Inference of interaction networks using generalized Lotka Volterra equations on time-series data with package gLVInterNetworks

Lukas Hirsch

2016-12-24

Generation of in silico data

To download the package from my github account use

```
devtools::install_github("lkshrsch/gLVInterNetworks", build_vignettes = TRUE)
```

To load the package into the R environment

```
library(gLVInterNetworks)
```

To generate in silico data sets we use the following command with the following mandatory arguments:

- species: The amount of variables in the system
- number_of_interactions: The amount of interactions between different variables
- timepoints: Starting from 0, the number of measurements the output will have
- noise: The standard deviation of the stochasticity added to the data (variance of the gaussian noise)
- testData: The percentage of ending datapoints left out for validation purposes.

```
data <- gLVgenerateData(species = 2,
                        number_of_interactions = 2,
                        timepoints = 40,
                        noise = 0.01,
                        testData = 20)
```

Note that depending on the size of the system to simulate (number of species, etc) this command can output messages which can be ignored regarding the difficulty of the numerical integration of the solution.

```
data("exampleData2")
```

The variable data is a list containing the following attributes

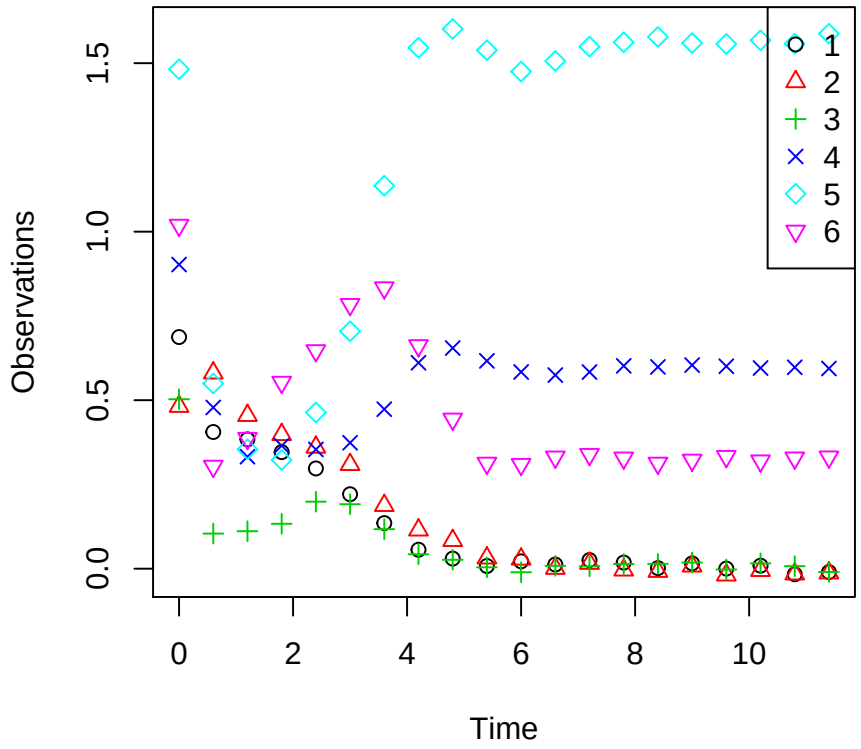
```
names(exampleData2)
```

```
## [1] "species"    "timepoints" "Parms"      "noise"      "sparsity"
## [6] "obs"        "testData"
```

From which *species*, *timepoints*, *noise*, *sparsity*, and *testData* correspond to the input arguments.

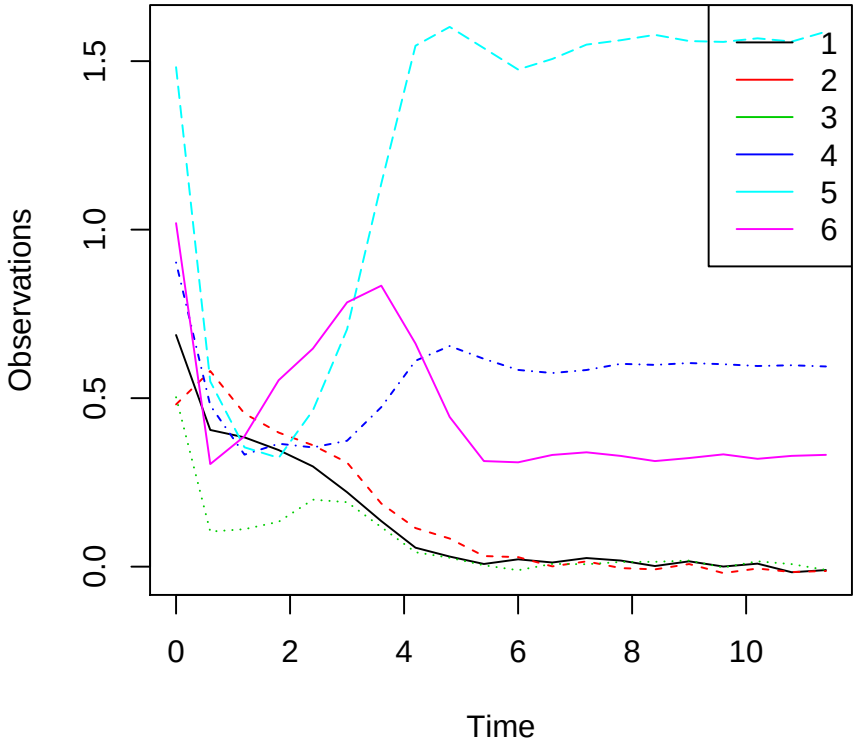
- Parms : Is the randomly generated parameter matrix under the given input constraints. It contains one row per variables / species in the system and the parameters correspond to :
- obs : Corresponds to the solution using the estimated model parameters for the time interval given. This are the values plotted when using

```
plot(exampleData2, pch = 1:exampleData2$species)
legend("topright",
       legend = colnames(exampleData2$obs[,-1]),
       pch = 1:exampleData2$species,
       col=1:exampleData2$species)
```



Or alternatively with the legend flag set to TRUE for an easy plot generation

```
plot(exampleData2, legend = T)
```



Loading data sets

The package comes with a small example of raw data stored in a csv file with a time column with the dates when the observations were taken. The .csv file is a text file with entries sepparated by semicolons ";". To get the absolute path of the file 'example1.csv' installed in the package directory 'rawdata/' use

```
fpath <- system.file("rawdata", "example1.csv", package="gLVInterNetworks")
```

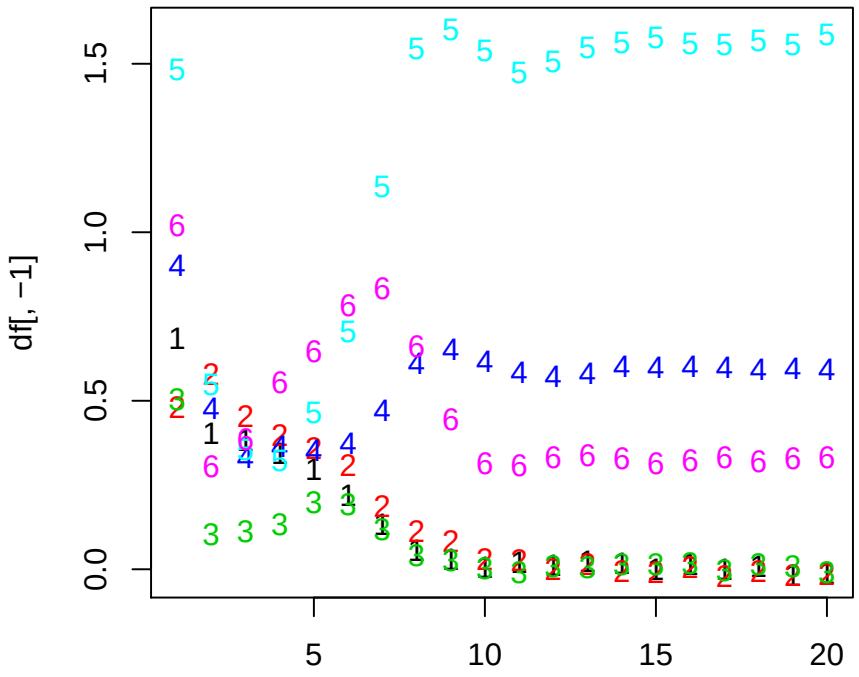
The load the file

```
df <- read.csv2(file = fpath, header = TRUE)
```

```
head(df)
```

```
##      time     Gate1     Gate2     Gate3    Gate.C     Gate5    Gate.6
## 1 01. Feb 0.6873188 0.4814508 0.5028041 0.9024577 1.4820795 1.0192897
## 2 02. Feb 0.4056058 0.5804809 0.1040970 0.4784449 0.5495137 0.3043241
## 3 03. Feb 0.3836464 0.4544713 0.1112002 0.3323488 0.3541356 0.3869106
## 4 04. Feb 0.3459959 0.3978596 0.1333045 0.3648185 0.3228151 0.5533130
## 5 05. Feb 0.2974755 0.3602963 0.1988122 0.3538549 0.4635254 0.6470609
## 6 06. Feb 0.2209130 0.3089127 0.1913458 0.3735745 0.7044591 0.7841929
```

```
matplot(df[,-1])
```



We can convert the data.frame to a numerical matrix using

```
exampleData <- data.matrix(df)
```

With which we have transformed the dates into numerical values. The original measurement time rate (per day) will be the changing rate for the estimated parameters after fitting the model to the data.

```
head(exampleData)
```

```
##      time     Gate1     Gate2     Gate3    Gate.C     Gate5    Gate.6
## [1,]    1 0.6873188 0.4814508 0.5028041 0.9024577 1.4820795 1.0192897
## [2,]    2 0.4056058 0.5804809 0.1040970 0.4784449 0.5495137 0.3043241
## [3,]    3 0.3836464 0.4544713 0.1112002 0.3323488 0.3541356 0.3869106
## [4,]    4 0.3459959 0.3978596 0.1333045 0.3648185 0.3228151 0.5533130
## [5,]    5 0.2974755 0.3602963 0.1988122 0.3538549 0.4635254 0.6470609
## [6,]    6 0.2209130 0.3089127 0.1913458 0.3735745 0.7044591 0.7841929
```

Fitting the generalized Lotka Volterra model to the data

The generalized Lotka Volterra model consists in a set of differential equations, nonlinear in the variables x. It describes the rate of change in time of the abundance of a variable (species, or taxonomical/functional unit)

$$\frac{dx_i}{dt} = \alpha_i x_i + \sum \beta_{ij} x_i x_j$$

Or written simpler

$$\dot{x_i} = x_i(\alpha_i + \sum \beta_{ij} x_j)$$

Written in matrix form for a two species system:

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \times \left[\begin{pmatrix} \alpha_1 & \beta_{11} & \beta_{12} \\ \alpha_2 & \beta_{21} & \beta_{22} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \right]$$

This matrix form for the parameter matrix with the growth term in the first column and interaction parameters in the right-side quadratic matrix corresponds to the output of the parameter in the R code:

```
data("exampleData1")
print(exampleData1$Parms)
```

```
##           growth
## [1,]  1.5334121 -1.653507 -2.652309
## [2,] -0.7558098  5.358952 -1.692742
```

$$= \begin{pmatrix} \alpha_1 & \beta_{11} & \beta_{12} \\ \alpha_2 & \beta_{21} & \beta_{22} \end{pmatrix}$$

And for bigger systems such as the six species simulation of "exampleData2"

```
data("exampleData2")
print(exampleData2$Parms)
```

```
##           growth
## [1,]  2.1118070 -1.297752  0.000000  3.997974 -2.9417880  0.000000
## [2,] -0.7378096  0.000000 -1.867226  1.589395  5.4730475 -2.216080
## [3,]  3.9810420  2.710835 -6.945811 -1.537990  0.0000000 -3.820136
## [4,]  0.1564243  3.865123 -3.941819 -3.679745 -1.3240106  0.000000
## [5,]  1.9821135 -3.323225 -3.122314  0.000000  0.8075217 -1.989126
## [6,]  3.6807751  0.000000  0.000000  0.000000 -8.9701972  1.306043
##
## [1,] -2.625256
## [2,]  0.000000
## [3,]  0.000000
## [4,]  1.906246
## [5,]  2.062700
## [6,] -1.251722
```

Model parameters for variables x_1 to x_6

$$= \begin{pmatrix} \alpha_1 & \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} & \beta_{15} & \beta_{16} \\ \alpha_2 & \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} & \beta_{25} & \beta_{26} \\ \alpha_3 & \beta_{31} & \beta_{32} & \beta_{33} & \beta_{34} & \beta_{35} & \beta_{36} \\ \alpha_4 & \beta_{41} & \beta_{42} & \beta_{43} & \beta_{44} & \beta_{45} & \beta_{46} \\ \alpha_5 & \beta_{51} & \beta_{52} & \beta_{53} & \beta_{54} & \beta_{55} & \beta_{56} \\ \alpha_6 & \beta_{61} & \beta_{62} & \beta_{63} & \beta_{64} & \beta_{65} & \beta_{66} \end{pmatrix}$$

Nonlinear Regression to estimate values for the model parameters

To estimate the parameter matrix from the equations above, use

```
nlrData2 <- gLVnonlinearRegression(data = exampleData2)
```

The resulting list containts 12 objects, named Parms, SSR, residual_SD, SE, residuals_t.test, message, obs, Fit, df, quantitative, qualitative1, qualitative2

The algorithm achieved a fit with a total sum of squared residuals of

```
print(nlrData2$SSR)
```

```
## [1] 0.008053986
```

The estimated parameter matrix is

```
print(nlrData2$Parms)
```

```
##             [,1]          [,2]        [,3]        [,4]       [,5]
## [1,]  1.79223499 -1.6710904970 -0.04198621  3.53269933 -1.4976110
## [2,] -1.14900477  4.4813918003 -3.63452284  2.71661123  2.3632050
## [3,]  5.68942535 -0.0005482157 -8.21869778  1.05303570  1.7927713
## [4,]  0.08948867  5.5097018443 -5.47304844 -4.05040893 -0.7140135
## [5,]  1.81274708 -3.4934328640 -3.30201388 -0.08410942  1.6976769
## [6,]  3.57599179  2.9824639899 -1.69752061  0.48877777 -9.7911085
##            [,6]       [,7]
## [1,] -0.3371221 -2.4281236
## [2,] -0.7749937 -0.1214798
## [3,] -3.9802883 -2.2094892
## [4,] -0.2387796  2.2040078
## [5,] -2.2471820  2.2061703
## [6,]  1.7306970 -1.4043691
```

With standard deviations

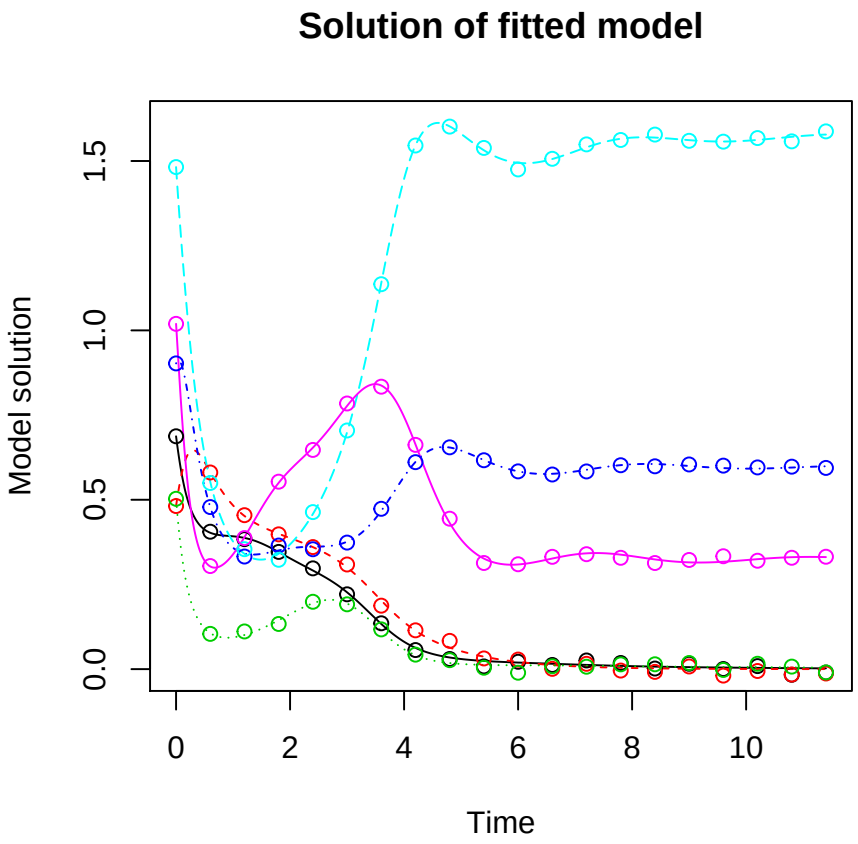
```
summary(nlrData2)
```

```
##          Estimate      StdErr t.value   p.value
##  [1,]  1.79223499  0.46961585  3.8164 0.0002698 ***
##  [2,] -1.14900477  0.54870800 -2.0940 0.0395064 *
##  [3,]  5.68942535  0.85978069  6.6173 4.173e-09 ***
```

```
##  [4,]  0.08948867  0.41054537  0.2180 0.8280178
##  [5,]  1.81274708  0.54137626  3.3484 0.0012536 **
##  [6,]  3.57599179  0.89334751  4.0029 0.0001416 ***
##  [7,] -1.67109050  3.64758521 -0.4581 0.6481285
##  [8,]  4.48139180  3.91428001  1.1449 0.2557586
##  [9,] -0.00054822  8.85457181 -0.0001 0.9999508
## [10,]  5.50970184  2.88580155  1.9092 0.0599081 .
## [11,] -3.49343286  3.28617960 -1.0631 0.2910312
## [12,]  2.98246399  4.98891756  0.5978 0.5516933
## [13,] -0.04198621  1.82769429 -0.0230 0.9817311
## [14,] -3.63452284  1.95242783 -1.8615 0.0664351 .
## [15,] -8.21869778  4.91626804 -1.6717 0.0985839 .
## [16,] -5.47304844  1.68613405 -3.2459 0.0017261 **
## [17,] -3.30201388  1.93846243 -1.7034 0.0924712 .
## [18,] -1.69752061  2.74310962 -0.6188 0.5378310
## [19,]  3.53269933  2.21432123  1.5954 0.1146698
## [20,]  2.71661123  2.24557865  1.2098 0.2300247
## [21,]  1.05303570  3.56257752  0.2956 0.7683344
## [22,] -4.05040893  1.55263263 -2.6087 0.0108885 *
## [23,] -0.08410942  1.24756004 -0.0674 0.9464206
## [24,]  0.48877777  1.94766130  0.2510 0.8025079
## [25,] -1.49761104  3.08391217 -0.4856 0.6285976
## [26,]  2.36320500  2.89647244  0.8159 0.4170473
## [27,]  1.79277129  5.86571855  0.3056 0.7606965
## [28,] -0.71401352  1.13928425 -0.6267 0.5326721
## [29,]  1.69767687  1.20417987  1.4098 0.1625679
## [30,] -9.79110852  2.13858372 -4.5783 1.745e-05 ***
## [31,] -0.33712206  1.30350650 -0.2586 0.7966043
## [32,] -0.77499375  1.28646795 -0.6024 0.5486421
## [33,] -3.98028829  2.54563956 -1.5636 0.1219667
## [34,] -0.23877962  0.68687232 -0.3476 0.7290520
## [35,] -2.24718204  0.68033025 -3.3031 0.0014452 **
## [36,]  1.73069704  1.13444722  1.5256 0.1311577
## [37,] -2.42812362  0.74038135 -3.2796 0.0015552 **
## [38,] -0.12147983  0.73655803 -0.1649 0.8694265
## [39,] -2.20948918  1.09072892 -2.0257 0.0462156 *
## [40,]  2.20400779  0.32405784  6.8013 1.877e-09 ***
## [41,]  2.20617034  0.27332327  8.0717 6.836e-12 ***
## [42,] -1.40436911  0.39934207 -3.5167 0.0007317 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can plot the solution of the fitted model with estimated parameter

```
plot(nlrData2, type="l", main="Solution of fitted model")
points(exampleData2)
```



Linear Regression

Interpretation of results and model identifiability

```
data <- exampleData2
ident <- sensitivityAnalysis(Parms = nlrData2$Parms)
```

```
head(ident$sens)
```

```
head(ident$coll)
```

```
##   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20
## 1  1  1  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 2  1  0  1  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 3  1  0  0  1  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 4  1  0  0  0  1  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 5  1  0  0  0  0  1  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 6  1  0  0  0  0  0  1  0  0   0   0   0   0   0   0   0   0   0   0   0
##   X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   X39 X40 X41 X42 N collinearity
## 1   0   0   0   0 2          1.2
## 2   0   0   0   0 2          1.5
## 3   0   0   0   0 2          2.9
## 4   0   0   0   0 2          1.7
## 5   0   0   0   0 2          4.5
## 6   0   0   0   0 2          2.7
```

```
ident$coll[ident$coll[,"N"]==42,]
```

```
##   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20
## 1  1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   1   1   1   1
##   X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38
## 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##   X39 X40 X41 X42  N collinearity
## 1   1   1   1   1 42         2519
```

```
coll_index <- ident$coll[ident$coll[,"N"]==42,"collinearity"]
```

Thus the least level of precision required for a unique solution is ‘r 1/coll_index’. This can be compared to the level of precision that the model solution fits the original data, described by the standard deviation of the residuals left between them:

Level of precision achieved by the solution:

```
nlrData2$residual_SD
```

```
## [1] 0.008185737
```

Level of precision achievable by different parameterizations (different numeric solutions for the whole parameter matrix):

```
1/coll_index
```

```
## [1] 0.000396906
```

Is the solution unique?

```
nlrData2$residual_SD < 1/coll_index
```

```
## [1] FALSE
```

How is the effect of changing values of each parameter single handedly to the output? (sensitivity analysis)

```
sSens <- summary(ident$sens)
print(sSens)
```

```
##       value    scale      L1      L2     Mean      Min     Max   N
## 1   1.79223  1.79223 5.50281 7.6e-01  4.5e+00  -9.1635 2.3e+01 120
## 2  -1.14900 -1.14900 1.00904 1.4e-01  3.8e-01  -5.5945 2.9e+00 120
## 3   5.68943  5.68943 4.03823 5.5e-01 -1.1e+00 -12.1952 2.0e+01 120
## 4   0.08949  0.08949 0.32623 4.8e-02  2.6e-01  -0.4064 1.9e+00 120
## 5   1.81275  1.81275 2.51148 4.4e-01 -1.9e+00 -23.3270 4.0e+00 120
## 6   3.57599  3.57599 4.12006 5.3e-01 -2.8e+00 -14.4177 9.6e+00 120
## 7  -1.67109 -1.67109 1.23081 1.8e-01 -8.4e-01  -5.8143 3.5e+00 120
## 8   4.48139  4.48139 0.97930 1.3e-01 -4.5e-01  -3.7198 3.1e+00 120
## 9  -0.00055 -0.00055 0.00013 1.8e-05  9.6e-05  -0.0002 5.5e-04 120
## 10  5.50970  5.50970 3.12947 4.4e-01  1.7e+00 -10.1712 1.3e+01 120
## 11 -3.49343 -3.49343 0.89469 1.2e-01 -2.9e-01  -3.2139 2.6e+00 120
## 12  2.98246  2.98246 0.89067 1.2e-01 -2.4e-01  -3.5493 3.1e+00 120
## 13 -0.04199 -0.04199 0.03901 5.5e-03 -2.7e-02  -0.1805 1.1e-01 120
## 14 -3.63452 -3.63452 0.96371 1.3e-01  4.9e-01  -3.1658 3.7e+00 120
## 15 -8.21870 -8.21870 2.47821 3.2e-01  1.8e+00  -4.0186 9.8e+00 120
## 16 -5.47305 -5.47305 3.83910 5.4e-01 -2.2e+00 -16.8511 1.2e+01 120
## 17 -3.30201 -3.30201 1.02866 1.4e-01 -2.9e-01  -3.5767 3.0e+00 120
## 18 -1.69752 -1.69752 0.62857 8.5e-02  1.9e-01  -2.1808 2.5e+00 120
## 19  3.53270  3.53270 1.26892 1.7e-01  9.6e-01  -2.7954 5.3e+00 120
## 20  2.71661  2.71661 0.27490 3.5e-02 -2.2e-01  -1.1356 6.4e-01 120
## 21  1.05304  1.05304 0.11252 1.5e-02 -8.3e-02  -0.3997 1.6e-01 120
## 22 -4.05041 -4.05041 1.12513 1.5e-01 -8.2e-01  -5.2936 2.8e+00 120
## 23 -0.08411 -0.08411 0.00835 1.0e-03 -8.3e-05  -0.0280 3.8e-02 120
## 24  0.48878  0.48878 0.06289 8.5e-03 -3.0e-02  -0.2814 2.0e-01 120
## 25 -1.49761 -1.49761 2.18480 3.0e-01 -1.7e+00 -10.2651 3.8e+00 120
## 26  2.36321  2.36321 1.11606 1.6e-01 -2.1e-01  -3.0115 7.6e+00 120
## 27  1.79277  1.79277 0.68527 1.0e-01 -8.8e-03  -1.6090 4.6e+00 120
## 28 -0.71401 -0.71401 1.37309 2.1e-01 -1.1e+00  -9.0312 1.6e+00 120
## 29  1.69768  1.69768 1.38575 2.6e-01 -1.1e+00 -13.8241 1.7e+00 120
## 30 -9.79111 -9.79111 5.70152 7.4e-01  3.9e+00 -11.9581 2.4e+01 120
## 31 -0.33712 -0.33712 0.90277 1.4e-01 -7.7e-01  -5.5181 1.0e+00 120
## 32 -0.77499 -0.77499 0.88542 1.5e-01  8.4e-02  -7.0927 3.1e+00 120
## 33 -3.98029 -3.98029 2.94273 5.8e-01 -1.1e+00 -30.0480 6.1e+00 120
## 34 -0.23878 -0.23878 1.02844 1.8e-01 -8.5e-01  -8.1009 6.2e-01 120
## 35 -2.24718 -2.24718 4.88671 9.9e-01  4.2e+00  -2.5101 5.1e+01 120
## 36  1.73070  1.73070 2.15542 3.3e-01 -1.6e+00 -12.8612 2.5e+00 120
## 37 -2.42812 -2.42812 4.04551 5.6e-01 -3.4e+00 -14.5913 5.7e+00 120
## 38 -0.12148 -0.12148 0.05925 8.7e-03  4.1e-02  -0.1494 2.5e-01 120
## 39 -2.20949 -2.20949 0.66959 9.5e-02  3.7e-01  -1.4293 2.5e+00 120
## 40  2.20401  2.20401 4.00782 5.7e-01  3.4e+00  -4.6491 2.0e+01 120
## 41  2.20617  2.20617 1.41180 2.2e-01 -1.1e+00  -9.9819 2.5e+00 120
## 42 -1.40437 -1.40437 0.87123 1.1e-01  6.3e-01  -1.8172 3.3e+00 120
```

Less sesntivity (standard: lower L2 value) means that the parameter is harder to identify and the result is more correlated to the values that other parameter obtain. This is seen in the summary statistic of the parameter matrix with each standard deviation. Note that only more sensitive parameters (parameters) are the ones with narrower standard deviation (significantly different from zero):

```
sSens[order(sSens$L2, decreasing = TRUE),c("value","L2")]
```

```
##       value      L2
## 35 -2.24718 9.9e-01
## 1   1.79223 7.6e-01
## 30 -9.79111 7.4e-01
## 33 -3.98029 5.8e-01
## 40  2.20401 5.7e-01
## 37 -2.42812 5.6e-01
## 3   5.68943 5.5e-01
## 16 -5.47305 5.4e-01
## 6   3.57599 5.3e-01
## 5   1.81275 4.4e-01
## 10  5.50970 4.4e-01
## 36  1.73070 3.3e-01
## 15 -8.21870 3.2e-01
## 25 -1.49761 3.0e-01
## 29  1.69768 2.6e-01
## 41  2.20617 2.2e-01
## 28 -0.71401 2.1e-01
## 7  -1.67109 1.8e-01
## 34 -0.23878 1.8e-01
## 19  3.53270 1.7e-01
## 26  2.36321 1.6e-01
## 22 -4.05041 1.5e-01
## 32 -0.77499 1.5e-01
## 2  -1.14900 1.4e-01
## 31 -0.33712 1.4e-01
## 17 -3.30201 1.4e-01
## 8   4.48139 1.3e-01
## 14 -3.63452 1.3e-01
## 11 -3.49343 1.2e-01
## 12  2.98246 1.2e-01
## 42 -1.40437 1.1e-01
## 27  1.79277 1.0e-01
## 39 -2.20949 9.5e-02
## 18 -1.69752 8.5e-02
## 4   0.08949 4.8e-02
## 20  2.71661 3.5e-02
## 21  1.05304 1.5e-02
## 38 -0.12148 8.7e-03
## 24  0.48878 8.5e-03
## 13 -0.04199 5.5e-03
## 23 -0.08411 1.0e-03
## 9  -0.00055 1.8e-05
```

Most sensitive parameters: 35, 1, 30, 33, 40, 37, 3, 16, 6, 5 ...

```
s <- cbind(1:42,summary(nlrData2))
s[order(s[,5]),]
```

```
##               Estimate    StdErr       t.value      p.value
```

```
##  [1,] 41  2.2061703360 0.2733233  8.0716520129 6.835962e-12
##  [2,] 40  2.2040077859 0.3240578  6.8012790095 1.877011e-09
##  [3,]  3  5.6894253515 0.8597807  6.6172983502 4.173052e-09
##  [4,] 30 -9.7911085211 2.1385837 -4.5783143524 1.745095e-05
##  [5,]  6  3.5759917927 0.8933475  4.0029123744 1.415518e-04
##  [6,]  1  1.7922349925 0.4696158  3.8163852354 2.697588e-04
##  [7,] 42 -1.4043691082 0.3993421 -3.5167071615 7.316644e-04
##  [8,]  5  1.8127470798 0.5413763  3.3484051899 1.253632e-03
##  [9,] 35 -2.2471820363 0.6803302 -3.3030753039 1.445232e-03
## [10,] 37 -2.4281236237 0.7403814 -3.2795580464 1.555161e-03
## [11,] 16 -5.4730484445 1.6861340 -3.2459153866 1.726125e-03
## [12,] 22 -4.0504089278 1.5526326 -2.6087361921 1.088848e-02
## [13,]  2 -1.1490047656 0.5487080 -2.0940186289 3.950636e-02
## [14,] 39 -2.2094891841 1.0907289 -2.0256996439 4.621558e-02
## [15,] 10  5.5097018443 2.8858016  1.9092448804 5.990811e-02
## [16,] 14 -3.6345228407 1.9524278 -1.8615401719 6.643507e-02
## [17,] 17 -3.3020138850 1.9384624 -1.7034190816 9.247116e-02
## [18,] 15 -8.2186977797 4.9162680 -1.6717350855 9.858394e-02
## [19,] 19  3.5326993267 2.2143212  1.5953870106 1.146698e-01
## [20,] 33 -3.9802882882 2.5456396 -1.5635710359 1.219667e-01
## [21,] 36  1.7306970414 1.1344472  1.5255862186 1.311577e-01
## [22,] 29  1.6976768727 1.2041799  1.4098200039 1.625679e-01
## [23,] 20  2.7166112334 2.2455786  1.2097600031 2.300247e-01
## [24,]  8  4.4813918003 3.9142800  1.1448827839 2.557586e-01
## [25,] 11 -3.4934328640 3.2861796 -1.0630681475 2.910312e-01
## [26,] 26  2.3632050028 2.8964724  0.8158907254 4.170473e-01
## [27,] 28 -0.7140135230 1.1392842 -0.6267211407 5.326721e-01
## [28,] 18 -1.6975206079 2.7431096 -0.6188307588 5.378310e-01
## [29,] 32 -0.7749937479 1.2864679 -0.6024197876 5.486421e-01
## [30,] 12  2.9824639899 4.9889176  0.5978178541 5.516933e-01
## [31,] 25 -1.4976110385 3.0839122 -0.4856205221 6.285976e-01
## [32,]  7 -1.6710904970 3.6475852 -0.4581361096 6.481285e-01
## [33,] 34 -0.2387796205 0.6868723 -0.3476331994 7.290520e-01
## [34,] 27  1.7927712948 5.8657186  0.3056354100 7.606965e-01
## [35,] 21  1.0530356982 3.5625775  0.2955825360 7.683344e-01
## [36,] 31 -0.3371220579 1.3035065 -0.2586270623 7.966043e-01
## [37,] 24  0.4887777705 1.9476613  0.2509562466 8.025079e-01
## [38,]  4  0.0894886708 0.4105454  0.2179751075 8.280178e-01
## [39,] 38 -0.1214798279 0.7365580 -0.1649290666 8.694265e-01
## [40,] 23 -0.0841094158 1.2475600 -0.0674191328 9.464206e-01
## [41,] 13 -0.0419862101 1.8276943 -0.0229722280 9.817311e-01
## [42,]  9 -0.0005482157 8.8545718 -0.0000619133 9.999508e-01
```

Most certain parameter values: 41, 40, 3, 30, 6, 1, 42, 5, 35, 37 ...

In these two list, eight out of ten parameters coincide (40, 3, 30, 6, 1, 5, 35, 37)

Standard error of model residuals vs Parameter collinearity, Summary of parameter estimates

Network structures