This document includes five web appendices to the paper:

Nguyen TQ, Webb-Vargas Y, Koning IM, Stuart EA. Causal mediation analysis with a binary outcome and multiple continuous or ordinal mediators: Simulations and application to an alcohol intervention. *Structural Equation Modeling: A Multidisciplinary Journal.*

including:

Appendix A: 22 scenarios representing variation in path and mediator residual correlation signs

Appendix B: Results from variations in path strengths and correlation strengths

Appendix C: Additional simulations: Semi-positive-definite mediator residual correlation matrix

Appendix D: Mplus input template/samples and corresponding R-code

Appendix E: Mplus input for the illustrative example

The Mplus inputs and R-code in Appendices D and E are also available as several separate Mplus and R files (to come).

Appendix A: 22 scenarios representing variation in path and mediator residual correlation signs

While 7 paths (including exposure-to-mediator paths $\alpha_1, \alpha_2, \alpha_3$, mediator-to-outcome paths $\beta_1, \beta_2, \beta_3$ and exposure-to-outcome path γ) plus 3 mediator residual correlations ($\rho_{12} = \text{corr}(M^{[1]}, M^{[2]}), \rho_{13} = \text{corr}(M^{[1]}, M^{[3]}), \rho_{23} = \text{corr}(M^{[2]}, M^{[3]})$) imply 2^{10} permutation of signs, many of these can be converted to other permutations simply by either switching the locations of the mediators, or flipping the signs (or reversing the category order) of certain mediator(s) and/or the outcome. By the reasoning below, we reduce the number of scenarios investigated to 22.

First, we fix $\gamma > 0$; a negative γ can be turned positive by flipping the outcome.

Second, for mediator residual correlations, we consider situations where: (1) all three correlations are positive, and (2) when two are positive and one is negative, the latter including three situations: (i) $\rho_{12} < 0$, (ii) $\rho_{13} < 0$, and (iii) $\rho_{23} < 0$. The other situations (all negative, and one positive two negative) can be converted to these cases by flipping the sign of one of the mediators.

Third, there are $2^6 = 64$ permutations of the α paths and β paths. Half of these, however, can be converted to the other half by flipping the signs of all the mediators, resulting in all α and β paths flipping signs but the residual correlation matrix remaining unchanged. Also, mediators with same-sign α paths and same-sign β paths are exchangeable, so swapping their locations results in the 32 permutations collapsing into 10 combinations.

Combining the 10 path sign combinations with the all-positive mediator residual correlation matrix results in 10 scenarios to be investigated. Combining them with the three one-negative-residual correlation options gives rise to 30 possible scenarios, but some of these are equivalent to others by swapping locations and/or flipping signs of some mediator(s); they are thus reduced to 12 scenarios.

Appendix B: Results from variations in path strengths and correlation strengths

The method performs well, with all estimators, across the variations examined in magnitudes of path coefficients and mediator residual correlations, with bias, MSE and coverage comparable to those in the base scenarios. The tables below present results from WLSMV estimator with ordinal mediators. Results from other estimator and mediator type combinations are similar. See Tables 1 and 2.

Appendix C: Additional simulations: Semi-positive-definite mediator residual correlation matrix

Using sign scenario 1a as base, we replace the mediator residual correlation matrix with each of several semi-positive-definite matrices to compare how the estimators perform.

In all cases, the ML estimator fails; Bayes estimator completes estimation for all simulations; and WLSMV completes estimation for a large majority, but fails with a significant minority, of simulations. As an example, Table 3 presents results from WLSMV and Bayes estimators for one of the scenarios; results from the other scenarios show similar patterns.

With respect to RD-based effects, both estimators perform well in estimating TE, with bias, RMSE and estimated SE being similar and small. For NDEs and NIEs, Bayes estimator

Table 1

Bias, root mean square error (RMSE) and coverage (by 95% confidence intervals), over path strength variations (using WLSMV estimator with ordinal mediators)

	For RD-based effects									**For RR-based effects**								
	Bias			RMSE			Coverage			Standardized bias			Standardized RMSE			Coverage		
Effect	base	symdiff	asymdiff	base	symdiff	asymdiff	base	symdiff	asymdiff	base	symdiff	asymdiff	base	symdiff	asymdiff	base	symdiff	asymdiff
Using 4a as base scenario																		
TE	0.000	-0.000	0.000	0.042	0.040	0.045	0.944	0.941	0.947	0.003	0.003	0.004	0.104	0.109	0.095	0.947	0.945	0.948
$NDE(\cdot 0)$	-0.001	0.000	-0.001	0.033	0.026	0.037	0.943	0.955	0.946	0.001	0.001	0.001	0.045	0.037	0.056	0.951	0.953	0.948
$NIE(1\cdot)$	0.001	-0.000	0.001	0.035	0.030	0.039	0.945	0.947	0.947	0.003	0.001	0.003	0.088	0.094	0.076	0.946	0.950	0.951
$NIE(0\cdot)$	0.001	-0.001	0.001	0.035	0.030	0.037	0.942	0.951	0.951	0.004	-0.001	0.005	0.159	0.138	0.141	0.946	0.943	0.937
$NDE(\cdot 1)$	-0.000	0.000	-0.001	0.033	0.026	0.037	0.947	0.954	0.944	0.019	0.016	0.015	0.164	0.140	0.145	0.946	0.955	0.944
Using 5p as base scenario																		
TE	-0.001	-0.000	-0.001	0.042	0.035	0.042	0.947	0.948	0.943	0.007	0.022	0.005	0.102	0.151	0.101	0.949	0.956	0.947
$NDE(\cdot 0)$	0.001	-0.000	-0.002	0.047	0.041	0.045	0.945	0.942	0.945	0.006	0.012	0.000	0.106	0.166	0.101	0.947	0.944	0.941
$NIE(1\cdot)$	-0.001	-0.000	0.000	0.049	0.042	0.047	0.949	0.945	0.952	0.006	0.023	0.009	0.099	0.153	0.096	0.943	0.950	0.950
$NIE(0\cdot)$	-0.001	-0.001	0.000	0.049	0.042	0.047	0.948	0.953	0.955	0.003	0.018	0.005	0.112	0.137	0.108	0.947	0.954	0.954
$NDE(\cdot 1)$	0.001	0.000	-0.002	0.047	0.041	0.045	0.945	0.942	0.945	0.010	0.004	0.004	0.095	0.061	0.089	0.946	0.942	0.938
Using 6p as base scenario																		
TE	0.001	0.000	0.001	0.030	0.032	0.036	0.956	0.949	0.944	0.033	0.028	0.016	0.184	0.171	0.134	0.950	0.948	0.956
$NDE(\cdot 0)$	0.000	0.001	0.001	0.053	0.041	0.057	0.938	0.947	0.935	0.024	0.020	0.014	0.233	0.192	0.171	0.937	0.952	0.936
$NIE(1\cdot)$	0.001	-0.000	-0.000	0.049	0.040	0.055	0.941	0.959	0.931	0.036	0.025	0.019	0.195	0.169	0.150	0.943	0.953	0.944
$NIE(0\cdot)$	0.000	-0.001	0.000	0.048	0.040	0.056	0.938	0.961	0.926	0.027	0.023	0.011	0.160	0.153	0.132	0.950	0.946	0.945
$NDE(\cdot 1)$	0.001	0.001	0.001	0.053	0.041	0.057	0.937	0.953	0.935	0.006	0.004	0.010	0.077	0.058	0.097	0.929	0.948	0.930

'base' = base scenario (uniform path strengths). 'symdiff' = path strengths varying symmetrically. 'asymdiff' = path strengths varying asymmetrically. Standardized bias (RMSE) is bias (RMSE) divided by the true effect.

Table 2

Bias, root mean square error (RMSE) and coverage (by 95% confidence intervals), over variations in mediator residual correlations (using WLSMV estimator with ordinal mediators)

	For RD-based effects												**For RR-based effects**											
	Bias				RMSE				Coverage				Standardized bias				Standardized RMSE				Coverage			
Effect	base	mixed	low	high	base	mixed	low	high	base	mixed	low	high	base	mixed	low	high	base	mixed	low	high	base	mixed	low	high
Using 2a as base scenario																								
TE	0.001	0.001	0.000		0.041	0.042	0.041		0.950	0.941	0.952		0.010	0.011	0.009		0.105	0.107	0.105		0.950	0.950	0.945	
$NDE(\cdot 0)$	0.001	0.001	0.001		0.049	0.044	0.044		0.949	0.946	0.950		0.007	0.009	0.006		0.114	0.105	0.105		0.947	0.949	0.950	
$NIE(1\cdot)$	0.000	-0.000	-0.000		0.047	0.042	0.042		0.956	0.950	0.952		0.008	0.006	0.006		0.095	0.084	0.085		0.951	0.951	0.951	
$NIE(0\cdot)$	0.000	-0.000	-0.000		0.047	0.042	0.042		0.958	0.952	0.954		0.006	0.006	0.005		0.109	0.102	0.103		0.961	0.953	0.952	
$NDE(\cdot 1)$	0.001	0.001	0.000		0.049	0.044	0.044		0.951	0.948	0.947		0.009	0.009	0.007		0.098	0.087	0.085		0.947	0.948	0.949	
Using 7p as base scenario																								
TE	-0.000	-0.001	-0.000	-0.001	0.046	0.046	0.046	0.046	0.936	0.934	0.937	0.941	0.004	0.003	0.003	0.003	0.093	0.094	0.093	0.094	0.942	0.943	0.945	0.951
$NDE(\cdot 0)$	-0.000	-0.001	0.000	-0.001	0.050	0.044	0.044	0.067	0.941	0.947	0.945	0.933	0.003	0.002	0.004	0.002	0.082	0.074	0.073	0.102	0.946	0.948	0.949	0.946
$NIE(1\cdot)$	-0.000	-0.000	-0.001	0.001	0.049	0.041	0.042	0.065	0.943	0.953	0.944	0.946	0.003	0.002	0.001	0.007	0.079	0.070	0.071	0.099	0.947	0.956	0.951	0.950
$NIE(0\cdot)$	-0.000	-0.000	-0.001	0.001	0.049	0.041	0.042	0.064	0.945	0.951	0.948	0.942	0.000	0.001	-0.002	0.003	0.159	0.133	0.128	0.233	0.940	0.949	0.943	0.939
$NDE(\cdot 1)$	-0.000	-0.001	0.000	-0.001	0.050	0.044	0.044	0.067	0.942	0.943	0.946	0.936	0.026	0.016	0.018	0.058	0.175	0.145	0.139	0.288	0.942	0.950	0.950	0.931
Using 10c as base scenario																								
TE	0.000	0.001	0.001		0.045	0.045	0.045		0.948	0.947	0.946		0.005	0.006	0.006		0.091	0.092	0.090		0.954	0.952	0.957	
$NDE(\cdot 0)$	0.001	0.000	0.002		0.081	0.104	0.044		0.942	0.923	0.947		0.004	0.004	0.005		0.125	0.154	0.073		0.946	0.934	0.949	
$NIE(1\cdot)$	-0.001	0.001	-0.001		0.081	0.103	0.044		0.944	0.920	0.948		0.013	0.024	0.002		0.132	0.175	0.073		0.943	0.928	0.951	
$NIE(0\cdot)$	-0.001	0.001	-0.001		0.082	0.103	0.044		0.944	0.921	0.949		-0.001	0.002	-0.002		0.247	0.343	0.130		0.939	0.922	0.939	
$NDE(\cdot 1)$	0.001	0.001	0.002		0.081	0.104	0.044		0.940	0.920	0.947		0.075	0.158	0.021		0.330	0.581	0.139		0.929	0.915	0.947	

'base' = base scenario (uniform correlation strengths). 'mixed' = mixed correlation strengths. 'low' = low correlation strengths. 'high' = high correlation strengths. Standardized bias (RMSE) is bias (RMSE) divided by the true effect.

Table 3

Comparing WLSMV and Bayes estimators when the mediator residual matrix is semi-positive-definite

		With continuous mediators								With ordinal mediators							
	True	Bias		RMSE		Average SE		Median SE		Bias		RMSE		Average SE		Median SE	
	value	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes	WLSMV	Bayes
RD-based effects: all metrics on the raw proportion scale																	
TE	0.711	0.001	-0.002	0.031	0.030	0.031	0.032	0.031	0.032	0.001	0.005	0.031	0.031	0.031	0.031	0.031	0.031
$NDE(\cdot 0)$	0.153	-0.000	-0.009	0.018	0.019	3.271	0.232	1.375	0.232	0.022	0.050	0.126	0.147	2.064	0.189	0.836	0.194
$NIE(1\cdot)$	0.557	0.001	0.013	0.029	0.031	3.271	0.231	1.373	0.231	-0.022	-0.047	0.129	0.149	2.064	0.190	0.836	0.195
$NIE(0\cdot)$	0.557	0.001	0.007	0.028	0.028	3.268	0.231	1.379	0.232	-0.022	-0.046	0.129	0.148	2.073	0.190	0.812	0.194
$NDE(\cdot 1)$	0.153	-0.000	-0.011	0.018	0.020	3.268	0.232	1.377	0.233	0.023	0.049	0.127	0.147	2.073	0.189	0.811	0.193
RR-based effects: all metrics standardized by dividing by the true effects																	
TE	5.917	0.029	0.021	0.175	0.161	0.165	0.173	0.158	0.166	0.029	0.045	0.175	0.177	0.165	0.171	0.158	0.164
$NDE(\cdot 0)$	2.062	0.014	-0.025	0.108	0.102	11.132	0.815	4.644	0.802	0.092	0.185	0.461	0.517	7.211	0.694	2.842	0.676
$NIE(1\cdot)$	2.870	0.011	0.052	0.095	0.110	11.212	7.029	4.672	6.124	21.197	0.056	637.518	0.561	337.587	6.313	2.905	2.785
$NIE(0\cdot)$	4.855	0.026	0.048	0.159	0.155	4.774	0.358	2.029	0.352	-0.007	-0.048	0.236	0.249	3.110	0.328	1.206	0.319
$NDE(\cdot 1)$	1.219	0.001	-0.015	0.027	0.028	4.668	1.654	1.960	1.470	96792.888	0.137	3047466.568	0.358	5050218.236	1.387	1.295	0.915

This scenario combines the mediator residual correlation matrix $\begin{pmatrix} 1.00 & -0.82 & 0.30 \\ -0.82 & 1.00 & 0.30 \\ 0.30 & 0.30 & 1.00 \end{pmatrix}$ with path coefficients from scenario 1a. Results presented are from simulations for which WLSMV did not fail (990 and 992 out of 1000 simulations with continuous and ordinal mediators, respectively). RMSE = root mean squared error. SE = estimated standard error. RD = risk difference. RR = risk ratio.

is slightly more biased than WLSMV. Both estimators overestimate variance, and this problem is much worse for WLSMV, with very large SEs (average and median over 3.2 and 1.3 with continuous, and 2.0 and 0.8 with ordinal, mediators) for effects that are differences in proportions. Bayes SEs (average and median both about 0.23 with continuous, and 0.19 with ordinal, mediators) are much more modest.

For RR-based effects, with continuous mediators, both estimators perform quite well in terms of bias and RMSE. Both overestimate variance, and this problem is worse for WLSMV. With ordinal mediators, the WLSMV estimator is severely biased for effects that involve either of the truly counterfactual outcome probabilities (p_{10} or p_{01}) in the denominator. Bayes performs better with regards to bias, RMSE as well as estimated SEs.

Appendix D: Mplus input template/samples and corresponding R-code

This appendix includes an Mplus input template to fit models using WLSMV/ML estimator combined with R-code for bootstrapping; and an Mplus input sample to fit models using Bayes estimator combined with R-code to process the posterior distribution after model fitting. Both Mplus inputs are easy to tailor to a specific model. The reason one is called a template and the other a sample is that it is relatively straightforward to fit a model using WLSMV/ML (so you can quickly tailor the template and run it), but Bayesian analysis is likely to involve running the same model multiple times each time re-specifying options while assessing convergence, auto-correlation and effective sample size (so the ANALYSIS statement in the sample is suggestive rather than prescriptive).

Mplus input template for WLSMV/ML

```
TITLE: To be used with R-code for bootstrapping WLSMV/ML.
       Includes 2 continuous and 2 ordinal mediators, plus 2 confounders.

DATA: FILE = adat.dat;

VARIABLE:
  NAMES = z1 z2 x m1 m2 m3 m4 y;
  USEVAR = z1 z2 x m1 m2 m3 m4 y; ! confounders, exposure, all mediators, and outcome
  CATEGORICAL = y m3 m4;          ! binary outcome and the ordinal mediators

ANALYSIS:
  ESTIMATOR = WLSMV;
  PARAMETERIZATION = THETA;  ! need this when using WLSMV to ensure correct scaling

! If wish to use ML (which ONLY applies to ALL CONTINUOUS mediators),
! use these two lines instead:
  ESTIMATOR = ML;
  LINK = PROBIT;

MODEL:
  y ON x  (gamma)
       z1 (delta1)
       z2 (delta2)
       m1 (beta1)
       m2 (beta2)
       m3 (beta3)
       m4 (beta4);

  [y$1] (ytau); ! outcome threshold

  y@1; ! residual outcome variance. REMOVE if using ML.

  m1 ON x  (alpha1)
        z1 (lambda11)
        z2 (lambda12);

  m2 ON x  (alpha2)
        z1 (lambda21)
        z2 (lambda22);
```

```
  m3 ON x  (alpha3)
        z1 (lambda31)
        z2 (lambda32);

  m4 ON x  (alpha4)
        z1 (lambda41)
        z2 (lambda42);

  [m1] (m1mu); ! intercepts of continuous mediators
  [m2] (m2mu);

  m1 (var1); ! residual variances of continuous mediators
  m2 (var2);
  m3@1; ! residual variances of ordinal mediators. REMOVE these two lines if using ML.
  m4@1;

  m1 WITH m2 (rho12); ! residual covariances/correlations between mediators
  m1 WITH m3 (rho13);
  m1 WITH m4 (rho14);
  m2 WITH m3 (rho23);
  m2 WITH m4 (rho24);
  m3 WITH m4 (rho34);

MODEL CONSTRAINT:      ! Declare the new parameters from equation (6), including
  NEW(constant         ! the constant term,
      z1_coef          ! coefficients of the confounders z1
      z2_coef          ! and z2,
      x_coef           ! coefficient of exposure condition x,
      xp_coef          ! coefficient of exposure condition x',
      scale);          ! and scale parameter.

! Define these new parameters:

  constant = -ytau
            +beta1*m1mu  ! one of these terms for each continuous mediator
            +beta2*m2mu;

  z1_coef = delta1+
           beta1*lambda11+
           beta2*lambda21+
           beta3*lambda31+
           beta4*lambda41;

  z2_coef = delta2+
           beta1*lambda12+
           beta2*lambda22+
           beta3*lambda32+
           beta4*lambda42;

  x_coef  = gamma;

  xp_coef = beta1*alpha1+beta2*alpha2+beta3*alpha3+beta4*alpha4;

  scale = sqrt(beta1*beta1*var1+    ! one of these terms per continuous mediator
               beta2*beta2*var2+
```

```
               beta3*beta3+         ! one of these terms per ordinal mediator
               beta4*beta4+
               2*beta1*beta2*rho12+ ! one of these terms per pair of mediators
               2*beta1*beta3*rho13+
               2*beta1*beta4*rho14+
               2*beta2*beta3*rho23+
               2*beta2*beta4*rho24+
               2*beta3*beta4*rho34+
               1);                  ! Please take care to include this number 1.

! END OF TEMPLATE!


! A USEFUL TRICK: Since the new parameters will be used in subsequent computations, if
! they  are small, you may want more precision than that provided by the three decimal
! places in Mplus output. In the MODEL CONSTRAINT statement, simply multiply ALL OF THEM
! by, say, 100 (see below). This is equivalent to adding two decimal places!
!
! constant = (-ytau
!             +beta1*m1mu
!             +beta2*m2mu)*100;
!
! z1_coef = (delta1+
!            beta1*lambda11+
!            beta2*lambda21+
!            beta3*lambda31+
!            beta4*lambda41)*100;
!
! z2_coef = (delta2+
!            beta1*lambda12+
!            beta2*lambda22+
!            beta3*lambda32+
!            beta4*lambda42)*100;
!
! x_coef  = gamma*100;
!
! xp_coef = (beta1*alpha1+beta2*alpha2+beta3*alpha3+beta4*alpha4)*100;
!
! scale = (sqrt(beta1*beta1*var1+
!               beta2*beta2*var2+
!               beta3*beta3+
!               beta4*beta4+
!               2*beta1*beta2*rho12+
!               2*beta1*beta3*rho13+
!               2*beta1*beta4*rho14+
!               2*beta2*beta3*rho23+
!               2*beta2*beta4*rho24+
!               2*beta3*beta4*rho34+
!               1))*100;
!
! While these quantities are inflated, the potential outcome probabilities and causal
! mediation effects will not be affected, because the multiplier cancels itself out.
```

R-code to bootstrap when using WLSMV/ML

```
# This code calls on Mplus to fit the structural equation model to the dataset; extract
# the new parameters defined in the MODEL CONSTRAINT statement; and combine them with
# data on the confounders to compute point estimates of causal mediation effects.

# It also does the same with 500 bootstrap samples drawn from the dataset, which results
# in 500 sets of bootstrap estimates of the causal mediation effects; and uses these to
# derive their standard errors and confidence intervals.

### Preparations
#################

# Put the data (dat.dat) and Mplus input (sem.inp) in the same folder.
# Set it as the working directory.
setwd("C:/Projects/CausalMediation") # change this to specify the path to your folder

# Install and load the package required to work with Mplus.
install.packages("MplusAutomation")  # skip if already installed
library(MplusAutomation)

# Read the dataset into R.
dat <- read.table("dat.dat")

# Check that the data are good. If needed, revise the NAMES = statement in sem.inp.
names(dat); dim(dat); head(dat); tail(dat)
n <- nrow(dat)  # get sample size

# Extract data on the confounders only.
confounders <- dat[,1:2] # here two confounders are in columns 1 and 2 of the dataset

# Make four "counterfactual datasets", which will be combined with the new parameters to
# compute causal mediation effects. These datasets differ only by the last two columns.

dat.00 <- cbind(rep(1,n),    # a column of 1's (to multiply with parameter constant)
                confounders, # confounder data (to multiply with z1_coef and z2_coef)
                rep(0,n),    # a column of 0's for x = 0 (to multiply with x_coef)
                rep(0,n))    # a column of 0's for x'= 0 (to multiply with xp_coef)

dat.11 <- cbind(rep(1,n),confounders,
                rep(1,n),rep(1,n)) # columns of 1's and 1's for x = 1 and x'= 1
dat.10 <- cbind(rep(1,n),confounders,
                rep(1,n),rep(0,n)) # columns of 1's and 0's for x = 1 and x'= 0

dat.01 <- cbind(rep(1,n),confounders,
                rep(0,n),rep(1,n)) # columns of 0's and 1's for x = 0 and x'= 1

#### Main analysis to get point estimates
##########################################

# Write out an Mplus data file named adat.dat (the name referred to in sem.inp).
prepareMplusData(dat,"adat.dat")

# Call Mplus to run models in the folder, which is the only one model sem.inp.
runModels()
```

```
# Open the output file sem.out and check if everything about the model is correct!

# Extract estimates of the new parameters from sem.out.
main.est <- paramExtract(extractModelParameters("sem.out")$unstandardized,params="new")

# Look at main.est. It is a table and the point estimates are in column 3.
main.est

# Grab the scale parameter from the last row of main.est.
main.scale <- main.est[nrow(main.est),3]

# Grab the coefficients (constant, z1_coef, z2_coef, x_coef, xp_coef).
main.coefs <- main.est[-nrow(main.est),3]

# Compute the four potential outcome probabilities, by:
#   multiplying the conterfactual datasets with the coefficients extracted,
#   then dividing by the scale parameter extracted,
#   then taking the inverse-probit,
#   and finally averaging over the sample -- all in the same lines of code.
main.p00 <- mean(pnorm((dat.00%*%main.coefs)/main.scale))
main.p11 <- mean(pnorm((dat.11%*%main.coefs)/main.scale))
main.p10 <- mean(pnorm((dat.10%*%main.coefs)/main.scale))
main.p01 <- mean(pnorm((dat.01%*%main.coefs)/main.scale))

# Compute point estimates of causal mediation effects.
# Here we include both RD- and RR-based effects and both ways of decomposing TE.
# Select the effects of interest in your analysis.

main.TE.RD    <- main.p11-main.p00
main.NDE.0.RD <- main.p10-main.p00
main.NIE.1.RD <- main.p11-main.p10
main.NIE.0.RD <- main.p01-main.p00
main.NDE.1.RD <- main.p11-main.p01

main.TE.RR    <- main.p11/main.p00
main.NDE.0.RR <- main.p10/main.p00
main.NIE.1.RR <- main.p11/main.p10
main.NIE.0.RR <- main.p01/main.p00
main.NDE.1.RR <- main.p11/main.p01
### Bootstrap for standard errors and confidence intervals
############################################################

## First, fit the model to bootstrap samples and collect estimates of new parameters.

boot.n <- 500         # specify number of bootstrap samples
boot.est.mat <- NULL  # initialize matrix to hold estimates of new parameters
set.seed(67890)       # set seed for reproducible results (you can use another seed)
j <- 0                # initialize counter of bootstrap steps
while(j<boot.n){
  cat(paste("BOOTSTRAP STEP",j+1))                 # print step number to screen
  boot <- dat[sample(1:nrow(dat),replace=TRUE),]  # draw a bootstrap dataset
  # From here the code mimics main analysis.
  prepareMplusData(boot,"adat.dat")
  runModels()
  boot.est <- paramExtract(extractModelParameters("sem.out")$unstandardized,params="new")
```

```
  boot.est <- boot.est[,3]
  if(class(boot.est)=="numeric"){                # if the run was good,
    boot.est.mat <- rbind(boot.est.mat,boot.est)  # store estimates of new parameters
    j <- j+1
  }
}
rownames(boot.est.mat) <- NULL

## Second, combine new parameters with data to compute potential outcome probabilities
## (for each of all the bootstrap datasets).

boot.coefs.mat <- boot.est.mat[,-ncol(boot.est.mat)]
boot.scale.vec <- boot.est.mat[, ncol(boot.est.mat)]
boot.scale.mat <- matrix(rep(boot.scale.vec,n),ncol=n)

boot.p00 <- rowMeans(pnorm((boot.coefs.mat%*%t(dat.00))/boot.scale.mat))
boot.p11 <- rowMeans(pnorm((boot.coefs.mat%*%t(dat.11))/boot.scale.mat))
boot.p10 <- rowMeans(pnorm((boot.coefs.mat%*%t(dat.10))/boot.scale.mat))
boot.p01 <- rowMeans(pnorm((boot.coefs.mat%*%t(dat.01))/boot.scale.mat))

boot.pxx.eff <- data.frame(cbind(boot.p00,boot.p11,boot.p10,boot.p01))
names(boot.pxx.eff) <- c("p00","p11","p10","p01")

## Third, compute causal mediation effects (for each of all the bootstrap datasets).
## Again, here we include both RD- and RR-based effects and both ways of decomposing TE.
## Select the effects of interest in your analysis.

boot.pxx.eff$TE.RD    <- boot.pxx.eff$p11-boot.pxx.eff$p00
boot.pxx.eff$NDE.0.RD <- boot.pxx.eff$p10-boot.pxx.eff$p00
boot.pxx.eff$NIE.1.RD <- boot.pxx.eff$p11-boot.pxx.eff$p10
boot.pxx.eff$NIE.0.RD <- boot.pxx.eff$p01-boot.pxx.eff$p00
boot.pxx.eff$NDE.1.RD <- boot.pxx.eff$p11-boot.pxx.eff$p01

boot.pxx.eff$TE.RR    <- boot.pxx.eff$p11/boot.pxx.eff$p00
boot.pxx.eff$NDE.0.RR <- boot.pxx.eff$p10/boot.pxx.eff$p00
boot.pxx.eff$NIE.1.RR <- boot.pxx.eff$p11/boot.pxx.eff$p10
boot.pxx.eff$NIE.0.RR <- boot.pxx.eff$p01/boot.pxx.eff$p00
boot.pxx.eff$NDE.1.RR <- boot.pxx.eff$p11/boot.pxx.eff$p01
## Fourth, derive standard errors for causal mediation effects from bootstrap estimates

boot.TE.RD.se    <- sd(boot.pxx.eff$TE.RD)
boot.NDE.0.RD.se <- sd(boot.pxx.eff$NDE.0.RD)
boot.NIE.1.RD.se <- sd(boot.pxx.eff$NIE.1.RD)
boot.NIE.0.RD.se <- sd(boot.pxx.eff$NIE.0.RD)
boot.NDE.1.RD.se <- sd(boot.pxx.eff$NDE.1.RD)

boot.TE.RR.se    <- sd(boot.pxx.eff$TE.RR)
boot.NDE.0.RR.se <- sd(boot.pxx.eff$NDE.0.RR)
boot.NIE.1.RR.se <- sd(boot.pxx.eff$NIE.1.RR)
boot.NIE.0.RR.se <- sd(boot.pxx.eff$NIE.0.RR)
boot.NDE.1.RR.se <- sd(boot.pxx.eff$NDE.1.RR)

## Fifth, derive confidence intervals for effects from bootstrap estimates

boot.TE.RD.ci    <- quantile(boot.pxx.eff$TE.RD    ,probs=c(.025,.975))
```

```
boot.NDE.0.RD.ci <- quantile(boot.pxx.eff$NDE.0.RD,probs=c(.025,.975))
boot.NIE.1.RD.ci <- quantile(boot.pxx.eff$NIE.1.RD,probs=c(.025,.975))
boot.NIE.0.RD.ci <- quantile(boot.pxx.eff$NIE.0.RD,probs=c(.025,.975))
boot.NDE.1.RD.ci <- quantile(boot.pxx.eff$NDE.1.RD,probs=c(.025,.975))

boot.TE.RR.ci    <- quantile(boot.pxx.eff$TE.RR   ,probs=c(.025,.975))
boot.NDE.0.RR.ci <- quantile(boot.pxx.eff$NDE.0.RR,probs=c(.025,.975))
boot.NIE.1.RR.ci <- quantile(boot.pxx.eff$NIE.1.RR,probs=c(.025,.975))
boot.NIE.0.RR.ci <- quantile(boot.pxx.eff$NIE.0.RR,probs=c(.025,.975))
boot.NDE.1.RR.ci <- quantile(boot.pxx.eff$NDE.1.RR,probs=c(.025,.975))

### Combine point estimates, SEs and CIs in a summary table
############################################################

estimate <- c(main.TE.RD,main.NDE.0.RD,main.NIE.1.RD,main.NIE.0.RD,main.NDE.1.RD,
              main.TE.RR,main.NDE.0.RR,main.NIE.1.RR,main.NIE.0.RR,main.NDE.1.RR)

se <- c(boot.TE.RD.se,boot.NDE.0.RD.se,boot.NIE.1.RD.se,boot.NIE.0.RD.se,boot.NDE.1.RD.se,
        boot.TE.RR.se,boot.NDE.0.RR.se,boot.NIE.1.RR.se,boot.NIE.0.RR.se,boot.NDE.1.RR.se)

ll <- c(boot.TE.RD.ci[1],
        boot.NDE.0.RD.ci[1],boot.NIE.1.RD.ci[1],
        boot.NIE.0.RD.ci[1],boot.NDE.1.RD.ci[1],
        boot.TE.RR.ci[1],
        boot.NDE.0.RR.ci[1],boot.NIE.1.RR.ci[1],
        boot.NIE.0.RR.ci[1],boot.NDE.1.RR.ci[1])
ul <- c(boot.TE.RD.ci[2],
        boot.NDE.0.RD.ci[2],boot.NIE.1.RD.ci[2],
        boot.NIE.0.RD.ci[2],boot.NDE.1.RD.ci[2],
        boot.TE.RR.ci[2],
        boot.NDE.0.RR.ci[2],boot.NIE.1.RR.ci[2],
        boot.NIE.0.RR.ci[2],boot.NDE.1.RR.ci[2])

summary <- cbind(estimate,se,ll,ul)
rownames(summary) <- c("TE.RD","NDE.0.RD","NIE.1.RD","NIE.0.RD","NDE.1.RD",
                       "TE.RR","NDE.0.RR","NIE.1.RR","NIE.0.RR","NDE.1.RR")
colnames(summary) <- c("estimate","SE","2.5%","97.5%")
```

Mplus input sample for Bayes estimator

```
TITLE: To be used with R-code for processing posterior distribution from Bayes model.
       Includes 2 continuous and 2 ordinal mediators, plus 2 confounders.
DATA: FILE = dat.dat;

VARIABLE:
  NAMES = x m1 m2 m3 m4 y;
  USEVAR = x m1 m2 m3 m4 y; ! exposure, all mediators, and outcome
  CATEGORICAL = y m3 m4;    ! binary outcome and the ordinal mediators

ANALYSIS: ! This section includes options that need (re-)specifying/adjusting.
  ESTIMATOR = BAYES;
  MEDIATOR = LATENT; ! if ordinal mediators
  CHAINS = 3;
  PROCESSORS = 3;
  BITERATIONS = 50000 (3000);
  BCONVERGENCE = 0.001
  THIN = 50;

MODEL:
  y ON x;
  y ON z1;
  y ON z2;
  y ON m1;
  y ON m2;
  y ON m3;
  y ON m4;

  m1 ON x;
  m1 ON z1;
  m1 ON z2;

  m2 ON x;
  m2 ON z1;
  m2 ON z2;

  m3 ON x;
  m3 ON z1;
  m3 ON z2;

  m4 ON x;
  m4 ON z1;
  m4 ON z2;

  m1 WITH m2;
  m1 WITH m3;
  m1 WITH m4;
  m2 WITH m3;
  m2 WITH m4;
  m3 WITH m4;

SAVEDATA:
  BPARAMETERS = "bparas.dat" ! This saves the posterior distribution of model parameters.
```

R-code to process posterior distribution from Bayesian model

```
# Use this AFTER finishing fitting the model with Bayes estimator in Mplus.

### Preparations
#################

# Install and load required R packages
install.packages("MplusAutomation") # skip if already installed
install.packages("coda")            # skip if already installed
library(MplusAutomation)
libary(coda)

# Read in data and get sample size
dat <- read.table("dat.dat")
n <- nrow(dat)

# Extract data on the confounders only.
confounders <- dat[,1:2] # here two confounders are in columns 1 and 2 of the dataset

# Make four "counterfactual datasets", which will be combined with the new parameters to
# compute causal mediation effects. These datasets differ only by the last two columns.

dat.00 <- cbind(rep(1,n),    # a column of 1's (to multiply with parameter constant)
                confounders, # confounder data (to multiply with z1_coef, z2_coef)
                rep(0,n),    # a column of 0's for x = 0 (to multiply with x_coef)
                rep(0,n))    # a column of 0's for x'= 0 (to multiply with xp_coef)

dat.11 <- cbind(rep(1,n),confounders,
                rep(1,n),rep(1,n)) # columns of 1's and 1's for x = 1 and x'= 1

dat.10 <- cbind(rep(1,n),confounders,
                rep(1,n),rep(0,n)) # columns of 1's and 0's for x = 1 and x'= 0

dat.01 <- cbind(rep(1,n),confounders,
                rep(0,n),rep(1,n)) # columns of 0's and 1's for x = 0 and x'= 1

### Extract the posterior distribution of model parameters from file bparas.dat
################################################################################

post <- getSavedata_Bparams("bparas.dat",discardBurnin=TRUE)
post <- as.matrix(post,iters=TRUE)

# Open the matrix post and look at the contents of the columns
head(post)
names(post)

# Rename the columns in post using: iteration, chain for the first 2 columns, and then
# parameter names, in the order they appear in the matrix. The command below is only a
# suggestion, as the order of the parameters may be different.



colnames(post) <- c("iteration","chain",
                    "m1mu","m2mu",   # intercepts of continuous mediators
```

```
                        "beta1","beta2", # beta coefficients for continuous mediators
                        "lambda11","lambda12","alpha1", # lambda and alpha coefficients,
                        "lambda21"."lambda22","alpha2", # one row per mediator
                        "lambda31","lambda32","alpha3",
                        "lambda41","lambda42","alpha4",
                        "beta3","beta4",       # beta coefficients for ordinal mediators
                        "delta1","delta2",     # delta coefficients for confounders
                        "gamma",               # gamma coefficient
                                               # residual covariance matrix, including
                        "var1","cov12","var2", # residual variances of continuous mediators
                        "cov13","cov14","cov23","cov24","cov34",
                        "m3tau1","m3tau2","m3tau3", # thresholds of ordinal mediators
                        "m4tau1","m4tau2","m4tau3",
                        "ytau")                     # threshold of the outcome

### Derive the posterior distribution of the new parameters from equation (6)
###############################################################################

post <- data.frame(post)

post$constant <- -post$ytau +
                  post$beta1*post$m1mu+ # one of these lines for each continuous mediator
                  post$beta2*post$m2mu

post$z1.coef <- post$beta1*post$lambda11+ # one of these lines per mediator
                post$beta2*post$lambda21+
                post$beta3*post$lambda31+
                post$beta3*post$lambda41

post$z2.coef <- post$beta1*post$lambda12+ # one of these lines per mediator
                post$beta2*post$lambda22+
                post$beta3*post$lambda32+
                post$beta3*post$lambda42

post$x.coef  <- post$gamma

post$xp.coef <- post$beta1*post$alpha1+ # one of these lines per mediator
                post$beta2*post$alpha2+
                post$beta3*post$alpha3+
                post$beta3*post$alpha3

post$scale <- sqrt(post$beta1*post$beta1*post$var1+ # one of these lines per continuous mediator
                   post$beta2*post$beta2*post$var2+
                   post$beta3*post$beta3+           # one of these lines per ordinal mediator
                   post$beta4*post$beta4+
                   2*post$beta1*post$beta2*post$cov12+ # one of these lines per pair of mediators
                   2*post$beta1*post$beta3*post$cov13+
                   2*post$beta1*post$beta4*post$cov14+
                   2*post$beta2*post$beta3*post$cov23+
                   2*post$beta2*post$beta4*post$cov24+
                   2*post$beta3*post$beta4*post$cov34 +
                   1)                                  # Please be sure to include this number 1.
### Compute potential outcome probabilities's posterior distribution
######################################################################
```

```
# This is done by multiplying the conterfactual datasets with the posterior distribution
# of equation (6) coefficients,then dividing by the posterior distribution of the scale
# parameter, then taking the inverse-probit, and finally averaging over the sample.

post.coefs.mat <- cbind(post$constant,post$z1.coef,post$z2.coef,post$x.coef,post$xp.coef)
post.scale.mat <- matrix(rep(post$scale,n),ncol=n)

post$p00 <- rowMeans(pnorm((post.coefs.mat%*%t(dat.00))/post.scale.mat))
post$p11 <- rowMeans(pnorm((post.coefs.mat%*%t(dat.11))/post.scale.mat))
post$p10 <- rowMeans(pnorm((post.coefs.mat%*%t(dat.10))/post.scale.mat))
post$p01 <- rowMeans(pnorm((post.coefs.mat%*%t(dat.01))/post.scale.mat))

### Compute the posterior distribution of causal mediation effects
###################################################################

# Here we include both RD- and RR-based effects and both ways of decomposing TE.
# Select the effects of interest in your analysis.

post$TE.RD    <- post$p11-post$p00
post$NDE.0.RD <- post$p10-post$p00
post$NIE.1.RD <- post$p11-post$p10
post$NIE.0.RD <- post$p01-post$p00
post$NDE.1.RD <- post$p11-post$p01

post$TE.RR    <- post$p11/post$p00
post$NDE.0.RR <- post$p10/post$p00
post$NIE.1.RR <- post$p11/post$p10
post$NIE.0.RR <- post$p01/post$p00
post$NDE.1.RR <- post$p11/post$p01

### Check the effective sample size of the effects of interest
###############################################################

effectiveSize(post[,(ncol(post)-9):ncol(post)])

# If the effective sample size for any of the effects of interest is small, you might
# want to go back to the Mplus modeling stage and get longer MCMC chains or increase
# thinning to reduce auto-correlation. If this is not a problem, go to the next step.

### Derive point estimates, standard errors and credible intervals of the causal
### mediation effects from the posterior distribution
################################################################################

# Median point estimates

post.TE.RD.median    <- median(post$TE.RD)
post.NDE.0.RD.median <- median(post$NDE.0.RD)
post.NIE.1.RD.median <- median(post$NIE.1.RD)
post.NIE.0.RD.median <- median(post$NIE.0.RD)
post.NDE.1.RD.median <- median(post$NDE.1.RD)

post.TE.RR.median    <- median(post$TE.RR)
post.NDE.0.RR.median <- median(post$NDE.0.RR)
post.NIE.1.RR.median <- median(post$NIE.1.RR)
post.NIE.0.RR.median <- median(post$NIE.0.RR)
```

```
post.NDE.1.RR.median <- median(post$NDE.1.RR)

# Standard errors

post.TE.RD.se    <- sd(post$TE.RD)
post.NDE.0.RD.se <- sd(post$NDE.0.RD)
post.NIE.1.RD.se <- sd(post$NIE.1.RD)
post.NIE.0.RD.se <- sd(post$NIE.0.RD)
post.NDE.1.RD.se <- sd(post$NDE.1.RD)

post.TE.RR.se    <- sd(post$TE.RR)
post.NDE.0.RR.se <- sd(post$NDE.0.RR)
post.NIE.1.RR.se <- sd(post$NIE.1.RR)
post.NIE.0.RR.se <- sd(post$NIE.0.RR)
post.NDE.1.RR.se <- sd(post$NDE.1.RR)

# Quantile credible intervals

post.TE.RD.ci    <- quantile(post$TE.RD   ,probs=c(.025,.975))
post.NDE.0.RD.ci <- quantile(post$NDE.0.RD,probs=c(.025,.975))
post.NIE.1.RD.ci <- quantile(post$NIE.1.RD,probs=c(.025,.975))
post.NIE.0.RD.ci <- quantile(post$NIE.0.RD,probs=c(.025,.975))
post.NDE.1.RD.ci <- quantile(post$NDE.1.RD,probs=c(.025,.975))

post.TE.RR.ci    <- quantile(post$TE.RR   ,probs=c(.025,.975))
post.NDE.0.RR.ci <- quantile(post$NDE.0.RR,probs=c(.025,.975))
post.NIE.1.RR.ci <- quantile(post$NIE.1.RR,probs=c(.025,.975))
post.NIE.0.RR.ci <- quantile(post$NIE.0.RR,probs=c(.025,.975))
post.NDE.1.RR.ci <- quantile(post$NDE.1.RR,probs=c(.025,.975))

# Summary table

summary <-
  cbind(c(post.TE.RD.median,
          post.NDE.0.RD.median,post.NIE.1.RD.median,post.NIE.0.RD.median,post.NDE.1.RD.median,
          post.TE.RR.median,
          post.NDE.0.RR.median,post.NIE.1.RR.median,post.NIE.0.RR.median,post.NDE.1.RR.median),
        c(post.TE.RD.se,post.NDE.0.RD.se,post.NIE.1.RD.se,post.NIE.0.RD.se,post.NDE.1.RD.se,
          post.TE.RR.se,post.NDE.0.RR.se,post.NIE.1.RR.se,post.NIE.0.RR.se,post.NDE.1.RR.se),
        c(post.TE.RD.ci[1],
          post.NDE.0.RD.ci[1],post.NIE.1.RD.ci[1],post.NIE.0.RD.ci[1],post.NDE.1.RD.ci[1],
          post.TE.RR.ci[1],
          post.NDE.0.RR.ci[1],post.NIE.1.RR.ci[1],post.NIE.0.RR.ci[1],post.NDE.1.RR.ci[1]),
        c(post.TE.RD.ci[2],
          post.NDE.0.RD.ci[2],post.NIE.1.RD.ci[2],post.NIE.0.RD.ci[2],post.NDE.1.RD.ci[2],
          post.TE.RR.ci[2],
          post.NDE.0.RR.ci[2],post.NIE.1.RR.ci[2],post.NIE.0.RR.ci[2],post.NDE.1.RR.ci[2]))
rownames(summary) <- c("TE.RD","NDE.0.RD","NIE.1.RD","NIE.0.RD","NDE.1.RD",
                       "TE.RR","NDE.0.RR","NIE.1.RR","NIE.0.RR","NDE.1.RR")
colnames(summary) <- c("estimate","SE","2.5%","97.5%")
```

Appendix E: Mplus input for the illustrative example

```
TITLE: Adolescent Alcohol Prevention -- Model with four mediators
DATA: FILE = "adat.dat";

VARIABLE:
NAMES = id condition school class
    age0 gender edu christian islamic othreligion paredu
    drink0 wkdrink2
    acntrl0 acntrl1 arule0 arule1 aatt0 aatt1
    prule0 prule1 patt0 patt1;
MISSING=.;
USEVAR = age0 gender edu paredu christian islamic othreligion
    condition drink0 wkdrink2
    acntrl0 acntrl1 arule0 arule1 aatt0 aatt1 patt0 patt1;
CATEGORICAL = wkdrink2;
CLUSTER = class;

ANALYSIS:
TYPE = COMPLEX;
ESTIMATOR = MLR;
LINK = PROBIT;
INTEGRATION = MONTECARLO;

MODEL:
acntrl1 ON condition (alpha1)
    age0        (lambda101)
    gender      (lambda102)
    edu         (lambda103)
    paredu      (lambda104)
    christian   (lambda105)
    islamic     (lambda106)
    othreligion (lambda107)
    acntrl0     (lambda108)
    arule0      (lambda109)
    aatt0       (lambda110)
    drink0      (lambda112)
    ;
arule1 ON condition (alpha2)
    age0        (lambda201)
    gender      (lambda202)
    edu         (lambda203)
    paredu      (lambda204)
    christian   (lambda205)
    islamic     (lambda206)
    othreligion (lambda207)
    acntrl0     (lambda208)
    arule0      (lambda209)
    aatt0       (lambda210)
    patt0       (lambda211)
    drink0      (lambda212)
    ;
aatt1 ON condition (alpha3)
    age0        (lambda301)
    gender      (lambda302)
```

```
    edu         (lambda303)
    paredu      (lambda304)
    christian   (lambda305)
    islamic     (lambda306)
    othreligion (lambda307)
    acntrl0     (lambda308)
    arule0      (lambda309)
    aatt0       (lambda310)
    drink0      (lambda312)
    ;
patt1 ON condition (alpha4)
    age0        (lambda401)
    gender      (lambda402)
    edu         (lambda403)
    paredu      (lambda404)
    christian   (lambda405)
    islamic     (lambda406)
    othreligion (lambda407)
    patt0       (lambda411)
    drink0      (lambda412)
    ;

[acntrl1] (m1mu);
[arule1]  (m2mu);
[aatt1]   (m3mu);
[patt1]   (m4mu);

acntrl1 (var1);
arule1  (var2);
aatt1   (var3);
patt1   (var4);

acntrl1 WITH arule1 (cov12);
acntrl1 WITH aatt1  (cov13);
acntrl1 WITH patt1  (cov14);
arule1  WITH aatt1  (cov23);
arule1  WITH patt1  (cov24);
aatt1   WITH patt1  (cov34);

wkdrink2 ON condition (gamma)
    acntrl1 (beta1)
    arule1  (beta2)
    aatt1   (beta3)
    patt1   (beta4)
    age0        (delta01)
    gender      (delta02)
    edu         (delta03)
    paredu      (delta04)
    christian   (delta05)
    islamic     (delta06)
    othreligion (delta07)
    drink0      (delta12)
    ;

[wkdrink2$1] (ytau);
```

```
MODEL CONSTRAINT:
NEW(constant
    z01_coef
    z02_coef
    z03_coef
    z04_coef
    z05_coef
    z06_coef
    z07_coef
    z08_coef
    z09_coef
    z10_coef
    z11_coef
    z12_coef
    x_coef
    xp_coef
    scale);

constant=(-ytau + beta1*m1mu+beta2*m2mu+beta3*m3mu+beta4*m4mu)*1000;
z01_coef=(beta1*lambda101+beta2*lambda201+beta3*lambda301+beta4*lambda401 + delta01)*1000;
z02_coef=(beta1*lambda102+beta2*lambda202+beta3*lambda302+beta4*lambda402 + delta02)*1000;
z03_coef=(beta1*lambda103+beta2*lambda203+beta3*lambda303+beta4*lambda403 + delta03)*1000;
z04_coef=(beta1*lambda104+beta2*lambda204+beta3*lambda304+beta4*lambda404 + delta04)*1000;
z05_coef=(beta1*lambda105+beta2*lambda205+beta3*lambda305+beta4*lambda405 + delta05)*1000;
z06_coef=(beta1*lambda106+beta2*lambda206+beta3*lambda306+beta4*lambda406 + delta06)*1000;
z07_coef=(beta1*lambda107+beta2*lambda207+beta3*lambda307+beta4*lambda407 + delta07)*1000;
z08_coef=(beta1*lambda108+beta2*lambda208+beta3*lambda308                         )*1000;
z09_coef=(beta1*lambda109+beta2*lambda209+beta3*lambda309                         )*1000;
z10_coef=(beta1*lambda110+beta2*lambda210+beta3*lambda310                         )*1000;
z11_coef=(                beta2*lambda211                +beta4*lambda411         )*1000;
z12_coef=(beta1*lambda112+beta2*lambda212+beta3*lambda312+beta4*lambda412 + delta12)*1000;
x_coef  =gamma*1000;
xp_coef =(beta1*alpha1+beta2*alpha2+beta3*alpha3+beta4*alpha4)*1000;
scale   =(sqrt(beta1*beta1*var1+
               beta2*beta2*var2+
               beta3*beta3*var3+
               beta4*beta4*var4+
               2*beta1*beta2*cov12+
               2*beta1*beta3*cov13+
               2*beta1*beta4*cov14+
               2*beta2*beta3*cov23+
               2*beta2*beta4*cov24+
               2*beta3*beta4*cov34 + 1))*1000;


! The *1000 above is the trick mentioned in Appendix D for increased precision.
```