ROBUST REVERSE ENGINEERING OF DYNAMIC GENE NETWORKS UNDER SAMPLE SIZE HETEROGENEITY

ANKUR P. PARIKH, WEI WU, ERIC P. XING

School of Computer Science, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
**E-mail: apparikh@cs.cmu.edu, weiwu2@cs.cmu.edu, epxing@cs.cmu.edu*

Simultaneously reverse engineering a collection of condition-specific gene networks from gene expression microarray data to uncover dynamic mechanisms is a key challenge in systems biology. However, existing methods for this task are very sensitive to variations in the size of the microarray samples across different biological conditions (which we term *sample size heterogeneity in network reconstruction*), and can potentially produce misleading results that can lead to incorrect biological interpretation. In this work, we develop a more robust framework that addresses this novel problem. Just like microarray measurements across conditions must undergo proper normalization on their magnitudes before entering subsequent analysis, we argue that networks across conditions also need to be "normalized" on their density when they are constructed, and we provide an algorithm that allows such normalization to be facilitated while estimating the networks. We show the quantitative advantages of our approach on synthetic and real data. Our analysis of a hematopoietic stem cell dataset reveals interesting results, some of which are confirmed by previously validated results.

Keywords: gene network reconstruction, dynamic, sample size heterogeneity

1. Introduction

Capturing and understanding the differential usage (i.e. rewiring) of cellular pathways and regulatory structures as a result of various biological processes and responses to external stimuli is an important problem in systems biology. Some examples include embryonic development, cell cycle, differentiation, and carcinogenesis. One promising technique to help uncover complex gene interactions governing these processes is to use computational methods to reverse engineer gene networks from microarray data. The macro-topology of the recovered network as well as the individual interactions among the genes can then be analyzed to shed more light into the underlying regulatory mechanisms.

To model the evolving nature of these phenomena, it often does not suffice to reconstruct one static snapshot of the underlying regulatory structure since this cannot uncover dynamic functional roles played by various genes in different cellular stages or at different times. Consider an example of the human hematopoietic system shown in Figure 1. Hematopoietic stem cells (located at the root) differentiate into more specialized cells along the lineages, eventually becoming red blood cells, platelets, or white blood cells. It would be inappropriate to pool together various samples to reconstruct a single network representing a common regulatory structure for different cell states, e.g., red and white blood cells, since they have distinct morphologies and play completely different roles in biological systems, and thus their respective regulatory structures must also be considerably different. Instead it is more suitable to reconstruct a *collection* of networks, one for each cell state. Different functional roles of various genes across the different cell states can then be analyzed.

However, the problem of simultaneously recovering a collection of networks over different cell states poses unique challenges that do not appear in the static recovery case. The key challenge we face in this work is that different cell states have different numbers of microarray samples, which we term *sample size heterogeneity in network reconstruction.* This phenomenon is quite common in biological datasets due to a variety of reasons such as samples having to be discarded if the quality of the microarrays is poor, or constrains on acquisition of certain biomedical samples.

Even though sample size heterogeneity can pose considerable challenges for many existing network reconstruction methods in different ways, in this work we choose to focus on addressing its effect on a class of state-of-the-art methods that are based on sparse, regularized regression.[1–3] These methods are designed for the high dimensional setting common in biology, where the number of genes can be substantially larger than the number of samples, and allow us to uncover more sophisticated dependencies than can be obtained by measuring simpler quantities such as correlation or mutual information. Building upon the regularized regression based network learning paradigm, several methods[4–6] have recently proposed leveraging similarities of multiple networks corresponding to biological conditions considered to be related for more accurate multi-network joint estimation, under evolving network scenarios. This strategy is very valuable in the scenario we consider in this work, where the number of samples for each cell state is small (e.g., as few as 4 per cell state, clearly statistically insignificant for inferring a network alone), and thus information sharing between related cell states is crucial and can increase the effective sample size and consequently the power of network learning. Such methods have helped reveal the dynamic interactions in embryonic development[4] as well as cancer progression and reversion.[6]

Despite being statistically powerful, network learning approaches based on regularized regression can suffer from sample size heterogeneity, which can substantially bias the density of the networks recovered. In particular, with existing sparse regression methods, cell states with more samples will tend to have considerably denser networks than those with fewer samples, a phenomena depicted in Figure 1. Intuitively, this is because the algorithm is more confident about estimating networks with more samples and thus these networks are denser.

The resultant artificial difference may be acceptable in certain applications (e.g. features for a downstream classifier). However, in many cases, we are interested in a comparative analysis of the networks, both in terms of macro-topology (e.g. density, centrality) or micro-topology (e.g. neighborhoods of individual genes). In this scenario, sample size heterogeneity can lead to misleading biological conclusions, since it will be unclear which differences among the networks are manifestations of the actual changes in regulatory mechanisms across different cell states and which are the artifacts due to sample size heterogeneity.

One simple approach to handle sample size heterogeneity is to make each cell state have the same number of samples by discarding excess samples in some states. The downside of this approach is the waste of the precious data in the small-sample-size scenarios common in biological studies. For example, in the hematopoietic stem cell dataset we consider, using this strategy would lead to a reduction of the total sample size by approximately 40 percent.

Another approach is to post-process the networks to be more calibrated, e.g. normalizing all the edge weights across the cell states and then applying some threshold. However, this may produce adverse effects. Namely, since edges can only be *deleted*, and not *added* during post-processing, the original networks learned using sparse regression have to be denser than desired, and then further sparsified via post-processing. The resulting edge set from this procedure would then be suboptimal compared to the edge set constructed by just learning a sparser network with the regularized regression.

1.1. *Our Contribution*

In this work, we identify a novel problem of sample size heterogeneity, which to our knowledge has not been systematically analyzed or addressed before. Although it can affect many classes of network estimation algorithms, we focus on a class of sparse regression methods for dynamic network reconstruction, and propose a solution to address the challenge in this paradigm. In particular, we propose a novel regularization technique to ensure the resulting networks are balanced and thus more easily comparable. We refer to our approach as *ROMGL* (*RObust Multi-network Graphical Lasso*).

The important novelty we emphasize here is that our network calibration is not introduced as a post-processing of the inferred networks, but an integral part within the network inference procedure, in the form of a new and calibrated network estimator, and therefore more effective and statistically justifiable.

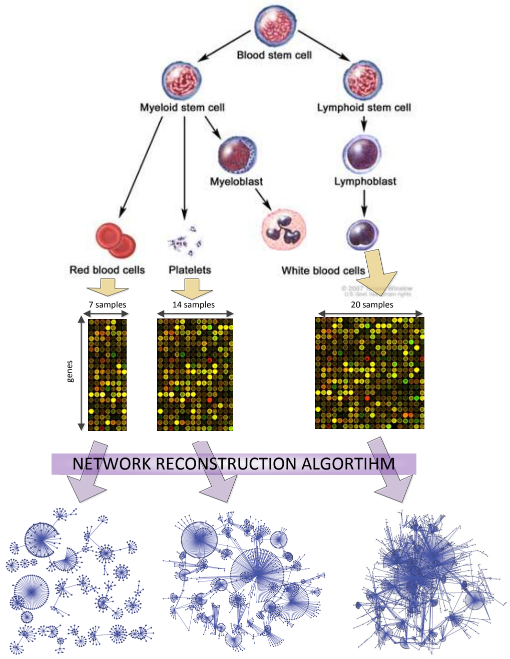


Fig. 1. Illustration of a hematopoietic stem cell genealogy and how more samples bias existing reconstruction methods to give artificially denser networks. [a]

The rest of the work is outlined as follows. We first present the general framework of reconstructing gene networks via sparse regression methods and concretely illustrate the problem that sample size heterogeneity poses. We then present our robust method. Lastly, we evaluate our approach on synthetic data as well as on a human hematopoietic stem cell dataset.

2. Background: Recovering Gene Networks via Gaussian Graphical Models

Consider the problem of modeling a set of gene regulatory networks, denoted by $\mathcal{Z}$ (where $|\mathcal{Z}| = Z$), each corresponding to a different cell state $z \in \mathcal{Z}$ with S_z *i.i.d.* microarray measurements of all genes in cell state z. $\mathcal{Z}$ could represent a set of networks over time or over a genealogy. Let $\mathcal{G}^{(z)} = (\mathcal{V}, \mathcal{E}^{(z)})$ represent a network in cell state z, where $\mathcal{V}$ denotes the set of p genes (fixed for all z), and $\mathcal{E}^{(z)}$ denotes the set of edges over vertices. An edge $(u, v) \in \mathcal{E}^{(z)}$ can

[a] http://www.siteman.wustl.edu/CancerDetails.aspx?id=661&xml=CDR257990.xml

represent a relationship (e.g., influence or interaction) between genes u and v in cell state z. Let $\boldsymbol{X}^{(s,z)} = (X_1^{(s,z)}, \ldots, X_p^{(s,z)})'$, where $s \in \{1, \ldots, S_z\}$, be a vector of gene expression values that are real valued and standardized, such that each dimension has mean 0 and variance 1.

A gene network can be represented by a probabilistic graphical model.[7,8] While there are many other ways to represent gene networks, the advantage of using graphical models is that the graph structure encodes conditional independence relations among the genes, and is thus able to model more nuanced relationships than simple statistical quantities such as correlation or mutual information. In this work, we assume that $\boldsymbol{X}^{(z)}$ follows a multivariate Gaussian distribution with mean 0 and covariance matrix $\Sigma^{(z)}$, so that the conditional independence relationships among the genes can be encoded as a Gaussian graphical model (GGM).[9] It is well known that for GGMs, edges in the graph correspond to non-zero elements in the inverse covariance matrix (known as the precision matrix), which we denote by $\boldsymbol{\Omega}^{(z)} := (\omega_{uv}^{(z)})_{u,v\in[p]}$. Thus, estimating the graph structure is equivalent to selecting the non-zero elements of the precision matrix.

As commonly done, instead of directly estimating the precision matrix elements $\omega_{uv}^{(z)}$, we estimate the partial correlation coefficients $\boldsymbol{\rho}^{(z)}$, which are proportional to the precision matrix elements: $\rho_{uv}^{(z)} = -\frac{\omega_{uv}^{(z)}}{\sqrt{\omega_{uu}^{(z)}\omega_{vv}^{(z)}}}$. Thus, $\rho_{uv}^{(z)}$ is zero if and only if $\omega_{uv}^{(z)}$ is zero. Thus the network resultant from the non-zero $\rho_{uv}^{(z)}$ is equivalent to that from the nonzero $\omega_{uv}^{(z)}$. Furthermore, the partial correlation is intuitive in the sense that a high positive value of $\rho_{uv}^{(z)}$ indicates that the genes u and v are strongly positively correlated (conditioned on the other genes), while a low negative value indicates the genes are strongly negatively correlated (conditioned on the other genes), and $\rho_{uv}^{(z)} = 0$ for all $(u, v) \notin \mathcal{E}^{(z)}$. As a result, we simply consider estimating the partial correlation coefficients and designate these as the edge values in $\mathcal{G}^{(z)}$: $\mathcal{E}^{(z)} = \{\rho_{uv}^{(z)} : |\rho_{uv}^{(z)}| > 0\}$.

2.1. *Neighborhood Selection*

Estimating $\boldsymbol{\rho}_{uv}$ is challenging because biological data is often high dimensional (tens of thousands of genes) while the number of samples is small (in the tens). One approach is neighborhood selection[2] based on ℓ_1-norm regularized regression, which has strong theoretical guarantees and also works well in practice. We first discuss it in the context of estimating a collection of networks independently, which is also the foundation of existing approaches on time-varying network estimation that leverage information among similar states.[4–6]

Here the neighborhood of each gene u is estimated independently and the neighborhoods are then combined to form a network. In every neighbor estimation step, gene u is treated as a response variable, all the other genes are the covariates, and the regression weights are proportional to the partial correlation coefficients between the other genes and u. More formally, let $\boldsymbol{X}_{\setminus u}$ indicate the $p-1$ vector of the values of all genes except u. Similarly, $\boldsymbol{\beta}_{\setminus u} := \{\beta_{uv} : v \in \mathcal{V} \setminus u\}$. It is a well known result , that the partial correlation coefficients can be related to the following regression model[10] : $X_u^{(z)} = \sum_{v\neq u} X_v^{(z)}\beta_{uv}^{(z)} + \epsilon_u^{(z)}, \quad u \in [p]$, where $\epsilon_u^{(z)}$ is uncorrelated with $\boldsymbol{X}_{\setminus u}^{(z)}$ if and only if $\beta_{uv}^{(z)} = -\frac{\omega_{uv}^{(z)}}{\omega_{uu}^{(z)}} = \rho_{uv}^{(z)}\sqrt{\frac{\omega_{vv}^{(z)}}{\omega_{uu}^{(z)}}}$. Some algebra gives that $\rho_{uv}^{(z)} = \mathrm{sign}(\beta_{uv}^{(z)})\sqrt{\beta_{uv}^{(z)}\beta_{vu}^{(z)}}$. The above equations basically indicate that we can solve for the regression coefficients using a linear regression, where the response variable corresponds to

X_u and the covariates correspond to $\boldsymbol{X}_{\setminus u}$. The corresponding partial correlation coefficients can be recovered via the algebraic relations. An ℓ_1 penalty is applied to encourage a sparse solution, as in the lasso.[1] We can estimate the neighborhood of gene u for all cell states $z \in \mathcal{Z}$ using this strategy, as depicted in Eq. 1.

$$\hat{\boldsymbol{\beta}}^{(1)}_{\setminus u}, ..., \hat{\boldsymbol{\beta}}^{(Z)}_{\setminus u} = \underset{\boldsymbol{\beta}^{(1)}_{\setminus u}, ..., \boldsymbol{\beta}^{(Z)}_{\setminus u}}{\operatorname{argmin}} \sum_{z \in \mathcal{Z}} \mathcal{L}^u(\boldsymbol{X}^{(z)}, \boldsymbol{\beta}^{(z)}_{\setminus u}) + \lambda \sum_{z \in \mathcal{Z}} \|\boldsymbol{\beta}^{(z)}_{\setminus u}\|_1 \tag{1}$$

where $\mathcal{L}^u(\boldsymbol{X}^{(z)}, \boldsymbol{\beta}^{(z)}_{\setminus u}) := \sum_{s=1}^{S_z} \left(x_u^{(s,z)} - \sum_{v \neq u} \beta^{(z)}_{uv} x_v^{(s,z)} \right)^2$. Note that the optimization problem decouples into Z separate problems. This procedure is repeated to estimate the neighborhood of every gene $u \in \mathcal{V}$. It has been shown that under certain conditions, one can obtain an estimator of the edge set $\mathcal{E}$ that is *sparsistent*,[2,11] i.e. the correct network structure can be attained as a function of the number of genes, samples, and topology of the network.

2.2. *Neighborhood Selection and Sample Size Heterogeneity*

However, applying the same λ to all $z \in \mathcal{Z}$ such as in Eq. 1 can be problematic under sample size heterogeneity. Consider two cell states z_1 and z_2 and assume that $S_{z_1} > S_{z_2}$. This implies that $\mathcal{L}^u(\boldsymbol{X}^{(z_1)}, \boldsymbol{\beta}^{(z_1)}_{\setminus u} = \boldsymbol{0})$ will generally be larger than $\mathcal{L}^u(\boldsymbol{X}^{(z_2)}, \boldsymbol{\beta}^{(z_2)}_{\setminus u} = \boldsymbol{0})$. Applying the same λ to both of them will then tend to lead to a more sparse solution for z_2 than z_1. This is because networks with different sample sizes should be learned with different amounts of regularization.

At first glance, it seems simple scaling/normalization (such as dividing $\mathcal{L}^u(\boldsymbol{X}^{(z)}, \boldsymbol{\beta}^{(z)}_{\setminus u})$ by S_z) would be sufficient. Asymptotic theory[12] dictates that in addition to dividing each $\mathcal{L}^u(\boldsymbol{X}^{(z)}, \boldsymbol{\beta}^{(z)}_{\setminus u})$ by S_z, λ should be divided by $\sqrt{S_z}$ as shown in Eq 2:

$$\hat{\boldsymbol{\beta}}^{(1)}_{\setminus u}, ..., \hat{\boldsymbol{\beta}}^{(Z)}_{\setminus u} = \underset{\boldsymbol{\beta}^{(1)}_{\setminus u}, ..., \boldsymbol{\beta}^{(Z)}_{\setminus u}}{\operatorname{argmin}} \left(\sum_{z \in \mathcal{Z}} \frac{1}{S_z} \mathcal{L}^u(\boldsymbol{X}^{(z)}, \boldsymbol{\beta}^{(z)}_{\setminus u}) + \sum_{z \in \mathcal{Z}} \frac{\lambda}{\sqrt{S_z}} \|\boldsymbol{\beta}^{(z)}_{\setminus u}\|_1 \right) \tag{2}$$

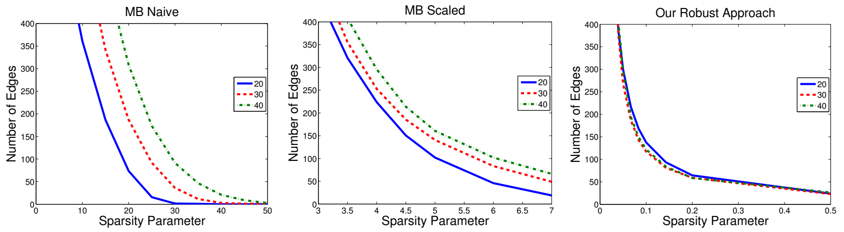
However, this scaling is based on several theoretical assumptions on the underlying model. As a result, it may behave erratically in practice on microarray data as we show in Section 7. Even when all the theoretical assumptions hold, the $\sqrt{S_z}$ factor is correct only *asymptotically*, and not necessarily for smaller sample sizes. To illustrate the problem, we present an example shown in Figure 2. (More quantitative results will be given in Section 6.) Here, a single network with 100 vertices and 200 edges was randomly generated. Then, 10 sets with 20 samples, 10 sets with 30 samples, and 10 sets with 40 samples were generated, all from the same network. We vary the sparsity parameter λ, and plot the mean edge count for each sample size. Figure 2(a) shows the results of optimizing Eq. 1 without scaling [a]. As one can see, although all the samples were generated from the same network, the networks learned from the 40 samples have many more edges than those from fewer samples. Figure 2(b) shows the results for optimizing Eq. 2 (with scaling). This works better, but networks learned from the 40 samples still have considerably more edges than those from 20.

One possible strategy is to assign each network a different regularization parameter and tune these manually according to known biological interactions. Unfortunately, this requires

[a] MB stands for Meinshausen and Buhlmann who proposed neighborhood selection[2] for GGMs.

that we have enough prior knowledge about *all* the networks, which is unlikely for many systems. Instead, it is preferable to develop an approach that only requires prior knowledge about a small subset of the networks for the purposes of parameter tuning.

3. A More Robust Formulation



(a) MB Naive (b) MB Scaled (c) ROMGL

Fig. 2. Comparison of non-robust vs. robust approaches on a simple example. Our robust approach, *ROMGL*, produces networks that are much more balanced than the naive and scaled methods (MB Naive and MB Scaled). See text for details.

In order to calibrate the networks to mitigate the artifacts caused by sample size heterogeneity, we propose the following approach. We require that the sum of the absolute edge weights to be the same for all networks reconstructed. This is in some sense similar to the assumptions made in microarray data pre-processing via *normalization* which rely on less than ideal yet necessary assumptions in order to remove systematic dye bias from the data, e.g., quantile normalization in RMA assumes an identical distribution of gene expression values in all samples in a dataset.[13,14]

Rather than post-processing the networks, we integrate this assumption into our network algorithm, thus allowing for a more principled and effective approach.

Unfortunately, it is difficult to directly modify neighborhood selection described in the previous section to incorporate this assumption, because we are constraining the *entire* networks to have the same sum of absolute edge weights, rather than the individual neighborhoods. The former assumption is much more realistic, since the latter implies all the nodes have similar degrees. However, since neighborhood selection estimates each neighborhood independently, it cannot incorporate this assumption in its procedure. Instead, we build our solution from SPACE[15] which is a procedure that simultaneously performs neighborhood selection on all neighborhoods. First define,

$$\mathcal{M}(\boldsymbol{X}^{(z)},\boldsymbol{\rho}^{(z)},\boldsymbol{\sigma}^{(z)}) := \sum_{u\in\mathcal{V}}\sum_{s=1}^{S_z}\left(x_u^{(s,z)} - \sum_{v\neq u}\boldsymbol{\beta}_{uv}^{(z)}x_v^{(s,z)}\right)^2 = \sum_{u\in\mathcal{V}}\sum_{s=1}^{S_z}\left(x_u^{(s,z)} - \sum_{v\neq u}\rho_{uv}^{(z)}\sqrt{\frac{\sigma_{vv}}{\sigma_{uu}}}x_v^{(s,z)}\right)^2 \tag{3}$$

Then, using SPACE to estimate each network $z \in \mathcal{Z}$ separately will give the following optimization problem:

$$\begin{aligned}\hat{\boldsymbol{\rho}}^{(1)},...,\hat{\boldsymbol{\rho}}^{(Z)} &= \underset{\boldsymbol{\rho}^{(1)},...,\boldsymbol{\rho}^{(\mathcal{Z})}}{\operatorname{argmin}}\left(\sum_{z\in\mathcal{Z}}\frac{1}{S_z}\mathcal{M}(\boldsymbol{X}^{(z)},\boldsymbol{\rho}^{(z)},\boldsymbol{\sigma}^{(z)})\sum_{z\in\mathcal{Z}}\frac{\lambda}{\sqrt{S_z}}\|\boldsymbol{\rho}^{(z)}\|_1\right)\\ \text{subject to} \quad & \rho_{uv}^{(z)} = \rho_{vu}^{(z)} \quad \forall z, \forall u \neq v \end{aligned} \tag{4}$$

Similar to the previous sections, the objective above decouples into Z separate problems.

Here $\sigma_{uu}^{(z)} = 1/var(\epsilon_u^{(z)})$, where $\epsilon_u^{(z)}$ was defined in Section 2.1. Note that SPACE estimates $\boldsymbol{\rho}$ directly instead of $\boldsymbol{\beta}$. This is because while $\rho_{uv}^{(z)} = \rho_{vu}^{(z)}$, $\beta_{uv}^{(z)} \neq \beta_{vu}^{(z)}$ due to the relation in Section 2.1.

Note that SPACE has the same problem as neighborhood selection with varying sample sizes. However, because we estimate all the neighborhoods jointly, we can propose a new formulation that enforces our assumption. This can be done by requiring the ℓ_1 norm of the absolute value of the edge weights to be equal to C for all $z \in \mathcal{Z}$.

$$\begin{aligned}\hat{\boldsymbol{\rho}}^{(1)}, ..., \hat{\boldsymbol{\rho}}^{(Z)} = \text{argmin}_{\boldsymbol{\rho}^{(1)},...,\boldsymbol{\rho}^{(Z)}} \textstyle\sum_{z \in \mathcal{Z}} \frac{1}{S_z} \mathcal{M}(\boldsymbol{X}^{(z)}, \boldsymbol{\rho}^{(z)}, \boldsymbol{\sigma}^{(z)}) \\ \text{subject to} \quad \rho_{uv}^{(z)} = \rho_{vu}^{(z)} \quad \forall z, \forall u \neq v, \quad \|\boldsymbol{\rho}^{(1)}\|_1 = C, \|\boldsymbol{\rho}^{(2)}\|_1 = C, ..., \|\boldsymbol{\rho}^{(Z)}\|_1 = C \end{aligned} \tag{5}$$

The formulation above represents the foundation of our approach, which we call *ROMGL* (*RObust Multi-network Graphical Lasso*). Note that this formulation is different than that in Eq. 4, because if we write it in Lagrangian form with λ's instead of constraints, then it is equivalent to a different λ for each constraint

$$\begin{aligned}\hat{\boldsymbol{\rho}}^{(1)}, ..., \hat{\boldsymbol{\rho}}^{(Z)} &= \underset{\boldsymbol{\rho}^{(1)},...,\boldsymbol{\rho}^{(Z)}}{\text{argmin}} \left(\sum_{z \in \mathcal{Z}} \frac{1}{S_z} \mathcal{M}(\boldsymbol{X}^{(z)}, \boldsymbol{\rho}^{(z)}, \boldsymbol{\sigma}^{(z)}) + \sum_{z \in \mathcal{Z}} \lambda_z \|\boldsymbol{\rho}^{(z)}\|_1 \right) \\ \text{subject to} &\quad \rho_{uv}^{(z)} = \rho_{vu}^{(z)} \quad \forall z, \forall u \neq v \end{aligned} \tag{6}$$

Moreover, without solving the optimization problem, the correspondence between C and the set of equivalent $\{\lambda_z\}_{z \in \mathcal{Z}}$ is unknown. Thus, the advantage of our approach is that we only have to explicitly set one parameter C instead of a different λ for each $z \in \mathcal{Z}$ (since $|\mathcal{Z}|$ might be quite large). We demonstrate our approach in Figure 2. Unlike the non-robust methods, our approach returns edge counts that are more similar across the different sample sizes.

4. Sharing Information Across States

So far, we have discussed robustly estimating a collection of networks without sharing information among different cell states. However, in the small-sample-size scenarios prevalent in regulatory genomics, this can result in poor estimation quality of the networks. For example, in the hematopoietic stem cell dataset we consider, some of the cell states have only 4 microarray samples, which is clearly statistically insufficient for reliable network estimation. However, since in many cases the gene networks are related, such as in a time series or a genealogy, we can leverage this interconnectedness of the networks for more accurate network reconstruction.

We assume we have *prior knowledge* of which networks are biologically related, and this information is encoded as a graph over the cell states $\mathcal{Z}$, which we denote by $\mathcal{H} = (\mathcal{Z}, \Gamma)$. $\mathcal{H}$ is constructed such that cell states closer to one another in the graph are assumed to be more biologically similar than those farther apart. For cells over a tree genealogy (e.g. stem cell differentiation), $\mathcal{H}$ represents a tree, and cell state z is connected to its parent and sibling cell states. As stated earlier, several methods[4–6] have recently proposed leveraging similarities of multiple networks for more accurate multi-network estimation. KELLER[4] proposes kernel smoothing, which estimates a given network by pooling a weighted average of related samples. TESLA and Treegl propose total variation regularization.[5,6]

However, these methods do not account for sample size heterogeneity. In fact, when sharing information among related states, robustness to sample size heterogeneity is even more crucial. This is because different cell states may have different numbers of neighbors in $\mathcal{H}$, and thus some may be able to share more information than others.

For simplicity, we only discuss how our robust formulation can be incorporated with kernel smoothing. Consider a smoothing kernel $K_h(z, y)$ that defines a similarity between cell state z and cell state y. We use the Epanechnikov kernel: $K_h(z, y) = 1 - \left(\frac{d(z,y)}{h}\right)^2$ if $\frac{d(z,y)}{h} \leq 1$, and 0 otherwise. Here we define $d(z, y)$ to be the shortest path from z to y in $\mathcal{H}$. Intuitively, this means that cell states closer to one another in the graph are assumed to be more biologically similar than those farther apart. Note that this is a more general setting than Song et al.,[4] who merely consider smoothing over time. We can then estimate a network for a cell state using a weighted average of samples from all cell states via the kernel:

$$\hat{\boldsymbol{\rho}}^{(1)}, ..., \hat{\boldsymbol{\rho}}^{(Z)} = \underset{\boldsymbol{\rho}^{(1)},...,\boldsymbol{\rho}^{(Z)}}{\operatorname{argmin}} \sum_{z \in \mathcal{Z}} \sum_{y \in \mathcal{Z}} K_h(z, y) \mathcal{M}(\boldsymbol{X}^{(y)}, \boldsymbol{\rho}^{(z)}, \boldsymbol{\sigma}^{(z)})$$

$$\text{subject to} \quad \rho_{uv}^{(z)} = \rho_{vu}^{(z)} \quad \forall z, \forall u \neq v, \quad \|\boldsymbol{\rho}^{(1)}\|_1 = C, ..., \|\boldsymbol{\rho}^{(Z)}\|_1 = C \tag{7}$$

We term this approach *ROMGL-Smooth* (an abbreviation for *Kernel-Smoothed ROMGL*).

5. Optimization

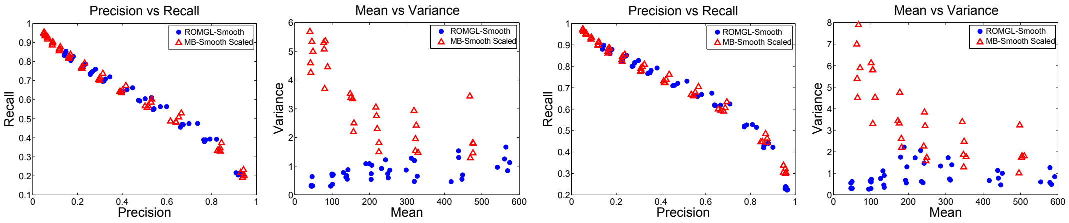
We briefly describe how to optimize Eq. 7. The objective is separable in $z \in \mathcal{Z}$, and thus each $\{\boldsymbol{\rho}^{(z)}, \boldsymbol{\sigma}^{(z)}\}$ pair can be optimized separately from the other $z' \neq z$. However, Eq. 7 is not jointly convex in both $\boldsymbol{\rho}^{(z)}$ and $\boldsymbol{\sigma}^{(z)}$. Fortunately, given a fixed $\boldsymbol{\sigma}^{(z)} = \bar{\boldsymbol{\sigma}}^{(z)}$, the problem is convex in $\boldsymbol{\rho}^{(z)}$. Similarly, given a fixed $\boldsymbol{\rho}^{(z)} = \bar{\boldsymbol{\rho}}^{(z)}$ we can update $\boldsymbol{\sigma}^{(z)}$. Thus, we proceed by alternatively updating $\boldsymbol{\rho}^{(z)}$ and $\boldsymbol{\sigma}^{(z)}$.

To optimize $\boldsymbol{\rho}^{(z)}$ given a fixed $\bar{\boldsymbol{\sigma}}^{(z)}$, we use a projected gradient method, where after updating the current value of $\boldsymbol{\rho}^{(z)}$ in the direction of the gradient, it is projected back onto the constraint set. For our constraint, the projection can be done very efficiently in $O(n \log n)$ time using the method of Duchi et al.[16] Updating $\boldsymbol{\sigma}^{(z)}$ given a fixed $\bar{\boldsymbol{\rho}}^{(z)}$ can be done using a similar update to traditional SPACE: $\frac{1}{\bar{\sigma}_u^{(z)}} \leftarrow \frac{1}{\sum_{y \in \mathcal{Z}} K_h(z,y)} \sum_{y \in \mathcal{Z}} K_h(z, y) \mathcal{M}^u(\boldsymbol{X}^{(y)}, \bar{\boldsymbol{\rho}}^{(z)}, \bar{\boldsymbol{\sigma}}^{(z)})$.

6. Synthetic Evaluation

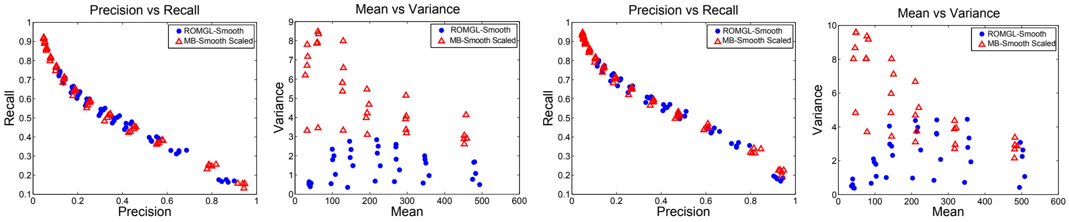
We first focus on synthetic data where the modelling assumptions hold. Our *ROMGL-Smooth* (Eq. 7) can naturally be compared with a Gaussian Graphical Model (GGM) version of KELLER[4] which also uses kernel smoothing. We find that in this case the $\sqrt{S_z}$ scaling (Eq. 2) performs better than the naive approach (Eq. 1), and therefore only compare our approach to GGM KELLER with scaling (which we refer to as *MB-Smooth Scaled*) in this section.

We performed the experiments with two types of graphs: Erdos Renyi random graphs and sparse graphs with hubs. For each type, we generate a sequence of graphs of length 25. Each graph in the sequence has 100 vertices and 200 edges, and is created by randomly deleting and adding 10 edges from the previous graph. The sample size is 30 for the first five graphs, 35 for the next 5, and so on up to 50 for the last 5 graphs. Note that all graphs have the same number of edges (even though they are not identical). We run both methods for $h = \{2, 3\}$, for a variety of regularization parameters, and repeat each experiment for 5 different graph



(a) Prec/Rec, h=2 (b) Mean/Var h=2 (c) Prec/Rec, h=3 (d) Mean/Var, h=3

Fig. 3. Comparison of our robust approach, *ROMGL-Smooth* (blue circles), with an existing non-robust method, *MB-Smooth Scaled* (red triangles), on synthetic Erdos Renyi random graphs. See text for details.



(a) Prec/Rec, h=2 (b) Mean/Var h=2 (c) Prec/Rec, h=3 (d) Mean/Var, h=3

Fig. 4. Comparison of our approach, *ROMGL-Smooth* (blue circles), with an existing non-robust method, *MB-Smooth Scaled* (red triangles), on synthetic sparse graphs with hubs. See text for details.

sequences. The methods are evaluated on two different criteria. To measure accuracy of the approaches in recovering the structures we plot precision/recall curves. The precision is defined as $prec = \frac{1}{Z}\sum_{z\in\mathcal{Z}} \frac{\hat{\mathcal{E}}^{(z)}\cap\mathcal{E}^{(z)}}{|\hat{\mathcal{E}}^{(z)}|}$ and the recall is defined as $rec = \frac{1}{Z}\sum_{z\in\mathcal{Z}} \frac{\hat{\mathcal{E}}^{(z)}\cap\mathcal{E}^{(z)}}{|\mathcal{E}^{(z)}|}$.

We also propose a quantitative measure of robustness. Let $\hat{\boldsymbol{e}} = (|\hat{\mathcal{E}}^{(1)}|, ..., |\hat{\mathcal{E}}^{(\mathcal{Z})}|)$ be the vector of edge counts of the networks recovered by a method. Intuitively, if a method is robust to sample size heterogeneity, the variance of $\hat{\boldsymbol{e}}$ should be small, since all the true graphs have the same number of edges. Thus, we propose the quantity $var(\hat{\boldsymbol{e}})/mean(\hat{\boldsymbol{e}})$ as a measure of robustness (scaling by $mean(\hat{\boldsymbol{e}})$ provides for easier comparison).

The precision/recall curves show that both methods perform comparably according to this metric (Figures 3(a), 3(c), 4(a), and 4(c)), indicating that our new robust approach generates results with comparable accuracy as the scaling method. However, our new approach yields results with considerably lower variance, indicating that it is more robust than the scaling method (Figures 3(b), 3(d), 4(b) and 4(d)). This is especially true when the recovered graphs are sparser, since *MB-Smooth Scaled* has very high variance in this case. This is the most prevalent scenario, since on many real biology datasets, the sample size is small, so we are more likely to select sparse graphs. Furthermore, as we will see, the scaling method performs much worse on real data than synthetic data.

7. Application to the Hematopoietic Stem Cell Dataset

We applied our method to the human hematopoietic stem cell dataset analyzed in Novershtern et al.[17] There are 38 cell states in the tree-shaped multi-lineage stem cell genealogy. We focus on a subset of 732 genes from the entire dataset for the experiments in this section.

First, we quantitatively compare our approach (*ROMGL-Smooth*) to the non-robust approaches: naive (*MB-Smooth Naive*) and scaling (*MB-Smooth Scaled*). The bandwidth for these algorithms was fixed to 5. For a given setting of the regularization parameter (λ or C), we plot the average edge count over all the 38 cell states on the x-axis and the difference between the largest edge count and the smallest edge count on the y-axis. As shown in Figure 5, the non-robust methods produce networks with very different sizes, e.g., some of the networks have less than 100 edges while others have thousands. Our robust approach produces much more calibrated results.

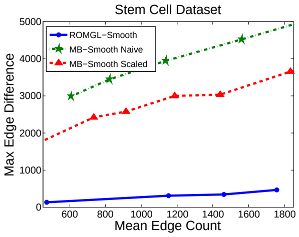
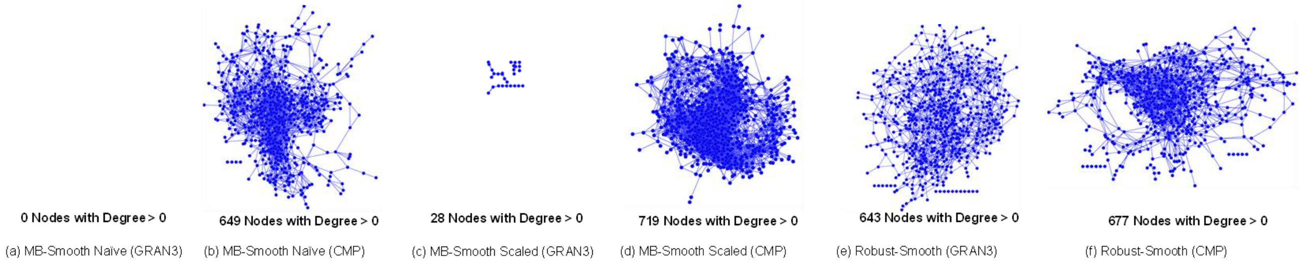


Fig. 5. Our approach, denoted by *ROMGL-Smooth* (blue) compared with *MB-Smooth Scaled* (red) and *MB-Smooth Naive* (green) on the hematopoietic stem cell dataset. Our approach returns networks that are much more calibrated with more similar edge counts.

To examine these differences further, we show cell-specific networks for two cell states, granulocytes (GRAN3) and common myeloid progenitors (CMP), recovered by the three approaches in Figure 6. GRAN3 is a leaf in the cell genealogy; it has few neighbors and the lowest effective sample size (14.92) when the smoothing kernel is applied. In constrast, CMP is an internal node in the genealogy that can differentiate into megakaryocytes, erythrocytes, granulocytes, and monocytes, and thus has many neighbors; it has the highest effective sample size (60.52). As one can see, for the naive approach (Figures 6(a) and 6(b)), sample size heterogeneity is such a problem that the GRAN3 network has zero edges while the CMP network has 4532. Similarly, the scaling approach also performs poorly. The GRAN3 network has only 72 edges (Figure 6(c)) while the CMP network has 2944 edges (Figure 6(d)). Thus, with both of these approaches, it is practically impossible to analyze the GRAN3 network in relation to the other networks. In contrast, our approach gives much more balanced results; the GRAN3 network has 1269 edges (Figure 6(e)) while the CMP network has 1614 edges (Figure 6(f)).

Next, we examined the results generated by our robust approach in more detail. Novershtern et al.[17] discovered various gene modules and their corresponding regulators active in different cell states in the hematopoietic stem cell dataset. It is unknown, however, how genes in these modules interact with one another. We compare and contrast our results to theirs on the two modules 721 and 817 described in Novershtern et al.[17] The former module is induced in granulocytes and monocytes (GRAN/MONO), while the other in B cells, T cells, and granulocytes (BCELL/TCELL/GRAN).

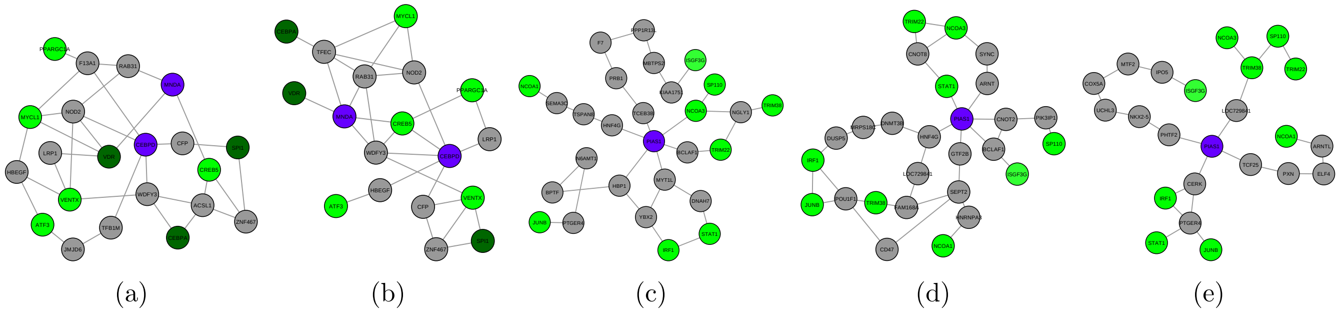
The subnetworks corresponding to the GRAN/MONO 721 module we recovered in the granulocytes and monocytes are shown in Figure 7 (a) and (b). It can be seen that we recovered all the genes in the module for both subnetworks, which include both experimentally verified ones (shown in dark purple and dark green) and unverified ones (light green). Note almost all of the proposed genes in the module are within 2-3 hops from the regulators CEBPD and MNDA in the GRAN3 and MONO2 subnetworks. Moreover, our results reveal interaction patterns of the genes in these subnetworks (only a list of genes in the module was shown in Novershtern et al.[17]). A closer examination of the two subnetworks reveals that they contain


0 Nodes with Degree > 0
649 Nodes with Degree > 0
28 Nodes with Degree > 0
719 Nodes with Degree > 0
643 Nodes with Degree > 0
677 Nodes with Degree > 0

(a) MB-Smooth Naïve (GRAN3) (b) MB-Smooth Naïve (CMP) (c) MB-Smooth Scaled (GRAN3) (d) MB-Smooth Scaled (CMP) (e) Robust-Smooth (GRAN3) (f) Robust-Smooth (CMP)

Fig. 6. The cell-state-specific networks for granulocytes (GRAN3) and common myeloid progenitors (CMP) recovered by the three approaches. The robust approach (*ROMGL-Smooth*), shown in (e) and (f), produces substantially more balanced networks than the other two approaches.

two modules with similar gene interaction patterns, one is a large 10-gene module with MNDA, CREB5, VDR, RAB31, NOD2, CEBPD, CFP, MYCL1, WDFY3, and VENTX, and the other is a small 2-gene module with HBEGF and ATF3. Interestingly, 7 out of these 12 genes were also proposed by Novershtern et al.



(a) (b) (c) (d) (e)

Fig. 7. The *ROMGL-Smooth* reconstructed subnetworks corresponding to (a) module 721 in granulocytes (GRAN3), and (b) module 721 monocytes (MONO2) (c) PIAS1 module in B cells (BCELLa3), (d)PIAS1 module in T cells (TCELL3), and (e) PIAS1 module in granulocytes (GRAN3). Purple represents genes that are regulators of the module and were experimentally validated in Novershtern et al.[17] Dark green represents other genes in the module that were experimentally validated. Light green represents the genes in the module which were not experimentally validated. All the other genes are colored gray.

Finally, we examined the reconstructed subnetworks in B cells (BCELLa3), T cells (TCELL3), and granulocytes (GRAN3) corresponding to the BCELL/TCELL/GRAN 817 module in Novershtern et al.[17] (Figure 7 (c),(d),(e)). In this case, the topologies of the subnetworks are very different. The only gene module shared between the BCELLa3 and TCELL3 subnetworks is HNF4G–PIAS1–BCLAF1. In addition, the topology of the GRAN3 subnetwork corresponding to the BCELL/TCELL/GRAN 817 module is distinctly different from the BCELLa3 and TCELL3 subnetworks. These findings are consistent with the fact that both B cells and T cells are lymphocytes and closer in the genealogy than granulocytes.

8. Discussion

In conclusion, we have identified the problem of sample size heterogeneity in multi-network reconstruction and proposed a principled solution that works well in practice. Our method assumes that all networks have approximately the same number of edges. However, more complex assumptions are possible if we have prior knowledge about the network densities. For example, we can assume cell states in a certain category each have sum of absolute edge weights equal to C_1, while cell states in another category are associated with parameter C_2.

Acknowledgements This research was made possible by Grants NIH 1R01GM093156 and NIH 1R01GM087694, and an NSF Graduate Fellowship (Grant No. 0946825) to APP

References

1. R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* **58**, 267 (1996).
2. N. Meinshausen and P. Bühlmann, High-dimensional graphs and variable selection with the Lasso, *Annals of Statistics* **34**, 1436 (2006).
3. J. Friedman, T. Hastie and R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, *Biostatistics* **9**, 432 (2008).
4. L. Song, M. Kolar and E. Xing, Time-Varying Dynamic Bayesian Networks, *Bioinformatics* **25**, p. i128 (2009).
5. A. Ahmed and E. Xing, Recovering time-varying networks of dependencies in social and biological studies, *Proceedings of the National Academy of Sciences* **106**, p. 11878 (2009).
6. A. Parikh, W. Wu, R. Curtis and E. Xing, TREEGL: reverse engineering tree-evolving gene networks underlying developing biological lineages, *Bioinformatics* **27**, i196 (2011).
7. E. Segal, H. Wang and D. Koller, Discovering molecular pathways from protein interaction and gene expression data, *Bioinformatics-Oxford* **19**, 264 (2003).
8. A. Dobra, C. Hans, B. Jones, J. Nevins, G. Yao and M. West, Sparse graphical models for exploring gene expression data, *Journal of Multivariate Analysis* **90**, 196 (2004).
9. D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques* (MIT press, 2009).
10. S. Lauritzen, *Graphical models* (Oxford University Press, USA, 1996).
11. M. Wainwright, P. Ravikumar and J. Lafferty, High-Dimensional Graphical Model Selection Using l1-Regularized Logistic Regression, *Advances in Neural Information Processing Systems* **19**, p. 1465 (2007).
12. M. Wainwright, Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using Constrained Quadratic Programs, *Information Theory, IEEE Transactions on* **55**, 2183 (2009).
13. R. Irizarry, B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, U. Scherf and T. Speed, Exploration, normalization, and summaries of high density oligonucleotide array probe level data, *Biostatistics* **4**, 249 (2003).
14. B. Bolstad, R. Irizarry, M. Åstrand and T. Speed, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics* **19**, 185 (2003).
15. J. Peng, P. Wang, N. Zhou and J. Zhu, Partial correlation estimation by joint sparse regression models, *Journal of the American Statistical Association* **104**, 735 (2009).
16. J. Duchi, S. Shalev-Shwartz, Y. Singer and T. Chandra, Efficient projections onto the l1-ball for learning in high dimensions, in *Proceedings of the 25th international conference on Machine learning*, 2008.
17. N. Novershtern, A. Subramanian, L. Lawton, R. Mak, W. Haining, M. McConkey, N. Habib, N. Yosef, C. Chang, T. Shay *et al.*, Densely interconnected transcriptional circuits control cell states in human hematopoiesis, *Cell* **144**, 296 (2011).