

# **strum** package - examples

Yeunjoo E. Song, Catherine M. Stein, Nathan J. Morris

May 11, 2015

This document contains the whole analysis process for the first three example models from the introduction document. Note that the input data used for these examples are not necessarily simulated to give a meaningful result for each analysis.

```
> library(strum)
```

## 1 Genetic association analysis

This is an example of a typical genetic association analysis model with a latent trait (similar to MIMIC model). Suppose that there are three measurements (P1, P2 and P3), and it is hypothesized that there is a single latent trait (L1) underlying the three measurements. The latent variable L1 is influenced by a SNP and a set of variance components, polygenic(p) and random environmental, tal(e). Each trait is also influenced by its own random environmental factor. This is the model diagram.

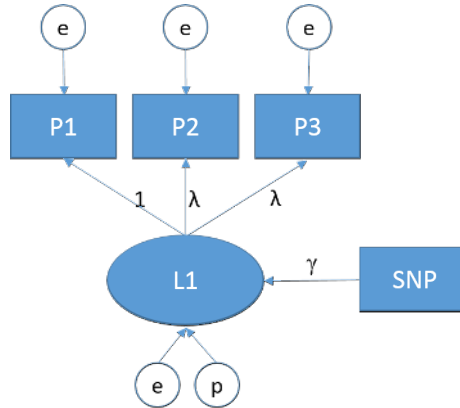


Figure 1: Genetic association analysis model

## 1.1 Construct model

The first step is to construct a **strumModel** object specifying the above model using *createStrumModel()* function.

```
> assoForm1 =
+   'L1 =~ P1 + P2 + P3 + <e>
+   L1 ~ aSNP + <p,e>
+   '
> myAssoModel = createStrumModel(formulas = assoForm1)

Creating strumModel ..... Done

> myAssoModel

Basic properties of the model:
      Model Class ..... strumModel
      Ascertainment ..... FALSE

List of all variables:
      Obs Covariate InEita   InY Exogen.
L1    FALSE      FALSE  TRUE FALSE   FALSE
aSNP   TRUE       TRUE   FALSE FALSE    NA
P1     TRUE      FALSE  FALSE TRUE    NA
P2     TRUE      FALSE  FALSE TRUE    NA
P3     TRUE      FALSE  FALSE TRUE    NA

Model formulas:
      L1 =~ P1 + P2 + P3 + <e>
      L1 ~ aSNP + <p,e>
```

## 1.2 Prepare data

The next step is to prepare data. In this example, the data must be a data.frame with 4 required fields - family, id, father, mother, since the model includes the polygenic variance component (p). To run a strum analysis, you need to construct a **strumData** object created by the *createStrumData()* function with a data.frame. The following code shows the step using the example input file "chr1Ped.csv".

```
> dName = system.file("extdata/example_ped.csv", package = "strum")
> dF = read.csv(dName, header=T)[,1:18]
> names(dF) = c("family","id", "father","mother",names(dF)[5:18])
> myAssoData = createStrumData(dF, "Pedigree")

Creating strumData ..... Done

> myAssoData
```

Data type: Pedigree  
Data size: 477 entries, 18 variables

First 5 rows of data values:

	family	id	father	mother	sex	disease	proband		P1	P2	P3
1	1	1	0	0	0	0	0	0.4093955	0.44450079	-0.3867515	
2	1	2	0	0	1	0	0	-1.5037814	1.52582608	0.8832360	
3	1	3	1	2	0	0	0	1.5850090	0.08833692	0.9322619	
4	1	4	1	2	1	0	0	1.6246356	0.60065352	1.0895325	
5	1	5	1	2	1	0	0	-0.4111477	0.08588345	-0.6477336	

	SBP	DBP	A1	A2	S1	S2	aSNP	rs6040343
1	3.39	4.1800	1.300	1.630	1.9000	-0.0613	1	1
2	-3.49	-2.7200	-0.784	-3.550	-2.8900	-2.1300	0	0
3	-3.40	0.0815	-1.820	-4.390	-1.6700	-3.0900	1	0
4	-7.04	-3.6500	-0.183	-4.740	-2.9800	-2.3500	0	0
5	4.60	4.9900	2.440	-0.117	-0.0408	-0.4350	1	1

phi object contains 75 matrices:

First matrix:

```
$`1`
      1  2  3  4  5  6  7
1 1.0 0.0 0.5 0.5 0.5 0.5 0.5
2 0.0 1.0 0.5 0.5 0.5 0.5 0.5
3 0.5 0.5 1.0 0.5 0.5 0.5 0.5
4 0.5 0.5 0.5 1.0 0.5 0.5 0.5
5 0.5 0.5 0.5 0.5 1.0 0.5 0.5
6 0.5 0.5 0.5 0.5 0.5 1.0 0.5
7 0.5 0.5 0.5 0.5 0.5 0.5 1.0
```

Empty IBD object.

### 1.3 Run analysis

Now, run the association analysis by the function call *strum()* with two previously constructed objects as the arguments.

```
> myAssoResult = strum(myAssoModel, myAssoData)
```

Start STRUM analysis ...

Fitting model step 1 ..... Done

Fitting model step 2 ..... Done

Testing model fit ..... Done

Analysis completed!

## 1.4 Result

The result object contains the model description and two result tables. The first table contains the fitted parameter values with standard errors, confidence intervals, and p-values. The second table contains the information on the model fit from four different measures. For association analysis, you would test  $H_0: \gamma = 0$  versus  $H_1: \gamma \neq 0$ . In this model,  $\gamma$  is the parameter  $L1 \sim \text{aSNP}$ , which equals to 0.9445929 with the  $p\text{value} = 1.591018\text{e-}21$ .

```
> myAssoResult
```

```
=====
      Model
=====
```

Basic properties of the model:

```
Model Class ..... strumFittedModel
Ascertainment ..... FALSE
```

List of all variables:

	Obs	Covariate	InEita	InY	Exogen.
L1	FALSE	FALSE	TRUE	FALSE	FALSE
aSNP	TRUE	TRUE	FALSE	FALSE	NA
P1	TRUE	FALSE	FALSE	TRUE	NA
P2	TRUE	FALSE	FALSE	TRUE	NA
P3	TRUE	FALSE	FALSE	TRUE	NA

Model formulas:

```
L1 =~ P1 + P2 + P3 + <e>
L1 ~ aSNP + <p,e>
```

```
=====
      Result
=====
```

Parameter estimates:

	estimate	stdError	lowerCI	upperCI	pValue
L1=~P2	0.9885029	0.05314867	0.88433341	1.0926724	3.284230e-77
L1=~P3	1.0105280	0.05822325	0.89641252	1.1246435	1.773988e-67
L1~aSNP	0.9445929	0.09913024	0.75030117	1.1388846	1.591018e-21
P1~[intercept]	0.2880181	0.13404076	0.02530298	0.5507331	3.165543e-02
P2~[intercept]	0.1829933	0.13133455	-0.07441771	0.4404043	1.635179e-01
P3~[intercept]	0.3505190	0.12747443	0.10067375	0.6003643	5.964614e-03
P1~~P1<e>	1.1318519	0.18199793	0.77514251	1.4885613	2.501075e-10
P2~~P2<e>	1.2400615	0.14775629	0.95046454	1.5296585	2.377245e-17
P3~~P3<e>	0.6908993	0.19387808	0.31090526	1.0708934	1.829182e-04

L1~~L1<p>	0.9445110	0.26947683	0.41634613	1.4726759	2.283205e-04
L1~~L1<e>	1.1478948	0.20752417	0.74115493	1.5546347	1.588615e-08

Chi-square fit statistics:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	2.719114	7	0.90971460
Mean adjusted	0.2057025	13.218676	7	0.06695628
Mean-Variance adjusted	0.1599908	16.995440	9	0.04878743
Theoretically corrected	NA	11.586595	7	0.11500000

Model fit indices:

	value
Comparative Fit Index (CFI)	1

## 2 Genetic linkage analysis

In this section, we show an example of a typical genetic linkage analysis model with a latent trait using IBD information. Suppose again that there are three measurements as above (P1, P2 and p3) and a single latent trait (L1) underlying the three measurements. The latent variable L1 is influenced by a set of genetic and random variance components. Each trait is also influenced by its own random environmental factor. The model diagram looks like following.

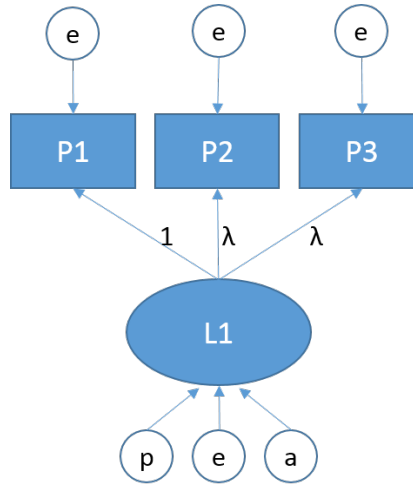


Figure 2: Genetic linkage analysis model

### 2.1 Construct model

The above linkage model can be constructed as a **strumModel** object using *createStrumModel()* function.

```

> linkForm1 =
+   'L1 =~ P1 + P2 + P3 + <e>
+   L1 ~ <a,p,e>
+   '
> myLinkModel = createStrumModel(formulas = linkForm1)

Creating strumModel ..... Done

> myLinkModel

Basic properties of the model:
  Model Class ..... strumModel
  
```

```
Ascertainment ..... FALSE
```

List of all variables:

	Obs	Covariate	InEita	InY	Exogen.
L1	FALSE	FALSE	TRUE	FALSE	FALSE
P1	TRUE	FALSE	FALSE	TRUE	NA
P2	TRUE	FALSE	FALSE	TRUE	NA
P3	TRUE	FALSE	FALSE	TRUE	NA

Model formulas:

```
L1 =~ P1 + P2 + P3 + <e>
L1 ~ <a,p,e>
```

## 2.2 Prepare data

From the above linkage analysis model, **a** represents the major gene variance components, which requires the ibd information to be imported. The ibd information for the family data can be imported by specifying the name of ibd file into *ibdFileName* argument for *createStrumData()*. The use of the example ibd file “GENIBD.chr1Ped.ibd”, which contains the ibd information of family data in “chr1Ped.csv”, is shown in the following code. We use the data.frame, dF, created for the previous association analysis model.

```
> iName = system.file("extdata/GENIBD.chr1Ped.ibd", package = "strum")
> myLinkData = createStrumData(dF, "Pedigree", ibdFileName=iName)
```

```
Importing S.A.G.E. IBD file ..... Done
Creating strumData ..... Done
```

## 2.3 Run analysis

Now, run the linkage analysis by the function call *strum()*. If you want to perform the linkage analysis on all markers exist in the IBD file, you don't need to specify the marker name as an argument for *strum()* function. In this case, each marker will be analysed one by one, and the result object will contain a list of the linkage analysis results for all markers.

```
> myLinkResultAll = strum(myLinkModel, myLinkData)
```

To analyze a subset of IBD markers, then you can specify the names of them as follows;

```
> mNames = c("chr1marker1", "chr1marker2")
> myLinkResult = strum(myLinkModel, myLinkData, ibdMarkers=mNames)
```

```
Start STRUM analysis ...
```

```
chr1marker1:
  Fitting model step 1 ..... Done
  Fitting model step 2 ..... Done
  Testing model fit ..... Done

chr1marker2:
  Fitting model step 1 ..... Done
  Fitting model step 2 ..... Done
  Testing model fit ..... Done
```

Analysis completed!

## 2.4 Result

The result object again contains the model description and result tables. The first table contains the fitted parameter values with standard errors, confidence intervals, and p-values. The second table contains the information on the model fit from four different measures. For linkage analysis, you would test  $H_0: \alpha = 0$  versus  $H_1: \alpha \neq 0$ . In this model,  $\alpha$  is the parameter  $L1 \sim L1<a>$ , which equals to 0.2787365 with the pvalue = 2.594076e-01.

```
> myLinkResult[[1]]
```

```
=====
Model
=====
```

Basic properties of the model:

```
Model Class ..... strumFittedModel
Ascertainment ..... FALSE
```

List of all variables:

	Obs	Covariate	InEita	InY	Exogen.
L1	FALSE	FALSE	TRUE	FALSE	FALSE
P1	TRUE	FALSE	FALSE	TRUE	NA
P2	TRUE	FALSE	FALSE	TRUE	NA
P3	TRUE	FALSE	FALSE	TRUE	NA

Model formulas:

```
L1 =~ P1 + P2 + P3 + <e>
L1 ~ <a,p,e>
```

```
=====
Result
=====
```



Parameter estimates:

	estimate	stdError	lowerCI	upperCI	pValue
L1=~P2	0.8678181	0.07587710	0.7191018	1.016535	2.726799e-30
L1=~P3	0.8914807	0.09331546	0.7085858	1.074376	1.254982e-21
P1~[intercept]	1.1238196	0.11825192	0.8920501	1.355589	2.027470e-21
P2~[intercept]	1.0432875	0.11808091	0.8118532	1.274722	9.977489e-19
P3~[intercept]	1.1673092	0.12331581	0.9256147	1.409004	2.907264e-21
P1~~P1<e>	0.8806093	0.30655087	0.2797806	1.481438	2.035310e-03
P2~~P2<e>	1.3941225	0.19786635	1.0063116	1.781933	9.221350e-13
P3~~P3<e>	0.6740625	0.21180676	0.2589289	1.089196	7.301963e-04
L1~~L1<p>	1.3112519	0.52164084	0.2888546	2.333649	5.973493e-03
L1~~L1<e>	1.4094234	0.35748785	0.7087601	2.110087	4.030537e-05
L1~~L1<a>	0.2788592	0.43189124	-0.5676320	1.125351	2.592465e-01

Chi-square fit statistics:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	3.246504	10	0.9750134
Mean adjusted	0.4180031	7.766697	10	0.6516137
Mean-Variance adjusted	0.3483359	9.320036	12	0.6753792
Theoretically corrected	NA	9.066546	10	0.5258000

Model fit indices:

	value
Comparative Fit Index (CFI)	1

### 3 Structural Equation Model

This is an example of a SEM model with latent variables and polygenic effect. Suppose that there are six measurements and three underline latent variables. anger is a latent variable which underlies the two measurements (A1, A2), bp is a latent variable which underlies the two measurements (SBP, DBP) and stress is a latent variable which underlies the two measurements (S1, S2). bp is caused by anger and stress, and stress is caused by anger and a SNP (rs6040343). All traits and latent variables are also influenced by their own polygenic and random variance components except stress, which the variance is fixed at 0.1 for both polygenic and random components. The model diagram looks like following.

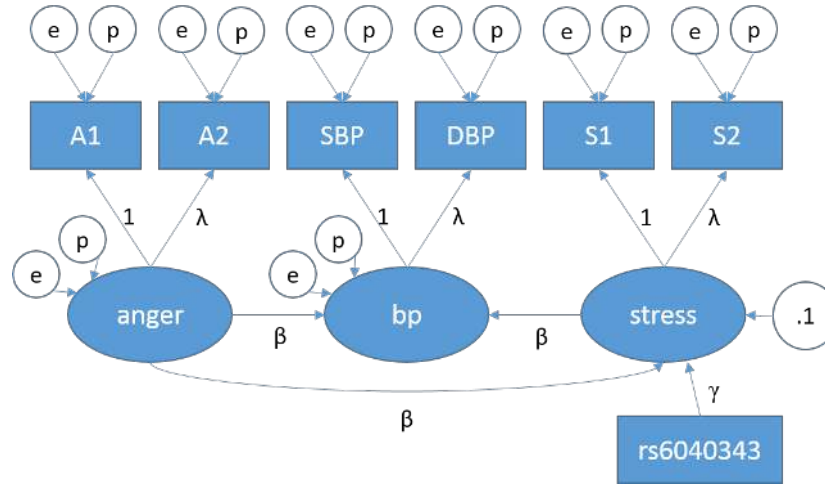


Figure 3: Structural equation model

#### 3.1 Construct model

The above SEM model can be constructed as a **strumModel** object using *createStrumModel()* function.

```
> semForm1 =
+   'bp =~ SBP + DBP
+   anger =~ A1 + A2
+   stress =~ S1 + S2
+   bp ~ anger + stress
+   stress ~ anger + rs6040343
+   var(stress) = .1
+   '
> mySemModel = createStrumModel(formulas = semForm1)
```

```

Creating strumModel ..... Done

> mySemModel

Basic properties of the model:
      Model Class ..... strumModel
      Ascertainment ..... FALSE

List of all variables:
      Obs Covariate InEita   InY Exogen.
bp      FALSE      FALSE   TRUE FALSE   FALSE
anger   FALSE      FALSE   TRUE FALSE   TRUE
stress  FALSE      FALSE   TRUE FALSE   FALSE
rs6040343 TRUE      TRUE   FALSE FALSE   NA
SBP      TRUE      FALSE   FALSE TRUE    NA
DBP      TRUE      FALSE   FALSE TRUE    NA
A1       TRUE      FALSE   FALSE TRUE    NA
A2       TRUE      FALSE   FALSE TRUE    NA
S1       TRUE      FALSE   FALSE TRUE    NA
S2       TRUE      FALSE   FALSE TRUE    NA

Model formulas:
bp =~ SBP + DBP
anger =~ A1 + A2
stress =~ S1 + S2
bp ~ anger + stress
stress ~ anger + rs6040343
var(stress) = .1

```

### 3.2 Prepare data

The next step is to prepare data. Note again that the data must be a `data.frame` with 4 required fields - family, id, father, mother, since the model includes the polygenic variance component (p) by default. A **strumData** object is created by `createStrumData()` function with a `data.frame`. Again, we use the `data.frame`, `dF`, created above using the example input file “chr1Ped.csv”.

```

> mySemData = createStrumData(dF, "Pedigree")

Creating strumData ..... Done

> mySemData

Data type: Pedigree
Data size: 477 entries, 18 variables

First 5 rows of data values:

```

	family	id	father	mother	sex	disease	proband		P1	P2	P3
1	1	1	0	0	0	0	0	0.4093955	0.44450079	-0.3867515	
2	1	2	0	0	1	0	0	-1.5037814	1.52582608	0.8832360	
3	1	3	1	2	0	0	0	1.5850090	0.08833692	0.9322619	
4	1	4	1	2	1	0	0	1.6246356	0.60065352	1.0895325	
5	1	5	1	2	1	0	0	-0.4111477	0.08588345	-0.6477336	

	SBP	DBP	A1	A2	S1	S2	aSNP	rs6040343
1	3.39	4.1800	1.300	1.630	1.9000	-0.0613	1	1
2	-3.49	-2.7200	-0.784	-3.550	-2.8900	-2.1300	0	0
3	-3.40	0.0815	-1.820	-4.390	-1.6700	-3.0900	1	0
4	-7.04	-3.6500	-0.183	-4.740	-2.9800	-2.3500	0	0
5	4.60	4.9900	2.440	-0.117	-0.0408	-0.4350	1	1

phi object contains 75 matrices:

First matrix:

```
$`1`
      1  2  3  4  5  6  7
1 1.0 0.0 0.5 0.5 0.5 0.5 0.5
2 0.0 1.0 0.5 0.5 0.5 0.5 0.5
3 0.5 0.5 1.0 0.5 0.5 0.5 0.5
4 0.5 0.5 0.5 1.0 0.5 0.5 0.5
5 0.5 0.5 0.5 0.5 1.0 0.5 0.5
6 0.5 0.5 0.5 0.5 0.5 1.0 0.5
7 0.5 0.5 0.5 0.5 0.5 0.5 1.0
```

Empty IBD object.

### 3.3 Run analysis

Now, run the analysis by the function call *strum()* with two previously constructed objects as the arguments.

```
> mySemResult = strum(mySemModel, mySemData)
```

Start STRUM analysis ...

Fitting model step 1 ..... Done

Fitting model step 2 ..... Done

Testing model fit ..... Done

Analysis completed!

### 3.4 Result

The result object again contains the model description and result tables. To test the SNP effect to stress, you would test  $H_0: \gamma = 0$  versus  $H_1: \gamma \neq 0$ . In this model,  $\gamma$  is the parameter **stress** ~ **rs6040343**, which equals to 1.013427436 with the pvalue = 2.000891e-12.

```
> mySemResult
```

```
=====
Model
=====
```

Basic properties of the model:

```
Model Class ..... strumFittedModel
Ascertainment ..... FALSE
```

List of all variables:

	Obs	Covariate	InEita	InY	Exogen.
bp	FALSE	FALSE	TRUE	FALSE	FALSE
anger	FALSE	FALSE	TRUE	FALSE	TRUE
stress	FALSE	FALSE	TRUE	FALSE	FALSE
rs6040343	TRUE	TRUE	FALSE	FALSE	NA
SBP	TRUE	FALSE	FALSE	TRUE	NA
DBP	TRUE	FALSE	FALSE	TRUE	NA
A1	TRUE	FALSE	FALSE	TRUE	NA
A2	TRUE	FALSE	FALSE	TRUE	NA
S1	TRUE	FALSE	FALSE	TRUE	NA
S2	TRUE	FALSE	FALSE	TRUE	NA

Model formulas:

```
bp =~ SBP + DBP
anger =~ A1 + A2
stress =~ S1 + S2
bp ~ anger + stress
stress ~ anger + rs6040343
var(stress) = .1
```

```
=====
Result
=====
```

Parameter estimates:

	estimate	stdError	lowerCI	upperCI	pValue
bp=~DBP	1.034984664	0.04257819	0.95153295	1.1184364	1.618860e-130
anger=~A2	1.003405013	0.08030019	0.84601953	1.1607905	7.882569e-36
stress=~S2	1.054993930	0.08276525	0.89277702	1.2172108	3.247140e-37
bp~anger	0.823535846	0.19648258	0.43843707	1.2086346	2.772460e-05
stress~anger	0.942734335	0.07702719	0.79176382	1.0937049	1.924329e-34
bp~stress	1.052784855	0.16354335	0.73224578	1.3733239	1.215819e-10
stress~rs6040343	1.013517766	0.14609692	0.72717307	1.2998625	3.996716e-12
SBP~[intercept]	-0.117910157	0.30223817	-0.71028608	0.4744658	6.964454e-01

DBP~[intercept]	-0.302734202	0.30931966	-0.90898960	0.3035212	3.277234e-01
A1~[intercept]	0.018375775	0.22433007	-0.42130308	0.4580546	9.347151e-01
A2~[intercept]	-0.030596804	0.21819352	-0.45824825	0.3970546	8.884800e-01
S1~[intercept]	0.118914921	0.21176180	-0.29613059	0.5339604	5.744224e-01
S2~[intercept]	0.009989619	0.19519661	-0.37258870	0.3925679	9.591843e-01
SBP~~SBP<p>	0.476876861	0.51400045	-0.53054551	1.4842992	1.767621e-01
DBP~~DBP<p>	1.345964880	0.60228069	0.16551643	2.5264133	1.271590e-02
A1~~A1<p>	1.133270488	0.32775409	0.49088427	1.7756567	2.724188e-04
A2~~A2<p>	0.807378677	0.38682325	0.04921904	1.5655383	1.843489e-02
S1~~S1<p>	1.097072908	0.39554259	0.32182368	1.8723221	2.772076e-03
S2~~S2<p>	0.680992694	0.35439049	-0.01359989	1.3755853	2.732878e-02
SBP~~SBP<e>	1.259598679	0.30646269	0.65894285	1.8602545	1.977261e-05
DBP~~DBP<e>	0.624480268	0.36310715	-0.08719666	1.3361572	4.273225e-02
A1~~A1<e>	0.702416998	0.20541629	0.29980847	1.1050255	3.137041e-04
A2~~A2<e>	1.224986062	0.27512020	0.68576039	1.7642117	4.242850e-06
S1~~S1<e>	1.273526304	0.25231863	0.77899087	1.7680617	2.240559e-07
S2~~S2<e>	1.234374633	0.25196340	0.74053544	1.7282138	4.815705e-07
bp~~bp<p>	0.587268608	0.60016699	-0.58903707	1.7635743	1.639114e-01
anger~~anger<p>	1.154519946	0.25287980	0.65888464	1.6501553	2.491657e-06
bp~~bp<e>	1.078159808	0.40991675	0.27473774	1.8815819	4.266831e-03
anger~~anger<e>	0.886105506	0.19373164	0.50639847	1.2658125	2.393857e-06

Chi-square fit statistics:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	7.315514	25	0.9997751
Mean adjusted	0.5440466	13.446483	25	0.9705453
Mean-Variance adjusted	0.4857559	15.060060	28	0.9777947
Theoretically corrected	NA	17.950139	25	0.8445000

Model fit indices:

	value
Comparative Fit Index (CFI)	1

## 4 SessionInfo

```
> sessionInfo();

R version 3.1.2 (2014-10-31)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] grid      stats      graphics  grDevices  utils      datasets  methods
[8] base

other attached packages:
[1] strum_0.6.1      Rgraphviz_2.10.0 graph_1.44.1    pedigree_1.4
[5] reshape_0.8.5    HaploSim_1.8.4   Matrix_1.1-4

loaded via a namespace (and not attached):
[1] BiocGenerics_0.12.1 lattice_0.20-29  MASS_7.3-35
[4] parallel_3.1.2    plyr_1.8.1      Rcpp_0.11.3
[7] stats4_3.1.2      tools_3.1.2
```

## References

- Morris, N.J., Elston, R.C., & Stein, C.M. (2010). A framework for structural equation models in general pedigrees. *Human heredity*, 70:278–286.
- Song, Y.E., Stein, C.M., & Morris, N.J. (2015). strum: an R package for structural modeling of latent variables for general pedigrees. *BMC Genetics*, 16:35.