

### Online Supplement S-3

#### Example of a CoTiMA using 2 empirical correlation matrices following the steps shown in the flowchart of Appendix B

```

1. # Step 0 ##### install.packages('ctsem') # only required if not already installed #####
2. require(ctsem) # load ctsem to perform continuous time structural equation modeling and CoTiMA
3. require(MASS) # load MASS package to create multivariate normal pseudo raw data from correlation matrices

4. noOfVars <- 2 # The number of (manifest) variables analyzed (satisfaction & performance; is used later)

5. model1 <- ctModel( # General ctsem model for all studies.
6.   n.latent=2, # 2 latent variables: satisfaction & performance.
7.   n.manifest=noOfVars, # 2 manifest variables: satisfaction & performance.
8.   Tpoints=2, # The largest number of waves among all primary studies.
9.   manifestNames=c("X", "Y"), # Use "X" and "Y" as names for satisfaction and performance.
10.  DRIFT=matrix(c("Auto_X", "Cross_XToY", # Define the drift matrix. Assigning names (e.g. "Auto_X")
11.                "Cross_YToX", "Auto_Y"), # makes the output easier to read and sets parameters free.
12.                2, 2), # It is a 2 x 2 drift matrix.
13.  LAMBDA=matrix(c( 1, 0, # Matrix of Loadings of the 2 manifest indicators on the 2 latent
14.                  0, 1), # factors. Using numbers fixes parameters to these values.
15.                  2, 2), # It is a 2 row (latent) x 2 column (manifest) matrix.
16.  MANIFESTVAR=matrix(c( 0, 0, # Matrix of error variances and error covariances of manifest
17.                        0, 0), # indicators. Since there is only 1 manifest indicator per
18.                        2, 2), # factor here, variances are fixed to 0. It is also a 2 x 2 matrix.
19.  MANIFESTMEANS = matrix(c( 0, 0), # Means of manifest indicators. Fixed to 0 because means are not
20.                           2, 1), # analyzed. It is a 2 x 1 matrix (a vector).
21.  TOMEANS = matrix(c( 0, 0), # Means of latent factors. Also fixed to 0 because means are not
22.                    2, 1)) # analyzed. Also a 2 x 1 matrix (a vector).

23. noOfStudies <- 2 # This is used below for extracting start values from preliminary results.
24. varNames <- c('X_T0', 'Y_T0', 'X_T1', 'Y_T1', 'dT1') # Specifies the variable names for the pseudo raw data created below.

25. # Step 1 ##### primary study 1 (Ashforth & Saks, 1996) & primary study 2 (Bechtold et al., 1981) #####
26. lag01 <- 6 # The lag was 6 months in the Ashforth & Saks study.
27. N01 <- 222 # Their sample size was 222.
28. waves01 <- 2 # They used 2 waves.
29. corr01 <- matrix(c( 1.00, 0.11, 0.64, 0.14, # The correlations reported by Ashforth & Saks.
30.                    0.11, 1.00, 0.20, 0.69,
31.                    0.64, 0.20, 1.00, 0.21,
32.                    0.14, 0.69, 0.21, 1.00),
33.                    4,4) # It is a 4 x 4 matrix (2 manifest variables at 2 waves).
34. lag02 <- 18 # The lag was 6 months in the Bechtold et al. study.
35. N02 <- 64 # Their sample size was 64.
36. waves02 <- 2 # They used 2 waves.
37. corr02 <- matrix(c( 1.00, 0.15, 0.53, 0.17, # The correlations reported by Bechtold et al.
38.                    0.15, 1.00, 0.19, 0.57,
39.                    0.53, 0.19, 1.00, 0.21,
40.                    0.17, 0.57, 0.21, 1.00),
41.                    4, 4) # It is a 4 x 4 matrix (2 manifest variables at 2 waves).

42. # Step 2 ##### create pseudo raw data #####
43. matrixOfLags01 <- matrix(lag01, N01, waves01-1) # Create 22 x 1 matrix filled with the time lag of Study 1.
44. pseudoRaw01 <- mvnorm( n=N01, # Create pseudo raw data for N = 22 cases.
45.   mu=matrix( 0, # Matrix (vector) of mean values: All are 0 in a
46.              1, # matrix with 1 row

```

```

47.          waves01*noOfVars),      # and 2 * 2 columns (4 variables).
48.          Sigma=corr01,          # The correlations to be reproduced by the pseudo raw data.
49.          empirical=TRUE)        #
50. raw01 <- cbind(pseudoRaw01, matrixOfLags01) # Combine pseudo raw data and time lag.
51. colnames(raw01) <- varNames     # Assign variable names.

52. matrixOfLags02 <- matrix(lag02, N02, waves02-1) # Create 64 x 1 matrix filled with the time lag of Study 2.
53. pseudoRaw02 <- mvnorm( n=N02,          # Create pseudo raw data for N = 64 cases.
54.          mu=matrix(0,                # Matrix (vector) of mean values: All are 0 in a
55.          1,                          # matrix with 1 row
56.          waves01*noOfVars),          # and 2 * 2 columns (4 variables).
57.          Sigma=corr01,                # The correlations to be reproduced by the pseudo raw data.
58.          empirical=TRUE)              #
59. raw02 <- cbind(pseudoRaw02, matrixOfLags02) # Combine pseudo raw data and time lag.
60. colnames(raw02) <- varNames          # Assign variable names.

61. # Step 3 ##### ctsem model for each primary study #####
62. fit01 <- ctRefineTo(datawide=raw01, model1) # Fit model1 to pseudo raw data of Study 1.
63. fit01CI <- ctCI(fit01, confidenceintervals = 'DRIFT') # Compute confidence intervals for drift matrices.
64. summary(fit01CI, verbose=TRUE)           # Display parameter estimates and fit indices for Study 1.
65. fit02 <- ctRefineTo(datawide=raw02, model1) # Fit model1 to pseudo raw data of Study 2.
66. fit02CI <- ctCI(fit02, confidenceintervals = 'DRIFT') # Compute confidence intervals for drift matrices.
67. summary(fit02CI, verbose=TRUE)          # Display parameter estimates and fit indices for Study 2.

68. startValues <- rbind( fit01$mxobj$output$estimate, # Extract parameter estimates (into a matrix)
69.          fit02$mxobj$output$estimate) # to be used as start values later.
70. parNames <- colnames(startValues)           # Extract parameter names.
71. noOfParams <- length(parNames)              # Determine the number of estimate parameters.
72. startValues <- c(t(startValues))            # Transform matrix of start values in column vector.
73. prefixG1 <- paste0("g", rep(c("1_", noOfParams))) # Prefixes for parameter names for group1 (= "1_").
74. prefixG2 <- paste0("g", rep(c("2_", noOfParams))) # Prefixes for parameter names for group2 (= "2_").
75. prefixALL <- c(prefixG1, prefixG2)         # Combine prefixes.
76. names(startValues) <- paste0(prefixALL, parNames) # Assign names to start values.

77. # Step 4 ##### Combine Data of all primary studies #####
78. dataAll <- rbind(raw01, raw02)             # Combine data: 86 (22 + 64) x 5 (4 variables + time lag) matrix.
79. colnames(dataAll) <- varNames             # Assign names.
80. grps <- paste0("g", rep(c(1,2), c(N01, N02))) # Create a vector with N01 = 22 "g1" and N02 = 64 "g2", which is used to identify
81. # data of primary studies in subsequent multi-group models.

82. # Step 5 ##### ctsem multigroup heterogeneity model #####
83. fitHet <- ctMultigroupFit( dat=dataAll,          # Fit the model. Use the matrix of combined data.
84.          groupings=grps,                # Use the grouping vector to identify primary studies.
85.          ctmodelobj=model1,              # Fit the data to the previously defined model1.
86.          omxStartValues=startValues)      # Use the previously extracted start values.
87. fitHetCI <- ctCI(fitHet, confidenceintervals = 'DRIFT') # Compute confidence intervals for drift matrices.
88. summary(fitHetCI, group="g1")           # Fit indices and coefficients for Study 1
89. summary(fitHetCI, group="g2")           # Fit indices and coefficients for Study 2

90. # Step 6 ##### fit full drift set homogeneity model #####
91. fixModel <- model1                        # Copy model1 into fixModel.
92. fixModel$DRIFT <- matrix(("groupfixed"), 2,2) # Make drift matrix homogenous across primary studies.
93. fitHomFull <- ctMultigroupFit( dat=dataAll,      # Use the matrix of combined data.
94.          groupings=grps,                  # Use the grouping vector to identify primary studies.
95.          ctmodelobj=model1,              # Fit the data to the previously defined model1
96.          fixedmodel=fixModel,            # using fixModel to specify free/constrained parameters.
97.          omxStartValues=startValues)      # Use the previously extracted start values.

```

```

98. fitHomFullCI <- ctCI( fitHomFull,
99.     confidenceintervals = 'DRIFT')           # Compute confidence intervals for drift matrices.
100. summary(fitHomFullCI, group="g1")         # Fit indices and coefficients for Study 1
101. summary(fitHomFullCI, group="g2")         # Fit indices and coefficients for Study 2

102. # Step 7 ##### plot cross effects from full drift set homogeneity model #####
103. x <- matrix(NA, 80,2)                     # Create empty object x to store discrete cross effect sizes
104.                                           # (for X->Y and Y->X) across 80 waves.
105. driftMatrix <- summary(fitHomFullCI, group="g1")$DRIFT # Copy drift coefficients into driftMatrix.
106. for(i in 1:80) {                          # Loop through effect size computation 80 times.
107.     x[i,] <- expm(driftMatrix * i)[c(2,3)] # Compute matrix exponential of drift matrix
108.                                           # and store the 2nd & 3rd (cross effects) into x.
109. }                                          # End loop.

110. matplot( x,                               # Plot discrete effect sizes across 80 waves.
111.     type="l",                               # Draw a line (not dots or other symbols).
112.     lty=1,                                  # Draw solid line (not dashed).
113.     xlab="Months",                          # Label x-axis.
114.     ylab="Cross Effect (XtoY=black, YtoX=grey)", # Label y-axis.
115.     ylim=c(0,.2),                          # Scale the y-axis from 0 to .20.
116.     col=c(1,8))                            # Use color 1 (black) for X->Y & color 8 (grey) for Y->X.

117. dev.copy(png, "Cross_XtoY.png",           # Save plot in png format with filename="Cross_XtoY.png".
118.     width=8, height=8, units='in',         # Size of the figure in inches.
119.     res=300)                               # Resolution of the figure in dpi.
120. dev.off()                                # Close the graphics device.

```

A CoTiMA could be performed in eight steps (Step 0 to Step 7), which correspond to the eight script sections above. (Step 1 to Step 7 correspond to the seven Steps in the flowchart shown in Appendix C). In Section 0 (#1-24), the ctsem package is installed and a basic ctsem model is set up as a template for later use. Here, two possibilities for psychometric meta-analysis exist. The model here uses only a single manifest indicator per latent variable. If more manifest indicators are available, the model could be changed so that error variance and true variance are separated. The second option would be to fix the error variances of the single manifest indicator to 1 – reliability.

In Section 1 (#25-41), information about primary study data are imported containing information on used time lag (lag), sample size (N), number of waves (waves) and the correlation matrix (corr). Here, another possibility for psychometric meta-analysis exists. One could also add reliabilities in this step and correct the correlations for attenuation.

In Section 2 (#42-60), these primary study data are then used to create a corresponding ‘pseudo’ raw data set and names are assigned to the variables in the raw data set; this step is needed because, so far, ctsem can only handle raw data.

In Section 3 (#61-76), CTSEM are fitted for each primary study. Note this step is not essential and it could be skipped. However, there are two potential benefits. First, the heterogeneity model, which is tested later and which fits all primary studies simultaneously, yields identical results but may take considerably more time if the number of primary studies is large. Second, lines #68-76 show how to extract the estimated parameters and store them in an object ‘startValues’. Indeed, they can be used in later models as starting values, which ensures faster convergence and improved replicability because random fluctuations in starting values are avoided.

In Section 4 (#77-81), the pseudo raw data of all primary studies are combined into a single data set. Further, A grouping variable is composed, which is used later to assign each line in the data set to the respective primary study.

In Section 5 (#82-89), continuous time parameters are estimated using a multi-sample heterogeneity model. As noted earlier, the resulting estimates are identical to the estimates obtained in Section 4. Further, the minus 2 log-likelihood value of this heterogeneity model is identical to the sum of the minus 2 loglikelihood values of the models fitted in Section 4.

In Section 6 (#90-101), the estimated continuous time parameters are aggregated. A homogeneity model is fitted, in which all four continuous time parameters (two auto effects and two cross effects) are constrained to be equal across all primary studies. When fitting the homogeneity model using maximum likelihood (ML) estimation, the invariant drift parameters are chosen such that the overall fit function is minimized, which is the sum of the fit functions of the multiple groups (e.g., Voelkle et al., 2012, Eq. 12):

$$F_{ML} = \log|\Sigma| + \text{tr}(\mathbf{S}\Sigma^{-1}) - \log|\mathbf{S}| - (V+1) \quad (9)$$

with  $V$  denoting the number of (manifest) variables,  $\mathbf{S}$  denoting the observed and  $\Sigma$  denoting the model implied correlation matrix.

Finally, Section 7 (#102-120) is used to plot the course of the discrete cross-lagged effects over time. The black line illustrates the course of the aggregated effect of satisfaction on performance over time, and the grey line illustrates the course of the aggregated reversed effect of performance on satisfaction.