

Simultaneous gene network and mode of action inference with mNIr

Mario Lauria and Diego di Bernardo
TIGEM - Napoli, Italy
{lauria, dibernardo}@tigem.it

Christine Nardini
DEIS - University of Bologna, Italy
christine.nardini@unibo.it

1 Introduction

The problem of “reverse engineering” gene expression data can be stated as follows: given a set of gene expression data obtained from multiple array experiments, how can we infer the network of genes that produced such data? A number of different approaches have been proposed, based on mutual information/relevance networks (ARACNE [5]), on bayesian networks (BANJO [6]), on clustering algorithms [3], and on deterministic Ordinary Differential Equations (ODE) based methods (NIR [4], MNI [4]).

In particular, the NIR tool follows the ODE-based approach with the additional hypothesis that the gene expression machinery can be approximated as a linear system. NIR is able to infer networks with high accuracy, however its use is restricted to the few cases when information on which genes were perturbed is available. The new approach to gene inference described in this paper builds upon NIR and seeks to obtain a more general tool by removing such restriction. For this purpose, we couple NIR to MNI as described in the next section. MNI is a tool previously developed by diBernardo et al. to infer the mode of action of compounds [2].

2 The design of mNIr

In the assumption of (local) linearity of the system, a formal formulation of the problem can be given as follows: given X , find a pair A, P that satisfies the equation $AX = -P$, where X is the matrix of expression data (of dimensions $N \times M$, one column for experiment, one row per gene), A is the matrix of gene coupling coefficients, and P is the matrix of perturbations.

Based on this formulation, the problem can then be decomposed in the two following consecutive steps: i) given X , infer the matrix P_{est} of most likely gene perturbations (mode of action), then ii) using X and P_{est} , infer the matrix A . Each one of these two steps represents a well known gene expression related problem, for both of which a number of methods have been proposed recently.

Dataset (<i>gene</i> × <i>expts.</i>)	mNir		ARACNE		Banjo		NIR		Random PPV
	PPV	Se	PPV	Se	PPV	Se	PPV	Se	
100 × 100 (<i>u</i>)	0.30	0.06	0.33	0.24	0.33	0.09	0.97	0.87	0.19
100 × 100 (<i>d</i>)	0.22	0.04					0.96	0.86	0.10

Table 1: Performance comparison of existing network inference algorithms. Positive Predictive Value (PPV) represents the accuracy of the inferred network and is defined as $\frac{TP}{TP+FP}$, Sensitivity (Se) is defined as $\frac{TP}{TP+FN}$, with TP: true positive, FP: false positive, FN: false negative. The values were obtained assuming both directed and undirected graph (shown as (*d*) and (*u*) respectively).

Our solution, implemented in a tool called mNir, is to use MNI for step i) (i.e. estimate P from X), and then to use NIR for step ii) (i.e. find A , given X, P). A number of heuristics were developed to maximize the performance of this combination of tools (Figure 1).

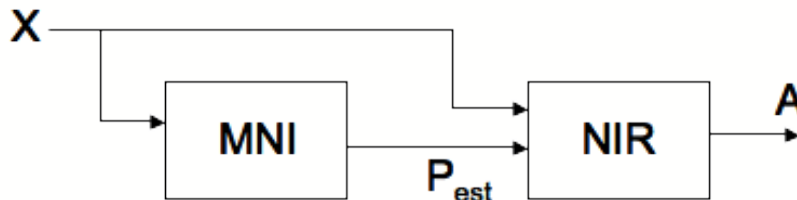


Figure 1: Block scheme of mNir.

3 Results

We implemented mNir using Matlab (ver. 7.1.0.246 (R14)). A set of baseline results obtained using a set of synthetic data are shown in Table 1. The synthetic data was obtained by taking the synthetic networks used in the review paper by Bansal et al. [1]. The matrix of perturbation P employed was one in which 10 genes were perturbed per each experiment - specifically, genes $i, i + 1, \dots, (i + 10) \bmod N$ were perturbed in experiment i . Noise was introduced as described in Bansal et al. [1]. A sample of the results obtained from the simulations are shown in Table 1.

The results from the other algorithms were obtained by running the tools with the default value of their parameters. The column 'Random' refers to the expected performance of an algorithm that selects pairs of genes randomly and then 'infers' an edge between them. The performance of mNir is comparable to Banjo and ARACNE in terms of PPV, even if it does not measure up to the best of the class in terms of combined PPV/Sensitivity. At the same time, the P matrix was estimated with an average accuracy of .67/.79 in terms of PPV/Sensitivity respectively.

References

- [1] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(78), 2007.
- [2] D di Bernardo, MJ Thompson, TS Gardner, SE Chobot, EL Eastwood, AP Wojtovich, SJ Elliott, SE Schaus, and JJ Collins. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature Biotechnology*, 23:377–383, 2005.
- [3] M Eisen, PT Spellman, PO Brown, and D Botstein. Genetics cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, page 1486314868, 1998.
- [4] TS Gardner, D di Bernardo, D Lorenz, and JJ Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301:102–105, 2003.
- [5] A Margolin, I Nemenman, K Basso, C Wiggins, G Stolovitzky, R Della Favera, and A Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, pages S1 (arXiv: q-bio.MN/0410037), 2006.
- [6] Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004.