GRENITS:
Gene Regulatory Network Inference Using Time Series

Edward R. Morrissey
Systems Biology Doctoral Training Centre
University of Warwick

October 27, 2020

Overview

GRENITS offers four network inference statistical models using Dynamic Bayesian Networks and Gibbs Variable Selection. A linear interaction model, two linear interaction models with added experimental noise (Gaussian and Student distributed) and a non-linear interaction model ([1] and [2]). The package is intended to be used by both users with a background in Bayesian Inference, as well as casual users. To this end, prior parameters and MCMC (Markov chain Monte Carlo) parameters have been set by default to values that in our experience are adequate for a large range of data sets. As well as this, some basic diagnostic and analysis plots are provided.

The example contained in this document shows how to run the Linear Network model, as well as how to analyse the output and how to make an explicit network prediction.

Example: Network Inference For Simulated Data

Simulated Data

The data we will use for this example is data generated by an Ordinary Differential Equation (ODE) model of *A. thaliana*'s circadian clock [4]. The model was deterministically simulated using COPASI [3]. The data was subsampled to produce hourly time spacing and the log was taken.

```
> data(Athaliana_ODE)
> dim(Athaliana_ODE)

[1]  5 50
```

The data matrix has genes in rows (5) and time points in columns (50) (fig: 1).

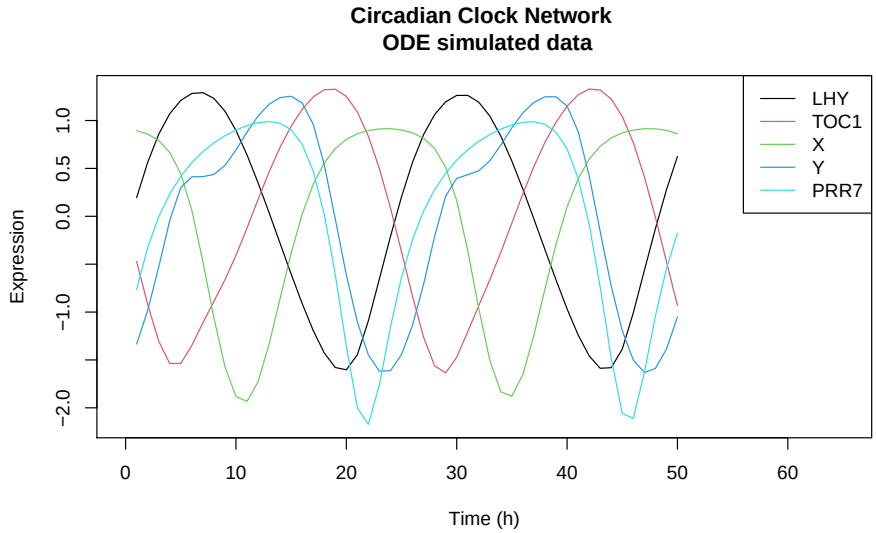

Figure 1: *A. thaliana* Circadian Clock. Simulated data from ODE model

Inference

First we run the MCMC function

```
> output.folder <- paste(tempdir(), "/Example_LinearNet", sep="")
> LinearNet(output.folder, Athaliana_ODE)
```

This will run two MCMC chains with the default parameters. For non default parameter values, a parameter vector can be provided (See ?mcmc.defaultParams_Linear). The output folder contains the raw output from the MCMC run (chain1 and chain2), gene names (geneNames.txt) and the values of the parameters used (runInfo.txt). Next we analyse the output.

```
> # Analyse raw results, place analysis plots and files in output.folder
> analyse.output(output.folder)
```

The results of the analysis are placed in the run folder.

```
> dir(output.folder)
```

```
[1] "AnalysisPlots.pdf"              "ConvergencePlots.pdf"
[3] "NetworkProbability_List.txt"    "NetworkProbability_Matrix.txt"
[5] "ProbNumParents.txt"             "chain1"
[7] "chain2"                         "geneNames.txt"
[9] "runInfo.txt"
```

The Convergence file (ConvergencePlots.pdf) contains plots associated to the adequate convergence of the MCMC. This is crucial for further analysis as, if convergence has not been reached, the results are not trustworthy. The most important parameters to check are the connection probabilities (fig 2). A certain amount of ”jitter” is acceptable. If any link has a probability difference larger

than 0.2 a warning will be issued (both on screen and as a file) and it would be advisable to re-run the MCMC function with a larger number of iterations (See ?mcmc.defaultParams_Linear).

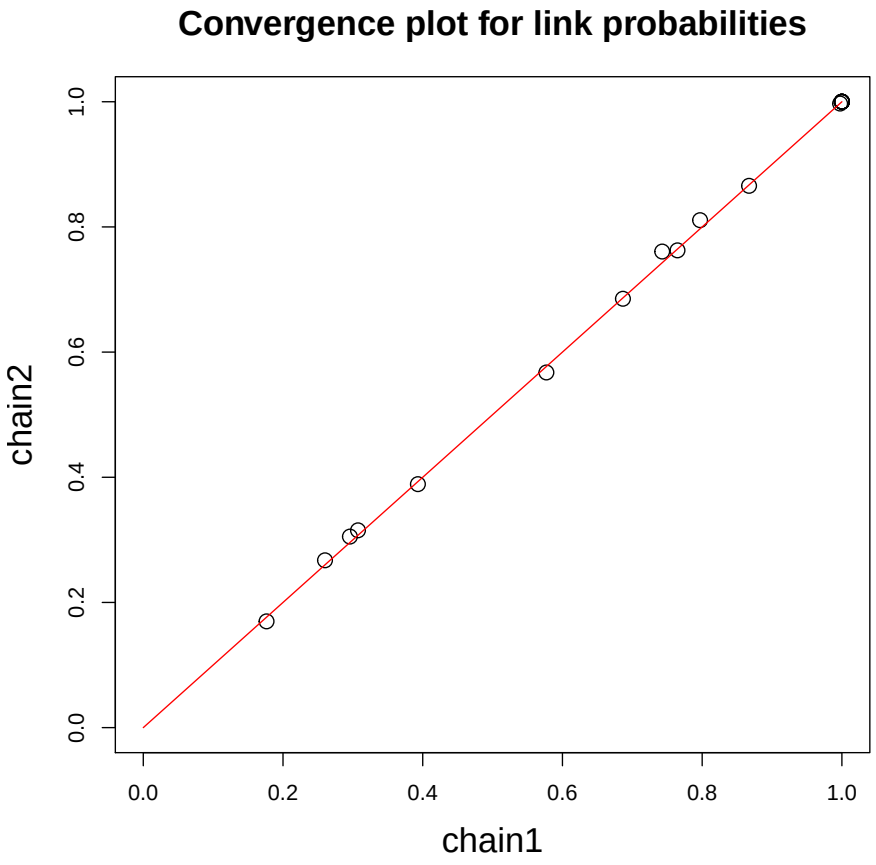

Figure 2: Convergence plot

More formal convergence analysis can also be carried out by reading the chains (?read.chain) and using an MCMC convergence package such as "coda". Although in our experience with the parameters chosen, convergence is not too problematic. Normally monitoring the link probabilities (fig 2) and if necessary running longer runs is sufficient to achieve convergence.

To predict a network we first load the inferred network probabilities

```
> prob.file <- paste(output.folder, "/NetworkProbability_Matrix.txt", sep = "")
> prob.mat  <- read.table(prob.file)
> print(prob.mat)
```

```
           LHY      TOC1         X         Y      PRR7
LHY  0.0000000 0.2601111 1.0000000 1.0000000 1.0000000
TOC1 0.2957778 0.0000000 0.8671111 1.0000000 1.0000000
X    0.6865556 0.5770000 0.0000000 0.7427778 0.7970000
```

```
Y    0.7646667 0.3931111 0.3073333 0.0000000 0.9974444
PRR7 1.0000000 1.0000000 0.9998889 0.1764444 0.0000000
```

By applying a threshold (e.g 0.8) on the link probability (not a p-value!), we can make a network prediction

```
> inferred.net <- 1*(prob.mat > 0.8)
> print(inferred.net)
```

```
     LHY TOC1 X Y PRR7
LHY    0    0 1 1    1
TOC1   0    0 1 1    1
X      0    0 0 0    0
Y      0    0 0 0    1
PRR7   1    1 1 0    0
```

A plot of the inferred network can be seen in fig 3 (right), whereas fig 3 (left) shows how the link probabilities are distributed. The latter plot (found in the file "AnalysisPlots.pdf") is useful when deciding what threshold to use, especially if the links separate into two groups.

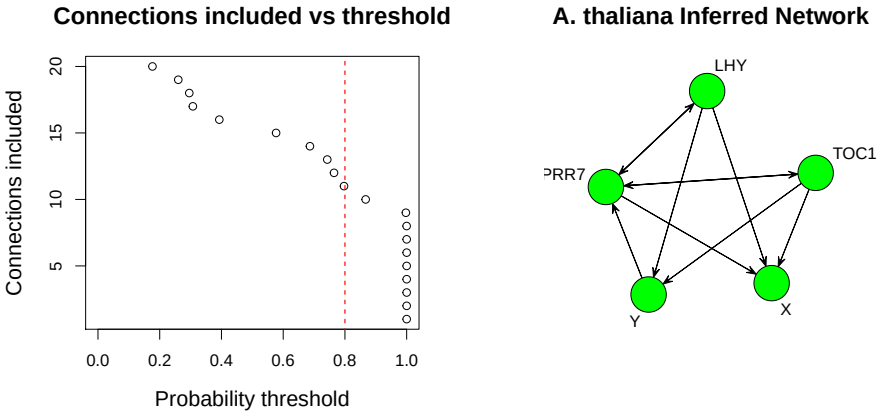

Figure 3: Inferred network. Data used is Simulated data from ODE model

For information on the sign of the interaction (activating/inhibiting) see NetworkProbability_List.txt

```
> prob.list.file <- paste(output.folder, "/NetworkProbability_List.txt", sep = "")
> prob.list      <- read.table(prob.list.file, header = T)
> above.08       <- (prob.list[,3] > 0.8)
> print(prob.list[above.08,])
```

```
   From   To Probability    Strength
5   LHY PRR7   1.0000000  1.88780933
10 TOC1 PRR7   1.0000000  1.43981833
11    X  LHY   1.0000000  0.07761011
12    X TOC1   0.8671111 -0.07565456
```

```
15    X PRR7   0.9998889  0.40993644
16    Y  LHY   1.0000000 -0.12428289
17    Y TOC1   1.0000000  0.43008678
21 PRR7  LHY   1.0000000 -0.12054067
22 PRR7 TOC1   1.0000000 -0.25325467
24 PRR7    Y   0.9974444  0.36404822
```

This file can be used for further analysis using for example cytoscape [5].

References

[1] Morrissey, E.R., Juarez, M.A., Denby, K.J. and Burroughs, N.J. *On reverse engineering of gene interaction networks using time course data with repeated measurements.* Bioinformatics 2010.

[2] Morrissey, E.R., Juarez, M.A., Denby, K.J. and Burroughs, N.J. *Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully Bayesian spline autoregression* Biostatistics 2011.

[3] Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. and Kummer, U. *COPASI a COmplex PAthway SImulator.* Bioinformatics 2006.

[4] Locke, J.C.W., Kozma-Bognar, L., Gould, P.D., Feher, B., Kevei, E., Nagy, F., Turner, M.S., Hall, A. and Millar, A.J. *Experimental validation of a predicted feedback loop in the multi-oscillator clock of Arabidopsis thaliana.* Molecular Systems Biology 2006.

[5] Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B and Ideker T. *Cytoscape: a software environment for integrated models of biomolecular interaction networks.* Genome Research 2003.