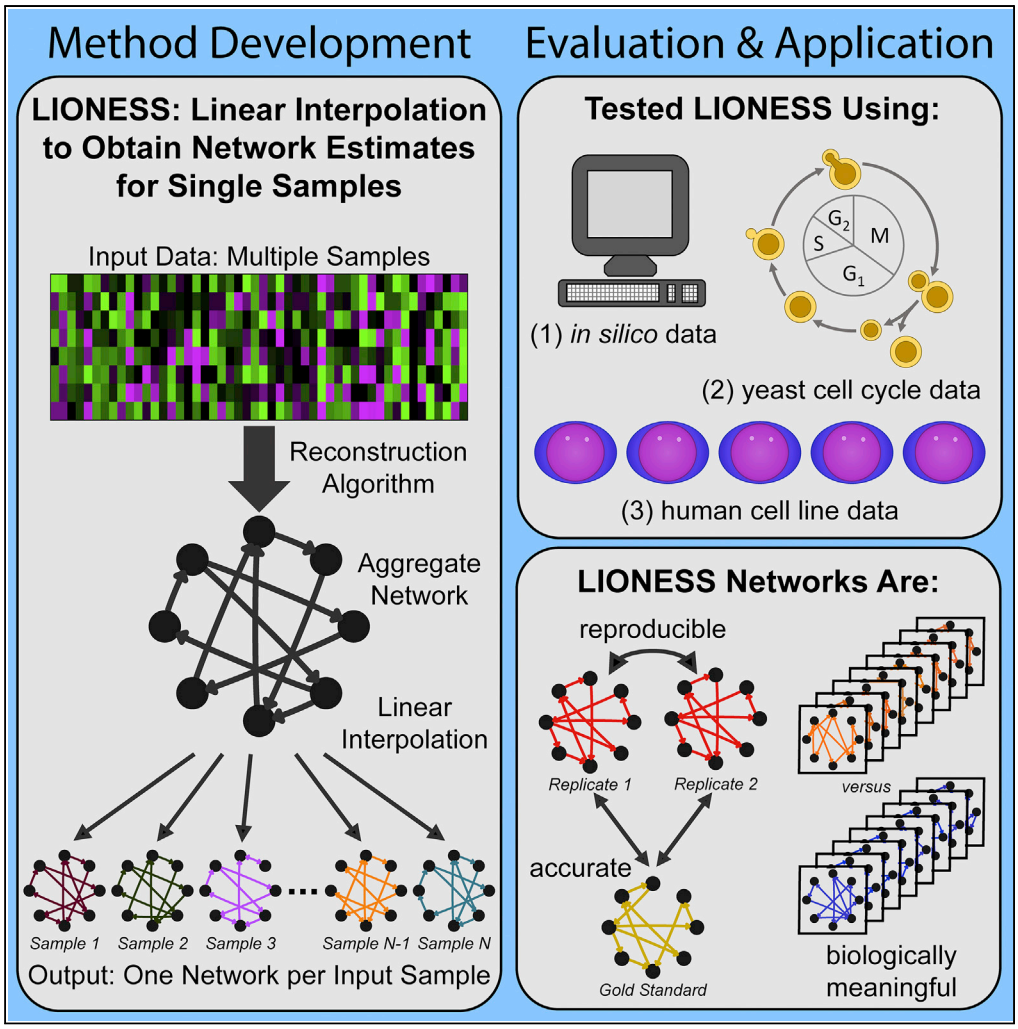
iScience

CellPress

Article

Estimating Sample-Specific Regulatory Networks



Marieke Lydia Kuijjer, Matthew George Tung, GuoCheng Yuan, John Quackenbush, Kimberly Glass

kimberly.glass@channing.harvard.edu

HIGHLIGHTS

We developed LIONESS to extract single-sample networks from aggregate models

We tested LIONESS using *in silico*, yeast, and human expression data

LIONESS-estimated networks are reproducible, accurate, and biologically meaningful

Single-sample network analysis highlights important biological processes

DATA AND SOFTWARE AVAILABILITY

GSE4987
GSE19480
GSE31388

Kuijjer et al., iScience 14, 226–240
April 26, 2019
https://doi.org/10.1016/j.isci.2019.03.021

iScience

CellPress

Article

Estimating Sample-Specific Regulatory Networks

Marieke Lydia Kuijjer,[1,7] Matthew George Tung,[2,7] GuoCheng Yuan,[3,4] John Quackenbush,[3,5,6] and Kimberly Glass[5,6,8,*]

SUMMARY

Biological systems are driven by intricate interactions among molecules. Many methods have been developed that draw on large numbers of expression samples to infer connections between genes (or their products). The result is an aggregate network representing a single estimate for the likelihood of each interaction, or "edge," in the network. Although informative, aggregate models fail to capture population heterogeneity. Here we propose a method to reverse engineer *sample-specific* networks from aggregate networks. We demonstrate our approach in several contexts, including simulated, yeast microarray, and human lymphoblastoid cell line RNA sequencing data. We use these sample-specific networks to study changes in network topology across time and to characterize shifts in gene regulation that were not apparent in the expression data. We believe that generating sample-specific networks will greatly facilitate the application of network methods to large, complex, and heterogeneous multi-omic datasets, supporting the emerging field of precision network medicine.

INTRODUCTION

In many instances, especially when analyzing complex traits and diseases, a single gene or pathway cannot fully explain a particular phenotype. In these cases, biological processes are often characterized as complex networks whose structures are altered as the phenotype changes. Studying the pattern of connections between biological components, and how these structures change between cell states, can yield new insights into the mechanisms driving disease. However, accurately reconstructing these networks in a way that captures both the properties and complexities of each phenotype remains a significant challenge.

Biological and phenotypic variability is a prominent feature in many complex traits and diseases. The generation of large multi-omic resources, including The Cancer Genome Atlas, the Encyclopedia of DNA Elements (ENCODE Project Consortium, 2012), and the Genotype-Tissue Expression (GTEx Consortium, 2015; GTEx Consortium et al., 2017) project, as well as the recent rise of single-cell genomic technologies and the cataloging of individual cell types in the Human Cell Atlas (Rozenblatt-Rosen et al., 2017), have brought this issue to the forefront. We now recognize that diversity in the regulatory processes active in different cells, across multiple tissues, between various phenotypes, and even in response to environmental exposures, all contribute to the complexity of observed disease manifestations. It is also increasingly clear that the cumulative effect of multiple individual-specific variations, each with a relatively small effect size, likely play an important role in the manifestation of many different diseases, including rare disease subtypes (McClellan and King, 2010). These observations speak to a multifactorial process. In other words, rather than individual molecules, it is alterations in biological processes, characterized as complex networks, that play a critical role in mediating the observed diversity (Loscalzo et al., 2007). Effectively capturing this network-level heterogeneity is critical as we seek to understand how gene expression and regulatory processes manifest at an increasingly individualized level.

Existing methods for estimating biological networks often rely upon combining information from large quantities of data (most commonly gene expression data). This means that even when the data represent a spectrum of phenotypes, these approaches, by default, estimate only a single "aggregate" network (De Smet and Marchal, 2010; Marbach et al., 2012). Although these types of aggregate networks have allowed us to gain important insights across a wide range of biological systems and diseases, they only capture the regulatory processes shared across a population of samples. More recently, several approaches have been suggested for exploring sample-level network information (Alvarez et al., 2016; Liu et al., 2015, 2016). However, these methods are severely limited. In particular, current single-sample methods rely upon differential analysis of the underlying expression data, thereby masking any information shared across the

[1]Centre for Molecular Medicine Norway (NCMM), Nordic EMBL Partnership, University of Oslo, 0318 Oslo, Norway

[2]Department of Anesthesiology, Critical Care, and Pain Medicine, Massachusetts General Hospital, Boston, MA 02114, USA

[3]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA 02115, USA

[4]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA 02215, USA

[5]Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA 02115, USA

[6]Department of Medicine, Harvard Medical School, Boston, MA 02115, USA

[7]These authors contributed equally

[8]Lead Contact

*Correspondence: kimberly.glass@channing.harvard.edu

https://doi.org/10.1016/j.isci.2019.03.021

population (see "Transparent Methods" and Table S1). Regulatory processes act on a network that contains both common and context-specific interactions (Sonawane et al., 2017). However, there are currently no existing approaches designed to reconstruct the complete network for each sample in a population.

To fill this gap and effectively model the regulatory processes active in each sample in a population, we have developed a method to reverse engineer sample-specific networks. We call this approach LIONESS (Linear Interpolation to Obtain Network Estimates for Single Samples). LIONESS estimates individual sample networks by applying linear interpolation to the predictions made by existing aggregate network inference approaches. In this article, we demonstrate the accuracy, robustness, and applicability of LIONESS in the context of multiple aggregate network reconstruction approaches and in several datasets, including simulated data, microarray expression data from synchronized yeast cells, and RNA sequencing (RNA-seq) data collected from human lymphoblastoid cell lines (Figure 1A; Table S2). We also show how the predictions from LIONESS can be used to model regulatory network changes over time and to characterize the regulatory processes active in individual samples. Ultimately, we find that analyzing single-sample regulatory networks provides a view of biological systems that is distinct from, but complementary to, other sources of multi-omic data.

RESULTS

Complex Relationships in Biological Networks

Many widely used network inference methods start by calculating a score or statistic for each gene pair based on shared information across a set of input gene expression samples (De Smet and Marchal, 2010; Marbach et al., 2012). These scores are sometimes augmented to better account for regulatory complexity (Faith et al., 2007; Margolin et al., 2006; Langfelder and Horvath, 2008) but are ultimately used to infer the presence or absence of "interactions" between genes. This collection of genes and their corresponding complex set of inferred interactions are conceptualized as a network in which "nodes" represent genes and "edges" represent the interactions between those genes. In this context, heterogeneity in the underlying input samples is often essential for correctly estimating a network model, as variance in the data can amplify gene co-variation patterns, leading to more robust network predictions. However, at the same time, building this type of consensus, or "aggregate," network model largely ignores the fact that there may be multiple different underlying regulatory networks represented across the individual input samples.

Consider the collection of cells within a tissue. We now recognize that within this system, each cell may have its own unique gene expression profile and corresponding unique active gene regulatory network. In the same way, each individual person in a group manifests a phenotype in a slightly different fashion, meaning that his or her gene expression profile and the gene regulatory network driving it should be subtly different. We have started to embrace this complexity in analyzing gene expression, whereas it has been largely ignored in the analysis of gene regulatory networks.

To better model network-level diversity across a population, we sought to develop a method that could model sample-specific networks. In developing our approach, we recognized that there are two types of relationships that needed to be considered: (1) intra-network relationships, or the connections among the nodes (genes) *within* a biological network, and (2) inter-network relationships, or the relationships *between* multiple different biological networks. The first of these (intra-network relationships) is an area that has been highly studied. It is now widely recognized that relationships among nodes within a biological network are very complex and that these networks are often characterized by nonlinear regulatory dynamics and synergistic effects. Fortunately, there are many approaches that have already been developed to model these complex interactions (Wang and Huang, 2014; Marbach et al., 2012), as outlined above. In contrast, the comparative study of networks (inter-network relationships) is still a relatively young field. However, a number of recent studies have used linear approaches to analyze and cluster sets of networks (Marbach et al., 2012; Schlauch et al., 2017; Mucha et al., 2010; Onnela et al., 2012).

LIONESS: Linear Interpolation to Obtain Network Estimates for Single Samples

With the above in mind, we developed our approach by using a linear framework to relate a set of networks, each representing a different biological sample. In other words, we suggest that an "aggregate" network predicted from a set of N samples can be thought of as the average of individual component networks reflecting the contributions from each member in the input sample set. Mathematically, this means that the

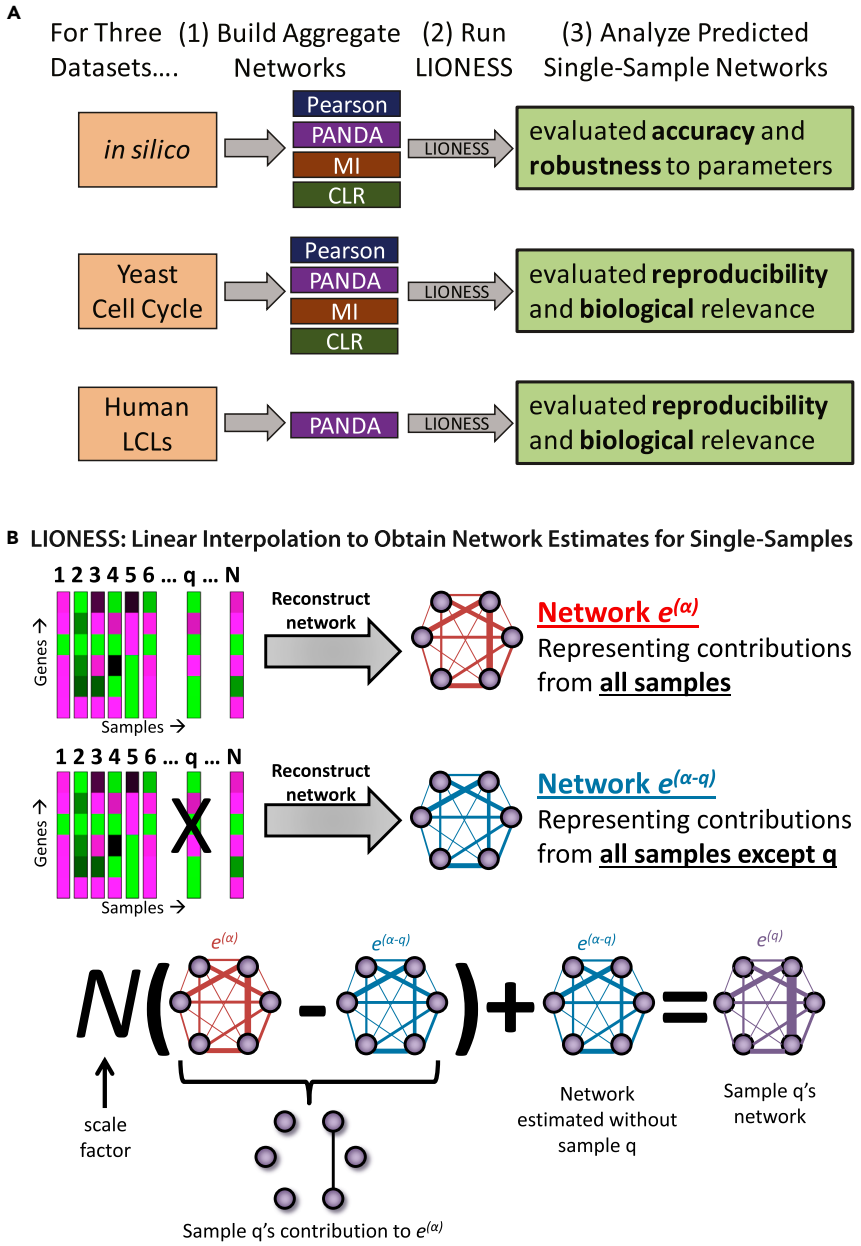


Figure 1. Overview of LIONESS Approach and Evaluation

(A) Flow diagram summarizing the analyses performed in this article to evaluate the LIONESS approach. LIONESS was applied to multiple aggregate network reconstruction approaches including Pearson correlation coefficient, PANDA (Passing Attributes between Networks for Data Assimilation), MI (mutual information), and CLR (Context Likelihood of Relatedness).

(B) Visual illustration of how LIONESS estimates the network for a single sample based on two aggregate network models, one reconstructed using all biological samples in a given dataset and the other using all except the sample of interest (q). See also Table S2 and Figure S1.

weight of an edge, $e_{ij}^{(\alpha)}$ between two nodes (i and j) in an aggregate network derived using all samples (α) can be modeled as the linear combination of the weight of that edge across a set of networks:

$$e_{ij}^{(\alpha)} = \sum_{s=1}^{N} w_s^{(\alpha)} e_{ij}^{(s)}, \qquad \text{(Equation 1)}$$

where $\sum_{s=1}^{N} w_s^{(\alpha)} = 1$.

In this equation, each network ($e_{ij}^{(s)}$) in the set directly corresponds to one of the samples (s) used to reconstruct the aggregate network ($e_{ij}^{(\alpha)}$), and $w_s^{(\alpha)}$ represents the relative contribution of that sample to the aggregate model; we note that the complex relationships between the nodes in the aggregate network ($e_{ij}^{(\alpha)}$) can be calculated using any aggregate network reconstruction approach. This allows us to ensure that higher-order, nonlinear relationships, such as those commonly found in complex biological networks, can be included in the network models.

Next, we also suggest that, as in Equation 1, the weight of an edge in a network reconstructed using all but one of the samples ($\alpha - q$) can be written as:

$$e_{ij}^{(\alpha-q)} = \sum_{s \neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)}, \qquad \text{(Equation 2)}$$

where $\sum_{s \neq q}^{N} w_s^{(\alpha-q)} = 1$.

Comparing Equations 1 and 2, we find that $w_q^{(\alpha)} = 1 - w_s^{(\alpha)} / w_s^{(\alpha-q)}$ so long as the relative contribution of each of the samples (s) has the same proportional contribution to the network reconstructed using all samples ($w_s^{(\alpha)}$) when compared with the network reconstructed using all samples except one ($w_s^{(\alpha-q)}$) (implying that $w_s^{(\alpha)} / w_s^{(\alpha-q)}$ is a constant; for more information see Equation E8 in Supplemental Information). This comparison also allows us to solve exactly for the network for an individual sample q. In particular, by subtracting the above equations we find:

$$\begin{aligned} e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)} &= w_q^{(\alpha)} e_{ij}^{(q)} + \sum_{s \neq q}^{N} \left(w_s^{(\alpha)} - w_s^{(\alpha-q)} \right) e_{ij}^{(s)} \\ &= w_q^{(\alpha)} e_{ij}^{(q)} - w_q^{(\alpha)} \sum_{s \neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)} \end{aligned} \qquad \text{(Equation 3)}$$

The network specific to sample q in terms of the aggregate networks is then:

$$e_{ij}^{(q)} = \frac{1}{w_q^{(\alpha)}} \left(e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)} \right) + e_{ij}^{(\alpha-q)}. \qquad \text{(Equation 4)}$$

In summary, the edge scores for a given individual network are equal to the difference in edge scores for an aggregate network constructed using all the samples and an aggregate network reconstructed using all but the sample of interest, multiplied by a scaling factor, and added to the edge scores of the network reconstructed using all but the sample of interest (Figure 1B). What this means is that we can use pairs of aggregate network models to "extract" networks for each of the individual input samples. In the following analysis we give samples equal weight ($w_q^{(\alpha)} = 1/N$), although one could, in principle, weight samples differently based on the quality of the data for individual samples or some other measure. A more detailed version of the LIONESS derivation is provided in the Supplemental Information.

We note that the mathematical framework presented in Equation 4 is independent of the inference method used to estimate the aggregate network edge weights. In other words, LIONESS can be thought of as a mathematical "wrapper" that can be applied to estimate networks based on any aggregation model. With this in mind, we have performed a detailed exploration of the behavior of Equation 4 when the aggregate network model is calculated using Pearson correlation or mutual information (MI), two measures commonly applied to quantify the level of a linear or nonlinear association between variables, respectively. For both measures, we are able to show that the inter-network linearity assumption of LIONESS (Equation 1) holds in the context of large sample size (see Supplemental Information). Simulation analysis also illustrates how LIONESS consistently assigns similar edge weights to the samples that most contribute to an expected relationship and correctly identifies and re-weights edges for the samples that are most inconsistent with an expected relationship (Figure S1).

LIONESS Accurately and Robustly Predicts Networks Using *In Silico* Data

To systematically evaluate LIONESS, we created a series of datasets where the underlying networks corresponding to each input expression sample are known. We used these data to (1) evaluate whether LIONESS

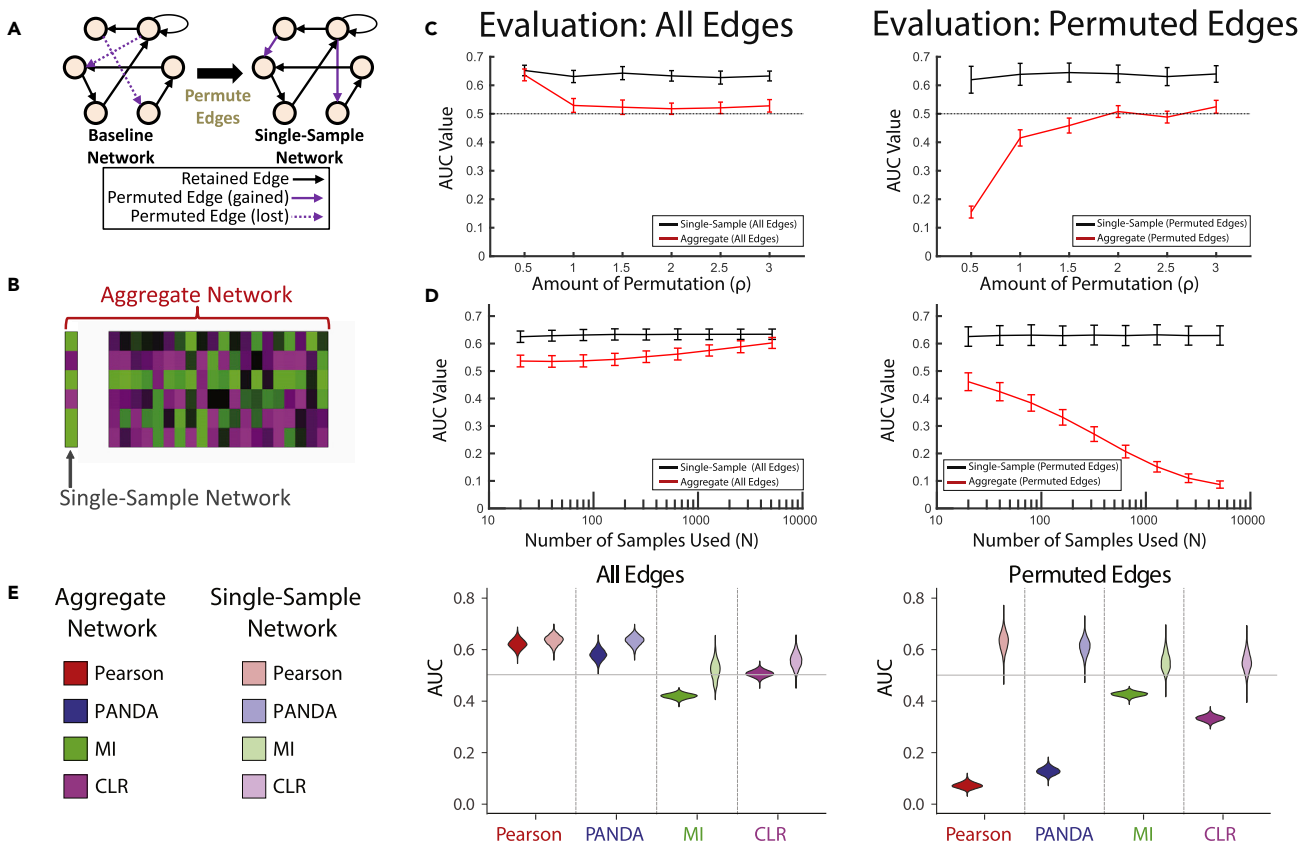


Figure 2. Evaluation of LIONESS' Ability to Recover Known Single-Sample Networks in *In Silico* Data

(A) Toy example of how we create a single-sample network from an underlying baseline network.

(B) Illustration of the gene expression samples used to build a single-sample network. We evaluated the accuracy of both the aggregate network derived using all samples (red) and the LIONESS-estimated single-sample network (black) by benchmarking against the corresponding "gold-standard" single-sample network.

(C) The mean and standard deviation of the AUC values of the aggregate (red) and LIONESS-predicted single-sample networks (black) estimated from *in silico* datasets representing varying levels of heterogeneity.

(D) The mean and standard deviation of the AUC values of the aggregate (red) and LIONESS-predicted single-sample networks (black) estimated using increasing numbers of input expression samples. For each sample size, 10,000 random subsets of samples were used.

(E) Violin plots showing the distribution of AUC values for aggregate and LIONESS-predicted single-sample networks estimated using four different aggregate network reconstruction approaches. For (C–E) AUCs were calculated using all possible edges, and for edges that differ from the baseline model (permuted edges), see (A).

See also Figures S2–S4.

accurately predicts individual sample networks, (2) to explore how sensitive these predictions are to the properties of the underlying data, and (3) to assess whether LIONESS is able to recover sample-specific network relationships (i.e., edges specific to a given sample's network).

Briefly, to create a benchmark *in silico* dataset, we started with a baseline network containing *M* nodes and random edges. We then permuted the edges within this baseline network, creating a single-sample network with the same degree distribution (Figure 2A). We repeated this *N* times, creating *N* "gold-standard" single-sample networks. To derive corresponding expression profiles for each of these networks, we generated 1,000 random initial expression states (0 or 1 corresponding to whether the gene is "on" or "off," respectively) and applied a Boolean model (see "Transparent Methods") to determine the corresponding network attractors (Wuensche, 1998). We averaged over all states defined within these 1,000 attractors to generate "expression" values for the *M* nodes (which represent genes) in each network. This gave us an *M*-by-*N* matrix of expression values, one for each of the nodes (genes) in each network. An overview of our approach is shown in Figure S2. The generated *in silico* datasets are included in Data S1.

We first evaluated LIONESS' predictions in the context of varying heterogeneity. To do this, we generated six different *in silico* datasets using the same baseline network but varying the amount of permutation used to obtain the single-sample network models. For this analysis we chose a network size of $M=100$ nodes and $N=100$ samples and used Pearson correlation to calculate an aggregate network before applying Equation 4 to reconstruct each of the individual sample networks. We evaluated the accuracy of the Pearson correlation aggregate network and each of the LIONESS-estimated single-sample networks (Figure 2B) by comparing with the original "gold-standard" networks and calculating the area under the receiver operator characteristic curve (AUCROC, or more simply AUC).

We observe that in the context of greater heterogeneity among the single-sample networks (increased permutation) the LIONESS-predicted networks are much more accurate than the aggregate network (Figure 2C). On the other hand, in the context of low heterogeneity, the accuracy of the LIONESS-predicted networks is similar to that of the aggregate network; this is to be expected because the aggregate network should not be significantly different from the single-sample networks in this context. Most interesting, however, is the fact that the accuracy of the permuted edges (those that appear in the single-sample network but not the baseline network, see Figure 2A) is *independent* of sample heterogeneity. These edges are not accurately captured in the aggregate network model, especially in the case of low heterogeneity.

We have repeated this analysis on *in silico* data for networks (1) of various sizes (contain more nodes) and (2) with varying levels of noise added to their associated expression data. We find that LIONESS' performance is independent of the size of the network models (Figures S3A and S3B) and retains its ability to predict networks even in the presence of expression data noise (Figure S3C).

Next, we evaluated LIONESS' predictions in the context of varying sample size. To do this, we generated an additional *in silico* dataset based on the same 100-node baseline network as the previous analysis. We used a moderate level of permutation ($\rho=1$) to generate a dataset with 10,000 paired network and expression samples. We selected subsets of this dataset containing $N+1$ samples, where N varied from 20 to 5,000; applied LIONESS to estimate the $(N+1)^{th}$ sample's network; and evaluated the accuracy of that network as well as the corresponding aggregate network from which it was derived (Figure 2D). We observe that as we increase the number of samples (N), the accuracy of LIONESS single-sample networks remains constant, both overall and for the sample-specific permuted edges. However, although including more samples improves the accuracy of the aggregate network model, the sample-specific permuted edges within the aggregate model are very poorly estimated with increasing sample size. This behavior is expected; including more samples provides increasing information that can help accurately estimate edges that are in the baseline network (those that are most likely to be common across all the single-sample networks). These edges are, by definition, the opposite of the sample-specific permuted edges.

We next assessed how sensitive LIONESS networks are to the chosen set of "background" samples. Using the same *in silico* dataset described above, we evaluated the similarity between pairs of single-sample networks that represent the same expression sample (q), but which were constructed using independent sets of background samples. We found high reproducibility, in particular as we increased the number of background samples (Figure S4A). We also tested how robust LIONESS predictions are when there are distinct subtypes represented in the background samples. To do this, we generated a separate *in silico* dataset that contained seven subtypes of different sizes (for more information see "Transparent Methods"). We found that LIONESS' performance was similar when using a background consisting of all samples, or a background consisting of only subtype-specific samples (Figure S4B, p value = 0.639 for the overall analysis). Simulation analysis also illustrates how, in the case of multiple subtypes in the underlying expression data, using all samples allows for a robust estimation of single-sample edge weights (Figure S4C).

Finally, we tested the generalizability of LIONESS by estimating single-sample networks from aggregate models derived using several common network reconstruction approaches, including Pearson correlation, Passing Attributes between Networks for Data Assimilation (PANDA) (Glass et al., 2013), MI, and Context Likelihood of Relatedness (CLR) (Faith et al., 2007). These methods represent several commonly used reconstruction approaches, including both linear (Pearson) and nonlinear (MI) models, that infer either directed (PANDA) or undirected (Pearson, MI, CLR) networks (for more information, see "Transparent Methods"). Figure 2E shows the distribution in AUC values for the aggregate and LIONESS single-sample network predictions for each of these approaches. We find that LIONESS consistently and accurately

predicts single-sample networks for all four network inference methods. Interestingly, although the difference in AUC between the overall aggregate and single-sample models is fairly similar for all four approaches, the AUC values are lowest for networks estimated using MI, a nonlinear approach for assessing correlation. This may reflect that our *in silico* data do not fully represent the complexity found in biological systems or that MI is not the optimal measure to use when estimating a regulatory network from expression data.

Estimating Single-Sample Networks Using Experimental Data from Yeast

We next tested LIONESS using experimental data from cell-cycle-synchronized yeast cells. We downloaded gene expression data (Gene Expression Omnibus: GSE4987) (Pramila et al., 2006) consisting of dye-swap technical replicates measured every 5 min for 120 min. We ma-normalized (Yang et al., 2009) these data, removed probe sets with missing information, batch-corrected using ComBat (Johnson et al., 2007), averaged probe sets mapping to the same open reading frame annotation, and quantile-normalized the resulting gene-by-sample matrix of expression values. We note that the 105-min time point was excluded in both replicates due to poor hybridization performance (Pramila et al., 2006).

We used four different network inference methods (Pearson Correlation, PANDA (Glass et al., 2013), MI, and CLR (Faith et al., 2007)) to reconstruct aggregate networks for this dataset and applied LIONESS to estimate the networks for each of the individual samples. The correlation between edge weights in each pair of the estimated sample-specific networks is shown in the first column of Figure 3 (R1&R2-from-R1&R2). We see that network estimates for the same technical replicate are highly similar, as evidenced by the strong diagonal in the upper-right and lower-left squares of each comparison; additional structure is also evident in off-diagonal similarities that reflect the fact that the time course data include more than one cell cycle.

To test if strong reproducibility was because of inclusion of replicates in the expression data, we also ran LIONESS separately on each individual replicate. This analysis produced 24 single-sample networks estimated using only the data in replicate one and 24 single-sample networks estimated using only the data in replicate two (R1-from-R1 & R2-from-R2). The correlation between these networks is shown in the second column of Figure 3. As before, we observe strong reproducibility in estimated edge weights between technical replicates. However, it is worth noting that even though we have corrected for batch effects in the expression data, several of the methods, especially CLR, appear to be sensitive to the "background" data used.

We note that this level of reproducibility is similar to that observed in the underlying expression data, demonstrating that we did not lose replicate information by applying LIONESS separately to the two sets of expression samples (Figure S5A). Interestingly, replicate PANDA networks had higher levels of similarity when compared with the other three reconstruction approaches. Based on these results, in the following analysis we focus on the single-sample networks derived using PANDA as the aggregate network inference method. Results for the other reconstruction approaches are presented in Figure S5B.

Single-Sample Networks Show Periodic Structure across the Cell Cycle

We next tested whether these single-sample networks could provide insight into gene regulation and dynamic cellular network processes. We averaged the aggregate networks and single-sample networks representing the same time point in each of the two replicates, identified the 1,000 edges with the highest variability across the individual networks, and visualized those edges as a heatmap in Figure 4A. The highly variable edges have a range of predicted weights in the aggregate network (left panel); however, we observe strong oscillatory patterns in edge weights (right panel), apparently reflecting changes in gene regulation across the cell cycle. Further investigation indicates that all these highly variable edges originate from one of four transcription factors (MBP1, SWI4, SWI6, and STB1), each of which is known to play a key role in regulating the yeast cell cycle (Ho et al., 1999).

We examined the genes for which there is strong evidence of targeting by these transcription factors (average edge weight across all LIONESS networks greater than zero). In Figure 4B we plot the average weight of these high-evidence interactions for each regulating transcription factor and the average expression of their target genes. It is immediately apparent that oscillation in edge weights occurs at exactly twice

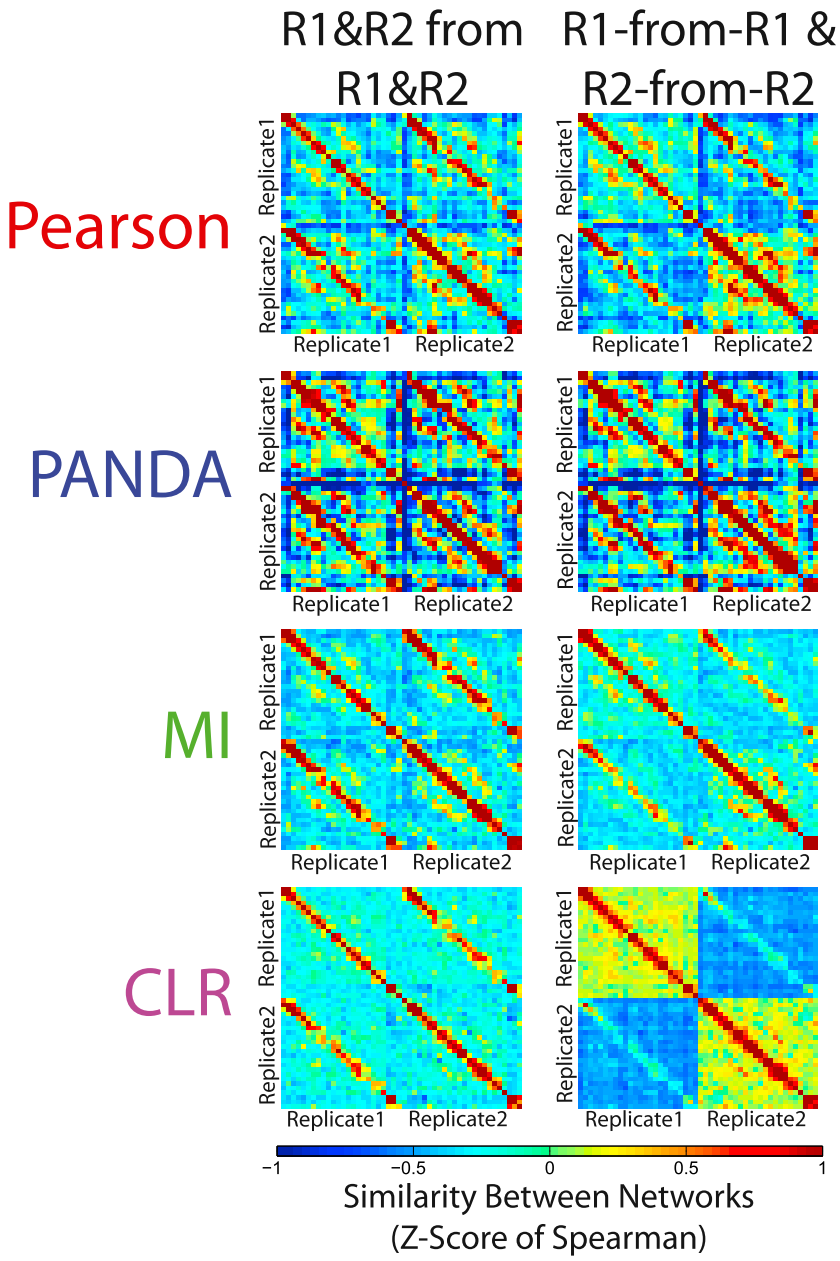


Figure 3. Analysis of LIONESS Networks Predicted for 48 Expression Samples Collected across a Yeast Cell Cycle Time Course Experiment

LIONESS was used to predict networks for each sample in the expression data by applying four different aggregate network reconstruction approaches. For each approach we built the aggregate models either using all samples (R1&R2 from R1&R2), or only the samples from the same technical replicate (R1-from-R1 & R2-from-R2). The Spearman correlation was used to evaluate how similar these networks are to each other. See also Figure S5.

the frequency of the oscillation in gene expression, and that the gene expression oscillates with a period approximately equal to that of the yeast cell cycle.

To understand this result we have to recognize that PANDA interprets correlation in target gene expression as an indication of co-regulation by an upstream transcription factor. Consequently, PANDA assigns greater edge weights when a transcription factor's targets are all coordinately increasing (activated) or decreasing (de-activated or repressed) their expression levels. High edge weights should be interpreted

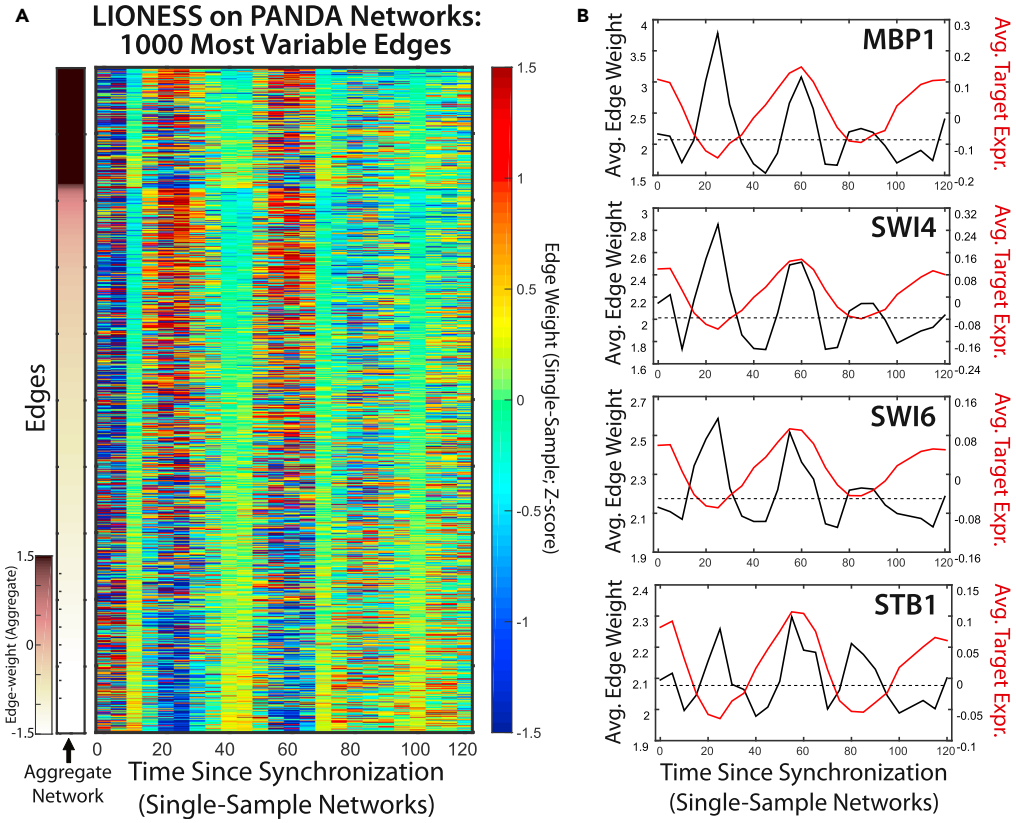


Figure 4. Characterizing Networks across the Yeast Cell Cycle

(A) A heatmap of the edge weights for the 1,000 most variable edges across the sample-specific network models. The left panel shows the weights of these edges in the aggregate network, and the right panel shows the edge weights across the single-sample networks. For the right panel rows are *Z* score normalized for visualization purposes.

(B) The average expression of genes targeted by the four transcription factors that were identified as regulatory nodes of the 1,000 topmost variable edges as well as the average weight of high-confidence edges that extend between those transcription factors and their target genes. The average weight of these edges in the aggregate network is shown as a dashed line.

See also Figure S5.

as evidence for information flow from a transcription factor (TF) to its targets, which could be due to a physically present TF actively regulating its downstream targets, and could also reflect a strong *lack* of regulation by that TF. In this light, the "turn on/turn off" behavior is exactly what one would predict given how PANDA estimates network relationships and is further evidence that LIONESS is extracting meaningful single-sample networks.

Reconstructing Single-Sample Networks for Human Lymphoblastoid Cell Lines

Lastly, we applied LIONESS to infer individual-specific human gene regulatory networks. We used a set of 155 RNA-seq samples from immortalized lymphoblastoid cell lines representing 65 different individuals (Pickrell et al., 2010). We downloaded raw fastq files from the Pritchard lab website (http://eqtl.uchicago.edu/) and aligned samples to hg19 using Bowtie (Langmead et al., 2009); subsequent quality control analysis using RNA-SeQC (DeLuca et al., 2012) excluded two samples due to low expression profile efficiency scores. This left us with a final set of 153 RNA-seq experiments that includes replicates and represents 65 distinct individuals. We normalized this dataset using DEseq2 (Love et al., 2014). For additional data processing and normalization information, see Transparent Methods.

Based on our results when applying LIONESS to network models in the simulated and yeast cell cycle data, we chose PANDA as our aggregate network reconstruction method for the human data. We used PANDA to estimate aggregate gene regulatory network models for the collection of 153 RNA-seq samples. We

then applied LIONESS to these aggregate models, resulting in 153 single-sample networks, one for each of the RNA-seq expression samples. A hierarchical clustering (complete linkage, Spearman correlation) of the network edge weights demonstrates that networks for the same individual nearly always cluster more strongly with each other than with networks representing different individuals (Figure S6). This analysis demonstrates that even when constructing networks using biological data from higher-order organisms such as human, the sample-specific networks predicted by LIONESS are reproducible.

Complex Relationships between Network Targeting and Gene Expression

As with yeast, we investigated the relationship between gene targeting and expression in human networks. First, we averaged single-sample networks that represent the same individual, resulting in 65 "person-specific" regulatory networks. We then selected high-evidence regulatory interactions for each transcription factor (average edge weight across all single-sample networks greater than zero), and directly compared the mean edge weight for these interactions in each of the single-sample networks to the average expression of the targeted genes in the original expression samples.

We found nonlinear relationships between targeting and expression, with the highest average edge weights occurring when target genes have either high or low expression levels (Figure 5A); this is consistent with what we observed in our yeast analysis (Figure 4B). Coloring by the transcription factor expression level in each sample reveals additional patterns with some transcription factors primarily acting as activators (increased target gene expression upon increased TF expression and targeting) and others generally acting as repressors (decreased target gene expression upon increased TF expression and targeting). However, the relationship between a transcription factor and its target genes is not always simple, indicating that other regulatory mechanisms, such as co-activators, post-translational modifiers, or epigenetic mechanisms, are likely playing an important role in mediating these regulatory events.

Increased Network Targeting Corresponds to Open Chromatin

DNase hypersensitivity profiling data are also available for these 65 lymphoblastoid cell lines (Degner et al., 2012), and we used it to investigate how network structures reflect epigenetic state. We downloaded the data from the Pritchard lab website (http://eqtl.uchicago.edu/) and called DNase "peaks" for each sample using Model-based Analysis for ChIP-seq (MACS) (Zhang et al., 2008). When a peak fell within the promoter region of a gene, we assigned that gene a sample-specific score reflecting the significance level of the associated peak call. We found 12,424 genes with a DNase promoter peak in at least one sample and 3,488 genes with a promoter peak in all samples. For details on the DNase data processing, see "Transparent Methods".

A DNase hypersensitivity peak represents a region of open chromatin that is often presumed to be occupied by one or more regulatory proteins, including transcription factors. We wanted to determine if differences in chromatin state between the 65 individuals is reflected in alterations in transcription factor targeting within our single-sample networks. We assigned each edge in each sample a score by combining (1) the weight of that interaction in our single-sample network models (because this value indicates whether information is flowing between that transcription factor and target gene in the PANDA model) and (2) the expression level of the transcription factor itself (because this value indicates whether the TF is physically present in the cell (Figure 5B)). This resulted in a set of expression-modified edge weights for each sample. For more information on how we calculated these edge weights, see "Transparent Methods".

We next used the sum of the edge weights associated with each gene to estimate the number of transcription factors regulating that gene in each of the 65 person-specific networks ($k^{(L+e)}$). For comparison, we calculated gene targeting in two other ways: (1) using LIONESS edge weight estimates in the absence of gene expression information ($k^{(L)}$) and (2) using gene expression information in the absence of LIONESS-predicted edge weights ($k^{(m+e)}$); for the second measure we combined transcription factor expression in each sample with the motif information used for PANDA's prior (see "Transparent Methods"). We note that this last approach is conceptually similar to current methods for approximating sample-specific network information (see Introduction).

To evaluate the association of network targeting with chromatin state, for each gene we calculated the Spearman correlation between gene targeting across the networks and the significance scores of that gene's promoter

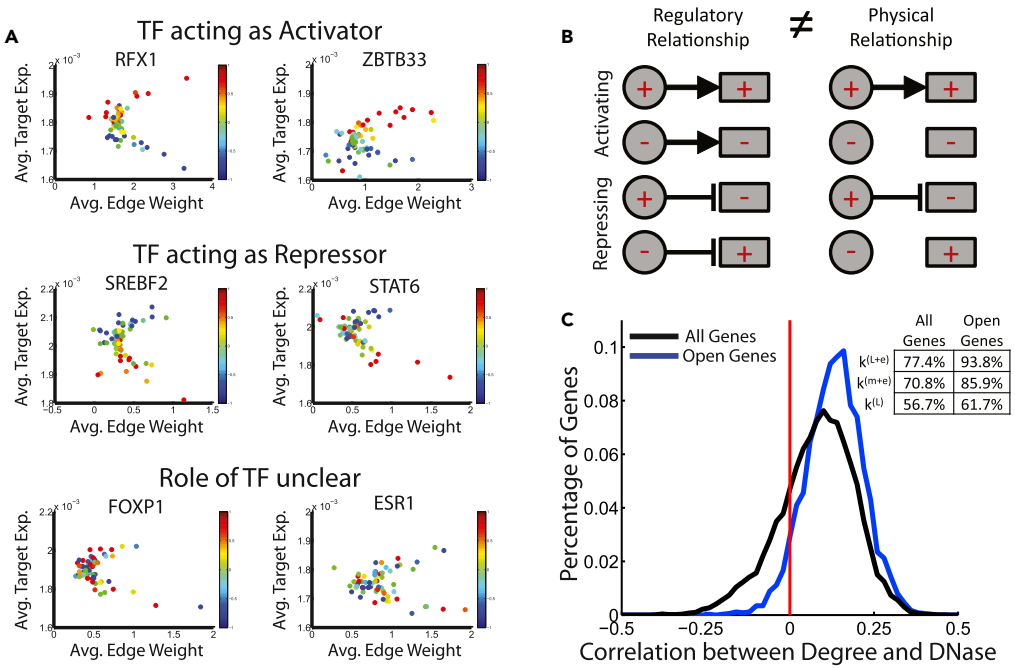


Figure 5. Comparison of Gene Regulation, Gene Expression, and DNase Hypersensitivity Data

(A) For six representative transcription factors, the mean expression of target genes and the mean weight of the edges targeting those genes across the 65 samples are plotted. For each sample, the expression of the TF is shown as a color, scaled to the normal distribution for visualization purposes.

(B) A cartoon illustrating how high edge weights and thus regulatory activity is not necessarily equivalent to the presence of a physical interaction.

(C) The distribution of the Spearman correlation values when comparing gene targeting (calculated by combining LIONESS predictions with TF expression; $k^{(L+e)}$) and the significance level of DNase hypersensitivity in a gene's promoter across all the samples. We also show the percentage of genes whose targeting positively correlates with DNase hypersensitivity when targeting is calculated using only the LIONESS-predicted edge weight ($k^{(L)}$: no expression considered) or a combination of expression and motif information ($k^{(m+e)}$). We performed these analyses either using all 12,424 genes included in our network model or for the set of 3,488 genes with a DNase peak called in all 65 samples (open genes).

See also Figure S6.

DNase across the corresponding cell lines. We find that gene targeting in the expression-modified LIONESS model ($k^{(L+e)}$) is very strongly correlated with promoter-DNase events, especially when only considering genes with measured chromatin information across all the cell lines (Figure 5C). This association is greater than when using only expression and motif information ($k^{(m+e)}$), demonstrating that the LIONESS approach provides additional information on chromatin state not apparent in the data used to seed the algorithm.

Differential Targeting of Genes Highlights Important Biological Processes

Finally, we wanted to determine if there are common structures across these single-sample regulatory networks that might be reflective of important biological processes. We performed a hierarchical clustering (complete linkage, Spearman correlation) on the edge weights in the 65 single-sample lymphoblastoid networks and identified two distinct groups of samples defined by sample-specific edge weights (Figure 6A). In parallel, we performed a hierarchical clustering using gene expression values (Figure 6B) and found two groups of samples that are distinct from the groups defined by the edge weight clustering.

We then used Gene Set Enrichment Analysis (GSEA) (Subramanian et al., 2005) to compare groups of samples defined in the expression-based and network-based clustering. First, we compared the expression levels of genes between groups of individuals defined in the expression-based clustering. Although there were many differentially expressed genes between the expression-based groups of samples (2,620 with false discovery rate [FDR] < 0.01), GSEA found no enrichment for known biological functions (Figure 6C). Next, we defined the targeting level of a gene in a sample as the sum of all edges pointing to that gene

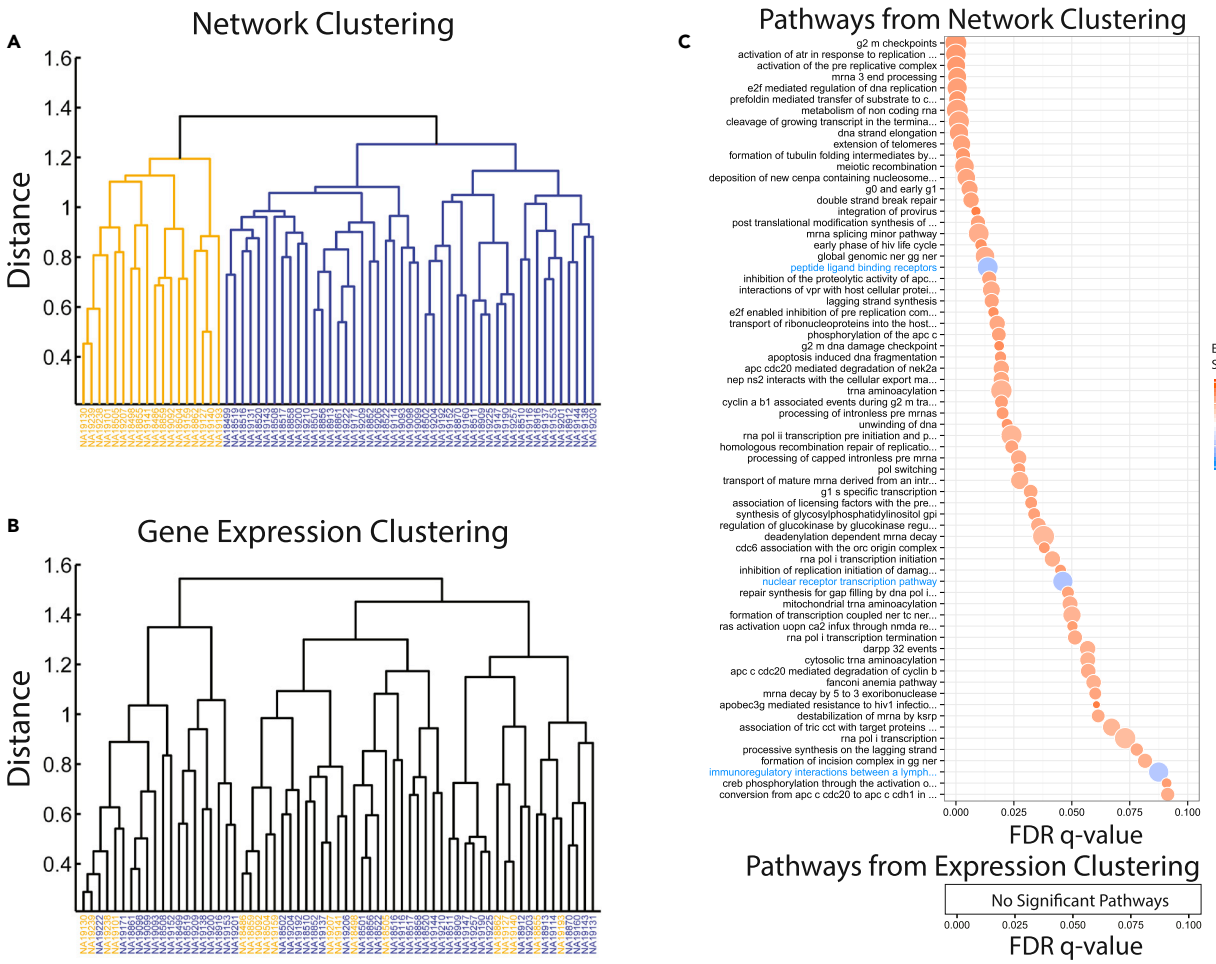


Figure 6. Analysis of Human Lymphoblastoid Cell Line Networks

(A) A hierarchical clustering on the edge weights for 65 regulatory networks, one for each distinct subject-derived lymphoblastoid cell line included in the RNA-seq dataset.

(B) An equivalent hierarchical clustering on gene expression values for these 65 individuals. This clustering is distinct from the one based on the network edge weights. Subject labels are colored based on the network clustering.

(C) Reactome pathways enriched based on GSEA using gene targeting instead of gene expression and comparing samples from the right and left groups of networks presented in (A). No Reactome pathways were identified when comparing the expression values of genes in the different groups defined by the hierarchical clustering presented in (B).

See also Figure S6.

in a single-sample network. We then used GSEA to compare the targeting levels of genes between the two groups of individuals defined by the network-based clustering (Glass et al., 2014). In contrast to the expression-based analysis, in the differential-targeting analysis GSEA found enrichment for many cellular processes related to cell proliferation (in the smaller "orange" cluster; n = 18) and immune function (in the larger "blue" cluster; n = 45; Figure 6C).

Unfortunately, there is little phenotypic information for the 65 individuals in this study, and those available (Choy et al., 2008) are not significantly associated with the groups defined by clustering on either the expression or the network information. However, given our functional enrichment results, we believe that the regulatory differences we observe between the network groups is likely related to differences in cellular growth rate induced by variable Epstein-Barr virus (EBV) levels in the cell lines. EBV is used to transform human B cells into immortalized lymphoblastoids and is known to activate nuclear factor (NF)-κB transcriptional response (Cahir McFarland et al., 1999). Consistent with this hypothesis, we find the signature "Activation of NF-κB in B-cells" highly targeted in the small, "cell proliferation" cluster (ES = 0.5, FDR = 2×10^{-3}).

We also performed GSEA on gene targeting in the aggregate network modeled using all 65 samples. This identified one significant pathway "Activation of IRF3/IRF7 mediated by TBK1/IKK epsilon." This pathway is activated upon cell stimulation with viruses. In fact, Interferon Regulatory Factor 7 (IRF7) was originally identified in the context of EBV infection (Zhang and Pagano, 1997). Although this indicates that this pathway is likely to be highly regulated in all 65 lymphoblastoid samples, it also shows that many of the subtle differences between the individual cell lines cannot be detected with an aggregate network approach. Overall, these results indicate that evaluating single-sample networks can lend insight into the biological processes active in different individuals even when a similar analysis of the gene expression data or the aggregate network model does not.

DISCUSSION

In this article, we present LIONESS as a method for estimating sample-specific regulatory networks. The core principle behind LIONESS is that the addition or removal of even a single sample will slightly perturb an aggregate network model. This perturbation can be used to estimate the contribution of a sample to the aggregate network, and therefore the network of that sample. Importantly, by relying on independent and existing aggregate models to capture the network of the interactions between genes, and a linear interpolation to estimate individual-level differences in the associated edge weights, LIONESS is able to reconstruct network estimates for each sample while preserving the biological complexity of the gene-gene interactions.

There are many network reconstruction methods, but there is no consensus as to the "correct" or "best" one to use—if in fact there is a single method that works best for all data types (Marbach et al., 2012). In the analysis presented here, we used four representative gene expression network reconstruction approaches: Pearson correlation, MI, CLR, and PANDA. These were chosen because they illustrate network reconstruction methods that use either a linear (Pearson) or nonlinear (MI) correlation measure, and the extensions of those measures to better capture true regulatory interactions instead of simple correlative effects. Within this representative collection of methods, our analysis suggests that applying LIONESS to aggregate networks reconstructed using PANDA has the greatest potential for reconstructing accurate network models that can be used to interpret phenotypic differences.

We also note that although we tested our approach in the context of using gene expression to reverse engineer regulatory networks, the linear algebraic framework at the heart of LIONESS is generalizable and can be applied in other settings where aggregate relationships are inferred from a collection of samples. In principle, this includes the application of LIONESS not only to other network inference methods but also in contexts where network relationships are inferred from other multi-sample omics data, such as metabolomics data, genetic/variant data, or epigenetic regulatory information such as CpG methylation. Incorporating this information into single-sample network models will be an important step in understanding the complexity of metazoan gene regulation.

Looking forward, LIONESS provides a way to unite the extensive literature and methodologies for modeling complex network relationships, with statistical analysis techniques that use sample-level information to model heterogeneity. Great progress has been made in assigning patients to disease subgroups based on gene expression profiles, or in using mutational profiles to match individual patients to specific therapies. LIONESS provides a framework in which one could imagine using a similar approach to analyze networks for precision medicine applications. For example, the network-interactions and properties predicted using LIONESS could be directly associated with patient phenotype, genotype, progression, survival, drug response, etc. Therefore LIONESS not only addresses the problem of estimating multiple networks for populations with significant phenotypic of biological heterogeneity but also provides a means of estimating and analyzing networks when samples of a particular phenotype or disease subtype are rare. Ultimately, one could imagine using the LIONESS approach to identify and target the regulatory pathways active in an individual patient (rather than using mutations or gene expression as surrogates for those pathways).

In summary, our approach to modeling single-sample networks is the first single-sample approach that estimates each sample's complete network rather than simply re-purposing differential expression information for network-based analysis. More importantly, LIONESS fills a critical gap, enabling the predictions made by existing network reconstruction methodologies to be evaluated using the same statistical

techniques widely applied in other areas of genomic data analysis. The mathematical framework of LIONESS is highly generalizable and has the potential to be used to study many different and important questions in the fields of precision medicine, health, and biomedical research.

Limitations of Study

One central feature of LIONESS is that it is not a stand-alone algorithm. Rather it is designed to be applied to the output of other aggregate network approaches, and thus is subject to the limitations inherent in those approaches. In this work we focused on testing and evaluating LIONESS in the context of gene regulatory network reconstruction approaches, which primarily leverage expression data to estimate edges and edge weights. It should be mathematically straightforward to apply LIONESS in other contexts, such as to aggregate approaches that consider other network features (e.g., node weights), or to methods that reconstruct different types of networks and/or leverage other types of omics data. Research in these areas is undergoing rapid development. Therefore it will be critical to systematically evaluate LIONESS in these contexts to fully interpret the method's predictions. Finally, it is important to recognize that the accuracy of LIONESS' single-sample networks depends on the accuracy of the method used for inferring the underlying aggregate network. As genomic data continue to grow and methods for network inference improve, this relationship implies that the accuracy of the single-sample networks inferred by LIONESS will only continue to improve.

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

DATA AND SOFTWARE AVAILABILITY

The data sets generated during and/or analyzed during the current study are available in Data S1 (in silico data), and from the Gene Expression Omnibus under accession number GSE4987 (yeast data), GSE19480 (human RNA-seq data), and GSE31388 (human DNase data). The human data is also available online at http://eqtl.uchicago.edu/. Fully processed and normalized versions of the yeast and human data used in this study are also available from the authors upon request.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2019.03.021.

ACKNOWLEDGMENTS

This work was supported by grants from the US National Heart, Lung, and Blood Institute of the National Institutes of Health (R01HL111759, P01HL105339, K25HL133599, 1R35CA220523), from the Charles A. King Trust Postdoctoral Research Fellowship Program, Sara Elizabeth O'Brien Trust, Bank of America, N.A., Co-Trustees, and from the Norwegian Research Council, Helse Sør-Øst, and University of Oslo through the Centre for Molecular Medicine Norway (NCMM). We would also like to thank Farrah Roy, Abhijeet Sonawane, and John Platig for useful insights and suggestions in drafting this manuscript.

AUTHOR CONTRIBUTIONS

Conceptualization, M.G.T. and K.G.; Software, Validation, and Formal Analysis, M.L.K. and K.G.; Methodology and Investigation, M.L.K., M.G.T., and K.G.; Resources, G.C.Y., J.Q., and K.G.; Data Curation and Writing – Original Draft, M.L.K., M.G.T., and K.G.; Writing – Review & Editing, all authors; Visualization, M.L.K. and K.G.; Supervision, J.Q., K.G., and G.C.Y.; Funding Acquisition, M.L.K., G.C.Y., J.Q., and K.G.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: September 10, 2018
Revised: January 30, 2019
Accepted: March 21, 2019
Published: April 26, 2019

REFERENCES

Alvarez, M.J., Shen, Y., Giorgi, F.M., Lachmann, A., Ding, B.B., Ye, B.H., and Califano, A. (2016). Functional characterization of somatic mutations in cancer using network-based inference of protein activity. Nat. Genet. *48*, 838–847.

Cahir McFarland, E.D., Izumi, K.M., and Mosialos, G. (1999). Epstein-barr virus transformation: involvement of latent membrane protein 1-mediated activation of NF-kappaB. Oncogene *18*, 6959–6964.

Choy, E., Yelensky, R., Bonakdar, S., Plenge, R.M., Saxena, R., de Jager, P.L., Shaw, S.Y., Wolfish, C.S., Slavik, J.M., et al. (2008). Genetic analysis of human traits in vitro: drug response and gene expression in lymphoblastoid cell lines. PLoS Genet. *4*, e1000287.

Degner, J.F., Pai, A.A., Pique-Regi, R., Veyrieras, J.B., Gaffney, D.J., Pickrell, J.K., de Leon, S., Michelini, K., Lewellen, N., Crawford, G.E., et al. (2012). DNase I sensitivity QTLs are a major determinant of human expression variation. Nature *482*, 390–394.

DeLuca, D.S., Levin, J.Z., Sivachenko, A., Fennell, T., Nazaire, M.D., Williams, C., Reich, M., Winckler, W., and Getz, G. (2012). RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics *28*, 1530–1532.

ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. Nature *489*, 57–74.

Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., and Gardner, T.S. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol. *5*, e8.

Glass, K., Huttenhower, C., Quackenbush, J., and Yuan, G.C. (2013). Passing messages between biological networks to refine predicted interactions. PLoS One *8*, e64832.

Glass, K., Quackenbush, J., Silverman, E.K., Celli, B., Rennard, S.I., Yuan, G.C., and Demeo, D.L. (2014). Sexually-dimorphic targeting of functionally-related genes in COPD. BMC Syst. Biol. *8*, 118.

GTEx Consortium (2015). Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans. Science *348*, 648–660.

GTEx Consortium, Laboratory, Data Analysis &Coordinating Center (LDACC)—Analysis Working Group, Statistical Methods groups—Analysis Working Group; Enhancing GTEx (eGTEx) groups, NIH Common Fund, NIH/NCI, NIH/NHGRI, NIH/NIMH, NIH/NIDA, Biospecimen Collection Source Site—NDRI, Biospecimen Collection Source Site—RPCI, Biospecimen Core Resource—VARI, Brain Bank Repository—University of Miami Brain Endowment Bank, Leidos Biomedical—Project Management, ELSI Study, Genome Browser Data Integration &Visualization—EBI; Genome Browser Data Integration &Visualization—UCSC Genomics Institute, University of California Santa Cruz;Lead analysts, Laboratory, Data Analysis &Coordinating Center (LDACC), NIH program management (2017). Genetic effects on gene expression across human tissues. Nature *550*, 204–213.

Ho, Y., Costanzo, M., Moore, L., Kobayashi, R., and Andrews, B.J. (1999). Regulation of transcription at the Saccharomyces cerevisiae start transition by Stb1, a Swi6-binding protein. Mol. Cell. Biol. *19*, 5267–5278.

Johnson, W.E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics *8*, 118–127.

Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics *9*, 559.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S.L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. *10*, R25.

Liu, C., Louhimo, R., Laakso, M., Lehtonen, R., and Hautaniemi, S. (2015). Identification of sample-specific regulations using integrative network level analysis. BMC Cancer *15*, 319.

Liu, X., Wang, Y., Ji, H., Aihara, K., and Chen, L. (2016). Personalized characterization of diseases using sample-specific networks. Nucleic Acids Res. *44*, e164.

Loscalzo, J., Kohane, I., and Barabasi, A.L. (2007). Human disease classification in the postgenomic era: a complex systems approach to human pathobiology. Mol. Syst. Biol. *3*, 124.

Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. *15*, 550.

Marbach, D., Costello, J.C., Kuffner, R., Vega, N.M., Prill, R.J., Camacho, D.M., Allison, K.R., Kellis, M., Collins, J.J., and Stolovitzky, G. (2012). Wisdom of crowds for robust gene network inference. Nat. Methods *9*, 796–804.

Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., and Califano, A. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics *7* (*Suppl 1*), S7.

McClellan, J., and King, M.C. (2010). Genetic heterogeneity in human disease. Cell *141*, 210–217.

Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., and Onnela, J.P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. Science *328*, 876–878.

Onnela, J.P., Fenn, D.J., Reid, S., Porter, M.A., Mucha, P.J., Fricker, M.D., and Jones, N.S. (2012). Taxonomies of networks from community structure. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *86*, 036104-36104.

Pickrell, J.K., Marioni, J.C., Pai, A.A., Degner, J.F., Engelhardt, B.E., Nkadori, E., Veyrieras, J.B., Stephens, M., Gilad, Y., and Pritchard, J.K. (2010). Understanding mechanisms underlying human gene expression variation with RNA sequencing. Nature *464*, 768–772.

Pramila, T., Wu, W., Miles, S., Noble, W.S., and Breeden, L.L. (2006). The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. Genes Dev. *20*, 2266–2278.

Rozenblatt-Rosen, O., Stubbington, M.J.T., Regev, A., and Teichmann, S.A. (2017). The human cell Atlas: from vision to reality. Nature *550*, 451–453.

Schlauch, D., Glass, K., Hersh, C.P., Silverman, E.K., and Quackenbush, J. (2017). Estimating drivers of cell state transitions using gene regulatory network models. BMC Syst. Biol. *11*, 139.

De Smet, R., and Marchal, K. (2010). Advantages and limitations of current network inference methods. Nat. Rev. Microbiol. *8*, 717–729.

Sonawane, A.R., Platig, J., Fagny, M., Chen, C.Y., Paulson, J.N., Lopes-Ramos, C.M., Demeo, D.L., Quackenbush, J., Glass, K., and Kuijjer, M.L. (2017). Understanding tissue-specific gene regulation. Cell Rep. *21*, 1077–1088.

Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., and Mesirov, J.P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. U S A *102*, 15545–15550.

Wang, Y.X., and Huang, H. (2014). Review on statistical methods for gene network reconstruction using expression data. J. Theor. Biol. *362*, 53–61.

Wuensche, A. (1998). Discrete dynamical networks and their attractor basins. Complex. Int. *6*, 3–21.

Yang, Y.H., Paquet, A., and Dudoit, S. (2009). marray: Exploratory Analysis for Two-Color Spotted Microarray Data, https://www.bioconductor.org/packages/release/bioc/html/marray.html.

Zhang, L., and Pagano, J.S. (1997). IRF-7, a new interferon regulatory factor associated with Epstein-Barr virus latency. Mol. Cell. Biol. *17*, 5748–5757.

Zhang, Y., Liu, T., Meyer, C.A., Eeckhoute, J., Johnson, D.S., Bernstein, B.E., Nusbaum, C., Myers, R.M., Brown, M., Li, W., and Liu, X.S. (2008). Model-based analysis of ChIP-seq (MACS). Genome Biol. *9*, R137.

ISCI, Volume *14*

Supplemental Information

Estimating Sample-Specific Regulatory Networks

Marieke Lydia Kuijjer, Matthew George Tung, GuoCheng Yuan, John Quackenbush, and Kimberly Glass

Supplemental Items

1. SUPPLEMENTAL FIGURES AND LEGENDS

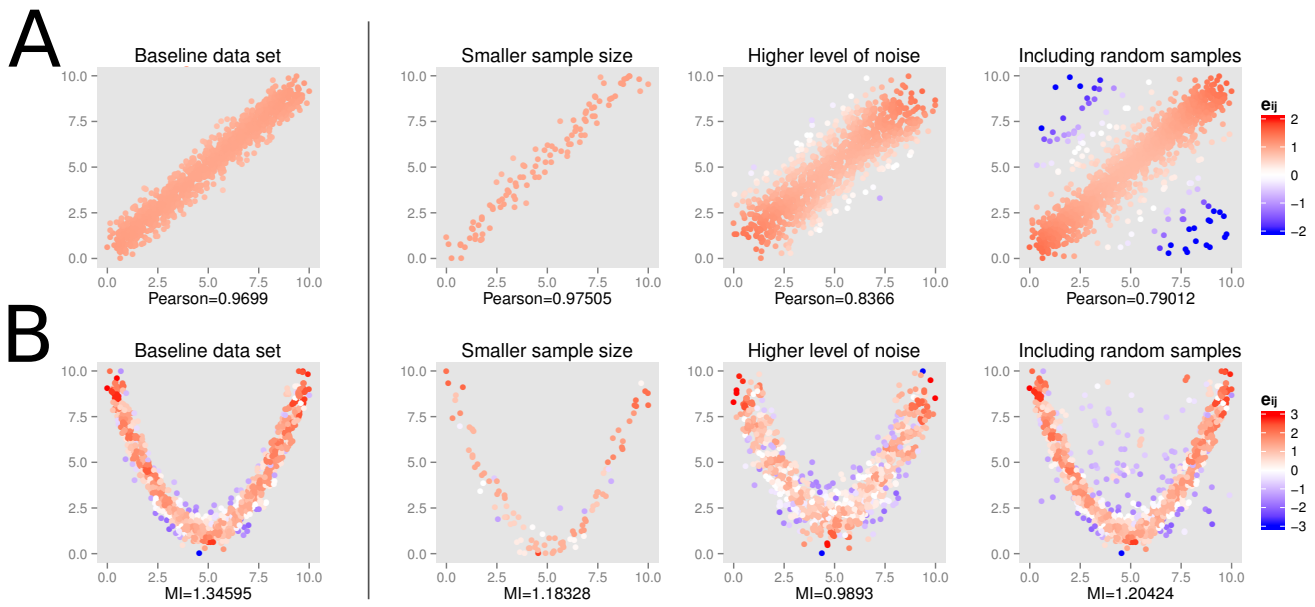


Figure S1: **Examples of single-sample edge weights estimated using LIONESS**, related to Figure 1. LIONESS applied to (A) Pearson correlation or (B) mutual information (MI). Plots show simulated values for two variables that represent the expression levels for two genes across a set of samples. For (A) the samples follow a linear relationship, $x = y$, and for (B) they follow a nonlinear relationship, $y = 10 \cdot (x/5 - 1)^2$. Each dot corresponds to an individual sample. A dot's color indicates the edge weight (e_{ij}) estimated for that sample using LIONESS (red: positive edge weights, blue: negative edge weights, see color legends). The aggregate statistics are shown below the plots. The plots on the left of the vertical bar show a "baseline" data set consisting of $N = 1000$ samples with a low level of Gaussian noise (standard deviation, $sd = 0.1$ for (A) and $sd = 0.05$ for (B)) added to the x versus y relationship. The three plots to the right of the vertical bar show data sets with (1) a lower number of input samples ($N = 100$), (2) a higher level of Guassian noise ($sd = 0.25$ for (A), $sd = 0.1$ for (B)), or (3) additional, randomly-distributed outlier samples ($0.25 \cdot N$ for (A), $0.1 \cdot N$ for (B)). Outliers were generated based on Gaussian noise with mean $= 5$, $sd = 3$, and bounded by $[0, 10]$ using the "truncnorm" package in R.

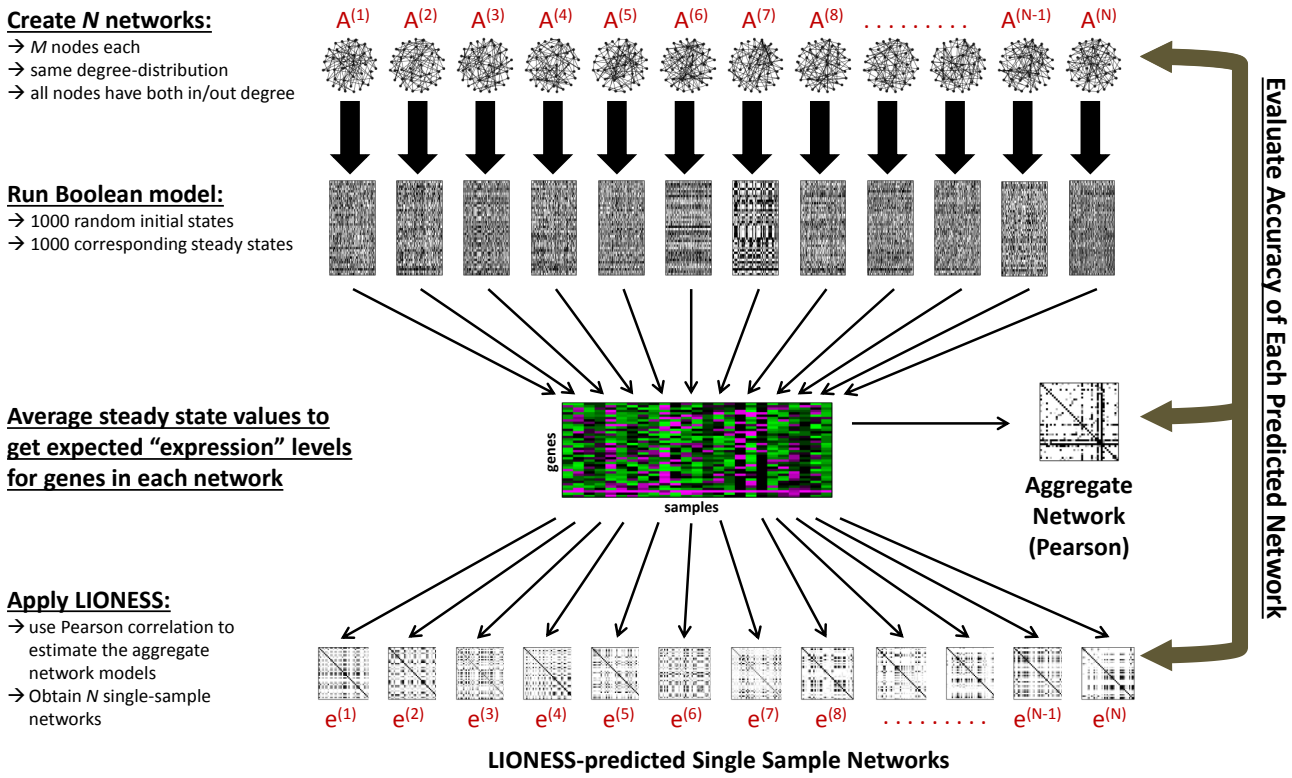


Figure S2: **A schematic overview of how we generated** *in silico* **expression data for a set of known underlying gene regulatory network models**, related to Figure 2.

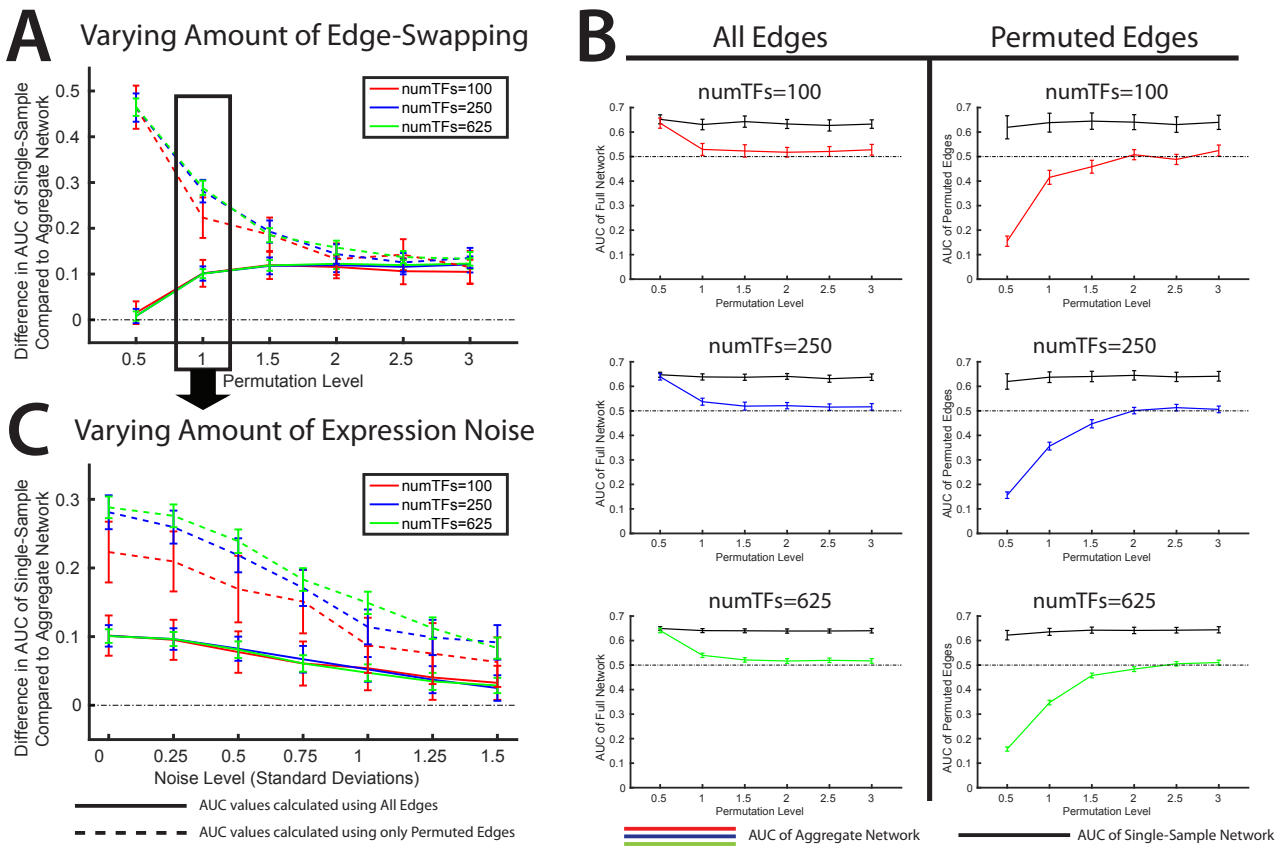


Figure S3: **Results from applying LIONESS to** *in silico* **data sets**, related to Figure 2. (A) A plot showing the difference in AUC values for the aggregate versus the LIONESS single-sample networks across different networks sizes and for different permutation levels. Solid lines show the evaluation across all edges in the network models, dashed lines show the evaluation when only considering edges that are "permuted" (different from the original "seed" network model). The mean and standard deviation across the 100 samples are shown. (B) Plots showing the average AUC values for the single-sample (black) and aggregate (colored) network models across different levels of edge-permutation. The range of AUC values, based on the standard deviation, is indicated by the error bars. The left panel shows the evaluation using all edges and the right shows the evaluation using only "permuted" edges. We see that the aggregate network models do a very poor job at accurately predicting the permuted edges, which are the edges that are truly sample-specific. (C) A plot showing the difference in AUC values between the aggregate and single-sample networks for different networks sizes and with different levels of Gaussian noise added to the expression information. Solid lines show the evaluation across all edges in the network models, dashed lines show the evaluation when only considering "permuted" edges. The mean and standard deviation across the 100 samples are shown.

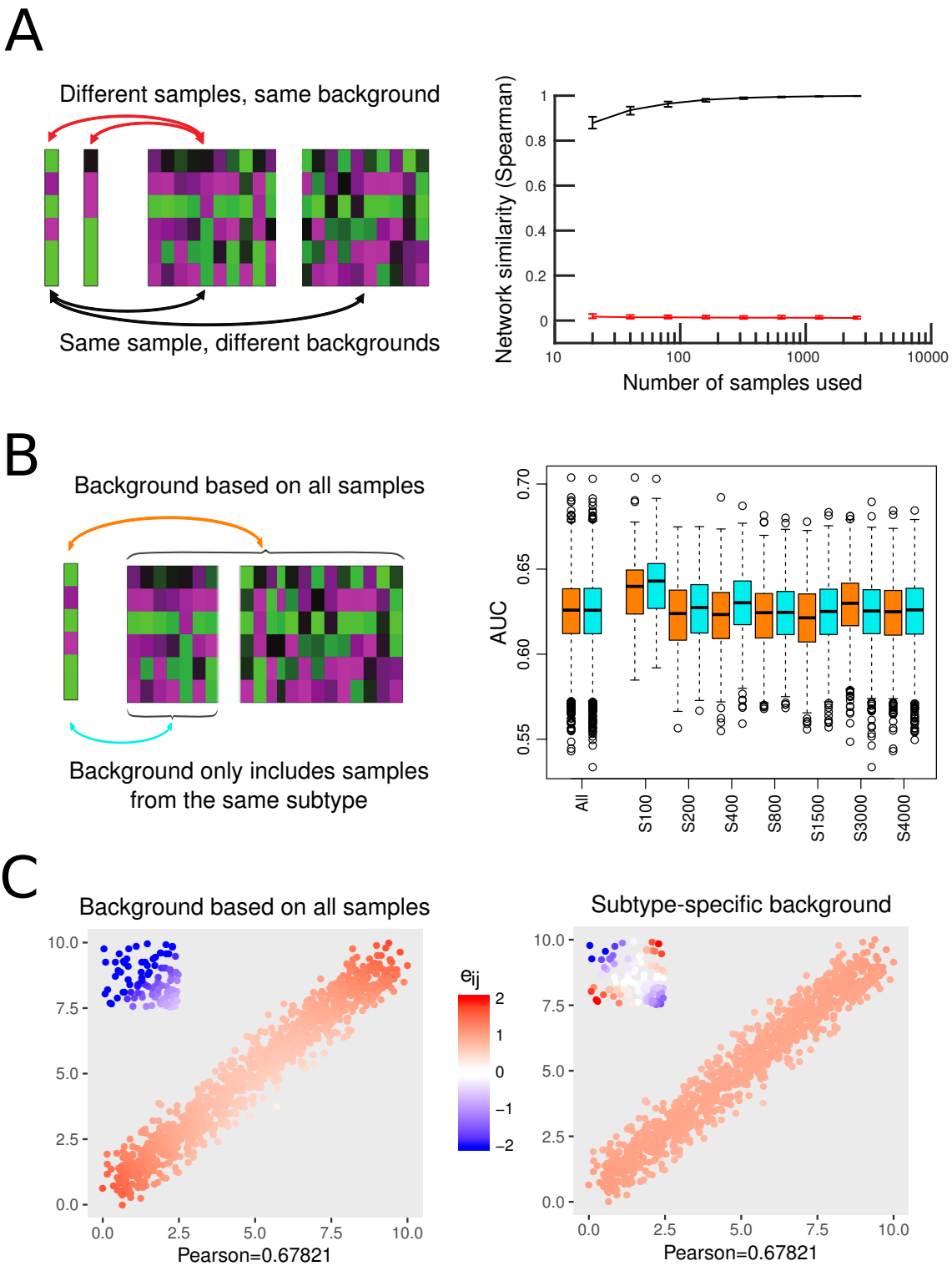


Figure S4: **Results from applying LIONESS using different background data sets**, related to Figure 2. (A) The median and interquartile range of the similarity (calculated using the Spearman correlation) between pairs of networks, as a function of the number of background samples used to calculate the aggregate model. Red: similarity between networks modeled for different samples on the same background, black: similarity between networks modeled for the same sample on different backgrounds. (B) Boxplots visualizing AUCs calculated for networks modeled on the background including all samples (orange) are similar to those calculated for networks modeled on subtype-specific samples (cyan). Boxplots are shown for all 10,000 samples together ("All"), as well as for each subtype separately ("S100"–"S4000"). (C) Examples of single-sample edge weights estimated by applying LIONESS to Pearson correlation for two different subtypes, using all samples in the background (left) or subtype-specific samples only (right).

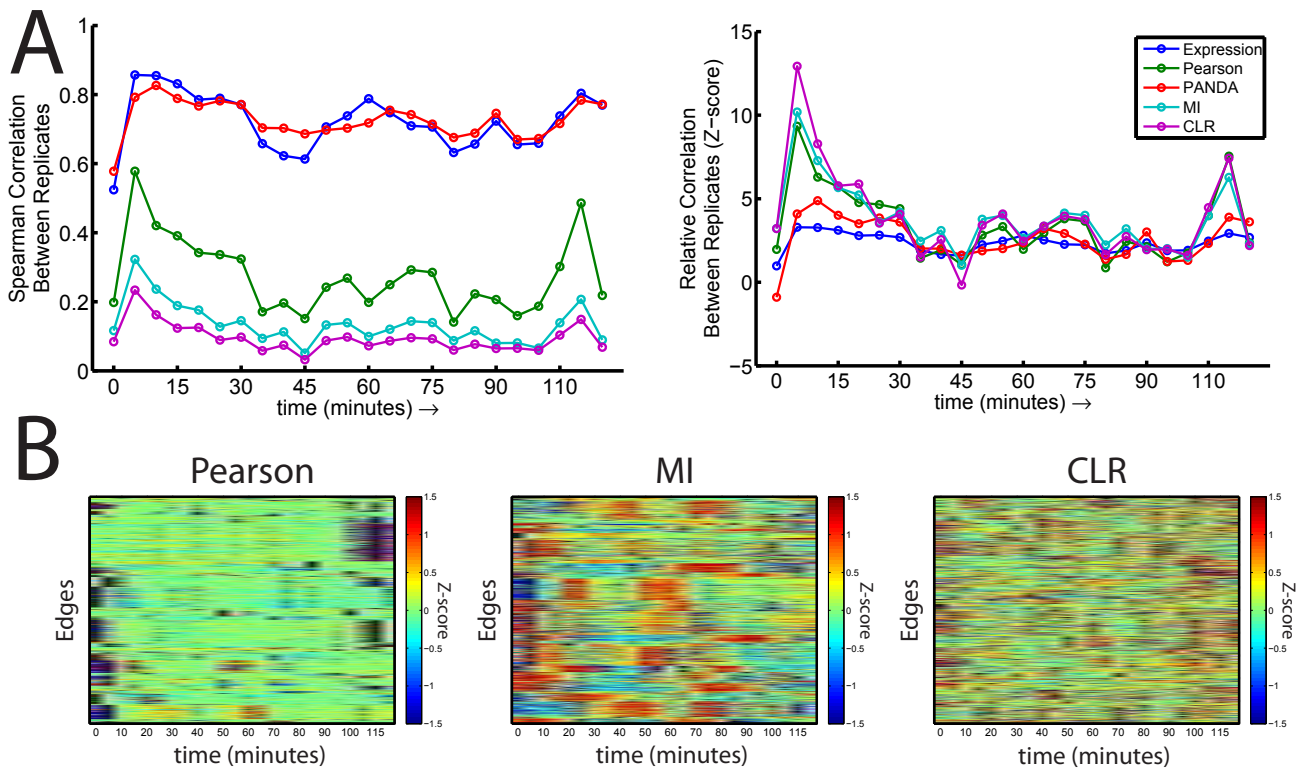


Figure S5: **Results from applying LIONESS to yeast cell cycle data using different aggregate network approaches**, related to Figures 3 and 4. (A) The left panel shows the similarity, as determined by the Spearman correlation, between replicates in the expression data, and the similarity between networks corresponding to those same replicates in the "R1-from-R1 & R2-from-R2" reconstruction. The right panel shows the relative similarity. In this case the Spearman correlation values from the left panel have been converted into Z-scores, based on the mean and standard deviation across the correlation values between all possible pairs of samples (compare with Figure 3 in the main text). (B) Heat maps showing the top 1000 most variables edges identified when applying LIONESS to yeast cell cycle data using various other aggregate reconstruction approaches. Although LIONESS is a generalizable approach, these results highlight the importance of selecting a robust underlying aggregate reconstruction algorithm when applying LIONESS.

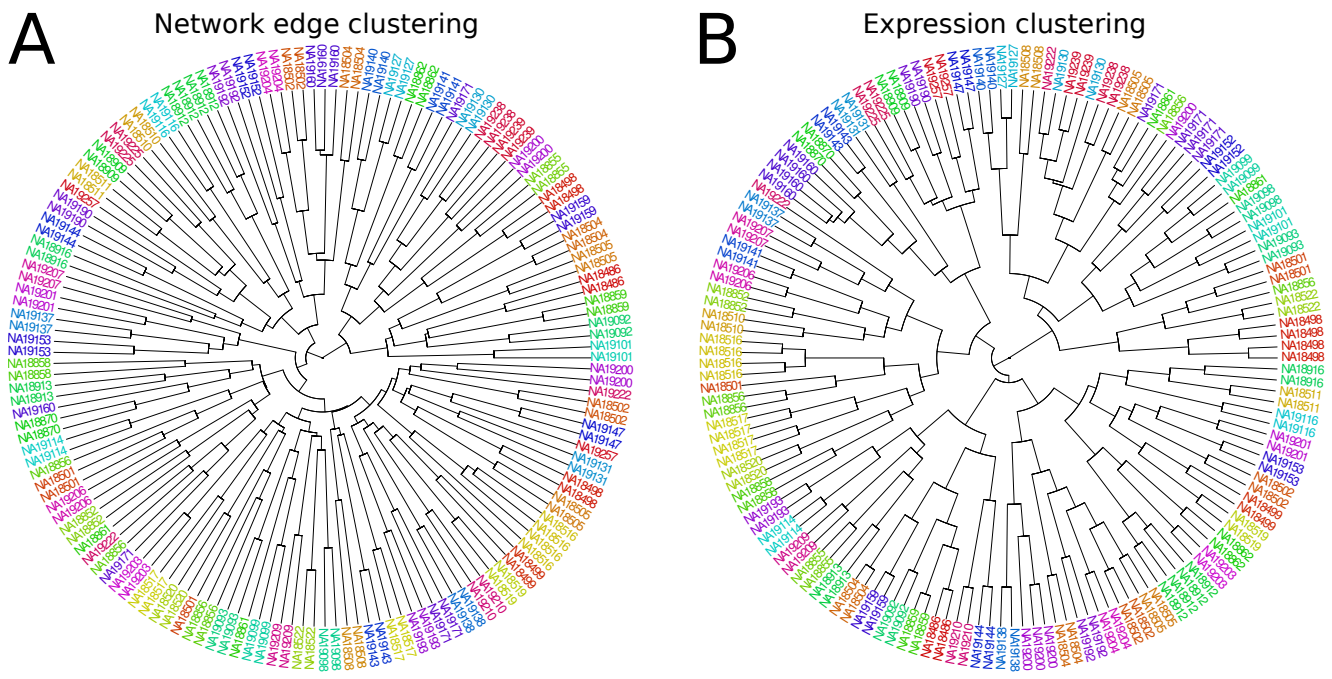


Figure S6: **Hierarchical clustering of lymphoblastoid cell line networks and expression data**, related to Figures 5 and 6. A hierarchical clustering of (A) the 153 single-sample lymphoblastoid cell line networks predicted using LIONESS and (B) the gene expression data that was used to build the networks. Closer inspection reveals that technical replicates (different experimental samples assaying the same cell line) tend to cluster together. Different replicates cluster together in the network and expression dendrograms; this includes 121/153 of the single-sample networks and 121/153 of gene expression samples.

2. SUPPLEMENTAL TABLES

Method(s)	General Approach	Assumptions / Limitations	Input Information	Predictions / Output	Estimates Common Network Structures?
ssMARINA[1] VIPER[2]	Quantify **ssDE** of genes, evaluate association of ssDE profile with gene set	Only evaluates node-level enrichment (not edges)	gene sets	Proteins/TFs associated with ssDE genes	no
DERA[3]	Quantify **ssDE** of genes, identify edges in input network connected to ssDE genes	Cannot recover ss-edges not associated with differential-expression or not in input network	literature-curated / "known" network	edges in input network connected to ssDE genes	no
ssPCC[4][5]	Quantify **ssDC** (probability an edge is specific to each sample)	Assumes positive linear correlation indicates edge; only identifies ss-edges	Pearson correlation	edges specific to a sample (excludes common edges)	no
LIONESS	Use interpolation to estimate the complete network of each sample	Assumes ss-networks, on average, represent the aggregate network	aggregate network (reconstructed using existing method)	networks for each sample (both sample-specific and common edges)	yes

Abbreviations: ssDE = single-sample differential-expression; ssDC = single-sample differential-correlation; ssPCC = single-sample Pearson correlation; ss-edges = single-sample edges (edges that are specific to an individual sample); ss-networks = single-sample networks (a set of networks, each associated with an individual sample)

References:

[1] Aytes, A. *et al.* Cross-species regulatory network analysis identifies a synergistic interaction between FOXM1 and CENPF that drives prostate cancer malignancy. *Cancer Cell* **25**, 638-651, doi:10.1016/j.ccr.2014.03.017 (2014).

[2] Alvarez, M. J. *et al.* Functional characterization of somatic mutations in cancer using network-based inference of protein activity. *Nat Genet* **48**, 838-847, doi:10.1038/ng.3593 (2016).

[3] Liu, C., Louhimo, R., Laakso, M., Lehtonen, R. & Hautaniemi, S. Identification of sample-specific regulations using integrative network level analysis. *BMC Cancer* **15**, 319, doi:10.1186/s12885-015-1265-2 (2015).

[4] Zhang, W., Zeng, T. & Chen, L. EdgeMarker: Identifying differentially correlated molecule pairs as edge-biomarkers. *J Theor Biol* **362**, 35-43, doi:10.1016/j.jtbi.2014.05.041 (2014).

[5] Liu, X., Wang, Y., Ji, H., Aihara, K. & Chen, L. Personalized characterization of diseases using sample-specific networks. *Nucleic Acids Res* **44**, e164, doi:10.1093/nar/gkw772 (2016).

Table S1: **A summary of single-sample analysis approaches**, related to Figure 1.

Application:	*in silico* data (Figure 2C, SFigure 3)	*in silico* data (Figure 2D-E, SFigure 4)	Yeast data (Figures 3-4, SFigure 5)	Human data (Figures 5-6, SFigure 6)
# TFs/Genes/Samples:	Various[1]-genes/100-Samples	100-genes/10000-Samples[2]	105-TFs/3551-genes/48-Samples	158-TFs/12424-genes/153-Samples
Expression Data Used to Reconstruct Networks:	Generated using a Boolean model (various parameters)[3]	Generated using a Boolean model	mRNA expression across yeast cell-cycle [1]	RNA-seq on sixty-five lymphoblastoid cell-lines [2]
Methods Used to Reconstruct Networks:	Pearson	Pearson, PANDA, MI, CLR	Pearson, PANDA, MI, CLR	PANDA
Prior Data Used to Initialize PANDA networks:	N/A[4]	N/A[4]	TF-motif scan [3,4]	TF-motif scan [5,6]
Benchmarks for each Single-sample network:	Models used to generate the expression data	Models used to generate the expression data	N/A	N/A[5]

Table Footnotes:

(1) Size={100,250,625}-genes. Edge perturbation level, ρ=[0.5:0.5:3].
(2) Permuted networks are based on the same seed network as in the Figure 1C / SFigure 3 *in silico* data. Edge permutation level, ρ =1.
(3) For the ρ =1 data, generated expression data that contained Gaussian noise = [0:0.25:1.5]*σ.
(4) An identity matrix was used to initialize PANDA for the *in silico* analysis.
(5) Although there is no single-sample network benchmark for human data, we do have DNase1 hypersensitivity profiles for each of the individuals in the dataset, which we used to corroborate the network findings.

References for Data Sources:

[1] Pramila, T., Wu, W., Miles, S., Noble, W.S., and Breeden, L.L. (2006). The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. Genes Dev *20*, 2266-2278.
[2] Degner, J.F., Pai, A.A., Pique-Regi, R., Veyrieras, J.B., Gaffney, D.J., Pickrell, J.K., De Leon, S., Michelini, K., Lewellen, N., Crawford, G.E., *et al.* (2012). DNase I sensitivity QTLs are a major determinant of human expression variation. Nature *482*, 390-394.
[3] Harbison, C. T. *et al.* Transcriptional regulatory code of a eukaryotic genome. *Nature* **431**, 99-104, doi:10.1038/nature02800 nature02800 (2004).
[4] http://fraenkel.mit.edu/Harbison/release_v24/txtfiles/
[5] Mathelier, A., Zhao, X., Zhang, A.W., Parcy, F., Worsley-Hunt, R., Arenillas, D.J., Buchman, S., Chen, C.Y., Chou, A., Ienasescu, H., *et al.* (2014). JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. Nucleic Acids Res *42*, D142-147.
[6] Pinello, L., Xu, J., Orkin, S.H., and Yuan, G.C. (2014). Analysis of chromatin-state plasticity identifies cell-type-specific regulators of H3K27me3 patterns. Proc Natl Acad Sci U S A *111*, E344-353.

Table S2: **A summary of the analyses performed in this manuscript and the data used for each**, related to Figure 1.

3. TRANSPARENT METHODS

This section contains additional information regarding the data processing and analyses presented in the main text of "Estimating Sample-Specific Regulatory Networks". A summary of the data and analyses described in this supplement, and presented in both the main text, figures, and supplemental figures, is presented in Table S2.

3.1. Current single-sample analysis approaches

Several existing methods quantify the expression differences associated with a single-sample (as compared to a background set of samples) and use that information in a network-type of analysis. These approaches all use differential analyses—either differential-expression or differential-correlation—to highlight information specific to a single sample. In other words, by design, they cannot estimate network structures common across all samples. These common structures are critical in analyzing networks and are needed to correctly quantify topological characteristics such as community structure, as well as node and edge centralities (Sonawane et al., 2017). An overview of existing single-sample analysis approaches is included below and the methods are summarized in Table S1.

Single-sample differential-expression

The main way others have used single-sample information in a network analysis, is to start with a single "known" network and then overlay sample-specific expression information to identify the parts of this network that may be relevant in a sample-specific context. For these approaches, a single-sample differential-expression (ssDE) profile is first constructed by comparing the expression of genes in a given sample with an expected distribution of expression values across a background set of samples. One common way to quantify ssDE is using a Z-score approach. In this case, the mean and standard deviation of a gene's expression across samples is calculated; the expression of that gene in a given sample is then normalized by subtracting the mean and dividing by the standard deviation. Single-sample differential-expression has been used in network analysis approaches in two main ways.

Single-sample gene-set enrichment: Both ssMARINa (single-sample MAster Regulator INference algorithm) (Aytes et al., 2014) and VIPER (Virtual Inference of Protein activity by Enriched Regulon) (Alvarez et al., 2016) use "sample-specific signatures" obtained from ssDE analysis together with profiles for transcription factor targets to estimate the overall activity of transcriptional regulators and/or proteins in individual samples. However, these methods do not provide an estimate of the actual single-sample networks that may be leading to this differential activity.

ssDE layered onto an input network: In DERA (Differentially Expressed Regulation Analysis) (Liu et al., 2015), a prior biological network is built from public databases, the various sample-specific portions of this network are estimated by "coloring" genes based on the ssDE results. These sample-specific subnetworks are analyzed to identify a core set of interactions commonly-identified across a group of samples. We note that using this type of approach, an edge that is specific to an individual sample, but not present in this original prior network, will never be identified. One can imagine that these interactions may be biologically important, such as when a mutation causes a protein to change its interacting partners (Wang et al., 2015).

Single-sample differential-correlation

Another way others have used single-sample expression information in a network analysis is to apply a statistical approach to quantify single-sample differential-correlation (ssPCC) (Liu et al., 2016; Zhang et al., 2014). Pearson correlation follows a normal distribution, therefore the difference between two distributions of Pearson correlations can be statistically quantified using the Z-score. In other words, by calculating the Pearson correlation both with and without a sample of interest, this known relationship can be used to transform the difference between those correlations into a value representing the ssPCC.

One mathematical assumption made by ssPCC is that every edge has the same distribution of weight-values across the predicted single-sample models. In other words, all edges have equal probability of being identified across the population. Therefore, to remove false positives and generate interpretable results, ssPCC has been used to differentially-weight edges in a known biological network (e.g. documented protein-protein interactions in StringDB) (Liu et al., 2016). As with DERA (see above), we point out that by doing this filtering, an edge that is specific to an individual sample, but not present in this original prior network, will never be identified.

Contrast with the LIONESS approach

We emphasize that none of the above approaches for analyzing single-sample information are designed to directly estimate sample-specific networks. Both ssDE are ssPCC are conceptually quite similar. Importantly, by only quantifying how specific a node (ssDE) or edge (ssPCC) is to a specific sample, both approaches effectively mask any network relationships that may be common across the samples. In contrast, LIONESS is designed to estimate both sample-specific and common network relationships. In other words, LIONESS is fundamentally different from these existing single-sample analysis methods in that it estimates each sample's complete network rather than simply re-purposing differential-expression information for network-based analysis.

Furthermore, we point out that the approaches outlined above ignore both the extensive literature on network reconstruction methods, as well as the fact that regulatory networks are often characterized by nonlinear dynamics and synergistic effects. This is especially true of ssPCC, which is simply a re-framing of the residuals obtained from running a linear Pearson correlation analysis.

3.2. Aggregate network reconstruction approaches

Many methods have been developed for inferring biological networks. In our manuscript we analyze the application of LIONESS to four specific methods: (1) Pearson correlation, (2) PANDA (Passing Attributes between Networks for Data Assimilation), (3) mutual information, and (4) Context Likelihood of Relatedness (CLR). These methods were chosen because they represent a set of network reconstruction methods that use either a linear (Pearson) or nonlinear (mutual information) correlation measure, and methods that extend those measures to try to better capture regulatory interactions.

Pearson Correlation: Pearson correlation evaluates the degree of a linear relationship between two variables. Regulatory networks can be reconstructed by calculating the Pearson correlation coefficient between the expression levels of each TF and each target gene. These coefficients are measures of whether TFs and target genes are being co-expressed, which may indicate a regulatory event.

Passing Attributes between Networks for Data Assimilation (PANDA): PANDA (Glass et al., 2013) builds regulatory networks by starting with a prior of possible interactions between TFs and target genes, for example TF motif binding information. PANDA integrates this regulatory prior with gene expression information and protein-protein interaction data, using a message passing approach to determine information flow between the different data types. The message passing algorithm used in PANDA is based on the assumption that if the expression of two genes correlate, those genes are more likely to be regulated by similar sets of TFs than two genes that do not show correlation in expression.

Mutual Information (MI): MI compares the joint probability distribution of two variables to the products of their corresponding marginal distributions. Similar to Pearson correlation, MI is a measure of association between TFs and target genes, which may indicate a regulatory event. This method does not assume linearity or continuity of the data used for building the network.

Context Likelihood of Relatedness (CLR): CLR (Faith et al., 2007) is based on MI, but applies a double Z-score transformation to the MI scores to produce a "normalized" value for each edge (z_{ij}). This transformation normalizes each TF-gene interaction based on the background distribution of MI values for each gene (z_j), and the background distribution of MI values for each TF (z_i): $z_{ij} = \sqrt{z_i^2 + z_j^2}$; note that any edge for which either $z_i < 0$ or $z_j < 0$ is given a final weight of zero ($z_{ij} = 0$).

Running LIONESS on networks reconstructed using PANDA

We used the *corr()* function and a version of the PANDA algorithm implemented in MATLAB (Glass et al., 2015) to reconstruct networks using Pearson correlation and PANDA, respectively. PANDA requires a prior regulatory network in addition to gene expression information. For the *in silico* data we used an identity matrix (corresponding to each transcription factor targeting only itself) as the prior regulatory network. For the yeast and human data, a prior regulatory network was constructed based on transcription sequence-motif information (see below). It is worth noting that although PANDA can optionally take protein-protein interaction (PPI) information as an input, we did not use PPI data to provide a fair comparison with other network reconstruction approaches.

To calculate the mutual information and reconstruct networks based on the Context Likelihood of Relatedness (CLR), we used the *build.mim()* and *clr()* functions within the "minet" package in R/Bioconductor (Meyer et al., 2008). For the mutual information application in *in silico* data we used an estimator based on the entropy of the empirical probability distribution (estimator="mi.empirical"), with 100 bins of equal width (disc="equalwidth"); the application in yeast used default parameters. It is worth noting that these algorithms create symmetric gene by gene matrices of predicted edge scores. Therefore, in the context of reconstructing single-sample yeast *regulatory* networks, we reduced these aggregate networks by selecting only the portion of the predicted gene-by-gene matrix that corresponds to edges from a transcription factor to a target gene.

3.3. Building an intuition for LIONESS using Pearson correlation and mutual information

One advantage of the LIONESS approach is that the input edge-weight estimates can, in theory, come from any network inference method that leverages information across a set of samples. With this in mind, in order to gain a better understanding and intuition for LIONESS, we performed a detailed exploration of its behavior when the aggregate network model is calculated using two widely-used measures: Pearson correlation and mutual information. In particular, to gain a better intuition for the values estimated by LIONESS for these two measures, we simulated data for pairs of nodes that had either (1) a strong linear ($\rho = 0.9699$), or (2) a strong nonlinear ($MI = 1.34595$) relationship across multiple samples and applied LIONESS to these data.

In the linear case (Figure S1A), we used Pearson correlation as our aggregate model and applied the LIONESS equation (Equation 4 in the main text) to estimate edge-weights for each sample in the data. We observe that all samples are given similar edge-weights by LIONESS, with an average value of 0.9702. We then repeated this analysis for (1) a smaller number of samples, (2) increased noise, and (3) the introduction of outliers that are inconsistent with the expected linear relationship. We find that the edge-weights estimated by LIONESS are very robust to sample-size. Increasing the noise does decrease the edge-weight estimates for the samples that are farthest from the linear trend-line, but this is a desired outcome as these samples are the least consistent with the expected relationship. Similarly, when we add in samples to the data that are, by design, inconsistent with the expected linear relationship, LIONESS correctly identifies and gives strong negative edge-weights to those outlier samples.

In the nonlinear case (Figure S1B) we used mutual information to estimate the nodes' aggregate relationship before applying the LIONESS equation. We find that mutual information is a slightly more sensitive measure than Pearson correlation, with samples that differ from the expected relationship more readily identified and down-weighted by LIONESS. Other than this, however, the conclusions from the mutual information and Pearson correlation analyses are very similar. LIONESS is highly robust to the number of samples used in both cases, and gives higher edge-weights to samples near the expected relationship but lower weights to the samples that are inconsistent with the overall expected relationship.

LIONESS-predicted single-sample edge weights in the case of multiple expression subtypes: We also explored LIONESS' behavior when including only subtype-specific data in the background or when including a more heterogeneous background data set (for more information, see the "Analysis with multiple subtypes" below). We used data from the Pearson correlation example (see Figure S1 legend) as the "main subtype," which showed a strong positive correlation between two genes. We then simulated an additional subtype by adding samples for which the two genes did not correlate. We generated expression levels in this subtype using the truncnorm package in R, with settings mean=5, sd=3, and bounded between [0, 2.5] for values on the x-axis and between [7.5, 10] for values on the y-axis. We next estimated LIONESS edge weights based on a background including all samples from both subtypes (Figure S4C, left) and based on a background consisting of subtype-specific samples only (Figure S4C, right). When including all samples in the aggregate network model, we observe a clear difference in edge weights estimated for these two different subtypes. Samples in the subtype in which these genes are correlated receive high edge weights (red), while samples from the subtype in which the genes are not correlated receive low edge weights (blue). When only including subtype-specific samples in the aggregate network model (right), this clear distinction is lost. In addition to the expected negative edge weights for the subtype in which the two genes are not correlated, neutral and positive edge weights are assigned to a subset of samples in this subtype.

3.4. Data generation, normalization, and relevant pre-processing steps

Generation of the in silico *expression data and regulatory networks*

Data generation: To test LIONESS's ability to reconstruct sample-specific network models we generated a set of network models and a corresponding associated set of gene expression profiles (Figure S2). To begin, we created a single "seed" network model with M nodes. To approximate the structure of biological networks (Albert, 2005), the out-degree of nodes in this seed model were given a power-law degree distribution (generated using the approach published in (Clauset et al., 2009), with $\alpha = 3$) with their targets selected randomly. We ensured that the out-degree and in-degree of all nodes in this seed network was greater than zero.

Next we randomized this "seed" network model, holding the degree distribution fixed, by performing $\rho \times N_e$ edge-swaps (where N_e is the number of edges and ρ allows us to control how different the permuted network is from the initial seed network). Then we generated a set of initial Boolean states for each node in the network, and determined the subsequent states of the nodes using Stouffer's Z-score method:

$$S_j^{(t+1)} = round\left[CDF^{-1}\left(\frac{\sum_i Z_{ij} S_i^{(t)}}{\sqrt{\sum_i S_i^{(t)}}}\right)\right] \tag{1}$$

where CDF^{-1} is the inverse cumulative distribution function for the normal distribution, Z_{ij} is the Z-scored weight of an edge from node i to node j, and $S_i^{(t)}$ is the state of node i at time t. This Boolean model was run until an attractor solution was found. In total we generated 1000 random initial Boolean states for each randomized network, resulting in 1000 attractor solutions. The expression level of a node in the randomized network was then estimated as the average across these steady-state solutions. This entire process was then repeated N times to create N total randomized versions of the "seed" network model and N corresponding matched expression samples. We applied this approach to generate sets of *in silico* networks that (1) are of various size (varying M), (2) have different levels of inter-network variability (varying ρ), (3) have varying levels of noise added to their associated expression data (see below), or (4) have a large number of samples (increasing N).

Analysis with varying levels of edge permutation: For our analyses, we created three initial "seed" networks of different sizes, with M equal to 100, 250, and 625 nodes. To evaluate the impact of between-sample heterogeneity on LIONESS's prediction, for each of these seed networks we created six different sets of network-models based on different levels of edge-permutation, with ρ equal to 0.5, 1, 1.5, 2, 2.5, or 3. Next, we ran the Boolean model described above on each of the generated networks. In total this process created 18 sets of $N = 100$ "gold-standard" networks and the corresponding gene-expression levels for these 100 samples. These data sets represent networks of different sizes and permutation levels relative to the initial "seed". For each of these data sets, we constructed all 100 single-sample networks by applying LIONESS to aggregate networks based on the Pearson correlation in gene expression levels. We benchmarked these 100 single-sample networks against the 100 "gold-standard" networks in the data set. We also separately benchmarked only the "permuted" edges. To identify permuted edges we compared the edges in each of the gold-standards with the original "seed" network from which those standards were derived, and identified the subset of edges that only exist in either the "seed" or the "gold-standard" network. We then evaluated the AUC of the LIONESS-predicted single-sample edge-weights for this subset of edges. We observe that LIONESS estimates these edges incredibly well, with an overall accuracy similar to the other "non-permuted" edges (Figure S3A–B). In contrast, the aggregate network is completely unable to estimate the "permuted" sample-specific edges, especially in cases of low heterogeneity (low values of ρ, which correspond to fewer edge-swaps).

Analysis with varying levels of expression noise: We took the data sets associated with an edge-permutation level of $\rho = 1$. This included three data sets, representing network-sizes of M equal to 100, 250, or 625 nodes. For each of these data sets, we determined the standard deviation (σ_i) of each gene's (i) expression across the samples. Then, for each gene, we used the *normrnd()* function in MATLAB to generate 100 noisy expression-values (one for each sample) based on a Gaussian distribution centered at that gene's original "correct" expression level in the sample, and with a standard deviation set to $r \times \sigma_i$. We did this for a range of values for r: 0.25, 0.5, 0.75, 1, 1.25, and 1.5. This resulted in eighteen additional *in silico* expression-sets (six levels of noise for the three data sets associated with different network sizes). Note that setting $r = 0$ is equivalent to using the original expression-set without any additional noise. For each of these expression-sets we constructed the 100 corresponding single-sample networks by applying LIONESS to aggregate networks based on the Pearson correlation. We benchmarked both the full network models and the "permuted edges" in these 100 single-sample networks against the 100 "gold-standard" networks in

the original data set. We find that as noise is added, the overall difference in AUC of the single-sample networks relative to the aggregate network model does decrease, but remains greater than zero (Figure S3C). Importantly, we do not lose the ability to accurately predict the truly sample-specific "permuted" edges.

Analysis with varying numbers of samples: We selected the seed network associated with the $M = 100$ nodes and created a data set with a moderate amount of heterogeneity among the networks ($\rho = 1$) and containing $N = 10,000$ samples. We selected subsets of these data consisting of various numbers of samples and applied LIONESS to aggregate networks build using Pearson correlation. We observe that including more samples increased the accuracy of the overall aggregate network, but that this corresponded to a poor prediction of sample-specific (permuted) edges. On the other hand, for the LIONESS single-sample networks, both the overall accuracy and the accuracy of the sample-specific edges is robust to the number of samples used. We also used this data set to assess the accuracy of LIONESS networks built using different aggregate reconstruction approaches: Pearson correlation, PANDA, mutual information, and CLR. For this final evaluation we applied each of the four reconstruction approaches to build four aggregate networks based on all 10,000 *in silico* expression samples. We then applied LIONESS to each of these aggregate networks, generating 10,000 sample-specific networks for each of the approaches.

Analysis with various sample backgrounds: In order to solve for a single-sample network, a "background" set of samples is used to reconstruct and , or aggregate networks which do and do not include the sample for which we are building the single-sample network model, respectively. For the *in silico* data, we performed an analysis wherein we compared two single-sample networks which represent the same expression-sample (q), but which were constructed using independent "background" samples. To select samples for the two "backgrounds" in this analysis, we first randomly ordered all samples in our in silico data; the first M samples were assigned to background one, and samples $(M + 1)$ to $2M$ were assigned to background two. This ensured that the two "backgrounds" contained completely independent sets of samples. To assess how sensitive LIONESS networks are to the chosen set of background samples, we evaluated the similarity between these pairs of single-sample networks. We observe high reproducibility, especially as we increase the number of background samples used to reconstruct the aggregate models, with near identity for large background sample sizes (Figure S4A). As a control, we also compared two single-sample networks representing different gene expression samples that were derived using the same set of background samples in the aggregate model. We see almost no similarity between these networks, especially with increasing numbers of background samples.

Analysis with multiple subtypes: To assess whether LIONESS performs best when using a homogeneous background dataset (for example, consisting of samples from a specific subtype) or whether LIONESS returns better predictions when including a more heterogeneous set of background samples, we generated an in silico data set that included 10,000 total samples representing seven different subtypes with varying sample size. We started by creating a "seed" network with $M = 100$ nodes generated as described above. We then created seven distinct permutations of this seed network, based on swapping the labels of 50% of the nodes in the network; this ensured that the networks had the same basic underlying structure (same density and degree distribution), but also distinct basins of attraction (which correspond to distinct gene expression patterns). Next, for each of these seven networks, we performed edge-permutations with $\rho = 1$, as described above. For each subtype we created a different number of corresponding sample-specific networks based on these permutations. This resulted in seven sets of subtype-specific seed networks with sample size equal to 100, 200, 400, 800, 1,500, 3,000, and 4,000. We generated expression data for each of these "gold standard" networks by running the Boolean model described above (Equation 1). We then constructed single-sample networks by applying LIONESS to aggregate networks modeled using Pearson correlation on the expression levels. We did this using two approaches—one that included all samples in the background dataset, and one that included only samples from the specific subtype in the background. Finally, we evaluated the AUC of the LIONESS-predicted edge weights based on the original "gold standard" networks. We found minimal differences in the accuracy of sample-specific networks when either using all samples as a background, or only those samples that belonged to the same subtype as the sample in question (Figure S4B).

Processing the yeast cell cycle expression data

GPR files associated with (Pramila et al., 2006) were downloaded from the Gene Expression Omnibus (GEO; accession GSE4987). Each of two replicates were separately ma-normalized using the *maNorm()* function in the "marray" package in R/Bioconductor (Yang et al., 2007). The data was batch-corrected using the *ComBat()* function in the "sva" package (Leek et al., 2014) and probe-sets mapping to the same gene were averaged, resulting in expression values for 5088 genes across fifty samples, twenty-five from each of the two replicate data sets. Two samples (corresponding to the 105 minute time point) were excluded for data-quality reasons, as noted in the original publication, and genes without motif information (see below) were then removed, giving a final expression data-set containing 48 samples and 3551 genes. These data were quantile-normalized and used in all subsequent analyses.

Generating the yeast motif prior data for PANDA

PANDA requires a prior regulatory network structure in addition to gene expression information. To construct a motif prior network for yeast we downloaded predicted binding sites for 204 yeast transcription factors (Fraenkel Lab, 2004; Harbison et al., 2004; MacIsaac et al., 2006). These data include 4360 genes with tandem promoters. 3551 of these genes are also covered on the gene expression array (see above). 105 total transcription factors in this data set target the promoter of one of these 3551 genes. The motif map between these 105 transcription factors and 3551 target genes was used as a prior regulatory network input to the PANDA algorithm. A subset of 65 of these transcription factors that also had expression information was used to reduce the size of the Pearson, MI, and CLR predicted networks to edges that extend between transcription factors and genes.

Processing the human RNA-Seq data

RNA sequencing (RNA-Seq) data (Pickrell et al., 2010) were downloaded from the Pritchard lab website (`http://eqtl.uchicago.edu/`, accessed April 2014; also available on the Gene Expression Omnibus, GEO: GSE19480). 173 different samples corresponding to 74 different cell lines were available for download. We aligned all samples to the hg19/GRCh37 reference genome using Bowtie (Langmead et al., 2009), with options -n 3 and -m 1, allowing for not more than 3 mismatches in the seed (28 bases on the high-quality end of the read, the default in Bowtie), and suppressing all non-unique alignments. We used RNA-SeQC (DeLuca et al., 2012) to determine the quality of the reads, using an expression profiling efficiency cut-off of 0.75. Cell lines NA19119 and NA18853 fell below this cut-off. Next we used subread (Liao et al., 2013) to count reads, and the subread algorithm featureCounts to assign and summarize counts to genes. Finally, we removed samples with poor quality reads, and samples for which we did not have good quality DNase hypersensitivity data available (see below), leaving us with 153 samples corresponding to 65 different cell lines.

We used the "DESeq2" (Love et al., 2014) package to analyze read counts for these 153 samples and adjusted for different library sizes using the *estimateSizeFactors* function. Only genes that had raw counts in at least 50% of all 153 samples (21516/57820, or 37% of all genes) were retained for further analysis. Correction for gene length was also performed; each intensity value was divided by the length of the corresponding gene (defined as the total length of the genomic region covered by the features/exons) in the "gene_id" meta-feature in featureCounts. Finally, Ensembl gene ids were converted to HGNC gene symbols (16901/21516 genes) using R package "biomaRt" (Durinck et al., 2005, 2009). This gene list was subsetted to only include genes for which we found at least one transcription factor binding site in the adjacent promoter (see below), and for which we had at least one DNase hypersensitivity peak-call in the adjacent promoter (see below). This resulted in a matrix of 12424 HGNC symbols by 153 expression samples.

Generating the human motif prior for PANDA

We downloaded JASPAR motifs (`http://jaspar.genereg.net/`) (Mathelier et al., 2013) and then used Haystack (Pinello et al., 2014) to scan the entire hg19 genome for these motifs. Of the 205 motifs in the JASPAR database, only 158 had genomic hits that met our significance threshold ($p < 10^{-6}$). We used HOMER (Heinz et al., 2010) (`http://homer.salk.edu/homer/ngs/index.html`) to get the distances of these motif hits to the nearest transcriptional start site (TSS). We used these reported distances to parse the motif hits based on their TSS proximity, keeping only those hits within the "promoter" region of a gene, which we define as $[-750, +250]$ around the TSS. We then filtered for genes to include only those in the RNA-seq data (see above). This information was used to make a prior transcription factor to gene map that we used when running PANDA.

Processing the human DNase hypersensitivity data

Raw DNAse hypersensitivity data (Degner et al., 2012) was downloaded from the Pritchard lab website (`http://eqtl.uchicago.edu/dsQTL_data/RAW_DATA/`, accessed June 2014; also available on GEO: GSE31388). A total of 204 different samples corresponding to 70 different cell lines were available for download. Data were aligned to the hg19/GRCh37 reference genome using Bowtie (Langmead et al., 2009), with options -v 1 and -m 10, allowing for not more than 1 mismatch, and suppressing any alignments for reads having more than 10 reportable alignments. Quality control using Bowtie output identified 16 samples with a high percentage (greater than 80%) of failed reads. We called DNase hypersensitivity peaks using MACS (Zhang et al., 2008). Peaks with significance score of less than 10^{-5} were mapped to the nearest gene using HOMER (Heinz et al., 2010). When the peak fell within the promoter

region of the gene, we assigned a score to that sample-gene pair equal to $-10 \cdot log_{10}p$; otherwise the sample-gene pair was given a default score of zero. We then removed the 16 poor quality samples, as well as samples for which we did not have good quality RNA-seq data (see above), leaving us with 177 samples corresponding to 65 cell lines.

To obtain a promoter-DNase score specific for each cell line, we averaged technical replicates. We then filtered these data to include only genes for which we also had RNA-seq data available and genes that were present in our motif prior (see above). The result was a matrix of DNase-promoter scores that included 12424 genes that had a DNase promoter-peak in at least one sample (at least one row-entry greater than zero). 3488 genes had a promoter-peak in all samples (all row-entries greater than zero).

3.5. Analysis of the human single-sample networks

Calculating gene degree and comparing with DNase hypersensitivity data

When comparing with the DNase information, we calculated gene degree three different ways:

$$\begin{aligned} k_j^{(L)} &= \sum_i p_{ij}^{(L)}; \\ k_j^{(L+e)} &= \sum_i p_{ij}^{(L)} p_i^{(e)}; \\ k_j^{(m+e)} &= \sum_i p_{ij}^{(m)} p_i^{(e)} \end{aligned} \tag{2}$$

where $p_i^{(e)}$ is the "probability" that TF i is expressed, calculated by taking the inverse CDF of the Z-score of the TF's expression compared to the background of its expression in all other samples; $p_{ij}^{(L)}$ is the "probability" of the edge from the LIONESS networks, found by taking the inverse CDF of the predicted edge-weight score (which is in Z-score units); $p_{ij}^{(m)}$ is the "probability" of the edge from the motif data, either 0 or 1 based on whether the motif of TF i was found in the promoter of gene j.

Clustering networks/expression and running GSEA

We created clusters of networks and expression samples by clustering either edge-weights, or gene expression levels, respectively. To do this we performed a hierarchical clustering in which we row-normalized edge-weights (or gene expression) across samples, calculated distance based on the Spearman correlation, and performed a complete-linkage clustering. Spearman correlation was used as a distance metric because the network edges are often not normally distributed. For each clustering performed, we took the primary cut of the dendrogram to make exactly two groups of samples for further analysis.

We performed a LIMMA (Smyth, 2004) analysis to identify either differentially-expressed or differentially-targeted genes between the two groups of samples defined by the hierarchical clustering. A gene's targeting was defined as the sum of all edges pointing to that gene in that sample (the gene's "in-degree"). We also calculated the log fold-change in gene expression/targeting between the two groups of samples and ran a pre-ranked GSEA analysis with 1000 iterations. For differences in gene-targeting in network-defined clusters we observed 68 significant Reactome pathway signatures (Liberzon et al., 2011) with an FDR < 0.1 and a gene set size less than forty. An equivalent clustering/LIMMA/GSEA analysis evaluating differential-expression on expression-defined clusters resulted in no significant Reactome pathways.

Data and software availability

The data sets generated during and/or analyzed during the current study are available in Data S1 (*in silico* data), and from the Gene Expression Omnibus under accession number GSE4987 (yeast data), GSE19480 (human RNA-seq data), and GSE31388 (human DNase data). The human data is also available online at `http://eqtl.uchicago.edu/`. Fully processed and normalized versions of the yeast and human data used in this study are also available from the authors upon request.

4. SUPPLEMENTAL REFERENCES

Albert, R. (2005). Scale-free networks in cell biology, *Journal of cell science* **118**(21): 4947–4957.

Alvarez, M. J., Shen, Y., Giorgi, F. M., Lachmann, A., Ding, B. B., Ye, B. H. and Califano, A. (2016). Functional characterization of somatic mutations in cancer using network-based inference of protein activity, *Nature genetics* **48**(8): 838.

Aytes, A., Mitrofanova, A., Lefebvre, C., Alvarez, M. J., Castillo-Martin, M., Zheng, T., Eastham, J. A., Gopalan, A., Pienta, K. J., Shen, M. M. et al. (2014). Cross-species regulatory network analysis identifies a synergistic interaction between foxm1 and cenpf that drives prostate cancer malignancy, *Cancer cell* **25**(5): 638–651.

Clauset, A., Shalizi, C. R. and Newman, M. E. (2009). Power-law distributions in empirical data, *SIAM review* **51**(4): 661–703.

Degner, J. F., Pai, A. A., Pique-Regi, R., Veyrieras, J.-B., Gaffney, D. J., Pickrell, J. K., De Leon, S., Michelini, K., Lewellen, N., Crawford, G. E. et al. (2012). Dnase I sensitivity qtls are a major determinant of human expression variation, *Nature* **482**(7385): 390–394.

DeLuca, D. S., Levin, J. Z., Sivachenko, A., Fennell, T., Nazaire, M.-D., Williams, C., Reich, M., Winckler, W. and Getz, G. (2012). Rna-seqc: Rna-seq metrics for quality control and process optimization, *Bioinformatics* **28**(11): 1530–1532.

Durinck, S., Moreau, Y., Kasprzyk, A., Davis, S., De Moor, B., Brazma, A. and Huber, W. (2005). Biomart and bioconductor: a powerful link between biological databases and microarray data analysis, *Bioinformatics* **21**(16): 3439–3440.

Durinck, S., Spellman, P. T., Birney, E. and Huber, W. (2009). Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart, *Nature protocols* **4**(8): 1184–1191.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J. and Gardner, T. S. (2007). Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles, *PLoS biology* **5**(1): e8.

Fraenkel Lab (2004). Regulatory Map formatted for spreadsheet import, `http://fraenkel.mit.edu/Harbison/release_v24/txtfiles/`. [Online; accessed Jul 2011].

Glass, K., Huttenhower, C., Quackenbush, J. and Yuan, G.-C. (2013). Passing messages between biological networks to refine predicted interactions, *PloS one* **8**(5): e64832.

Glass, K., Quackenbush, J. and Kepner, J. (2015). High performance computing of gene regulatory networks using a message-passing model, *High Performance Extreme Computing Conference (HPEC), 2015 IEEE* pp. 1–6.

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B., Reynolds, D. B., Yoo, J. et al. (2004). Transcriptional regulatory code of a eukaryotic genome, *Nature* **431**(7004): 99–104.

Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H. and Glass, C. K. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities, *Molecular cell* **38**(4): 576–589.

Langmead, B., Trapnell, C., Pop, M. and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome, *Genome Biol* **10**(3): R25.

Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E. and Storey, J. D. (2014). Package 'sva'.

Liao, Y., Smyth, G. K. and Shi, W. (2013). The subread aligner: fast, accurate and scalable read mapping by seed-and-vote, *Nucleic acids research* p. gkt214.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P. and Mesirov, J. P. (2011). Molecular signatures database (msigdb) 3.0, *Bioinformatics* **27**(12): 1739–1740.

Liu, C., Louhimo, R., Laakso, M., Lehtonen, R. and Hautaniemi, S. (2015). Identification of sample-specific regulations using integrative network level analysis, *BMC cancer* **15**(1): 319.

Liu, X., Wang, Y., Ji, H., Aihara, K. and Chen, L. (2016). Personalized characterization of diseases using sample-specific networks, *Nucleic acids research* **44**(22): e164–e164.

Love, M. I., Huber, W. and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2, *Genome biology* **15**(12): 550.

MacIsaac, K. D., Wang, T., Gordon, D. B., Gifford, D. K., Stormo, G. D. and Fraenkel, E. (2006). An improved map of conserved regulatory sites for saccharomyces cerevisiae, *BMC bioinformatics* **7**(1): 113.

Mathelier, A., Zhao, X., Zhang, A. W., Parcy, F., Worsley-Hunt, R., Arenillas, D. J., Buchman, S., Chen, C.-y., Chou, A., Ienasescu, H. et al. (2013). Jaspar 2014: an extensively expanded and updated open-access database of transcription factor binding profiles, *Nucleic acids research* p. gkt997.

Meyer, P. E., Lafitte, F. and Bontempi, G. (2008). minet: Ar/bioconductor package for inferring large transcriptional networks using mutual information, *BMC bioinformatics* **9**(1): 461.

Pickrell, J. K., Marioni, J. C., Pai, A. A., Degner, J. F., Engelhardt, B. E., Nkadori, E., Veyrieras, J.-B., Stephens, M., Gilad, Y. and Pritchard, J. K. (2010). Understanding mechanisms underlying human gene expression variation with rna sequencing, *Nature* **464**(7289): 768.

Pinello, L., Xu, J., Orkin, S. H. and Yuan, G.-C. (2014). Analysis of chromatin-state plasticity identifies cell-type–specific regulators of h3k27me3 patterns, *Proceedings of the National Academy of Sciences* **111**(3): E344–E353.

Pramila, T., Wu, W., Miles, S., Noble, W. S. and Breeden, L. L. (2006). The forkhead transcription factor hcm1 regulates chromosome segregation genes and fills the s-phase gap in the transcriptional circuitry of the cell cycle, *Genes & development* **20**(16): 2266–2278.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments, *Statistical applications in genetics and molecular biology* **3**(1): 1–25.

Sonawane, A. R., Platig, J., Fagny, M., Chen, C.-Y., Paulson, J. N., Lopes-Ramos, C. M., DeMeo, D. L., Quackenbush, J.,

Glass, K. and Kuijjer, M. L. (2017). Understanding tissue-specific gene regulation, *Cell reports* **21**(4): 1077–1088.

Wang, Y., Sahni, N. and Vidal, M. (2015). Global edgetic rewiring in cancer networks, *Cell systems* **1**(4): 251–253.

Yang, Y., Paquet, A. and Dudoit, S. (2007). marray: Exploratory analysis for two-color spotted microarray data, *Version 1.16* .

Zhang, W., Zeng, T. and Chen, L. (2014). Edgemarker: identifying differentially correlated molecule pairs as edge-biomarkers, *Journal of theoretical biology* **362**: 35–43.

Zhang, Y., Liu, T., Meyer, C. A., Eeckhoute, J., Johnson, D. S., Bernstein, B. E., Nusbaum, C., Myers, R. M., Brown, M., Li, W. et al. (2008). Model-based analysis of chip-seq (macs), *Genome Biol* **9**(9): R137.

5. SUPPLEMENTAL DERIVATIONS

In section 5.1, we show how we can derive networks for individual samples from an aggregate network by applying the LIONESS equation. The LIONESS equation works independently of the network reconstruction algorithm that is used to infer the aggregate network, and can be applied to both linear and non-linear network reconstruction algorithms. To demonstrate this, we show that we can exactly solve the LIONESS equation in its application to Pearson correlation networks, a linear network reconstruction algorithm, in section 5.2. In addition, we show the applicability of the LIONESS equation to the non-linear Mutual Information network reconstruction algorithm in section 5.3. Finally, in 5.4, we show that LIONESS networks do not converge in the limit of a large number of samples.

5.1. Derivation to find regulatory networks for individual samples in a collection

To begin, we assume that the value of a given edge ($e_{ij}^{(\alpha)}$) from a transcription factor (i) to a gene (j) predicted by a network reconstruction algorithm using a collection of samples (α) is the linear combination of the value of that edge across networks specific to each of the input samples ($e_{ij}^{(s)}$), where $w_s^{(\alpha)}$ represents the relative contribution of sample (s):

$$e_{ij}^{(\alpha)} = \sum_{s=1}^{N} w_s^{(\alpha)} e_{ij}^{(s)}, \text{ where } \sum_{s=1}^{N} w_s^{(\alpha)} = 1 \tag{E1}$$

Given this assumption, we can calculate two "aggregate" networks, one using all samples ($e_{ij}^{(\alpha)}$), as described above, and the other using all but one of the samples ($e_{ij}^{(\alpha-q)}$):

$$e_{ij}^{(\alpha-q)} = \sum_{s\neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)}, \text{ where } \sum_{s\neq q}^{N} w_s^{(\alpha-q)} = 1 \tag{E2}$$

Now, subtracting these two "aggregate" network estimates we get:

$$e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)} = \sum_{s=1}^{N} w_s^{(\alpha)} e_{ij}^{(s)} - \sum_{s\neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)} \tag{E3}$$

$$= w_q^{(\alpha)} e_{ij}^{(q)} + \sum_{s\neq q}^{N} w_s^{(\alpha)} e_{ij}^{(s)} - \sum_{s\neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)} \tag{E4}$$

$$= w_q^{(\alpha)} e_{ij}^{(q)} + \sum_{s\neq q}^{N} (w_s^{(\alpha)} - w_s^{(\alpha-q)}) e_{ij}^{(s)} \tag{E5}$$

We can then solve for the network specific to a single sample, $e_{ij}^{(q)}$:

$$e_{ij}^{(q)} = \frac{1}{w_q^{(\alpha)}}\left[e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)} + \sum_{s\neq q}^{N}(w_s^{(\alpha-q)} - w_s^{(\alpha)})e_{ij}^{(s)}\right] \tag{E6}$$

$$= \frac{1}{w_q^{(\alpha)}}\left[e_{ij}^{(\alpha)} - \sum_{s\neq q}^{N} w_s^{(\alpha)} e_{ij}^{(s)}\right] \tag{E7}$$

If we then stipulate that the weights used to estimate $e_{ij}^{(\alpha)}$ are related to the weights used to estimate $e_{ij}^{(\alpha-q)}$ by a constant, $w_s^{(\alpha)} = C w_s^{(\alpha-q)}$ and combine with Equations E1 and E2, we observe that:

$$1 = \sum_{s\neq q}^{N} w_s^{(\alpha-q)} = \sum_{s=1}^{N} w_s^{(\alpha)} = w_q^{(\alpha)} + \sum_{s\neq q}^{N} w_s^{(\alpha)} = w_q^{(\alpha)} + C\sum_{s\neq q}^{N} w_s^{(\alpha-q)} = w_q^{(\alpha)} + C \tag{E8}$$

This then implies that $C = 1 - w_q^{(\alpha)}$ (or, equivalently, $w_q^{(\alpha)} = 1 - w_s^{(\alpha)}/w_s^{(\alpha-q)}$), which we can substitute into Equation E7 in order to calculate the values of the edges in the single-sample network, $e_{ij}^{(q)}$, in terms of the two "aggregate" networks:

$$e_{ij}^{(q)} = \frac{1}{w_q^{(\alpha)}}\left[e_{ij}^{(\alpha)} - C\sum_{s\neq q}^{N} w_s^{(\alpha-q)} e_{ij}^{(s)}\right] \tag{E9}$$

$$= \frac{1}{w_q^{(\alpha)}}\left[e_{ij}^{(\alpha)} - (1 - w_q^{(\alpha)})e_{ij}^{(\alpha-q)}\right] \tag{E10}$$

$$= \frac{1}{w_q^{(\alpha)}}\left[e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)}\right] + e_{ij}^{(\alpha-q)} \tag{E11}$$

Finally, we can simplify this equation by (optionally) assuming that each sample is given equal weight ($w_q^{(\alpha)} = \frac{1}{N}$):

$$e_{ij}^{(q)} = N\left[e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)}\right] + e_{ij}^{(\alpha-q)} \tag{E12}$$

5.2. Application to Pearson correlation

To begin, we remember that the Pearson correlation (r) between two variables, X and Y can be defined as:

$$r = \frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right), \text{ where } \bar{X} = \frac{1}{N}\sum_{i}^{N}X_i \text{ and } S_X = \sqrt{\frac{1}{N-1}\sum_{i}^{N}(X_i-\bar{X})^2} \tag{E13}$$

We can then use the Pearson correlation to calculate two "aggregate" networks, one using all samples (resulting in r), and the other using all samples except for sample q (resulting in r'):

$$r' = \frac{1}{N-2}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right), \text{ where } \bar{X}' = \frac{1}{N-1}\sum_{i\neq q}^{N}X_i \text{ and } S'_X = \sqrt{\frac{1}{N-2}\sum_{i\neq q}^{N}(X_i-\bar{X}')^2} \tag{E14}$$

Now, using the "LIONESS" equation for deriving single samples we see that:

$$e_{ij}^{(q)} = N(e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)}) + e_{ij}^{(\alpha-q)} \tag{E15}$$

$$r_{xy}^{(q)} = N(r_{xy} - r'_{xy}) + r'_{xy} \tag{E16}$$

$$= N\left[\frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \frac{1}{N-2}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right] + \frac{1}{N-2}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right) \tag{E17}$$

$$= \frac{N}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \frac{N-1}{N-2}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right) \tag{E18}$$

$$= \frac{N}{N-1}\left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) + \frac{N}{N-1}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \frac{N-1}{N-2}\sum_{i\neq q}^{N}\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right) \tag{E19}$$

$$= \frac{N}{N-1}\left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) + \sum_{i\neq q}^{N}\left[\left(\frac{N}{N-1}\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(\frac{N-1}{N-2}\right)\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right] \tag{E20}$$

Next, if we average over all possible values of q we find that:

$$\frac{1}{N}\sum_{i}^{N}r_{xy}^{(i)} = \frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) + \sum_{i}^{N}\sum_{i\neq q}^{N}\left[\left(\frac{1}{N-1}\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(\frac{N-1}{N(N-2)}\right)\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right] \tag{E21}$$

We also note that (from equations E13 and E14 above):

$$\bar{X} = \frac{1}{N}\sum_{i}^{N} X_i \tag{E22}$$

$$= \frac{1}{N}X_q + \frac{1}{N}\sum_{i\neq q}^{N} X_i \tag{E23}$$

$$= \frac{1}{N}X_q + \frac{N-1}{N}\bar{X}' \tag{E24}$$

$$= \bar{X}' + \frac{1}{N}(X_q - \bar{X}') \tag{E25}$$

Thus when the difference between X_q and $\bar{X}'$ is much less than the total number of samples being considered (N), $\bar{X}' \to \bar{X}$ and $S'_X \to S_X$. This is most likely for large values of N. In this limit Equation E20 can be simplified as follows:

$$\lim_{N\to\infty} r_{xy}^{(q)} = \lim_{N\to\infty}\left[\frac{N}{N-1}\left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) + \sum_{i\neq q}^{N}\left[\left(\frac{N}{N-1}\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(\frac{N-1}{N-2}\right)\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right]\right] \tag{E26}$$

$$= \left(1\right)\left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) + \sum_{i\neq q}^{N}\left[\left(1\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(1\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right)\right] \tag{E27}$$

$$= \left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) \tag{E28}$$

Similarly Equation E21 can be simplified as follows:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N} r_{xy}^{(i)} = \lim_{N\to\infty}\left[\frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) + \sum_{i}^{N}\sum_{i\neq q}^{N}\left[\left(\frac{1}{N-1}\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(\frac{N-1}{N(N-2)}\right)\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right]\right] \tag{E29}$$

$$= \lim_{N\to\infty}\left[\frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) + \left(\frac{1}{N-1}\right)\sum_{i}^{N}\sum_{i\neq q}^{N}\left[\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(\frac{(N-1)^2}{N(N-2)}\right)\left(\frac{X_i-\bar{X}'}{S'_X}\right)\left(\frac{Y_i-\bar{Y}'}{S'_Y}\right)\right]\right] \tag{E30}$$

$$= \frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) + \left(0\right)\sum_{i}^{N}\sum_{i\neq q}^{N}\left[\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) - \left(1\right)\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right)\right] \tag{E31}$$

$$= \frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_i-\bar{X}}{S_X}\right)\left(\frac{Y_i-\bar{Y}}{S_Y}\right) + \left(0\right)\sum_{i}^{N}\sum_{i\neq q}^{N}\left(0\right) \tag{E32}$$

$$= \frac{1}{N-1}\sum_{i}^{N}\left(\frac{X_q-\bar{X}}{S_X}\right)\left(\frac{Y_q-\bar{Y}}{S_Y}\right) \tag{E33}$$

Which is, reassuringly, equal to r (equation E13). We note that in order to accurately obtain this final equation we must use the fact that, although $1/(N-1) \to 0$ for $N \to \infty$, here $1/(N-1)$ is multiplied by a summation over N finite values. Thus, when we take the limit of large N we have a very large number divided by another very large number of the same order of magnitude, resulting in a finite value.

5.3. Application to mutual information

Mutual information is a way to capture non-linear relationships between two variables (a and b) and can be used to define a network relationship $a \rightarrow b$ when there is a strong dependence (high mutual information) between those variables. In order to estimate the network for a single-sample, the LIONESS equation makes a linear assumption concerning the relationship of an edge $a \rightarrow b$ in one network (represented by the quantity $e_{ab}^{(1)}$), with the edge $a \rightarrow b$ in *another* network (represented by the quantity $= e_{ab}^{(2)}$). In other words, it only cares about the relationship between the quantities $e_{ab}^{(1)}$ and $e_{ab}^{(2)}$); it doesn't stipulate anything about the linear (or non-linear) relationship between a and b within either of those networks or the method that was used to derive the quantities $e_{ab}^{(1)}$ and $e_{ab}^{(2)}$.

Here we show that the linear assumption of LIONESS makes about the relationship of an edge between two networks holds true in the limit of large number of samples (large N), even when the magnitude of that edge $a \rightarrow b$ is calculated using a non-linear measure – in this case mutual information. We do this by explicitly calculating the form of the mutual information for a single-sample based on the LIONESS equation. We then explicitly test the linear assumption made by LIONESS, and show that averaging over all such single-sample networks reduces to the aggregate mutual-information network in the limit of a large number of total samples (large N). The derivation presented in this section follows the same basic approach as the one testing the application of LIONESS to networks derived using Pearson correlation (see section 5.2).

To begin, we remember that for $\mathcal{X}$ discrete states captured by variable x, and $\mathcal{Y}$ discrete states captured by variable y, mutual information can be defined as:

$$I(x,y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{E34}$$

In practice, in order to estimate $p(x)$, $p(y)$ and $p(x,y)$ samples will be binned into the $\mathcal{X}$ and $\mathcal{Y}$ discrete states. To better understand how this works, let us assume we are trying to find the mutual information between two vectors of data: $\vec{a}$ and $\vec{b}$. Based on this we can define functions $x_i = f_x(\vec{a}_i)$ and $y_i = f_y(\vec{b}_i)$ to return the binned "state" of each sample (or data-point) i in the (x,y) bins in $\mathcal{X}$ and $\mathcal{Y}$ space.

Using these functions we can then define a matrix A_{xy} whose entries are defined as total number of data-points binned into each (x,y) state:

$$A_{xy} = \sum_i \delta(x_i, x)\delta(y_i, y) \tag{E35}$$

where $\delta()$ is the Kronecker delta function. Let us also define:

$$X_y = \sum_i \delta(y_i, y) \text{ and } Y_x = \sum_i \delta(x_i, x). \tag{E36}$$

Note that X_y and Y_x are the row and column sums of A_{xy}, respectively, and represent the total number of data points binned into each of the (x) and (y) states.

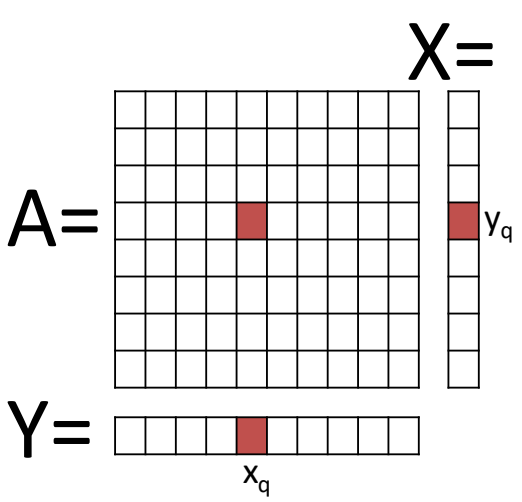
Based on the above definitions, we can re-write the mutual information based on A_{xy}, X_y and Y_x. More specifically, we note that $p(x,y) = {}^{A_{xy}}/_{N}$, $p(x) = {}^{Y_x}/_{N}$ and $p(y) = {}^{X_y}/_{N}$. Subbing these into Equation E34:

$$I(x,y) = \frac{1}{N} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} A_{xy} log\left(\frac{A_{xy}N}{X_y Y_x}\right) \tag{E37}$$

This form of mutual information sums over binned-states (x and y), but one can also equivalently write a form that sums over data-points. For this we define $A_{xy}^{(i)}$ as value of the A_{xy} matrix that contains data-point i:

$$A_{xy}^{(i)} = \sum_j \delta(x_j, x_i)\delta(y_j, y_i) \tag{E38}$$

and similarly $Y_x^{(i)} = \sum_j \delta(x_j, x_i)$ and $X_y^{(i)} = \sum_j \delta(y_j, y_i)$ as the values of Y_x and X_y that include the data-point i.



Definitions:

A = matrix defining where data-points belong in (x,y) space; A_{xy}=Np(x,y)
X = row sums of A, a vector defining where data-points belong in (y)-space; X_y=Np(y)
Y = column sums of A, a vector defining where data-points belong in (x)-space; Y_x=Np(x)
y_q = state in y that contains information for data-point q
x_q = state in x that contains information for data-point q

→ Note removing a single data-point (q) will affect values associated with y_q and x_q in A, X and Y by one.

We also note that in order to calculate mutual information as a sum over data points (instead of bin-states) the contribution of each data-point to the sum should be equivalent to one over the total number of data-points in the same (x, y) bin-state as that data-point, or $A_{xy}^{(i)}$. Based on this information, we can re-write mutual information as:

$$I(x,y) = \frac{1}{N}\sum_i \frac{A_{xy}^{(i)}}{A_{xy}^{(i)}} log\left(\frac{A_{xy}^{(i)} N}{X_y^{(i)} Y_x^{(i)}}\right) = \frac{1}{N}\sum_i log\left(\frac{A_{xy}^{(i)} N}{X_y^{(i)} Y_x^{(i)}}\right) \tag{E39}$$

Next, we define three new additional quantities. First, a matrix summarizing how many data-points would be assigned to each (x, y) bin were we to remove a single data-point (q) from $\vec{a}$ and $\vec{b}$:

$$A'^{(i)}_{xy} = \sum_{j\neq q} \delta(x_j, x_i)\delta(y_j, y_i) = \begin{cases} A_{xy}^{(i)} - 1 & x_i = x_q, y_i = y_q \\ A_{xy}^{(i)} & otherwise \end{cases} \tag{E40}$$

A vector summarizing how many data-points would be assigned to each y state, were we to remove data-point q:

$$X'^{(i)}_{y} = \sum_{j\neq q} \delta(y_j, y_i) = \begin{cases} X_y^{(i)} - 1 & y_i = y_q \\ X_y^{(i)} & otherwise \end{cases} \tag{E41}$$

And a vector summarizing how many data-points would be assigned to each x state, were we to remove data-point q:

$$Y'^{(i)}_{x} = \sum_{j\neq q} \delta(x_j, x_i) = \begin{cases} Y_x^{(i)} - 1 & x_i = x_q \\ Y_x^{(i)} & otherwise \end{cases} \tag{E42}$$

The only stipulation in defining these quantities is that the same x and y states that were used to define A_{xy}, X_y and Y_x are also used to define ${A'}_{xy}$, ${X'}_y$ and ${Y'}_x$. Based on these quantities, we can then define the mutual information between two vectors $\vec{a}$ and $\vec{b}$ after removing a single data-point from those vectors (q, previously been assigned to bin (x_q, y_q)).

$$I'(x,y) = \frac{1}{N-1}\sum_{i\neq q} log\left(\frac{{A'}_{xy}^{(i)}N}{{X'}_y^{(i)}{Y'}_x^{(i)}}\right) \tag{E43}$$

Now, reiterating the LIONESS equation we see:

$$\begin{aligned}
I(q) &= N(I(x,y) - I'(x,y)) + I'(x,y) && \text{(E44)}\\
&= NI(x,y) - (N-1)I'(x,y) && \text{(E45)}\\
&= \sum_i log\left(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\right) - \sum_{i\neq q} log\left(\frac{{A'}_{xy}^{(i)}N}{{X'}_y^{(i)}{Y'}_x^{(i)}}\right) && \text{(E46)}\\
&= log\left(\frac{A_{xy}^{(q)}N}{X_y^{(q)}Y_x^{(q)}}\right) + \sum_{i\neq q} log\left(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\right) - \sum_{i\neq q} log\left(\frac{{A'}_{xy}^{(i)}N}{{X'}_y^{(i)}{Y'}_x^{(i)}}\right) && \text{(E47)}\\
&= log\left(\frac{A_{xy}^{(q)}N}{X_y^{(q)}Y_x^{(q)}}\right) + \sum_{i\neq q} log\left(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\right) - log\left(\frac{{A'}_{xy}^{(i)}N}{{X'}_y^{(i)}{Y'}_x^{(i)}}\right) && \text{(E48)}\\
&= log\left(\frac{A_{xy}^{(q)}N}{X_y^{(q)}Y_x^{(q)}}\right) + \sum_{i\neq q} log\left(\frac{N}{N-1}\frac{A_{xy}^{(i)}}{{A'}_{xy}^{(i)}}\frac{{X'}_y^{(i)}}{X_y^{(i)}}\frac{{Y'}_x^{(i)}}{Y_x^{(i)}}\right) && \text{(E49)}
\end{aligned}$$

We note that, from above, we know the relationship between $A_{xy}^{(i)}$ and ${A'}_{xy}^{(i)}$ (see Equation E40), the relationship between $X_y^{(i)}$ and ${X'}_y^{(i)}$(see Equation E41), as well as the relationship between $Y_x^{(i)}$ and ${Y'}_x^{(i)}$ (see Equation E42). Based on this information, let us divide the data-points in the right-hand sum of Equation E49 into four parts: (1) P_1: those data-points that are not in either the x_q or y_q bins, (2) P_2: those data-points in the x_q bin but not the y_q bin, (3) P_3: those data-points in the y_q bin but not the x_q bin, and (4) P_4: those data-points that are with q in the (x_q, y_q) bin:

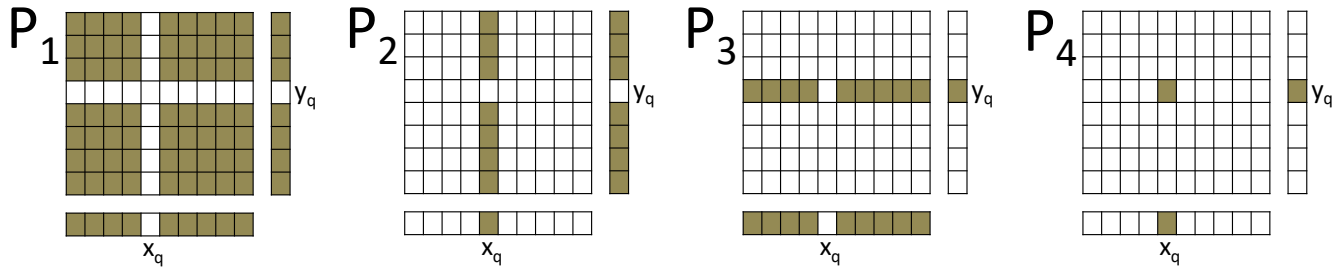
$$I(q) = log\left(\frac{A_{xy}^{(q)}N}{X_y^{(q)}Y_x^{(q)}}\right) + P_1 + P_2 + P_3 + P_4. \tag{E50}$$

Let's start with the simplest scenario: the data-points that are not in either the x_q or the y_q bin (P_1). We can calculate the total number of data-points in this category as:

$$N_1 = N - X_y^{(q)} - Y_x^{(q)} + A_{xy}^{(q)} \tag{E51}$$

Within the category the following also holds for these data-points: $A_{xy}^{(i)} = {A'}_{xy}^{(i)}$, $X_y^{(i)} = {X'}_y^{(i)}$, and $Y_x^{(i)} = {Y'}_x^{(i)}$. This is true for all data-points i that are not in either the x_q or y_q bins. Based on this, the P_1 reduces to:

$$P_1 = N_1 log\left(\frac{N}{N-1}\right) \tag{E52}$$



The second scenario (P_2) is a data-point that is in the x_q bin, but not the y_q bin. We can calculate the number of data-points in this category as:

$$N_2 = Y_x^{(q)} - A_{xy}^{(q)} \tag{E53}$$

Within the category the following also holds for these data-points: $A_{xy}^{(i)} = A'^{(i)}_{xy}$, $X_y^{(i)} = X'^{(i)}_y$, and $Y'^{(i)}_x = Y'^{(q)}_x = Y_x^{(q)} - 1$. In this case, the right hand sum in Equation E49 reduces to:

$$P_2 = N_2 log\left(\frac{N}{N-1}\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) = N_2\left[log\left(\frac{N}{N-1}\right) + log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right)\right] \tag{E54}$$

Similarly, the third scenario (P_3) is a data-point that is in the y_q bin, but not in the x_q bin. We can calculate the number of data-points in this category as:

$$N_3 = X_y^{(q)} - A_{xy}^{(q)} \tag{E55}$$

Within the category the following also holds for these data-points: $A_{xy}^{(i)} = A'^{(i)}_{xy}$, $X'^{(i)}_y = X'^{(q)}_y = X_y^{(q)} - 1$, and $Y_x^{(i)} = Y'^{(i)}_x$. In this case, the right hand sum in Equation E49 reduces to:

$$P_3 = N_3 log\left(\frac{N}{N-1}\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) = N_3\left[log\left(\frac{N}{N-1}\right) + log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right)\right] \tag{E56}$$

Finally, the last scenario is a data-point that is in the same (x_q, y_q) bin as q (but is not q itself). The number of data-points meeting this criteria is precisely $A_{xy}^{(q)} - 1$. The following also holds for these data-points: $A'^{(i)}_{xy} = A'^{(q)}_{xy} = A_{xy}^{(q)} - 1$, $X'^{(i)}_y = X'^{(q)}_y = X_y^{(q)} - 1$, and $Y'^{(i)}_x = Y'^{(q)}_x = Y_x^{(q)} - 1$. In this case, the right hand sum in Equation E49 reduces to:

$$P_4 = (A_{xy}^{(q)} - 1) log\left(\frac{N}{N-1}\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\frac{X_y^{(q)} - 1}{X_y^{(q)}}\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \tag{E57}$$

$$= (A_{xy}^{(q)} - 1)\left[log\left(\frac{N}{N-1}\right) + log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) + log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) + log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right)\right] \tag{E58}$$

Now let's aggregate together all the components of the right-hand sum:

$$P_1 + P_2 + P_3 + P_4 = (N - X_y^{(q)} - Y_x^{(q)} + A_{xy}^{(q)}) log\left(\frac{N}{N-1}\right) \tag{E59}$$

$$+ (Y_x^{(q)} - A_{xy}^{(q)}) \left[log\left(\frac{N}{N-1}\right) + log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \right] \tag{E60}$$

$$+ (X_y^{(q)} - A_{xy}^{(q)}) \left[log\left(\frac{N}{N-1}\right) + log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) \right] \tag{E61}$$

$$+ (A_{xy}^{(q)} - 1) \left[log\left(\frac{N}{N-1}\right) + log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) + log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) + log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \right] \tag{E62}$$

Re-grouping we then find that:

$$P_1 + P_2 + P_3 + P_4 = (N - X_y^{(q)} - Y_x^{(q)} + A_{xy}^{(q)} + Y_x^{(q)} - A_{xy}^{(q)} + X_y^{(q)} - A_{xy}^{(q)} + A_{xy}^{(q)} - 1) log\left(\frac{N}{N-1}\right) \tag{E63}$$

$$+ (Y_x^{(q)} - A_{xy}^{(q)} + A_{xy}^{(q)} - 1) log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \tag{E64}$$

$$+ (X_y^{(q)} - A_{xy}^{(q)} + A_{xy}^{(q)} - 1) log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) \tag{E65}$$

$$+ (A_{xy}^{(q)} - 1) log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) \tag{E66}$$

Or simplified:

$$P_1 + P_2 + P_3 + P_4 = (N-1) log\left(\frac{N}{N-1}\right) + (Y_x^{(q)} - 1) log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \tag{E67}$$

$$+ (X_y^{(q)} - 1) log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) + (A_{xy}^{(q)} - 1) log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) \tag{E68}$$

So then, remembering Equation E49 we observe that:

$$I(q) = log\left(\frac{A_{xy}^{(q)} N}{X_y^{(q)} Y_x^{(q)}}\right) + (N-1) log\left(\frac{N}{N-1}\right) + (Y_x^{(q)} - 1) log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) \tag{E69}$$

$$+ (X_y^{(q)} - 1) log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) + (A_{xy}^{(q)} - 1) log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) \tag{E70}$$

If we then average over all possible values of q, we find:

$$\frac{1}{N}\sum_{i}^{N} I(i) = \frac{1}{N}\sum_{i}^{N}\Bigg[log\bigg(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\bigg) + (N-1)log\bigg(\frac{N}{N-1}\bigg) + (Y_x^{(q)}-1)log\bigg(\frac{Y_x^{(q)}-1}{Y_x^{(q)}}\bigg) \tag{E71}$$

$$+ (X_y^{(q)}-1)log\bigg(\frac{X_y^{(q)}-1}{X_y^{(q)}}\bigg) + (A_{xy}^{(q)}-1)log\bigg(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)}-1}\bigg)\Bigg] \tag{E72}$$

We are most interested in the behavior of this quantity in the limit of a large number of samples ($N \to \infty$). In this case:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N} I(i) = \lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N}\Bigg[log\bigg(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\bigg) + (N-1)log\bigg(\frac{N}{N-1}\bigg) + (Y_x^{(q)}-1)log\bigg(\frac{Y_x^{(q)}-1}{Y_x^{(q)}}\bigg) \tag{E73}$$

$$+ (X_y^{(q)}-1)log\bigg(\frac{X_y^{(q)}-1}{X_y^{(q)}}\bigg) + (A_{xy}^{(q)}-1)log\bigg(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)}-1}\bigg)\Bigg] \tag{E74}$$

We note that the latter terms in the sum are constants (not dependent on i) and thus can be brought out of the sum. In addition, based on the rules of limits, we can divide the above equation into five separate limits:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N} I(i) = \lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N} log\bigg(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\bigg) + \lim_{N\to\infty}(N-1)log\bigg(\frac{N}{N-1}\bigg) + \lim_{N\to\infty}(Y_x^{(q)}-1)log\bigg(\frac{Y_x^{(q)}-1}{Y_x^{(q)}}\bigg) \tag{E75}$$

$$+ \lim_{N\to\infty}(X_y^{(q)}-1)log\bigg(\frac{X_y^{(q)}-1}{X_y^{(q)}}\bigg) + \lim_{N\to\infty}(A_{xy}^{(q)}-1)log\bigg(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)}-1}\bigg) \tag{E76}$$

Solving each of these limits we first note that, based on Equation E39:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i}^{N} log\bigg(\frac{A_{xy}^{(i)}N}{X_y^{(i)}Y_x^{(i)}}\bigg) = \lim_{N\to\infty} I(x,y) = I(x,y) \tag{E77}$$

Next we observe that the remaining limits take two main forms. The limit applied to the N and the $A_{xy}^{(q)}$ portions of the equation are of the form:

$$\lim_{z\to c}(z-1)log\bigg(\frac{z}{z-1}\bigg) = \lim_{z\to c}\frac{log(z)-log(z-1)}{(z-1)^{-1}} \tag{E78}$$

Then, applying l'Hopital's rule (and re-arranging, and applying l'Hopital again, etc):

$$\lim_{z\to c}\frac{log(z)-log(z-1)}{(z-1)^{-1}}=\lim_{z\to c}\frac{\frac{1}{z}-\frac{1}{z-1}}{-1(z-1)^{-2}} \tag{E79}$$

$$=\lim_{z\to c}(z-1)^2\left(\frac{1}{z-1}-\frac{1}{z}\right) \tag{E80}$$

$$=\lim_{z\to c}(z-1)^2\frac{z-(z-1)}{z^2-z} \tag{E81}$$

$$=\lim_{z\to c}\frac{(z-1)^2}{z^2-z} \tag{E82}$$

$$=\lim_{z\to c}\frac{2(z-1)}{2z-1} \tag{E83}$$

$$=\lim_{z\to c}\frac{2z-2}{2z-1} \tag{E84}$$

Similarly, the limit applied to the $Y_x^{(q)}$ and $X_y^{(q)}$ portions of the equation takes the form:

$$\lim_{z\to c}(z-1)log\left(\frac{z-1}{z}\right)=\lim_{z\to c}\frac{log(z-1)-log(z)}{(z-1)^{-1}} \tag{E85}$$

Then, applying l'Hopital's rule (and re-arranging, and applying l'Hopital again, etc):

$$\lim_{z\to c}(z-1)log\left(\frac{z-1}{z}\right)=\lim_{z\to c}\frac{log(z-1)-log(z)}{(z-1)^{-1}} \tag{E86}$$

$$=\lim_{z\to c}\frac{\frac{1}{z-1}-\frac{1}{z}}{-1(z-1)^{-2}} \tag{E87}$$

$$=\lim_{z\to c}(z-1)^2\left(\frac{1}{z}-\frac{1}{z-1}\right) \tag{E88}$$

$$=\lim_{z\to c}(z-1)^2\frac{z-1-z}{z(z-1)} \tag{E89}$$

$$=\lim_{z\to c}\frac{-(z-1)^2}{z^2-z} \tag{E90}$$

$$=\lim_{z\to c}\frac{-2(z-1)}{2z-1} \tag{E91}$$

$$=\lim_{z\to c}\frac{-(2z-2)}{2z-1} \tag{E92}$$

This implies that for $z=N$ and $c=\infty$:

$$\lim_{N\to\infty}(N-1)log\left(\frac{N}{N-1}\right)=1 \tag{E93}$$

Although less intuitive, for $N \to \infty$, all relevant values of $A_{xy}^{(q)}$ will also approach infinity. This can be seen by revisiting the definition of $A_{xy} = N * p(x,y)$. We can then note that $p(x,y)$ is bounded between 0 and 1. This implies that for any value of $0 < p(x,y) \leq 1$, as $N \to \infty$, $A_{xy} \to \infty$. On the other hand, if $p(x,y) = 0$, then A_{xy} is also equal to zero. In this latter case, removing a q data-point will *never* effect the bin ($A_{xy}^{(i \neq q)} = A'^{(i \neq q)}_{xy}$), since that element of A_{xy} contains no data-points; in other words, it will never exist in the limit in question. Based on this we can say that for $N \to \infty$:

$$\lim_{A_{xy}^{(q)} \to \infty} (A_{xy}^{(q)} - 1) log\left(\frac{A_{xy}^{(q)}}{A_{xy}^{(q)} - 1}\right) = 1 \tag{E94}$$

Based on a similar argument as was made for the limit applied to $A_{xy}^{(q)}$ we also see that for $N \to \infty$ the values of $X_y^{(q)}$ and $Y_x^{(q)}$ will also increasingly grow and approach infinity implying:

$$\lim_{X_y^{(q)} \to \infty} (X_y^{(q)} - 1) log\left(\frac{X_y^{(q)} - 1}{X_y^{(q)}}\right) = -1, \text{ and } \lim_{Y_x^{(q)} \to \infty} (Y_x^{(q)} - 1) log\left(\frac{Y_x^{(q)} - 1}{Y_x^{(q)}}\right) = -1 \tag{E95}$$

Finally, subbing the evaluations of these limits back into Equation E76 we find that:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i}^{N} I(i) = I(x,y) + 1 - 1 - 1 + 1 = I(x,y) \tag{E96}$$

Which is, reassuringly, precisely the linear assumption made by the LIONESS equation. Thus, we note that for large values of N Equation E76 reduces to the original data-point version of mutual information defined in Equation E39.

We note that this evaluation relies on the *theoretical* limit of large N. However, as is often the case in mathematical biology, calculating MI in practice relies on real-world, finite data; we test this type of application explicitly in the main text of the manuscript. In the context of large, but finite, data one could postulate that the above limits might not always hold. This is true. However, we emphasize that correctly binning data is a recognized issue for correctly calculating *aggregate* MI-networks and is independent of whether one would then want to apply LIONESS to these network-models. The importance of binning data in a way that prevents the existence of bins with only a small number of data-points is well-recognized by scientists that use MI to estimate network models. Indeed, many computational approaches that compute MI bin data into only 3-4 bins to ensure that enough data-points are in each bin.

5.4. Convergence properties of the LIONESS equation

Here we explore the behavior of the LIONESS equation in the limit of a large number of samples. For simplicity, let us define a new variable, $D = e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)}$ and reiterate the LIONESS equation (Equation E12) in terms of D:

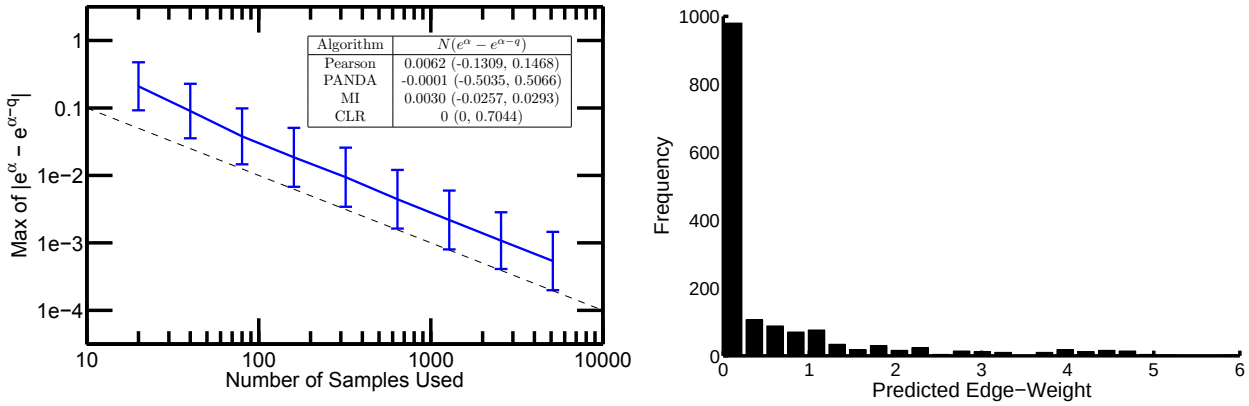
$$e_{ij}^{(q)} = N(e_{ij}^{(\alpha)} - e_{ij}^{(\alpha-q)}) + e_{ij}^{(\alpha-q)} \tag{E97}$$

$$= ND + e_{ij}^{(\alpha-q)} \tag{E98}$$

As the number of samples (N) approaches infinity, we note that D approaches zero. In this limit the LIONESS equation can be thought of as a very large number times a very small number, plus a constant:

$$e_{ij}^{(q)} = [N \to \infty][D \to 0] + [e_{ij}^{(\alpha-q)}]. \tag{E99}$$

This implies that, in the limit of large N, any sort of potential convergence behavior of the LIONESS equation is dependent on the relationship between D and N. We have systematically investigated the consequence of this relationship in the context of the analysis presented in the main text and present the results in the figure below.



Left: A plot of the maximum absolute value of D across all edges when applying LIONESS to Pearson aggregate networks. Values were calculated for varying numbers of samples (N) and with different sets of samples. The mean plus/minus one standard deviation for these sets of samples is shown. The inset table shows the median and interquartile-range for the ND values associated with the 10000 networks estimated using the four different reconstruction approaches explored in the main text. Right: The distribution of the aggregate network ($e^{(\alpha)}$) edge-weights predicted by applying CLR to the *in silico* data. The values predicted by CLR are normalized Z-scores, but are always non-negative.

To begin, we used the Pearson correlation to estimate aggregate networks using the *in silico* data. We determined D by subtracting the values estimated for edges in an aggregate network reconstructed using N samples with the values estimated for edges in an aggregate network based on those same N samples minus one. We then took the absolute value of these differences and determined the maximum across all edges in the network. We repeated this 1000 times using different sets of N samples, and in the figure above, we plot the mean, plus and minus the standard deviation, of the maximum-difference values across the 1000 sample-sets, and for varying values of N (blue line). For reference a line at $y = x/N$ is also plotted (dashed black line). We see that the difference between $e^{(\alpha)}$ and $e^{(\alpha-q)}$ consistently reaches values greater than $1/N$. This indicates that ND does not converge to zero and the LIONESS equation does not converge to $e_{ij}^{(\alpha-q)}$.

In the main text we apply LIONESS to four different network reconstruction approaches: Pearson Correlation, PANDA, Mutual Information (MI) and Context Likelihood of Relatedness (CLR). For each of these approaches, we have determined the median and interquartile range for all values of D used to calculate the $N = 10000$ single-sample networks based on the *in silico* expression data-set. We report these values (times 10000) in the subset table in the figure above. Unsurprisingly, the median value for ND for all the methods is very close to zero. In addition, all approaches have an ND interquartile range that is non-zero, and typically varies between 0.01 and 1.

In examining these values, we feel is it important to point out that each of these four reconstruction approaches estimates scores for network edges in a different way. As a consequence, the distribution of predicted edge weights in the associated aggregate models is different for each approach; these differences are reflected in D. For example, when reconstructing a network, CLR performs a joint Z-score normalization on a Mutual Information Matrix (MIM). As a part of this normalization step, it removes any edge with a MI value below either its row or column's mean in the MIM, assigning it a weight of zero. This step "removes" 51.4% of edges in the CLR *in silico* aggregate network (assigning them a predicted weight of zero). This can be observed by examining the distribution of edge weights predicted for the CLR aggregate network (see figure above). Consistently, 51.4% of D values based on CLR aggregate networks are also zero, as evidenced by the quartiles shown in the inset table in the figure above.