

# strum package - examples

Yeunjoo E. Song, Nathan J. Morris

June 30, 2014

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 (e). Each trait is also influenced by its own random environmental factor. This is the model diagram.

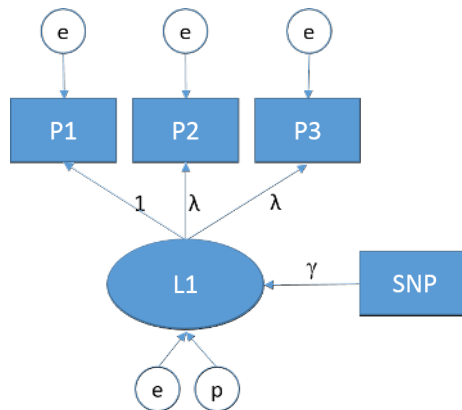


Figure 1: Genetic association analysis model

## 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	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 pvalue = 7.955090e-22.

```
> 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	1.642115e-77
L1=~P3	1.0105280	0.05822325	0.89641252	1.1246435	8.869942e-68
L1~aSNP	0.9445929	0.09913024	0.75030117	1.1388846	7.955090e-22
P1~[intercept]	0.2880181	0.13404076	0.02530298	0.5507331	1.582771e-02
P2~[intercept]	0.1829933	0.13133455	-0.07441771	0.4404043	8.175896e-02
P3~[intercept]	0.3505190	0.12747443	0.10067375	0.6003643	2.982307e-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 statistics of fit:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	2.755859	7	0.99357480
Mean adjusted	0.2057025	13.397307	7	0.06299978
Mean-Variance adjusted	0.1599908	17.225108	9	0.04530531
Theoretically corrected	1.0000000	11.687927	7	0.11130000

## 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.

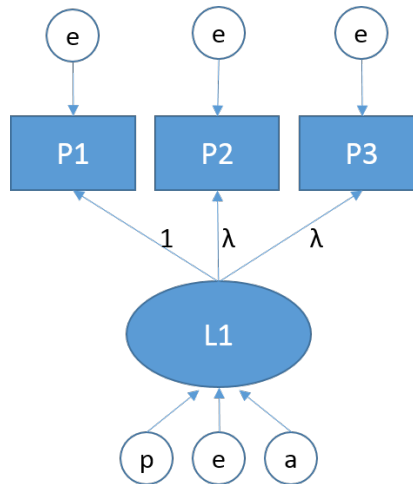


Figure 2: Genetic linkage analysis model

### 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,  $\mathbf{a}$  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;

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

```
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  $L1 \sim L1\langle p \rangle$ , which equals to 1.3114935 with the pvalue = 5.973887e-03.

```
> 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	0.8678695	0.07590839	0.7190918	1.016647	1.428314e-30
L1=~P3	0.8910605	0.09321847	0.7083556	1.073765	5.954188e-22
P1~[intercept]	1.1238196	0.11825192	0.8920501	1.355589	1.013733e-21
P2~[intercept]	1.0432875	0.11808091	0.8118532	1.274722	4.988756e-19
P3~[intercept]	1.1673092	0.12331581	0.9256146	1.409004	1.453634e-21
P1~~P1<e>	0.8799191	0.30659124	0.2790113	1.480827	2.052312e-03
P2~~P2<e>	1.3934769	0.19780518	1.0057859	1.781168	9.293497e-13
P3~~P3<e>	0.6745702	0.21172640	0.2595941	1.089546	7.211563e-04
L1~~L1<p>	1.3114935	0.52174178	0.2888984	2.334089	5.973887e-03
L1~~L1<e>	1.4101136	0.35762894	0.7091738	2.111053	4.024244e-05
L1~~L1<a>	0.2787365	0.43203390	-0.5680344	1.125507	2.594076e-01

Chi-square statistics of fit:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	3.295630	10	0.9862138
Mean adjusted	0.4177845	7.888349	10	0.6397420
Mean-Variance adjusted	0.3481538	9.466019	12	0.6626947
Theoretically corrected	1.0000000	9.054915	10	0.5269000

### 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.

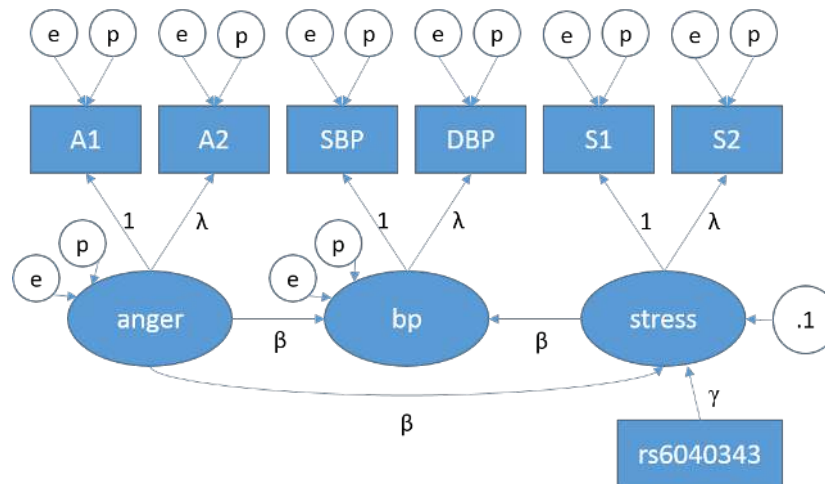


Figure 3: Structural equation model

#### 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	1.035032181	0.04257460	0.95158750	1.1184769	7.494250e-131
anger=~A2	1.003172952	0.08028815	0.84581106	1.1605348	3.991982e-36
stress=~S2	1.055164010	0.08278765	0.89290320	1.2174248	1.652860e-37
bp~anger	0.823477032	0.19646223	0.43841813	1.2085359	1.385408e-05
stress~anger	0.942564976	0.07702239	0.79160386	1.0935261	9.793374e-35
bp~stress	1.052862536	0.16355422	0.73230215	1.3734229	6.077212e-11
stress~rs6040343	1.013427436	0.14608766	0.72710087	1.2997540	2.000891e-12
SBP~[intercept]	-0.117910325	0.30223817	-0.71028625	0.4744656	3.482225e-01

DBP~[intercept]	-0.302734413	0.30931966	-0.90898981	0.3035210	1.638615e-01
A1~[intercept]	0.018375752	0.22433007	-0.42130310	0.4580546	4.673576e-01
A2~[intercept]	-0.030596742	0.21819352	-0.45824819	0.3970547	4.442401e-01
S1~[intercept]	0.118914923	0.21176180	-0.29613058	0.5339604	2.872112e-01
S2~[intercept]	0.009989653	0.19519661	-0.37258867	0.3925680	4.795921e-01
SBP~~SBP<p>	0.477083708	0.51395234	-0.53024436	1.4844118	1.766352e-01
DBP~~DBP<p>	1.345743296	0.60224112	0.16537238	2.5261142	1.272316e-02
A1~~A1<p>	1.133010959	0.32774658	0.49063946	1.7753825	2.731402e-04
A2~~A2<p>	0.807655100	0.38678524	0.04956996	1.5657402	1.839338e-02
S1~~S1<p>	1.097210966	0.39553172	0.32198304	1.8724389	2.768455e-03
S2~~S2<p>	0.680742260	0.35435561	-0.01378198	1.3752665	2.736138e-02
SBP~~SBP<e>	1.259787424	0.30649356	0.65907108	1.8605038	1.975534e-05
DBP~~DBP<e>	0.624278077	0.36314414	-0.08747135	1.3360275	4.279884e-02
A1~~A1<e>	0.702376222	0.20543107	0.29973873	1.1050137	3.142170e-04
A2~~A2<e>	1.225357642	0.27516889	0.68603653	1.7646788	4.231747e-06
S1~~S1<e>	1.273773004	0.25233960	0.77919648	1.7683495	2.234023e-07
S2~~S2<e>	1.234330774	0.25197404	0.74047073	1.7281908	4.825049e-07
bp~~bp<p>	0.586876864	0.60011805	-0.58933290	1.7630866	1.640531e-01
anger~~anger<p>	1.154779475	0.25292433	0.65905691	1.6505020	2.489016e-06
bp~~bp<e>	1.078347705	0.40995946	0.27484193	1.8818535	4.264518e-03
anger~~anger<e>	0.886146282	0.19375800	0.50638758	1.2659050	2.398569e-06

Chi-square statistics of fit:

	kappa	chiStat	df	pValue
Un-adjusted	1.0000000	7.407714	25	0.9999831
Mean adjusted	0.5440739	13.615273	25	0.9680314
Mean-Variance adjusted	0.4857802	15.249105	28	0.9756850
Theoretically corrected	1.0000000	18.067778	25	0.8395000

## 4 SessionInfo

```
> sessionInfo();

R version 3.0.2 (2013-09-25)
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_0.4      Rgraphviz_2.6.0 graph_1.40.0  pedigree_1.4
 [5] reshape_0.8.4  plyr_1.8      HaploSim_1.8.4 Matrix_1.1-0

loaded via a namespace (and not attached):
 [1] BiocGenerics_0.8.0 lattice_0.20-24  MASS_7.3-29      parallel_3.0.2
 [5] Rcpp_0.11.2     stats4_3.0.2    tools_3.0.2
```

## 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.