

DREAM3: Network Inference Using Dynamic Context Likelihood of Relatedness and the Inferelator

Aviv Madar¹, Alex Greenfield², Eric Vanden-Eijnden^{2,3}, Richard Bonneau^{1,2,4*}

1 Center for Genomics and Systems Biology, Department of Biology, New York University, New York, New York, United States of America, **2** Computational Biology Program, New York University School of Medicine, New York, New York, United States of America, **3** Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York, New York, United States of America, **4** Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, New York, United States of America

Abstract

Background: Many current works aiming to learn regulatory networks from systems biology data must balance model complexity with respect to data availability and quality. Methods that learn regulatory associations based on unit-less metrics, such as Mutual Information, are attractive in that they scale well and reduce the number of free parameters (model complexity) per interaction to a minimum. In contrast, methods for learning regulatory networks based on explicit dynamical models are more complex and scale less gracefully, but are attractive as they may allow direct prediction of transcriptional dynamics and resolve the directionality of many regulatory interactions.

Methodology: We aim to investigate whether scalable information based methods (like the Context Likelihood of Relatedness method) and more explicit dynamical models (like Inferelator 1.0) prove synergistic when combined. We test a pipeline where a novel modification of the Context Likelihood of Relatedness (mixed-CLR, modified to use time series data) is first used to define likely regulatory interactions and then Inferelator 1.0 is used for final model selection and to build an explicit dynamical model.

Conclusions/Significance: Our method ranked 2nd out of 22 in the DREAM3 100-gene *in silico* networks challenge. Mixed-CLR and Inferelator 1.0 are complementary, demonstrating a large performance gain relative to any single tested method, with precision being especially high at low recall values. Partitioning the provided data set into four groups (knock-down, knock-out, time-series, and combined) revealed that using comprehensive knock-out data alone provides optimal performance. Inferelator 1.0 proved particularly powerful at resolving the directionality of regulatory interactions, i.e. “who regulates who” (approximately 93% of identified true positives were correctly resolved). Performance drops for high in-degree genes, i.e. as the number of regulators per target gene increases, but not with out-degree, i.e. performance is not affected by the presence of regulatory hubs.

Citation: Madar A, Greenfield A, Vanden-Eijnden E, Bonneau R (2010) DREAM3: Network Inference Using Dynamic Context Likelihood of Relatedness and the Inferelator. PLoS ONE 5(3): e9803. doi:10.1371/journal.pone.0009803

Editor: Mark Isalan, Center for Genomic Regulation, Spain

Received: August 13, 2009; **Accepted:** January 18, 2010; **Published:** March 22, 2010

Copyright: © 2010 Madar et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The work of E.V.-E. was partially supported by National Science Foundation (NSF) grants DMS02-09959, DMS02-39625 and DMS07959, DMS02-39625 and DMS07-08140, and Office of Naval Research (ONR) grant N00014-04-1-0565. The work of R.B. was partially supported by NSF grant DBI-08207, and Department of Defense (DOD) grant USAMRAA W81XWH-04-1-0307. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: Bonneau@nyu.edu

Introduction

For decades the biological community has had a keen interest in characterizing the genetic regulatory networks that are largely responsible for an organism's ability to adapt to its constantly changing environment. An ever increasing number of functional genomics projects continue to make this a key problem in modern biology. It remains, however, unclear what constitutes the most efficient paradigm for characterizing regulatory networks, i.e. what experiments to perform, data to collect, and methods to use for learning biological regulatory networks. Moreover, the number of proposed methods for learning regulatory networks from systems data is growing and it is difficult to compare the relative merit of these methods unless methods are evaluated on similar datasets using similar metrics. The DREAM (Dialogue for Reverse Engineering Assessments and Methods) project [1,2] aims to shed

light on which paradigm is most useful for characterizing regulatory networks. It does so by posing a set of challenges to the computational biology community at large, allowing for the comparison of different methods on identical footing.

There are several broad classes of regulatory network inference methods that aim to reconstruct and model the underlying regulatory networks at varying degrees of detail. It is beyond the scope of this introduction to review more than a small subset of these methods, as they represent a very large body of work, for a more thorough review of network reconstruction methods we refer the reader to [3–10]. Here we focus on two classes of methods, namely: Mutual Information (MI) and Ordinary Differential Equation (ODE) based methods.

Mutual information based methods [11–17] are often formulated such that it is not necessary to assume a functional form for the effect of a regulator on its target(s); mutual information does not assume a

linear relationship between any given pair of genes (or any parametric relationship, for that matter). These methods often scale well to genome-wide regulatory networks, providing advantages over more detailed models in cases where the functional forms of regulatory interactions are unknown, complex, or when there is insufficient data to learn more intricate models. However, MI based methods as previously formulated, provide limited insight into the dynamic behavior of the system, and hence have limited use in predicting new observations—a key property for estimating a model’s relevance when the ground truth is unknown.

Ordinary Differential Equation based methods [18–30], aim to learn a set of ODEs describing the time evolution of target genes as a function of their likely regulators. These methods can provide deeper understanding of the system’s dynamic behavior, and can be used to predict new observations. However, they can be computationally demanding, and may require accurate measurement of a large number of parameters.

Here we employ a modified version of the MI based method Context Likelihood of Relatedness (CLR) [14] (modified to use time-series data) to reduce the space of possible regulatory interactions for a more detailed, scalable, ODE based method, Inferelator 1.0 [25]. CLR is an easily scalable MI based method for network reconstruction. In order to identify regulatory interactions CLR computes the MI between the expression levels of every possible regulator-target gene pair, and then computes a score for each pair. The score for a pair is a function of two z-scores resulting from comparing the pair’s MI value with: all MI values involving the regulator (to generate the regulator z-score), and all MI values involving the target (to generate the target z-score). CLR takes advantage of the fact that biological networks are, on average, quite sparse and assumes the majority of MI values involving a given target or regulator are insignificant, and thus constitute a background MI distribution. The method has been successfully used in the past to learn previously validated, as well as novel, transcriptional regulatory interactions in *E. coli* [14]. This method, as originally published, cannot resolve causality, i.e. which gene is the regulator and which gene is the target for a given significant interaction, as it relies solely on the symmetric MI matrix and can not take advantage of the kinetics represented in time series data.

The Inferelator 1.0 is a scalable method that uses an additive ODE model to approximate regulatory dynamics. At the core of the method is an l_1 -norm constrained regression algorithm, LASSO [31] (implemented using Least Angle Regression (LARS) [32]), that is used to efficiently choose a parsimonious set of likely regulators for each target gene, and to estimate the kinetic parameters associated with these interactions. Inferelator 1.0, like CLR, takes account of the sparsity found in biological networks by imposing an l_1 constraint on the kinetic parameters, resulting in a sparse (parsimonious) regulatory model. The method has been used successfully in the past to learn a large portion of *H. salinarum* transcriptional regulatory network, and was able to predict mRNA levels of 85 percent of the genes in the genome over new experimental conditions [33]. Two similar methods have also been successfully applied to the learning of human regulatory networks mediating TLR-5 response in macrophages [34], and to the DREAM2 50-gene *in silico* network challenge [27]. The Inferelator 1.0 as originally published included interactions between regulators in the ODE model.

Several network reconstruction methods, including the method described here, restrict the number of considered regulatory interactions using a correlation or MI based pre-processing step. For example, the Sparse Candidate Algorithm, a Bayesian network approach for learning biological regulatory networks, employed a mutual information pre-processing step, aimed at reducing complexity and improving the scaling of the algorithm to the genome scale [35]. Here, we describe and test an overall pipeline in which a modified version of CLR, a version that computes dynamic and static mutual information values for regulator-target pairs (mixed-CLR), is used as a pre-processing step for a more detailed ODE based method—Inferelator 1.0. We show that this overall dynamic pipeline identifies more directed true regulatory interactions when compared to pipelines that are based on a static model.

Methods

Here we describe the three step pipeline (Figure 1) that we have applied to the DREAM3 in-silico network challenge [1,2].

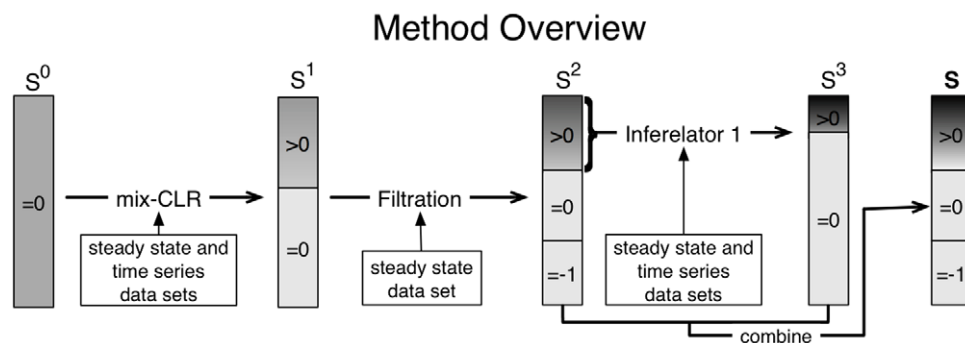


Figure 1. Method outline. For each regulatory interaction, $x_j \rightarrow x_i$, we define a confidence score s_{ij}^m , where $m=1,2,3$ indicates the step in our pipeline. We store these confidence scores in a corresponding $N \times N$ matrix, S^m (eq. 2), which we depict in the figure as a sorted list (from high to low confidence) of regulatory interactions. We schematically represent true positives (TPs) density (within any subset) as a gray scale, where black indicates high TP density. All possible pair-wise regulatory interactions are first scored using mixed-CLR, resulting in a matrix S^1 . We then filter out the least likely regulatory interactions based on the knock-out and knock-down steady-state observations, resulting in a matrix S^2 (the confidence score of each removed regulatory interaction was set to minus one, and thus sent to the back of the list). Lastly, we evaluate regulatory interactions in the TP enriched subset using Inferelator 1.0, by building an ODE model for each target gene. The kinetic weights from these ODE models were converted into confidence scores (S^3) and combined with S^2 to produce the final ranked list, S (eq. 32). The regulatory interactions scored in S , when ranked from high to low, represent our final ranking for each regulatory interaction.
 doi:10.1371/journal.pone.0009803.g001

Problem Set Up

The dynamical variables available from observations are the simulated mRNA levels of genes:

$$x(t) = (x_1(t), \dots, x_N(t))^T. \tag{1}$$

We are given data sets that contain observations taken from five different networks [36]. Each data set is composed of multiple sets of time series observations—where the system was perturbed and then measured at equal time intervals—and steady state observations—where the system was perturbed and then measured once it reached a steady state. Perturbations for time series observation consist of changing the initial expression levels of all genes. Perturbations for steady state observation consist of either knocking out one gene at a time, i.e. one of the gene’s initial expression level is set close to zero, or knocking down one gene at a time, i.e. one of the gene’s initial expression level is set close to half its wild-type expression level.

The DREAM3 *in silico* network challenge required participants to produce a ranked list of all possible pair-wise regulatory interactions, $x_j \rightarrow x_i$ (x_j regulates x_i), ordered by confidence. A perfect ranking would have all of the true regulatory interactions, i.e. true positives (TPs), ranked before all of the false regulatory interactions, i.e. true negatives (TNs). This paper does not address the relative strengths of the regulatory interactions or the kinetic constants learned by Inferelator 1.0, as predicting topology of a regulatory network was the main focus of this challenge.

To determine rankings, we will define a confidence score, $s(x_i, x_j)$, at each step in our pipeline to indicate our confidence in any given regulatory interaction, $x_j \rightarrow x_i$. We store these values in the form of a $N \times N$ matrix of confidence scores:

$$S^m = \begin{pmatrix} s^m(x_1, x_1) & s^m(x_1, x_2) & \cdots & s^m(x_1, x_N) \\ s^m(x_2, x_1) & s^m(x_2, x_2) & \cdots & s^m(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ s^m(x_N, x_1) & s^m(x_N, x_2) & \cdots & s^m(x_N, x_N) \end{pmatrix} \tag{2}$$

where the superscript, $m \in \{1, 2, 3\}$, indicates our measure of confidence after steps, one, two, and three in our pipeline, respectively. Note that columns in S^m correspond to regulators, and rows correspond to targets.

Without loss of generality we can assume that time-series observations resulted from one perturbation experiment, i.e. we can write them in the form of a $N \times K$ matrix of observations

$$X^{ts} = \begin{pmatrix} x_1(t_1) & x_1(t_2) & \cdots & x_1(t_K) \\ x_2(t_1) & x_2(t_2) & \cdots & x_2(t_K) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(t_1) & x_N(t_2) & \cdots & x_N(t_K) \end{pmatrix} \tag{3}$$

where (t_1, t_2, \dots, t_K) are the observation times.

There are two main sets of steady-state experiments: measurements of all genes when one gene (per experiment), x_i , is knocked out, which we denote as $x(x_i(-/-))$; and measurements of all genes when one gene (per experiment), x_i , is knocked down, which we denote as $x(x_i(-/+))$. For diploid cells, cells that contain two sets of chromosomes (one set

donated from each parent), the notations $(-/-)$ and $(-/+)$ are often used to indicate that both copies of a gene are non-functional or that one copy of a gene is non-functional, respectively. These two sets of experiments are complemented with one steady-state experiment that describe the system at the lack of any perturbation, so called wild type expression levels, which we denote as $x(w.t.)$.

Denote by e the vector of all steady-state experiments, i.e.

$$e = (w.t., x_1(-/-), \dots, x_N(-/-), x_1(-/+), \dots, x_N(-/+)), \tag{4}$$

then we can write all of the steady-state observations in the form of a $N \times M$ matrix

$$X^{st} = \begin{pmatrix} x_1(e_1) & x_1(e_2) & \cdots & x_1(e_M) \\ x_2(e_1) & x_2(e_2) & \cdots & x_2(e_M) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(e_1) & x_N(e_2) & \cdots & x_N(e_M) \end{pmatrix}. \tag{5}$$

where $M = 2N + 1$ indicates the total number of steady-state experiments.

Note that unlike typical genome-wide mRNA observations, the observations given in DREAM3 ranged from zero to one (e.g. microarray and RNA-seq can exhibit multiple \log_{10} units of range). This suggested to us that the DREAM3 data set, as provided, was properly normalized and thus we did not take any further data-normalization steps.

Step 1.a: Computing static and dynamic Mutual Information between Regulators and Targets

As the first step in our pipeline we apply our modified CLR algorithm (mixed-CLR) to reduce the number of likely regulators for each target (i.e. gene). This procedure has two parts: 1) computing static and dynamic Mutual Information (MI) between each potential regulator and target pair, followed by 2) a background correction step, for which we use the procedure originally described in [14].

We use MI as a metric of statistical dependency between two genes. MI between two random variables X and Y can be defined as [37,38]

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{6}$$

where $p(x, y)$ is the joint probability distribution function of X and Y , and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y , respectively, i.e. the probability that $X = x$ and $Y = y$, respectively.

When computing MI from continuous data a binning approach is often used [11]. Binning can lead to crude estimates of the probabilities involved, especially for small data sets. Fuzzy binning (smoothing), where each point is assigned to a number of bins with an associated weight, can alleviate this situation, leading to better estimates of probabilities. Here, we compute mutual information using a smoothing B-spline approach proposed by [39], with ten bins, and third-order B-splines (for a detailed description we refer the reader to [39]). An R [40] package for this method is available from the authors upon request (code is based in part on [14,39]).

Using both time-series and steady-state observations (the full set of provided experiments) we compute the static MI between

the observed expression levels of every gene pair, $I(x_i; x_j)$, and store their values in the form of a $N \times N$ matrix,

$$M^{\text{stat}} = \begin{pmatrix} I(x_1; x_1) & I(x_1; x_2) & \cdots & I(x_1; x_N) \\ I(x_2; x_1) & I(x_2; x_2) & \cdots & I(x_2; x_N) \\ \vdots & \vdots & \ddots & \vdots \\ I(x_N; x_1) & I(x_N; x_2) & \cdots & I(x_N; x_N) \end{pmatrix}. \quad (7)$$

Computing MI between the expression levels of genes with the purpose of characterizing regulatory interactions has two major limitations: 1) a pair of genes can often have a high MI value due to many reasons other than a regulatory interaction, e.g. a pair of genes can share a regulator; and 2) MI between the expression levels of two genes is a symmetric quantity, and thus can not resolve causality, i.e. can not resolve the directionality of the regulatory interaction. To partially resolve these limitations we compute dynamic MI values, derived from a linear additive ODE model, motivated by our previous work on Inferelator 1.0 [25].

We assume that the time evolution in the x 's can be approximated by the linear ODE:

$$\frac{dx_i(t)}{dt} = -\alpha_i x_i + \sum_{\substack{j=1 \\ j \neq i}}^N \beta_{i,j} x_j(t), \quad i=1, \dots, N \quad (8)$$

where $\alpha_i > 0$ is the first-order degradation rate of x_i , and

$$\beta = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,N} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N,1} & \beta_{N,2} & \cdots & \beta_{N,N} \end{pmatrix} \quad (9)$$

is a set of parameters to be estimated. Note that the matrix β is typically sparse, i.e. most entries are 0, and that it is given that auto-regulatory interactions do not exist in any of the DREAM3 networks, i.e. $\beta_{i,i} = 0$ for all i .

The next two steps aim to separate the terms in (8) that involve the putative regulators (i.e. the explanatory variables) from the terms in (8) that involve the target (i.e. the response), first for time-series experiments and then for steady-state experiments.

For time-series experiments we can write (8) using a finite difference approximation as

$$\tau_i \frac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k} + x_i(t_k) = \tau_i \sum_{\substack{j=1 \\ j \neq i}}^N \beta_{i,j} x_j(t_k), \quad (10)$$

$$i=1, \dots, N \quad k=1, \dots, K-1$$

where $\tau_i = \frac{1}{\alpha_i}$ is related to the half-life of x_i by $t_{1/2} = \tau_i \ln(2)$, and is set throughout this work to 10 minutes (i.e. half-life time of 7 minutes). This value is in the range of many known mRNA half-life times for *E. coli* [41], and previous work has shown that error in τ_i can be compensated for via an overall scaling of β

(Bonneau et al., unpublished). Thus, for every regulator (x_j) target (x_i) pair we can define a time-series response variable, $y_i(t_{k+1})$, as

$$y_i(t_{k+1}) = \tau_i \frac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k} + x_i(t_k), \quad (11)$$

with a corresponding explanatory variable, $x_j(t_k)$; both derived from the left- and right-hand-sides of (10), respectively.

For steady state experiments we can write (8) by setting the derivative to zero as

$$x_i(e_l) = \tau_i \sum_{\substack{j=1 \\ j \neq i}}^N \beta_{i,j} x_j(e_l), \quad i=1, \dots, N, \quad l=1, \dots, M. \quad (12)$$

Thus, for every regulator (x_j) target (x_i) pair we can define a steady-state response variable, $y_i(e_l)$, as

$$y_i(e_l) = x_i(e_l), \quad (13)$$

with a corresponding explanatory variable, $x_j(e_l)$; both derived from the left and right-hand-sides of (12), respectively.

Combining the time-series and steady-state response variables, we get the response vector:

$$y_i = (y_i(t_2), \dots, y_i(t_K), y_i(e_1), \dots, y_i(e_M)). \quad (14)$$

Combining the corresponding time-series and steady-state explanatory variables together, we get the explanatory variables vector:

$$x_j = (x_j(t_1), \dots, x_j(t_{K-1}), x_j(e_1), \dots, x_j(e_M)). \quad (15)$$

Note that for time-series, each explanatory variable ($x_j(t_k)$) is time-lagged with respect to its corresponding response variable ($y_i(t_{k+1})$). Given well-sampled time series, it is easy to see how this may help resolve causation. However, we also consider the MI between the pair ($x_j(t_k), y_i(t_{k+1})$) helpful at reducing statistical dependencies that are not due to direct regulatory interactions (when compared to the MI between the pair ($x_j(t_k), x_i(t_k)$)). This is based on the simple, yet biologically relevant, assertion that a transcription factor (x_j) directly affects the rate of change of its target mRNA (approximated by y_i) and not the accumulated amount of that target gene mRNA. In the Result section we shall see that for DREAM3 100-gene networks this is indeed the case.

We compute the dynamic MI between every pair of response-vector and explanatory-variable vector, $I(y_i; x_j)$, and store their values in the form of a $N \times N$ matrix,

$$M^{\text{dyn}} = \begin{pmatrix} I(y_1; x_1) & I(y_1; x_2) & \cdots & I(y_1; x_N) \\ I(y_2; x_1) & I(y_2; x_2) & \cdots & I(y_2; x_N) \\ \vdots & \vdots & \ddots & \vdots \\ I(y_N; x_1) & I(y_N; x_2) & \cdots & I(y_N; x_N) \end{pmatrix}. \quad (16)$$

Note that static- and dynamic-MI values are estimated using the same number of observations. Next we describe how to use

dynamic MI values as part of a modified CLR background correction.

Step 1.b: Context Likelihood of Relatedness (CLR). At the core of both the original CLR method and our modified CLR variant, mixed-CLR, is a background correction step that computes the significance of a given regulator-target MI value by comparing that value to all MI values for that regulator and all MI values for the given target. This background correction step can be briefly described as follows:

Let M be a $N \times N$ matrix, with each entry, M_{ij} , equals the pair-wise MI between a pair of variables, $I(x_i; x_j)$. In order to derive a CLR score for that pair of variables, $z(x_i, x_j)$, first compute a positive Z-score for M_{ij} with respect to the entries in the i 'th row of M , i.e.

$$z_i(x_i, x_j) = \max \left(0, \frac{M_{ij} - \frac{\sum_j M_{ij}}{N}}{\sigma_i} \right), \quad (17)$$

where σ_i is the standard deviation of the entries in the i 'th row of M . Second, compute a positive Z-score for M_{ij} with respect to the entries in the j 'th column of M , i.e.

$$z_j(x_i, x_j) = \max \left(0, \frac{M_{ij} - \frac{\sum_i M_{ij}}{N}}{\sigma_j} \right), \quad (18)$$

where σ_j is the standard deviation of the entries in the j 'th column of M . Lastly, combine the previous two positive Z-scores into a CLR pseudo Z-score, as:

$$z(x_i, x_j) = \sqrt{z_i(x_i, x_j)^2 + z_j(x_i, x_j)^2}. \quad (19)$$

We have computed the pseudo z-scores in three variations:

CLR: We have applied CLR background correction to M^{stat} (7), resulting in a $N \times N$ matrix of CLR pseudo Z-scores,

$$Z^{\text{stat}} = \begin{pmatrix} z(x_1, x_1) & z(x_1, x_1) & \cdots & z(x_1, x_N) \\ z(x_2, x_1) & z(x_2, x_2) & \cdots & z(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ z(x_N, x_1) & z(x_N, x_2) & \cdots & z(x_N, x_N) \end{pmatrix}. \quad (20)$$

dynamic-CLR: We have applied CLR background correction to M^{dyn} (16), resulting in a $N \times N$ matrix of dynamic-CLR pseudo Z-scores,

$$Z^{\text{dyn}} = \begin{pmatrix} z(y_1, x_1) & z(y_1, x_2) & \cdots & z(y_1, x_N) \\ z(y_2, x_1) & z(y_2, x_2) & \cdots & z(y_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ z(y_N, x_1) & z(y_N, x_2) & \cdots & z(y_N, x_N) \end{pmatrix}. \quad (21)$$

Mixed-CLR, using dynamic and static MI values: Here, we propose and describe a CLR background correction that is based in part on M^{stat} , and in part on M^{dyn} . The motivation to

use dynamic MI values is that they may be more appropriate to resolve true regulatory interactions from spurious dependencies. Although dynamic MI values may reduce false dependencies in the data, they will not completely remove them. The expected distribution of false or indirect dependencies is best represented by the static MI values. For these reasons we decided to evaluate a mixed (dynamic + static) CLR procedure.

To apply this procedure we first, as was done for dynamic-CLR above, compute the Z-score of M_{ij}^{dyn} with respect to the entries in the i 'th row of M^{dyn} , i.e.

$$z'_i(y_i, x_j) = \max \left(0, \frac{M_{ij}^{\text{dyn}} - \frac{\sum_j M_{ij}^{\text{dyn}}}{N}}{\sigma_i} \right), \quad (22)$$

where σ_i is the standard deviation of the entries in the i 'th column of M^{dyn} .

Second, we compute the Z-scores of M_{ij}^{dyn} with respect to the background distribution of MI entries in the j 'th column of M^{stat} , i.e.

$$z'_j(y_i, x_j) = \max \left(0, \frac{M_{ij}^{\text{dyn}} - \frac{\sum_i M_{ij}^{\text{stat}}}{N}}{\sigma_j} \right), \quad (23)$$

where σ_j is the standard deviation of the entries in the j 'th column of M^{stat} . Note that z'_j compares the dynamic MI value with the observed distribution of static MI values, and in order for this background correction step to be effective, it assumes that both dynamic and static MI values are in the same range (this was a wrong assumption as later we show that, at least for the DREAM3 100-gene networks, static MI values are in general larger than dynamic MI values).

Lastly, we combine the previous two Z-scores into a pseudo Z-score, $z'(y_i, x_j)$, as described in (19), resulting in a $N \times N$ matrix of mixed-CLR Z-scores,

$$Z^{\text{mix}} = \begin{pmatrix} z'(y_1, x_1) & z'(y_1, x_2) & \cdots & z'(y_1, x_N) \\ z'(y_2, x_1) & z'(y_2, x_2) & \cdots & z'(y_2, x_N) \\ \vdots & \ddots & \vdots & \\ z'(y_N, x_1) & z'(y_N, x_2) & \cdots & z'(y_N, x_N) \end{pmatrix}. \quad (24)$$

Note that Z^{stat} is symmetric, and thus can not be used to resolve directionality of regulatory interactions, while Z^{dyn} , and Z^{mix} are not symmetric.

In order to decide which CLR variant to use for DREAM3 predictions, we evaluated the performance of all three CLR variants, on the two DREAM2 50-gene networks, applying the top method from this test to the DREAM3 challenge. Mixed-CLR outperformed both CLR and dynamic-CLR. Thus, our matrix of confidence scores after step one is:

$$S^1 = Z^{\text{mix}}. \quad (25)$$

The regulatory interactions scored in S^1 , when ranked from high to low, represent our ranking for the regulatory interaction list

after step one. Note, S^1 sets the confidence scores of many regulatory interactions to zero, thus removing them from consideration, while ranking the remaining interactions.

Step 2: Using Genetic Perturbations to Remove Least Likely Regulatory Interactions from Consideration

For our second step we perform crude filtration to remove the most unlikely regulatory interactions given our knowledge of gene knock-outs and knock-downs. This step is solely based on the genetic perturbations collected as steady-state observations.

For each interaction, $x_j \rightarrow x_i$, we compute the relative change in mRNA level of x_i when x_j is knocked out:

$$r(x_j \rightarrow x_i | x_j(-/-)) = \frac{|x_i(w.t.) - x_i(x_j(-/-))|}{x_i(w.t.)}. \quad (26)$$

Similarly, we compute the relative change in mRNA level of x_i when x_j is knocked down:

$$r(x_j \rightarrow x_i | x_j(-/+)) = \frac{|x_i(w.t.) - x_i(x_j(-/+))|}{x_i(w.t.)}. \quad (27)$$

Given a cutoff, $c \in [0, 1]$, we filter out an interaction, $x_j \rightarrow x_i$, iff $r(x_j \rightarrow x_i | x_j(-/-)) < c$ AND $r(x_j \rightarrow x_i | x_j(-/+)) < c$. In other words, we filter out a regulatory interaction, $x_j \rightarrow x_i$, if a large drop in expression levels of x_j have only resulted in a smaller than $c\%$ change in expression levels of x_i . For every regulatory interaction, $x_j \rightarrow x_i$, that was filtered this way, we have set $s_{ij} = -1$. The actual value of -1 does not matter, as negative scores are sent to the end of the ranked regulatory interaction list and not considered further. We denote the matrix of confidence scores, S^1 , after applying filtration as S^2 .

The regulatory interactions scored in S^2 , when ranked from high to low, represent our ranking for the regulatory interaction list after step two. We now apply the final step of our procedure, an ODE-based constrained linear regression approach—Inferelator 1.0.

Step 3.a: Inferelator 1.0

Here we use the results of the previous two steps, contained in S^2 , to remove low ranked regulatory interactions from consideration by Inferelator 1.0 [25], improving overall model selection performance. Furthermore, we want to force Inferelator 1.0 to consider only high confidence regulatory interactions (i.e. high rank regulatory interactions), strengthening the connection between mix-CLR and Inferelator 1.0. Thus, as possible regulators (explanatory variables) of x_i , we consider the P highest confidence regulators from S^2 , i.e. the P x_j 's corresponding to the highest strictly-positive s_{ij} 's, where $j = 1, \dots, N$. We denote $P^i \leq P$ to represent the actual number of regulators chosen, as in general a target gene, x_i , may have less than P regulators with $s_{ij} > 0$. We denote this x_i specific subset of likely regulators as $x^i(t)$.

We use Inferelator 1.0 to learn a sparse ODE model for each $x_i(t)$ as a function of $x^i(t)$ by assuming that the time evolution in the x_i 's is governed by

$$\frac{dx_i(t)}{dt} = -\alpha_i x_i + \sum_{j=1}^{P^i} \beta_{ij} x_j^i(t), \quad i = 1, \dots, N \quad (28)$$

which is exactly (8) with the modification that we only consider a subset of regulators (high confidence ones) for each target gene.

Least Angle Regression (LARS) [32] is used to efficiently implement an l_1 constraint [31] on β , which minimize the following objective function, amounting to a least-square estimate based on the ODE (28):

$$\mathcal{E}(\beta) = \sum_{i=1}^N \mathcal{E}_i(\beta) \quad (29)$$

where

$$\mathcal{E}_i(\beta) = \sum_{k=1}^{K-1} \left| \frac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k} + \alpha_i x_i(t_k) - \sum_{j=1}^{P^i} \beta_{ij} x_j^i(t_k) \right|^2 \quad (30)$$

under an l_1 -norm penalty on regression coefficients,

$$\sum_{j=1}^{P^i} |\beta_{ij}| \leq s \sum_{j=1}^{P^i} |\beta_{ij}^{\text{ols}}| \quad (31)$$

where β^{ols} is the over-fit ordinary least-squares estimate (i.e. the minimizer of (30) with no penalty), and s is a number between 0 and 1 referred to as the shrinkage parameter; setting $s=1$ corresponds to ordinary least-square regression. Note that, as before, we use steady-state observations by setting the first term under the summation on the right-hand-side of (30) to zero.

Ten fold cross validation is used to select the minimum value of s that results in models with good generalization, i.e. good predictive performance on new data. Each resulting model is then an ODE describing the time evolution of $x_i(t)$. The full set of models, one for each target, constitutes the full network model. For Inferelator 1.0 we have assumed that each target gene has no more than ten regulators, i.e. we have chosen $P=10$ corresponding to $P_i \leq 10$ in (28). This assumption turned out to be very wrong for one out of the five 100-gene networks (Yeast3), which had genes that were regulated by as many as 24 regulators.

To produce the ranks required by the challenge we combine the Inferelator 1.0 model weights (β) with the mixed-CLR measures of confidence (S^2) using a simple heuristic designed to give each method roughly equal influence. We describe this heuristic in the following two sub-sections.

Step 3.b: Converting Inferelator 1.0 Weights into Confidence Scores. Regulatory interactions that were supported by mixed-CLR and not filtered out all have corresponding confidence scores $s_{ij}^2 > 0$, in S^2 . The previous, Inferelator 1.0, step gave us a sparse matrix, β , with a small number of entries $|\beta_{ij}| > 0$, chosen from the regulatory interactions with $s_{ij}^2 > 0$.

To ensure that Inferelator 1.0 confidence scores are on equal footing with the previous confidence scores, stored in S^2 , we first assigned all Inferelator 1.0 weights to S^3 , i.e. $S^3 = \beta$, and then replaced the non-zero values (weights) in S^3 with a corresponding confidence scores of equal rank in S^2 . For example, the regulatory interaction with the highest absolute value, $|s_{ij}^3|$ in S^3 , was assigned the highest value from S^2 , while the interaction with the second highest absolute value in S^3 , was given the second highest value from S^2 . We continued in such a way until we assign a confidence

score from S^2 to each interaction in S^3 that had a non-zero weight.

Step 3.c: Combining Results from Mixed-CLR and Inferelator 1.0 to Produce Final Ranks. We store our final confidence scores for regulatory interactions that were supported by mixed-CLR and Inferelator 1.0, in S , with every entry s_{ij} equal to

$$s_{ij} = \sqrt{(s_{ij}^2)^2 + (s_{ij}^3)^2} \quad i=1,2,\dots,N \quad j=1,2,\dots,N. \quad (32)$$

Note that most confidence scores in S^3 equal zero (the Inferelator 1.0 weight was zero) and thus have no effect on the final confidence scores. This step can be considered as re-organization (pushing up the ranking list) of regulatory interactions with $s_{ij}^2 > 0$ that also had an Inferelator 1.0 model weight $|\beta_{ij}| > 0$.

The regulatory interactions scored in S , when ranked from high to low, represents our final ranking for the regulatory interaction list. It is given that auto-regulatory interactions do not exist in the DREAM3 challenge networks, thus we have not considered auto regulatory interactions, $x_i \rightarrow x_i$, for all i .

We have implemented all the steps in our pipeline using the R statistical language [40]. Code is freely available from the authors upon request.

Comparing Results to Inferelator 1.0 alone

Inferelator 1.0 as previously described in [25] used a dynamic correlation matrix, similar to M^{dyn} (16) (with the only difference being that dynamic correlation was used as a measure of similarity instead of dynamic mutual information), to initially choose high confidence regulators. In order to compare performance of Inferelator 1.0 alone to the pipeline described above, we also computed the correlation between every dynamic pair, $\text{cor}(y_i; x_j)$ (see equations (14) and (15)), and stored the values in the form of a $N \times N$ matrix, C^{dyn} . We then performed step 3.a using C^{dyn} instead of S^2 and ranked chosen regulatory interactions, i.e. regulatory interactions corresponding to $|\beta_{ij}| > 0$, based on absolute value weights. The resulting ranked list was used to evaluate the performance of Inferelator 1.0 alone.

Judging Performance

After a network inference method suggests potential regulatory interactions, validation of these interactions typically requires significant effort (often requiring the coordination of multiple experiments). Hence, a regulatory network inference method should ideally produce a small number of false positives (FP) even at the expense of a higher false negative (FN) rate. When testing such a method, the performance metric should be sensitive to the method's ability to avoid FPs. Therefore, throughout this section we used area-under-curve of precision ($\frac{TP}{TP+FP}$) vs. recall ($\frac{TP}{TP+FN}$) plot, where TP stands for true positives, as a measure of performance, since it degrades quickly with FPs.

Results

Mixed-CLR and Inferelator 1.0 Proved Complimentary, Outperforming Other Methods and Combinations of Methods

We used the DREAM2 50-gene data for testing our pipeline prior to the DREAM3 100-gene challenge. On both this pre-competition data and the actual DREAM3 data, Mixed-CLR with Inferelator 1.0 outperformed other potential pipelines we evaluated, and was thus the method we initially used for the DREAM3 competition. From

Performance

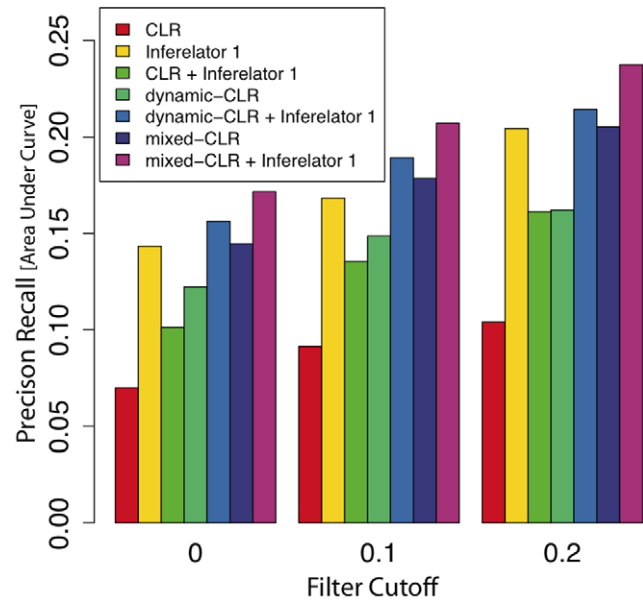


Figure 2. Mean area-under precision vs. recall curves for DREAM3 five 100-gene networks. We evaluated the performance of Inferelator 1.0 and three different versions of CLR—namely: original-CLR (CLR), dynamic-CLR, and mixed-CLR—with or without Inferelator 1.0, at three levels of knock-out filtration, $c=0,0.1,0.2$. To make DREAM3 predictions we used mixed-CLR with Inferelator 1.0 (with filtration cutoff $c \approx 0.1$), resulting in area-under precision vs. recall curve of 0.20 (p-value, 10^{-56}), and area-under receiver operating characteristic curve of 0.78 (p-value, 10^{-36}). We show that the pipeline we used to make DREAM3 predictions produced optimal performance, compared to other tested CLR/Inferelator 1.0 combinations. Error bars for methods involving Inferelator 1.0 (variability due to cross validation) are approximately within 1% of Precision vs. Recall area-under-curve values and are thus not shown. doi:10.1371/journal.pone.0009803.g002

Figure 2 we can see that: 1) mixed-CLR outperformed dynamic-CLR and CLR, regardless of filtration cutoff (for the DREAM3 networks we used a mean filtration cutoff of $c \approx 0.1$, so as to filter approximately one third of all regulatory interactions for each network); 2) our simple knock-out filtration step boosted performance of any method combination we tested but did not alter the performance ranks of the methods tested for any cutoff value; and 3) Mixed-CLR and Inferelator 1.0 are complimentary, providing superior performance when compared to each method alone.

The same trend, in which mixed-CLR coupled with Inferelator 1.0 outperforms the other evaluated method combinations for a large range of tested filtration cutoffs, holds for DREAM3 50-gene networks (for which our method ranked 4th out of 27) and 10-gene networks (for which our method ranked 5th out of 29) (data not shown). As for the DREAM3 50-gene networks, our pipeline did not outperform the DREAM2 50-gene challenge best performers [27–29,42] (data not shown). Note that our method is based in part on computing z-scores. As network (or system) size decreases, our estimates for underlying probability distributions decreases as well, making our z-score estimates crude. This perhaps explains the decline in performance (relative to other participating methods in DREAM3) for the smaller networks in this challenge.

For DREAM3 100-Gene Networks, Knock Out Observations Contributed Most to Performance

One important question that the DREAM initiative aims to answer is what data sets are most useful for characterizing

regulatory network. We compared the performance of five methods (CLR, mixed-CLR, Inferelator 1.0, and mixed-CLR or CLR with Inferelator 1.0) over four partitions of the full range of provided experiments, namely: knock-down, knock-out, time-series, and the former three combined. From Figure 3 we can see that: 1) the dynamical methods, mixed-CLR and Inferelator 1.0 were more powerful at utilizing time-series observations than the static method CLR; 2) for all dataset partitions tested, mixed-CLR and Inferelator 1.0 proved complimentary and had optimal performance; and 3) for all tested methods, knock-out data (100 observations) was most instrumental for learning the regulatory networks, followed by time-series and knock-down data (966 and 100 observations, respectively).

Inferelator 1.0 and Use of Knock-out Information Effectively Resolved Causation

Determining causation (the directionality of regulatory interactions) is one of the tougher problems to solve when inferring regulatory networks. In practice, *a priori* knowledge is often used to suggest which genes are regulating a given target or target set (for example knowing that one gene codes for an enzyme and one for a transcription factor gives us the ability to resolve directionality). However solving for the directionality between pairs of regulators remains a critical challenge. It could be argued, for example, that

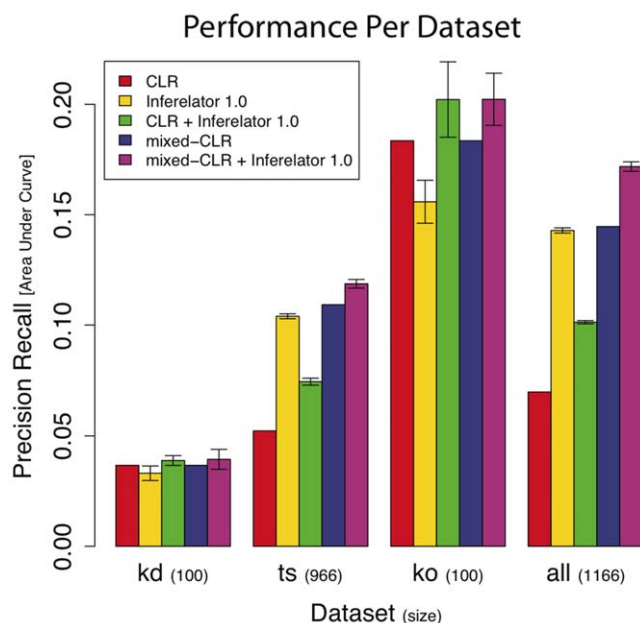


Figure 3. Performance as a function of data set used. We evaluated the contribution of each data set (namely: knock-down ('kd'), time-series ('ts'), knock-out ('ko'), and all three combined ('all')) to performance of CLR, mixed-CLR, Inferelator 1.0, and CLR or mixed-CLR with Inferelator 1.0 (no filtration was used, $c=0$). Note, mixed-CLR is a generalization of CLR that takes advantage of time-series data, when time-series data is not used (i.e. for 'kd' and 'ko') the two are equivalent. For all tested methods 'ko' data contributes the most to performance (followed by 'ts' and 'kd' data respectively). The inclusion of a dynamical model allowed mixed-CLR and Inferelator 1.0 to take advantage of 'ts' data (compare to CLR above 'ts' and 'all' data partitions). Mixed-CLR and Inferelator 1.0 are complimentary, as evidenced by the improvement in performance when the two methods are combined. For 'ts', 'ko', and 'all' data partitions, mixed-CLR with Inferelator 1.0, the method we used to make predictions for DREAM3, gave optimal performance. Error bars for methods involving Inferelator 1.0 are drawn at one standard deviation (estimated from ten Inferelator 1.0 runs). doi:10.1371/journal.pone.0009803.g003

determining regulatory interactions between pairs of regulators is a more important problem than resolving other regulatory interactions, as interactions between regulators are key to the cell's ability to process and integrate information.

We compared the relative merit of five methods (CLR, mixed-CLR, Inferelator 1.0, and mixed-CLR or CLR with Inferelator 1.0) with or without knock-out filtration to determine causation. From Figure 4 we can see that: 1) Out of the five methods, Inferelator 1.0 best resolved causation ($\approx 93\%$); 2) mixed-CLR had some power at resolving causation when compared to the static version of the algorithm; and 3) removal of unlikely regulatory interactions based on the knock-out filtration, was very useful for resolving causation and complimentary to the other methods we tested.

Performance Degrades with Increasing Network In-Degree

Biological regulatory networks are typically sparse, i.e. they have a relatively small number of regulatory edges when compared to the total number of possible edges. Network sparsity is commonly used to glean at what the dynamic complexity of that network would be if it could be simulated or observed (where the more

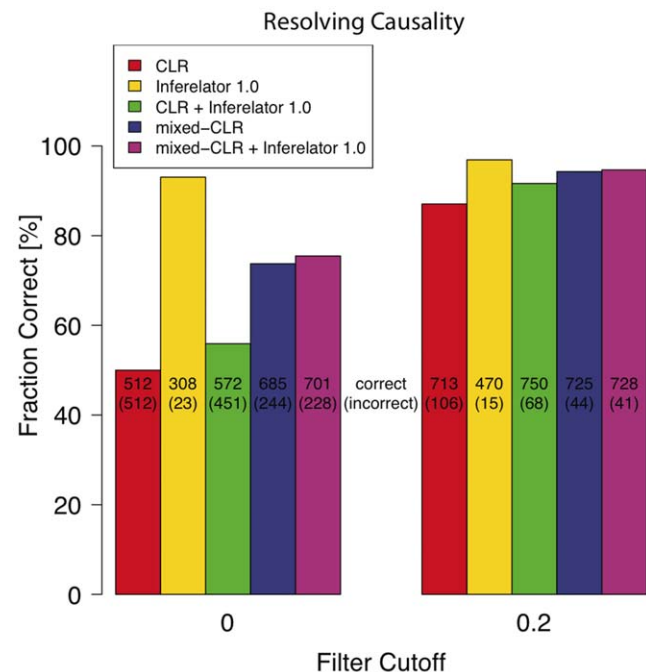


Figure 4. Resolving causation of regulatory interactions. We present the relative merit of five methods, with and without knock-out filtration, to resolve causation (i.e. directionality of regulatory interactions). For each method we computed the fraction of correctly resolved true regulatory interactions (true positives, TPs) out of the total number of TPs the method had identified. We define a TP interaction, $x_j \rightarrow x_i$, as correctly resolved, if its score, $s_{i,j}$ (according to each method or method combination), was bigger than the confidence score of the reverse (false) regulatory interaction, $s_{j,i}$. The original CLR method without filtration results in symmetric confidence scores, $s_{i,j} = s_{j,i}$, and thus cannot resolve causation (fraction correct = 50%). In each bar plot we report the absolute number of correctly (incorrectly) resolved interactions. We show that, without filtration, Inferelator 1.0 has the most power at resolving causation ($\approx 93\%$ correct), and that for all methods knock-out filtration helps resolve causation. For Inferelator 1.0 filtration helps recover more TPs. Error bars for methods involving Inferelator 1.0 are less than 1% and are not shown. doi:10.1371/journal.pone.0009803.g004

sparse a network is, the simpler its dynamic behaviour becomes). Network sparsity in turn can be separated into two more detailed measures: network in-degree distribution, derived from the distribution of regulatory edges entering each target gene, and network out-degree distribution, derived from the distribution of regulatory edges leaving each regulator. Each distribution when summed equals to the number of regulatory edges in the network. We find that, as expected, our method's median error increases with genes median in-degree (see Figure 5) ($R^2=0.976$), i.e. performance drops for targets under the control of many regulators, but interestingly is not correlated to median out-degree ($R^2=0.017$), i.e. performance does not drop for regulators controlling many target genes.

For DREAM3 100-Genes Networks, Mixed-CLR Did Not Effectively Correct for Background

One unexpected problem with mixed-CLR (that the DREAM3 challenge revealed) is that we have no guarantee that static and dynamic MI values will be in the same range for a given data set (which we assumed when constructing mixed-CLR). Indeed, from Figure 6 we can see that the majority of dynamic-MI values were below the mean static MI value. Since for background correction mixed-CLR computed the positive z-score of each regulatory interaction's dynamic MI value, assuming it was taken from the distribution of static MI values, most of these z-scores ended up being zero. Thus, for DREAM3 100-gene networks mixed-CLR in practice was the result of determining z-scores for each regulatory interaction based on the dynamic MI values alone (dynamic MI z-scores).

Also, we can see from Figure 6 that the dynamic MI distribution had a smaller standard deviation (2.68) than the static MI distribution (3.57), possibly making it easier to resolve true regulatory interactions from false regulatory interactions.

Dynamic MI Identified True Regulatory Interactions Better Than Static MI

We hypothesized that dynamic-MI will decrease false statistical dependencies between gene pairs (i.e. dependencies that are not due to direct regulatory interactions), assisting in the identification of true regulatory interactions. To test this hypothesis we computed MI between the expression levels of every gene pair (static MI), and between every pair of dynamic response and explanatory-variable (dynamic MI). For both static and dynamic MI values, we computed a z-scores for each true regulatory interaction (true positive, TP) and false regulatory interaction (true negative, TN) by assuming its MI value is taken from the distribution of MI values involving the target in that interaction, i.e. the first z-score from dynamic-CLR or mixed-CLR. Indeed, from Figure 7 we can see that TPs are better separated from TNs by dynamic MI z-scores than by static MI z-scores.

Top Ranked Predictions are Largely Correct

As mentioned previously, in biology it is desired that methods have high precision even in the expense of recall (completeness). Here we take a look at precision for several recall values ranging from low to high recall (2%–50%). We show in table 1 the performance for the method's best predicted network, Ecoli 2, and in table 2 the performance for the method's worst predicted network, Yeast 3. Our full pipeline produced good precision results, especially for the lower recall values. The boost in performance by Inferelator 1.0, that seems to be mostly apparent in the lower recall values, is probably confounded by an overly strong sparsity penalty at higher recall values for more complex models, i.e. Inferelator 1.0 identifies interactions with a high precision, but it seems to be too parsimonious to identify a substantial portion of the true regulatory interactions for high in-degree networks, e.g. the Yeast 3 network (where target genes

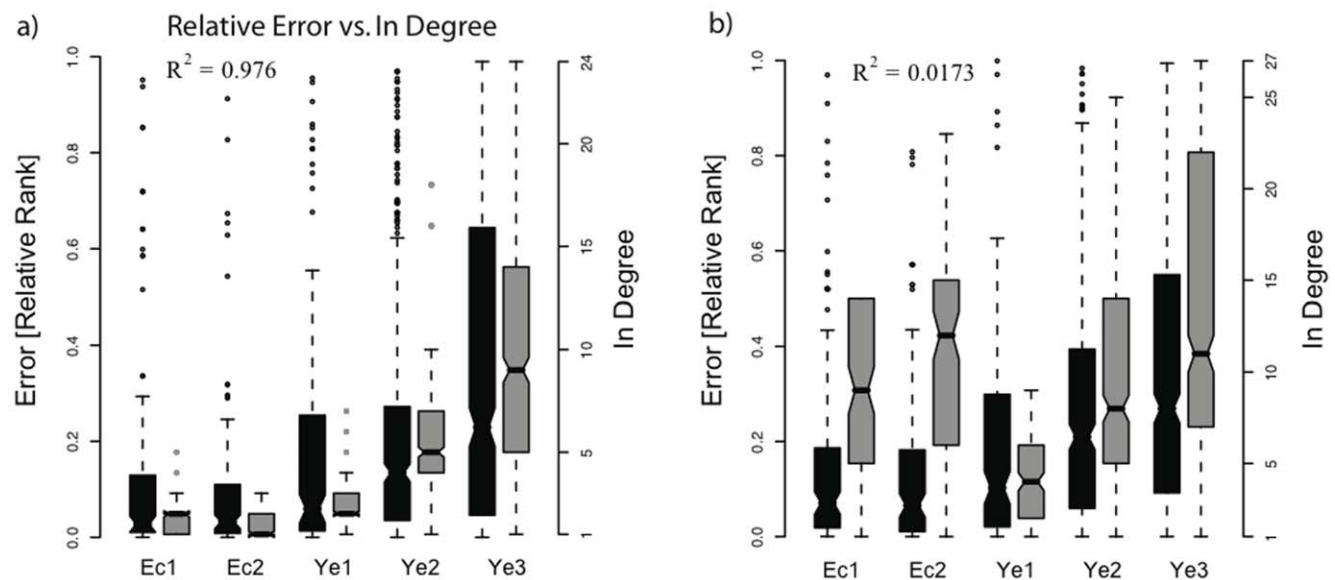


Figure 5. Error as a function of gene in degrees and gene out degrees. Here we evaluate the performance of mixed-CLR, filtration cutoff of $c=0.1$, and Inferelator 1.0—the pipeline we applied to make DREAM3 predictions. Box plots for error distributions for each of the five predicted networks are shown in black in both panels, gray box plots show in-degree and out-degree distributions for **a.** and **b.** respectively. We estimated error in the following manner: Denote by L the total number of possible regulatory interactions, and by l the rank we gave to a regulatory interaction, $x_j \rightarrow x_i$, the relative rank (error) of $x_j \rightarrow x_i$ is defined to be $\frac{l}{L}$. **a)** Median relative rank (Error) increases as the networks' median in-degree increases ($R^2 = .976$). **b)** Median relative rank (Error) is not correlated with median out-degree ($R^2 = .0173$). doi:10.1371/journal.pone.0009803.g005

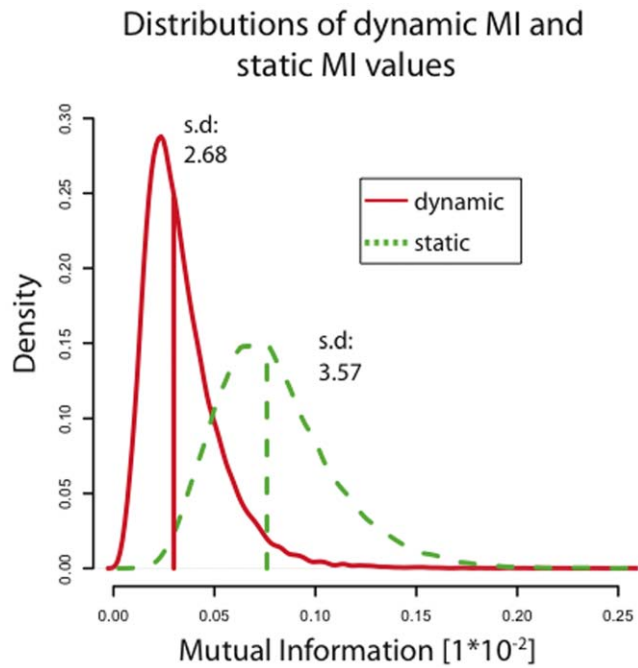


Figure 6. Distributions of static vs. dynamic mutual information values. We computed static and dynamic Mutual Information (MI) values for every possible regulatory interaction. Vertical lines represent distribution means. We present the combined probability densities for the five 100-gene networks. We show that: 1) both dynamic and static MI densities are right skewed, consistent with the assumption that MI values of true positives would be higher than MI values of true negatives; 2) the standard deviations for static MI z-scores, 3.57, is larger than for dynamic MI z-scores, 2.68, possibly making it easier to recover TPs from the dynamic MI z-scores; and 3) most dynamic MI values are smaller than the mean of the static MI values; this shift confounds mixed-CLR. Note that both static- and dynamic-MI values were estimated from the same number of observations, using the same number of bins. Thus, dataset size or bin number differences do not explain the shift in distributions. doi:10.1371/journal.pone.0009803.g006

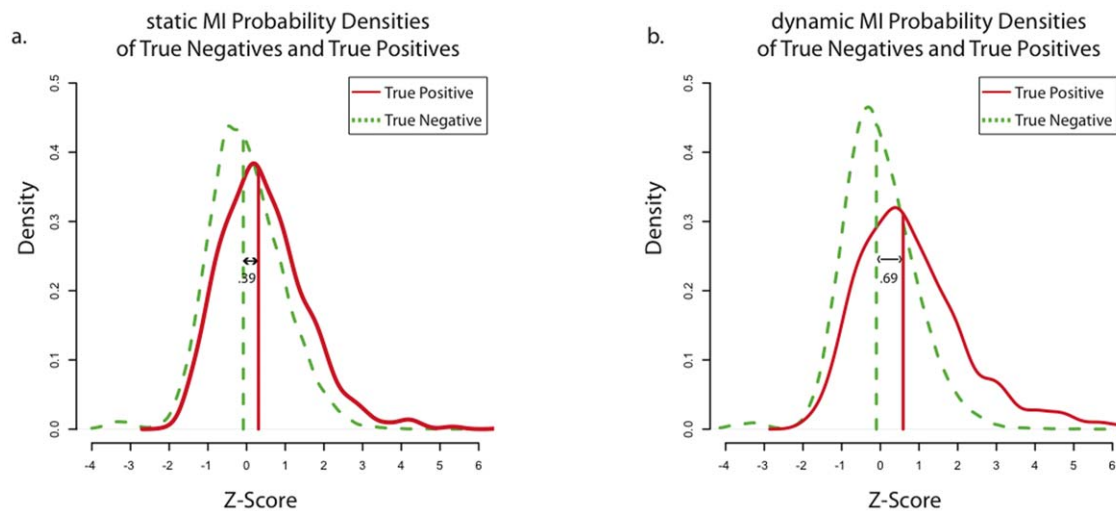


Figure 7. Probability densities of static and dynamic mutual information values for true positive and true negative regulatory interactions. We computed static and dynamic Mutual Information (MI) values for every possible regulatory interaction for all five 100-gene networks. For both static and dynamic MI values, we computed z-scores for true regulatory interactions (true positives, TPs) and false regulatory interactions (true negatives, TNs). We present the static (a.) and dynamic (b.) z-scores densities (combined over the five 100-gene networks) for TPs (red) and TNs (green). Vertical lines represent median z-scores. We show that TPs are better separated from TNs by the dynamic MI z-scores, consistent with the improved performance of mixed- and dynamic-CLR. doi:10.1371/journal.pone.0009803.g007

exhibit in-degrees exceeding 5–10, see Figure 5). Nevertheless our high accuracy in these low-recall settings is well matched to typical biological laboratory settings.

Discussion

We have shown that explicitly modeling dynamics using a simple ODE model increases the ability of our pipeline to identify true regulatory interactions (when compared to a static model), and help resolve the directionality of these interactions. Specifically, analysis of our performance on the DREAM3 100-gene networks show that: 1) the full pipeline (mixed-CLR followed by, knock-out filtration and Inferelator 1.0) outperformed other tested combinations of dynamic and static methods (Figure 2); 2) knock out data was instrumental for learning regulatory interactions (Figure 3); 3) Inferelator 1.0 was instrumental for resolving regulatory causation ($\approx 93\%$ of identified regulatory interactions were correctly resolved, Figure 4). 4) mixed-CLR and Inferelator 1.0 proved complimentary (Figure 2 and 3); and 5) dynamic MI values (mixed-CLR) separated true regulatory interactions from false, but otherwise dependent, pair-wise interactions better than static MI values (Figure 7).

We observed a drop in performance as the median in-degree of a network increases (Figure 5.a). This is to be expected and could be due to many reasons, including: 1) the dynamic behavior of a target gene becomes more condition-dependent as the number of regulators increases, fragmenting the data set among distinct conditions, and making it harder to resolve regulatory interactions from expression data no matter the method used (dynamic or static). This is supported by the observation that any method combination we have tested under-performed on the high in-degree networks (e.g. table 2), compared to low in-degree networks (e.g. table 1); 2) the model we have used was too simple to describe the dynamic behaviour of high in-degree target genes; 3) with Inferelator 1.0 we have imposed an l_1 constraint on model weights (i.e. a constraint on in-degree) that may have been too restrictive for high in-degree target genes; and 4) with this use of the Inferelator 1.0 we enforced a strict ten predictor cutoff that proved

Table 1. Network 2 (*E.coli*-2): Methods precision for low-to-high recall values.

	2%	5%	10%	20%	30%	40%	50%
mixed-CLR+Inferelator 1.0	100	100	100	75	39	12	10
mixed-CLR	100	100	92	48	17	13	9
dynamic-CLR+Inferelator 1.0	100	100	86	49	29	13	7
dynamic-CLR	75	86	75	42	17	7	6
CLR+Inferelator 1.0	75	86	71	29	17	8	5
Inferelator 1.0	100	100	100	69	14	7	5
CLR	75	60	29	16	6	4	4

In this table we present a more detailed view of performance for our method's best predicted network (119 total regulatory interactions with up to 3 regulators controlling each gene). The table inline method precision [%] at varying degrees of completeness (recall [%]).
doi:10.1371/journal.pone.0009803.t001

too stringent for two of the 100-gene networks, which had a significant number of target genes under the control of more than ten regulators.

Interestingly, we have not observed a similar drop in performance as networks median out-degree increased (Figure 5.b). In principle, as a networks' median out degree increases one expects that the number of indirect regulatory interactions (mediated through regulators of regulators) will increase, and with it the underlying complexity of the system's dynamic behaviour. However, there may be many reasons why this was not observed, including: 1) a change in a regulator's mRNA, followed by a corresponding change in a target gene's mRNA requires a time delay (note that for the DREAM3 *in silico* challenges mRNA and protein levels were modeled, albeit observations were only given for mRNA). This time delay will increase for indirect regulatory interactions. Since our method uses consecutive observations (here sampled every 20 minutes) a change in an indirect regulator's mRNA levels may not have the time to effectively propagate to its indirect targets. In other words, the observations were sampled finely enough to make direct regulatory interactions resolvable from indirect ones; 2) our approach was centered around the target gene; we modeled the change in rate of expression of each gene separately, i.e. we assumed an un-coupled system of ODEs. Thus our model complexity is largely determined by the number of regulators a gene has (in-degree), but not by number of targets a regulator has (out-degree); and 3) two (out of five) of the networks, the ones responsible for the observed lack of correlation between out-degree

and performance, were based on *E.coli*'s topology. There is evidence that for *E.coli* the number of indirect regulatory interactions (and thus complexity) is much smaller than expected by its out-degree distribution [43,44], and that its transcriptional network has primarily a feedforward structure, resulting in less complex dynamics due to the relative lack of feedback loops that would otherwise keep information propagating in the network (and thus increase complexity) [43,44].

We learned that the Inferelator 1.0 l_1 -norm regularizer (LARS [31,32]) proved to be too parsimonious for the two most complex (in terms of target in-degree) 100-gene networks, leaving many true regulatory interactions out of the model. One limitation of using an l_1 constraint is that in cases where several explanatory variables are correlated (or anti-correlated), the procedure will tend to pick either one of them or none, potentially leading to overly sparse models. This suggests that using a method which is more robust to the "one or none" problem, such as the elastic-net [45] (an l_1 and l_2 norm constrained regression), will improve performance.

We were encouraged to see that even a very simple dynamical model was able to significantly increase performance (compared to static model) at identifying true regulatory interactions and resolving their causation. Moreover, the two dynamic methods mixed-CLR and Inferelator 1.0 proved complimentary.

Knock-out observations were instrumental for characterizing the DREAM3 100-gene regulatory networks (Figure 3). This is in line with our observation that even the crude filter we used (based in part on knock-out data) to remove the least likely regulatory

Table 2. Network 5 (*Yeast*-3): Methods precision for low-to-high recall values.

	2%	5%	10%	20%	30%	40%	50%
mixed-CLR+Inferelator 1.0	92	51	34	20	13	12	10
mixed-CLR	75	50	31	18	13	12	10
dynamic-CLR+Inferelator 1.0	75	51	36	20	17	12	10
dynamic-CLR	67	36	25	20	15	12	10
CLR+Inferelator 1.0	57	34	18	15	12	10	9
Inferelator 1.0	86	54	25	5	5	5	5
CLR	28	22	17	13	12	10	9

In this table we present a more detailed view of performance for our method's poorest predicted network (551 total regulatory interactions with up to 24 regulators controlling each gene). The table inline method precision [%] at varying degrees of completeness (recall [%]).
doi:10.1371/journal.pone.0009803.t002

interactions proved very effective in identifying true regulatory interactions (Figure 2) and resolving their direction (Figure 4). Importantly, when knock-out data was used alone, all tested methods achieved optimal performance (Figure 3). Furthermore, only 100 knock-out observations were needed, compared to the 966 provided time-series observations (46 time-series experiments, each containing 23 observations). In our previous works we assumed that Inferelator 1.0 would implicitly use genetic information (e.g. knock-out data) by incorporating the steady-state data into the learning procedure, hence not requiring explicit constraints to be derived from genetic perturbations. The DREAM3 results suggest that we need to develop better explicit methods to incorporate constraints from such genetic perturbations into Inferelator 1.0 or similar methods. However, it is typically not possible to obtain knock-out information for each gene in such a comprehensive manner. Even when knock-out information can be obtained, the knocked out gene may not be active under the “wild type” conditions, thus not revealing any regulatory information. Therefore, it will prove helpful to also incorporate other types of constraints, for example constraints derived from TF-DNA binding experiments such as ChIP-chip [46] and ChIP-seq [47,48].

References

- Stolovitzky G, Prill RJ, Califano A (2009) Lessons from the dream2 challenges. *Annals of the New York Academy of Sciences* 1158: 159–195.
- Stolovitzky G, Monroe D, Califano A (2007) Dialogue on reverse-engineering assessment and methods: the dream of high-throughput pathway inference. *Ann N Y Acad Sci* 1115: 1–22.
- de Jong H (2002) Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology* 9: 67–103.
- Alm E, Arkin AP (2003) Biological networks. *Current Opinion in Structural Biology* 13: 193–202.
- Herrgard MJ, Covert MW, Palsson BO (2004) Reconstruction of microbial transcriptional regulatory networks. *Curr Opin Biotechnol* 15: 70–77.
- Hayete B, Gardner TS, Collins JJ (2007) Size matters: network inference tackles the genome scale. *Molecular systems biology* 3: 77.
- Bansal M, Belcastro V, Ambesi-Impombato A, di Bernardo D (2007) How to infer gene networks from expression profiles. *Molecular systems biology* 3.
- Bonneau R (2008) Learning biological networks: from modules to dynamics. *Nature chemical biology* 4: 658–664.
- Babu MM, Lang B, Aravind L (2009) Methods to reconstruct and compare transcriptional regulatory networks. *Methods Mol Biol* 541: 163–180.
- Madar A, Bonneau R (2009) Learning global models of transcriptional regulatory networks from data. *Methods Mol Biol* 541: 181.
- Butte AJ, Kohane IS, Kohane IS (2000) Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing* 5: 415–426.
- Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS (2000) Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks. *Proc Natl Acad Sci U S A* 97: 12182–12186.
- Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, et al. (2006) Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7 Suppl 1.
- Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, et al. (2007) Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology* 5: e8.
- Meyer PE, Kontos K, Lafitte F, Bontempi G (2007) Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J Bioinform Syst Biol.* 79879.
- Zhao W, Serpedin E, Dougherty ER (2008) Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Trans Comput Biol Bioinform* 5: 262–274.
- Watkinson J, Liang KC, Wang X, Zheng T, Anastassiou D (2009) Inference of regulatory gene interactions from expression data using three-way mutual information. *Ann N Y Acad Sci* 1158: 302–313.
- Chen KC, Csikasz-Nagy A, Gyorfy B, Val J, Novak B, et al. (2000) Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol Biol Cell* 11: 369–391.
- Cross FR, Archambault V, Miller M, Klovstad M (2002) Testing a mathematical model of the yeast cell cycle. *Mol Biol Cell* 13: 52–70.
- Monk NAM (2003) Oscillatory expression of hes1, p53, and nf-kappab driven by transcriptional time delays. *Curr Biol* 13: 1409–1413.
- Jaeger J, Blagov M, Kosman D, Kozlov KN, Myasnikova E, et al. (2004) Dynamical analysis of regulatory interactions in the gap gene system of drosophila melanogaster. *Genetics* 167: 1721–1737.
- Jaeger J, Surkova S, Blagov M, Janssens H, Kosman D, et al. (2004) Dynamic control of positional information in the early drosophila embryo. *Nature* 430: 368–371.
- Chang WC, Li CW, Chen BS (2005) Quantitative inference of dynamic regulatory pathways via microarray data. *BMC Bioinformatics* 6: 44.
- Wang Y, Joshi T, Zhang XS, Xu D, Chen L (2006) Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics* 22: 2413–2420.
- Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, et al. (2006) The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology* 7: R36.
- Wang RS, Wang Y, Zhang XS, Chen L (2007) Inferring transcriptional regulatory networks from high-throughput data. *Bioinformatics* 23: 3056–3064.
- Gustafsson M, Hörnquist M, Lundström J, Björkegren J, Tegnér J (2009) Reverse Engineering of Gene Networks with LASSO and Nonlinear Basis Functions. *New York Academy Sciences Annals* 1158: 265–275.
- Lauria M, Iorio F, di Bernardo D (2009) NIREst: A Tool for Gene Network and Mode of Action Inference. *New York Academy Sciences Annals* 1158: 257–264.
- Scheinine A, Mentzen WI, Fotia G, Pironi E, Maggio F, et al. (2009) Inferring Gene Networks: Dream or Nightmare? *New York Academy Sciences Annals* 1158: 287–301.
- Madar A, Greenfield A, Ostrer H, Vanden-Eijnden E, Bonneau R (2009) The inferelator 2.0: A scalable framework for reconstruction of dynamic regulatory network models. *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS.* pp 5448–5451.
- Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B (Methodological)* 58: 267–288.
- Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. *Ann Statist* 32: 407–499.
- Bonneau R, Facciotti MT, Reiss DJ, Schmid AK, Pan M, et al. (2007) A predictive model for transcriptional control of physiology in a free living cell. *Cell* 131: 1354–1365.
- Gilchrist M, Thorsson V, Li B, Rust AG, Korb M, et al. Systems biology approaches identify atf3 as a negative regulator of toll-like receptor 4. *Nature* 451: 1022.
- Friedman N, Nachman I, Pe'er D (1999) Learning bayesian network structure from massive datasets: The “sparse candidate” algorithm. pp 206–215.
- Marbach D, Schaffter T, Mattiussi C, Floreano D (2009) Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology* 16: 229–239.
- Shannon CE (1948) A mathematical theory of communication. *Bell system technical journal* 27.
- Cover TM, Thomas JA (1991) Elements of information theory. New York NY, USA: Wiley-Interscience. URL <http://portal.acm.org/citation.cfm?id=129837>.
- Daub CO, Steuer R, Selbig J, Kloska S (2004) Estimating mutual information using b-spline functions—an improved similarity measure for analysing gene expression data. *BMC bioinformatics* 5: 118.

40. R Development Core Team (2008) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>. ISBN 3-900051-07-0.
41. Bernstein JA, Khodursky AB, Lin PH, Lin-Chao S, Cohen SN (2002) Global analysis of mrna decay and abundance in escherichia coli at single-gene resolution using two-color fluorescent dna microarrays. *Proceedings of the National Academy of Sciences of the United States of America* 99: 9697–9702.
42. Gowda T, Vrudhula S, Kim S (2009) Prediction of Pairwise Gene Interaction Using Threshold Logic. *New York Academy Sciences Annals* 1158: 276–286.
43. Cosentino, Jona P, Bassetti B, Isambert H (2007) Hierarchy and feedback in the evolution of the escherichia coli transcription network. *PNAS* 104: 5516–5520.
44. Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics* 31: 64–68.
45. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B* 67: 301–320.
46. Horak CE, Snyder M (2002) Chip-chip: a genomic approach for identifying transcription factor binding sites. *Methods Enzymol* 350: 469–483.
47. Johnson DS, Mortazavi A, Myers RM, Wold B (2007) Genome-wide mapping of in vivo protein-dna interactions. *Science* 316: 1497–1502.
48. Robertson G, Hirst M, Bainbridge M, Bilenky M, Zhao Y, et al. (2007) Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing. *Nat Methods* 4: 651–657.