tie is higher with a same X status; "egoX" represents the hypothesis that an actor is less/more likely to form a tie based on own X status; "altX" represents the hypothesis that an actor is less/more likely to form a tie based on others' X status; "avSim" represents the hypothesis that likelihood of changing behavior is related to the average similarity of all actors' behavior status; "totSim" represents the hypothesis that likelihood of changing behavior is related to the total similarity of all actors' behavior status; "recip" represents the hypothesis that the tendency for ties is related to the reciprocity and "transTrip" represents the hypothesis that the tendency for ties is related to the transitivity.

I Specify the model in slide #168

1. Create the RSiena variables

We load the four network datasets from Coevole data. Then we use function `as.matrix(get.adjacency())`to get the adjacency matrix of network. For example, for data 1, we use `fr_w1 <- Coevolve$fr_w1` to load the data and use fr_w1 to represent the first network data. And use the function `matw1 <- as.matrix(get.adjacency(fr_w1))` to get and save the adjacency matrix. The similar procedures will be applied to other 3 datasets. Then we use function `fr4wav<-sienaDependent(array( c(matw1, matw2, matw3, matw4), dim=c(37,37,4)), sparse=FALSE)` to create the time-varying dependent network variable. And use `gender <- coCovar(V(fr_w1)$gender,centered = FALSE)` to create the time-stable covariate gender with the same vertex values of network fr_w1 since the option center does not make sense here so we use FALSE here. Finally, the behavior variable is created by function `smokebeh <- sienaDependent(array(c(V(fr_w1)$smoke,V(fr_w2)$smoke,` `V(fr_w3)$smoke,V(fr_w4)$smoke),dim=c(37,4)), type = "behavior")`. After all different types of objects are ready, we combine them into a single RSiena object by function `friend <- sienaDataCreate(fr4wav, smokebeh, gender)`

2. Create the effects of RSiena object

Then we use function `frndeff <- getEffects(friend)`to get the basic effects of the RSiena object. Then after comparing it with the result on slide #169, we find we need to add 7 more effects in the RSiena object. They are the transitivity effect, the "sameX" effect for gender, the "altX" effect for smoke behavior, the "egoX" for smoke behavior, the "sameX" effect for smoke behavior, "avSim" and "totSim" effects for smoke behavior. Then we use function `includeEffects` to realize the goal. And the codes are shown below:

```
frndeff <- includeEffects(frndeff,sameX,interaction1="gender",name="fr4wav")

frndeff <- includeEffects(frndeff,egoX,interaction1="smokebeh",name="fr4wav")

frndeff <- includeEffects(frndeff,altX,interaction1="smokebeh",name="fr4wav")

frndeff <- includeEffects(frndeff,sameX,interaction1="smokebeh",name="fr4wav")

frndeff <- includeEffects(frndeff,avSim,interaction1="fr4wav",name="smokebeh")

frndeff <- includeEffects(frndeff,totSim,interaction1="fr4wav",name="smokebe")

frndeff <- includeEffects(frndeff,transTrip,name="fr4wav")
```

Then we get the same effects with slide #169 as show below:

```
frndeff
```

```
##    name     effectName                          include fix   test
## 1  fr4wav   constant fr4wav rate (period 1) TRUE    FALSE FALSE
## 2  fr4wav   constant fr4wav rate (period 2) TRUE    FALSE FALSE
## 3  fr4wav   constant fr4wav rate (period 3) TRUE    FALSE FALSE
## 4  fr4wav   outdegree (density)             TRUE    FALSE FALSE
## 5  fr4wav   reciprocity                     TRUE    FALSE FALSE
## 6  fr4wav   transitive triplets             TRUE    FALSE FALSE
## 7  fr4wav   same gender                     TRUE    FALSE FALSE
## 8  fr4wav   smokebeh alter                  TRUE    FALSE FALSE
## 9  fr4wav   smokebeh ego                    TRUE    FALSE FALSE
## 10 fr4wav   same smokebeh                   TRUE    FALSE FALSE
## 11 smokebeh rate smokebeh (period 1)        TRUE    FALSE FALSE
## 12 smokebeh rate smokebeh (period 2)        TRUE    FALSE FALSE
## 13 smokebeh rate smokebeh (period 3)        TRUE    FALSE FALSE
## 14 smokebeh smokebeh linear shape           TRUE    FALSE FALSE
## 15 smokebeh smokebeh average similarity     TRUE    FALSE FALSE
## 16 smokebeh smokebeh total similarity       TRUE    FALSE FALSE
```

However, to get the model in slide #168 we still need to add the "altX" and "egoX" effects for gender:

```
frndeff <- includeEffects(frndeff,egoXinteraction1="gender",name="fr4wav")

frndeff <- includeEffects(frndeff,altX,interaction1="gender",name="fr4wa")
```

Then we get the same effects with slide #168 as show below:

```
##    name     effectName                         include fix   test
## 1  fr4wav   constant fr4wav rate (period 1)  TRUE    FALSE FALSE
## 2  fr4wav   constant fr4wav rate (period 2)  TRUE    FALSE FALSE
## 3  fr4wav   constant fr4wav rate (period 3)  TRUE    FALSE FALSE
## 4  fr4wav   outdegree (density)               TRUE    FALSE FALSE
## 5  fr4wav   reciprocity                       TRUE    FALSE FALSE
## 6  fr4wav   transitive triplets               TRUE    FALSE FALSE
## 7  fr4wav   gender alter                      TRUE    FALSE FALSE
## 8  fr4wav   gender ego                        TRUE    FALSE FALSE
## 9  fr4wav   same gender                       TRUE    FALSE FALSE
## 10 fr4wav   smokebeh alter                   TRUE    FALSE FALSE
## 11 fr4wav   smokebeh ego                     TRUE    FALSE FALSE
## 12 fr4wav   same smokebeh                    TRUE    FALSE FALSE
## 13 smokebeh rate smokebeh (period 1)          TRUE    FALSE FALSE
## 14 smokebeh rate smokebeh (period 2)          TRUE    FALSE FALSE
## 15 smokebeh rate smokebeh (period 3)          TRUE    FALSE FALSE
## 16 smokebeh smokebeh linear shape            TRUE    FALSE FALSE
## 17 smokebeh smokebeh average similarity       TRUE    FALSE FALSE
## 18 smokebeh smokebeh total similarity         TRUE    FALSE FALSE
```

II Re-run the model specified in slide #168

Then we use the function `sienaAlgorithmCreate(projname='coevol')`to create the algorithm and function `siena07` to estimate the model with RSiena object generated from data Coevolve, option returnDeps=TRUE, batch=TRUE, verbose=FALSE, useCluster=TRUE, initC=TRUE, nbrNodes=3 in siena07() and set.seed(999). The details are shown below:

```
myalgorithm <- sienaAlgorithmCreate(projname='coevol', seed=999)
RSmod1 <- siena07(myalgorithm,data = friend,effects = frndeff,batch=TRUE,
verbose=FALSE,useCluster=TRUE,initC=TRUE,nbrNodes=3,returnDeps=TRUE)
```

The results are shown below:

```
## Estimates, standard errors and convergence t-ratios
##                                          Estimate   Standard    Convergence
##                                                       Error        t-ratio
## Network Dynamics
##    1. rate constant fr4wav rate (period 1)   1.1535 (0.6114)    0.0256
##    2. rate constant fr4wav rate (period 2)   1.1510 (0.3725)   -0.0278
##    3. rate constant fr4wav rate (period 3)   1.1506 (0.2138)    0.0874
##    4. eval outdegree (density)              -3.1033 (1.8159)    0.0820
##    5. eval reciprocity                       0.7944 (0.4644)    0.0133
##    6. eval transitive triplets               0.0935 (0.0756)    0.0664
```

```
##    7. eval gender alter                         -0.3260 (1.1787)    0.0794
##    8. eval gender ego                            0.3087 (0.6028)    0.0596
##    9. eval same gender                           1.2623 (1.9439)    0.0872
##   10. eval smokebeh alter                        0.7076 (0.3769)    0.0010
##   11. eval smokebeh ego                         -0.0547 (0.4443)    0.0981
##   12. eval same smokebeh                         1.1636 (1.2834)    0.0223
## Behavior Dynamics
##   13. rate rate smokebeh (period 1)              0.2961 (1.0825)     0.1017
##   14. rate rate smokebeh (period 2)              0.3162 (0.6081)     0.0569
##   15. rate rate smokebeh (period 3)              0.3147 (0.5767)     0.0985
##   16. eval smokebeh linear shape                18.4840 (4300.5717)   0.1286
##   17. eval smokebeh average similarity   138.7131 (18520.1485)   -0.3288
##   18. eval smokebeh total similarity     -17.2877 (2056.9832)    -0.1838
```

III Use the first set of simulated data to construct models

1. Transform the simulated data

We can get the first simulated data from the previous model by `RSmod1$sims[[1]][[1]][[1]][[1]]`. In the simulated data, there is only information of the network after each period. Since the simulated data is not in the network form, we still need to transform the simulated data into network form. Then we use the function `el2sim <- RSmod1$sims[[1]][[1]][[1]][[1]]` to get the information of edgelist and `sb2sim <- RSmod1$sims[[1]][[1]][[2]][[1]]` to get the information of smoke behavior. Then use the function `fr_w2_sim <- graph.data.frame(el2,directed = TRUE)` to create the graph and `V(fr_w2_sim)$smoke <- sb2sim` as well as`V(fr_w2_sim)$gender <- V(fr_w2)$gender` to set the values of smoke and gender vertex attributes. It is the same procedure for the network information after period 2 and 3 respectively. Then we can get `fr_w2_sim`, `fr_w3_sim` and `fr_w4_sim` three simulated network datasets.

2. Create the RSiena variables from simulated data

We load the one original and three simulated network datasets from Coevole data. Then we use function `as.matrix(get.adjacency())`to get the adjacency matrix of network. For example, for data 1, we use `fr_w1 <- Coevolve$fr_w1` to load the data and use fr_w1 to represent the first network data. And use the function `matw1 <- as.matrix(get.adjacency(fr_w1))` to get and save the adjacency matrix. The similar procedures will be applied to other 3 datasets. Then we use function `fr4wav_sim<-sienaDependent(array(c(matw1,matw2_sim,matw3_sim,matw4_sim),dim=c(37,37,4)), sparse=FALSE)` to create the time-varying dependent network variable. And use `gender_sim <- coCovar(V(fr_w1)$gender,centered = FALSE)` to create the time-stable covariate gender with the same vertex values of network fr_w1 since the option center does not make sense here so we use FALSE here. to create the time-stable covariate gender with the same vertex values of network fr_w1. Finally, the behavior variable is created by function `smokebeh_sim <- sienaDependent(array(c(V(fr_w1)$smoke,V(fr_w2_sim)$smoke, V(fr_w3_sim)$smoke,V(fr_w4_sim)$smoke),dim=c(37,4)), type = "behavior")`. After all different types of objects are ready, we combine them into a single RSiena object by function `friend_sim <- sienaDataCreate(fr4wav_sim, smokebeh_sim, gender_sim)`.

3. Create the effects of model 1 for RSiena object with simulated data

Then we use function `frndeff_sim <- getEffects(friend_sim)`to get the basic effects of the RSiena object. Then after comparing it with the result on slide #168, we find we need to add 9 more effects in the RSiena object. They are the transitivity effect, the "sameX" effect for gender_sim, the "altX" effect for simulated smoke behavior and gender_sim, the "egoX" for simulated smoke behavior and gender_sim, the "sameX" effect for simulated smoke behavior, "avSim" and "totSim" effects for simulated smoke behavior. Then we use function `includeEffects` to realize the goal. And the codes are shown below:

```
frndeff_sim <- includeEffects(frndeff_sim,sameX,interaction1="gender_sim",name="fr4wav_sim")

frndeff_sim <- includeEffects(frndeff_sim,egoX,interaction1="smokebeh_sim",name="fr4wav_sim")

frndeff_sim <- includeEffects(frndeff_sim,altX,interaction1="smokebeh_sim",name="fr4wav_sim")

frndeff_sim <- includeEffects(frndeff_sim,sameX,interaction1="smokebeh_sim",name="fr4wav_sim")

frndeff_sim <- includeEffects(frndeff_sim,avSim,interaction1="fr4wav_sim",name="smokebeh_sim")

frndeff_sim <- includeEffects(frndeff_sim,totSim,interaction1="fr4wav_si",name="smokebeh_sim")

frndeff_sim <- includeEffects(frndeff_sim, transTrip,name="fr4wav_s")

frndeff_sim <- includeEffects(frndeff_sim,egoX,interaction1="gender_sim",name="fr4wav_sim")

frndeff_sim <- includeEffects(frndeff_sim,altX,interaction1="gender_sim",name="fr4wav_sim")
```

Then we get the effects of model 1 for simulated data as show below:

```
##    name          effectName                              include fix   test
## 1  fr4wav_sim    constant fr4wav_sim rate (period 1) TRUE    FALSE FALSE
## 2  fr4wav_sim    constant fr4wav_sim rate (period 2) TRUE    FALSE FALSE
## 3  fr4wav_sim    constant fr4wav_sim rate (period 3) TRUE    FALSE FALSE
## 4  fr4wav_sim    outdegree (density)                  TRUE    FALSE FALSE
## 5  fr4wav_sim    reciprocity                          TRUE    FALSE FALSE
## 6  fr4wav_sim    transitive triplets                  TRUE    FALSE FALSE
## 7  fr4wav_sim    gender_sim alter                     TRUE    FALSE FALSE
## 8  fr4wav_sim    gender_sim ego                       TRUE    FALSE FALSE
## 9  fr4wav_sim    same gender_sim                      TRUE    FALSE FALSE
## 10 fr4wav_sim    smokebeh_sim alter                   TRUE    FALSE FALSE
## 11 fr4wav_sim    smokebeh_sim ego                     TRUE    FALSE FALSE
## 12 fr4wav_sim    same smokebeh_sim                    TRUE    FALSE FALSE
## 13 smokebeh_sim rate smokebeh_sim (period 1)          TRUE    FALSE FALSE
## 14 smokebeh_sim rate smokebeh_sim (period 2)          TRUE    FALSE FALSE
## 15 smokebeh_sim rate smokebeh_sim (period 3)          TRUE    FALSE FALSE
## 16 smokebeh_sim smokebeh_sim linear shape             TRUE    FALSE FALSE
## 17 smokebeh_sim smokebeh_sim average similarity       TRUE    FALSE FALSE
## 18 smokebeh_sim smokebeh_sim total similarity         TRUE    FALSE FALSE
```

4. Create the effects of model 2 for RSiena object with simulated data

There are two paths to create the effects of model 2. One is that we use function `frndeff_sim2 <- getEffects(friend_sim)` to get the basic effects of the RSiena object. Then after comparing it with the result on slide #172, we find we only need to add 2 more effects in the RSiena object. They are the "sameX" effects for gender_sim and simulated smoke behavior. Then we use function `includeEffects` to realize the goal. And the codes are shown below:

```
frndeff2_sim <- includeEffects(frndeff2_sim,sameX,interaction1="gender_sim",name="fr4wav_sim")

frndeff2_sim <- includeEffects(frndeff2_sim,sameX,interaction1="smokebeh_sim",name="fr4wav_sim")
```

The other one is that we use the results of the effects of model 1 and use function `includeEffects` to delete some effects to meet the requirement of model 2. Then after comparing the result on slide #172 with the result on slide #168, we find we need to delete 7 more effects in the RSiena object. They are the transitivity effect, the "altX" effect for simulated smoke behavior and gender_sim, the "egoX" for simulated smoke behavior and gender_sim, "avSim" and "totSim" effects for simulated smoke behavior. Then we use function `includeEffects` to realize the goal. And the codes are shown below:

```
Frndeff2_sim <- includeEffects(frndeff_sim,egoX,interaction1="smokebeh_sim",name="
fr4wav_sim", include = FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim,altX,interaction1="smokebeh_sim",name=
"fr4wav_sim", include = FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim,avSim,interaction1="fr4wav_sim",name="
smokebeh_sim", include = FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim,totSim,interaction1="fr4wav_si",name="
smokebeh_sim", include = FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim, transTrip,name="fr4wav_s", include =
FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim,egoX,interaction1="gender_sim",name="f
r4wav_sim", include = FALSE)

Frndeff2_sim <- includeEffects(frndeff2_sim,altX,interaction1="gender_sim",name="f
r4wav_sim", include = FALSE)
```

Then we get the effects of model 2 for simulated data as show below:

```
##    name         effectName                            include fix   test
## 1  fr4wav_sim   constant fr4wav_sim rate (period 1) TRUE    FALSE FALSE
## 2  fr4wav_sim   constant fr4wav_sim rate (period 2) TRUE    FALSE FALSE
## 3  fr4wav_sim   constant fr4wav_sim rate (period 3) TRUE    FALSE FALSE
## 4  fr4wav_sim   outdegree (density)                  TRUE    FALSE FALSE
## 5  fr4wav_sim   reciprocity                          TRUE    FALSE FALSE
## 6  fr4wav_sim   same gender_sim                      TRUE    FALSE FALSE
## 7  fr4wav_sim   same smokebeh_sim                    TRUE    FALSE FALSE
## 8  smokebeh_sim rate smokebeh_sim (period 1)         TRUE    FALSE FALSE
```

```
## 9  smokebeh_sim rate smokebeh_sim (period 2)       TRUE    FALSE FALSE
## 10 smokebeh_sim rate smokebeh_sim (period 3)       TRUE    FALSE FALSE
## 11 smokebeh_sim smokebeh_sim linear shape          TRUE    FALSE FALSE
```

5. Estimate the model 1 and model 2

Then we use the function `sienaAlgorithmCreate(projname='coevol_sim1')`to create the algorithm and function `siena07` to estimate the model 1 with RSiena object generated from simulated data, option returnDeps=TRUE, batch=TRUE, verbose=FALSE, useCluster=TRUE, initC=TRUE, nbrNodes=3 in siena07() and set.seed(999). The details for model 1 are shown below:

```
myalgorithm1 <- sienaAlgorithmCreate(  projname='coevol_sim1', seed=999)
RSmod1_sim <- siena07(myalgorithm1,data = friend_sim,effects = frndeff_sim,
batch=TRUE,verbose=FALSE,useCluster=TRUE,initC=TRUE,nbrNodes=3,returnDeps=
TRUE)
```

The results for model 1 are shown below:

```
## Estimates, standard errors and convergence t-ratios
##                                                Estimate   Standard   Convergence
##                                                             Error      t-ratio
## Network Dynamics
##    1. rate constant fr4wav_sim rate (period 1)  1.3155   (0.2255)   -0.0085
##    2. rate constant fr4wav_sim rate (period 2)  3.7087   (0.4918)   -0.1015
##    3. rate constant fr4wav_sim rate (period 3)  3.2133   (0.4326)   -0.0261
##    4. eval outdegree (density)                 -2.8236   (0.3723)    0.0255
##    5. eval reciprocity                          0.7756   (0.1694)    0.0020
##    6. eval transitive triplets                  0.1556   (0.0480)    0.0194
##    7. eval gender_sim alter                    -0.0029   (0.2083)    0.0280
##    8. eval gender_sim ego                       0.1967   (0.2100)    0.0308
##    9. eval same gender_sim                      1.1430   (0.2176)    0.0165
##   10. eval smokebeh_sim alter                   0.1068   (0.2789)   -0.0560
##   11. eval smokebeh_sim ego                     0.2497   (0.2692)    0.0261
##   12. eval same smokebeh_sim                    0.2047   (0.3473)    0.0039
## Behavior Dynamics
##   13. rate rate smokebeh_sim (period 1)         0.2138   (0.1631)   -0.0186
##   14. rate rate smokebeh_sim (period 2)         0.7383   (0.3556)   -0.0143
##   15. rate rate smokebeh_sim (period 3)         2.3657   (1.9074)   -0.0682
##   16. eval smokebeh_sim linear shape            0.1174   (0.4236)    0.0549
##   17. eval smokebeh_sim average similarity      4.8444   (9.4918)    0.0184
##   18. eval smokebeh_sim total similarity       -0.6137   (1.8038)    0.0164
```

Then we use the function `sienaAlgorithmCreate(projname='coevol_sim2')`to create the algorithm and function `siena07` to estimate the model 2 with RSiena object generated from simulated data, option returnDeps=TRUE, batch=TRUE, verbose=FALSE, useCluster=TRUE, initC=TRUE, nbrNodes=3 in siena07() and set.seed(999). The details for model 2 are shown below:

```r
myalgorithm2 <- sienaAlgorithmCreate(projname='coevol_sim2', seed=999)
RSmod2_sim <- siena07(myalgorithm2,data = friend_sim,effects = frndeff2_sim, batch=TRUE,verbose=FALSE,useCluster=TRUE,initC=TRUE,nbrNodes=3,returnDeps=TRUE)
```

The results for model 2 are shown below:

```
## Estimates, standard errors and convergence t-ratios
##                                                   Estimate   Standard   Convergence
##                                                                Error      t-ratio
## Network Dynamics
##    1. rate constant fr4wav_sim rate (period 1)  1.2708  (0.2227)    0.0340
##    2. rate constant fr4wav_sim rate (period 2)  3.5522  ( 0.4450)   -0.0124
##    3. rate constant fr4wav_sim rate (period 3)  3.1825  ( 0.4129)   -0.0164
##    4. eval outdegree (density)                  -2.3531  (0.2482)   -0.0584
##    5. eval reciprocity                           0.9065  (0.1572)   -0.0444
##    6. eval same gender_sim                       1.3749  (0.1933)   -0.0620
##    7. eval same smokebeh_sim                     0.0430  (0.2890)   -0.0608
## Behavior Dynamics
##    8. rate rate smokebeh_sim (period 1)          0.1546  (0.1040)    0.0007
##    9. rate rate smokebeh_sim (period 2)          0.5907  (0.2560)   -0.0079
##   10. rate rate smokebeh_sim (period 3)          2.2181  (2.0580)    0.0580
##   11. eval smokebeh_sim linear shape            -0.1376  (0.3937)    0.0392
```

IV Compare the results

1. Results analysis of two models

Table 1 Results for model 1

		Estimate	Stderror	t-stats	t-ratio
Network Dynamics	1. rate constant fr4wav_sim rate (period 1)	1.3155	0.2255	***5.8337***	-0.0085
	2. rate constant fr4wav_sim rate (period 2)	3.7087	0.4918	***7.5410***	-0.1015
	3. rate constant fr4wav_sim rate (period 3)	3.2133	0.4326	***7.4278***	-0.0261
	4. eval outdegree (density)	-2.8236	0.3723	***-7.5842***	0.0255
	5. eval reciprocity	0.7756	0.1694	***4.5785***	0.0020
	6. eval transitive triplets	0.1556	0.0480	***3.2416***	0.0194
	7. eval gender_sim alter	-0.0029	0.2083	-0.0139	0.0280
	8. eval gender_sim ego	0.1967	0.2100	0.9366	0.0308
	9. eval same gender_sim	1.1430	0.2176	***5.2527***	0.0165
	10. eval smokebeh_sim alter	0.1068	0.2789	0.3829	-0.0560
	11. eval smokebeh_sim ego	0.2497	0.2692	0.9275	0.0261
	12. eval same smokebeh_sim	0.2047	0.3473	0.5894	0.0039
Behavior Dynamics	13. rate rate smokebeh_sim (period 1)	0.2138	0.1631	1.3108	-0.0186
	14. rate rate smokebeh_sim (period 2)	0.7383	0.3556	***2.0762***	-0.0143
	15. rate rate smokebeh_sim (period 3)	2.3657	1.9074	1.2402	-0.0682
	16. eval smokebeh_sim linear shape	0.1174	0.4236	0.2771	0.0549
	17. eval smokebeh_sim average similarity	4.8444	9.4918	0.5103	0.0184

	18. eval smokebeh_sim total similarity	-0.6137	1.8038	-0.3402	0.0164

For the results, the RSiena manual suggests that absolute values less than 0.10 indicate excellent convergence, and absolute values less than 0.15 are reasonable. Here we see that all of the parameters except rate constant fr4wav_sim rate (period 2), have excellent convergence and the parameter of rate constant fr4wav_sim rate (period 2) shows only reasonable convergence.

The significance of these evaluation function weights can be determined by dividing the estimates by their standard errors. These are distributed as t-statistics, so any absolute values greater than 2 are significant at the 0.05 significance level.

For our results, we can see the t-stats of three rates in network dynamics, outdegree, reciprocity, transitivity, same gender_sim and rate of smokebeh_sim (period 2) are greater than 2 so these corresponding estimates are significant.

Specifically, we can see that our friendship formation is more likely with people who have the same gender. Conversely, it appears that the main effects of ego smoking and gender, same smoking, alter smoking and gender are not significant predictors of tie formation. Outdegree, transitivity and reciprocity are significant structural predictors.

Table 2 Results for model 2

		Estimate	Stderror	t-stats	t-ratio
Network Dynamics	1. rate constant fr4wav_sim rate (period 1)	1.2708	0.2227	***5.7063***	0.0340
	2. rate constant fr4wav_sim rate (period 2)	3.5522	0.4450	***7.9824***	-0.0124
	3. rate constant fr4wav_sim rate (period 3)	3.1825	0.4129	***7.7076***	-0.0164
	4. eval outdegree (density)	-2.3531	0.2482	***-9.480***	-0.0584
	5. eval reciprocity	0.9065	0.1572	***5.7665***	-0.0444
	6. eval same gender_sim	1.3749	0.1933	***7.1127***	-0.0620
	7. eval same smokebeh_sim	0.0430	0.2890	0.1487	-0.0608
Behavior Dynamics	8. rate rate smokebeh_sim (period 1)	0.1546	0.1040	1.4865	0.0007
	9. rate rate smokebeh_sim (period 2)	0.5907	0.2560	***2.3074***	-0.0079
	10. rate rate smokebeh_sim (period 3)	2.2181	2.0580	1.0777	0.0580
	11. eval smokebeh_sim linear shape	-0.1376	0.3937	-0.349	0.0392

For the results, the RSiena manual suggests that absolute values less than 0.10 indicate excellent convergence, and absolute values less than 0.15 are reasonable. Here we see that all of the parameters have excellent convergence.

The significance of these evaluation function weights can be determined by dividing the estimates by their standard errors. These are distributed as t-statistics, so any absolute values greater than 2 are significant at the 0.05 significance level.

For our results, we can see the t-stats of three rates in network dynamics, outdegree, reciprocity, transitivity, same gender_sim and rate of smokebeh_sim (period 2) are greater than 2 so these corresponding estimates are significant.

Specifically, we can see that our friendship formation is more likely with people who have the same gender. Conversely, it appears that the main effects of ego smoking and gender, same smoking, alter smoking and gender are not significant predictors of tie formation. Outdegree, transitivity and reciprocity are significant structural predictors.

2. Results comparison of two models

From the results, we can see three rates in network dynamics, outdegree, reciprocity, transitivity, same gender_sim and rate of smokebeh_sim (period 2) are all significant no matter for model 1 or model 2. Also, the estimates of these significant parameters are similar for these two models. And the results for two models in convergence accessing are similar. That is, all parameters have reasonable convergence. For model 1, all parameters except rate constant fr4wav_sim rate (period 2), have excellent convergence and the parameter of rate constant fr4wav_sim rate (period 2) shows only reasonable convergence. For model 2, all parameters have excellent convergence.

Appendix

```
library(igraph)
library("UserNetR")
library(RSiena)

data(Coevolve) # Load the data
fr_w1 <- Coevolve$fr_w1 # Get the data 1
fr_w2 <- Coevolve$fr_w2 # Get the data 2
fr_w3 <- Coevolve$fr_w3 # Get the data 3
fr_w4 <- Coevolve$fr_w4 # Get the data 4

matw1 <- as.matrix(get.adjacency(fr_w1))# Get the adjacency matraix and tra
nsform it into matrix form storing in matw1
matw2 <- as.matrix(get.adjacency(fr_w2))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw2
matw3 <- as.matrix(get.adjacency(fr_w3))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw3
matw4 <- as.matrix(get.adjacency(fr_w4))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw4
fr4wav<-sienaDependent(array(c(matw1,matw2,matw3,matw4),dim=c(37,37,4)), s
parse=FALSE)# Create thetime-varying dependent network variable.
fr4wav# Show the fr4wav

gender_vect <- V(fr_w1)$gender# Get the gender values of graph fr_w1
gender <- coCovar(gender_vect,centered = FALSE) #create the time-stable cov
ariate gender with the same vertex values of network fr_w1 since the option
 center does not make sense here so we use FALSE here.

smoke <- array(c(V(fr_w1)$smoke, V(fr_w2)$smoke,V(fr_w3)$smoke, V(fr_w4)$s
moke),
dim=c(37,4))#Preparation for Creating the behavior variable


smokebeh <- sienaDependent(smoke,type = "behavior")# Create the behavior va
riable

friend <- sienaDataCreate(fr4wav,smokebeh,gender)#After all different types
 of objects are ready, we combine them into a single RSiena object

print01Report(friend, modelname="sample")# Generate the initial report

frndeff <- getEffects(friend)#get the basic effects of the RSiena object.
frndeff# Show the basic effect
```

```r
frndeff <- includeEffects(frndeff,sameX,
interaction1="gender",name="fr4wav")# Add the "sameX" effect for gender

frndeff <- includeEffects(frndeff,egoX,
interaction1="smokebeh",name="fr4wav")

# add the "egoX" effect for smokebehavior

frndeff <- includeEffects(frndeff,altX,
interaction1="smokebeh",name="fr4wav")

# add the "altX" effect for smokebehavior

frndeff <- includeEffects(frndeff,sameX,
interaction1="smokebeh",name="fr4wav")

# add the "sameX" effect for smokebehavior

frndeff <- includeEffects(frndeff,avSim,
interaction1="fr4wav",name="smokebeh")

# add the "avSim" effect for smokebehavior

frndeff <- includeEffects(frndeff,totSim,interaction1="fr4wav",name="smoke
beh")

# add the "totSim" effect for smokebehavior
frndeff <- includeEffects(frndeff,recip,transTrip,
name="fr4wav")

# add the transitivity effect
frndeff # Show the effect

frndeff <- includeEffects(frndeff,egoX,
interaction1="gender",name="fr4wav")

# add the "egoX" effect for gender

frndeff <- includeEffects(frndeff,altX,
interaction1="gender",name="fr4wav")

# add the "altX" effect for gender
frndeff#SHow the effects

myalgorithm <- sienaAlgorithmCreate(
   projname='coevol',seed =999)# Create the akgorithm and set the seed=999

RSmod1 <- siena07(myalgorithm,data = friend,
```

```r
   effects = frndeff,
   batch=TRUE,
   verbose=FALSE,useCluster=TRUE,
   initC=TRUE,nbrNodes=3,
   returnDeps=TRUE)# Estimate the model

summary(RSmod1)# Show the results of fitting
```

Simulation

fr_w2_sim

```r
el2 <- RSmod1$sims[[1]][[1]][[1]][[1]]# GET the edgelist of simulated netwo
rk for first wave
sb2 <- RSmod1$sims[[1]][[1]][[2]][[1]]# GET the smokebehavior of simulated
network network for first wave
fr_w2_sim <- graph.data.frame(el2,directed = TRUE)
# Create the first wave network
V(fr_w2_sim)$smoke <- sb2 # Set the smokebehavior values
V(fr_w2_sim)$gender <- V(fr_w2)$gender# Set the gender values

fr_w2_sim# SHow the network fr_w2_sim
```

fr_w3_sim

```r
el3 <- RSmod1$sims[[1]][[1]][[1]][[2]]
#GET the edgelist of simulated network for second wave
sb3 <- RSmod1$sims[[1]][[1]][[2]][[2]]
#GET the smokebehavior of simulated network network for second wave
fr_w3_sim <- graph.data.frame(el3,directed = TRUE)
# Create the second wave network
V(fr_w3_sim)$smoke <- sb3# Set the smokebehavior values
V(fr_w3_sim)$gender <- V(fr_w3)$gender# Set the gender values

fr_w3_sim# SHow the network fr_w3_sim
```

fr_w4_sim

```r
el4 <- RSmod1$sims[[1]][[1]][[1]][[3]]
#GET the edgelist of simulated network for third wave
sb4 <- RSmod1$sims[[1]][[1]][[2]][[3]]
#GET the smokebehavior of simulated network network for third wave
fr_w4_sim <- graph.data.frame(el4,directed = TRUE)
# Create the third wave network
V(fr_w4_sim)$smoke <- sb4# Set the smokebehavior values
```

```r
V(fr_w4_sim)$gender <- V(fr_w4)$gender
# Set the gender values
fr_w4_sim# SHow the network fr_w3_sim
```

Construct #168 model

```r
matw1 <- as.matrix(get.adjacency(fr_w1))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw1
matw2_sim <- as.matrix(get.adjacency (fr_w2_sim))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw2_sim
matw3_sim <- as.matrix(get.adjacency (fr_w3_sim))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw3_sim
matw4_sim <- as.matrix(get.adjacency (fr_w4_sim))
# Get the adjacency matraix and transform it into matrix form storing in ma
tw4_sim
fr4wav_sim<-sienaDependent(array(c(matw1,matw2_sim,matw3_sim,matw4_sim),di
m=c(37,37,4)), sparse=FALSE)# Create thetime-varying dependent network vari
able.
fr4wav_sim# Show fr4wav_sim

gender_vect <- V(fr_w1)$gender# Get the gender values of graph fr_w1
gender_sim <- coCovar(gender_vect,centered = FALSE)#create the time-stable
covariate gender with the same vertex values of network fr_w1 since the opt
ion center does not make sense here so we use FALSE here.

smoke_sim <- array(c(V(fr_w1)$smoke, V(fr_w2_sim)$smoke,V(fr_w3_sim)$smok
e, V(fr_w4_sim)$smoke),
dim=c(37,4))


smokebeh_sim <- sienaDependent(smoke_sim,type = "behavior")#Preparation for
 Creating the behavior variable
smokebeh_sim# Create the behavior variable

friend_sim <- sienaDataCreate(fr4wav_sim,smokebeh_sim,gender_sim)#After al
l different types of objects are ready, we combine them into a single RSien
a object

print01Report(friend_sim, modelname="sample")# Show the initial report

frndeff_sim <- getEffects(friend_sim)#Get the basic effects
frndeff_sim# Show the basic efects
```

```r
frndeff_sim <- includeEffects(frndeff_sim,sameX,
interaction1="gender_sim",name="fr4wav_sim")# Add the "sameX" effect for ge
nder_sim

frndeff_sim <- includeEffects(frndeff_sim,egoX,
interaction1="smokebeh_sim",name="fr4wav_sim")# Add the "egoX" effect for s
mokebeh_sim

frndeff_sim <- includeEffects(frndeff_sim,altX,
interaction1="smokebeh_sim",name="fr4wav_sim")# Add the "altX" effect for s
mokebeh_sim

frndeff_sim <- includeEffects(frndeff_sim,sameX,
interaction1="smokebeh_sim",name="fr4wav_sim")# Add the "sameX" effect for
smokebeh_sim

frndeff_sim <- includeEffects(frndeff_sim,avSim,
interaction1="fr4wav_sim",name="smokebeh_sim")# Add the "avSim" effect for
smokebeh_sim

frndeff_sim <- includeEffects(frndeff_sim,totSim,
interaction1="fr4wav_sim",name="smokebeh_sim")# Add the "totSim" effect for
 smokebeh_sim

frndeff_sim <- includeEffects(frndeff_sim,transTrip,name="fr4wav_sim")# Ad
d the "transitivity" effect

frndeff_sim <- includeEffects(frndeff_sim,egoX,
interaction1="gender_sim",name="fr4wav_sim")# Add the "egoX" effect for gen
der_sim

frndeff_sim <- includeEffects(frndeff_sim,altX,
interaction1="gender_sim",name="fr4wav_sim")# Add the "altX" effect for gen
der_sim

frndeff_sim# SHow the effects

frndeff2_sim <- getEffects(friend_sim)# Get the basic effets
frndeff2_sim# Show the basic effets

frndeff2_sim <- includeEffects(frndeff2_sim,sameX,
interaction1="gender_sim",name="fr4wav_sim")# Add the "sameX" effect for ge
nder_sim

##   effectName      include fix   test  initialValue parm
## 1 same gender_sim TRUE    FALSE FALSE          0   0
```

```r
frndeff2_sim <- includeEffects(frndeff2_sim,sameX,
interaction1="smokebeh_sim",name="fr4wav_sim")# Add the "sameX" effect for
smokebeh_sim
```

```
##   effectName        include fix   test  initialValue parm
## 1 same smokebeh_sim TRUE    FALSE FALSE          0   0
```

```r
frndeff2_sim# Show the effects

#frndeff_sim2 <- includeEffects(frndeff_sim2,egoX,interaction1="smokebeh_s
im",name="fr4wav_sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,altX,interaction1="smokebeh_s
im",name="fr4wav_sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,avSim,interaction1="fr4wav_si
m",name="smokebeh_sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,totSim,interaction1="fr4wav_s
im",name="smokebeh_sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,recip,transTrip,name="fr4wav_
sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,egoX,interaction1="gender_sim
",name="fr4wav_sim", include = FALSE)

#frndeff_sim2 <- includeEffects(frndeff_sim2,altX,interaction1="gender_sim
",name="fr4wav_sim", include = FALSE)
```

Estimation

mdoel 1

```r
myalgorithm1 <- sienaAlgorithmCreate(
   projname='coevol_sim1',seed = 999)# Create the algorithm and set seed =
999

RSmod1_sim <- siena07(myalgorithm1,data = friend_sim,
   effects = frndeff_sim,
   batch=TRUE,
   verbose=FALSE,useCluster=TRUE,
   initC=TRUE,nbrNodes=3,
   returnDeps=TRUE)# Run the model
```

```r
summary(RSmod1_sim)# SHow the results of fitting
```

model 2

```r
myalgorithm2 <- sienaAlgorithmCreate(
   projname='coevol_sim2',seed =999)# Create the akgorithm and set the seed
=999

RSmod2_sim <- siena07(myalgorithm2,data = friend_sim,
   effects = frndeff2_sim,
   batch=TRUE,
   verbose=FALSE,useCluster=TRUE,
   initC=TRUE,nbrNodes=3,
   returnDeps=TRUE)# Etimate the model

summary(RSmod2_sim)# SHow the results of fitting
```