Incorporating Existing Network Information into Gene Network Inference

Scott Christley[1,2,3,4]*, **Qing Nie**[1,3,4], **Xiaohui Xie**[2,3,4,5]*

1 Department of Mathematics, University of California Irvine, Irvine, California, United States of America, **2** Department of Computer Science, University of California Irvine, Irvine, California, United States of America, **3** Center for Mathematical and Computational Biology, University of California Irvine, Irvine, California, United States of America, **4** Center for Complex Biological Systems, University of California Irvine, Irvine, California, United States of America, **5** Institute for Genomics and Bioinformatics, University of California Irvine, Irvine, California, United States of America

Abstract

One methodology that has met success to infer gene networks from gene expression data is based upon ordinary differential equations (ODE). However new types of data continue to be produced, so it is worthwhile to investigate how to integrate these new data types into the inference procedure. One such data is physical interactions between transcription factors and the genes they regulate as measured by ChIP-chip or ChIP-seq experiments. These interactions can be incorporated into the gene network inference procedure as a priori network information. In this article, we extend the ODE methodology into a general optimization framework that incorporates existing network information in combination with regularization parameters that encourage network sparsity. We provide theoretical results proving convergence of the estimator for our method and show the corresponding probabilistic interpretation also converges. We demonstrate our method on simulated network data and show that existing network information improves performance, overcomes the lack of observations, and performs well even when some of the existing network information is incorrect. We further apply our method to the core regulatory network of embryonic stem cells utilizing predicted interactions from two studies as existing network information. We show that including the prior network information constructs a more closely representative regulatory network versus when no information is provided.

Citation: Christley S, Nie Q, Xie X (2009) Incorporating Existing Network Information into Gene Network Inference. PLoS ONE 4(8): e6799. doi:10.1371/journal.pone.0006799

Editor: Cathal Seoighe, National University of Ireland Galway, Ireland

Received May 6, 2009; **Accepted** July 27, 2009; **Published** August 27, 2009

Copyright: © 2009 Christley 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: This work was partially supported by National Institutes of Health grants R01GM75309, R01GM67247 and P50GM76516 to QN, and National Science Foundation grant DBI-0846218 to XX. 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: scott.christley@uci.edu (SC); I xhx@uci.edu (XX)

Introduction

Considerable progress has been obtained in the ability to infer gene regulatory networks from gene expression data. The three primary methodologies include probabilistic graphical models [1], information-theoretic approaches [2,3], and ordinary differential equations (ODEs) [4,5], see [6–9] for reviews of these and other approaches. However using only gene expression data will likely not be sufficient because the noise inherent in the measurements as well as the expense and difficulty to obtain numerous measurements under different experimental conditions implies the inference process on a whole-genome level will always be underdetermined with respect to the amount of data available. Recent techniques attempt to integrate additional data sources or introduce constraints to help guide the inference procedure. Such techniques consider including modeling of environmental and transcription factor interactions [10,11], incorporating DNA motif sequence in gene promoter regions [12–14], combining multiple microarray datasets from the same organism across multiple experiments [15,16] or from completely different organisms [2], and integrating proteomics and metabolomics [17]. Yet despite these advances, gene network inference remains an extremely difficult problem and new integrative techniques still need to be explored.

One particularly interesting type of data is experimentally determined physical interactions whereby the genes regulated by specific cis-acting transcription factors are identified. These experimental approaches use protocols such as ChIP-chip and ChIP-seq [18,19] to perform genome-wide measurements, and they have been used to construct putative regulatory networks [20–22] under the assumption that binding peaks discovered in gene promoter regions implies regulation of those genes. These protocols are also useful to measure data such as DNA methylation distributions [23], epigenetic state and chromatin structure [24,25], and transcription factor promoter occupancy [20]. While this data has been used with gene expression data for identification of regulation for a small set of genes [26,27], there currently is no research utilizing this type of data as part of the genome-wide computational inference of gene networks. This experimental strategy provides high-quality interaction data but is restricted in that the transcription factors must be known in advance and effective antibodies must be available for the ChIP protocol to work, therefore many interactions are missed and only provides a small subset of the regulatory network. However, these interactions can be utilized as a priori network information to help guide inference procedures. Probabilistic approaches can incorporate this existing network information through prior distributions [28,29], but these techniques are computationally expensive. ODE methods are more computationally tractable, however there is no existing research that shows how to systematically incorporate prior network information. We fill that void in this article by extending the ODE methodology into a general optimization framework that incorporates gene expression data with existing network information. Our approach continues upon earlier work by also utilizing regularization parameters to encourage network sparsity, and we show that various types of experimental data can be formulated into the framework. We prove that our estimator is asymptotically root-n consistent with the estimated weights converging to the true weights at a rate of $\frac{1}{\sqrt{n}}$, where n is the number of data observations, and that there is a corresponding probabilistic interpretation which is also asymptotically root-n consistent.

We test our method on simulated network data and show that existing network information improves performance, overcomes the lack of observations, and performs well even when some of the existing network information is incorrect. We demonstrate the applicability of our framework to real biological data by inferring the core regulatory network for embryonic stem cells. We utilize predicted interactions from two experimental studies, each using a different experimental technique, as existing network information, and we show that including the experimental network data constructs a more closely representative network versus when no information is provided.

Methods

Gene network inference based on ordinary differential equations (ODEs) describes gene regulation as a function of other genes:

$$\frac{dx_i(t)}{dt} = F_i\left(x_1(t), \ldots, x_p(t)\right) \tag{1.1}$$

where $x_i(t)$ is the concentration of mRNA for gene i measured at time t, $dx_i(t)/dt$ is the rate of change for the mRNA concentration of gene i, and p is the number of genes. Each function F_i represents all of the various factors that affect the amount of mRNA for gene i including processes such as transcription rate, degradation, post-transcriptional modifications, translation rate, etc. This representation indicates a causal interaction versus a conditional probability as with statistical approaches, but does not necessarily imply a physical interaction or a direct relationship as proteins, metabolites, transcription factor binding, and other regulatory processes are not explicitly represented. The advantage of this dynamical system representation is that the model can be expanded to include any of these more detailed interactions. Though this introduces additional flexibility as it adds more parameters to the model, little data is available for these intermediate processes, and the functional form is generally not known. Furthermore, the system may be amenable to analysis to deduce properties such as existence of steady-state solutions, bistability and sensitivity analysis of parameter values; and numerical simulation can be performed for quantitative prediction and validation.

Variations exist with the exact formulation depending on whether the available expression data is time-series or steady-state measurements, the assumed experimental noise model, and the particular function form for F_i that is chosen. For current simplicity of presentation we will take the form that approximates the gene regulatory network with a linear system of equations such as used by Gardner et al. [5] and expanded upon by others [4,16,30]; however we will show in later sections how different forms of data can be included as well as non-linear functions. The model considers a set of external perturbations u that have been applied to one or more gene resulting in the following set of linear ODEs:

$$\frac{dx}{dt} = Wx - u \tag{1.2}$$

where W is a $p \times p$ matrix containing the interaction coefficients and constitutes the network model to be inferred. Given n observations, $\left(x^1,u^1\right), \ldots, \left(x^n,u^n\right)$, of the mRNA concentrations for p genes and their perturbations, and under the assumption the observations are made at steady-state $(dx_i/dt = 0)$, inferring W in Eq. (1.2) can be expressed as a least-squares minimization problem:

$$\hat{W} = \arg\min_W f(W) = \sum_{j=1}^{n} \left\| Wx^j - u^j \right\|^2 \tag{1.3}$$

Regularization

For $p > n$, the linear system is underdetermined. Gardner et al. [5] with their network identification by multiple regression (NIR) algorithm argue that assuming a maximum of k incoming connections serves to transform the problem into an overdetermined system, and makes it robust to measurement noise and incomplete data. However, this restriction does not reliably prevent overfitting and all genes tend to have exactly k incoming connections to minimize the linear regression error, regardless of whether all those connections are valid. Computationally, the NIR algorithm has to run $\binom{p}{k}$ multiple linear regressions for each gene which becomes intractable for large p and modest values for k. Even if there are enough observations, regression tends to use as many genes as possible to explain the data and thus overfits by including all k network connections. A more appropriate methodology is one based on sparsity. Genes should only have enough connections to predict their expression data without overfitting, and genes should be allowed to have differing number of connections to properly reflect the underlying network structure implied by the data. Various regularization techniques have been introduced to prevent overfitting including ridge regression [31], LASSO [32–34], and elastic net [35]. Ridge regression uses an L_2-norm constraint to maintain the best predictors, but it does not encourage sparsity and is not necessarily the most parsimonious model. The LASSO (least absolute shrinkage and selection operator) method adds an L_1-norm constraint; this constraint tends to produce connection coefficients that are exactly zero, and thus acts to enforce parsimony. Elastic net combines both of these constraints. Gustafsson et al. [4] and the Inferelator [10] both use LASSO and provide evidence that it selects parsimonious models. We consider using LASSO as the basis of our algorithm to enforce network sparsity and will enhance it to include existing network information, so the minimization problem becomes:

$$\hat{W} = \arg\min_W g(W) = f(W) + \alpha \|W\|_1 \tag{1.4}$$

where $\|W\|_1 = \sum_{i=1}^{p} \sum_{j=1}^{p} \left|W_{ij}\right|$ and α is a positive parameter that enforces the level of sparsity in the gene network. The parameter α is learned through cross-validation with larger values for α

producing a more sparse matrix while $\alpha=0$ corresponds to the standard least-squares regression problem.

Incorporating existing network information

Existing network information can be incorporated into the minimization problem by adding an additional constraint for connections in the network. Given a W^0 matrix with positive entries $W^0_{ij} \geq 0$ indicating the lack of interaction for gene j on gene i, the problem becomes:

$$\hat{W} = \arg\min_{W} g(W) = f(W) + \alpha \|W\|_1 + \beta \left\| W \circ W^0 \right\|_1 \tag{1.5}$$

where $W \circ W^0$ denotes the entry-wise product between matrix W and W^0. This adds a penalty to edges in W that do not exist in W^0, making those edges less likely to be included. Notice that this formulation does not force edges provided by the existing network information to be included in the resultant network; instead those edges are just not penalized with $W^0_{ij}=0$ which will make them more likely to be picked over other edges. This allows the optimization to still pick a different network structure if it fits the data better. The strength of the penalty is determined by a positive parameter β, which is learned through cross-validation. If the existing network information is not beneficial to reducing the error of the inferred network model, then cross-validation will set $\beta=0$ to eliminate the penalty. However $\beta>0$ signifies that the existing network information is beneficial.

Optimization framework

We introduce a general optimization framework for various types of gene expression data that incorporates sparsity and existing network information. This formulation encompasses the standard least-squares problem as in Eq. (1.3), yet it is flexible enough to handle gene-specific problem alterations such as those required for certain kinds of gene expression perturbation data. Let $f(W)$ be a quadratic function of the $p \times p$ square matrix W, defined in the form:

$$f(W) = \frac{1}{2}\mathrm{tr}\left(W^T W \Sigma\right) - \mathrm{tr}(WU) + \frac{1}{2}\sum_{i=1}^{p} W_i D^i W_i^T \tag{1.6}$$

where W_i denotes the i-th row vector of the matrix W. Matrices Σ and D^i are symmetric and positive definite, that is $\Sigma \succ 0$ and $D^i \succ 0$ for all $i = 1, \cdots, p$. Under this definition $f(W)$ is a convex function of W. Our goal is to find W that minimizes $f(W)$ subject to sparsity constraint and existing network information:

$$\hat{W} = \arg\min_{W} g(W) = f(W) + \alpha \|W\|_1 + \beta \left\| W \circ W^0 \right\|_1 \tag{1.7}$$

We simplify the notation to the following:

$$\arg\min_{W} g(W) = f(W) + \|\Lambda \circ W\|_1 \tag{1.8}$$

by defining a $p \times p$ matrix Λ that combines the two parameters:

$$\Lambda_{ij} = \alpha + \beta W^0_{ij} \tag{1.9}$$

We use a coordinate descent algorithm to solve this optimization problem for a given Λ matrix [36]. The algorithm iteratively updates each W_{ij} matrix entry until $f(W)$ converges to its minimum value; convergence is guaranteed by the convexity of the function and the additivity of the L_1 regularization term [37]. The derivate of $f(W)$ with respect to W_{ij} is:

$$\frac{\partial f(W)}{W_{ij}} = W_{ij}\left(\Sigma_{jj} + D^i_{jj}\right) - \gamma_{ij} \tag{1.10}$$

where γ_{ij} is independent of W_{ij} and is defined to be:

$$\gamma_{ij} \equiv U_{ji} - \sum_{k=1, k\neq j}^{p} W_{ik}\left(\Sigma_{kj} + D^i_{jk}\right) \tag{1.11}$$

Now consider the derivate of $g(W)$ in Eq. (1.8) with respect to W_{ij}:

$$\frac{\partial g(W)}{W_{ij}} = W_{ij}\left(\Sigma_{jj} + D^i_{jj}\right) - \gamma_{ij} + \Lambda_{ij}\,\mathrm{sgn}\left(W_{ij}\right) \tag{1.12}$$

where

$$\mathrm{sgn}\left(W_{ij}\right) \equiv \begin{cases} 1 & W_{ij} > 0; \\ -1 & W_{ij} < 0; \\ \in[-1,1] & W_{ij} = 0 \end{cases} \tag{1.13}$$

Therefore the update rule in the coordinate descent algorithm is:

$$\hat{W}_{ij} = \begin{cases} \frac{\gamma_{ij} - \Lambda_{ij}}{\Sigma_{jj} + D^i_{jj}} & \gamma_{ij} > \Lambda_{ij}; \\ \frac{\gamma_{ij} + \Lambda_{ij}}{\Sigma_{jj} + D^i_{jj}} & \gamma_{ij} < -\Lambda_{ij}; \\ 0 & \mathit{otherwise} \end{cases} \tag{1.14}$$

The resultant network model $\hat{W}$ is dependent upon the values used for the two parameters (α, β), so we use cross-validation to find values that provide the minimum total testing error. K-fold cross-validation is common practice; however there tends to be few observations for gene expression data, so we have used leave-one-out cross-validation which puts just one observation into the testing set. Note that if all $\Lambda_{ij} > \Lambda^{\max} = \max_{i,j} \left| U_{ij} \right|$ then all matrix entries W_{ij} will be constrained to be zero, therefore:

$$\Lambda^{\max} = \max \left| U_{ij} \right| \tag{1.15}$$

This provides bounds $0 < \Lambda_{ij} < \Lambda^{\max}$ that need to be searched. We perform an exponential search starting from $\Lambda^{\max}$ and going down, using the W matrix as a warm start from one value to the next. In fact, Λ is comprised of two parameters so a matrix of α and β values is constructed and the minimum error for the parameter pair is chosen.

Perturbation gene expression data

Suppose we are given a set of n observation, $\left(x^1, u^1\right), \left(x^2, u^2\right), \ldots, \left(x^n, u^n\right)$, representing the activities of genes x_i in the network after input perturbation u_i, for $i = 1, \ldots, p$. We infer the network connections W between the genes by minimizing the following error function:

$$\hat{W} = \arg\min_{W} f(W) = \sum_{j=1}^{n} \left\| Wx^j - u^j \right\|^2 \qquad (1.16)$$

The error function can be rewritten as:

$$\begin{aligned} f(W) &= \sum_{j=1}^{n} (Wx^j - u^j)^T (Wx^j - u^j) \\ &= 2\left[\tfrac{1}{2}\operatorname{tr}(W^T W \Sigma) - \operatorname{tr}(WU)\right] + C \end{aligned} \qquad (1.17)$$

where C is a term independent of W thus dropping out of the minimization, and the matrices Σ and U are defined to be:

$$\Sigma = \sum_{j=1}^{n} x^j (x^j)^T \quad U = \sum_{j=1}^{n} x^j (u^j)^T \qquad (1.18)$$

Thus standard gene expression perturbation experiments fit in the optimization framework as described in Eq. (1.6) with $D = 0$.

Modified formulation for different perturbation experiments

The formulation as described in Eq. (1.3) requires a measurement of the external perturbation distinct from the expression values of the perturbed genes, which might not be available for all types of experiments. We propose an alternate formulation where we consider genetic perturbations $\Delta x^b = x^b - x^*$ away from the gene expression of the wild-type gene network x^*. We show in the next sections that we can still express the system as a least-squares minimization problem in the same form as Eq. (1.6) and use our optimization framework to find the best network model to explain the perturbation.

Null mutant gene expression data. Null mutant experiments remove or prevent a gene from being expressed; the simplest form is to knock out one gene per experiment then measure the steady-state expression values for the $p-1$ other genes. Suppose we are given p observations where a single different gene i is removed in each observation. Denote the observed steady state by the vector x_i^{null}, for all $i = 1, \cdots, p$ after gene i is removed. The perturbation of x_i^{null} away from the wild-type steady state is:

$$\Delta x_i^{null} = x_i^{null} - x^* \qquad (1.19)$$

We infer the connection model and strengths between the genes by minimizing the following error function:

$$f(W) = \sum_{i=1}^{p} \sum_{j=1, \neq i}^{p} \left[W \Delta x_i^{null} - \Delta x_i^{null} \right]_j^2 \qquad (1.20)$$

Gene i cannot be used to predict itself for the observation when gene i is removed, so the error function indicates this by excluding gene i for observation i. We reformulate the error function and cast into Eq. (1.6) where we have defined:

$$\begin{aligned} \Sigma &= \sum_{i=1}^{p} \Delta x_i^{null} \left(\Delta x_i^{null}\right)^T \\ D^i &= \Delta x_i^{null} \left(\Delta x_i^{null}\right)^T \\ U_{ij} &= \Sigma_{ij} - \Delta x_{ii}^{null} \Delta x_{ij}^{null} \end{aligned} \qquad (1.21)$$

Heterozygous knockdown gene expression data. Heterozygous knockdown experiments remove one of two copies of a gene; a series of experiments might knockdown one gene per experiment then measure the steady-state gene expression values for all p genes. Suppose we are given p observations where a single different gene i is knocked down in each observation. Denote the observed steady state by the vector x_i^h, for all $i = 1, \cdots, p$ after removing once copy of gene i. The perturbation of x_i^h away from the wild-type steady state is:

$$\Delta x_i^h = x_i^h - x^* \qquad (1.22)$$

However for the experiment with gene i knocked-down, only one copy of gene i can contribute so we denote the perturbation of the other copy of gene i by:

$$\tilde{\Delta} x_{ii}^h = x_{ii}^h - x_i^* / 2 \qquad (1.23)$$

We infer the connection model between the genes by minimizing the following error function:

$$f(W) = \sum_{i=1}^{p} \sum_{j=1, \neq i}^{p} \left[W \Delta x_i^h - \Delta x_i^h \right]_j^2 + \sum_{i=1}^{p} \left[\frac{1}{2} W_i \Delta x_i^h - \tilde{\Delta} x_{ii}^h \right]^2 \qquad (1.24)$$

We can reformulate the error function and cast into Eq. (1.6) with:

$$\begin{aligned} \Sigma &= \sum_{i=1}^{p} \Delta x_i^h \left(\Delta x_i^h\right)^T \\ D^i &= \tfrac{3}{4} \Delta x_i^h \left(\Delta x_i^h\right)^T \\ U_{ij} &= \Sigma_{ij} - \left(\Delta x_{ii}^h - \tfrac{1}{2} \tilde{\Delta} x_{ii}^h \right) \Delta x_{ij}^h \end{aligned} \qquad (1.25)$$

Time-series gene expression data

Time-series gene expression data has also been utilized in ODE methodology. The basic idea is to no longer assume the system has been measured at steady-state $(dx/dt = 0)$, and use the time-series data as an approximation to the derivative. We can still consider perturbations to the system but we should take into account that dynamics of different genes operate on different time scales. We are given a set of trajectories, each from a different initial perturbation, along with n observations for all genes at unit time intervals Δt as the system relaxes back to the steady state. The final measurement is not necessarily the steady state. We consider each trajectory as a time sequence, $x_i^1, x_i^2, \ldots, x_i^n$, for each gene i. We have a linear system of the form:

$$\tau \frac{dx}{dt} = Wx - u \qquad (1.26)$$

The derivative can be estimated in a number of ways, but we will consider here the Mean-Value Theorem approximation:

$$\frac{dx_i^t}{dt} \approx \frac{x_i^{t+1} - x_i^{t-1}}{2\Delta t} \qquad (1.27)$$

The problem is formulated as a least-squares minimization problem and we use our same optimization framework as before:

$$\hat{W} = \arg\min_{W} f(W) = \sum_{t=2}^{n-1} \left\| Wx^t - \left(u^t + \tau \frac{dx^t}{dt} \right) \right\|^2 \qquad (1.28)$$

However, we do not know τ_i for each gene so we will need to learn it iteratively with W. Assume that we have a reasonable initial value for τ_i, we first calculate $\hat{W}$ then optimize Eq. (1.26) for each τ_i:

$$E(\tau_i) = \tau_i A_i - B_i = 0 \tag{1.29}$$

where $A_i = \left(\frac{dx_i^2}{dt}, \cdots, \frac{dx_i^{n-1}}{dt}\right)$ and $B_i = u_i - \hat{W}_i x$. The optimal solution for τ_i is:

$$\tau_i = A_i^T B_i \left(A_i^T A_i\right)^{-1} \tag{1.30}$$

Use the determined τ_i to recalculate the derivative entries in Eq. (1.28) and compute a new $\hat{W}$, repeat this iteratively until convergence.

Nonlinear functional form

Consider just a single gene x_i for Eq. (1.3) where we have a nonlinear differentiable response function F_i and we want to find:

$$\hat{\beta}_i = \arg\min_{\beta} g(\beta_i) = \sum_{j=1}^{n} \left\| F_i\left(\beta_i^T x^j\right) - u_i^j \right\|^2 \tag{1.31}$$

where β_i are the interaction coefficients for gene i, essentially corresponding to row i of the network model W. To solve the problem, we first find a quadratic approximation of the objective function around an arbitrary point β_0:

$$\begin{aligned} g(\beta) = \sum_{j=1}^{n} \left\| F_i\left(\beta_i^T x^j\right) - u_i^j \right\|^2 &= g(\beta_0) + J(\beta_0)^T \Delta\beta \\ &+ \frac{1}{2} \Delta\beta^T H(\beta_0) \Delta\beta + o\left(\|\Delta\beta\|^2\right) \end{aligned} \tag{1.32}$$

where $\Delta\beta = \beta - \beta_0$, $J(\beta_0) = \nabla g(\beta_0)$ is the gradient of g at β_0, and $H(\beta_0)$ is the Hessian matrix. Specifically, we have:

$$\begin{aligned} J(\beta) &= 2 \sum_{j=1}^{n} \left[F\left(\beta_i^T x^j\right) - u_i^j \right] F'\left(\beta_i^T x^j\right) x^j \\ H(\beta) &= 2 \sum_{j=1}^{n} \left[\left[F\left(\beta_i^T x^j\right) - u^j \right] F''\left(\beta_i^T x^j\right) - F'\left(\beta_i^T x^j\right)^2 \right] x^j \left(x^j\right)^T \end{aligned} \tag{1.33}$$

Thus, around β_0, the problem fits into our convex optimization framework and can be solved with the same coordinate descent algorithm:

$$\hat{\beta}(\beta_0) = \arg\min_{\beta} f(\beta) = \frac{1}{2} \beta^T H(\beta_0) \beta + [J(\beta_0) - H(\beta_0)\beta_0] \beta \tag{1.34}$$

In summary, we propose the following algorithm to independently find the set of interaction coefficients for each gene:

1) Randomly choose $\beta_0 \in R^p$.
2) Find $\hat{\beta}(\beta_0)$ using Eq. (1.34) and the coordinate descent algorithm.
3) Perform line search: find $\hat{\alpha} \in [0,1]$ such that

$$\hat{\alpha} = \arg\min_{\alpha} g(\beta(\alpha))$$

where $\beta(\alpha) = \beta_0 + \alpha\left[\hat{\beta}(\beta_0) - \beta_0\right]$.

4) Set $\beta_0 = \beta\left(\hat{\beta}\right)$ and go to Step 2) if $\alpha \neq 0$.

Asymptotic properties

Consider Eq. (1.16) with a matrix Λ of non-negative entries and a parameter for L_1-norm regularization:

$$\begin{aligned} \hat{W} = \arg\min_{W} g(W) &= f(W) + \lambda \|\Lambda \circ W\|_1 \\ &= \sum_{j=1}^{n} \left\| W x^j - u^j \right\|^2 + \lambda \|\Lambda \circ W\|_1 \end{aligned} \tag{1.35}$$

This equation is equivalent to:

$$\hat{W} = \arg\min_{W} g(W) = \mathrm{tr}\left(W^T W \Sigma\right) - 2\mathrm{tr}\left(W U^T\right) + \lambda \|\Lambda \circ W\|_1 \tag{1.36}$$

with matrices Σ and U as defined in Eq. (1.18).

Consider the following noise model:

$$u = Wx + \varepsilon \tag{1.37}$$

where the noise term $\varepsilon \sim N\left(0, \sigma^2 I\right)$ follows normal distribution with a fixed variance.

Theorem 1

If $\lambda/\sqrt{n} \to \lambda_0 \geq 0$, $\Lambda = O(1)$, *and*

$$C = \lim_{n \to \infty} \left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) \tag{1.38}$$

is non-singular, then when $n \to \infty$,

$$\sqrt{n}\left(\hat{W} - W\right) \xrightarrow{D} \arg\min(V) \tag{1.39}$$

where

$$\begin{aligned} V(Z) &= \mathrm{tr}\left(Z^T Z C\right) - 2\mathrm{tr}\left(Z \Psi^T\right) \\ &+ \lambda_0 \mathrm{tr}\left\{ [I(W \neq 0) \circ \mathrm{sgn}(W) \circ Z + I(W = 0) \circ |Z|] \Lambda^T \right\} \end{aligned} \tag{1.40}$$

and Ψ *is a random matrix with normal distribution of mean 0 and covariance* $E\left[\Psi_{ij}\Psi_{kl}\right] = \Sigma_{ik}\delta_{jl}\sigma^2$ *for all i, j, k, l.*

Proof of the theorem is provided in Appendix S1. This theorem suggests that the estimate is root-n consistent with the estimated $\hat{W}$ converging to the true W at a rate of $1/\sqrt{n}$ when the penalty term is reduced at a rate of $1/\sqrt{n}$. The root-n consistency property is similar to the asymptotic property of the Lasso estimator first described by Knight and Fu [38].

A special case: orthogonal design. In this section, we show how prior network information can aid us in network inference with a special case where we can analytically solve for the error rate. Consider the case of an orthogonal design: $\Sigma = I$. Different entries of W decouple and have the optimal value in the form of a soft-threshold function:

$$\hat{W}_{ij} = \mathrm{sgn}\left(U_{ij}\right) \left[U_{ij} - \lambda_{ij}\right]_+ \tag{1.41}$$

Now consider a noise model of $U_{ij} = W_{ij} + \varepsilon_{ij}$ where $\varepsilon_{ij} \sim N\left(0, \sigma^2\right)$ for all i, j. We assume W is sparse with N^- zero entries and N^+ non-zero entries. Note that $N^- + N^+ = p^2$. Denote $\phi(x)$ the density of a standard normal distribution and $\Phi(x)$ its cumulative distribution. The probability of $\hat{W}_{ij} = 0$ is:

$$\begin{aligned}\Pr\left(\hat{W}_{ij}=0\right) &= \Pr\left(\left|W_{ij}+\varepsilon_{ij}\right| \le \lambda_{ij}\right) \\ &= \Phi\left(\frac{\lambda_{ij}-W_{ij}}{\sigma}\right) - \Phi\left(\frac{-\lambda_{ij}-W_{ij}}{\sigma}\right)\end{aligned} \tag{1.42}$$

If $W_{ij} = 0$, the probability of misidentifying $\hat{W}_{ij}$ as non-zero is:

$$1-\Pr\left(\hat{W}_{ij}=0\right) = 1-\Phi\left(\frac{\lambda_{ij}}{\sigma}\right) + \Phi\left(\frac{-\lambda_{ij}}{\sigma}\right) \tag{1.43}$$

The overall error rate of misidentifying non-zero entries as zero and zero entries as non-zero is:

$$\begin{aligned}&\sum_{i=1}^{p}\sum_{j=1}^{p} I\left(W_{ij}=0\right)\left[1-\Phi\left(\frac{\lambda_{ij}}{\sigma}\right)+\Phi\left(\frac{-\lambda_{ij}}{\sigma}\right)\right] \\ &+ I\left(W_{ij}\neq 0\right)\left[\Phi\left(\frac{\lambda_{ij}-W_{ij}}{\sigma}\right)-\Phi\left(\frac{-\lambda_{ij}-W_{ij}}{\sigma}\right)\right]\end{aligned} \tag{1.44}$$

For simplicity of discussion, assume $W_{ij} = w$ for all non-zero entries and $\lambda_{ij} = \lambda$ for all i, j. Then the error rate is:

$$E\left(\bar{\lambda}\right) = N^-\left[1-\Phi\left(\bar{\lambda}\right)+\Phi\left(-\bar{\lambda}\right)\right] + N^+\left[\Phi\left(\bar{\lambda}-\bar{w}\right)+\Phi\left(-\bar{\lambda}-\bar{w}\right)\right] \tag{1.45}$$

where $\bar{w} = w/\sigma$ and $\bar{\lambda} = \lambda/\sigma$. When $\bar{w} \gg 1$, the optimal $\bar{\lambda}$ that minimizes $E\left(\bar{\lambda}\right)$ is:

$$\bar{\lambda}^* = \min\left\{0,\ \frac{\bar{w}}{2}\frac{2\alpha}{1-\alpha}\right\} \tag{1.46}$$

where $\alpha = N^-/(N^- + N^+)$ is the proportion of zero entries, or the sparsity of the network. The optimal regularization $\bar{\lambda}^*$ can be derived through cross-validation in real applications if the number of observations is sufficiently large.

Next consider how to incorporate the existing network information (potentially incorrect). Suppose we are provided with the information that a subset of the connections is zero. For these connections, we increase the regularization parameter λ to $\lambda + \gamma$ with $\gamma \ge 0$. Suppose among the subset, K^- connections are indeed zero and K^+ connections are actually non-zero. In summary, we have 4 groups of connections: 1) K^- with true zero connection strength and parameter $\lambda + \gamma$; 2) $N^- - K^-$ with zero connection strength and parameter λ; 3) K^+ with non-zero connection strength and parameter $\lambda + \gamma$; and 4) $N^+ - K^+$ with non-zero connection strength and parameter λ. The total error rate is then:

$$\begin{aligned}E\left(\hat{\lambda},\hat{\gamma}\right) &= K^-\left[1-\Phi\left(\bar{\lambda}+\bar{\gamma}\right)+\Phi\left(-\bar{\lambda}-\bar{\gamma}\right)\right] \\ &\quad + K^+\left[\Phi\left(\bar{\lambda}+\bar{\gamma}-\bar{w}\right)-\Phi\left(-\bar{\lambda}-\bar{\gamma}-\bar{w}\right)\right] \\ &\quad + (N^- - K^-)\left[1-\Phi\left(\bar{\lambda}\right)+\Phi\left(-\bar{\lambda}\right)\right] \\ &\quad + (N^+ - K^+)\left[\Phi\left(\bar{\lambda}-\bar{w}\right)-\Phi\left(-\bar{\lambda}-\bar{w}\right)\right]\end{aligned} \tag{1.47}$$

where $\bar{\gamma} = \gamma/\sigma$. Similarly, if $\bar{w} \gg 1$, the optimal $\bar{\gamma}$ is:

$$\hat{\gamma}^* = \min\left\{0,\ \frac{1}{\bar{w}}\left(\frac{2\beta}{1-\beta}-\frac{2\alpha}{1-\alpha}\right)\right\} \tag{1.48}$$

where $\beta = K^-/(K^- + K^+)$ is the sparsity in the subset. Note that $\hat{\gamma}^* > 0$ if and only if $\beta > \alpha$. This suggests that by adjusting the parameter γ through cross-validation, we should be able to decrease the prediction error rate if the sparsity of the subset is lower than the sparsity of the entire network. On the other hand, if $\beta < \alpha$, this means that the prior information on the subset is actually worse than a random guess. In that case, the cross-validation should set $\gamma = 0$ thus ignoring the prior information.

Probabilistic Interpretation

There is correspondence of the least-square minimization model as described above with a probabilistic interpretation using Gaussian random fields to model the interaction between genes. Under this model, we assume the distribution of gene expression values are described by a multivariate normal distribution with mean μ and covariance matrix Σ. We are interested in the inverse of the covariance matrix, $\Omega = \Sigma^{-1}$, which encodes conditional dependency between two genes conditioned on all others, and therefore it contains information on the connectivity between genes. Our goal is to estimate Ω from a set of observations $\left\{x^1, x^2, \ldots, x^n\right\}$ where $x^j \in R^p$ is a vector with dimension p, and we assume each x_i is standardize with mean 0. Then the log likelihood for observing the data is:

$$l(x;\Sigma) = -\frac{n}{2}\log\det(\Sigma) - \frac{1}{2}\sum_{j=1}^{n}\left(x^j\right)^T\Sigma^{-1}x^j \tag{1.49}$$

up to a constant difference. After standardizing the data, we can rewrite the log likelihood as a function of Ω:

$$l(x,\Omega) = \frac{n}{2}\log\det(\Omega) - \frac{n}{2}\mathrm{tr}(S\Omega) \tag{1.50}$$

where $S = \sum_{j=1}^{n} x^j\left(x^j\right)^T \Big/ n$ is the empirical covariance matrix. The inverse covariance matrix Ω can be estimated by maximizing this log likelihood function. Non-zero entries, Ω_{ij}, indicate the existence of a connection between gene i and j. Also note that different from the linear network model, the probabilistic model infers a network with undirected edges as it is making a statement about conditional dependency between two genes. This problem can be solved using several convex optimization algorithms including interior point methods [39] and coordinate descent [34].

When $n < p$, S is singular and the solution is underdetermined. Even when $n \ge p$, we will likely overfit the data using Eq. (1.50) if n is not large. A solution is to add a regularization term to the log likelihood function. In fact, we can incorporate the same L_1-norm sparsity constraint that we did for the ODE model, thus providing us the capability to both enforce a parsimonious model as well as introduce existing network information. Given a matrix Λ of non-negative entries and a single non-negative parameter λ, we can formulate the following minimization problem:

$$\hat{\Omega} = \underset{\Omega \succ 0}{\arg\min} f(\Omega) = -\frac{n}{2}\log\det(\Omega) + \frac{n}{2}tr(S\Omega) + \lambda\|\Lambda \circ \Omega\|_1 \tag{1.51}$$

where $\Lambda \circ \Omega$ is the component-wise product of the two matrices.

Similar to the linear network inference described in Theorem 1, we can prove (provided in Appendix S1) that the estimated Ω converges to the true Ω at the rate of $\frac{1}{\sqrt{n}}$ as $n \to \infty$ thus showing that the estimate is still root-n consistent.

Theorem 2

If $\lambda/\sqrt{n}\rightarrow\lambda_0\geq 0$, $\Lambda=O(1)$, *and* Σ *is non-singular, then*

$$\sqrt{n}\left(\hat{\Omega}-\Omega\right)\overset{D}{\rightarrow}\arg\min(V) \tag{1.52}$$

where

$$\begin{aligned} V(\Delta)=&\frac{1}{4}\mathrm{tr}(\Sigma\Delta\Sigma\Delta)+\frac{1}{2}\mathrm{tr}(Z\Delta)\\ &+\lambda_0\mathrm{tr}\left\{[I(\Omega=0)\circ|\Delta|+I(\Omega\neq 0)\circ\mathrm{sgn}(\Omega)\circ\Delta]\Lambda^T\right\}\end{aligned} \tag{1.53}$$

and Z *is a random matrix with normal distribution of mean 0 and covariance* $E\left[Z_{ij}Z_{kl}\right]=\Sigma_{ij}\Sigma_{jl}+\Sigma_{il}\Sigma_{jk}$ *for all i, j, k, l.*

Results

Simulation Results

We generated a set of random linear network models to test the utility of our optimization framework and to characterize the effect of providing existing network information. Each network contains $p=10$ nodes with 2–3 uniform randomly selected incoming edges for a total of exactly 25 edges in the network; a weight for each edge was randomly drawn from the normal distribution $N(0,1)$. We verified that the generated network, W, was not singular with a valid inverse then generated a random perturbation matrix, u, with $n=15$ experiments (or observations) for all p nodes. Each random response value was drawn from the normal distribution $N(0,1)$ and the observation matrix, x, was calculated:

$$x=W^{-1}u \tag{1.54}$$

Experimental noise was added to both the perturbation and observation matrices. The noise for the perturbation matrix was drawn from $N(0,0.1)$ and for the observation matrix was drawn from $N(0,0.3)$. The larger standard deviation for the observation matrix signifies the additive noise for 2–3 incoming edges from the perturbations.

For our experiments utilizing existing network information, we considered the simplest network information that can be provided, the existence of a directed edge going from one gene to another as a boolean value. The set of existing edges is provided as a boolean network W^0 to our algorithm where an entry $W^0_{ij}=0$ indicates a directed interaction from gene j to gene i and thus is not penalized, while $W^0_{ij}=1$ for all other edges.

Fewer observations decreases prediction performance. One of the key challenges with inferring gene networks from gene expression data is the relatively few observations available compared to the large number of genes. This has a direct effect on how well an inferred model can predict the observed data as illustrated by Figure 1. While an underdetermined linear model can be constructed to fit the data exactly, this is clearly overfitting and cross-validation more correctly specifies the best fitting model. Figure 1 shows how when the number of observations decrease then the error increases for the best fitting model as determined by cross-validation. Furthermore, when the number of observations is greater than the number of variables, then the error stabilizes.

Existing network information overcomes fewer observations. We tested how providing existing network information could overcome the lack of observations by running our algorithm on a set of five randomly generated linear models. We systematically provided more valid edges to the algorithm going from zero edges up to the fully correct network. The results can be seen in Figure 2. For each of the five randomly generated linear network models, we ran our algorithm five times with a different set of edges randomly selected from the correct network. The cross-validation error results are averaged over the 25 total simulation runs for each number of valid edges provided as existing network information.

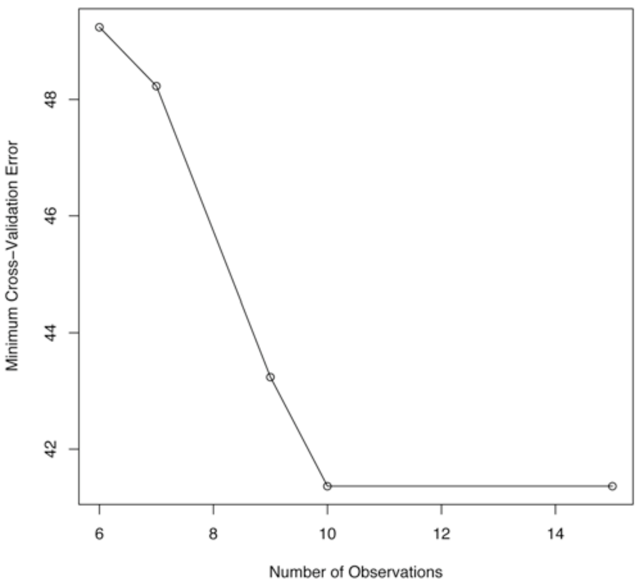

Figure 1. Minimum Cross-Validation Error given Number of Input Observations. The minimum error for the best fitting model as determined by cross-validation and averaged for five simulated linear network models. The network models have ten nodes, so once there are enough observations then the error stabilizes; however the error increases as less observations are provided to the inference algorithm.
doi:10.1371/journal.pone.0006799.g001

Figure 2 clearly illustrates that the error decreases as more valid edges are provided to the algorithm. An interesting observation is that the rate of decrease is significantly more for fewer observations; this indicates that each valid edge has a more important role in inferring the correct network model when the total information available is scarce. Therefore, even providing just a few valid edges can substantially improve the inference process. Furthermore note that the variation is greater for fewer observations, we could have made these curves smoother by running more simulations, but the variation illustrates another key point that not all valid edges are equal in their ability to reduce error. Some edges are more important than others, though which edges is typically not known beforehand.

Providing incorrect network information does not hurt prediction. While we showed in the previous section that providing correct edges as existing network information helps prediction, what about if we give the algorithm incorrect edges? Figure 3 shows the results. If we only provide invalid edges, it does not significantly hurt performance. The reason is apparent if we consider the constraint added to the optimization problem for incorporating network information. If the existing network information provided to the algorithm does not help to reduce the error, then cross-validation will determine the best minimum error is obtained by setting $\beta=0$, essentially ignoring the network information. The algorithm then performs equally as well as when no network information is provided.

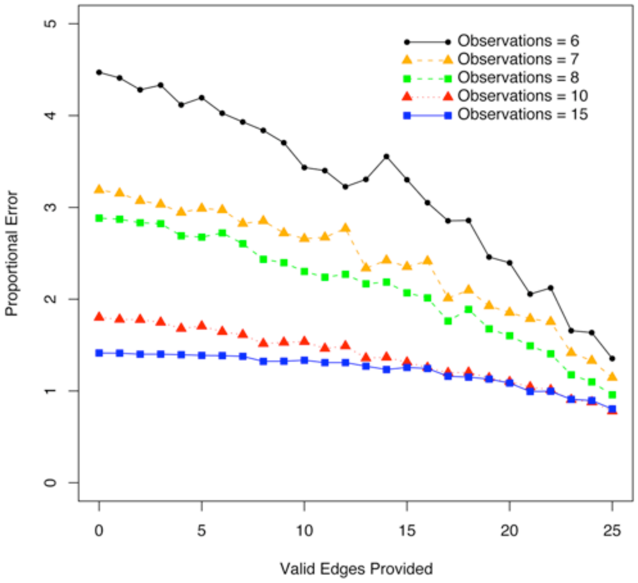

Figure 2. Error Decreases with More Valid Edges Provided as Existing Network Information. Going from zero edges to the fully correct network, randomly selected valid edges are provided as existing network information. The cross-validation error for five simulation runs for five randomly generated linear network models is averaged and plotted as proportional error versus the number of valid edges provided. Because we add experimental noise to the observations, the amount of error varies with the number of observations. Therefore to cancel this bias we calculate a proportional error, which is the minimum cross-validation error averaged across the simulation runs divided by the minimum least-squares error obtained linear regression.
doi:10.1371/journal.pone.0006799.g002

Combination of valid and invalid network information still performs better than providing no network information. It would not generally be the case that the provided existing network information is completely correct; it is more likely that it contains a mixture of valid and invalid edges. Figure 4 shows that our algorithm still performs well in this mixed situation. When the majority of the edges are valid then adding invalid edges increases the error but not significantly enough to detract from the usefulness of the valid edges. Even when there are only a few valid edges compared to the number of invalid edges, then the algorithm is able to still utilize those valid edges. In all cases, providing existing network information, even with some invalid edges, performs better than providing no network information.

Biological Results

The discovery that introduction of transcription factors into mouse and human somatic cells is sufficient to induce a pluripotent stem cell fate has generated considerable excitement [40–46]. Further experiments have been performed to elucidate the core transcriptional network involved with maintaining pluripotency including correlation of transcription factor binding data with gene expression [21], ChIP-seq [22] and biotin-mediated ChIP [20]. While such experiments suggest potential gene regulatory interactions, they do not provide definitive evidence because not all of the detected interactions may be functional [47]. However such data constitutes prior network information we can provide to our gene network inference procedure, and in this section we will compare an inferred network that includes this prior network information versus a network when no information is provided.

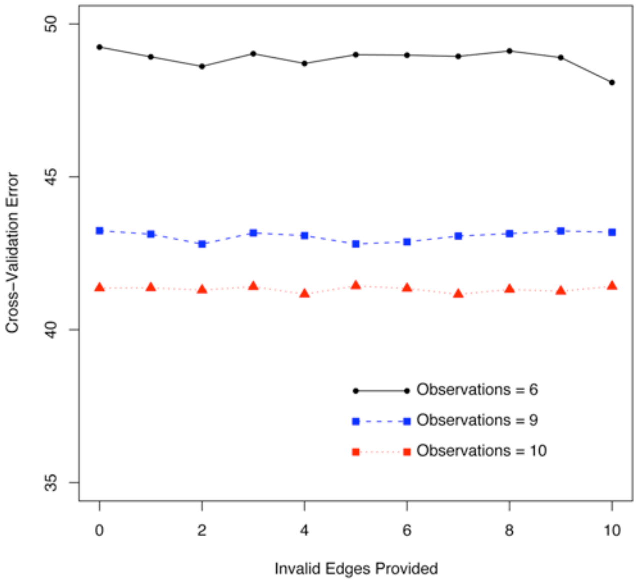

Figure 3. Invalid Edges Does Not Affect Performance. Randomly selected invalid edges provided as existing network information does not affect the minimum error for the best fitting model as determined by cross-validation and averaged for five simulated linear network models.
doi:10.1371/journal.pone.0006799.g003

We focus on the 49 total genes that are included in the regulatory networks constructed by Zhou et al. [21] and Kim et al. [20]. In Figure 5 we show this combined network for three core transcription factors (Nanog, Oct4 and Sox2) and the genes they are hypothesized to regulate. While this figure only shows 25 genes, we performed the computational analysis for all 49 genes (provided in Supplemental Text S1) but restrict our discussion for clarity to these three core factors. What can be seen from Figure 5

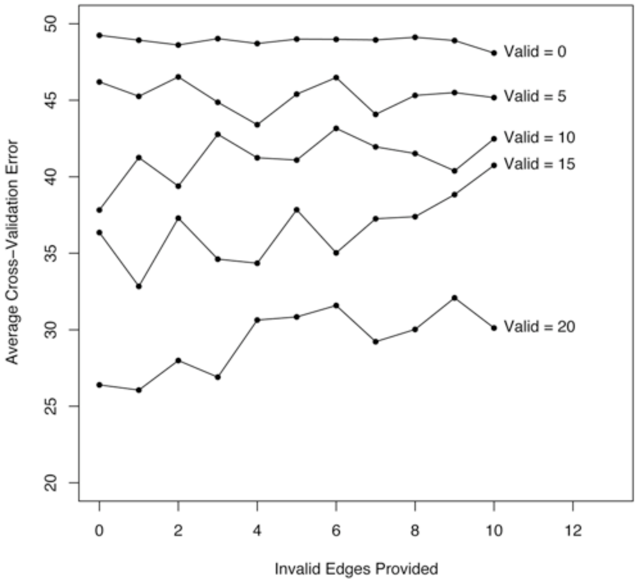

Figure 4. Mixture of Valid and Invalid Edges. Randomly selected valid and invalid edges provided as existing network information still performs well as determined by cross-validation and averaged for five simulated linear network models.
doi:10.1371/journal.pone.0006799.g004

is that the experimental data suggests cross-regulation between all three factors, a larger set of genes co-regulated by all three factors, and a few genes regulated by one or two of the core factors. We use the gene expression data from Ivanova et al. [48] which consists of a time course set of expression values, however we utilize just the final time points which most closely resemble steady state conditions for a total of 16 observations.

Using leave-one-out cross-validation, we find the values for the α (sparsity) and β (prior network) parameters for each gene that minimizes the total testing error. Figure 6 shows the resulting network inferred by our algorithm when given prior network information. For all three core factors, the algorithm gave more weight to the prior network over just the sparsity constraint; Nanog ($\beta = 0.03 > \alpha = 0.001$), Oct4 ($\beta = 0.03 > \alpha = 0.0$) and Sox2 ($\beta = 0.0078 > \alpha = 0.001$). This was not true for all genes, with 23 of the 49 genes having an αvalue less than or equal to the βvalue; the full set of learned parameter values are provided in Supplemental Text S1. We also used cross-validation to learn a network without prior information, so only finding the best α parameter to minimize the testing error. The resultant network is shown in Figure 7.

For both Figure 6 and 7, color is used to illustrate the comparison between the prior network in Figure 5 with the inferred network. Black edges indicate the inferred network predicted the same interaction as in the prior network, red indicates a prior network interaction not predicted in the inferred network, and blue edges are new interactions predicted in the inferred network. While the two inferred networks appear similar, there are significant differences. Overall the inferred network with prior information maintained more edges in correspondence with the prior network keeping 34 edges while the network without prior information kept only 25 edges. Both networks added about the same number of new interactions with 33 for the network with prior information and 32 for the network without prior information. Most notably the network with prior information maintained the co-regulatory interactions between the three core factors while this is lost in the other network.

Closer inspection of the interactions (red edges) present in the experimental network but not in the inferred networks show that for some genes all interactions with the core transcription factors were not predicted. These genes are Rif1, Rybp, Tle4, Tcf7 and

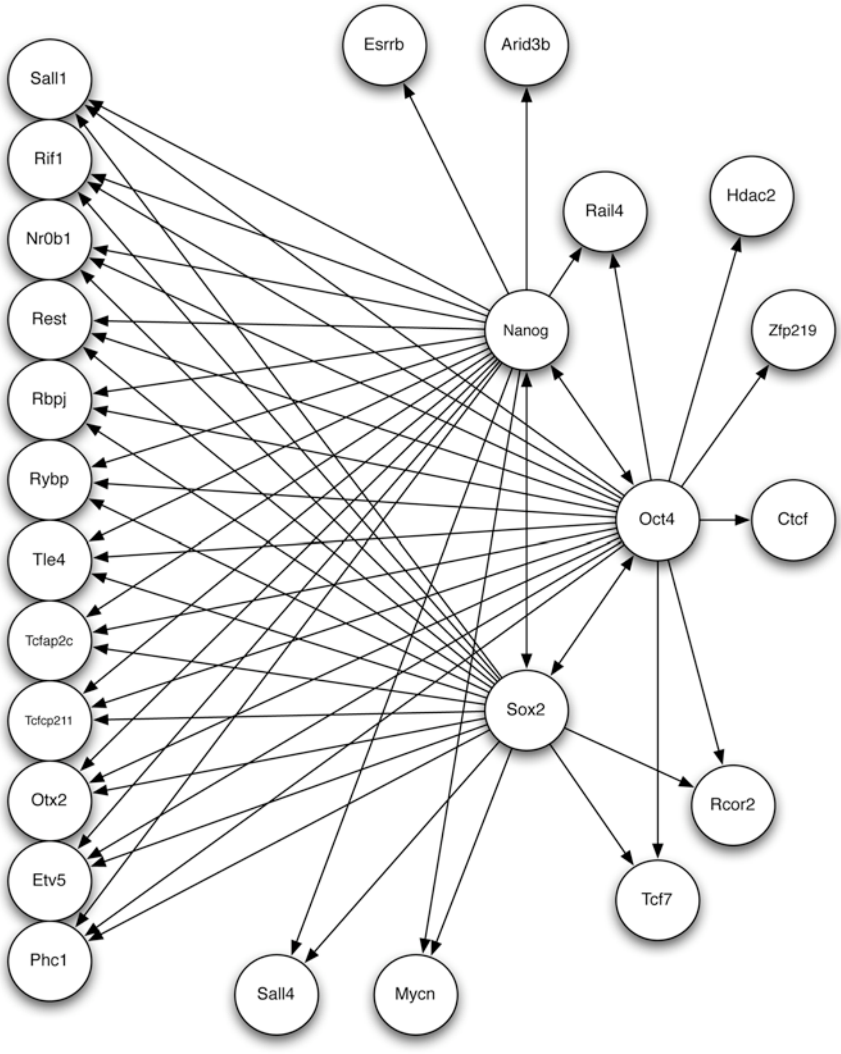

Figure 5. Experiment Transcriptional Network for Mouse Embryonic Stem Cells. Combining the hypothesized interactions obtained through experiments by Zhou et al. [21] and Kim et al. [20] forms prior network information to be provided to our gene network inference algorithm. There are a total of 49 genes but only the 25 genes regulated by the three core factors, Nanog, Oct4 and Sox2, are shown here.
doi:10.1371/journal.pone.0006799.g005

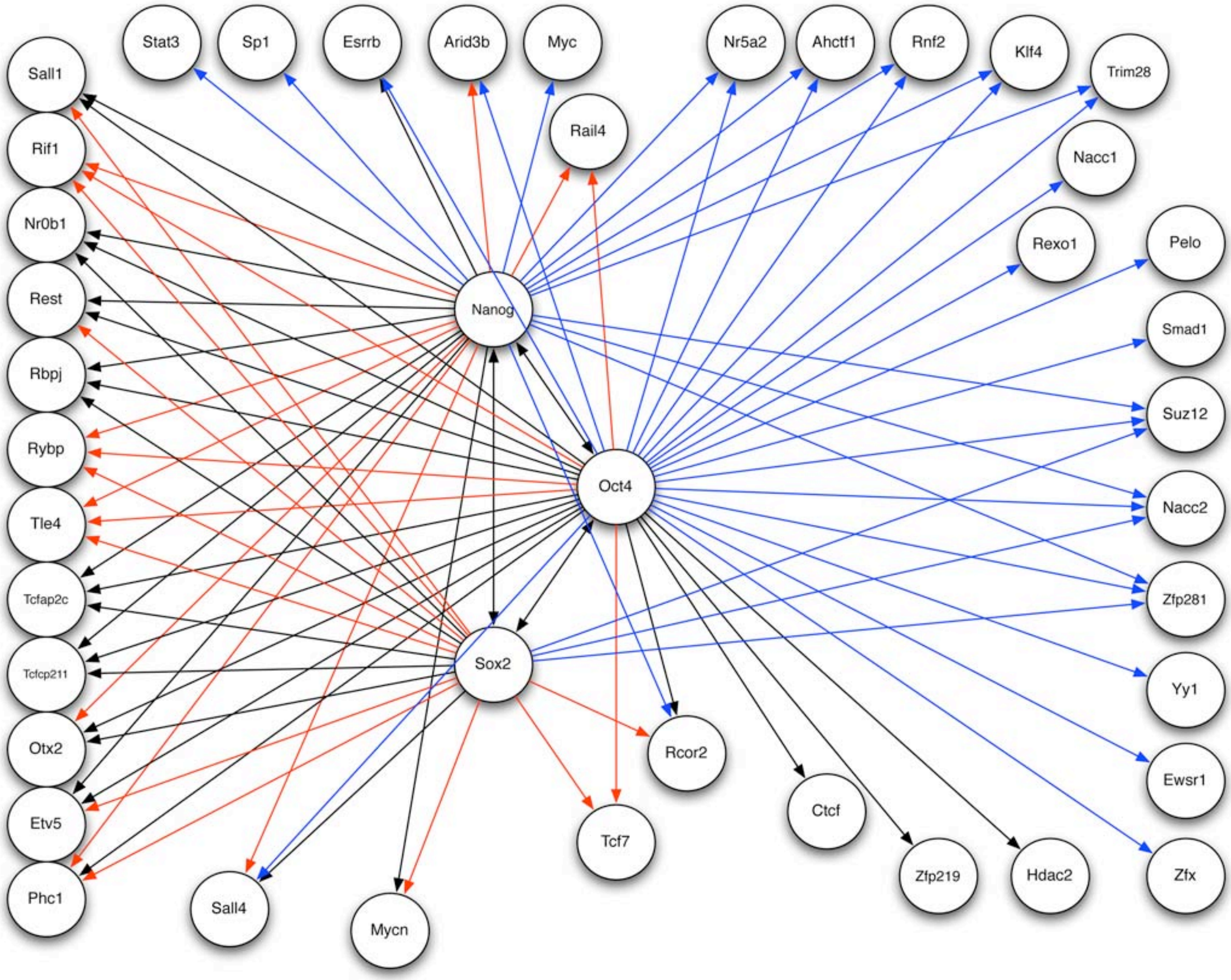

Figure 6. Inferred Network with Prior Information. The gene regulatory network learned through cross-validation with the experimental network provided as existing information. This figure shows just the interactions predicted for the three core factors, Nanog, Oct4 and Sox2. Black edges indicate the inferred interaction matches the experimental network, a red edge indicates an interaction in the experimental network not predicted in the inferred network, and blue edges are new interactions predicted in the inferred network.
doi:10.1371/journal.pone.0006799.g006

Rail4, and these represent the majority of interactions missing in the inferred networks. There are several possible explanations. One is anomalous expression data for the genes, however basic statistical analysis of the data shows the mean expression values for each gene is much greater than its standard deviation which indicates the expression data is not dominated by noise. A second possibility is that the interaction is sufficiently non-linear so a linear approximation fails to capture the connection, but an alternative explanation is that the inferred network failed to confirm a functional interaction as suggested by the experimental network. These five genes are new predictions in the experimental network, and there is little or no experimental confirmation so the lack of correspondence draws into question the validity of those interactions. While we have not checked all interactions, it is encouraging to note that some interactions shared between the experimental and inferred networks include genes known to be important for pluripotency such as Sall4 [49] and Esrrb [50] as well as new inferred interactions such as Oct4 to Yy1 [51]. The combination of gene network inference coupled with the incorporation of existing network information provides a stronger framework for predicting true functional interactions while also doing a better job of excluding false positives.

Discussion

We have taken the ODE methodology for inferring gene networks from gene expression data and extended it to incorporate a priori network information, such that might be obtained from additional biological data like ChIP-chip or ChIP-seq experiments. We have devised a general optimization framework and algorithm that can be applied to gene expression data for a variety of experiments such as perturbation, null mutant, and heterozygous knockdown. Though the focus of the presentation has been on linear ODEs, we have shown that a quadratic approximation of nonlinear functions can be used in the context of the optimization framework. While use of regularization parameters to encourage sparsity in inferred gene networks has been previously reported, we believe this is the first research that utilizes such regularization

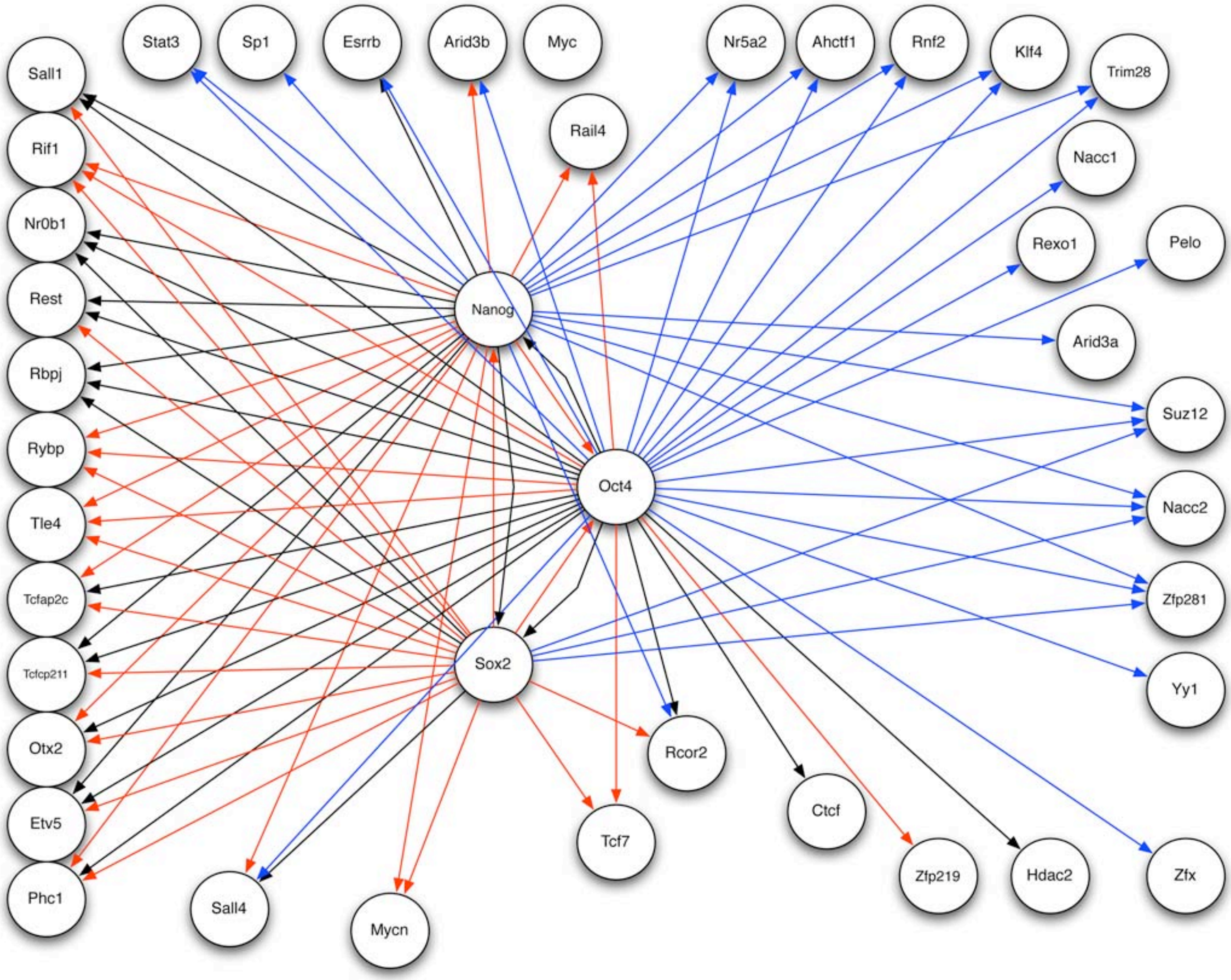

Figure 7. Inferred Network without Prior Information. The gene regulatory network learned through cross-validation with no prior network information provided. This figure shows just the interactions predicted for the three core factors, Nanog, Oct4 and Sox2. Black edges indicate the inferred interaction matches the experimental network, a red edge indicates an interaction in the experimental network not predicted in the inferred network, and blue edges are new interactions predicted in the inferred network.
doi:10.1371/journal.pone.0006799.g007

to integrate existing network information into the inference procedure. We have tested our algorithm on simulated linear network model data and showed that existing network information improves performance, can overcome the lack of observations, and performs well even when some of the existing network information is incorrect.

The method that we use to incorporate existing network information does not force edges to be present in the inferred network model. The penalty actually serves to prevent other edges from being included in the network, so it very much acts as a soft constraint. The algorithm is free to pick other edges if they ultimately fit the data better and cross-validation will insure this by appropriate computation of the parameter value. Though we do not describe the specifics, it is possible to force an edge to be included in the final network. This can be done by removing all of the L_1-norm regularization during the calculation for that specific edge. The coordinate descent algorithm in our optimization framework will then perform the standard least-squares calculation for that edge. Another advantage of our framework for existing network information is that a "confidence" value can be provided for each edge, thus penalizing some edges more or less, by using different values in the W^0 matrix. Furthermore, edges can be effectively removed from the network just by setting their $W_{ij}^0 = \Lambda^{\max}$.

Our presentation of the optimization framework indicates that there is a single value for the α and β parameters for the whole network. However, the interaction coefficients for each gene can be considered independent from the other genes. This means that it is possible to have a separate α and β for each gene. The algorithm would still use cross-validation to calculate the parameter values but it would be done separately for each gene. This introduces more parameters into the model so it has a greater chance of overfitting the data but it also has the advantage of allowing the framework to more selectively utilize existing network information, especially in the situation where the information is good for one gene but poor for another gene. With a single parameter value, the optimization framework has to balance good and bad information for the whole network. Splitting the calculation also makes the algorithm more amenable to parallel computation because each gene is optimized independently and can be performed by separate programs without any synchronization. The final W matrix is constructed by merging the results of the individual genes together. Even with single values for α and β, parallelization can still be utilized either for individual genes and/or for cross-validation, but synchronization is required at intermediate steps.

Computationally, a single optimization run is fast and efficient. For both the simulated and biological networks, a single optimization finished quickly in just a few seconds. What can actually take considerable time is cross-validation to learn the best α and β parameter values. Here one constructs a two-dimensional grid of parameter values, a lower-triangular matrix as entries above the diagonal correspond to solutions with $W=0$. Then for each grid entry a set of optimizations is run to infer a gene network for a subset of observations as well as calculate the cross-validation error. For the biological data with 16 observations, that entails 16 optimization runs per grid entry and with a resolution of 50 intervals for each parameter for a total of 1275 grid entries; inferring the complete gene network in Figure 6 requires over 20,000 optimization runs. Even so this takes only about 1.5 hours on a standard desktop Mac computer. Scaling to genome-wide networks with thousands of genes is certainly a challenge but can be mitigated by decomposition of the problem and parallel processing.

A limitation of our framework is also one that is shared by many other gene network inference procedures. Specifically that the simple network structure of a gene regulating another gene hides the true complexity of the transcriptional, translational and regulatory processes in the underlying biology, and it fails to provide mechanistic hypotheses for how a gene regulates other genes. Despite this limitation, this is a long-term goal that we and others in the field strive to attain by advancing these methods. Of particular note is that ODE methods have the inherent extensibility to incorporate more detailed and complex functional relationships that directly represent physical and causal mechanisms. The challenge is how to efficiently and correctly infer these functional relationships and associated parameters given the limited amount of biological data. We have hinted at one possibility for non-linear functions but considerable work still remains to improve the robustness and efficacy of these approaches.

While our framework presents a novel way to incorporate existing network information into the process of gene network inference, we believe it offers a more generic mechanism to incorporate other types of constraints. For example, research has indicated that different transcription factors have different occupancy levels in the promoter regions of the genes they regulate [20]. This might be modeled as constraints in our framework whereby transcription factors with low occupancy are given a high W_{ij}^0 value while a high occupancy transcription factor is given a low W_{ij}^0 value. Of course this makes the assumption, possibly incorrectly, that the occupancy level has a proportional effect on transcription rate, but this may be an appropriate approximation for some systems. In the future, we look forward to investigating the many ways that we can incorporate the growing body of biological data into our optimization framework.

Supporting Information

Appendix S1 Proofs of theorems
Found at: doi:10.1371/journal.pone.0006799.s001 (0.20 MB DOC)

Text S1 Complete results for embryonic stem cell network
Found at: doi:10.1371/journal.pone.0006799.s002 (0.08 MB DOC)

Acknowledgments

We thank Qiang Liu for careful reading of the manuscript.

Author Contributions

Conceived and designed the experiments: SC QN XX. Performed the experiments: SC. Analyzed the data: SC QN XX. Contributed reagents/materials/analysis tools: SC QN XX. Wrote the paper: SC QN XX.

References

1. Friedman N (2004) Inferring cellular networks using probabilistic graphical models. Science 303: 799–805.
2. 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 Biol 5: e8.
3. Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, et al. (2006) Reverse engineering cellular networks. Nat Protoc 1: 662–671.
4. Gustafsson M, Hornquist M, Lombardi A (2005) Constructing and analyzing a large-scale gene-to-gene regulatory network Lasso-constrained inference and biological validation. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2: 254–261.
5. Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301: 102–105.
6. de Jong H (2002) Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 9: 67–103.
7. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D (2007) How to infer gene networks from expression profiles. Mol Syst Biol 3: 78.
8. Gilbert D, Fuss H, Gu X, Orton R, Robinson S, et al. (2006) Computational methodologies for modelling, analysis and simulation of signalling networks. Brief Bioinformatics 7: 339–353.
9. Bonneau R (2008) Learning biological networks: from modules to dynamics. Nat Chem Biol 4: 658–664.
10. 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 Biol 7: R36.
11. Li H, Zhan M (2008) Unraveling transcriptional regulatory programs by integrative analysis of microarray and transcription factor binding data. Bioinformatics 24: 1874–1880.
12. Pilpel Y, Sudarsanam P, Church GM (2001) Identifying regulatory networks by combinatorial analysis of promoter elements. Nat Genet 29: 153–159.
13. Segal E, Raveh-Sadka T, Schroeder M, Unnerstall U, Gaul U (2008) Predicting expression patterns from regulatory sequence in Drosophila segmentation. Nature 451: 535–540.
14. Tamada Y, Kim S, Bannai H, Imoto S, Tashiro K, et al. (2003) Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. Bioinformatics 19 Suppl 2: ii227–236.
15. Mordelet F, Vert J (2008) SIRENE: supervised inference of regulatory networks. Bioinformatics 24: i76–82.
16. Wang Y, Joshi T, Zhang XS, Xu D, Chen L (2006) Inferring gene regulatory networks from multiple microarray datasets. Bioinformatics 22: 2413–2420.
17. Tan K, Tegner J, Ravasi T (2008) Integrated approaches to uncovering transcription regulatory networks in mammalian cells. Genomics 91: 219–231.
18. Mardis ER (2007) ChIP-seq: welcome to the new frontier. Nat Methods 4: 613–614.
19. Jiang C, Pugh BF (2009) Nucleosome positioning and gene regulation: advances through genomics. Nat Rev Genet 10: 161–172.
20. Kim J, Chu J, Shen X, Wang J, Orkin SH (2008) An extended transcriptional network for pluripotency of embryonic stem cells. Cell 132: 1049–1061.
21. Zhou Q, Chipperfield H, Melton DA, Wong WH (2007) A gene regulatory network in mouse embryonic stem cells. Proc Natl Acad Sci USA 104: 16438–16443.
22. Chen X, Xu H, Yuan P, Fang F, Huss M, et al. (2008) Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. Cell 133: 1106–1117.
23. Meissner A, Mikkelsen TS, Gu H, Wernig M, Hanna J, et al. (2008) Genome-scale DNA methylation maps of pluripotent and differentiated cells. Nature 454: 766–770.
24. Mikkelsen TS, Ku M, Jaffe DB, Issac B, Lieberman E, et al. (2007) Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. Nature 448: 553–560.
25. Bernstein BE, Meissner A, Lander ES (2007) The mammalian epigenome. Cell 128: 669–681.

26. Chen Y, Blackwell T, Chen J, Gao J, Lee A, et al. (2007) Integration of genome and chromatin structure with gene expression profiles to predict c-MYC recognition site binding and function. PLoS Comput Biol 3: e63.
27. Sharov A, Masui S, Sharova L, Piao Y, Aiba K, et al. (2008) Identification of Pou5f1, Sox2, and Nanog downstream target genes with statistical confidence by applying a novel algorithm to time course microarray and genome-wide chromatin immunoprecipitation data. BMC Genomics 9: 269.
28. Mukherjee S, Speed TP (2008) Network inference using informative priors. Proc Natl Acad Sci USA 105: 14313–14318.
29. Werhli AV, Husmeier D (2008) Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. Journal of bioinformatics and computational biology 6: 543–572.
30. Bansal M, Gatta GD, di Bernardo D (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. Bioinformatics 22: 815–822.
31. Kennard RW (2000) Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics. pp 80–86.
32. Tibshirani R (1996) Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society Series B (Methodological) 58: 267–288.
33. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32: 407–451.
34. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics (Oxford, England) 9: 432–441.
35. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J Roy Stat Soc B 67: 301–320.
36. Friedman J, Hastie T, Tibshirani R (2008) Regularization Paths for Generalized Linear Models via Coordinate Descent. http://www-stat.stanford.edu/~hastie/Papers/glmnet.pdf.
37. Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. J Optimiz Theory App 109: 475–494.
38. Knight K, Fu W (2000) Asymptotics for Lasso-Type Estimators. Ann Stat 28: 1356–1378.
39. Banerjee O, Ghaoui LE, d'Aspremont A, Natsoulis G (2006) Convex optimization techniques for fitting sparse Gaussian graphical models. ICML '06: Proceedings of the 23rd international conference on Machine learning. pp 89–96.
40. Takahashi K, Yamanaka S (2006) Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. Cell 126: 663–676.
41. Okita K, Ichisaka T, Yamanaka S (2007) Generation of germline-competent induced pluripotent stem cells. Nature 448: 313–317.
42. Wernig M, Meissner A, Foreman R, Brambrink T, Ku M, et al. (2007) In vitro reprogramming of fibroblasts into a pluripotent ES-cell-like state. Nature 448: 318–324.
43. Okita K, Nakagawa M, Hyenjong H, Ichisaka T, Yamanaka S (2008) Generation of mouse induced pluripotent stem cells without viral vectors. Science 322: 949–953.
44. Park I, Zhao R, West JA, Yabuuchi A, Huo H, et al. (2008) Reprogramming of human somatic cells to pluripotency with defined factors. Nature 451: 141–146.
45. Stadtfeld M, Nagaya M, Utikal J, Weir G, Hochedlinger K (2008) Induced pluripotent stem cells generated without viral integration. Science 322: 945–949.
46. Yu J, Vodyanik MA, Smuga-Otto K, Antosiewicz-Bourget J, Frane JL, et al. (2007) Induced pluripotent stem cell lines derived from human somatic cells. Science 318: 1917–1920.
47. Li XY, MacArthur S, Bourgon R, Nix D, Pollard DA, et al. (2008) Transcription factors bind thousands of active and inactive regions in the Drosophila blastoderm. PLoS Biol 6: e27.
48. Ivanova N, Dobrin R, Lu R, Kotenko I, Levorse J, et al. (2006) Dissecting self-renewal in stem cells with RNA interference. Nature 442: 533–538.
49. Tsubooka N, Ichisaka T, Okita K, Takahashi K, Nakagawa M, et al. (2009) Roles of Sall4 in the generation of pluripotent stem cells from blastocysts and fibroblasts. Genes to Cells 14: 683–694.
50. Zhang X, Zhang J, Wang T, Esteban MA, Pei D (2008) Esrrb activates Oct4 transcription and sustains self-renewal and pluripotency in embryonic stem cells. J Biol Chem 283: 35825–35833.
51. Donohoe ME, Silva SS, Pinter SF, Xu N, Lee JT (2009) The pluripotency factor Oct4 interacts with Ctcf and also controls X-chromosome pairing and counting. Nature 460: 128–132.

Appendix S1

Proof of Theorems

Consider a matrix Λ of non-negative entries and a parameter for L_1-norm regularization:

$$\hat{W} = \arg\min_{W} g(W) = f(W) + \lambda \left\| \Lambda \circ W \right\|_1 = \sum_{j=1}^{n} \left\| W x^j - u^j \right\|^2 + \lambda \left\| \Lambda \circ W \right\|_1 \tag{1.0}$$

This equation is equivalent to:

$$\hat{W} = \arg\min_{W} g(W) = \mathrm{tr}(W^T W \Sigma) - 2\mathrm{tr}(W U^T) + \lambda \left\| \Lambda \circ W \right\|_1 \tag{1.1}$$

with matrices Σ and U defined to be:

$$\Sigma = \sum_{j=1}^{n} x^j \left(x^j\right)^T \quad U = \sum_{j=1}^{n} x^j \left(u^j\right)^T \tag{1.2}$$

Consider the following noise model:

$$u = Wx + \varepsilon \tag{1.3}$$

where the noise term $\varepsilon \sim N(0, \sigma^2 I)$ follows normal distribution with a fixed variance.

Theorem 1

If $\lambda / \sqrt{n} \to \lambda_0 \geq 0, \Lambda = O(1)$*, and*

$$C = \lim_{n \to \infty} \left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) \tag{1.4}$$

is non-singular, then when $n \to \infty$*,*

$$\sqrt{n}\left(\hat{W} - W\right) \xrightarrow{D} \arg\min(V) \tag{1.5}$$

where

$$V(Z) = \mathrm{tr}(Z^T Z C) - 2\mathrm{tr}(Z \Psi^T) + \lambda_0 \mathrm{tr}\left\{ \left[I(W \neq 0) \circ \mathrm{sgn}(W) \circ Z + I(W = 0) \circ |Z| \right] \Lambda^T \right\} \tag{1.6}$$

and Ψ *is a random matrix with normal distribution of mean 0 and covariance* $E\left[\Psi_{ij}\Psi_{kl}\right] = \Sigma_{ik}\delta_{jl}\sigma^2$ *for all i, j, k, l.*

Proof. Let $\Delta\hat{W} = \hat{W} - W$. Then it follows from Eq. (1.1) that $\Delta\hat{W}$ is the solution of

$$\Delta\hat{W} = \arg\min_{\Delta W} g(\Delta W) = n\,\mathrm{tr}\left(\Delta W^T \Delta W \Sigma\right) - 2\sqrt{n}\,\mathrm{tr}\left(\Delta W \Psi^T\right) + \lambda\left\|\Lambda \circ (W + \Delta W)\right\|_1 \tag{1.7}$$

where

$$\Psi = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \varepsilon_i x_i^T \tag{1.8}$$

is a random matrix with mean 0 and covariance $E\left[\Psi_{ab}\Psi_{cd}\right] = C_{bd}^m \delta_{ac}\sigma^2$ for all *a, b, c, d*.

Here $C^m = \sum_{i=1}^{n} x_i x_i^T / n$ is the empirical covariance matrix of the input data. As $n \to \infty, C^m \to C$ and $\lambda \to \sqrt{n}\lambda_0$. Consequently, we must have $\Delta\hat{W} \to 0$ in Eq. (1.7) almost surely when *C* is non-singular. When $\Delta W \to 0$, the last term in the RHS of Eq. (1.7) can be rewritten as

$$\left\|\Lambda \circ (W + \Delta W)\right\|_1 = \mathrm{tr}\left\{\left[I(W \neq 0) \circ \mathrm{sgn}(W) \circ \Delta W + I(W = 0) \circ \left|\Delta W\right|\right]\Lambda^T\right\} \tag{1.9}$$

Now define $Z = \sqrt{n}\Delta W$. Then as $n \to \infty, \sqrt{n}\left(\hat{W} - W\right)$ minimizes

$$\mathrm{tr}\left(Z^T Z C\right) - 2\,\mathrm{tr}\left(Z\Psi^T\right) + \lambda_0\,\mathrm{tr}\left\{\left[I(W \neq 0) \circ \mathrm{sgn}(W) \circ \Delta W + I(W = 0) \circ \left|\Delta W\right|\right]\Lambda^T\right\} \tag{1.10}$$

□

Given a matrix Λ of non-negative entries and a single non-negative parameter λ, we can formulate the following minimization problem:

$$\hat{\Omega} = \arg\min_{\Omega \succ 0} f(\Omega) = -\frac{n}{2}\log\det(\Omega) + \frac{n}{2}tr(S\Omega) + \lambda\left\|\Lambda \circ \Omega\right\|_1 \tag{1.11}$$

where $\Lambda \circ \Omega$ is the component-wise product of the two matrices.

Theorem 2

If $\lambda/\sqrt{n} \to \lambda_0 \geq 0, \Lambda = O(1)$*, and* Σ *is non-singular, then*

$$\sqrt{n}\left(\hat{\Omega} - \Omega\right) \xrightarrow{D} \arg\min(V) \tag{1.12}$$

where

$$V(\Delta) = \frac{1}{4}\mathrm{tr}(\Sigma\Delta\Sigma\Delta) + \frac{1}{2}\mathrm{tr}(Z\Delta) + \lambda_0 \mathrm{tr}\left\{\left[I(\Omega = 0) \circ |\Delta| + I(\Omega \neq 0) \circ \mathrm{sgn}(\Omega) \circ \Delta\right]\Lambda^T\right\} \tag{1.13}$$

and Z is a random matrix with normal distribution of mean 0 and covariance

$E\left[Z_{ij}Z_{kl}\right] = \Sigma_{ij}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}$ *for all i, j, k, l.*

Proof. Let $\hat{Y} = \hat{\Omega} - \Omega$. Then from Eq. (1.11) we have:

$$\hat{Y} = \arg\min_{Y} f(Y) = -\frac{n}{2}\log\det(\Omega + Y) + \frac{n}{2}\mathrm{tr}(SY) + \lambda\|\Lambda \circ (\Omega + Y)\|_1 \tag{1.14}$$

$S = \sum_{j=1}^{n} x^j \left(x^j\right)^T / n$ is a random variable. When $n \to \infty$, S follows normal distribution and converges in distribution to:

$$S = \Sigma + \frac{1}{\sqrt{n}}Z \tag{1.15}$$

where Z follows normal distribution with mean 0 and covariance

$E\left[Z_{ij}Z_{kl}\right] = \Sigma_{ij}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}$ for all $i, j, k, l \in [1, p]$. Hence, we have:

$$n\,\mathrm{tr}(SY) = n\,\mathrm{tr}(\Sigma Y) + \mathrm{tr}(\sqrt{n}YZ) \tag{1.16}$$

When $n \to \infty, Y \to 0$ almost surely. Thus, we can perform Taylor expansion on $\log\det(\Omega + Y)$ around Ω which leads to:

$$\begin{aligned} \log\det(\Omega+Y) &= \log\det(\Omega)+\operatorname{tr}(\Omega^{-1}Y)-\frac{1}{2}\operatorname{tr}(\Omega^{-1}Y\Omega^{-1}Y)+o\left(\|Y\|_F^2\right) \\ &= \log\det(\Omega)+\operatorname{tr}(\Sigma Y)-\frac{1}{2}\operatorname{tr}(\Sigma Y\Sigma Y)+o\left(\|Y\|_F^2\right) \end{aligned} \tag{1.17}$$

When Y is small, the term $\left|\Lambda\circ\left(\Omega+Y\right)\right|$ can be rewritten as:

$$\left|\Lambda\circ\left(\Omega+Y\right)\right|_{ij}=\begin{cases} \Lambda_{ij}\left|Y_{ij}\right| & \Omega_{ij}=0 \\ \Lambda_{ij}\operatorname{sgn}(\Omega_{ij})(\Omega_{ij}+Y_{ij}) & \Omega_{ij}\neq 0 \end{cases} \tag{1.18}$$

Substituting Eqs. (1.16), (1.17) and (1.18) into Eq. (1.14) and ignoring terms independent of Y, we have:

$$\begin{aligned} &\hat{Y}=\arg\min_{Y} f(Y)=\frac{1}{4}\operatorname{tr}\left(\Sigma\sqrt{n}Y\Sigma\sqrt{n}Y\right)+\frac{1}{2}\operatorname{tr}\left(\sqrt{n}YZ\right) \\ &+\lambda_0\sum_{i=1}^{n}\sum_{j=1}^{n}\left[I\left(\Omega_{ij}=0\right)\Lambda_{ij}\left|\sqrt{n}Y_{ij}\right|+I\left(\Omega_{ij}\neq 0\right)\Lambda_{ij}\operatorname{sgn}\left(\Omega_{ij}\right)\sqrt{n}Y_{ij}\right] \end{aligned} \tag{1.19}$$

If we define $\Delta=\sqrt{n}Y=\sqrt{n}\left(\hat{\Omega}-\Omega\right)$, then we have $\hat{\Delta}=\arg\min_{\Delta} V\left(\Delta\right)$.

□