Supplemental Materials

Till Stress Do Us Part: On the Interplay Between Perceived Stress and Communication Network Dynamics

by Y. Kalish et al., 2015, *Journal of Applied Psychology*

http://dx.doi.org/10.1037/apl0000023

This online supplement includes four parts: (1) an introduction to SAB models; (2) an example that shows how SAB models calculate parameter estimates; (3) the complete RSiena code used for our analyses, with comments; (4) the complete results presented in a table with visuals, written descriptions of effects, and interpretations of significant effects.

1. An Introduction to Stochastic Actor Based (SAB) Models

Stochastic agent-based (SAB) models are models that express empirically observed changes in network ties and individual attributes as time-aggregated outcomes of a series of individual decisions (Steglich et al., 2010). Inputs to these models are a binary-directed network variable (*x)* representing the communication network, and a discrete behavioral variable *(z)*, representing the attribute (perceived stress), observed for the same actors at (at least 2) discrete time-points.

While data are only collected at discrete time-points, the model assumes that the underlying time parameter is continuous. This means that changes in relations and attributes unfold in arbitrarily-small time steps. The reason for employing a continuous-time model even if the observations are made only at discrete moments (3 in our study) is that SAB models represent the feedback processes that are inherent in network dynamics and in the mutual influence between network and individual behavior. Technically, these changes in network and attributes are the outcome of a Markov process. Thus, the model conditions on the first (network and attribute) observation point. The probabilities of change in the network and in attributes depend, at each moment, on the current combination of network and attribute variables for all actors.

Change in network ties and in attributes is generated by two unobserved, interdependent processes taking place continuously between observation moments. The first is the process of change in network ties—the "selection process," which may be affected by existing network ties as well as by individual attributes. The second is the process of change in attributes—which may be influenced by the attributes of peers as well as by network ties. Both processes act in parallel on the joint state space of network–behavior configurations. In this way, the selection process affects the opportunities and constraints under which the network and attributes influence each other.

SAB models assume that decisions are driven by actors' goals and constraints as expressed by two mathematical functions termed "objective functions". These objective functions are specified separately for network ties and actor attributes. In a utility-based approach, the objective functions are derived directly from the actors' utilities, where observed changes between measurements are modeled as consequences of a series of small changes, interpretable as decisions optimizing the objective functions plus a stochastic error term, discussed below.

At random instants (called microsteps and modeled separately by rate parameters), one probabilistically selected actor gets an opportunity to make one change to either his own network or his own attribute. A network microstep consists of the addition or deletion of one outgoing tie, i.e., one actor can create a tie to a new actor, terminate an existing tie or maintain an existing tie). A behavioral microstep consists of the increase, decrease or maintenance in the attribute score for *z*.

The two objective functions denoted by *f*, equation 1 (network objective function, f^{net}) and equation 2 (behavioral objective function, f^{beh}), determine the relative probabilities of actor *i* to move to the next measured values of the network and the attribute. The frequency at which microsteps occur is determined by stochastic waiting times, denoted by rate functions.

$$f_i^{net}(x,z) = \sum_k \beta_k^{net}\, s_{ik}^{net}(x,z) \qquad (1)$$

$$f_i^{beh}(x,z) = \sum_k \beta_k^{beh}\, s_{ik}^{beh}(x,z) \qquad (2)$$

where $f_i(x,z)$ is the value of the (network or behavioral, respectively) objective function for actor *(i)* given the current set of parameter estimates *(β),* the state of the network *(x)* and the level of the attribute *(z).* The *k* effects, represented as s_{ik} reflect change processes that depend on the network *(x)* and on individual attributes *(z)*. Snijders et al. (2010) provide recommendations as to which effects s_{ik} should be included in SAB models. Both functions contain baseline parameters (‘outdegree’ for network evolution, and a linear and a quadratic ‘shape’ parameter for behavior evolution) to account for network density and basic distributional properties of the attribute variable net of the other effects.

After evaluating all possible changes, an actor makes the change that maximizes both objective functions. These two objective functions translate into probabilities for change in network and attributes for actor *i* (see Ripley et al., 2013, pp. 67 for probability equations). The outcomes of all changes made by all actors are those that are most consistent with the next measurements.

Effects (*s*) that were included in our analysis were those suggested by Snijders et al., (2010) to represent endogenous network processes, as well as effects that pertain to our research hypotheses. The mathematical definition of the effects is provided below.

Effect name	Formula
Effects for network objective function	
Structural (endogenous) effects:	
Outdegree	$\sum_j x_{ij}$
Reciprocity	$\sum_j x_{ij}\, x_{ji}$
Cyclic triads	$\sum_{j,h} x_{ij}\, x_{jh} x_{hi}$
Transitive triplets	$\sum_{j,h} x_{ij}\, x_{ih} x_{jh}$
Indegree-related popularity (sqrt) effect	$\sum_j x_{ij} \sqrt{\sum_h x_{hj}}$

Outdegree-related popularity (sqrt) effect	$\sum_j x_{ij} \sqrt{\sum_h x_{jh}}$
Structural equivalence effect with respect to incoming ties	$\sum_j x_{ij}\, d_{ij}$ With $d_{ij} = \sum_{\substack{h=1 \\ h\neq i,j}}^{n} b_0 - \lvert x_{hi} - x_{hj} \rvert$
Attribute-related effects	
Attribute-alter effect (H3)	$\sum_j x_{ij}\, z_j$
Attribute-ego effect (H1)	$\sum_j x_{ij}\, z_i$
Attribute-similarity effect (H4)	$\sum_j x_{ij}\, (sim_{ij}^{z} - \widehat{sim^z})$ Where $\widehat{sim^z} = \frac{\Delta - \lvert z_{i-}z_j \rvert}{\Delta}$ with $\Delta = max_{ij} \lvert z_i - z_j \rvert$
Effects for the behavioral objective function z_i	
Behavior shape (linear)	z_i
Behavioral shape (quadratic)	z_i^2
Indegree effect	$z \sum_j x_{ji}$
Outdegree effect (H2)	$z_i \sum_j x_{ij}$

Note: *x* is the state of the (communication) network at a particular time; *i*,*j*,*h* are three network actors (individuals). x_{ij} represents a communication tie from actor *i* to actor *j*, which may be observed (x_{ij};=1) or not (x_{ij};=0). *z* represents the attribute (perceived stress)

The interested reader is referred to Snijders (2001) and particularly to Snijders et al. (2010) for a more complete treatment of SAB models, including creation and endowment effects, measuring goodness of fit, and dealing with compositional changes. Ripley et al. (2013) provide an excellent tutorial to running SAB models in RSiena.

2. A Simplified Example of SAB Model Calculations

To better explain the internal calculations of SAB models, we present a simple case, which does not include actor attributes. Consider the communication ties among four actors, denoted *A* to *D,* at two time-points, represented in the two matrices below. A “1” in cell (*i*,*j*) indicates that actor *i* (ego) communicates with actor *j* (alter) at that observed time-point.

	Time 1			
	A	B	C	D
A	0	0	0	1
B	1	0	0	0
C	0	1	0	0
D	1	0	0	0

	Time 2			
	A	B	C	D
A	0	0	0	1
B	0	0	1	0
C	1	1	0	1
D	1	0	0	0

Actor *A* reported communicating with actor *D* at time 1, a relationship that was maintained at time 2. Actor *B* reported communicating with actor *A* at time 1, a communication that was not maintained at time 2, to be replaced by the creation of a new communication tie to actor *C*.

The aim of SAB models is to estimate parameters that best describe the “decision rules” by which actors create, maintain or terminate ties so that the similarities and differences between observed time-points (i.e., the matrix on the left and the one on the right) are explained. To do this, the model breaks the discrete network measurements (matrix 1 and 2) into very small (unobserved) network microsteps, in which randomly selected actors control their outgoing ties (whom they communicate with).

The SAB model randomly selects an actor; let us assume the selected actor is actor *A*. At time 1 actor *A* has a tie only with actor *D*. Actor *A* now “considers” all his options: he can (a) create a new tie to actor *B*, (b) create a new tie to actor *C*, (c) drop his existing tie to *D*, or (d) do nothing. His final “decision” is based on the value of the objective function for each option. Let us assume for the sake of simplicity that the objective function comprises only two effects: an outdegree effect and a reciprocity effect, and let us further assume that initial values for those effects are estimated at (-0.4) and (+1.0) for outdegree and reciprocity, respectively. The meaning of the negative outdegree effect is that each new tie “costs” -0.4, and the positive and significant reciprocity effect means that actors "gain" +1.0 by reciprocating ties or dropping unreciprocated ties.

Let us consider actor *A*'s choices again: If he creates a new tie to actor *B*, the value of his objective function is now (-0.4)+(1.0) = 0.6 because he has created a new tie to an alter (*B*), but that new tie is now reciprocal. The value of a new tie to actor *C* would be (-0.4)+(-1.0)= -1.4, because he created a new tie that is not reciprocal. The termination of a tie to actor *D* would have the objective function value of (+0.4)+(-1.0) because *A* "frees" a tie, yet damages his reciprocity level. Finally, doing nothing (maintaining his tie to *D*) will elicit a value of 0. Each of these values translates into a probability according to the exponential transformation, $e^{0.6}$= 1.82, 0.24, 0.54 and 1, for each of the four options, respectively. This suggests that at this network microstep, and given the parameter values of (-0.4) and (+1) for outdegree and reciprocity, respectively, actor *A* is most likely to create a new tie to *B*.

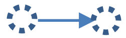
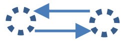
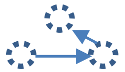
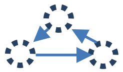
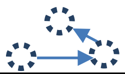
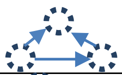
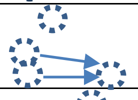
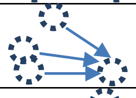
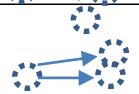
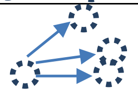
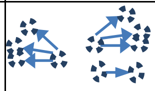
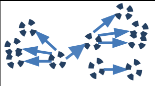
The model then selects another actor (in the next network microstep) and flips through the actors' choices again. Let us assume that given the parameter values for the objective function, actor *C* also decides to sever his existing tie to actor *B*. The model continues to randomly select actors. However, for the sake of simplicity, let us stop after two microsteps. At the end of these two microsteps, actor *A* created a tie to actor *B*, and actor *C* terminated his tie to actor *B*. Yet, in the matrix on the right (representing the next measured time-point), this is clearly not the case, since actor *A* has no tie to actor *B*, and actor *C* maintains his tie to actor *B*. As a result, parameter values are recalibrated so that they better fit the next time-point. In reality, estimation of the parameter estimates for the objective function is an iterative process using the "method of moments" estimation technique (Snijders, 2001), such that the final parameter values are the ones that best describe the "decision rules" with which the network progresses from the matrix on the left to that on the right.

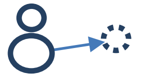
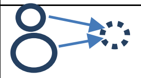
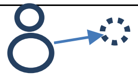
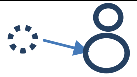
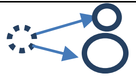
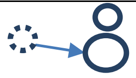
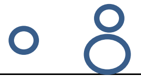
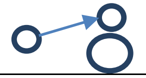
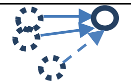
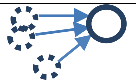
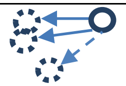
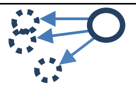
3. RSiena Script for the Evolution of Communication Ties and Perceived Stress

RSiena Syntax	**Explanation**
DATA PREPARATION	
library(RSiena)	# Calls RSiena program
setwd("[…path of folder]")	# set working directory to where the data are:
talk1 <- as.matrix(read.table("talk_t1.dat")) talk2 <- as.matrix(read.table("talk_t2.dat")) talk3 <- as.matrix(read.table("talk_t3.dat"))	# Read data sets. # Structural zeros are defined in the data as "10", as is the row and column information for people who voluntarily withdrew from the assessment bootcamp.
stress <- as.matrix(read.table("stress.dat",na.strings=c("99"))) stress[stress==99] <- NA	# read the stress data; missing values on stress are defined in the data as 99.
talknet <- sienaNet(array(c(talk1, talk2, talk3), dim=c(115,115,3))) stresslvl <- sienaNet(stress [, 1:3], type="behavior")	# Create network data structure and stress data structures as changing attribute:
analysisdata <- sienaDataCreate(talknet, stresslvl) analysisEffects <- getEffects(analysisdata)	#Create an RSIENA object for the analysis.
print01Report(analysisdata,analysisEffects, modelname='descriptives')	# generate initial descriptive report:
MODEL SPECIFICATION	# include effects; rate parameters, outdegree and reciprocity are included by default.
	#include endogenous effects.
analysisEffects <- includeEffects(analysisEffects, transTrip, name="talknet") analysisEffects <- includeEffects(analysisEffects,cycle3, name="talknet") analysisEffects <- includeEffects(analysisEffects,inPopSqrt,outPopSqrt, inStructEq, name="talknet")	#adds transitive triplet #adds three-cycle effect #adds in-popularity (sqrt) effect, out popularity (sqrt) effect and in-structural equivalence effects, respectively
	# includes effects for hypotheses.
analysisEffects <- includeEffects(analysisEffects,egoX, type='creation', interaction1='stresslvl',name="talknet")	#Hypotheis 1a.
analysisEffects <- includeEffects(analysisEffects,egoX, type='endow', interaction1='stresslvl',name="talknet")	#Hypothesis1b.
analysisEffects <- includeEffects(analysisEffects,name="stresslvl",indeg, outdeg, interaction1="talknet")	#Hypothesis 2 (indeg is a required control).
analysisEffects <- includeEffects(analysisEffects,altX, type='creation', interaction1='stresslvl',name="talknet")	#Hypothesis 3a.
analysisEffects <- includeEffects(analysisEffects,altX, type='endow', interaction1='stresslvl',name="talknet")	#Hypotheis 3b.
analysisEffects <- includeEffects(analysisEffects, simX, interaction1='stresslvl')	# Hypothesis 4.
First.modelNEW <- sienaModelCreate(useStdInits=FALSE,projname='model1NEW.first',	# create model for estimation:

cond=TRUE)	
FIRSTNEW.results <- siena07(First.modelNEW,data=analysisdata, effects=analysisEffects, batch=FALSE,verbose=FALSE, returnDeps=TRUE)	# estimate model:
summary(FIRSTNEW.results)	# view summary of results.
GOODNESS OF FIT	
source("sienaGOF.R")	# add the sienaGOF function from package RSienaTest
gof.indegree1 <- sienaGOF(FIRSTNEW.results,IndegreeDistribution,verbose=TRUE)	# Check GOF for indegrees, outdegrees, geodesic distance distribution and triad census.
plot(gof.indegree1,key=0:8)	
gof.outdegree1 <- sienaGOF(FIRSTNEW.results,OutdegreeDistribution,verbose=TRUE)	
plot(gof.outdegree1,key=0:8)	
gof.geodesic1 <- sienaGOF(FIRSTNEW.results,GeodesicDistribution,verbose=TRUE)	
plot(gof.geodesic1,key=1:8)	
gof.triads1 <- sienaGOF(FIRSTNEW.results,TriadCensus,verbose=TRUE)	
triad.keys <- c("003","012","102","021D","021U","021C","111D","111U",	

4. Complete SIENA Model Results for the Co-Evolution of Communication Networks and Perceived Stress

RSIENA effect name	Effect meaning	Visual explanation		Model1	Model2	result interpretation
		Initial configuration at t	Configuration observed at t+1			
Communication network as DV						
Rate function T1-T2				3.18 (0.21)	3.41 (0.23)	Control: number of times a random actor makes a network microstep between T1 and T2
Rate function T2-T3				5.17 (0.37)	5.74 (0.44)	Control: number of times a random actor makes a network microstep between T2 and T3
Outdegree	Creation of a communication tie to a random alter			-1.26 (0.70), p=0.07	-1.21 (0.84), p=0.15	
Reciprocity	Reciprocation of a communication tie between ego and alter			0.37* (0.16), p=0.02	0.41* (0.17), p=0.02	Over time, egos prefer to communicate with an alter who previously communicated with them
3-cycles	Receiving a communication from alter's communication partner			-0.13 (0.08), p=0.11	-0.13 (0.09), p=0.15	
Transitive triplets	Communicating with alter's communication partner			0.69** (0.22), p=0.002	0.68** (0.23), p=0.003	Over time, egos prefer to communicate with alters' communication partners
Indegree-popularity (sqrt)	Popular actors (actors who are sought after by alters) become more popular over time			1.41*** (0.27), p<.001	1.35*** (0.27), p<.001	Popular egos attract even more communication
Outdegree-popularity (sqrt)	Expansive actors (actors who communicate with many alters) become more expansive over time			-0.41 (0.26), p=0.12	-0.40 (0.35), p=0.26	
In-structural equivalence	Popular egos communicate with other popular alters, non-popular egos communicate with non-popular alters			0.57*** (0.12), p<.001	0.51*** (0.13), p<.001	Egos prefer to communicate with alters who are similar in communication popularity

Attribute-ego-creation	The higher ego's attribute level, the more likely he is to create a tie to a new alter				-3.49*** (0.84), p<.001	**Support for H1a** (negative effect): egos with higher levels of stress prefer to form fewer new communication ties
Attribute-ego-endowment	The higher ego's attribute level, the more likely he is to maintain a tie with an existing alter				2.12* (1.05), p=0.04	**Support for H1b**: egos with higher level of stress prefer to maintain their existing ties to alters
Attribute-alter-creation	The higher alter's attribute level, the more new ties are created to him				0.21 (0.16), p=0.19	**No support for H3a** (we expected to find a negative effect)
Attribute-alter-endowment	The higher alter's attribute level, the more ties are maintained with him				-0.69$ (0.35), p=0.05	**Marginal support for H3b (negative effect)**: ties to highly-stressed alters are not maintained
Attribute-similarity	The more similar ego's and alter's levels on an attribute, the more likely a tie between them				0.78* (0.33), p=0.02	**Support for H4**: egos prefer to communicate with alters who have similar levels of stress
Stress as DV						
Rate function T1-T2					0.81 (0.18)	Control: number of times a random actor makes a behavioral microstep between T1 and T2
Rate function T2-T3					0.95 (0.25)	Control: number of times a random actor makes a behavioral microstep between T2 and T3
Linear shape					1.32* (0.56), p=0.02	Control: perceived stress increases over the entire cohort
Quadratic shape					-0.61* (0.26), p=0.02	Control: perceived stress in the cohort centers on a main value
Indegree-attribute	Actors who are becoming more popular become higher on the attribute				0.08 (0.11), p=0.47	
outdegree-attribute	Actors who are becoming more expansive become higher on the attribute				-0.35* (0.16), p=0.02	**Support for H2** (negative effect): the fewer alters ego communicates with, the more stressed ego becomes

Notes: Full circles represent actors with the attribute, stress. Large and small circles represent actors with high and low levels of perceived stress, respectively. Dashed circles denote actors in general, irrespective of their stress level. Ego is always represented on the left of the diagram; alter – on the right.