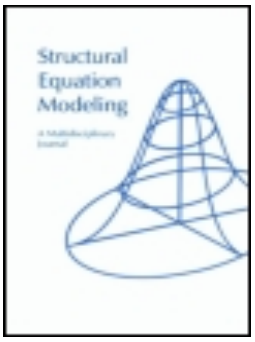


Structural
Equation
Modeling
A Multidisciplinary
Journal

Routledge
Taylor & Francis Group

Structural Equation Modeling: A Multidisciplinary Journal

ISSN: 1070-5511 (Print) 1532-8007 (Online) Journal homepage: http://www.tandfonline.com/loi/hsem20

Automated Selection of Robust Individual-Level Structural Equation Models for Time Series Data

Stephanie T. Lane & Kathleen M. Gates

To cite this article: Stephanie T. Lane & Kathleen M. Gates (2017): Automated Selection of Robust Individual-Level Structural Equation Models for Time Series Data, Structural Equation Modeling: A Multidisciplinary Journal, DOI: 10.1080/10705511.2017.1309978

To link to this article: http://dx.doi.org/10.1080/10705511.2017.1309978

Published online: 01 May 2017.

Structural Equation Modeling: A Multidisciplinary Journal, 00: 1–15, 2017
Copyright © Taylor & Francis Group, LLC
ISSN: 1070-5511 print / 1532-8007 online
DOI: https://doi.org/10.1080/10705511.2017.1309978

Routledge
Taylor & Francis Group

TEACHER'S CORNER

Automated Selection of Robust Individual-Level Structural Equation Models for Time Series Data

Stephanie T. Lane and Kathleen M. Gates
The University of North Carolina at Chapel Hill

In order to analyze intensive longitudinal data collected across multiple individuals, researchers frequently have to decide between aggregating all individuals or analyzing each individual separately. This paper presents an R package, `gimme`, which allows for the automatic specification of individual-level structural equation models that combine group-, subgroup-, and individual-level information. This R package is a complement of the GIMME program currently available via a combination of MATLAB and LISREL. By capitalizing on the flexibility of R and the capabilities of the existing structural equation modeling package lavaan, `gimme` allows for the automated specification and estimation of group-, subgroup-, and individual-level relations in time series data from within a structural equation modeling framework. Applications include daily diary data as well as functional magnetic resonance imaging data.

Keywords: structural equation modeling, R, time series

Across varied domains, researchers collect multivariate data for each individual unit of study across numerous measurement occasions. Frequently referred to as time series data (alternatively, intensive longitudinal data), examples include psychophysiological processes studied using neuroimaging (Beltz et al., 2013), daily diary studies (Shwinski, Smyth, Hofer, & Stawski, 2006) and observational coding of social interactions among dyads (Anzman-Frasca et al., 2013). A primary goal in acquiring these data is to understand temporal processes. Within the neuroimaging community, the process of interest is brain functioning and connectivity, where relations among spatially disparate regions across time offer insight into this phenomenon. Similarly, in daily diary studies, the process of interest might be the dynamics of psychological processes, such as emotion, over time. Methods for analyzing these processes vary greatly, but most have the same underlying goal: identifying the temporal relations that best describe a process over time.

Using time series data affords researchers the ability to pose different questions than those that could be answered with cross-sectional designs. Indeed, analyzing data across time often identifies different patterns of relations than when looking at cross-sectional data (Molenaar, 2004). A key benefit of utilizing time series data is the ability to investigate potential individual differences in patterns of relations. Both theory (Lamiell, 1981; Molenaar, 2008) and emerging results (Anzman-Frasca et al., 2013; Fair, Bathula, Nikolas, & Nigg, 2012; Gates, Molenaar, Iyer, Nigg. & Fair, 2014) suggest that individuals differ in many processes of interest to social scientists. Taken together, understanding these temporal processes on the individual level might assist social science researchers in providing improved diagnostic tools; in turn, this understanding may aid in the development of individually tailored prevention protocols and treatment programs.

Structural equation modeling (SEM) is a popular approach for analyzing time series data, as it can be used to obtain information regarding both lagged and contemporaneous effects frequently found in time series data. Although SEM can be applied at the individual level, two primary concerns preclude researchers from doing so. The first is gathering a sufficient number of observations across time. Should a sufficient number

Correspondence should be addressed to Stephanie T. Lane, CB # 3270, Department of Psychology and Neuroscience, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599. E-mail: slane@unc.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hsem.

be obtained, a secondary concern is that noise will drive the results in individual samples (MacCallum, Roznowski, & Necowitz, 1992). In an attempt to detect signal from noise, the current standard is to conduct group-level analysis by concatenating individual time series data; that is, each individual's data are pasted consecutively below the previous individual's data to arrive at one matrix. Here, the length of the matrix is the total number of observations across time (T_i, for each individual i) times the number of individuals (sample size N), where the number of columns is equal to the number of variables, p. Analysis that enables insight into the relations of multiple variables, such as SEM, is then conducted on these aggregated data to arrive at a nomothetic, or group-level, model that can then be applied to the population.

At its best, aggregating across individuals in this way might aid the researcher in detecting signal from noise. However, combining data sets assumes homogeneity in the relations among variables that explain the processes across individuals. That is, it requires that the data satisfy a strong assumption that one process is sufficient to describe the individuals comprising the sample. However, this assumption likely does not hold in many areas of study within the social sciences.

Thus, both individual-level and group-level approaches are associated with limitations, and it is a daunting task for the researcher to decide whether to analyze each individual separately or aggregate the individuals prior to model selection. This issue motivated the original development of group iterative multiple model estimation (GIMME), a toolbox available in MATLAB that relies on two proprietary programs: MATLAB (The MathWorks, 2010) and LISREL (Jöreskog & Sorbom, 2006), where MATLAB facilitates the user interface and LISREL provides model estimation and optimization. By looking across individuals for patterns of relations among variables at both the group and individual level, GIMME has been found to provide among the most reliable approaches available (Mumford & Ramsey, 2014), particularly in the presence of processes that are heterogeneous across individuals (Gates & Molenaar, 2012).

Requiring the use of two proprietary programs impedes usability for a number of reasons. First, due to licensing restrictions, LISREL cannot be placed on a server. Thus, researchers are unable to utilize cluster computing resources while running GIMME. Second, LISREL is only supported on Windows systems; it is not supported on Linux-based systems. Additionally, for researchers who do not use MATLAB or LISREL in other contexts, purchasing these two programs for the use of GIMME might be cost-prohibitive. Finally, updates in LISREL often change the format and nature of the output, which requires constant updating of the MATLAB shell that reads in the output. Taken together, there exists a great need for one publicly available, stand-alone program for users that can be used on servers and on all platforms. We capitalize on previously shown equivalences in the lavaan and LISREL estimates for SEM (Rosseel, 2012) to arrive at an R version of GIMME that is flexible and does not require the user to purchase any programs. The present project developed, extensively tested, and packaged `gimme` (Lane, Gates, & Molenaar, 2014) for R. This package contains adaptations, improvements, and extensions to the original GIMME MATLAB toolbox, ensuring that the R package performs as well or better than the previously evaluated version.

Importantly, the ability of the `gimme` package to accurately recover group-, subgroup-, and individual-level paths in the presence of heterogeneous data has been demonstrated previously (Gates, Lane, Varangis, Giovanello, & Guskiewicz, 2017; Gates & Molenaar, 2012). Our purpose here is simply to demonstrate the usage of the package and the navigation of the output.

SPECIFICATIONS OF MODEL FOR CURRENT PROGRAM

`gimme` uses the SEM framework to (a) identify the structure of relations among variables of interest, and (b) estimate the weights of these relations. To accommodate the sequential dependence found in time series data, `gimme` estimates the unified SEM (uSEM; also referred to as structural vector autoregression, or SVAR) to obtain relations among variables across time (Chen et al., 2011; Gates, Molenaar, Hillary, Ram, & Rovine, 2010; Kim, Zhu, Chang, Bentler, & Ernst, 2007). The uSEM estimates both lagged (up to a predefined order of Q) and contemporaneous relations (zero order) simultaneously as follows:

$$\eta_t = A\eta_t + \sum_{q=1}^{Q} \phi_q \eta_{t-q} + \zeta_t \tag{1}$$

where $\eta_t, t = 1, 2, ..., T$ contains the manifest p-variate time series (where t ranges across the time-ordered sequence of observations), A contains the (p, p)-dimension matrix of contemporaneous relations among variables (with zeros along the diagonal), ϕ_q contains the (p, p)-dimension matrix of the associations among variables at a lag of q, and ζ_t contains a $p \times 1$ vector white noise process. The parameters in A and ϕ are contained in the B matrix of standard SEM software packages (including LISREL, M*plus* [Múthen & Múthen, 2012], and lavaan). GIMME and `gimme` currently offer the option to have a lag of $q = 1$. All paths where the current time point would predict the previous time point are set to zero; that is, all B paths that would predict η_{t-1} are constrained to zero (Gates et al., 2010). Traditionally, these uSEM models are either applied separately for each individual's data or to data that have been aggregated across individuals. To circumvent the issues that arise from either approach, in a multistep process, `gimme` obtains a group-level model using a process robust to outliers and heterogeneity. This structure is then used as a starting point for identifying relations that exist for the individual in an iterative model search procedure. Previous work has demonstrated that beginning model selection with group-level relations greatly improves the recovery of individual-level paths, alleviating the concern that individual model selection will be driven by noise (Gates & Molenaar, 2012).

The gimme package uses lavaan for the estimation of structural equation models. gimme begins by estimating an empty model (i.e., no estimated relations in the A or ϕ submatrices of the B matrix) across each individual. At the beginning, the researcher may specify whether or not to begin estimation with the autoregressive (AR) paths freed for estimation. The estimates of AR effects indicate the degree to which a given variable at $t-1$ predicts itself at t; these effects are frequently found in time series data. In the gimme package, researchers can also specify additional paths they wish to have estimated. In this way, the model search can be considered a semiconfirmatory model.

Modification indexes are then obtained for each individual's model. Modification indexes indicate the extent to which the model would improve should the corresponding element of the B matrix be freely estimated. Because modification indexes are asymptotically $\chi^2(1)$ distributed, we can conduct significance tests for each element. Modification indexes corresponding to the diagonal of the A matrix are removed, as a variable cannot predict itself in contemporaneous time. Similarly, paths where a variable at t would predict a variable at $t-1$ are removed. Once the null model for each individual has been estimated, gimme proceeds by counting which element, if freed, would significantly (according to a Bonferroni-corrected alpha level) improve model fit for the greatest number of individuals. In the presence of a tie, the element with the highest average modification index across individuals is selected. If this path is significant for ≥75% of individuals, gimme adds this path to every individual's model and continues searching until no path meets this criteria. The cutoff value of >75% is typically found in neuroimaging research (van den Heuvel & Sporns, 2011), and it provides an appropriate heuristic for what constitutes the "majority." However, this cutoff value could be modified by the user. Importantly, if no paths exist that meet this criteria, then none will be chosen during the group-level search procedure. Thus, no group-level model will be forced onto data so heterogeneous that a group-level path would fail to describe individuals comprising the sample (details of this procedure can be found in Gates & Molenaar, 2012).

Once an appropriate group-level model is established, paths that are no longer significant for the majority are pruned in a manner similar to the iterative search procedure. Specifically, the z values associated with each path are evaluated, and if ≤75% of paths are significant, that path will be pruned. In the case of a tie, the path with the lowest average z value will be pruned. Once no paths fit this criteria, the group-level model is established. In this manner, gimme arrives at a group-level model that contains only paths that are significant for the majority of individuals comprising the sample that cannot be swayed by outlier cases. Although all individuals have these paths, the weights are allowed to vary across individuals at all steps of the procedure.

It might be the case that a researcher anticipates not only group- and individual-level paths, but also subgroup-level paths. In this case, gimme allows for the specification of subgroup = TRUE, which utilizes information following the group-level search using a robust community detection method known as Walktrap (Pons & Latapy, 2006). The similarity or adjacency matrix that is used to cluster individuals contains information regarding how similar each pair of individuals is in their temporal models. Specifically, each individual's group-level path and expected parameter change (EPC) estimates are used. EPCs are related to modification indexes but can take both positive and negative values and are normally distributed. gimme obtains a count for each pair of individuals i and j that reflects the number of significant B and EPC estimates that they both have and are in the same direction (i.e., positive or negative). This $N \times N$ adjacency matrix contains counts that indicate the number of temporal effects that the individuals have in common. Walktrap then returns a vector indicating the subgroup (or community) membership of each individual. gimme then proceeds by searching for paths specific to each subgroup in a manner similar to the group-level search. Once subgroup-level paths are added, a similar pruning procedure is then conducted. Finally, once this search is done, a final search is done to ensure that all group-level paths are still significant for the majority of individuals.

Finally, using any group-level and potentially subgroup-level paths as the starting model, individual-level models are then estimated. Modification indexes are again obtained, and the element with the highest modification index exceeding $\chi^2(1)_{,\alpha=.01}$ is freely estimated. The model search for the individual is terminated when an excellent fitting model is obtained as indexed by two of four fit indexes: root mean square error of approximation (RMSEA; Steiger, 1990), nonnormed fit index (NNFI; Bentler & Bonnett, 1980) comparative fit index (CFI; Bentler, 1990) and standardized root meansquare residual (SRMR; Bentler, 1995). For the RMSEA and SRMR, values less than .05 indicate an excellent fit; for the CFI and NNFI, values greater than .95 indicate an excellent fit (Brown. 2006). This approach is similar to the manner in which researchers identify the appropriate number of factors within the SEM framework, and it performed optimally in the original GIMME program. Formally, the final model obtained by gimme can be written as follows:

$$\eta_{t,i} = (A_i + A_i^g + A_i^s)\eta_{t,i} + (\phi_{1,i} + \phi_{1,i}^g + \phi_{1,i}^s)\eta_{t-1,i} + \zeta_{t,i} \quad (2)$$

where, as before, η_t indicates the manifest p-variate time series, A contains the contemporaneous paths, ϕ_1 contains the lag-1 paths, and ζ_t contains the errors in prediction. The subscript i indicates that the parameters in the matrix are unique estimates for individual i for i through N individuals. Matrices with superscript g contain estimates for paths in the group-level structure; that is, for each open path in the g superscript matrices, a path estimate exists for each individual. Matrices with superscript s contain paths for the subgroup-level structure. Please note that the matrices with s superscripts are not included should the researcher specify `subgroup = FALSE`. The matrices without the superscript g or s contain estimates for paths that exist for that individual and are not contained in the group structure. In this way, it is clear that there are two (or potentially three) submodels: one group-level structure, one subgroup-level structure, and one individual-level structure, all of which are estimated at the individual level (Figure 1).

PROGRAM

Considerations for Use

There are a number of considerations for use. First, `gimme` requires that all individuals have the same number of variables, and that these variables are in the same order in the data file or matrix containing each individual's data. The GIMME algorithm has been evaluated with as few as 5 variables and as many as 15 variables, and it has been shown to perform well in this range when recovering data-generating effects. However, as few as 3 variables can be used. An upper limit of no more than 20 variables is suggested given the increase in computation time. Importantly, we recommend at least 60 time points for researchers interested in using `gimme`, although more might be beneficial with a large number of variables (Lane, Gates, Pike, Beltz & Wright, 2016).

Additionally, `gimme` currently does not allow for a lag of greater than 1 to be estimated given the use of a block-Toeplitz structure. That is, the addition of additional lagged variables beyond an order of 1 would quickly become computationally burdensome. Missing data may be handled by `gimme` provided that the mechanism of missingness is random or completely at random. Additionally, it is recommended that the data were measured at equal intervals, an assumption that might be violated with certain forms of ecological momentary assessment data. Finally, `gimme` does not currently allow for data with time-varying effects, therefore, it might not be suitable for situations where effects are thought to vary with respect to time.

Instructions for Use

The `gimme` package has one major function, `gimmeSEM()`, that provides many options to the researcher. This function is used to analyze data using the algorithm described earlier. The program begins by accessing data files stored in a directory created by the user, or by accessing data already stored as named matrices or data frames within a list (explained in detail later) in the R environment.

Regardless of which option is used, there should exist a data set for each individual containing a $T_i \times p$ matrix, where the columns represent variables and the rows represent time. Here, we distinguish T_i because the number of

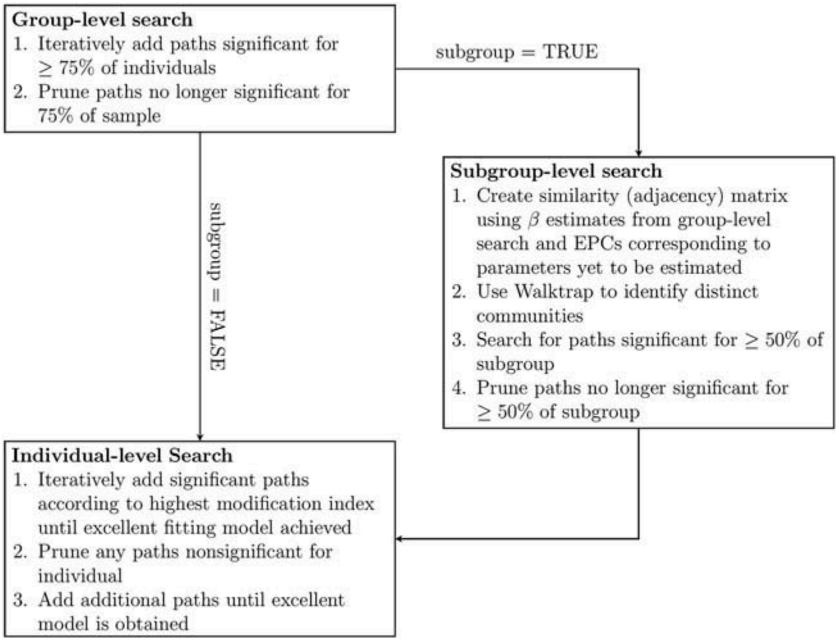

Group-level search
1. Iteratively add paths significant for ≥ 75% of individuals
2. Prune paths no longer significant for 75% of sample
subgroup = TRUE
subgroup = FALSE
Subgroup-level search
1. Create similarity (adjacency) matrix using β estimates from group-level search and EPCs corresponding to parameters yet to be estimated
2. Use Walktrap to identify distinct communities
3. Search for paths significant for ≥ 50% of subgroup
4. Prune paths no longer significant for ≥ 50% of subgroup
Individual-level Search
1. Iteratively add significant paths according to highest modification index until excellent fitting model achieved
2. Prune any paths nonsignificant for individual
3. Add additional paths until excellent model is obtained

FIGURE 1 Model search procedure.

time points can vary across individuals. The number of variables, p, however, cannot vary over individuals. Before analysis begins, the gimmeSEM() function accesses each data file and creates p additional variables to represent the variable at $T - 1$.

The user begins by installing the gimme package and loading the gimme library using the following code:

```
install.packages(“gimme”, dependencies = TRUE)
library(gimme)
```

To apply the gimme algorithm to a set of time series data across N individuals, a call to gimme could be structured as:

```
gimmeSEM(data                        = “C:/example1”,
out                                  = “C:/example1_out”,
sep                                  = “,”,
header                               = FALSE,
ar                                   = TRUE,
plot                                 = TRUE,
subgroup                             = FALSE,
paths                                = NULL)
```

The arguments within this function include data, the path to the directory of data files or the name of the list containing all data matrices; out, the path to the directory where results will be stored; sep, the spacing of the data files using standard R convention (“ for space-delimited, “\t” for tab-delimited, and “,” for comma-delimited); header, a logical indicating whether the data files have a header row; ar, a logical indicating whether or not model search should begin with AR paths open; plot, a logical indicating whether the user desires automatically generated plots from qgraph (Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012) depicting relations among variables; subgroup, a logical indicating whether the user would like the model search to include subgroup-level paths; and paths, an optional argument where the user can specify lavaan-style syntax with paths with which to begin model estimation. All logicals preceding indicate the default values. Thus, where the raw data files are stored in the directory provided for the data argument, the results will be stored in the directory specified by the out argument, the files are comma delimited and contain no header row, and model estimation will begin with autoregressive paths open. By default, plots (plot = TRUE), autoregressive paths are estimated (ar = TRUE), subgroups are not obtained (subgroup = FALSE), and no additional paths are specified with which to begin model estimation (paths = NULL).

To clarify the structure of the data directory, we present an example of the contents using:

```
head(list.files(“C:/example1”, full.names = TRUE))
```

```
[1] “C:/example1/group_1_1.csv” “C:/example1/
group_1_10.csv”
[3] “C:/example1/group_1_11.csv” “C:/example1/
group_1_12.csv”
[5] “C:/example1/group_1_2.csv” “C:/example1/
group_1_3.csv”
```

Here, we see the file path for the first six files in the data directory. All of these files contain comma-separated values (.csv), although text files containing values separated by spaces, tabs, or commas might also be provided. Each file contains an individual’s time series data with length T_i and p variables.

We can view the structure of an individual’s data file using:

```
head(read.csv(“C:/example1/group_1_1.csv”))
```

Here, we see the first six lines of this individual’s data set. Alternatively, the user might bypass the need for the sep and header arguments by providing a list of $T_i \times p$ data matrices directly to the data argument. For clarification, a list is defined as a “generic vector containing other objects.” By “list with named members,” we mean that each matrix within the list is named (e.g., a subject ID). Thus, a list can be provided, which is a vector containing named matrices or data frames for each individual. This use of the data argument could be useful for users who already have all individuals’ time series contained in a single list. If simData is the name of the list containing the named data frame for each individual, we can first see the named elements of the list using

which yields the ordered list of names for the simData object, as shown here:

Thus, we may view the data for individual group_1_1 using simData[[1]]:

Here, we see the first six lines of the named data frame for individual group_1_1. From this demonstration, we see that there is a data file for each individual containing a $T_i \times p$ time series (only the first six time points are presented here for illustrative purposes). If an output directory is specified by the user, multiple output files are produced. For each run of gimmeSEM, two subdirectories are produced, individual

```
V1          V2        V3        V4        V5       V6       V7        V8       V9       V10
–2.540     1.707    -3.374    -0.130    -1.047    2.164    -1.421    -3.205    0.130    -1.634
-1.434     4.197    -1.865     3.018    -1.879    2.461    -0.553    -1.002    1.876    -1.481
-1.180     5.302     0.592     5.310    -1.492    2.798     1.002    -0.416    2.179     0.053
0.881      3.797     0.088     3.211     0.085    1.415    -0.810    -2.463    4.044    -1.412
-0.567     2.682    -0.731     1.951    -1.071    1.366    -1.417    -2.827    3.840    -2.589
-3.587     2.227     0.413     5.580    -2.989    1.985     2.593    -2.036    0.250    -1.177
```

```
names(simData)
```

```
[1]        “group_1_1”        “group_1_10”        “group_1_11”        “group_1_12”
[5]        “group_1_2”        “group_1_3”         “group_1_4”         “group_1_5”
[9]        “group_1_6”        “group_1_7”         “group_1_8”         “group_1_9”
[13]       “group_2_13”       “group_2_14”        “group_2_15”        “group_2_16”
[17]       “group_2_17”       “group_2_18”        “group_2_19”        “group_2_20”
[21]       “group_2_21”       “group_2_22”        “group_2_23”        “group_2_24”
[25]       “group_2_25”
```

```
V1          V2        V3        V4        V5       V6       V7        V8       V9       V10
–2.540     1.707    -3.374    -0.130    -1.047    2.164    -1.421    -3.205    0.130    -1.634
-1.434     4.197    -1.865     3.018    -1.879    2.461    -0.553    -1.002    1.876    -1.481
-1.180     5.302     0.592     5.310    -1.492    2.798     1.002    -0.416    2.179     0.053
0.881      3.797     0.088     3.211     0.085    1.415    -0.810    -2.463    4.044    -1.412
-0.567     2.682    -0.731     1.951    -1.071    1.366    -1.417    -2.827    3.840    -2.589
-3.587     2.227     0.413     5.580    -2.989    1.985     2.593    -2.036    0.250    -1.177
```

and `subgroup` (if `subgroup = TRUE` is selected). Within the individual directory, three output files exist for each individual data file: a matrix containing *B* values for both contemporaneous and lagged relations, a matrix containing standard errors, and a plot summarizing the individual-level paths (if `plot = TRUE`). In this graphic, blue represents negative *B* values, red represents positive *B* weights, and the thickness of the line represents the magnitude of the edge weight. All other files are placed in the main output directory. Table 1 describes the output files and location.

Models are estimated using full information maximum likelihood (FIML). Consequently, we take advantage of the ability of FIML to handle missing data. Although the assumption of row-wise independence is violated in this instance, previous research has indicated that these quasi-maximum likelihood estimates approximate maximum likelihood estimates for AR processes (Hamaker, Dolan, & Molenaar, 2002). The syntax for each individual is iteratively updated on the addition and pruning of new paths using the aforementioned process, and `gimmeSEM()` proceeds by estimating group-, (potentially) subgroup-, and individual-level paths. Output is then directed to an (optional) directory specified by the user, and the user is notified on the completion of a successful search.

Two complementary functions exist that enable the user to compare results from `gimmeSEM` to current standard approaches for arriving at individual-level models and group-level models. First, `indSEM()` identifies the model for each individual independently and does not use shared information across individuals to inform model selection. As noted earlier, one criticism of this approach is that results might be driven by noise (Gates & Molenaar, 2012). No group-level model is generated. An additional function, `aggSEM()`, concatenates all of the individual data files to arrive at one data set. It then runs the `indSEM()` procedure on this data set. This procedure results in a group model and no individual-level paths; thus, no individual-level output or graphs (if `plot = TRUE`) are provided. A summary of the available functions is provided in Table 2.

TABLE 1
Summary of Output Files

Level	*File*	*Contents*
Group	indivPathEstimates	A .csv file containing each element estimated for each individual
	summaryPathCounts	A .csv file containing a breakdown of counts for each path, whether it was estimated at the group, subgroup, or individual level. Also available in matrix form in summaryPathCountMatrix.csv
	summaryFit	A .csv file containing model fit convergence, and subgroup membership for each individual
	summaryPathsPlot	A .pdf containing figure with group-, subgroup-, and individual-level paths for the sample. Black paths are group-level, green paths are subgroup- level, and gray paths are individual-level, where the thickness of the line represents the count
Subgroup	subgroup*k*PathCountsMatrix	A .csv containing counts of relations among lagged and contemporaneous variables for the *k* th subgroup
	subgroup*k*Plot	A .pdf containing a plot of group-, subgroup-, and individual- level paths for the *k* th subgroup. Black represents group-level paths, grey represents individual-level paths, and green represents subgroup-level paths
Individual	*filename*Betas	A .csv file containing the β matrix for individual contained in *filename*. The first p columns contain the φ matrix, and the next p columns contain the A matrix
	*filename*StdErrors	A .csv file containing the SE matrix for the individual contained in *filename*. The first p columns contain the SEs for the φ matrix, and the next p columns contain the SEs for the A matrix
	*filename*Plot	A .pdf containing the individual-level for the individual contained in *filename*. Red paths represent positive weights and blue paths represent negative weights

Note. SE = Standard error.

TABLE 2
Summary of Functions

Function	*Purpose*
gimmeSEM()	Conducts model search for group- (potentially) subgroup-, and individual-level paths to provide unique estimates for all individuals
indSEM()	Conducts model search and estimation of individual-level paths to provide unique structure and estimates for all individuals
aggSEM()	Concatenates data, conducts model search, and estimates only group-level paths

In the model search, the user can declare certain paths that are expected to exist that can be added at the start of estimation. For example, a paths argument for data containing no header row could be defined as shown next, where V2 indicates the variable located at column 2 in the data file. These paths represent that V4 predicts V2 contemporaneously, V3 at $t-1$ predicts V6 at t, and that model estimation should begin with these paths open for all individuals. Example code is shown here where two confirmatory paths are specified and data are read in from a list:

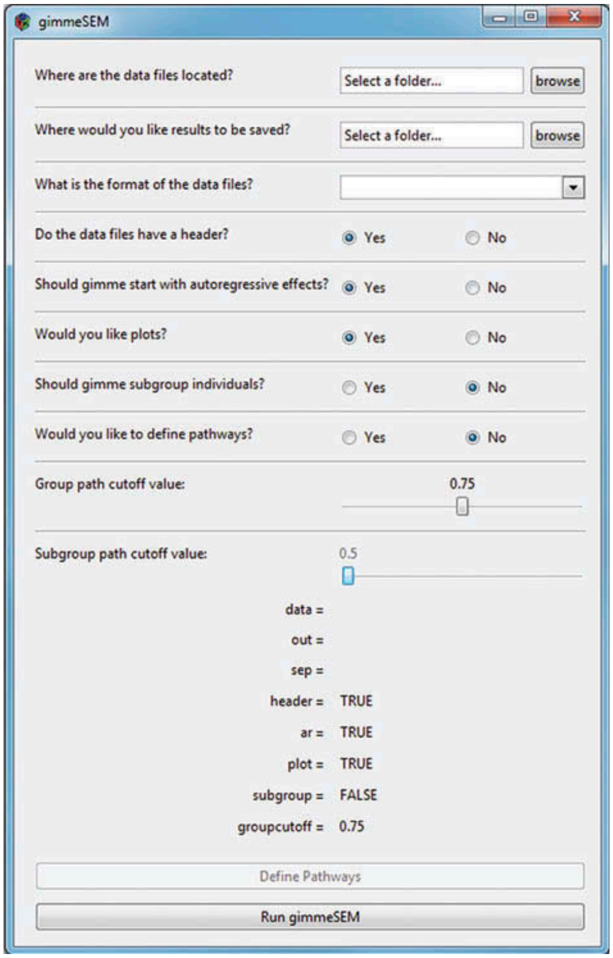
```
paths <- ’V2        ~ V4
V6                  ~ V3lag’

gimmeSEM(data       = “C:/example1”,
out                 = “C:/example1_out”
paths               = paths)
```

Alternatively, gimmeSEM() can be run by launching the graphical user interface (GUI) provided with the gimme package. To launch the GUI, simply type gimmeInteractive() into the R console. The GUI can be viewed in Figure 2.

SIMULATED DATA EXAMPLE

Here, we simulate data to demonstrate the functionality of gimme. These data are generated to mimic characteristics of functional MRI (FMRI) data, which are frequently characterized by large AR effects, multiple paths common to all individuals, and paths unique to an individual. In fMRI data, AR effects are consistently present and large in magnitude due to the lagged nature of the hemodynamic response following neural activation (Huettel, Song, & McCarthy, 2004). Additionally, though the Human Connectome Project aims to establish a “blueprint” of connectivity that exists across all individuals (Van Essen et al., 2013), it has also been acknowledged that a sizable amount of the connectome is thought to be unique to the individual (Barch et al., 2013). Moreover, the individual variability present in connectivity has been shown to be predictive of a host of cognitive and behavioral outcomes (van den Heuvel, Stam,



Where are the data files located?

Where would you like results to be saved?

What is the format of the data files?

Do the data files have a header? ● Yes ○ No

Should gimme start with autoregressive effects? ● Yes ○ No

Would you like plots? ● Yes ○ No

Should gimme subgroup individuals? ○ Yes ● No

Would you like to define pathways? ○ Yes ● No

Group path cutoff value: 0.75

Subgroup path cutoff value: 0.5

data =
out =
sep =
header = TRUE
ar = TRUE
plot = TRUE
subgroup = FALSE
groupcutoff = 0.75

FIGURE 2 Graphical user interface.

Kahn, & Pol, 2009). Therefore, we generate data here characterized by (a) sizable AR effects present across all persons, (b) shared connections across all persons, (c) connections shared across individuals in a subgroup, and (d) connections unique to a given individual.

Using simple algebraic substitution, Equation 1 can be rewritten in the following manner to generate data for one individual with a lag of one time point:

$$\eta_t = (I - A)^{-1}(\phi_{\eta_{t-1}} + \zeta_t) \quad (3)$$

where I is an identity matrix of order (p, p) and ζ_t is a vector of innovations with unit variance. Data of length $T = 200$ were generated for 25 replications (i.e., individuals). All individual replications had the group-level paths depicted in Figure 3a. The group-level paths have a weight of .5 for all individuals unless otherwise dictated by their subgroup membership. There were two equally-sized subgroups comprising the sample. These subgroups differed from each other in that (a) one of the group-level paths was made negative for one subgroup, and (b) each group had two additional subgroup-specific paths. These differences are depicted in Figure 3b and Figure 3c. Finally, at the individual level, individuals had an extra path in both the lagged and contemporaneous matrix at a probability of .01 (not depicted in Figure 3). These data replicate true data seen in the literature by having group, subgroup, and individual-level paths and weights that vary systematically across these levels. These data are loaded with the gimme package, and could be analyzed with the following code:

```
fit <- gimmeSEM(data = simData,
subgroup = TRUE)
```

Note that the output of the `gimmeSEM()` run is directed to an object named `fit` using the assignment operator, <-. After successful completion of the model search, summary information prints to the console:

```
gimme finished running normally
Number of subgroups = 2
Modularity = 0.14043
```

Two main options exist for viewing and interacting with output. First, if the user specifies a file path in the out argument, a copy of all relevant output will be placed in the specified directory. If the directory at the specified file path does not exist, it will be created. The user can also direct the output from `gimmeSEM` to an object (here, `fit`) and use predefined functions to access individual-, subgroup-, and group-level output.

Group-and Subgroup-Level Output

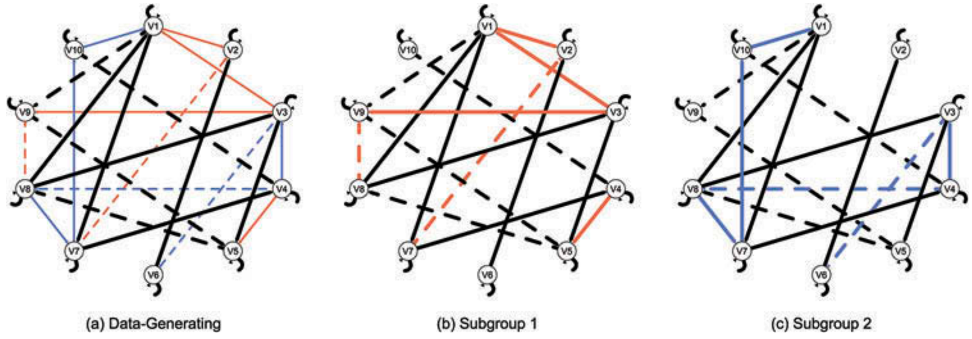
To access group-level and subgroup-level information, we may use a series of functions to inspect the `fit` object where output was directed. Note that output will be available in this object regardless of whether an output directory was specified. For example, to view a path diagram depicting relationships across the entire sample, we can use:

```
plot(fit)
```

This image is depicted in Figure 4a. To view plots specific to each subgroup, we can specify:

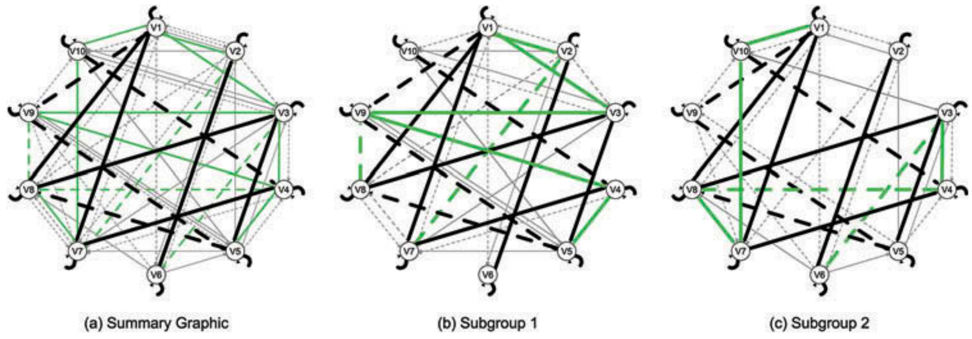
```
plot    (fit, subgroup = 1)
plot    (fit, subgroup = 2)
```

These images are depicted in Figure 4b and 4c.



(a) Data-Generating (b) Subgroup 1 (c) Subgroup 2

FIGURE 3 Data-generating models. Solid lines indicate a contemporaneous effect; dashed lines indicate a lagged effect. Black lines indicate a group-level effect; orange paths indicate an effect unique to subgroup 1; blue lines indicate a path unique to subgroup 2. Width of line corresponds to the count of individuals for whom the path was generated.



(a) Summary Graphic (b) Subgroup 1 (c) Subgroup 2

FIGURE 4 Output from gimmeSEM. Solid lines indicate a contemporaneous effect; dashed lines indicate a lagged effect. Black lines indicate a group-level effect; green paths indicate an effect at the subgroup level; gray paths indicate an effect unique to an individual. Width of line corresponds to the count of individuals for whom the path was estimated.

Similarly, to view a matrix containing the counts representing the number of individuals for whom a path was estimated, we can specify:

```
print (fit)
```

This matrix provides a first glance into the level of heterogeneity present in the sample. Here, with 25 individuals, a count of 25 represents that a path emerged at the grouplevel; that is, it is contained in every individual's final model. Counts of less than 25 indicate that the path was either estimated at the subgroup or the individual level. If a user wishes to view the average across the sample instead, `print(fit, mean = TRUE)` can be used. To partition this matrix by subgroup membership, we can use `print(fit, subgroup = 1)` to view the *count* matrix across individuals in Subgroup 1, and we can use `print(fit, subgroup = 1, mean = TRUE)` to view the *average* coefficient matrix across individuals in Subgroup 1. The latter coefficient matrix represents the average of the coefficient matrices across individuals in Subgroup 1.

Individual-Level Output

We can view an individual-level matrix using the `print()` function, which displays the contemporaneous and lagged coefficient matrices for a given individual. Here, if data were read in from a physical directory, the file name should be the original name without the file extension (e.g., `group_1_2` if the original file were named `group_1_2.csv`). If data were provided in a list format, the file name should be the name of that individual's data matrix in the list. For example, to view the coefficient matrix for `group_1_2` within the `simData` list, the following code can be used:

Here, the rows contain the dependent (predicted) variables and the columns contain the independent variables. For instance, the value of 0.56 at the intersection of `V1` and `V1lag` indicates that `V1lag` predicts `V1` with a β of 0.56. An individual-level path diagram depicting these

```
Please specify a file id for individual coefficient matrix.
Otherwise, a summary count matrix and sample average matrix are presented below.

Lagged Count Matrix for Sample
         V1lag    V2lag    V3lag    V4lag    V5lag    V6lag    V7lag    V8lag    V9lag   V10lag
V1          25        1        0        0        0        0        1        0        0        0
V2           1       25        0        0        0        0       13        0        0        0
V3           1        2       25        0        0        0        0        0        0        0
V4           0        0        1       25        1        0        0        0        0       25
V5           0        0        0        0       25        0        0        0       25        0
V6           2        1       12        0        0       25        0        0        1        0
V7           0        1        0        1        0        0       25        1        1        0
V8           1        2        0       12       25        0        2       25        0        0
V9          25        0        0        0        1        1        0       13       25        0
V10          2        0        1        0        0        0        0        0        1       25

Contemporaneous Count Matrix for Sample
            V1       V2       V3       V4       V5       V6       V7       V8       V9      V10
V1           0        1       13        0        0        0        0       25        0       12
V2          13        0        0        0        1       25        0        0        0        0
V3           0        0        0       12        1        0        0       25        0        2
V4           0        0        0        0        0        0       25        0       13        0
V5           0        0       25       13        0        1        0        0        1        0
V6           0        1        0        0        0        0        0        0        0        0
V7          25        0        1        0        1        0        0       12        0        0
V8           0        0        0        0        0        1        0        0        0        0
V9           0        0       13        0        1        0        0        0        0        0
V10          0        1        1        0        2        0       12        0        0        0
```

```
print(fit, file = “group_1_2”)
```

relations can be obtained using `plot(fit, file = “group_1_2”)`, where red paths indicate positive paths, blue paths indicate negative paths, and the width of the path represents its magnitude, (see Figure 5).

To access the fit indexes, convergence status, and subgroup membership of a given individual, the `fitMeasures` argument can be used within the `print() function`.

which yields the information for a single individual:

From this, we see the fit of the final model for individual `group_1_2` is $\chi^2(117) = 256.79$, $p<.05$, RMSEA $=$ .077, SRMR $=$ .017, NNFI $=$ .981, CFI $=$ .967, suggesting a model with good fit.

EMPIRICAL DATA EXAMPLE

To further illustrate the use and interpretation of the package, we use daily diary data collected by Borkenau and Ostendorf (1998). In this study, 22 individuals were asked for 90 consecutive days to respond to 30 self-report markers of the Big Five (Borkenau & Ostendorf, 1998). For illustrative purposes, we have selected six items pertaining to the Neuroticism dimension of the Big Five. These items include irritable (irr), emotionally stable (emot), calm, bad-tempered (badtemp), resistant (res), and vulnerable (vul). Analyzing these data will allow for insight into the lagged and contemporaneous relationships characterizing intraindividual variation over time.

If all individuals’ data matrices are contained within a list, a `gimme` run can be structured using:

Given that no output directory is specified, all relevant output will be directed to the `fit` object using the assignment operator (<-). We can view the summary matrix containing the count of paths across individuals using:

which displays:

From these results, we see that three sample-level paths emerged contemporaneously: vulnerability predicting emotional stability, vulnerability predicting resistance,

```
Lagged Matrix for group_1_2
          V1lag   V2lag   V3lag   V4lag   V5lag   V6lag   V7lag   V8lag   V9lag   V10lag
V1        0.56    0.00    0.00    0.00    0.00    0.0     0.00    0.000     .00    0.00
V2        0.00    0.58    0.00    0.00    0.00    0.0     0.00    0.00    0.00     0.00
V3        0.00    0.00    0.30    0.00    0.00    0.0     0.00    0.00    0.00     0.00
V4        0.00    0.00    0.00    0.53    0.00    0.0     0.00    0.00    0.00    -0.50
V5        0.00    0.00    0.00    0.00    0.53    0.0     0.00    0.00   -0.39     0.00
V6        0.00    0.00   -0.63    0.00    0.00    0.6     0.00    0.00    0.00     0.00
V7        0.00   -0.89    0.00    0.00    0.00    0.0     0.51    0.00    0.00     0.00
V8        0.00    0.00    0.00   -0.31   -0.43    0.0     0.00    0.62    0.00     0.00
V9       -0.55    0.00    0.00    0.00    0.00    0.0     0.00    0.00    0.63     0.00
V10      -0.48    0.00    0.00    0.00    0.00    0.0     0.00    0.00    0.00     0.56

Contemporaneous Matrix for group_1_2
           V1      V2      V3      V4      V5      V6      V7      V8      V9      V10
V1        0.00    0       0.00    0.00    0       0.00    0.00    0.76    0        0.47
V2        0.00    0       0.00    0.00    0       0.58    0.00    0.00    0        0.00
V3        0.00    0       0.00    0.15    0       0.00    0.00    0.84    0        0.00
V4        0.00    0       0.00    0.00    0       0.00    0.36    0.00    0        0.00
V5        0.00    0       0.57    0.00    0       0.00    0.00    0.00    0        0.00
V6        0.00    0       0.00    0.00    0       0.00    0.00    0.00    0        0.00
V7       -0.62    0       0.00    0.00    0       0.00    0.00    1.03    0        0.00
V8        0.00    0       0.00    0.00    0       0.00    0.00    0.00    0        0.00
V9        0.00    0       0.00    0.00    0       0.00    0.00    0.00    0        0.00
V10       0.00    0       0.00    0.00    0       0.00    0.31    0.00    0        0.00
```

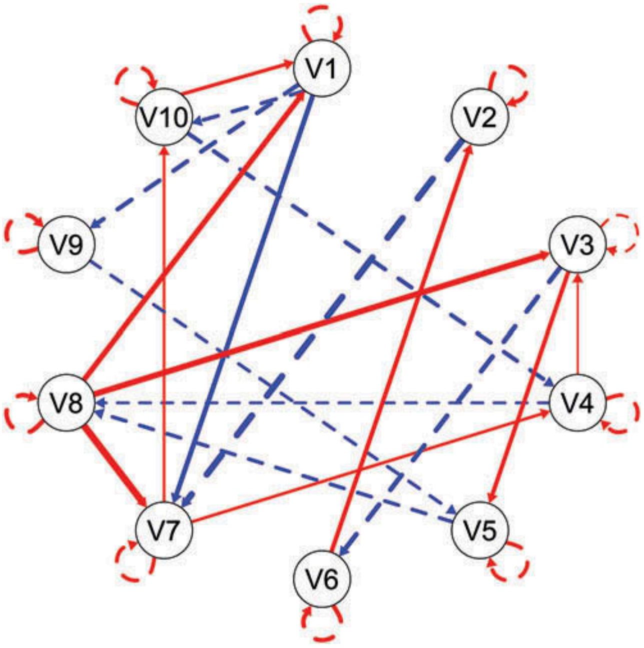


FIGURE 5 Individual-level diagram for “group_1_2” from simData. Solid lines indicate a contemporaneous effect; dashed lines indicate a lagged effect. Red lines indicate a positive effect; blue lines indicate a negative effect. The width of the line corresponds to the magnitude of the effect.

```
print(fit, file = “group_1_2”, fitMeasures = TRUE)
```

```
Fit for file group_1_2
chisq df pval rmsea srmr nnfi cfi status subgroup
256.7882 117 0 0.0773 0.0169 0.9812 0.9695
converged normally 2
```

```
fit <- gimmeSEM(data = borkenau)
```

```
print(fit)
```

and irritability predicting vulnerability. This indicates that these relationships were significant for greater than 75% of the sample. Interestingly, we see that all sample-level paths surfaced contemporaneously. This might be due to a difference in the speed of the process under observation and the time between measurement occasion. That is, when processes occur faster than the rate

```
Please specify a file id for individual coefficient matrix.
 Otherwise, summary count matrix is presented below.

 Lagged Count Matrix for Sample
                irrlag      emotlag      calmlag      badtemplag      reslag      vullag
irr                 22            2            0               1           0           0
emot                 1           22            0               1           0           0
calm                 0            0           22               1           0           0
badtemp              0            0            0              22           0           0
res                  0            0            1               1          22           0
vul                  0            1            0               1           0          22

Contemporaneous Count Matrix for Sample
                   irr         emot         calm         badtemp         res         vul
irr                  0            4            3               5           3           0
emot                 4            0            0               1           4          22
calm                 0           12            0               4           2           5
badtemp              5            7            0               0           1           3
res                  3            3            0               1           0          22
vul                 22            1            0               1           4           0
```

```
print(fit, mean = TRUE)
```

```
Please specify a file id for individual coefficient matrix.
Otherwise, a summary average matrix is presented below.

Lagged Average Matrix for Sample
                irrlag      emotlag      calmlag      badtemplag      reslag      vullag
irr               0.11         0.00         0.00           -0.01        0.00        0.00
emot             -0.01         0.08         0.00           -0.01        0.00        0.00
calm              0.00         0.00         0.05            0.01        0.00        0.00
badtemp           0.00         0.00         0.00            0.10        0.00        0.00
res               0.00         0.00         0.01           -0.01        0.05        0.00
vul               0.00        -0.01         0.00            0.01        0.00        0.05

Contemporaneous Average Matrix for Sample
                   irr         emot         calm         badtemp         res         vul
irr               0.00        -0.01        -0.11            0.11       -0.04        0.00
emot             -0.10         0.00         0.00           -0.02        0.09       -0.36
calm              0.00         0.31         0.00           -0.04        0.03       -0.09
badtemp           0.11        -0.17         0.00            0.00       -0.01        0.07
res              -0.09         0.05         0.00           -0.01        0.00       -0.33
vul               1.46         0.02         0.00            0.01        0.90        0.00
```

of observation, effects could surface contemporaneously (Granger, 1969).

We may view the average of the relationships across individuals using:

Here, the coefficients are averaged across all persons, including individuals for whom a coefficient might have been zero.

Similarly, the coefficients for a given individual can be accessed using their file name (without extension) or the name of their list. Thus, the coefficient matrix for individual “BorkInd4” can be accessed using:

```
print(fit, file = “BorkInd4”)
```

```
Lagged Matrix for BorkInd4
                 irrlag      emotlag      calmlag      badtemplag      reslag      vullag
irr               0.18        0.00         0.00         0.00            0.00        0.00
emot              0.00        0.11         0.00         0.00            0.00        0.00
calm              0.00        0.00         0.12         0.00            0.00        0.00
badtemp           0.00        0.00         0.00         0.24            0.00        0.00
res               0.00        0.00         0.23         0.00            0.17        0.00
vul               0.00        0.00         0.00         0.00            0.00        0.05

Contemporaneous Matrix for BorkInd4
                  irr         emot         calm         badtemp         res         vul
 irr              0.00        0.00         0            0               0           0.00
 emot            -0.29        0.00         0            0               0          -0.32
 calm             0.00        0.55         0            0               0           0.00
 badtemp          0.00       -0.39         0            0               0           0.00
 res              0.00        0.00         0            0               0           0.04
 vul              0.57        0.00         0            0               0           0.00
```

Here, we can see an individual’s coefficients for her final model, composed of group- and individual-level paths. For example, for the group-level path between irritability and vulnerability, we see $\beta = 0.57$. Similarly, for the group-level path between vulnerability and emotional stability, we see $\beta = -0.32$. Importantly, although these group-level paths are estimated across all persons, each person has a unique estimate.

We can also view this using the predefined `plot()` function, where the plot for the same individual can be viewed using:

This plot is displayed in Figure 6.

DISCUSSION

The R package `gimme` described in this article introduces an SEM-based method for identifying group-, subgroup-, and individual-level relations within time series data. This package promises to be useful for researchers analyzing a range of data, from establishing functional connectivity using fMRI data to investigating dynamic processes over time within daily diary data.

The `gimme` package is characterized by a number of strengths. It utilizes the popular and well-maintained lavaan package for estimation. Additionally, given the small number of commands required by the user, as well as the availability of a GUI, `gimme` can be used by both inexperienced and experienced R users. The automatic identification and estimation of models greatly reduces user burden. Importantly, this implementation improves on the original GIMME by offering a community detection based subgrouping procedure, automatically generated summary graphics using the qgraph package (Epskamp et al., 2012), and the ability to begin model estimation with semi confirmatory paths. Additionally, this implementation of GIMME only requires R, not a combination of proprietary software like the original GIMME toolbox.

Multiple aspects of the `gimme` package are open to further development. For example, the current implementation slows considerably when trying to estimate relations among more than 20 variables; however, work is underway to expand the package to allow for the estimation of relations among more variables, including both a measurement and structural model, using an alternative estimation procedure. Another extension could allow for estimation of exogenous variables that have been convolved with a hemodynamic response function for use fMRI data obtained during an event-related design. We encourage users to contribute to our Github page

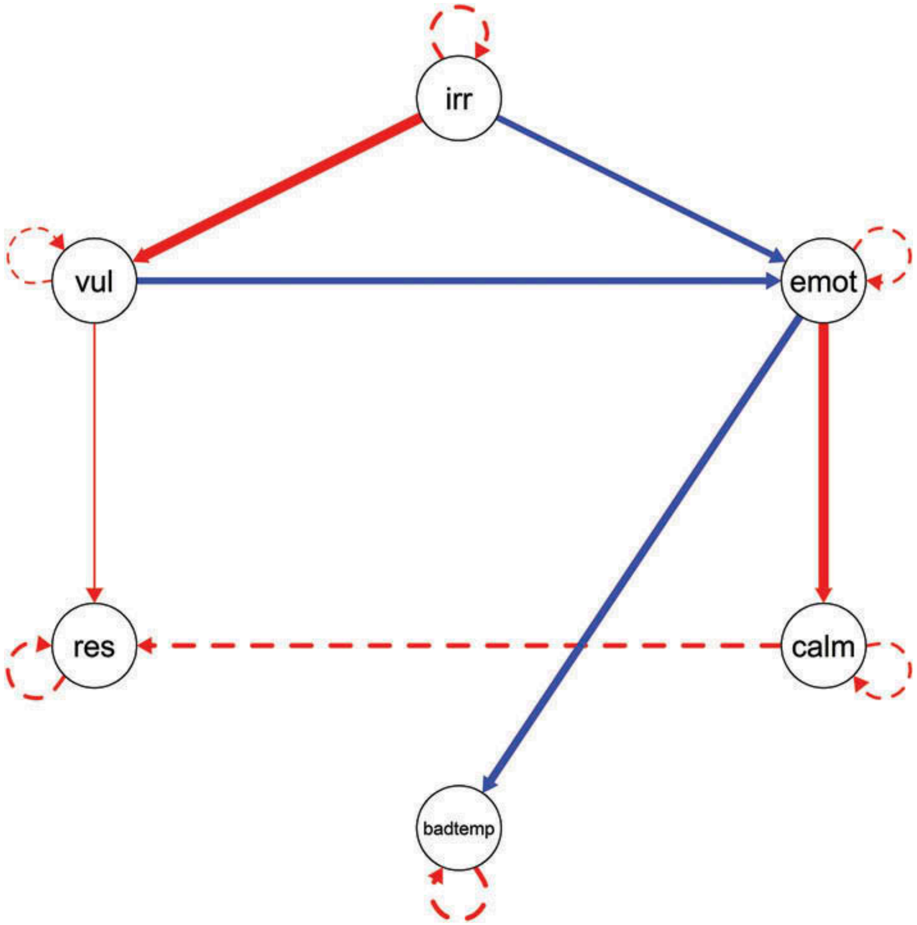


FIGURE 6 Individual-level diagram for "BorkInd4" from empirical example (data from Borkenau & Ostendorf, 1998). Solid lines indicate a contemporaneous effect; dashed lines indicate a lagged effect. Red lines indicate a positive effect; blue lines indicate a negative effect. The width of the line corresponds to the magnitude of the effect.

at `https://github.com/GatesLab/gimme`. In sum, `gimme` represents a flexible, user-friendly package to evaluate individual-level time series data using an automated search procedure rooted in SEM.

REFERENCES

Anzman-Frasca, S., Liu, S., Gates, K. M., Paul, I. M., Rovine, M. J., & Birch, L. L. (2013). Infants' transitions out of a fussing/crying state are modifiable and are related to weight status. *Infancy*, *18*, 662–686.

Barch, D. M., Burgess, G. C., Harms, M. P., Petersen, S. E., Schlaggar, B. L., Corbetta, M.,… Washington university in St. Louis–university of Minnesota Human Connectome Project Consortium. (2013). Function in the human connectome: Task-fMRI and individual differences in behavior. *NeuroImage*, *80*, 169–189.

Beltz, A. M., Gates, K. M., Engels, A. S., Molenaar, P. C., Pulido, C., Turrisi, R.,… Wilson, S. J. (2013). Changes in alcohol-related brain networks across the first year of college: A prospective pilot study using fMRI effective connectivity mapping. *Addictive Behaviors*, *38*, 2052–2059.

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, *107*, 238–246.

Bentler, P. M. (1995). EQS structural equations program manual [Computer software manual]. Encino, CA: Multivariate Software.

Bentler, P. M., & Bonnett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin*, *88*, 588–606.

Borkenau, P., & Ostendorf, F. (1998). The Big Five as states: How useful is the Five-Factor Model to describe intraindividual variations over time? *Journal of Research in Personality*, *32*, 202–221.

Brown, T. (2006). *Confirmatory factor analysis for applied research*. New York, NY: Guilford.

Chen, G., Glen, D. R., Saad, Z. S., Hamilton, J. P., Thompson, M. E., Gotlib, I. H., & Cox, R. W. (2011). Vector autoregression, structural equation modeling, and their synthesis in neuroimaging data analysis. *Computers in Biology and Medicine*, *41*, 1142–1115.

Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software, 48*(4), 1–18. Retrieved from http://www.jstatsoft.org/v48/i04/

Fair, D. A., Bathula, D., Nikolas, M. A., & Nigg, J. T. (2012). Distinct neuropsychological subgroups in typically developing youth inform heterogeneity in children with ADHD. *PNAS: Proceedings of the National Academy of the United States of America*, *109*, 6769–6774.

Gates, K. M., Lane, S. T., Varangis, E., Giovanello, K., & Guskiewicz, K. (2017). Unsupervised classification during time series model selection. *Multivariate Behavioral Research, 52*, 129–148.

Gates, K. M., & Molenaar, P. C. (2012). Group search algorithm recovers effective connectivity maps for individuals in homogeneous and heterogeneous samples. *NeuroImage*, *63*, 310–319.

Gates, K. M., Molenaar, P. C., Hillary, F. G., Ram, N., & Rovine, M. J. (2010). Automatic search for fMRI connectivity mapping: An alternative to Granger causality testing using formal equivalences among SEM path modeling, var, and unified SEM. *NeuroImage, 50*, 1118–1125.

Gates, K. M., Molenaar, P. C., Iyer, S. P., Nigg, J. T., & Fair, D. A. (2014). Organizing heterogeneous samples using community detection of gimme-derived resting state functional networks. *PLOS One*, *9*(3), 1–11.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, *37*, 424–438.

Hamaker, E., Dolan, C., & Molenaar, P. (2002). On the nature of SEM estimates of ARMA parameters. *Structural Equation Modeling*, *9*, 347–368.

Huettel, S. A., Song, A. W., & McCarthy, G. (2004). *Functional magnetic resonance imaging* (Vol. 1). Sunderland, MA:Sinauer Associates Sunderland.

Jöreskog, K., & Sörbom, D. (2006). *LISREL 8.8 for Windows* [Computer software manual]. Chicago, IL: Scientific Software International.

Kim, J., Zhu, W., Chang, L., Bentler, P., & Ernst, T. (2007). Unified structural equation modeling approach for the analysis of multisubject, multivariate functional MRI data. *Human Brain Mapping*, *28*, 85–93.

Lamiell, J. T. (1981). Toward an idiothetic psychology of personality. *American Psychologist*, *36*, 276–289.

Lane, S., Gates, K., & Molenaar, P. (2014). *gimme: Group iterative multiple model estimation* [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=gimme (R package version 0.1-7)

Lane, S. T., Gates, K., Pike, H., Beltz, A., & Wright, A. (2016). Uncovering general, shared, and unique temporal patterns in ambulatory assessment data.

MacCallum, R. C., Roznowski, M., & Necowitz, L. B. (1992). Model modifications in covariance structure analysis: The problem of capitalization on chance. *Psychological Bulletin*, *111*, 490–504.

Molenaar, P. C. (2004). A manifesto on psychology as idiographic science: Bringing the person back into scientific psychology, this time forever. *Measurement*, *2*, 201–218.

Molenaar, P. C. (2008). On the implications of the classical ergodic theorems: Analysis of developmental processes has to focus on intra-individual variation. *Developmental Psychobiology*, *50*, 60–69.

Mumford, J., & Ramsey, J. (2014). Bayesian networks for fMRI: A primer. *NeuroImage*, *86*, 573–582.

Múthen, L., & Múthen, B. (2012). *Mplus user's guide*, (7th ed). [Computer software manual]. Los Angeles, CA: Múthen & Múthen.

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191–218.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. Retrieved from http://www.jstatsoft.org/v48/i02/

Sliwinski, M. J., Smyth, J. M., Hofer, S. M., & Stawski, R. S. (2006). Intraindividual coupling of daily stress and cognition. *Psychological Aging*, *21*, 545–557.

Steiger, J. H. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, *25*, 173–180.

The MathWorks. (2010). *Matlab – The language of technical computing*, (Version 7.10.0) [Computer software manual]. Natick, MA: Author. Retrieved from http://www.mathworks.com/products/matlab

van den Heuvel, M. P., & Sporns, O. (2011). Rich-club organization of the human connectome. *The Journal of Neuroscience*, *31*, 15775–15786.

van den Heuvel, M. P., Stam, C. J., Kahn, R. S., & Pol, H. E. H. (2009). Efficiency of functional brain networks and intellectual performance. *The Journal of Neuroscience*, *29*, 7619–7624.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K.,. The Washington University in St. Louis–University of Minnesota Human Connectome Project Consortium. (2013). The Wu-Minn Human Connectome Project: An overview. *NeuroImage*, *80*, 62–79.