Using R Package RAMpath for Tracing SEM Path Diagrams and Conducting Complex Longitudinal Data Analysis

Zhiyong Zhang, Aki Hamagami, Kevin J. Grimm, and John J. McArdle

October 1, 2013

Abstract

In this paper, we introduce and demonstrate the application of a newly developed `R` package `RAMpath` for tracing path diagrams and conducting structural longitudinal data analysis. The package `RAMpath` was developed to preserve the essential features of the classic DOS-version RAMpath program (McArdle & Boker, 1990) and ease data analysis done through *Structural Equation Modeling* (SEMing). The applicability of `RAMpath` is demonstrated through a mediation model, a MIMIC model, several latent growth curve models, a univariate latent change score model, and a bivariate latent change score model. In addition to performing regular SEM analysis, `RAMpath` has unique features. First, it can generate path diagrams according to a given model. Second, it can display path tracing rules through path diagrams and decompose total effects into their respective direct and indirect effects as well as decompose variances and covariances into individual bridges. Furthermore, `RAMpath` can fit dynamic system models automatically based on latent change scores and generate vector field plots based upon results obtained from a bivariate dynamic system. `RAMpath` is provided as an open-source `R` package.

Using R Package RAMpath for Tracing SEM Path Diagram and Conducting Complex Longitudinal Data Analysis

Introduction to the R package `RAMpath`

History of RAMpath

The original RAMpath program was developed to "organize, display, analyze, and save linear structural equation models" (McArdle & Boker, 1990, p. I-2). A unique feature of the program is the automatic procedure to calculate and display the tracing rules of path models (e.g., Loehlin, 1998). The RAMpath program was based on the *Reticular Action Model* (RAM; e.g., McArdle, 1978, 2005; McArdle & McDonald, 1984; McArdle, 2005) and this notation has been used in popular SEM software such as SAS PROC CALIS and the R package `sem` (Fox, 2006).

RAM provides a simple but flexible way to specify a SEM using three matrices: the asymmetric square matrix **A**, the symmetric matrix **S**, and the filter matrix **F**. Assume there are p manifest variables and q latent variables. The **A** matrix is a $(p+q)\times(p+q)$ square matrix where a non-zero element a_{ij} represents a single-headed arrow starting from variable j *(j=1,…,p+q)* on the *j*th column of **A** and ending on variable *i (i=1,…,p+q)* in the *i*th row of **A**. The **S** matrix is a $(p+q)\times(p+q)$ square matrix where a non-zero element s_{ij} $(=s_{ji})$ represents a two-headed arrow between variable i and variable j on the *i*th row and *j*th column of **S**. Note that when $i \neq j$, s_{ij} represents the covariance between variable *i* and variable *j* and when $i = j$, s_{ii} represents the variance of variable i. The **F** matrix is a $p\times(p+q)$ rectangular matrix of zeros and ones. The element f_{ii} is equal to 1 if variable i is a manifest variable.

For a given path diagram, the corresponding model can be expressed using the **A**, **S**, and **F** matrices. With these three matrices, a path diagram can be reproduced. Also, with the three matrices, an expected covariance (or augmented moment) matrix based on a specified SEM is derived (see McArdle & McDonald, 1984; McArdle, 2005). The original RAMpath program uses only these three matrices to specify models, calculate model components, draw path diagrams, and organize results.

RAMpath as a New Development in R

The original RAMpath program (McArdle & Boker, 1990; see McArdle, 2005) was written in the C programming language for DOS-based machines. RAMpath is a newly developed R package that retains the important features of the original RAMpath program and extends the original program in a variety of ways. Like the original program, it is not our intention to develop our own SEM estimation routines. Rather, we utilize the R package lavaan (Rosseel, 2012) for model estimation, which can handle continuous and ordinal data, incomplete data, and multiple group analysis. RAMpath is not particular about the way it conducts parameter estimation, so, future releases will adopt and take advantage of other SEM packages such as sem (Fox, 2006) and OpenMx (Boker et al., 2011). The flexibility of applying existing code and packages is the major reason for us to develop RAMpath within R. Compared to the original RAMpath program, the current version of RAMpath adds the functionality to easily conduct longitudinal data analysis through growth curve models as well as univariate and bivariate latent change score models (e.g., McArdle, 2001; McArdle & Hamagami, 2001) in addition to regular SEM analysis.

To install `RAMpath` in `R`, use the following command: [1]

```
install.packages("RAMpath")
```

Note that because of its dependence on `lavaan`, `RAMpath` requires `R` version 2.14.0 or newer. To load the `RAMpath` package, use the following command within `R`:

```
library(RAMpath)
```

Additionally, in order to use the path diagram features of `RAMpath`, one has to install the free external program Graphviz, or use our online program discussed in the upcoming section on path diagram generation. Graphviz provides a method to render a path diagram script into a PDF figure.[2] It works on major operating systems such as Linux, Mac OS X, and Windows. Graphviz provides a graphical interface for easy installation.

In the following, we demonstrate the applicability and features of `RAMpath` using five examples. The first example is a common SEM, termed a mediation model, where we demonstrate the basic input and output of `RAMpath`. This example is also used to demonstrate a unique feature of `RAMpath`, visual displays of tracing rules of path diagrams. The second example is a multiple indicator multiple cause (MIMIC) model where we demonstrate how to integrate `RAMpath` and `lavaan` on a more complex model. In the third example, we show how to conduct basic growth curve analysis. In the fourth example, we demonstrate how to fit a univariate latent change score model and in the fifth example, we show how to estimate a bivariate latent change score model and construct an associated vector field plot to visualize the lead-lag association for the bivariate latent change score model.

[1] The quotation marks in R should be straight instead of curly. If R has trouble recognizing the commands used in this paper, please first try to change the quotation marks.

[2] Graphviz can be downloaded at `http://www.graphviz.org/Download.php`.

Example 1: A Mediation Model

We will first use a mediation model to demonstrate the basics of the RAMpath package. Mediation models are widely used in exploring the underlying mechanism of observed relationships by investigating the direct and indirect effects among variables (e.g., MacKinnon, 2008). The path diagram for a simple mediation model used in this example is given in Figure 1. Here, we test whether memory ability (measured by Hopkins verbal learning test, hvlt) is a mediator of the association between age and everyday problem solving (ept). The data come from the Advanced Cognitive Training for Independent and Vital Elderly (ACTIVE) study and are publicly available (e.g., Jobe et al., 2001, McArdle & Prindle, 2008).[3]

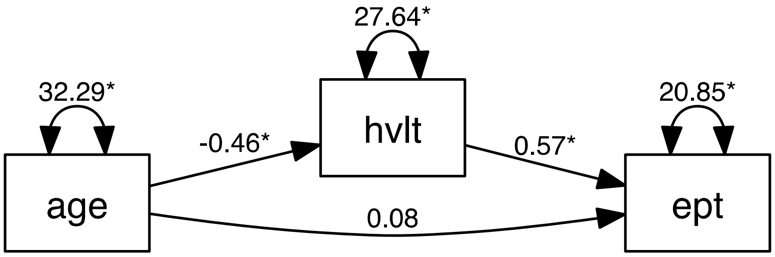

Figure 1. Path diagram for a mediation model. (Note: Generated by RAMpath).

RAMpath code to specify the mediation model in Figure 1 is shown in Code 1.

```
m1<-"
manifest=3
label=age,hvlt,ept
arrow(2,1)=?
arrow(3,1)=?
arrow(3,2)=?
sling(1,1)=?
sling(2,2)=?
sling(3,3)=?
"
```

Code 1. RAMpath specification for the mediation model in Figure 1.

[3] The URL to the data is http://dx.doi.org/10.3886/ICPSR04248.v3

First, the statements related to a model are quoted using either single or double straight quotation marks. The statement `manifest = 3` indicates there are three manifest variables in the model. If there were latent variables (e.g., 2 latent factors) one could specify them using the statement `latent = 2`. This is followed by `label=age,hvlt,ept`, which specifies the names of the observed variables (and latent variables if there are any) included in the model. Commas are used to separate the names of the variables. If latent variables were included in the model, their names would follow the manifest variable names. A path in SEM can be specified using either of two key words: `arrow` and `sling`. The `arrow` command is used to specify the single-headed paths (e.g., regression paths, factor loadings) or non-zero elements in the **A** matrix and the `sling` command is used to specify double-headed paths (e.g., variances and covariances) in the **S** matrix. For example, `arrow(2,1)` means that there is a single-headed path to the second variable, `hvlt`, from the first variable, `age`, and `sling(3,3)` means that there is double-headed arrow on `ept` to denote its variance. Note that the numbers in the parentheses correspond to the order of variables specified by `label`. Compared to the original RAMpath program, the **F** matrix is not required here because `RAMpath` automatically determines the manifest and latent variables based on the information specified by `manifest`, `latent`, and `label` statements.

A question mark, `?`, is used to indicate the specified path is free to be estimated. Starting values can be provided using a sequence of `?` and the starting value (see McDonald & Hartmann, 1992) such as `? 0.1` where `0.1` is the starting value for the estimated parameter at the beginning of the iterative estimation procedure. One could also provide a specific name to the path. For example, `arrow(2,1) = a ? 0.1` indicates that the path from `age` to `hvlt` is labeled as `a`, which is to be estimated and has a starting value `0.1`. To fix a parameter to a known value, the @ sign is used (as in the original RAMpath; see McArdle & Boker, 1990). For example, `sling(1,1) = @ 1` indicates that the variance of the first variable is fixed at 1.

Once a SEM model is specified through `RAMpath` commands, we can load an observed data set and conduct parameter estimation of the path model. The programming commands in Code 2 are used to read the data into R and estimate the mediation model in Figure 1.

```
## Either getting data from a physical file into R
ex1<-read.table(“ex1.txt”, header=T)

## or get the data embedded in the RAMpath package
data(ex1)

## Fit the model
res1<-ramFit(m1, ex1)
## lavaan model output
summary(res1$lavaan)
```

Code 2. Estimating the mediation model in Figure 1.

Specifically, the command

```
ex1<-read.table(“ex1.txt”, header=T)
```

reads the content of the data file `ex1.txt` that is physically on the hard drive into R. For the data file, the variable names are included on the top of the data file, which are read by R because of the `header=T` statement. The R name for the data is `ex1`. Each column of the data set represents a single variable, while each row represents a single subject. For longitudinal data, it should be in wide or full format where each column represents one occasion of measurement. Note that the data used here are also included in the `RAMpath` package and can be loaded into R using a command `data(ex1)`. The function `ramFit` automatically calls the `lavaan` package to estimate model parameters and outputs the RAM matrices. Specifically, using the following command,

```
res1 <- ramFit (m1, ex1)
```

we fit the mediation model, `m1,` using the sample data, `ex1`. Arguments for `lavaan` can also be provided in the `ramFit` function to control, for example, estimation methods and convergence criteria. `RAMpath` output from running the above code is given in Output 1. Each unknown element of the **A** and **S** matrices and its standard error are presented in matrix format.

```
--------------------
Parameter estimates:
--------------------

Matrix A

        age  hvlt ept
age       0     0   0
hvlt -0.463     0   0
ept   0.075 0.566   0

Matrix S

      age hvlt  ept
age  32.3    0    0
hvlt    0 27.6    0
ept     0    0 20.9

----------------------------------------
Standard errors for parameter estimates:
----------------------------------------

Matrix A

       age  hvlt ept
age      0     0   0
hvlt 0.138     0   0
ept  0.134 0.129   0

Matrix S

      age hvlt ept
age  6.81    0   0
hvlt    0 5.83   0
ept     0    0 4.4
```

Output 1. Output of running the mediation model in Figure 1.

All information regarding this data analysis including the RAM matrices and their standard errors is saved in the R object `res1` and can be accessed later. The names of object components saved in `res1` can be listed using the command `names(res1)`. For example, `res1$A` and `res1$Ase` provide the parameter estimates and their standard errors for the parameters in the **A** matrix. Additional information and output are also saved in `res1`. For example, `res1$lavaan` saves the original output of the `lavaan` function. Therefore, the code `summary(res1$lavaan)` will output the contents of the typical `lavaan` results as shown in Output 2.

```
lavaan (0.5-10) converged normally after 18 iterations

  Number of observations                            45

  Estimator                                         ML
  Minimum Function Chi-square                    0.000
  Degrees of freedom                                 0
  P-value                                        1.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  hvlt ~
    age              -0.463    0.138   -3.359    0.001
  ept ~
    age               0.075    0.134    0.560    0.575
    hvlt              0.566    0.129    4.369    0.000

Variances:
    age              32.294    6.808
    hvlt             27.641    5.827
    ept              20.853    4.396
```

Output 2. Sample output of `lavaan` for the mediation model.

`RAMpath` can generate a path diagram for a model automatically. For example, the command

```
plot(ramPathBridge(res1), "mediation")
```

generates the path diagram shown in Figure 1 and saves it into the PDF file called `mediation.pdf`.

The total effect between two variables, for example from `age` to `ept` can be decomposed into its respective direct and indirect effects. One approach to examining and understanding direct and indirect effects is through path tracing rules based on a path diagram (e.g., Wright, 1934). Furthermore, one can also trace the variance of a variable to see what elements or factors contribute to its variability. Additionally, for any two variables, their covariance can also be decomposed into individual components that contribute to the association between the two variables.

`RAMpath` provides a flexible way to decompose each effect (single direction relationship between variables such as the direct or total effect) and covariance (non-directional reciprocal relationship between two variables such as variance or covariance) into individual components. The code for doing the decomposition of effects and covariances from the mediation model is provided in Code 3. In the code, the `ramPathBridge` function with `indirect=TRUE` calculates all the individual paths using methods in Boker et al. (2002). The code `summary(bridge)` enumerates the decomposition of the effects into the table shown in Output 3 and the code `summary(bridge, type="bridge")` displays the decomposition of the covariances into the table shown in Output 4.

```
bridge<-ramPathBridge(res1, allbridge=TRUE, indirect=TRUE)
summary(bridge, se=TRUE)
summary(bridge, type="bridge")
```

Code 3. Decomposing the effects and covariances for the mediation model.

Output 3 lists all possible effects from one variable to another. For each effect, its components, including the direct and indirect effects, are identified and estimated. For example, for the total effect of `age` to `ept` there is the direct effect and one indirect effect passing through `hvlt`. In the output, the arrow (greater-than sign) `>` represents the direction of the paths. For example, `age > hvlt > ept` indicates the indirect effect, the multiplicative quantity of the path from `age` to `hvlt` and the path from `hvlt` to `ept`. Also, the first entry of `age > ept` refers to the total effect from `age` to `ept`, while the second entry of `age > ept` refers to the direct effect from `age` to `ept`. The total effect is equal to the sum of the direct effect and the indirect effect. The "`value`" column gives the estimated effects, the "`se`" provides the standard error estimates for the direct and indirect effects (obtained by setting `se=TRUE` in the `summary` function), and the "`percent`" column gives the percentage of each direct and indirect effect over the total effect. Note that the percentage can be larger than 100 or negative for a specific direct and indirect effect when negative effects involve.

```
Path and its decomposions:

Path Name                            Value           se    Percent
age > hvlt                  1       -0.463                 100.00
  age > hvlt                  1       -0.463        0.138    100.00
age > ept                   2       -0.187                 100.00
  age > ept                   1        0.075        0.134    -40.13
  age > hvlt > ept            2       -0.262        0.003    140.13
hvlt > ept                  1        0.566                 100.00
  hvlt > ept                  1        0.566        0.129    100.00
```

Output 3. List of all effects and their components.

Output 4 lists all possible covariances and variances (obtained by setting `allbridge=TRUE` in the `ramPathBridge` function) among the observed variables. For each covariance, its components are calculated. For example, the covariance (variance) between `ept` and `ept` is formed by six different components. In each component, the variables passed and the directions of paths are identified. Note that `<>` represents a double-headed path and `<`

and > represent single-headed paths in the respective directions. The decomposition of the variance and covariance is directly related to the tracing rules. For example, the total covariance between `ept` and `hvlt` is `15.64` and there are three components making up this covariance. The first component starts with `ept`, moves backward through the path `ept < hvlt`, and then goes through the variance component of `hvlt`. The coefficients passed are multiplied together so that `0.566*27.64=15.64`. A more complex component can be traced by going backwards from `ept` to `hvlt`, to `age`, passing through the variance of `age,` and finally going forward to `hvlt`.

```
Covariance and its bridges:

Path Name                                             Value   Percent
age <> age                                   1       32.294    100.00
   age <> age                                  1       32.294    100.00
hvlt <> age                                  1      -14.959    100.00
   hvlt < age <> age                           1      -14.959    100.00
hvlt <> hvlt                                 2       34.570    100.00
   hvlt <> hvlt                                1       27.641     79.96
   hvlt < age <> age> hvlt                     2        6.929     20.04
ept <> age                                   2       -6.039    100.00
   ept < age <> age                            1        2.423    -40.13
   ept< hvlt  < age <> age                     2       -8.462    140.13
ept <> hvlt                                  3       18.433    100.00
   ept < hvlt <> hvlt                          1       15.635     84.83
   hvlt < age <> age> ept                      2       -1.123     -6.09
   hvlt < age <> age> hvlt > ept               3        3.920     21.26
ept <> ept                                   6       30.827    100.00
   ept <> ept                                  1       20.853     67.65
   ept < age <> age> ept                       2        0.182      0.59
   ept< hvlt  < age <> age> ept                3       -0.635     -2.06
   ept < age <> age> hvlt > ept                4       -0.635     -2.06
   ept< hvlt  < age <> age> hvlt > ept         5        2.217      7.19
   ept < hvlt <> hvlt> ept                     6        8.845     28.69
```

Output 4. List of all variances and covariances and their components.

Decomposing the effects and covariances is very useful for teaching and learning SEM tracing rules. `RAMpath` can also graphically displays each regression effect and covariance path using path diagrams. Graphical outputs of variable relationships can be obtained by issuing the command, `plot`. The `plot` command takes five parameters: (1) an output of `ramPathBridge,` (2) a prefix label for multiple PDF files, (3) an originating variable, (4) a destination variable, and (5) an option of either a single directional relationship or a bi-directional relationship between two given variables. For example

```
plot(bridge, 'ex1effect', 'age', 'ept')
```

generates output plots for the direct and indirect paths from `age` to `ept`.[4] Using `bridge`, saved results by a `RAMpath` function, `ramPathBridge`, each path diagram is saved into a pdf file using a prefix, `ex1effect`, and sequentially named `ex1effect*.pdf` where * representing a number to identify the path. For the current example, two path diagrams are generated as shown in Figures 2 and 3. Note that in the path diagram, the solid lines represent the specific path evaluated and the dashed lines show paths or variables not directly related to the specific effect. The figure also shows the calculated effect.

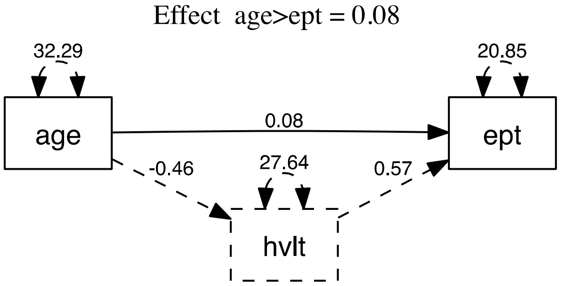

Figure 2. `ex1effect1.pdf`: the direct path from `age` to `ept`.

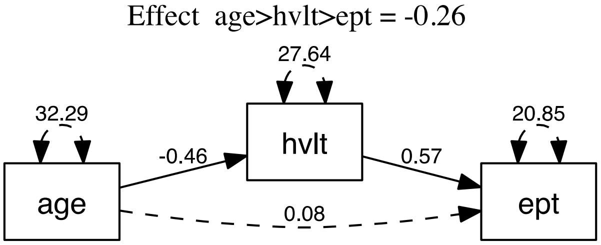

Figure 3. `ex1effect2.pdf`: the indirect path from `age` to `ept`.

[4] Note that because of rounding error, the coefficients on the path diagram are slightly different from the text output. One can change the digits on the plot by adding the option `digits=` in the plot the function.

Similarly, the covariance paths can also be plotted by specifying the plot type as `bridge`. For example, the R code

```
plot(bridge, 'ex1bridge', 'ept', 'hvlt', type='bridge')
```

generates three path diagrams shown in Figures 4-6, each representing a covariance component between `ept` and `hvlt`.

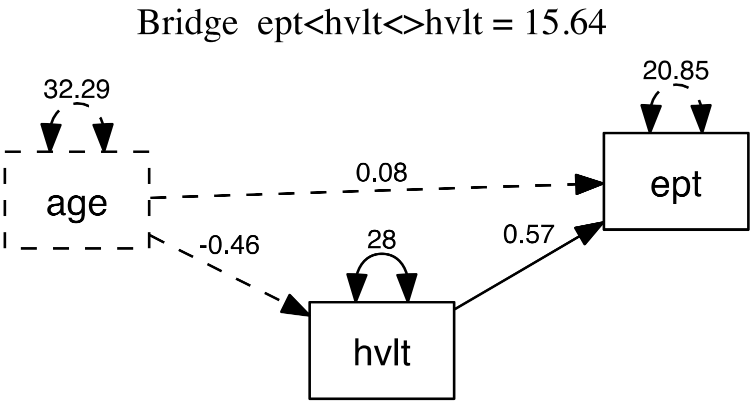

Figure 4. `ex1bridge1.pdf`: one component to form the covariance between `ept` and `hvlt`.

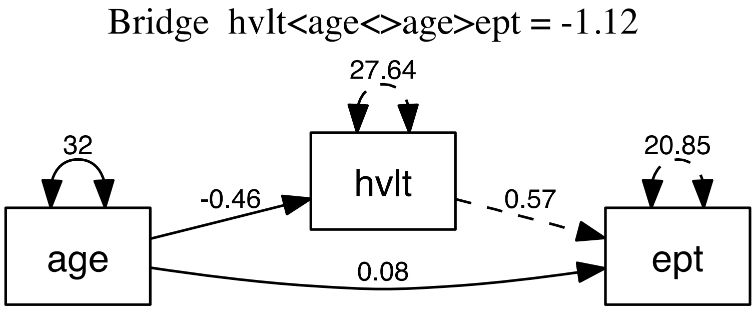

Figure 5. `ex1bridge2.pdf`: one component to form the covariance between `ept` and `hvlt`.

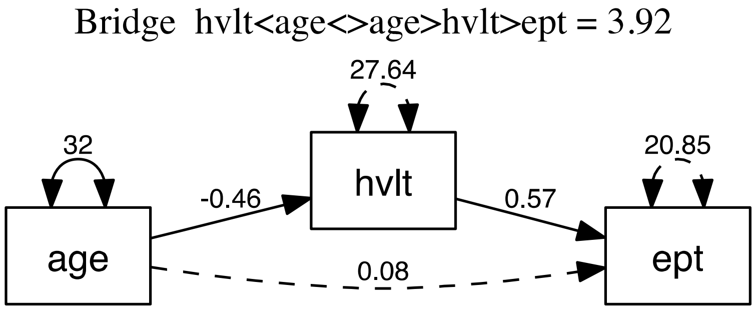

Figure 6. `ex1bridge3.pdf`: one component to form the covariance between `ept` and `hvlt`.

Example 2: A Multiple Indicator Multiple Cause Model

`RAMpath` can convert a `lavaan` object to RAM matrices for further analysis. Therefore, `RAMpath` can take full advantage of the estimation methods available through `lavaan`. In this example, we demonstrate the feature with a MIMIC model. The syntax in Code 4 estimates a MIMIC model in which the latent variable reasoning ability, `R`, is measured by three indicators – word sets (`ws1`), letter series (`ls1`), and letter sets (`lt1`) and this latent variable is predicted by education level (`edu`) and gender (`gender`). In `lavaan` notation, “~” represents the regression relationship and “=~” indicates a factor loading. Data for this example come from the ACTIVE study. The output for the MIMIC model is saved in `mimic.res`.

```
data(ex2)
mimic<-'
R =~ ws1 + ls1 + lt1
R ~ edu + gender
'
mimic.res<-sem(mimic, data=ex2, fixed.x=F)
```

Code 4. A MIMIC model fitted by `lavaan`

Using the function `lavaan2ram`, we can re-format the `lavaan` output in terms of RAM matrices. For example,

```
mimic.ram<-lavaan2ram(mimic.res)
```

gives the output shown in Output 5.

```
--------------------
Parameter estimates:
--------------------

Matrix A

       ws1 ls1 lt1  edu gender    R
ws1      0   0   0    0      0 1.00
ls1      0   0   0    0      0 1.19
lt1      0   0   0    0      0 0.43
edu      0   0   0    0      0    0
gender   0   0   0    0      0    0
R        0   0   0 0.78  0.056    0

Matrix S

       ws1 ls1 lt1   edu gender  R
ws1    5.4   0   0     0      0  0
ls1      0 4.7   0     0      0  0
lt1      0   0 3.9     0      0  0
edu      0   0   0  6.75  -0.26  0
gender   0   0   0 -0.26   0.18  0
R        0   0   0     0      0 15

----------------------------------------
Standard errors for parameter estimates:
----------------------------------------

Matrix A

       ws1 ls1 lt1   edu gender      R
ws1      0   0   0     0      0      0
ls1      0   0   0     0      0 0.033
lt1      0   0   0     0      0 0.016
edu      0   0   0     0      0      0
gender   0   0   0     0      0      0
R        0   0   0 0.051    0.3      0

Matrix S

        ws1  ls1  lt1   edu gender    R
ws1    0.44    0    0     0      0    0
ls1       0 0.57    0     0      0    0
lt1       0    0 0.18     0      0    0
edu       0    0    0 0.286 0.0337    0
gender    0    0    0 0.034 0.0075    0
R         0    0    0     0      0 0.87
```

Output 5. The RAM representation of the MIMIC model parameter estimates and standard errors.

The R object `mimic.ram` can be used to decompose effects and covariances and generate path diagrams. For example, the following code

```
bridge2<-ramPathBridge(mimic.ram)

plot(bridge2, 'mimic')
```

generates the path diagram shown in Figure 7. In the path diagram, "*" is used to indicate a parameter is significant at the 0.05 level. Therefore, in this example, education level is a significant predictor of reasoning ability while gender is not.

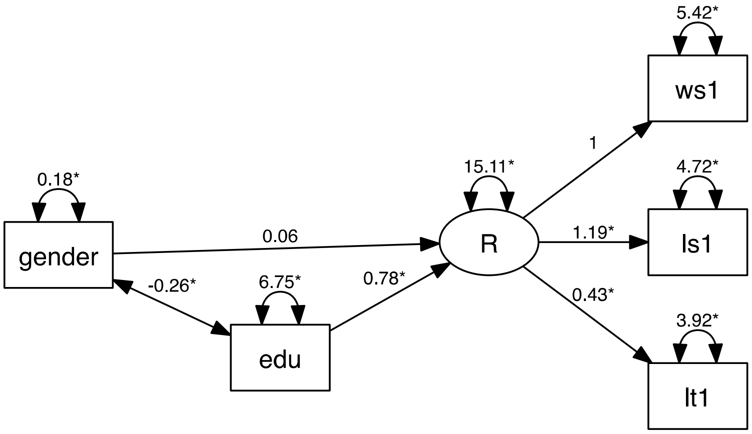

Figure 7. The path diagram for the MIMIC model.

Example 3: Latent curve models

Latent growth curve or latent curve models (LCM) are widely used in longitudinal data analysis (Rao, 1958; McArdle & Epstein, 1987; McArdle & Hamagami, 1991, 1992; Meredith & Tisak, 1990). The function `ramLCM` in `RAMpath` can be used to estimate the *no growth* model, the *linear* growth model, the *latent basis* growth model, and the *quadratic* growth model separately or simultaneously.

The no growth model is a model that assumes there is no change over time in the studied phenomenon. Using equations, it can be written as

$$\begin{aligned} y_{it} &= b_{i0} + e_{it} \\ b_{i0} &= \beta_{00} + \beta_{10}x_{i1} + \ldots + \beta_{q0}x_{iq} + v_{i0} \end{aligned}$$

where y_{it} is a measurement for person *i* at time *t*, and b_{i0} is the individual average score (or intercept). The random latent scores b_{i0} can be further predicted by a set of *q* predictor $x_{i1},\ldots,x_{iq}$ with coefficients $\beta_{10} - \beta_{q0}$.

The linear and latent growth curve model can be written as

$$\begin{aligned} y_{it} &= b_{i0} + A_t b_{i1} + e_{it} \\ b_{i0} &= \beta_{00} + \beta_{10}x_{i1} + \ldots + \beta_{q0}x_{iq} + v_{i0} \\ b_{i1} &= \beta_{01} + \beta_{11}x_{i1} + \ldots + \beta_{q1}x_{iq} + v_{i1} \end{aligned}$$

where A_t can be provided or estimated. If $A_t = t$, a linear growth curve model is defined. If two elements of *A* are fixed to different values (e.g., $A_1 = 0$ and $A_2 = 1$) and the remaining elements of *A* are estimated, a latent basis growth curve model is defined. In the `RAMpath` package, the elements of *A* can be flexibly specified as equally-spaced such as 1,2,3,4 or unequally-spaced such as 1, 3, 4, 7.

The quadratic growth curve model used in `RAMpath` is

$$\begin{aligned} y_{it} &= b_{i0} + A_t b_{i1} + A_t^2 b_{i2} + e_{it} \\ b_{i0} &= \beta_{00} + \beta_{10}x_{i1} + \ldots + \beta_{q0}x_{iq} + v_{i0} \\ b_{i1} &= \beta_{01} + \beta_{11}x_{i1} + \ldots + \beta_{q1}x_{iq} + v_{i1} \\ b_{i2} &= \beta_{02} + \beta_{12}x_{i1} + \ldots + \beta_{q2}x_{iq} + v_{i2} \end{aligned}$$

where b_{i2} is the quadratic slope and often time $A_t = t$.

RAMpath can be used to fit this series of growth curve models. For example, R code to fit all the four above-mentioned growth curve models and output results as RAM matrices simultaneously is

```
data(ex3)
gcm.all<-ramLCM(ex3, 1:6, ram.out=TRUE)
```

In addition, the fit statistics and fit indices for the four models are organized into the table shown in Output 6. Using the example data, ex3, the models can be compared immediately to arrive at the conclusion that the latent basis growth curve model fits the data best among the four competing models. Note that other, even better or more plausible, models can be constructed for a given set of data in addition to the four default models.

```
====================================================
Fit Statistics and Fit Indices for Model Comparisons
====================================================

                          No    Linear    Latent Quadratic
chisq               8.85e+03   978.024  9.52e+00  2.24e+01
df                  2.40e+01    21.000  1.70e+01  1.70e+01
chisq.per.df        3.69e+02    46.573  5.60e-01  1.32e+00
pvalue              0.00e+00     0.000  9.22e-01  1.71e-01
baseline.chisq      2.28e+03  2281.743  2.28e+03  2.28e+03
baseline.df         1.50e+01    15.000  1.50e+01  1.50e+01
baseline.pvalue     0.00e+00     0.000  0.00e+00  0.00e+00
cfi                 0.00e+00     0.578  1.00e+00  9.98e-01
tli                -1.43e+00     0.698  1.00e+00  9.98e-01
logl               -1.29e+04 -8924.848 -8.44e+03 -8.45e+03
unrestricted.logl -8.44e+03 -8435.836 -8.44e+03 -8.44e+03
npar                3.00e+00     6.000  1.00e+01  1.00e+01
aic                 2.57e+04 17861.696  1.69e+04  1.69e+04
bic                 2.57e+04 17886.984  1.69e+04  1.70e+04
ntotal              5.00e+02   500.000  5.00e+02  5.00e+02
bic2                2.57e+04 17867.940  1.69e+04  1.69e+04
rmsea               8.57e-01     0.302  0.00e+00  2.51e-02
rmsea.ci.lower      8.43e-01     0.286  0.00e+00  0.00e+00
rmsea.ci.upper      8.73e-01     0.318  1.38e-02  5.06e-02
rmsea.pvalue        0.00e+00     0.000  1.00e+00  9.45e-01
srmr                6.42e+00     0.203  1.89e-02  2.51e-02
srmr_nomean         7.08e+00     0.128  2.14e-02  2.23e-02
```

Output 6. The fit statistics for the growth curve models estimated by the ramLCM function.

Parameter estimates for the latent basis growth curve model are summarized in the path diagram in Figure 8. The R code to generate the path diagram is

```
bridge3<-ramPathBridge(gcm.all$ram$linear)
plot(bridge3, 'lgcm')
```

From the results, we can see there are individual differences in both initial level and change and there is a positive covariance between initial level and change.

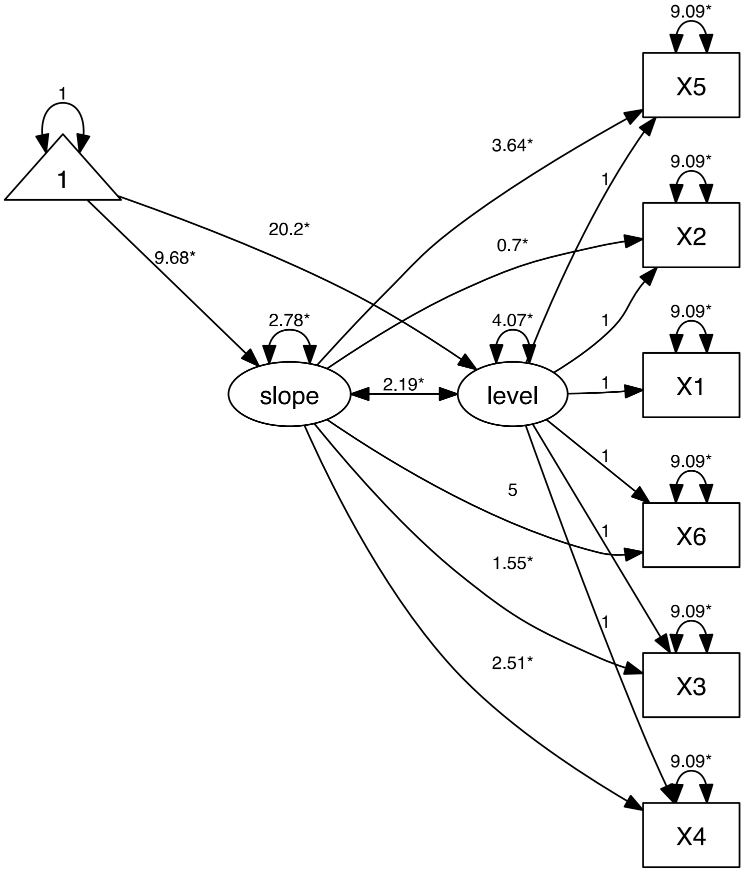

Figure 8. The path diagram with parameter estimates for the latent basis growth curve model.

The full usage of the `ramLCM` function can be seen using the command `?ramLCM` within R. For example, the following code fits a linear growth curve model with two predictors

```
gcm.pred<-ramLCM(ex3, c(2,3,4,5), model='linear',
          basis=c(2,3,4,5), predictor=c(1,6), ram.out=TRUE)
```

In this example, the data from time 2, 3, 4 and 5 are used to fit a linear growth curve model and the data at time 1 and 6 are used as predictors. This can be used to investigate how the data at the first and last occasions influence the change in the middle. Whether the model is practically justifiable is not considered here.

The independent and mutually uncorrelated error structure for the residuals is assumed for all the growth curve models discussed above by default. However, a user can fit other models based on the default ones using the function `ramReFit`. For example, if one would like to estimate the covariance between the residuals of `X1` and `X2` for the model with two predictors above, the following code can be used

```
gcm.pred.new<-ramReFit(gcm.pred, add='X1~~X2')
```

where the argument `add='X1~~X2'` adds new paths to the existing model.

Example 4: Univariate latent change score model

McArdle and Hamagami (2001) and McArdle (2001) introduced structural *latent change score* (LCS) models that incorporated methods of difference equations (Arminger, 1986; Coleman, 1968; Goldberg, 1986; Sheinerman, 1996; Tuma & Hanna, 1986) and latent curves (McArdle & Epstein, 1987; McArdle & Hamagami, 1991, 1992; Meredith & Tisak, 1990; Rao, 1958). The path diagram for a univariate latent change score model is presented in Figure 9.

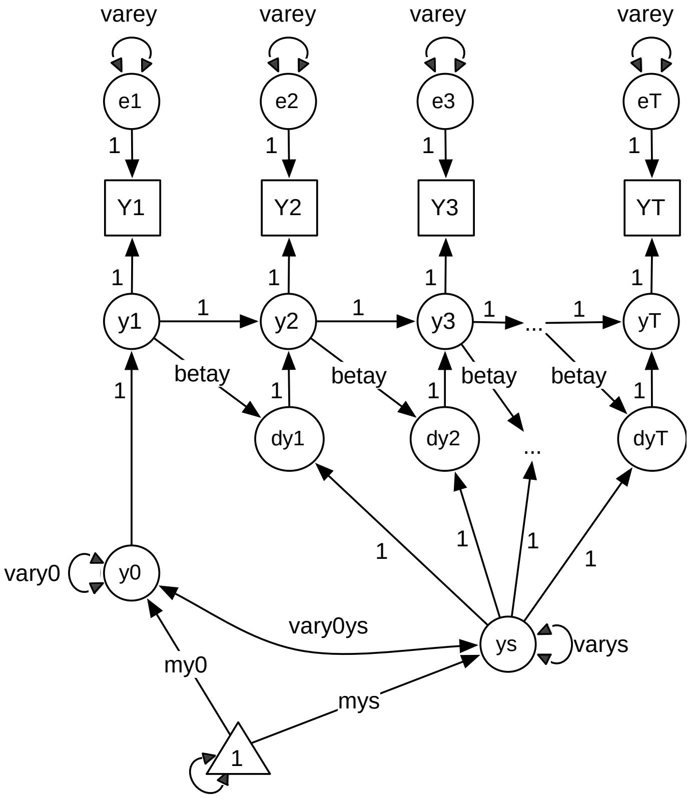

Figure 9. Path diagram for a univariate latent change score model.

In the univariate LCS model, we assume that a variable *Y* is repeatedly observed over time (t = 1 to T) on a sample of subjects (i = 1 to N). Also, we assume that a manifest score (Y_t) is the sum of a true score (y_t) and an unaccounted score (e_t):

$$Y_{it} = y_{it} + e_{it}$$

And we define Δy_{it} as the latent difference score between two successive occasions; i.e.,

$$\Delta y_{it} = y_{it} - y_{it-1} \, .$$

We apply a componential definition to a latent difference score at time *t* as

$$\Delta y_{it} = y_{is} + \beta y_{it-1} \, .$$

This representation simply means that a difference score between time *t* and time *t-1* (Δy_{it}) is determined as the sum of two terms: a self-feedback effect (βy_{it-1}), and a constant effect (y_{is}).

As a result, we express a system that has current scores as dependent variables and immediate past scores as predictors,

$$y_{it} = \Delta y_{it} + y_{it-1} = (1+\beta) y_{it-1} + y_{is} \;.$$

This true score dynamical model is then perturbed by an error term of the observed variable, i.e., uniqueness. Thus, observed scores are described as

$$Y_{it} = y_{it} + e_{it} = (1+\beta) y_{it-1} + y_{is} + e_{it}.$$

In order to account for sources of inter-individual differences, the level and linear slope components can be defined at a second level as

$$\begin{aligned} y_{0i} &= \mu_{y0} + v_{0i} \\ y_{si} &= \mu_{ys} + v_{1i} \end{aligned} \;,$$

where μ_{y0} and μ_{ys} are mean of the latent level and slope variables, while v_{0i} and v_{1i} are individual's deviation scores from means of latent level and slope scores. It is often assumed that these two latent variables are distributed as multivariate normal; i.e.,

$$\begin{pmatrix} y_s \\ y_0 \end{pmatrix} \sim MN\left[\begin{pmatrix} \mu_{ys} \\ \mu_{y0} \end{pmatrix}, \begin{pmatrix} \sigma_{ys}^2 & \sigma_{y0ys} \\ \sigma_{y0ys} & \sigma_{y0}^2 \end{pmatrix} \right].$$

Additional information about the technical details can be found in Hamagami, Zhang, & McArdle (2009).

The function `ramLCS` of the `RAMpath` package can be used to estimate the univariate latent change score model with very minimal input. For example, the R code below estimates a model for a dataset with 6 occasions of measurement. Output for the analysis is summarized in Table 1.

```
data(ex3)
test.lcs<-ramLCS(ex3, 7:12)
summary(test.lcs$lavaan, fit=T)
```

In the R code, `ex3` is the R data file that can be read into R from a file. In this example, the data can be loaded from our package directly using `data(ex3)`. The argument `7:12` tells R to use the 7th to the 12th variables in the data to fit a univarate latent change score model. Based on the model fit, the univaraite LCM fits the data well. The parameter betay is significant. Therefore, there is a significant self-feedback effect where one's change is positively related to one's previous score.

Table 1. Parameter estimates for the univariate latent change score model (χ^2_{20}=19.001, p-value=0.522; RMSEA=0.00; SRMR=0.035; CFI=1.000; TLI=1.001).

Parameter name	Estimate	Standard Error
betay	0.074	0.015
my0	19.988	0.146
mys	1.458	0.398
varey	8.762	0.306
vary0	2.603	0.559
varys	0.027	0.070
vary0ys	-0.083	0.164

The full usage of the `ramLCS` function is given below:

```
ramLCS(data, y, timey, ram.out = FALSE, betay, my0, mys,
       varey, vary0, varys, vary0ys, ...)
```

In this function, "`data`" specifies the data set to be used. "`y`" gives the indices for y variables. For example, 1:4 represents that the first four variables will be used. Non-continuous variable index can also be used here. For example `c(1,3,5,7)` would take out the four variables on the first, third, fifth and seventh columns of the data file to fit a model. "`timey`" specifies the time when y is observed. For example, `c(1,2,4,6)` indicates the variables/data are observed at time 1, 2, 4, and 6. The argument "`ram.out`" determines whether to print the RAM matrices in the output. For the LCS model, those matrices can be huge and therefore, the default is not to print the RAM matrices. The options betay, my0, mys, varey, vary0, varys, vary0ys specify the starting values for parameters in the univariate latent change score model. The names of the parameters correspond to those in the path diagram in Figure 8.

Example 5: Bivariate latent change score model

In bivariate dynamic systems, two change processes take place concurrently. In the bivariate dynamical models, longitudinal inter-relations between two change processes are investigated. A dynamic effect of one variable to the change score of another variable is coined a coupling effect. In algebraic form, a bivariate change score model (BLCS; e.g., Hamagami, Zhang, & McArdle, 2009) is expressed as

$$\begin{aligned}\Delta y_{it} &= y_{is} + \beta_y y_{it-1} + \gamma_y x_{it-1} \\ \Delta x_{it} &= x_{is} + \beta_x x_{it-1} + \gamma_x y_{it-1}\end{aligned},$$

where γ_y is a coupling effect from x to Δy, and γ_x is a coupling effect from y to Δx. When an error term is added to each process for the observed scores, we obtain simultaneous equations of the observed scores for two parallel processes, i.e.,

$$\begin{aligned}Y_{it} &= y_{is} + (1+\beta_y) y_{it-1} + \gamma_y x_{it-1} + ey_{it} \\ X_{it} &= x_{is} + (1+\beta_x) x_{it-1} + \gamma_x y_{it-1} + ex_{it}\end{aligned}.$$

Latent level and slope scores for y and x processes are assumed to evince inter-individual variability and are distributed multivariate normal, i.e.,

$$\begin{pmatrix} y_0 \\ y_s \\ x_0 \\ x_s \end{pmatrix} \sim MN\left[\begin{pmatrix} \mu_{y0} \\ \mu_{ys} \\ \mu_{x0} \\ \mu_{xs} \end{pmatrix}, \begin{pmatrix} \sigma^2_{y0} & \sigma_{y0ys} & \sigma_{y0x0} & \sigma_{y0xs} \\ \sigma_{y0ys} & \sigma^2_{ys} & \sigma_{x0ys} & \sigma_{xsys} \\ \sigma_{y0x0} & \sigma_{x0ys} & \sigma^2_{x0} & \sigma_{x0xs} \\ \sigma_{y0xs} & \sigma_{xsys} & \sigma_{x0xs} & \sigma^2_{xs} \end{pmatrix}\right].$$

The above model is called the bivariate latent change score (BLCS) model by McArdle (2001) and McArdle and Hamagami (2001). A path diagram for the BLCS model is presented in

Figure 10. Although the BLCS model is a very powerful tool in analyzing coupled dynamical systems (see Ferrer & McArdle, 2011), it is seen as generally difficult to specify and estimate in popular software. In fact, a dedicated program has been developed to ease the specification of univariate and bivariate latent change score models in Mplus (Zhang, 2006). The `ramBLCS` function simplifies the specification of a bivariate latent change score model so that it can be readily estimated by the R package `lavaan`.

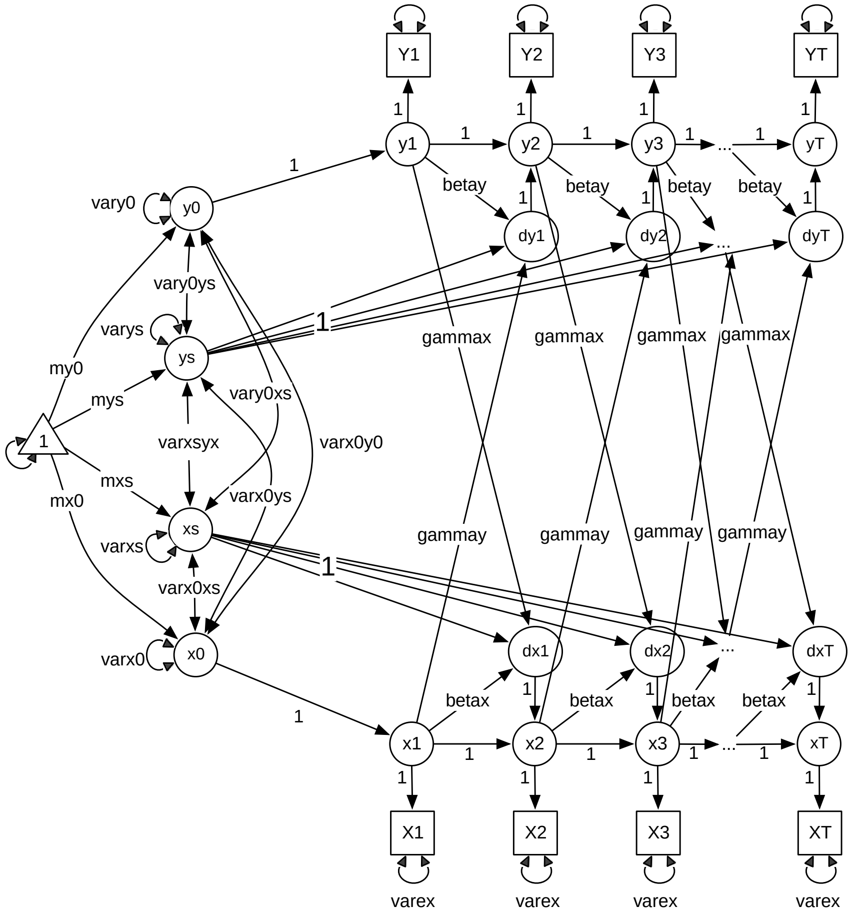

Figure 10. Path diagram for a bivariate latent change score model.

For example, the data set `ex3` includes the measured data for two variables from time 1 to time 6. The following code estimates a bivariate latent change score model,

```
test.blcs<-ramBLCS(ex3, 7:12, 1:6)
```

Note in the code, `7:12` indicates the 7th to the 12th columns in the data are from the first variable (*Y*) and `1:6` indicates the first 6 columns are from the second variable (*X*). To view the model fit and parameter estimates, one uses

```
summary(test.blcs$lavaan, fit=T)
```

For this example, the output is summarized in Table 2. The model apparently fits the data well. Since gammax is not significant while gammay is, one may conclude that *X* plays the leading role in the bivariate system.

Table 2. Parameter estimates for the bivariate latent change score model (χ^2_{70}=79.31, p-value=0.209; RMSEA=0.016; SRMR=0.033; CFI=.997; TLI=.997).

	Y			X	
Parameter name	Estimate	S.E.			S.E.
betay	0.079	0.101	betax	0.221	0.029
gammax	-0.003	0.037	gammay	-0.120	0.076
my0	19.975	0.146	mx0	20.213	0.144
mys	1.418	1.323	mxs	4.779	0.997
varey	8.779	0.325	varex	8.365	0.290
vary0	2.610	0.558	varx0	2.965	0.535
varys	0.027	0.073	varxs	0.626	0.136
vary0ys	-0.104	0.408	varx0xs	0.646	0.200
varx0y0	0.933	0.379			
vayx0ys	0.448	0.168			
vary0xs	0.512	0.363			
varxsys	0.499	0.081			

Note that the `lavaan` model specification for the bivariate latent change score model can be printed using command `ramShowModel(test.blcs)`. One can simply change the model specification to fit different alternative models to test different hypotheses as outlined by McArdle (2001).

The vector field plot has been used to graphically display the coupled dynamical systems represented by the BLCS model (Boker & McArdle, 1995). A vector field is an essential tool to decipher time-lag effects of complex longitudinal change processes. It is a representation of the simultaneous difference equations with a directional vector (arrow) in the location (X_0, Y_0) of each combination of initial values for *X* and *Y*. Each arrow then points in the expected direction of change (ΔX, ΔY) for any individual starting at the previous pair of values. RAMpath can readily generate the vector field plot. In addition to the typical vector field, the scatterplot of the selected actual data points and the 95% confidence ellipsoid are also added. Generally speaking, the arrows that should be taken most seriously are those inside the data ellipse. A practical application on the bivariate modeling of longitudinal changes in memory and changes in brain structures can be seen in McArdle et al. (2004).

For applications in the existing literature, one has to first estimate a BLCS model in one program such as Mplus and then plot the vector field using other software such as Matlab. In the RAMpath package, the function plot can be used to generate the vector field plot. For example, the R code below produces a vector field plot in Figure 10 for the above-fitted bivariate model.

```
plot(test.blcs, ylim=c(0,80), xlim=c(0,80), length=.05,
          xlab='X', ylab='Y',scale=.5, ninterval=9, n=50)
```

In the function, the first argument is the output from the ramBLCS function. The argument ylim gives the minimum and maximum values on the y-axis and xlim gives the minimum and maximum values on the x-axis of the vector field plot. The arguments xlab and ylab specify the labels for x-axis and y-axis, respectively. The argument ninterval specifies the number of columns and rows of arrows to plot and the argument length specifies the length of the arrow and the argument scale specifies the size of the arrow. Finally, the argument n specifies the number of points to plot in the scatterplot.

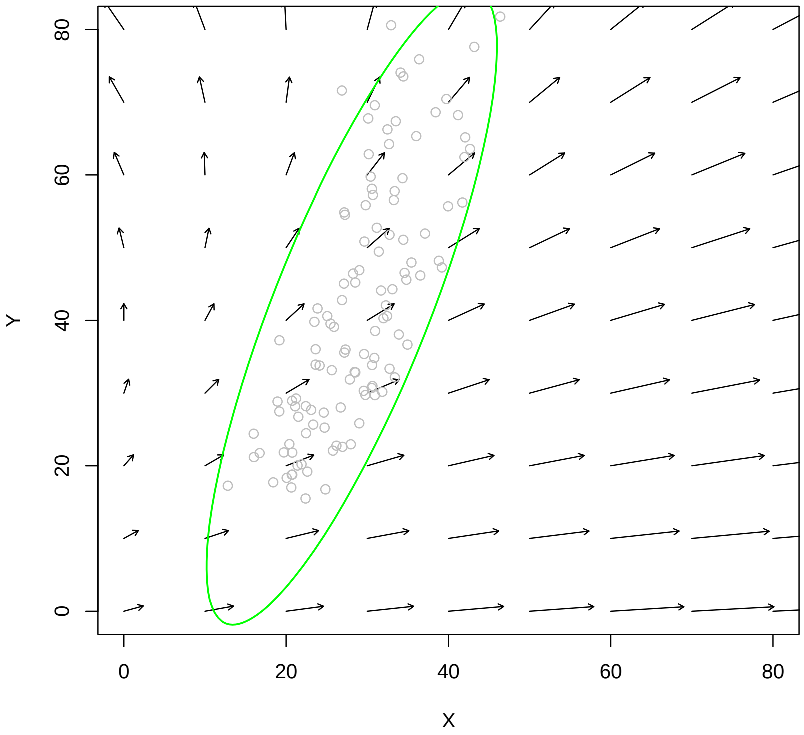

Figure 11. The vector field plot of the bivariate latent change score model.

Path Diagram Generation and Related Issues

Drawing path diagrams is not an easy task in general. Currently, `RAMpath` generates path diagrams using the free software Graphviz in two steps. First, the script language for Graphviz is generated within R and second, the script language is then processed by Graphviz outside of R to generate a PDF figure. The algorithms used by Graphviz automatically determine the layout of a path diagram. For many models such as the mediation model and the MIMIC model, the automatically generated path diagram is already laid out in a structural way. However, there are two main drawbacks. First, the generated path diagram cannot be edited and second, for very complex models such as the LCS and BLCS models, the path diagrams may not be laid out in a visually aesthetic style. In order to solve the drawbacks, a free online program has been developed which can be accessed at https://websem.psychstat.org/apps/drawsem/. The online program takes the output of `RAMpath` to generate an editable path diagram so that a user can easily modify it.

Taking the latent basis growth curve model in Example 3 as an example, although the path diagram itself is already sufficient to view the essential results, its layout is not structured well enough. To use the online program to edit it, one would first issue the following command in R,

```
plot(bridge)
```

Note that in the code, we did not specify a file name to save the figure. Therefore, the code will generate the output shown in Output 7. The output can be simply copied into the online program. Within the online program, one can then modify the path diagram to desired layout. Figure 12 illustrates the final path diagram after modification.

```
digraph "pathdiagram" {
  rankdir=LR;
  size="8,8";
  node [fontname="Helvetica" fontsize=14 shape=box];
  edge [fontname="Helvetica" fontsize=10];
  center=1;
  "level" [shape=ellipse]
  "slope" [shape=ellipse]
  "1" [shape=triangle]
  "1" -> "1" [label="1"   dir=both]
  "level" -> "X1" [label="1"];
  "level" -> "X2" [label="1"];
  "slope" -> "X2" [label="0.7*"];
  "level" -> "X3" [label="1"];
  "slope" -> "X3" [label="1.55*"];
  "level" -> "X4" [label="1"];
  "slope" -> "X4" [label="2.51*"];
  "level" -> "X5" [label="1"];
  "slope" -> "X5" [label="3.64*"];
  "level" -> "X6" [label="1"];
  "slope" -> "X6" [label="5"];
  "1" -> "level" [label="20.2*"];
  "1" -> "slope" [label="9.68*"];
  "X1" -> "X1" [label="9.09*"  dir=both];
  "X2" -> "X2" [label="9.09*"  dir=both];
  "X3" -> "X3" [label="9.09*"  dir=both];
  "X4" -> "X4" [label="9.09*"  dir=both];
  "X5" -> "X5" [label="9.09*"  dir=both];
  "X6" -> "X6" [label="9.09*"  dir=both];
  "level" -> "level" [label="4.07*"  dir=both];
  "slope" -> "level" [label="2.19*"  dir=both];
  "slope" -> "slope" [label="2.78*"  dir=both];
}
```

Output 7. Graphical script language for the latent basis growth curve model from R.

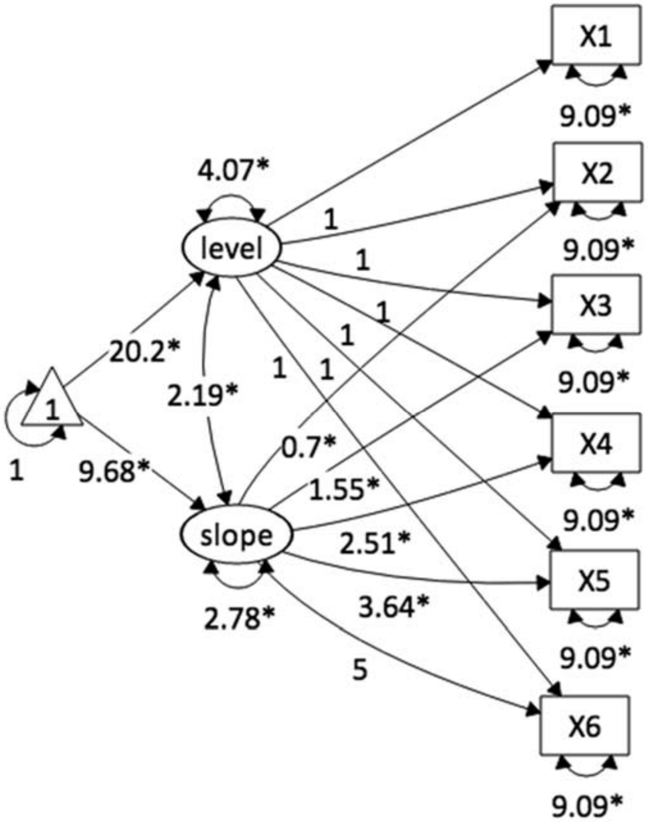

Figure 12. The path diagram for the latent basis growth curve model modified from the R output using our online program at https://websem.psychstat.org/apps/drawsem/.

Summary

In conclusion, `RAMpath` provides several unique programming feastures that are not currently available in either open source programs such as Mx, OpenMx, lavaan, and sem or commercial SEM software such as Mplus, AMOS, LISREL. The following table (Table 3) summarizes and highlights unique capability of RAMpath. In the table, '○' indicates availability of a specific feature, while '×' indicates non-availability of the feature.

Table 3. Summary of availability of general program features of RAMpath and other SEM software to perform SEM analyses.

Functionality	RAMpath	F1	F2	F3	C1	C2	C3
Perform Parameter estimation	○	○	○	○	○	○	○
Provide Model Fit	○	○	○	○	○	○	○
Handle Missing Data	○	○	○	○	○	○	○
Multiple Group Analysis	○	○	○	○	○	○	○
Summarize and decompose the total effect	○	×	×	×	○	×	×
Summarize and decompose variance and covariance	○	×	×	×	×	×	×
Generate Path Diagram	○	○	○	○	○	○	×
Edit Path Diagram	○	○	×	×	○	○	○
Graphical display for decomposed structural effects by path diagrams	○	×	×	×	×	×	×
Graphical display for decomposed variance and covariance among variables by path diagram	○	×	×	×	×	×	×
Streamlining of latent curve analyses	○	×	○	×	○	○	×
Streamlining of univariate latent score modeling	○	×	×	×	×	×	×
Streamlining of bivariate latent score modeling	○	×	×	×	×	×	×
Generate vector field (graphical representation of dynamical systems)	○	×	×	×	×	×	×
Free program	○	○	○	○	×	×	×
Open source (publicly modifiable)	○	○	○	○	×	×	×

Note. F1, F2, and F3 are three popular free SEM programs and C1, C2, and C3 are three popular commercial programs.

Discussion

The R package `RAMpath` was developed in order to preserve the essential features of the classical DOS-based RAMpath program and expedite data analysis through SEM. We have added built-in functions to make specification of longitudinal models simplified as well. The package `RAMpath` combines unique features of the RAM conceptual framework, functionality of the pre-existing RAMpath program, and flexibility of the R package `lavaan`. It also automates conventional longitudinal modeling and new longitudinal dynamic modeling. The package is provided as open-source software and can be extended to include more features by the R community. This paper highlights the main capabilities of `RAMpath`. The full functionality of `RAMpath` can be found in its built-in manual by issuing the command `?RAMpath` within R.

`RAMpath` retains the important features of the original RAMpath program. For example, it can specify an SEM model using RAM matrices. It can also generate a path diagram based on such specification. It also preserves the unique feature of RAMpath to decompose total effects into their direct and indirect effects and to decompose variances and covariances into individual bridges. Furthermore, it can display the decomposition through path diagrams.

`RAMpath` furthermore extends the capabilities of both the pre-existing RAMpath program and `lavaan` by adding automated procedures for latent curve modeling of longitudinal data analysis through SEM. For example, a user can simultaneously obtain parameter estimates, test statistics, and fit indices of several alternative latent curve models such as no growth, linear growth, latent basis growth, and quadratic growth model. If these are the dominant models of choice, a user can bypass a tedious process of separately writing a model specification of each latent growth model.

Moreover, `RAMpath` adds unique capability of dynamic system modeling. For example, it allows parameter estimation of univariate latent change score models and bivariate latent change score models for repeated measures. A user does not need to specify any details of latent change score models since RAMpath internally and conveniently specifies which parameters should be freed or fixed for users. In contrast, users have to specify detailed structural relationships in other SEM software packages. A detailed specification of SEMs can become a lengthy and time consuming process as the number of repeated measures increases.

In addition, RAMpath allows a user to generate a graphical representation of an SEM model and its components, and obtain PDF files thereof. It eliminates a time consuming process of drawing path diagrams using other graphic software. RAMpath also conveniently generates graphical representation of bivariate dynamic systems in the form of vector field plots. Again, a user does not need to use other scientific software or understand the mathematical details to generate these vector field plots. RAMpath bypasses these tedious processes for users.

Limitations and future directions

Many perspectives of the RAMpath package can and will be improved in subsequent releases. First, the current algorithm for decomposing the effect and covariance paths can be slow for large models. Therefore, more efficient algorithms can be developed. Second, in generating path diagrams, the software Graphviz is used to lay out the path diagram automatically and save it as a PDF file. But, currently, the automatically generated path diagram cannot be easily edited or modified. For this reason, we developed an online program to edit, modify, and improve aesthetics of path diagrams generated by RAMpath. In future releases, the RAMpath and the path diagram editing routine should be better integrated. Third, currently, the R package lavaan is used to estimate a model specified through RAMpath. One goal of the original RAMpath program is to utilize available SEM software and bridge different programs. Thus, other SEM software such as the sem package (Fox, 2006) and the OpenMx package

(Boker et al, 2011) can be incorporated into `RAMpath`. Fourth, the current `RAMpath` does not support more complex longitudinal data analysis such as growth mixture modeling. Future release will make this possible through the use of the `OpenMx` package.

References

Arminger, G. (1986). Linear stochastic differential equation models for panel data with unobserved variables. In N. Tuma (Ed.), *Sociological methodology 1986* (pp. 187-212). San Francisco: Jossey-Bass.

Boker, S. M., & McArdle, J. J. (1995). Statistical vector field analysis applied to mixed cross-sectional and longitudinal data. *Experimental Aging Research*, *21*, 77–93.

Boker, S. M., McArdle, J. J. & Neale, M. C. (2002) An algorithm for the hierarchical organization of path diagrams and calculation of components of covariance between variables. *Structural Equation Modeling, 9*(2), 174-194

Boker, S. M., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., Spies, J., Estabrook, R.,Kenny, S., Bates, T., Mehta, P., & Fox, J. (2011) OpenMx: An Open Source ExtendedStructural Equation Modeling Framework. *Psychometrika, 76* (2), 306-317.

Coleman, J. S. (1968). The mathematical study of change. In H. M. Blalock & A. B. Blalock (Eds.), *Methodology in social research* (pp. 428-475). New York: McGrow-Hill.

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling*, *13*, 465-486.

Goldberg, S. (1986). *Introduction to difference equation*. New York: Dover.

Hamagami, F., Zhang, Z., & McArdle, J. J. (2009). Modeling latent difference score models using Bayesian algorithms. In S.-M. Chow, E. Ferrer, & F. Hsieh (Eds), *Statistical methods for modeling human dynamics: An interdisciplinary dialogue* (pp. 319-348). New Jersey: Lawrence Erlbaum Associates.

Jobe, J. B., Smith, D. M., Ball, K., Tennstedt, S. L., Marsiske, M., Willis, S. L., . . . Kleinman, K. (2001). Active: A cognitive intervention trial to promote independence in older adults. *Controlled Clinical Trials, 22*(4), 453–479.

Loehlin, J. C. (1998). *Latent variable models: An introduction to factor, path, and structural analysis* (4th ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

MacKinnon, D.P. (2008). *Introduction to statistical mediation analysis*. London: Taylor & Francis.

McArdle, J. J. (1978). *A structural view of structural models.* Paper presented at the Winter Workshop on Latent Structure Models Applied to Developmental Data, University of Denver, December, 1978.

McArdle, J.J. (2001). A latent difference score approach to longitudinal dynamic structural analyses. In R. Cudeck, S. du Toit, & D. Sorbom (Eds.). *Structural Equation Modeling: Present and future*. Lincolnwood, IL: Scientific Software International (pp. 342-380).

McArdle, J. J. (2005). The development of the RAM Rules for Latent Variable Structural Equation Modeling. In A. Maydeu-Olivares & J. J. McArdle (Eds.), Contemporary Psychometrics (pp. 224-275). Mahwah, NJ: Lawrence Erlbaum Associates.

McArdle, J. J., & Boker, S. M. (1990). *RAMpath*. Hillsdale, NJ: Lawrence Erlbaum.

McArdle, J. J. & Epstein, D. (1987). Latent growth curves within developmental structural equation models. *Child development, 58*, 110-133.

McArdle, J. J. & Hamagami, F. (1991). Modeling incomplete longitudinal data using latent growth structural equation models. In L. Collins & J. L. Horn (Eds.), *Best methods for the analysis of change* (pp. 276-304). Washington, DC: American Psychological Association.

McArdle, J. J. & Hamagami, F. (1992). Modeling incomplete longitudinal data using latent growth structural models. *Experimental Aging Research, 18*, 145-166.

McArdle, J. J. & Hamagami, F. (2001). Latent difference score stuructural models for linear dynamic analyses with incomplete longitudinal data. In L. Collins & A. Sayer (Eds.). *New methods for the analysis of change. Decade of behavior*. (pp. 139-175). Washington, DC: APA Press.

McArdle, J. J., Hamagami, F., Jones, K., Jolesz, F., Kikinis, R., et al. (2004). Structural modeling of dynamic changes in memory and brain structure using longitudinal data from the normative aging study. *Journal of Gerontology: Psychological Sciences*, *59* (6), 294–304

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the Reticular Action Model for moment structures. *British Journal of Mathematical and Statistical Psychology, 37*, 234-251.

McArdle, J. J. & Prindle, J. J. (2008). A latent change score analysis of a randomized clinical trial in reasoning training. *Psychology and Aging, 23*(4), 702-719.

McDonald, R. P. & Hartmann, W. (1992). A procedure for obtaining initial values of parameters in the RAM model. *Multivariate Behavioral Research, 27*(1), 57-76.

Meredith, W., & Tisak, J. (1990). Latent curve analysis. *Psychometrika, 55*, 107–122.

Rao, C. R. (1958). Some statistical methods for the comparison of growth curves. *Biometrics, 14*, 1-17.

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software, 48*(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

Scheinerman, E. R. (1996). *Invitation to dynamical systems*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Tuma, N. & Hannan, M. (1984). *Social dynamics*. New York: Academic Press.

Wright, S. (1934). *Annals of Mathematical Statistics, 5*(3), 161-215

Zhang, Z. (2006). LDSM: A C++ program for generating codes for analyzing latent difference score model in Mplus. Retrievable from http://www.psychstat.org/us/article.php/38