Longitudinal Modeling

Nathan & Lindon

Template_Developmental_Twin_Continuous_Matrix.R

Template_Developmental_Twin_Ordinal_Matrix.R

jepq.txt

GenEpiHelperFunctions.R

Why run longitudinal models?

Map changes in the magnitude of genetic & environmental influence across time

ID same versus different genetic or environmental risks across development

ID factors driving change versus factors maintaining stability

Improve power to detect A, C & E

- using multiple observations from the same individual & the cross twin cross trait correlations

Common methods for longitudinal data analyses in genetic epidemiology

Cholesky Decomposition

- Advantages
  - Logical: organized such that all factors are constrained to impact later, but not earlier time points
  - Requires few assumptions, can predict any pattern of change
- Disadvantages
  - Not falsifiable
  - No predictions
  - Feasible for limited number of measurements

Latent Growth Curve Modeling

Simplex Modeling

Presentation layout

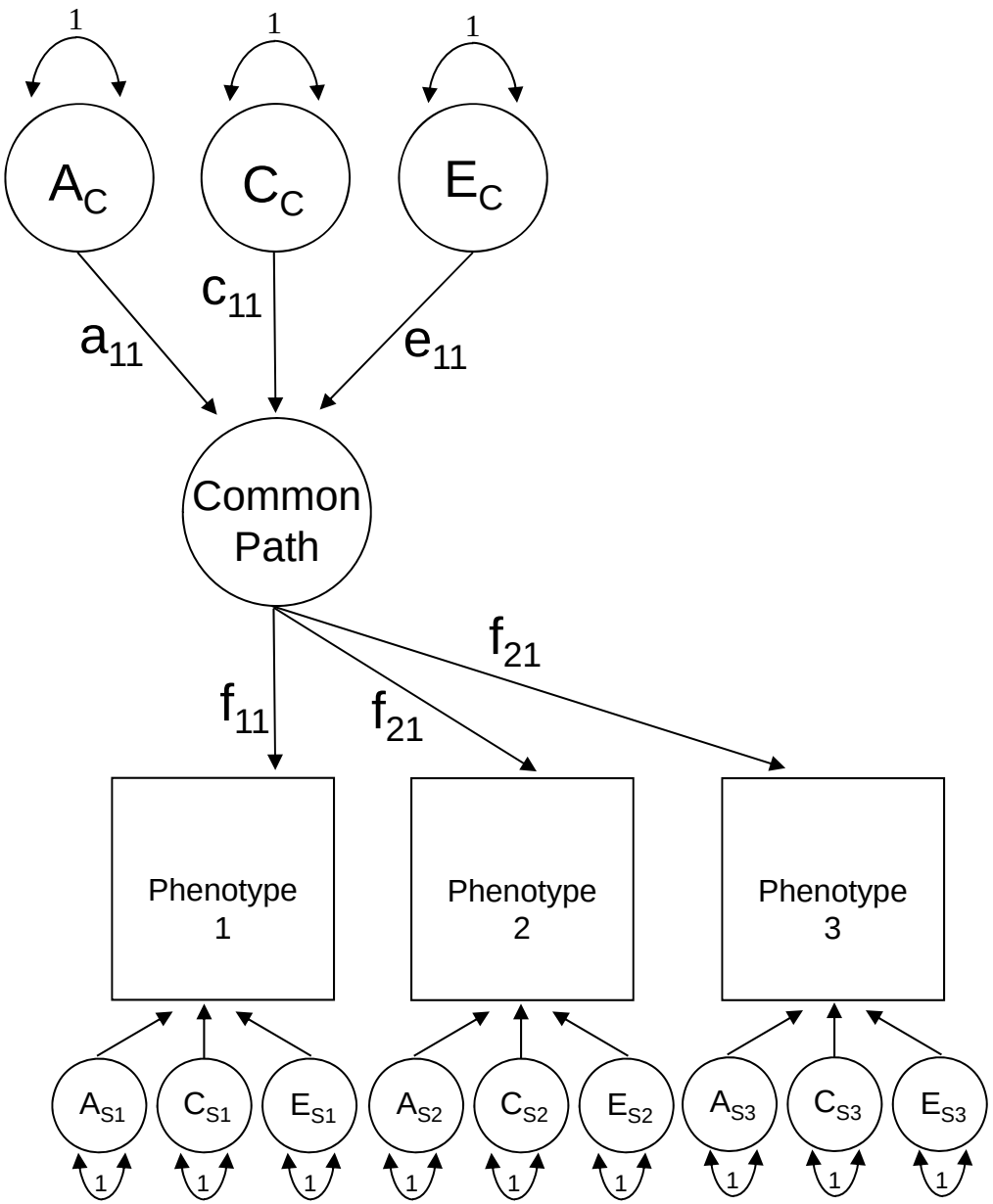
Recap common pathway model

Latent Growth Models

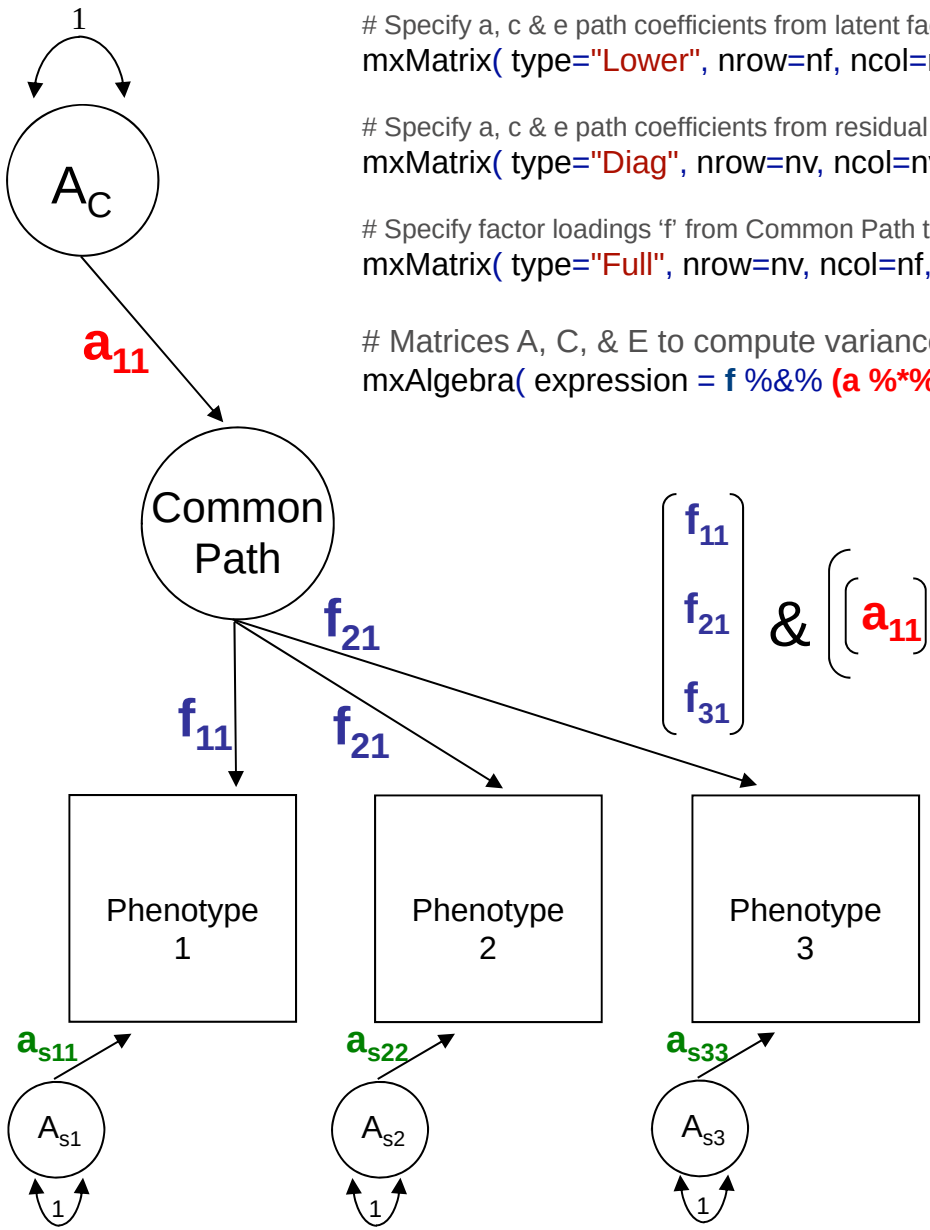
Simplex Models

Lindon’s caveat emptor

Common Pathway



Common Pathway: Genetic components of variance

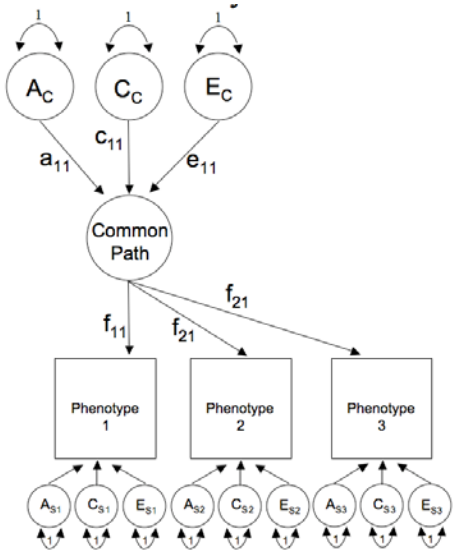



```
# Specify a, c & e path coefficients from latent factors A, C & E to Common Pathway
mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, name="a" ),

# Specify a, c & e path coefficients from residual latent factors As, Cs & Es to observed variables i.e. specifics
mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, name="as" ),

# Specify factor loadings 'f' from Common Path to observed variables
mxMatrix( type="Full", nrow=nv, ncol=nf, free=TRUE, values=15, name="f" ),

# Matrices A, C, & E to compute variance components
mxAlgebra( expression = f %&% (a %*% t(a)) + as %*% t(as), name="A" ),
```

$$\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \end{bmatrix} \& \left(\begin{bmatrix} a_{11} \end{bmatrix} X \begin{bmatrix} a_{11} \end{bmatrix}' \right) + \begin{bmatrix} a_{s11} & & \\ & a_{s22} & \\ & & a_{s33} \end{bmatrix} X \begin{bmatrix} a_{s11} & & \\ & a_{s22} & \\ & & a_{s33} \end{bmatrix}' = A$$

Common Pathway: Matrix algebra + variance components

```
# Matrices to store a, c, and e path coefficients for latent phenotype(s)
mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, name="a" ),
mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, name="c" ),
mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, name="e" ),

# Matrices to store a, c, and e path coefficients for specific factors
mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, name="as" ),
mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, name="cs" ),
mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, name="es" ),

# Matrix f for factor loadings from common pathway to observerd phenotypes
mxMatrix( type="Full", nrow=nv, ncol=nf, free=TRUE, values=15, name="f" ),

# Matrices A, C, & E to compute variance components
mxAlgebra( expression = f %&% (a %*% t(a)) + as %*% t(as), name="A" ),
mxAlgebra( expression = f %&% (c %*% t(c)) + cs %*% t(cs), name="C" ),
mxAlgebra( expression = f %&% (e %*% t(e)) + es %*% t(es), name="E" ),
```

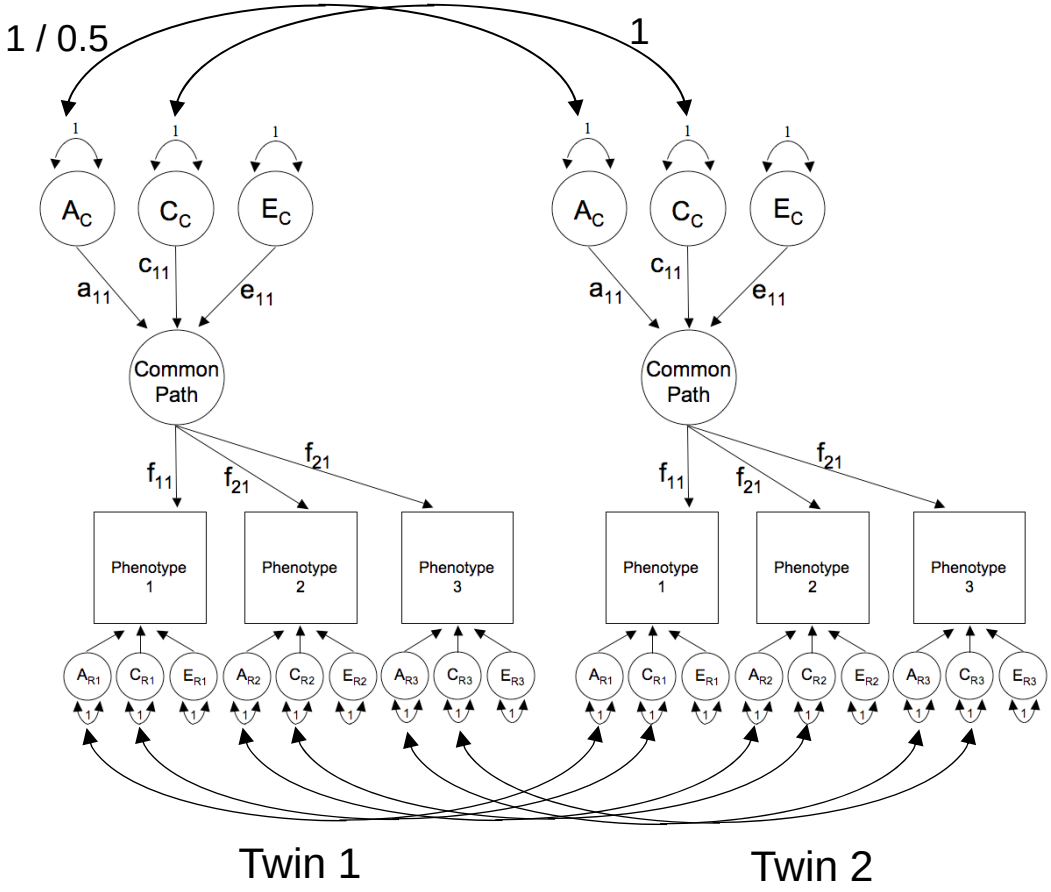


Within twin (co)variance

	T_1	T_2
T_1	A+C+E	
T_2		A+C+E

CP Model: Expected covariance

```
# Algebra for expected variance/covariance
covMZ     <- mxAlgebra( expression= rbind( cbind( A+C+E , A+C),
                                           cbind( A+C    ,  A+C+E)), name="expCovMZ" )

covMZ     <- mxAlgebra( expression= rbind( cbind(      A+C+E , 0.5%x%A+C),
                                           cbind(0.5%x%A+C , A+C+E)),    name="expCovMZ" )
```



MZ	T_1	T_2
T_1	A+C+E	A+C
T_2	A+C	A+C+E

DZ	T_1	T_2
T_1	A+C+E	$0.5_{@}$A+C
T_2	$0.5_{@}$A+C	A+C+E

Got longitudinal data?

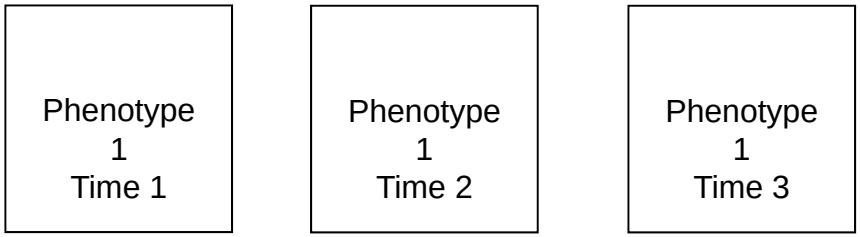
Do means & variance components change over time?

Are they stable?

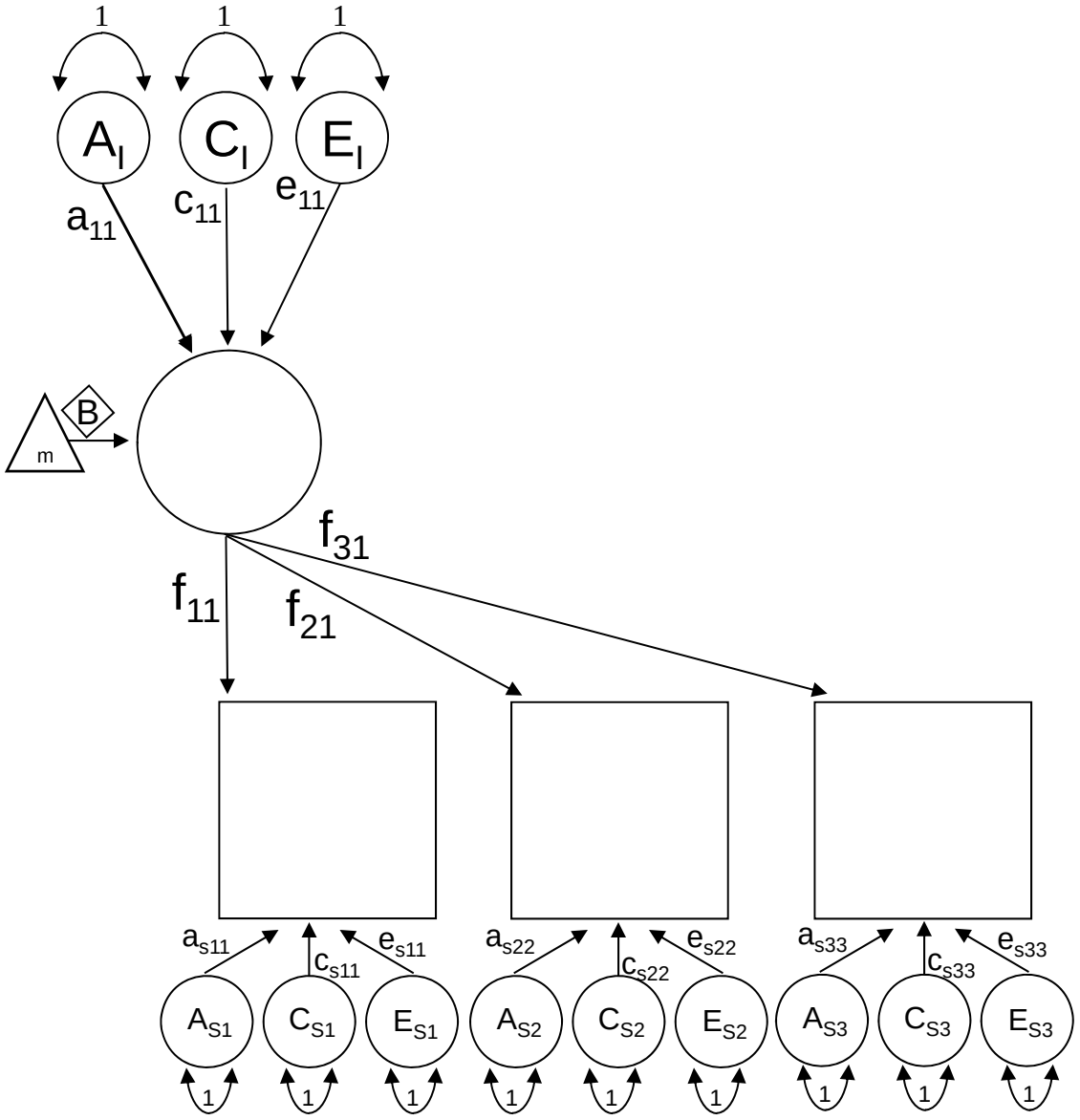
How to best explain change? Linear, non-linear?

One solution == latent growth model

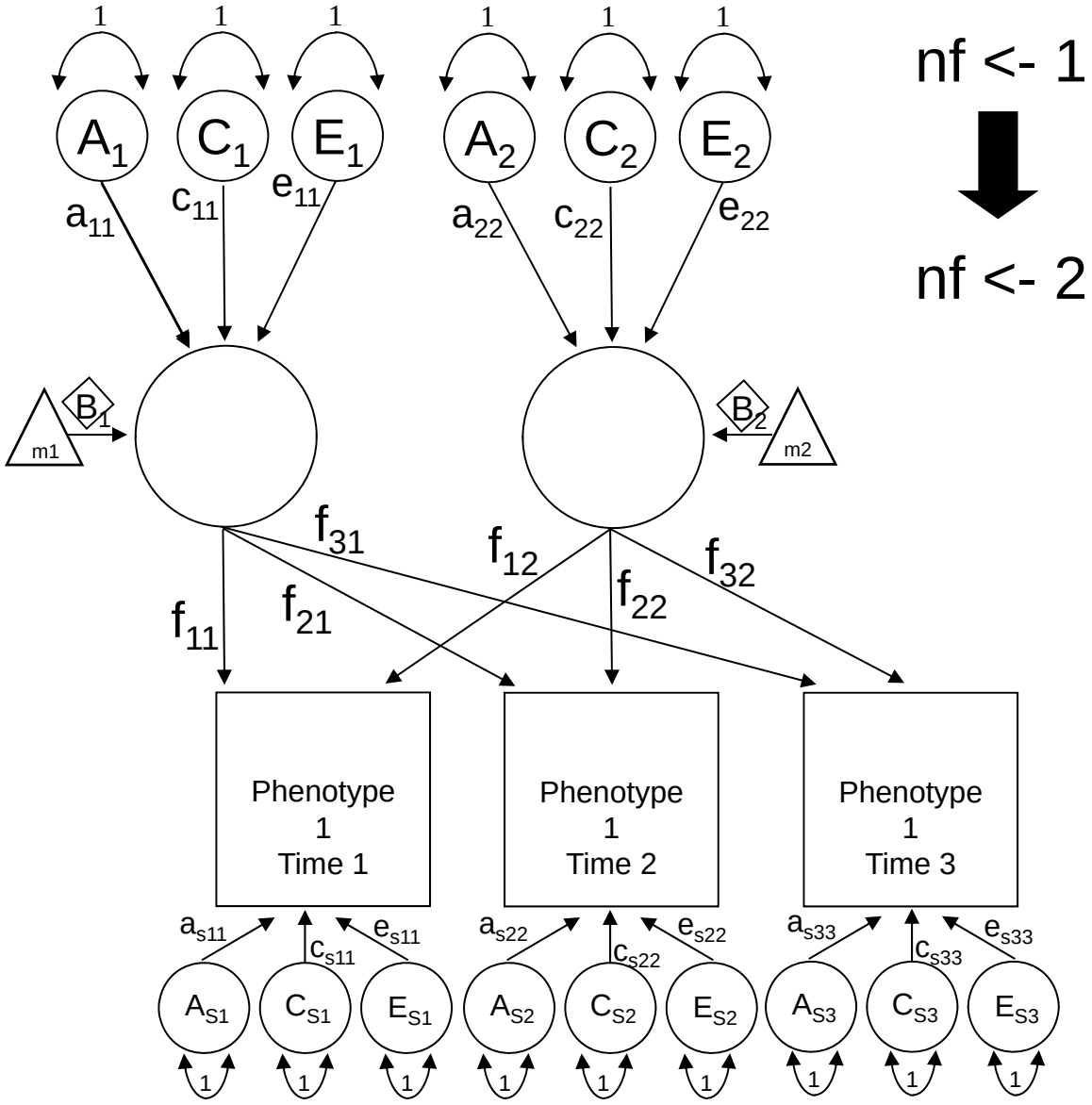
Build LGC from scratch



Common Pathway Model



Building a Latent Growth Curve Model



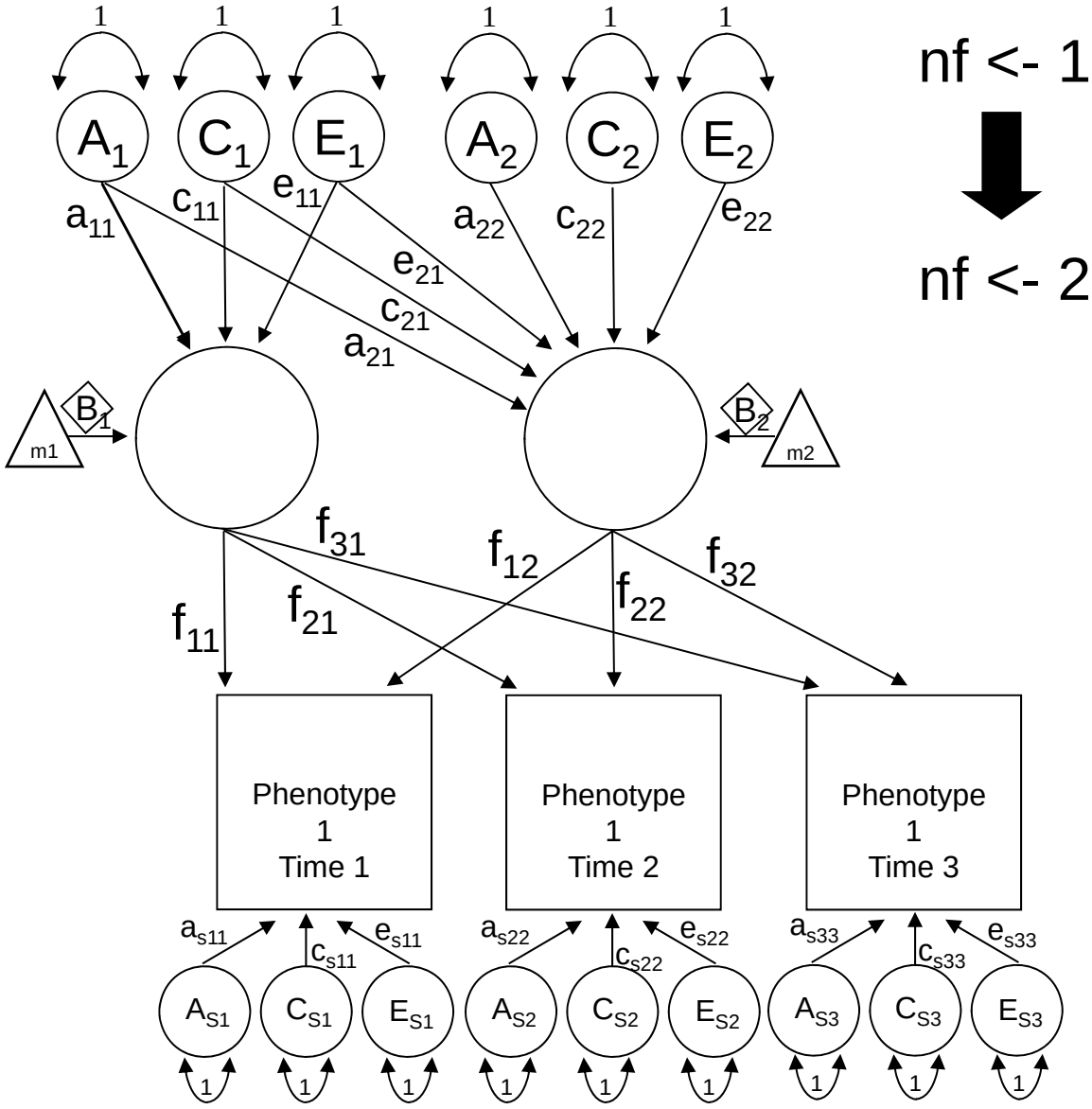
nf <- 1

nf <- 2

$$\begin{pmatrix} a_{11} \\ c_{11} \\ e_{11} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} & & \\ & c_{11} & \\ & & e_{11} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} & & \\ & c_{22} & \\ & & e_{22} \end{pmatrix}$$

$$\begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} \Rightarrow \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ f_{31} & f_{32} \end{pmatrix}$$

CP to Latent Growth Curve Model



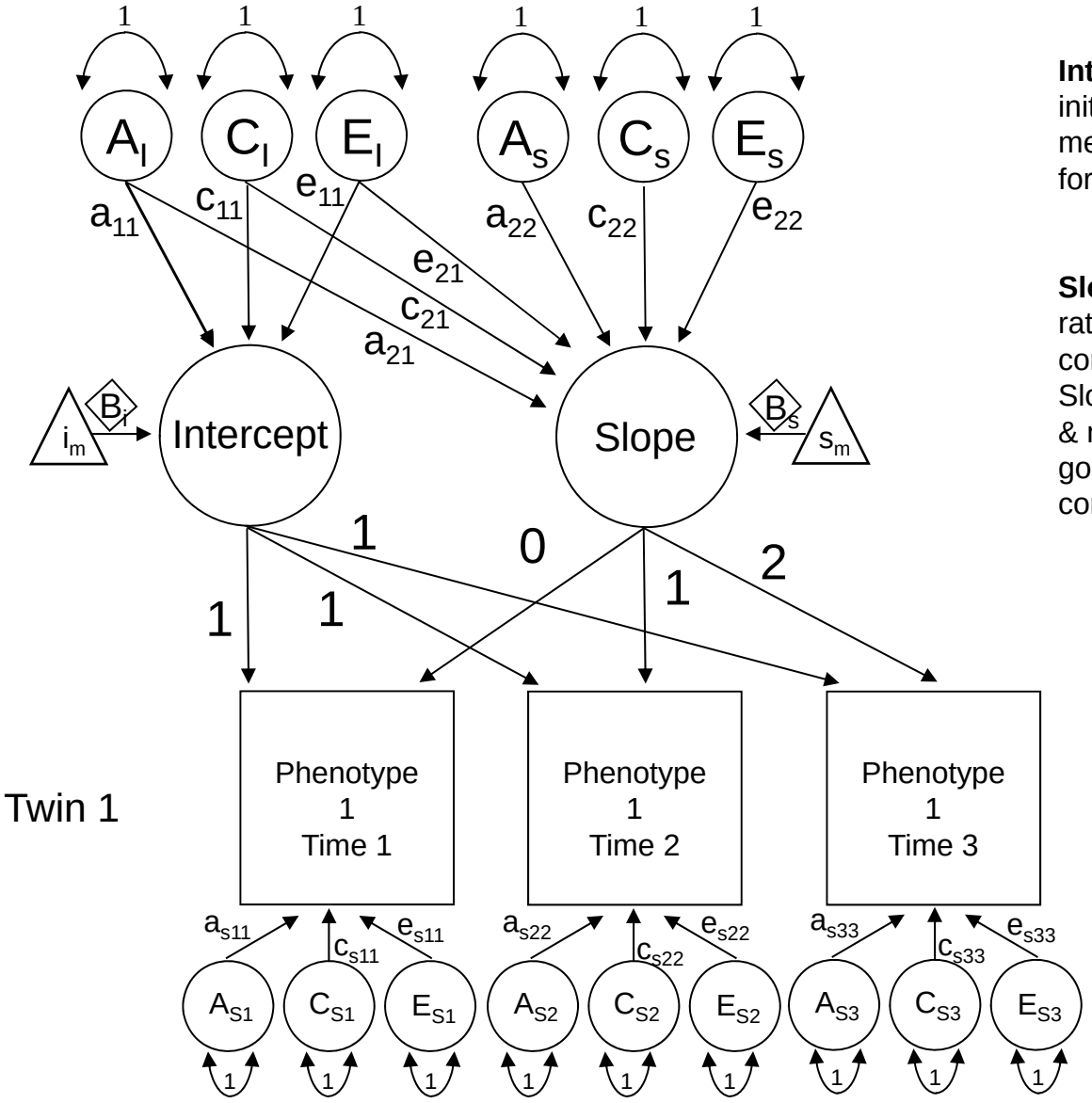
nf <- 1

nf <- 2

$$\begin{pmatrix} a_{11} \\ c_{11} \\ e_{11} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} & \\ a_{21} & a_{22} \\ c_{11} & \\ c_{21} & c_{22} \\ e_{11} & \\ e_{21} & e_{22} \end{pmatrix}$$

$$\begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} \Rightarrow \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ f_{31} & f_{32} \end{pmatrix}$$

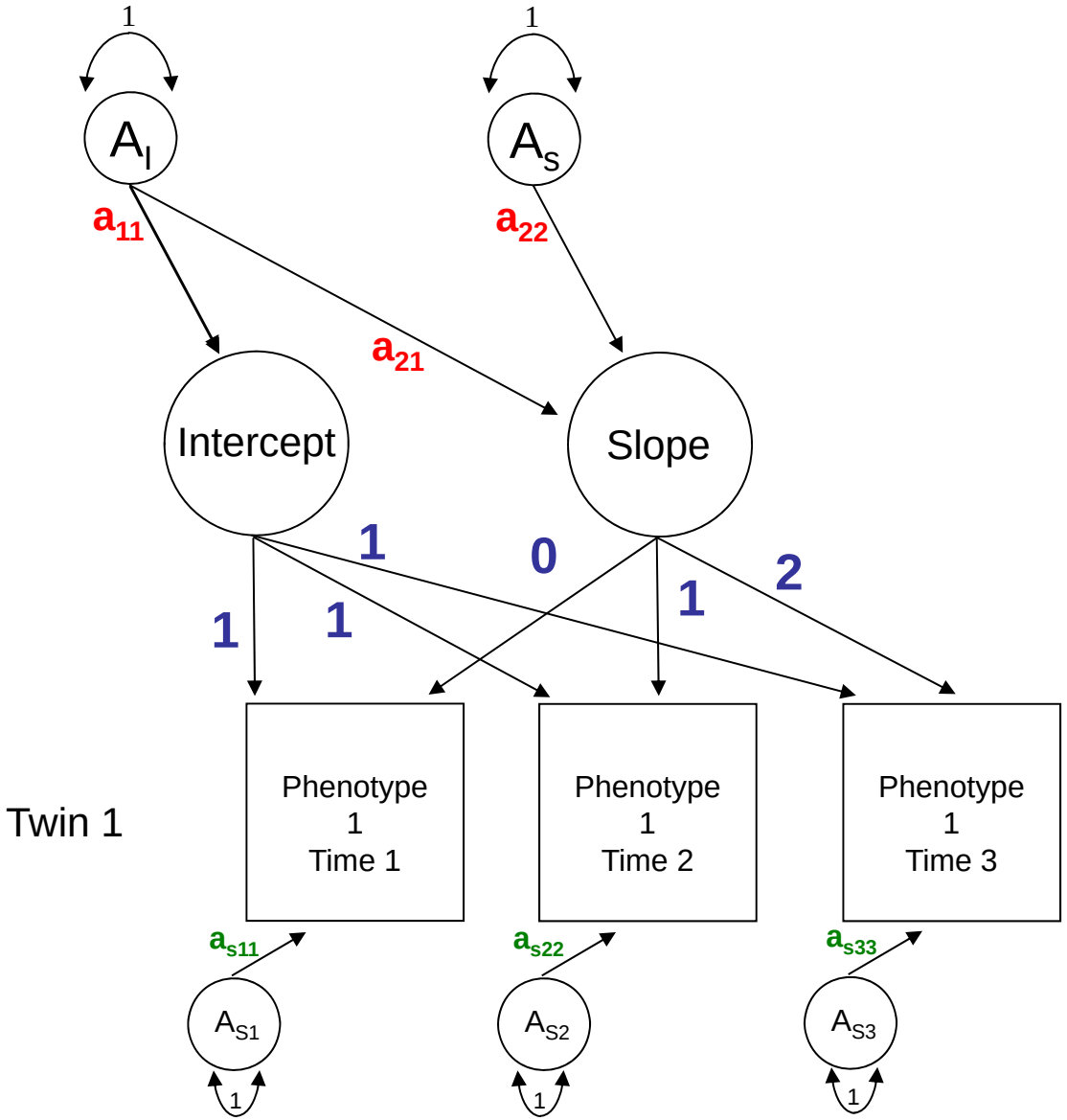
Latent Growth Curve Model



Intercept: Factor which explains initial variance components (and mean) for all measures. Accounts for the stability over time.

Slope: Factor which influences the rate of change in the variance components (and mean) over time. Slope(s) is (are) pre-defined: linear & non linear (quadratic, logistic, gompertz etc) hence factor loading constraints required.

LGC Model: Within twin genetic components of variance



$$\begin{pmatrix} a_{11} & \\ a_{21} & a_{22} \end{pmatrix}$$

Genetic pathway coefficients matrix

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

Factor loading matrix

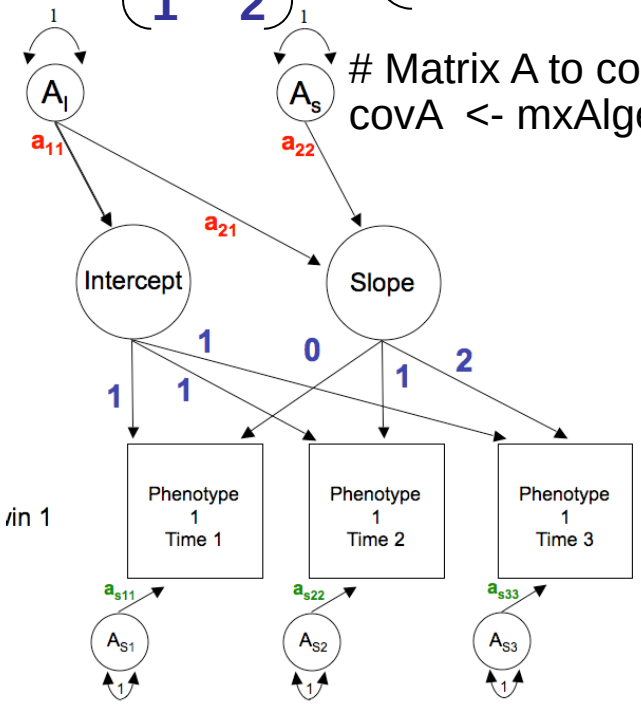
$$\begin{pmatrix} a_{S11} & & \\ & a_{S22} & \\ & & a_{S33} \end{pmatrix}$$

Residual genetic pathway coefficients matrix

LGC Model: Specifying variance components in R

```
# Matrix for a path coefficients from latent factors to Int' & Slope latent factors
pathAl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=AlLabs, name="al" )

# Matrix for a path coefficients from residuals to observed phenotypes
pathAs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, labels=AsLabs, name="as" )

# Factor loading matrix of Int & Slop on observed phenotypes
pathFl    <- mxMatrix( type="Full", nrow=nv, ncol=nf, free=FALSE, values=c(1,1,1,0,1,2), name="fl" )
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \& \left(\begin{pmatrix} a_{11} & \\ a_{21} & a_{22} \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{21} \\ & a_{22} \end{pmatrix}' \right) + \left(\begin{pmatrix} a_{S11} & & \\ & a_{S22} & \\ & & a_{S33} \end{pmatrix} \times \begin{pmatrix} a_{S11} & & \\ & a_{S22} & \\ & & a_{S33} \end{pmatrix}' \right) = A$$




```
# Matrix A to compute additive genetic variance components
covA  <- mxAlgebra( expression=fl %&% (al %*% t(al)) + as %*% t(as), name="A")
```

$$a_{11}^2 + a_{S11}^2 = A_{\text{var time 1}}$$

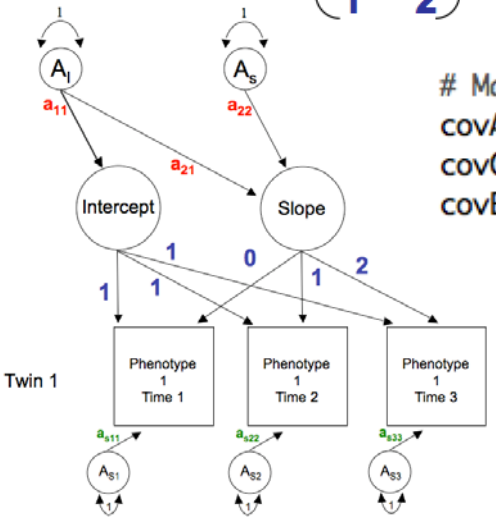
$$a_{11}^2 + a_{21}a_{11} + a_{21}a_{11} + a_{22}^2 + a_{S22}^2 = A_{\text{var time 2}}$$

$$a_{11}^2 + 2a_{21}a_{11} + 2a_{21}a_{11} + 2a_{22}^2 + a_{S33}^2 = A_{\text{var time 3}}$$

LGC Model: Specifying variance components in R

```
# Matrices ac, cc, and ec to store a, c, and e path coefficients from latent factors(s) to Int & Slope
pathAl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=AlLabs, name="al" )
pathCl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=ClLabs, name="cl" )
pathEl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=ElLabs, name="el" )

# Matrices as, cs, and es to store a, c, and e path coefficients for specific factors
pathAs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, labels=AsLabs, name="as" )
pathCs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, labels=CsLabs, name="cs" )
pathEs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=5, labels=EsLabs, name="es" )

# Matrix f for factor loadings on latent phenotype
pathFl    <- mxMatrix( type="Full", nrow=nv, ncol=nf, free=FALSE, values=c(1,1,1,0,1,2), name="fl" )
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \& \left(\begin{pmatrix} a_{11} & \\ a_{21} & a_{22} \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{21} \\ & a_{22} \end{pmatrix}' + \left(\begin{pmatrix} a_{S11} & & \\ & a_{S22} & \\ & & a_{S33} \end{pmatrix} \times \begin{pmatrix} a_{S11} & & \\ & a_{S22} & \\ & & a_{S33} \end{pmatrix}' \right) \right) = A$$




```
# Matrices A, C, and E compute variance components
covA      <- mxAlgebra( expression=fl %&% (al %*% t(al)) + as %*% t(as), name="A" )
covC      <- mxAlgebra( expression=fl %&% (cl %*% t(cl)) + cs %*% t(cs), name="C" )
covE      <- mxAlgebra( expression=fl %&% (el %*% t(el)) + es %*% t(es), name="E" )
```

LGC Model: Specifying covariance components in R

```
# Matrices A, C, and E compute variance components
covA      <- mxAlgebra( expression=fl %&% (al %*% t(al)) + as %*% t(as), name="A" )
covC      <- mxAlgebra( expression=fl %&% (cl %*% t(cl)) + cs %*% t(cs), name="C" )
covE      <- mxAlgebra( expression=fl %&% (el %*% t(el)) + es %*% t(es), name="E" )
```

MZ	T_1	T_2
T_1	A+C+E	A+C
T_2	A+C	A+C+E

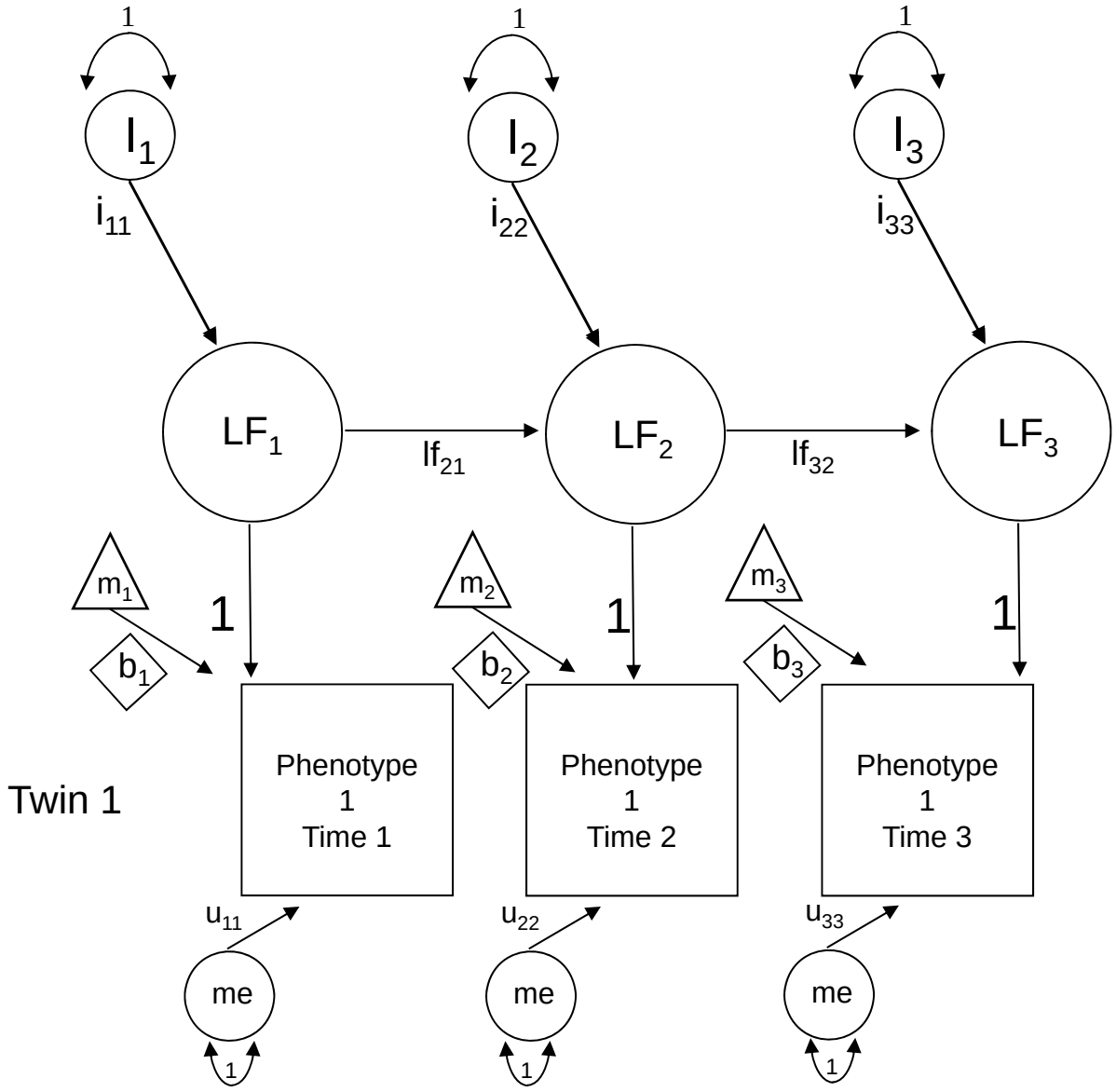
DZ	T_1	T_2
T_1	A+C+E	$0.5_{@}$A+C
T_2	$0.5_{@}$A+C	A+C+E

```
covMZ     <- mxAlgebra( expression= rbind( cbind(A+C+E , A+C),
                                           cbind(A+C   , A+C+E)),       name="expCovMZ" )
covDZ     <- mxAlgebra( expression= rbind( cbind(A+C+E     , 0.5%x%A+C),
                                           cbind(0.5%x%A+C , A+C+E)),   name="expCovDZ" )
```

LGC Model: 1.Continuous_Developmental_Twin_Matrix.R

```
# ------------------------------------------------------------------------------
# 2. Latent Growth Curve ACE Model + Sex effects for CONTINUOUS data
# ------------------------------------------------------------------------------

# Number of latent factors in model = INTECEPT + SLOPE
nf      <- 2

# Create Labels for Lower Triangular Matrices (fancy shorthand)
# Labels for a, c & e pathways from A, C & E latent factors to latent INTECEPT & SLOPE
AlLabs  <- paste("al", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")
ClLabs  <- paste("cl", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")
ElLabs  <- paste("el", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")

# Labels for factor loadings
FlLabs  <- paste("f",1:nv,1:nf,sep="_")

# Labels for a, c & e pathways from A, C & E residual to observed variables
AsLabs  <- paste("as",1:nv,1:nv,sep="_")
CsLabs  <- paste("cs",1:nv,1:nv,sep="_")
EsLabs  <- paste("es",1:nv,1:nv,sep="_")

# Prepare model = Specify all objects (matrices & matrix algebras)
# Matrices ac, cc, and ec to store a, c, and e path coefficients from latent factors(s) to Int & Slope
pathAl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=AlLabs, name="al" )
pathCl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=ClLabs, name="cl" )
pathEl    <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=.6, labels=ElLabs, name="el" )

# Matrices as, cs, and es to store a, c, and e path coefficients for specific factors
pathAs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, labels=AsLabs, name="as" )
pathCs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=4, labels=CsLabs, name="cs" )
pathEs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=5, labels=EsLabs, name="es" )

# Matrix f for factor loadings on latent phenotype
pathFl    <- mxMatrix( type="Full", nrow=nv, ncol=nf, free=FALSE, values=c(1,1,1,0,1,2), name="fl" )
#pathFL    <- mxMatrix( type="Full", nrow=nv, ncol=2, free=F, values=c(rep(1,nv), 0:nv-1), name="fl" ),  # general

# Matrices A, C, and E compute variance components
covA      <- mxAlgebra( expression=fl %&% (al %*% t(al)) + as %*% t(as), name="A" )
covC      <- mxAlgebra( expression=fl %&% (cl %*% t(cl)) + cs %*% t(cs), name="C" )
covE      <- mxAlgebra( expression=fl %&% (el %*% t(el)) + es %*% t(es), name="E" )

# Algebra to compute total variances and standard deviations (diagonal only)
covP      <- mxAlgebra( expression=A+C+E, name="V" )
matI      <- mxMatrix( type="Iden", nrow=nv, ncol=nv, name="I")
```

Simplex Models



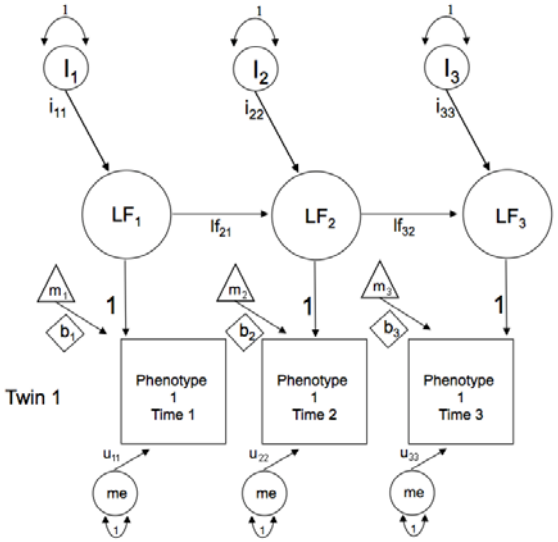
Simplex Models

Simplex designs model changes in the latent factor structure over time by fitting auto-regressive or Markovian chains

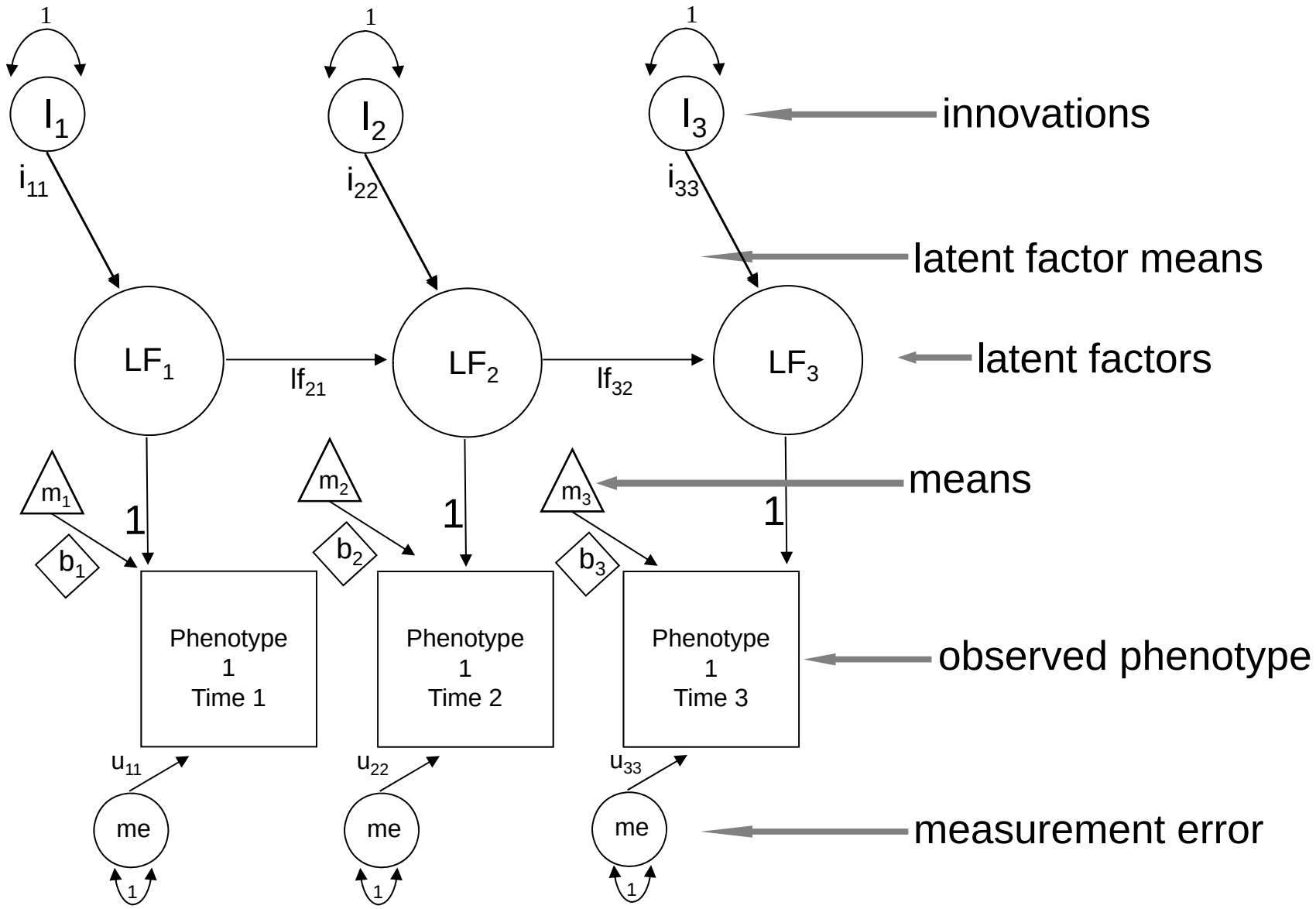
Determine how much variation in a trait is caused by stable & enduring effects versus transient effects unique to each time

The chief advantage of this model is the ability to partition environmental & genetic variation at each time point into:

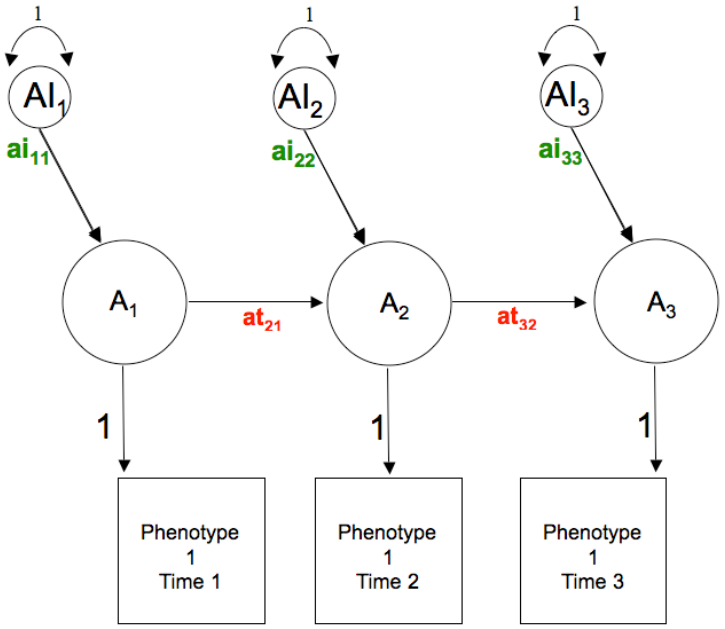
- genetic & environmental effects unique to each occasion
- genetic and environmental effects transmitted from previous time points



Simplex Models



Simplex Models: Within twin genetic variance



$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

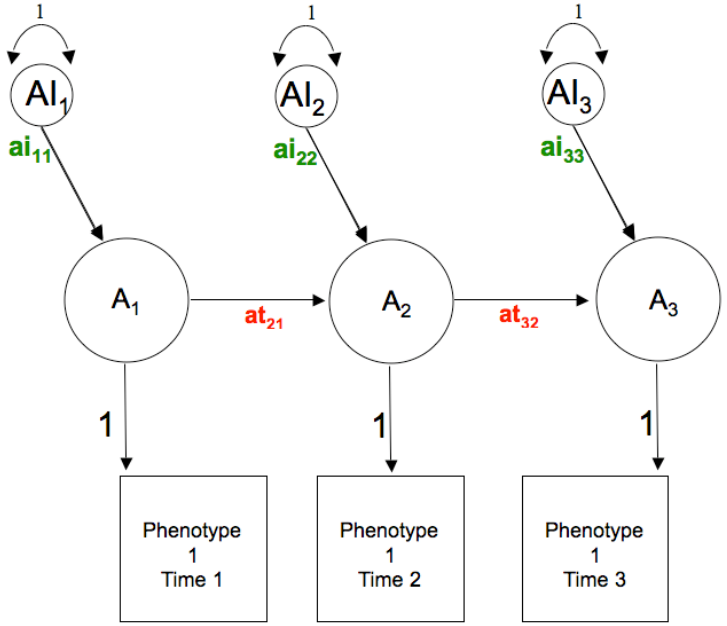
$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{pmatrix} ? & & \\ ? & ? & \\ ? & ? & ? \end{pmatrix} \end{array}$$

Transmission pathways

$$\begin{pmatrix} ai_{11} & & \\ & ai_{22} & \\ & & ai_{33} \end{pmatrix}$$

Innovation pathways

Simplex Models: Genetic variance



$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{pmatrix} 0 & & \\ at_{21} & 0 & \\ 0 & at_{32} & 0 \end{pmatrix} \end{array}$$

Transmission pathways

$$\begin{pmatrix} ai_{11} & & \\ & ai_{22} & \\ & & ai_{33} \end{pmatrix}$$

Innovation pathways

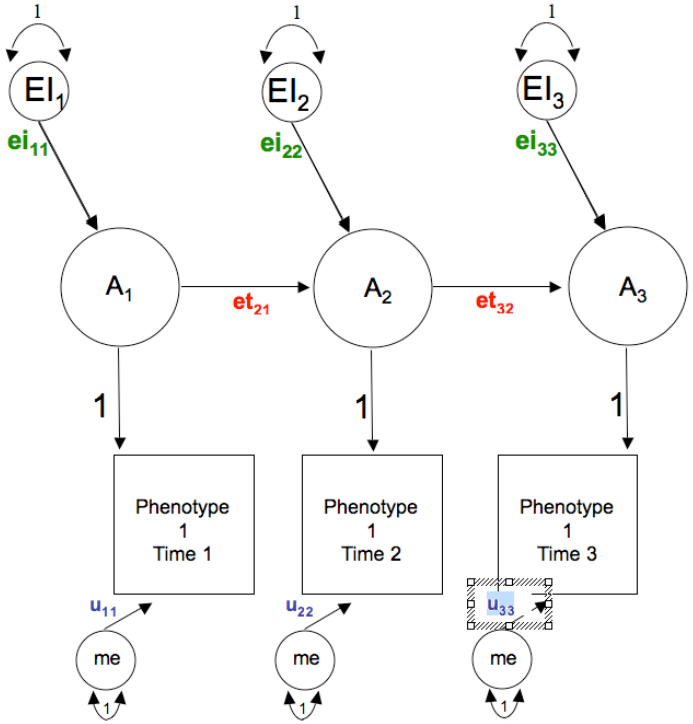
$$\left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & & \\ at_{21} & 0 & \\ 0 & at_{32} & 0 \end{pmatrix} \right)^{-1} \& \left(\begin{pmatrix} ai_{11} & & \\ & ai_{22} & \\ & & ai_{33} \end{pmatrix} * \begin{pmatrix} ai_{11} & & \\ & ai_{22} & \\ & & ai_{33} \end{pmatrix}' \right) = A$$

```
matI    <- mxMatrix( type="Iden", nrow=nv, ncol=nv, name="I")

pathAt  <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=tFree, values=ValsA, labels=AtLabs, name="at" )

pathAi  <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=iVals, labels=AiLabs, name="ai" )

covA    <- mxAlgebra( expression=solve( I - at ) %&% ( ai %*% t(ai)), name="A" )
```

Simplex Models: E variance + measurement error

```
matI     <- mxMatrix( type="Iden", nrow=nv, ncol=nv, name="I")
pathEt   <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=tFree, values=tValsE, labels=EtLabs, name="et" )
pathEi   <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=iVals, labels=EiLabs, name="ei" )
pathMe   <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, labels=c("u","u","u"), values=5, name="me" )

covE     <- mxAlgebra( expression=solve( I-et ) %&% (ei %*% t(ei))+ (me %*% t(me)), name="E" )
```

$$\left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & & \\ et_{21} & 0 & \\ 0 & et_{32} & 0 \end{pmatrix}\right)^{\sim} \& \begin{pmatrix} ei_{11} & & \\ & ei_{22} & \\ & & ei_{33} \end{pmatrix} * \begin{pmatrix} ei_{11} & & \\ & ei_{22} & \\ & & ei_{33} \end{pmatrix}' + \begin{pmatrix} u_{11} & & \\ & u_{22} & \\ & & u_{33} \end{pmatrix} * \begin{pmatrix} u_{11} & & \\ & u_{22} & \\ & & u_{33} \end{pmatrix}' = E$$



LGC Model: Specifying covariance components in R

```
# Matrices A, C, & E to compute variance components
covA      <- mxAlgebra( expression=solve(I-at) %&% (ai %*% t(ai)), name="A" )
covC      <- mxAlgebra( expression=solve(I-ct) %&% (ci %*% t(ci)), name="C" )
covE      <- mxAlgebra( expression=solve(I-et) %&% (ei %*% t(ei))+ (me %*% t(me)), name="E" )
```

MZ	T_1	T_2
T_1	A+C+E	A+C
T_2	A+C	A+C+E

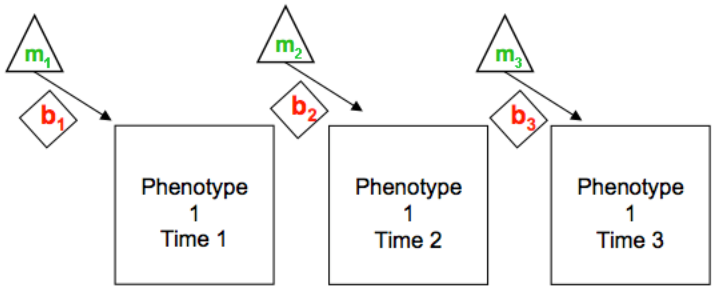
DZ	T_1	T_2
T_1	A+C+E	$0.5_{@}$A+C
T_2	$0.5_{@}$A+C	A+C+E

```
covMZ     <- mxAlgebra( expression= rbind( cbind(A+C+E , A+C),
                                           cbind(A+C   , A+C+E)),       name="expCovMZ" )
covDZ     <- mxAlgebra( expression= rbind( cbind(A+C+E     , 0.5%x%A+C),
                                           cbind(0.5%x%A+C , A+C+E)),   name="expCovDZ" )
```

Simplex Models: Means & sex in R

```
meanG     <- mxMatrix( type="Full", nrow=1, ncol=3, free=TRUE, labels=c("m1","m2","m3"), values=10.1, name="Mean" )
```

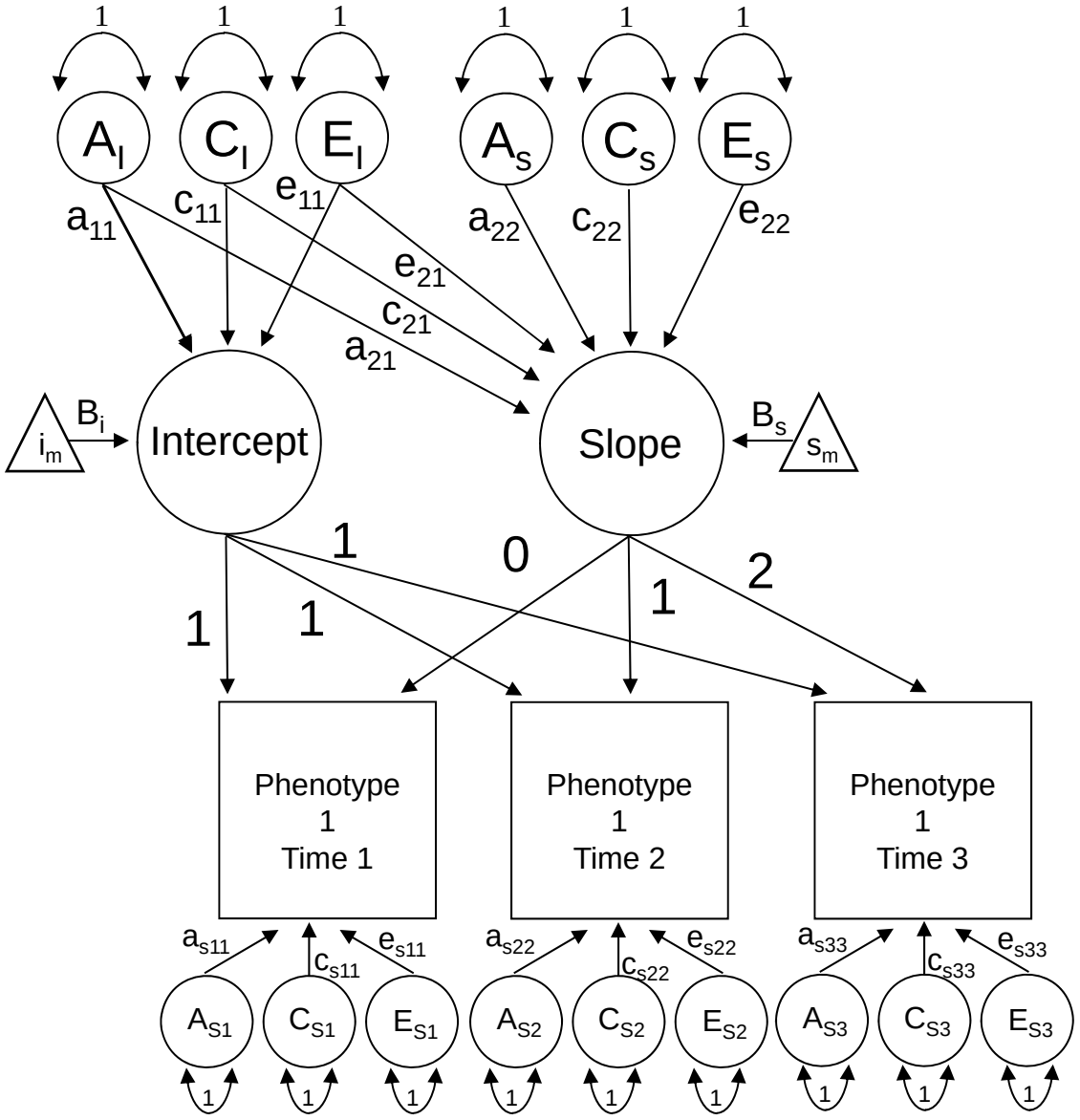
Time 1	Time 2	Time 3
m1	**m2**	**m3**



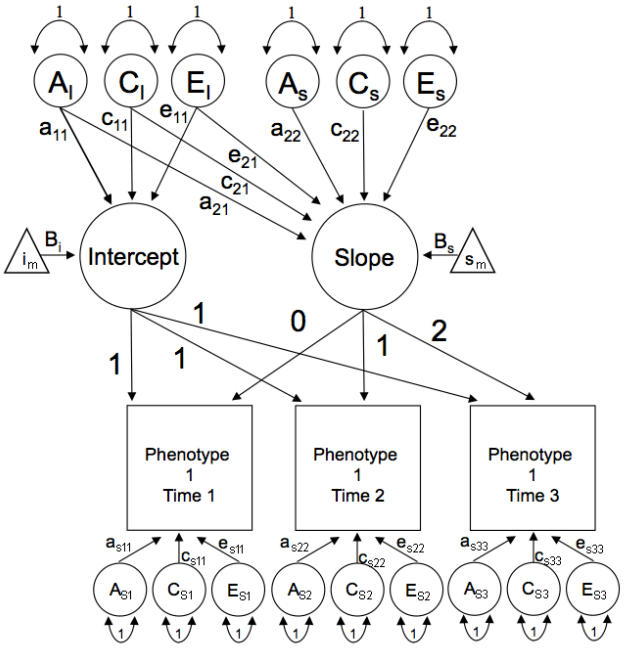
Simplex Model: 1.Continuous_Developmental_Twin_Matrix.R

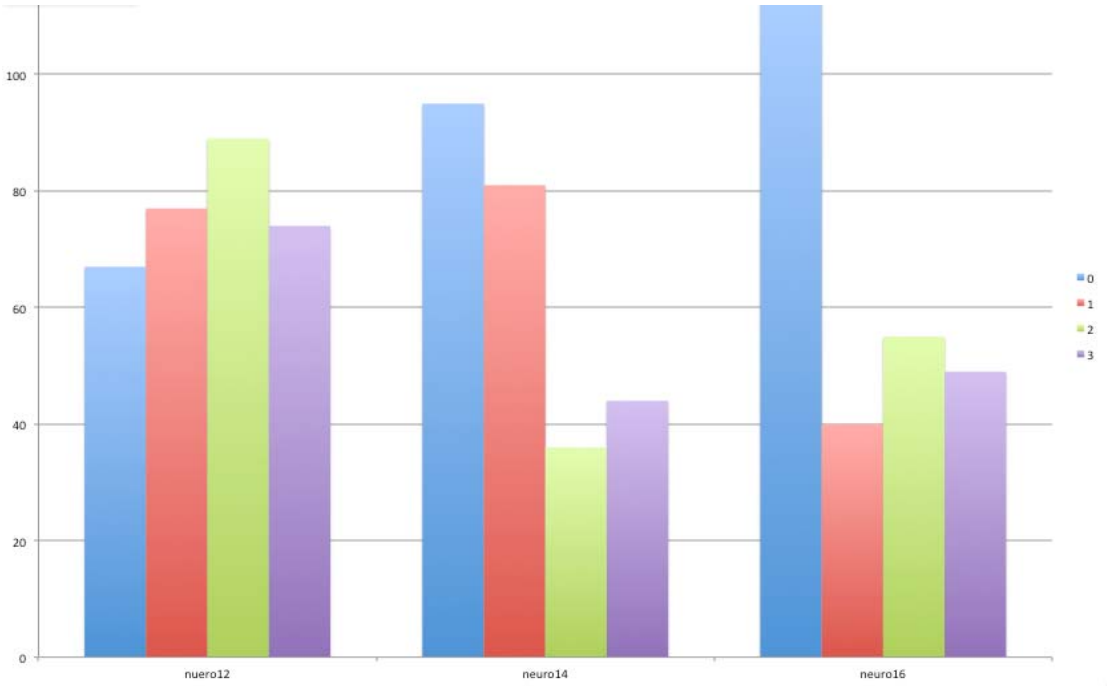
```
# ------------------------------------------------------------------------------
# 3. Simplex ACE Model - for continuous data
# ------------------------------------------------------------------------------

# Create Labels for SubDiagonal and Diagonal Matrices
tFree     <- c(F,T,F,F,T,F)         # Specify free vs fixed transmission elements
tValsA     <- c(0,1.5,0,0,1.5,0)        # Start values for free transmission elements
tValsC     <- c(0,1.2,0,0,1.3,0)        # Start values for free transmission elements
tValsE     <- c(0,0.5,0,0,0.8,0)        # Start values for free transmission elements

iFree     <- c(T,T,T)           # Specify free vs fixed transmission elements
iVals     <- c(2,2,2)           # Start values for free transmission elements


AtLabs    <- paste("at", do.call(c, sapply(seq(1, nv), function(x){ paste(x:nv, x,sep="_") })), sep="_")
CtLabs    <- paste("ct", do.call(c, sapply(seq(1, nv), function(x){ paste(x:nv, x,sep="_") })), sep="_")
EtLabs    <- paste("et", do.call(c, sapply(seq(1, nv), function(x){ paste(x:nv, x,sep="_") })), sep="_")
                                      # Labels (fancy) for transmissions

AiLabs    <- paste("ai",1:nv,1:nv,sep="_")  # Labels for A innovations
CiLabs    <- paste("ci",1:nv,1:nv,sep="_")  # Labels for C innovations
EiLabs    <- paste("ei",1:nv,1:nv,sep="_")  # Labels for E innovations

# Prepare model = Specify all objects (matrices & matrix algebras)
# Create matrices at, ct, & et to store a, c, & e path coefficients for transmissions
pathAt    <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=tFree, values=tValsA, labels=AtLabs, name="at" )
pathCt    <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=tFree, values=tValsC, labels=CtLabs, name="ct" )
pathEt    <- mxMatrix( type="Lower", nrow=nv, ncol=nv, free=tFree, values=tValsE, labels=EtLabs, name="et" )

# Created matrices ai, ci, & ei to store a, c, & e path coefficients for innovations
pathAi    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=iVals, labels=AiLabs, name="ai" )
pathCi    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=iVals, labels=CiLabs, name="ci" )
pathEi    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=iVals, labels=EiLabs, name="ei" )

# Matrix me for measurement error
pathMe    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, labels=c("u","u","u"), values=5, name="me" )

# Matrices A, C, & E to compute variance components
covA      <- mxAlgebra( expression=solve(I-at) %&% (ai %*% t(ai)), name="A" )
covC      <- mxAlgebra( expression=solve(I-ct) %&% (ci %*% t(ci)), name="C" )
covE      <- mxAlgebra( expression=solve(I-et) %&% (ei %*% t(ei))+ (me %*% t(me)), name="E" )

# Algebra to compute total variance and standard deviations (diagonal only)
```

Ordinal Data Latent Growth Curve Modeling



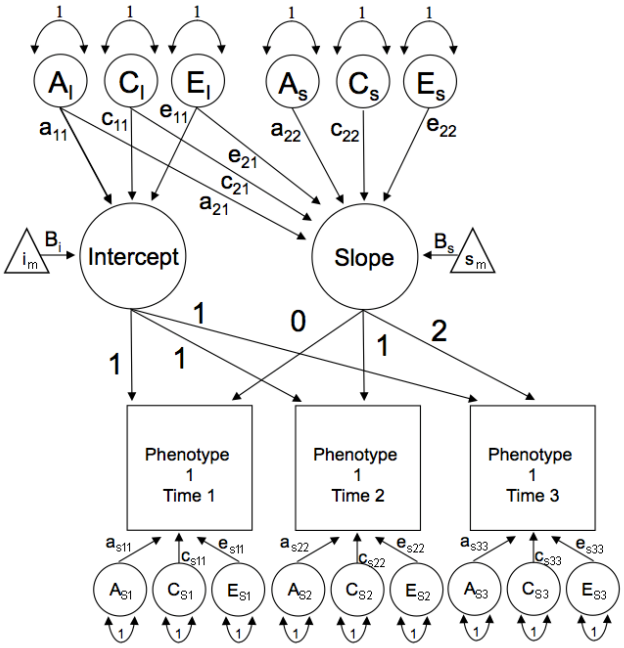
Ordinal Data Latent Growth Curve Modeling






```
# Matrix & Algebra for expected MEANS & THRESHOLDS
meanG     <- mxMatrix( type="Zero", nrow=1, ncol=nv, name="Mean" )
meanT     <- mxAlgebra( expression= cbind(Mean,Mean), name="expMean" )

threG     <- mxMatrix( type="Full", nrow=nth, ncol=nv, free=TRUE, name="Thre" )
Inc       <- mxMatrix( type="Lower", nrow=nth, ncol=nth, free=FALSE, values=1, name="Inc" )
threT     <- mxAlgebra( expression= cbind(Inc %*% Thre,Inc %*% Thre), name="expThre" )
# Standard ordinal data approach: Specify mean vector of zeroes & estimate thresholds
```

Ordinal Data Latent Growth Curve Modeling



Psychological Methods
2004, Vol. 9, No. 3, 301–333

1082-989X/04/$12.00 DOI: 10.1037/1082-989X.9.3.301

Squeezing Interval Change From Ordinal Panel Data: Latent Growth Curves With Ordinal Outcomes

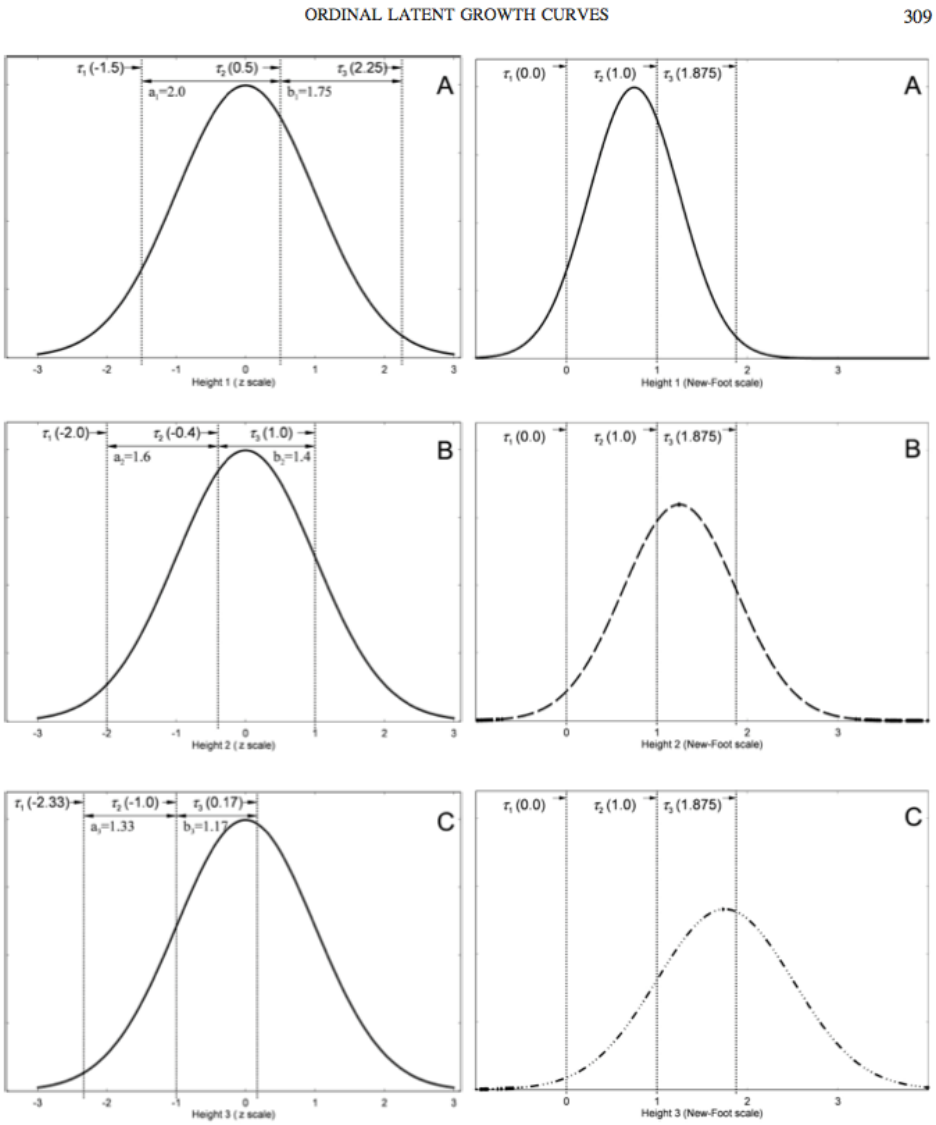
Paras D. Mehta
University of Illinois at Chicago

Michael C. Neale
Virginia Commonwealth University

Brian R. Flay
University of Illinois at Chicago

A didactic on latent growth curve modeling for ordinal outcomes is presented. The conceptual aspects of modeling growth with ordinal variables and the notion of threshold invariance are illustrated graphically using a hypothetical example. The ordinal growth model is described in terms of 3 nested models: (a) multivariate normality of the underlying continuous latent variables (y_t) and its relationship with the observed ordinal response pattern (Y_t), (b) threshold invariance over time, and (c) growth model for the continuous latent variable on a common scale. Algebraic implications of the model restrictions are derived, and practical aspects of fitting ordinal growth models are discussed with the help of an empirical example and Mx script (M. C. Neale, S. M. Boker, G. Xie, & H. H. Maes, 1999). The necessary conditions for the identification of growth models with ordinal data and the methodological implications of the model of threshold invariance are discussed.

Ordinal Data Latent Growth Curve Modeling

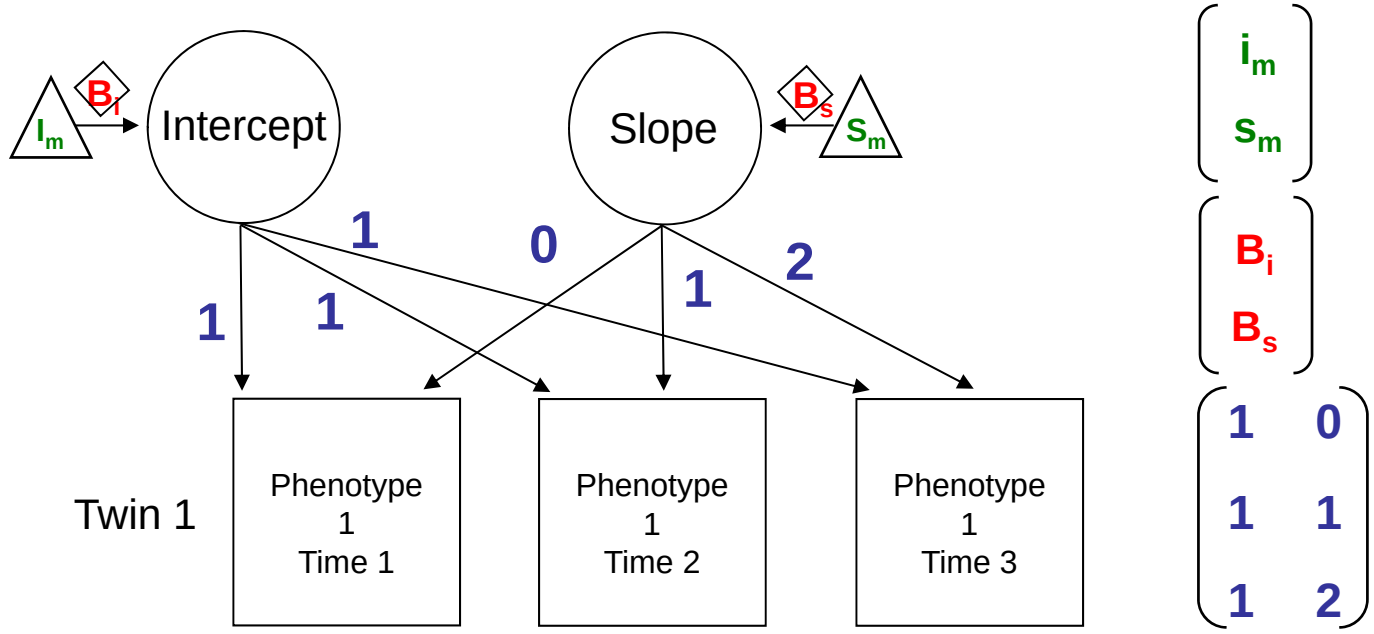



```
# Fix means at zero & allow thresholds to vary
                vs
# Fix the thresholds & allow means to vary
```

$$\boldsymbol{\tau}_{ct}^{nf} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ \boldsymbol{\tau}_3^{nf} & \boldsymbol{\tau}_3^{nf} & \boldsymbol{\tau}_3^{nf} \end{bmatrix}. \qquad (16)$$

LGC Model: Estimating means (& sex) in R

Means on observed phenotypes versus means on Intercept & Slope?



$$\begin{pmatrix} i_m \\ s_m \end{pmatrix} \quad \begin{pmatrix} B_i \\ B_s \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

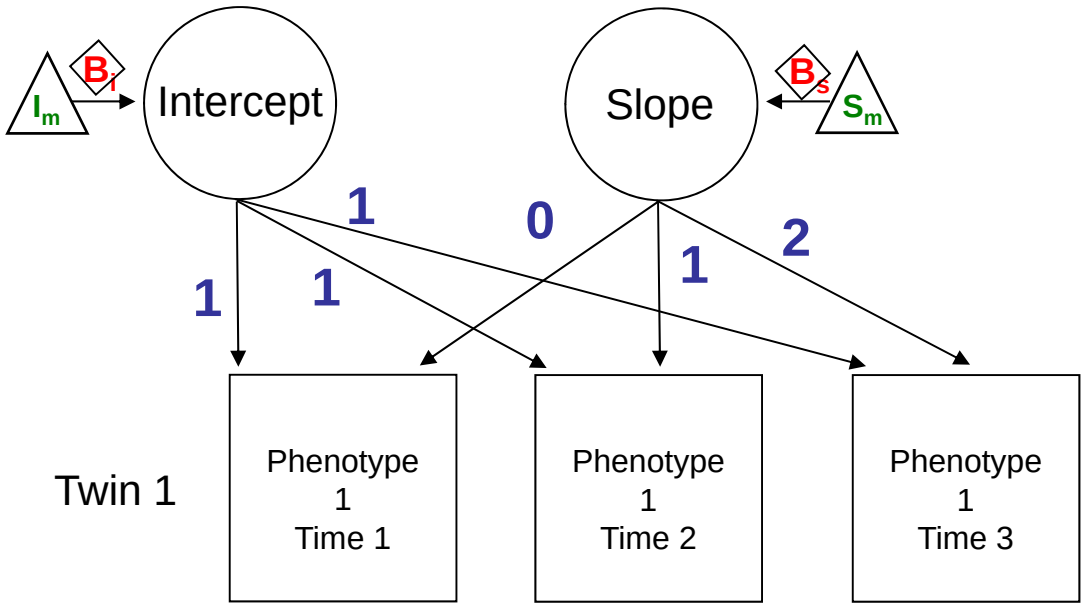
```
MeansIS   <- mxMatrix( type="Full", nrow=2, ncol=1, free=T, labels=c("Im","Sm"), values=c(5,2), name="LMeans" )

pathB     <- mxMatrix( type="Full", nrow=2, ncol=1, free=T, values=c(5,2), labels=c("Bi","Bs"), name="Beta" )

pathFl    <- mxMatrix( type="Full", nrow=nv, ncol=nf, free=FALSE, values=c(1,1,1,0,1,2), labels=FlLabs,name="fl" )
```

LGC Model: Means Algebra

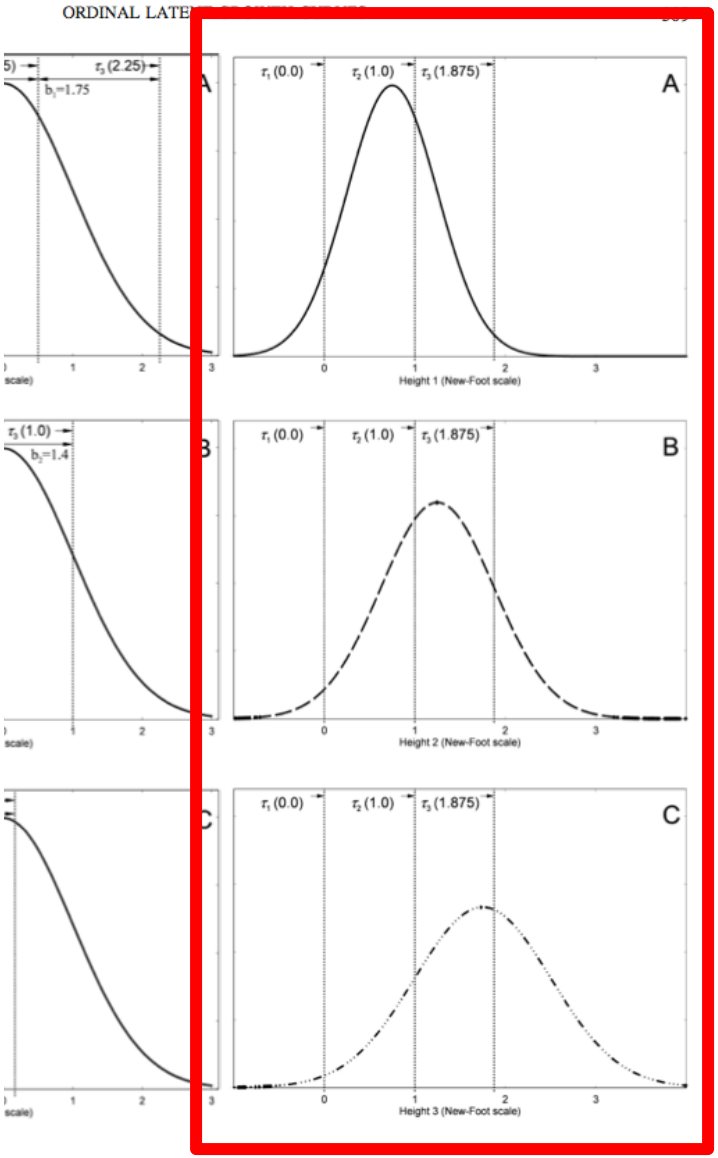
$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \times \left(\begin{pmatrix} i_m \\ s_m \end{pmatrix} + \begin{pmatrix} Sex_{T1} \end{pmatrix} @ \begin{pmatrix} B_i \\ B_s \end{pmatrix} \right) = \text{Expected means for Twin 1}$$

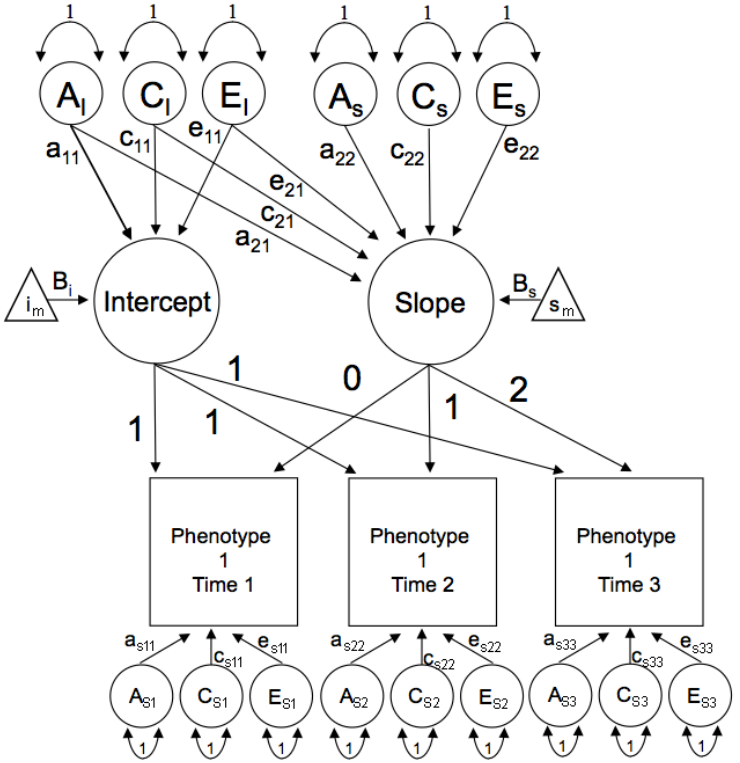
$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \times \begin{pmatrix} i_m + B_i Sex_{T1} \\ s_m + B_s Sex_{T1} \end{pmatrix} = \begin{pmatrix} (i_m + B_i Sex_{T1}) \\ (i_m + B_i Sex_{T1}) + 1(s_m + B_s Sex_{T1}) \\ (i_m + B_i Sex_{T1}) + 2(s_m + B_s Sex_{T1}) \end{pmatrix} \begin{matrix} \text{Time 1} \\ \text{Time 2} \\ \text{Time 3} \end{matrix}$$

Means1 <- mxAlgebra(expression= (t((**fl** %*% (**LMeans** - Sex1 %x% **Beta**)))), name="Mean1")



LGC: Threshold Invariance or Fixed Thresholds





$$\tau_{ct}^{nf} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ \tau_3^{nf} & \tau_3^{nf} & \tau_3^{nf} \end{bmatrix}.$$

LGC: Threshold Invariance or Fixed Thresholds

$$\tau_{ct}^{nf} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ \tau_3^{nf} & \tau_3^{nf} & \tau_3^{nf} \end{bmatrix}.$$

```
# Specify invariant (fixed) thresholds for ordinal data: 1st & 2nd thresholds are fixed, 3rd threshold is free
# NB: These constraints indentify the mean & standard deviation of the Intercept & Slope
thresh1   <- mxMatrix( type="Full", nrow=1, ncol=nv, free=F, labels=c("th1","th1","th1"), values=0.0, name="t1" )
thresh2   <- mxMatrix( type="Full", nrow=1, ncol=nv, free=F, labels=c("th2","th2","th2"), values=1.0, name="t2" )
thresh3   <- mxMatrix( type="Full", nrow=1, ncol=nv, free=T, labels=c("th3_v1","th3_v2","th3_v3"), values=5, name="t3" )
thresh123 <- mxAlgebra( expression=( rbind(t1,t2,t3) ), name="Th" )
low       <- mxMatrix( type="Lower", nrow=nth, ncol=nth, free=F, values=1, name="Inc" )
eThreshs    <- mxAlgebra( expression= cbind( Inc %*% Th, Inc %*% Th ), name="expThre")
```

LGC Model: 2.Ordinal_Template_Developmental_Twin_Matrix.R

```
# ------------------------------------------------------------------------------
# 3. Latent Growth Curve ACE Model + Sex effects
# ------------------------------------------------------------------------------

# Assumes threshold invariance + estimates latent factor means to squeeze out mean & variance information
# Mehta, P. D., Neale, M. C., & Flay, B. R. (2004). Squeezing interval change from ordinal panel data: Latent growth cur
 outcomes. Psychological Methods, 9(3), 301-333.

# SPECIFY START VALUES & create labels used in the script
# Labels for a, c & e pathways from A, C & E factors to INTECEPT & SLOPE factors
nf       <- 2       # Number of latent factors in model = INTECEPT + SLOPE
AlLabs   <- paste("al", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")
ClLabs   <- paste("cl", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")
ElLabs   <- paste("el", do.call(c, sapply(seq(1, nf), function(x){ paste(x:nf, x,sep="") })), sep="")

# Labels for LGC latent factor loadings
FlLabs   <- c(paste("f1",1:nv,sep=""),paste("f2",1:nv,sep=""))

# Labels for A, C & E specifics
AsLabs   <- paste("as",1:nv,1:nv,sep="_")
CsLabs   <- paste("cs",1:nv,1:nv,sep="_")
EsLabs   <- paste("es",1:nv,1:nv,sep="_")

# Lables for thresholds
ths      <- paste("thr",1:nv,sep="_")

#ISstartA     <- c(9,4,2)          # Start values for INTERCEP & SLOPE factor loadings
#ISstartE     <- c(4,4,2)          # Start values for INTERCEP & SLOPE factor loadings
#RESstart     <- c(5,5,5)          # Start values for INTERCEP & SLOPE factor loadings

# PREPARE MODEL = Specify all objects (matrices & matrix algebras)
# Matrices ac, cc, and ec to store a, c, and e path coefficients from latent factors(s) to Int & Slope
pathAl   <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=3, labels=AlLabs, name="al" )
pathCl   <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=3, labels=ClLabs, name="cl" )
pathEl   <- mxMatrix( type="Lower", nrow=nf, ncol=nf, free=TRUE, values=3, labels=ElLabs, name="el" )
# NB: No constraint on the lower & upper bounds

# Matrix f for fixed factor loadings from Intercept & Slope to observed variables
pathFl    <- mxMatrix( type="Full", nrow=nv, ncol=nf, free=FALSE, values=c(1,1,1,0,1,2), labels=FlLabs,name="fl" )

# Matrices as, cs, and es to store a, c, and e path coefficients for specific factors (residuals)
pathAs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=3, labels=AsLabs, name="as" )
pathCs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=3, labels=CsLabs, name="cs" )
pathEs    <- mxMatrix( type="Diag", nrow=nv, ncol=nv, free=TRUE, values=3, labels=EsLabs, name="es" )

# Matrices generated to hold A, C, and E computed Variance Components
covA      <- mxAlgebra( expression=fl %&% (al %*% t(al)) + as %*% t(as), name="A" )
covC      <- mxAlgebra( expression=fl %&% (cl %*% t(cl)) + cs %*% t(cs), name="C" )
covE      <- mxAlgebra( expression=fl %&% (el %*% t(el)) + es %*% t(es), name="E" )
```