NIRest: a tool for gene network and mode of action inference

Mario Lauria\textsuperscript{a,4}, Francesco Iorio\textsuperscript{a,4}, Diego di Bernardo\textsuperscript{a,4,5,*}

\textsuperscript{a} Systems Biology Lab, Telethon Institute of Genetics and Medicine, Naples, Italy
\textsuperscript{b} Dept of Computer and Systems Engineering, University of Naples “Federico II”, Naples, Italy
\textsuperscript{c} Dept of Mathematics and Computer Science, University of Salerno, Fisciano, Italy
\textsuperscript{*}corresponding author: dibernardo@tigem.it
\&: These authors contributed equally to this work

Reverse Engineering, Ordinary Differential Equations, Gene Networks

ABSTRACT
A novel algorithm for the identification of genetic networks from gene expression data is presented. Our approach is based on an Ordinary Differential Equations (ODE) model of the network, and on an assumption of linearity around an equilibrium point of the cell machinery. Here we start by describing the application of NIR (Network Identification by Reverse-engineering) to a state of the art in silico gene expression data set provided by the DREAM2 reverse-engineering competition (challenge 4). Then we present NIRest, a tool that builds upon the original NIR and that extends its use to the cases in which the generating perturbations are not known. The remarkable results obtained with in silico datasets support the validity of NIR and NIRest assumptions, and the effectiveness of our approach. Comparison with other leading methods based on Bayesian networks and Mutual Information shows the performance advantage of NIRest, which at the same time provides directed and undirected version of the inferred network and an estimate of the mode of action.

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, we would like to infer the network of genes that produced such data, i.e. gene-gene interactions describing the underlying biological process [cite gardner review and our review in MSB]. A number of different approaches have been proposed, based on mutual information/relevance networks (ARACNE [7]), on bayesian networks (BANJO [8]), on clustering algorithms [3], and on deterministic Ordinary Differential Equations (ODE) based methods (NIR [4], MNI [2]). 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 under the hypothesis that in each experiment a small number of genes has been perturbed (i.e. by overexpression or knockdown), and that the identity of these genes is known.

2. MODEL

As described in [4], in the NIR approach the gene network dynamics, describing the time evolution of the mRNA concentration transcribed by each gene, is modeled by a set of Ordinary Differential Equations:

\[
\frac{dx}{dt} = f(x,u)
\]

where \(x\) represents the mRNA concentrations of the genes in the network and \(u\) is a set of transcriptional perturbations. Assuming that the cell under investigation is at equilibrium near a stable steady-state point, we can apply a small perturbation to each of its genes. A perturbation is small if it does not drive the network out of the basin of attraction of its stable steady-state point and if the stable manifold in the
neighborhood of the steady-state point is approximately linear. With these assumptions the set of non-linear rate equations can be linearized near their stable steady-state point. Thus for each gene \( i \), in a network of \( N \) genes, we can write the equations

\[
\frac{dx_{il}}{dt} = \sum_{j=1}^{N} a_{ij} x_{jl} + u_{il} = a_{l}^T x_{i} + u_{il}, \quad i = 1, \ldots, N, \quad l = 1, \ldots, M
\]

(1)

where \( x_{il} \) is the mRNA concentration of gene \( i \) following the perturbation in experiment \( l \); \( a_{ij} \) represents the influence of gene \( j \) on gene \( i \); \( u_{il} \) is an external perturbation to the expression of gene \( i \) in experiment \( l \).

Identifying the gene interactions network means to derive the matrix \( A \) of the coefficient \( a_{ij} \) for each gene \( i \) in the model described above. This can be accomplished if we measure the mRNA concentration of all the \( N \) genes at steady state (i.e. \( \dot{x}_i = 0 \)) in \( M \) experiments and then solve the system of equations:

\[
AX = -U
\]

where \( X \) is an \( N \times M \) matrix whose columns are the \( x_i \) vectors and \( U \) is an \( N \times M \) matrix whose columns are the \( u_i \) vectors. This system can be solved only if \( M \geq N \), however the recovered weights \( A \) will be extremely sensitive to noise both in the data and in the perturbations and thus unreliable unless we overdetermine the system (increasing the number of experiments or assuming the number of regulators of each gene, \( k \), is very less than \( M \)).

3. APPLICATION TO IN SILICO DATA

Since in the NIR model the knowledge of the perturbation is required, we applied the NIR algorithm to the challenge number 4 of the DREAM competition (The In-Silico-Network Challenge). For this challenge, three In-Silico networks were created and endowed with a dynamics that simulate biological interactions. These networks were generated using a non-linear differential equations model with standard Hill kinetics and the data was obtained by simulating the network with the COPASI software [5, 10, 11]. The challenge consists of predicting the connectivity and some of the properties of one or more of these three networks. Three dataset were available and each of them was produced from the same gene network with 50 genes where the rate of synthesis of the mRNA transcribed by each gene is affected by the level of the mRNA transcribed by other genes. Each of these dataset were provided in three different components:

- **heterozygous**: steady state levels for the wild-type and 50 heterozygous knock-down strains for each gene (+/-). Values of gene expression were provided for a standard condition (steady state).
- **null-mutants**: steady state levels for the wild-type and 50 null mutant strains for each gene (-/-). Values of gene expression were provided for a standard condition (steady states).
- **Trajectories**: time courses of the network recovering from several external perturbations. 23 different perturbations and 26 time points were available.

The data we used was those in the first component (heterozygous) of the first dataset (In-Silico 1). Note that this problem satisfies the NIR-model requirements because we have a sufficient number of experiments and we also know which gene has been perturbed in each experiment. In order to estimate the coefficient of the gene interaction network we solved a linear regression problem for each equation in (1) assuming an upper bound of 10 regressors for each predicted gene, i.e. we assume that each gene can be regulated at most by 10 other genes. The regressor set was chosen according the residual sum of square error (RSS) minimization criterion. Due to the infeasibility of exhaustively searching the best set of regressors in the space of all the possible n-ples (for each value of n), we used the
forward/step-wise method \cite{9} to solve the variable selection problem, given a fixed cardinality \( k \) of regressors. We used the Akaike’s Final Prediction Error (FPE) to solve the model selection problem, i.e. to select \( k \) within a range of values. In our experiments we used a range of \([2, 10]\) unless specified otherwise.

3.1 Results

In order to assign a reliability score to each connection predicted by our method we sorted each coefficient of the obtained \( A \) matrix according to its absolute value. Note that the networks inferred by NIR are always directed and signed so in order to produce an Unsigned version we simply ignored the sign of the inferred network coefficients. In a similar way, for the Undirected version we symmetrized \( A \) by adding its own transpose and dividing the result by 2. The curves reported below have been obtained according to the guidelines of the DREAM2 Challenge Scoring Methodology by using the provided Matlab script.

The performance curves are shown in Figure 1. For the Undirected-Unsigned prediction we obtained a network with 148 edges, with a Positive Predictive Value (PPV) (also known as Recall) equal to 1 until the 33rd most reliable connection, and a PPV of 0.5 if considering the first 100 most reliable connections. For the Directed-Unsigned prediction the performance was essentially the same in terms of PPV, with 328 total connections inferred. For the Directed Signed version, the challenge rules required to separately specify the set of Excitatory connections and the set of Inhibitory connections. The 12 most reliable excitatory connections we predicted were all correct, and the same was true for the 25 most reliable inhibitory connections.

![Figure 1. NIR performance assessment obtained using the DREAM2 Challenge Scoring Methodology](image)

The results obtained show that NIR is a powerful network inference tool when information on the gene
perturbation is available. In next section we describe a novel approach to extend NIR to those cases in which no gene perturbation information is available, i.e. when we have no knowledge of which genes have had their expression level directly changed by the perturbation. This is normally the case for gene expression data obtained following treatment with drugs. In this case we rarely know the molecular targets of the drug and therefore we cannot use NIR as it is. With our proposed extension we can both infer the network in this case and, even more importantly, in principle we can identify the molecular targets of the perturbation, i.e. the mode of action of the drug.

4. EXTENSION OF THE NIR ALGORITHM

In this section we describe a novel approach to gene network inference. It builds upon our previous work and seeks to obtain a more general tool by removing precisely NIR main restriction, i.e. that complete knowledge must be available of which genes have been directly affected by the perturbation experiment (i.e. knowledge of the genes that have been exogeneusly over-expressed or knocked-down). Based on the NIR model, our approach to solving the network inference problem is described by a two-step procedure depicted in Figure 2:

i) given $X$, infer the matrix $U_{est}$ of most likely set of gene perturbations (also called mode of action);
ii) using $X$ and $U_{est}$, infer the matrix $A$ using NIR.

We have implemented these two steps in a tool called NIRest. We have tested a number of different methods to obtain an estimate of $U$, including for example MNI; MNI is an algorithm specifically developed for inferring the mode of action from a set of microarray data [2]. Based on the results of our tests (not shown) in the end we selected an indirect approach, consisting in obtaining a first rough estimate $A_{est}$ of $A$ with a fast statistical method, then deriving the perturbation matrix as $U_{est} = -A_{est}X$.

Then for the second of the two steps we fed $X$ and $U_{est}$ to NIR to obtain a better estimate of $A$ (Figure 2).

Figure 2. Block diagram of NIRest.

4.1 Design and implementation of NIRest

We implemented NIRest using Matlab. As a first step we obtained a matrix $A_{est}$ of coupling coefficients based on a measure of the degree of similarity between the expression profiles of each pair of genes. For each pair of genes we computed the correlation coefficients, and the corresponding $p$-value. For each gene we retained the coefficients associated with the smallest $k$ values of $p$, where $k$ represents the number of
connections to other genes. The use of the correlation coefficient is a time honored technique to estimate the degree of similarity between gene expression profiles [3]; we found that the use of p-values increases the accuracy of the $A$ matrix estimate.

In order to evaluate the performance of our new tool we used a mix of in silico datasets. The in-house datasets were obtained by taking five of the synthetic networks (Net1 to Net5) described in the review paper by Bansal et al. [1] and multiplying their inverse by a matrix $P$ describing the desired set of perturbations, according to the formula $X = -A^{-1} U$. Specifically, the presence of a perturbation to gene $i$ in experiment $j$ was represented by setting to 1 the corresponding element of $U$, i.e., $U_{ij} = 1$. The single perturbation matrix $U^1$ coincided with the identity matrix $I$; the multiple perturbation matrices $U^k$ were obtained by repeated shift-down and addition to itself of $I$. In other words, in the $p$-th experiment genes ($p$-th column of $U^k$) the genes $p, p + 1, \ldots, (p + k) \mod N$ were perturbed (elements in row $p, p + 1, \ldots, (p + k) \mod N$ of $U^k$ were set to 1). Finally, noise was added to $X$ as described in Bansal et al. [1] to simulate experimental errors.

In addition, we used the In-Silico 1 data set described in Section 3 (heterozygous dataset), this dataset is denoted by “IS1” in the tables reported below.

### 4.2 Performance of NIRest

A set of baseline results obtained using the single perturbation datasets is shown in Table 1. The values shown in the tables are the average of the values obtained with the five datasets, except for the IS1 measurements (IS1 is single dataset).

The results from the other algorithms were obtained by running the tools ARACNE, based on Mutual Information [7] and BANJO based on Bayesian networks [8], with their default 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. In addition to the superior performance in terms of Recall and Precision, NIRest has also the advantage of providing an estimate of both the directed and the undirected version of the network. This is an inherent limitation of tools like ARACNE based on a mutual information approach, in that they are able to infer the existence of an edge between two genes but not its direction.

<table>
<thead>
<tr>
<th>Graph type</th>
<th>NIRest</th>
<th>Aracne</th>
<th>Banjo</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Prec</td>
<td>Recall</td>
<td>Prec</td>
</tr>
<tr>
<td>Undir.</td>
<td>0.80</td>
<td>0.49</td>
<td>0.56</td>
<td>0.28</td>
</tr>
<tr>
<td>Dir.</td>
<td>0.47</td>
<td>0.43</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** Comparison on NIRest with other methods in terms of Recall and Precision

The task of inferring a gene network becomes more challenging as the number of perturbations increases, as the effects of each different perturbation is added to the contributions from the others as they propagate through the network. In order to study how its performance varies with expression data of increasing complexity, we tested NIRest on sets obtained from the same set of networks but with increasing numbers of perturbations. The results obtained from the simulations are shown in Table 2. Table 2 reports not only the Recall/Precision values computed on the $A$ estimated by NIRest, but also those computed on the $A_{est}$
Lauria et al.: NIRest: gene network inference

(“CC only” column), and those computed on the $A$ estimated by NIR using the true $U$ (“NIR” column). It is interesting to note that the performance of NIR is unaffected by the complexity of the data set, as long the correct $U$ is given as input. At the same time the degrading performance of the correlation coefficient points to the obfuscating effect of the multiple perturbations. Also, the vanishing advantage of running NIR on the $U_{est}$ derived from $A_{est}$ as the number of perturbations increases points to the sensitivity of NIR to errors in the estimate of $U$. For network IS1, NIRest performance is one order of magnitude better than the random algorithm. The real perturbations for network $A$ are not known; network IS1 was generated using a single perturbation per experiment.

<table>
<thead>
<tr>
<th>Pert per exp.</th>
<th>NIRest</th>
<th>CC only</th>
<th>NIR</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPV</td>
<td>Sens</td>
<td>PPV</td>
<td>Sens</td>
</tr>
<tr>
<td>1</td>
<td>0.80</td>
<td>0.49</td>
<td>0.77</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>0.61</td>
<td>0.37</td>
<td>0.58</td>
<td>0.39</td>
</tr>
<tr>
<td>5</td>
<td>0.35</td>
<td>0.21</td>
<td>0.35</td>
<td>0.22</td>
</tr>
<tr>
<td>10</td>
<td>0.22</td>
<td>0.13</td>
<td>0.22</td>
<td>0.13</td>
</tr>
<tr>
<td>IS1</td>
<td>0.55</td>
<td>0.45</td>
<td>0.56</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 2. Performance with in silico datasets

5. CONCLUSIONS

In this paper we have applied and described a mixed ODE/probabilistic approach to gene network inference. The performance of NIR and NIRest are substantially comparable or better than other state of the art tools for gene network inference, when few genes are perturbed in each experiment. In addition it has the advantage of providing an estimate of both directed and undirected versions of the gene network, and of the mode of action. The performance advantage of NIRest is larger for data expression sets that were generated with a small number of gene expression perturbations. Fortunately, the case of experiments for which a single gene is perturbed is not uncommon; this can be the result of intentional design (for example because a single gene of interest is over-expressed by insertion in a plasmid), or unwillingly because of the nature of the perturbation (for example a drug molecule targeting a specific gene). In many other cases the single perturbation assumption is not rigorously true but it represents an acceptable approximation, for example if a single perturbation happens to be much larger than the others.

6. FUNDING

The authors are partially funded by the Italian Fondazione Telethon, the Italian Fondazione AIRC, the EU FP6 Programme – AnEUploidy Project, and the Italian Ministry of University and Research – FIRB ITALBIONET Project.

7. ACKNOWLEDGMENT

We wish to thank Mukesh Bansal and Lucia Marucci for a number of insightful discussions and suggestions.
References


[10] Available at: http://wiki.c2b2.columbia.edu/dream/index.php/The_In-Silico-Network_Challenges._Description