

Inference of Networks

Practical – Topics in Computational Biology

16th of June 2010

Initially expression data was used to associate genes to conditions, and expression data analysis was mainly focused on the ability to detect whether a significant change in expression level was observed between two conditions. However, more recently the use of expression data to reconstruct the wiring plans of biology, in particular regulatory networks. In the absence of time series data, the causality information of the true regulatory network is impossible to obtain, but inferring networks of high direct dependencies between gene expressions will usually provide a good and highly informative approximation to the gene regulatory network. In this practical we will only focus on static data, i.e. expression data measured under varying conditions but not following the dynamic development as is the case with time series data. Time permitting you can use the VCell program to generate time series data to analyse with the banjo program.

1 ARACNE

In the conclusion of [1] ARACNE is mentioned as performing well on steady state data, even when only a few experiments are available. So we will start by using this program for network inference. It is available from amdec-bioinfo.cu-genome.org/html/ARACNE.htm in various formats. Beware that the Windows executable requires a Unix environment emulator called CygWin, so use the platform independent Java executable instead. The focus of the ARACNE program is solving an inference problem, so it does not come with a graphical user interface so you will have to run it from a command prompt (a graphical user interface should be available in the zip file found on the ARACNE download page, but it is likely to take you longer to get this to run than it will take figuring out how to run ARACNE from the command line). Navigate to where you downloaded the Java executable to, and make sure that `java -jar ARACNE-java.jar` works - this should print out a usage description, as no input file has been specified.

We will be using the RAF signalling pathway studied in [2]. The advantage of this is that it has been reconstructed with a high degree of accuracy, it small (only containing 11 genes) so it is possible to visually assess the results, and plenty of expression data is available for it. At www.stats.ox.ac.uk/~lyngsoe/ToCB/network_inference/aracne_format you should find the five subsamples of 100 expression level measurements used in [2], as well as one file

combining the five data sets. Try to run ARACNE on one of the small data sets. How is the inference represented in the output data file? Can you load this data file in Cytoscape? How well is the network predicted compared to the correct signalling pathway? The most interesting parameter to play around with in the command line options is that p-value determining how large mutual information needs to be to call an edge. At what p-value do you get the closest match to the true signalling pathway? Is there a difference in inference accuracy when using the combined data set rather than one of the smaller data sets?

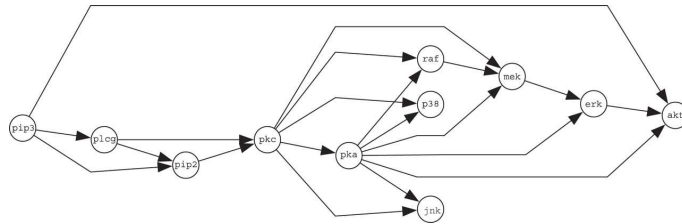


Figure 1: RAF signalling pathway (figure taken from [2])

2 Banjo

ARACNE builds a network by considering each possible edge independently, and adding it to the network if the mutual information between the data of the two nodes it connects is sufficiently high. Banjo, on the other hand, considers the fit to data of all the edges in a full network. It applies a Bayesian Network approach, interpreting the network as a statement of conditional independencies, and performs a heuristic search for the network best describing the observed data.

Start by downloading Banjo from www.cs.duke.edu/~amink/software/banjo/. This requires you to go through a registration process, and you will receive an email with a URL where you can obtain the software for a limited time. Banjo is distributed as an executable java file, and as for ARACNE you will need to run it from a command prompt. Once you have downloaded it, you may want to start the example run outlined in the user guide available in the `doc` subdirectory to make sure that everything at least in theory is working.

Banjo has quite a lot more options to tweak its performance than ARACNE, so you cannot specify all parameters of the run on the command line. Instead it uses a settings file. An example settings file for analysing the RAF pathway data, as well as the data files in a format suitable for Banjo should be available at www.stats.ox.ac.uk/~lyngsoe/ToCB/network_inference/banjo_format. In my experience, you need to specify a proper directory for input and output locations, so make sure to download the data files in the `data` subdirectory of the main Banjo directory. Also, in my experience, with standard Java memory settings the combined data file is too large to handle, so you may only want to attempt to analyse one or two of the smaller data sets. Remember to update the `observationsFile` specification in the settings file to indicate which data set you are analysing. As Banjo considers the full effect of the entire network

and attempts a search over all possible networks, it is significantly slower than ARACNE. How much slower is something you can specify in the settings file, as one of the settings is the maximum run time allowed. As this is just a practical to familiarise you with the software, you may want to lower this value, though evidently the thoroughness of the search will suffer.

Can you input the predicted network into Cytoscape? How well does Banjo predict the RAF pathway? Does the relative performances of ARACNE and Banjo match the conclusions of [1]? Banjo can only deal with discrete observations, and only a small number of such. However, it does contain built-in functionality for discretising real valued data. Where is the discretising strategy specified in the settings file, and what is it? Is there a difference in performance depending on whether interval or quantile discretisation is used? If you still have time available, try testing the performance of the move proposals (as outlined in the ‘The Porposers’ section of the Banjo user guide) and greedy instead of simulated annealing search.

References

- [1] Mukesh Bafna, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernado. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3:78, 2007.
- [2] Adriano V. Werhli, Marco Grzegorzcyk, and Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006.