Network Inference

Research Plan

Author: Daniel Morgan

Supervisor: Erik L.L. Sonnhammer

Co-Supervisor: Torbjorn Nordling

Introduction

It is my hope that understanding the biological relationships uncovered among genes, peptides or protein complexes will extend towards the creation of models of complex, resource dependent natural systems, to more efficiently test and effectively treat those biological systems. Proper characterization is impossible without knowledge of the surrounding environment, as it has long been known that removing singular components renders them disparate from their true nature, *i.e.* conclusions drawn from testing cells grown in isolated culture cannot be directly translated to patients.

One must then choose a perspective: A broad focus allows for investigation of obvious signals across an entire system, accounting for everything en masse and leaving results ambiguous, or to constrict interest to a specific subset from which to draw conclusions and base future research, in the hope of connecting the individual focus areas for a more meaningful observation.

I propose my studies focus on the formation of a systems perspective to biological experimentation, to showcase the need for higher resolution techniques of observation, and thereby the power and potential for understanding biological systems. The generic gene regulatory network (GRN) aims to capture and model all real interactions within a gene space, and no others. It is therefore crucial to evaluate the accuracy of GRN inference methods in the context of network and data properties to avoid false positive and negatives. Yet independent control of network topology, system dynamics, experimental design, data properties and noise is limited in the current network inference space; while packages with simulating, modeling, and analyzing capabilities are repeatedly introduced.

I aim to model the system space using proven control systems engineering techniques to sift through the litany of noise using methods of statistical probability to identify the signal. This theme will carry across several interconnected projects,beginning with the continued development of the flexible inference pipeline GeneSPIDER, which highlights the very necessity for such enhanced investigative abilities for its discussion on data property and their current disregard in biological research. Independent control is paramount to simulating biologically representative data, to determining which inference method best fits a given condition, and to predicting expected performance. I plan to expand the methods and techniques offered in GeneSPIDER to infer ever more accurate networks from the disparate datasets being opened for public use; If properly maintained and updated, the package would fulfill its design to become a catchall for adding knowledge of the biological systems to the scientific community in a secondary manner, consuming little time, money or resources. GeneSPIDER's ability to independently control network and data properties in simulation, together with its analytic and quality control features, enables more informative GRN inference performance than was previously possible, and its continued development will be a research-enabling force as well as a homogenizing force in the field.

Building upon the analysis of properties of experimentally characterized data, we are left with the biological sample itself, expensive, time consuming and highly variable. Biological samples capture only a single state of an otherwise dynamical system, and are thus in need of expanded coverage. The continued develop of our network inference toolkit could enlighten scientists to standards their experiments should strive for, and thus open network inference to more datasets, thereby increasing scientific knowledge.

In order to properly infer networks from data, we should understand the exact nature and character of the data. Simulation of data with properties reflective of biological data properties, embedded with tuning parameters, allows for real world variability to be modeled and thus understood. Modeling and evaluating a predicted outcome using an expanded bootstrap paradigm, thereby building statistical significance and biological meaningfulness, could guide the scientists questions, *i.e. in silico* conclusions guiding *in vitro* and *in vivo* hypothesis, and even raise the likelihood of experimental success.

We then implement this method of gathering support for our inference method to infer a network composed of a gene set known to contain cancer precursor genes, with a hub focused on the Myc pathway, itself known to be involved in several human cancer types. The Myc dataset contains 40 genes, perturbed individually in biological triplicate and technical duplicate, for a matrix of 80x135, and again in pairs to further challenge our network inference by means of deducing additive relationship versus some sort of semi-additive or semi-reductive interplay in resultant gene expression measurements. Comparison of networks inferred by single gene perturbations and double perturbations allows for dynamic scoring of the inference algorithm and model formation.

I hope to further develop this *in silico* technique to cut time and monetary investment for biological investigation, while maintaining statistical power. Our network inference methodology needs a benchmark to find out optimal scenarios for choosing one method over another. However, the task of choosing a suitable method becomes more challenging as the number of algorithms and methods grow. Creating wrappers for incorporation of various inference methods to be benchmarked within a unified environment such as GeneSPIDER brings homogeneity and comparability to the results, for the purpose of appraising the utility of the GeneSPIDER package, as well as improvements. If a thorough and sufficiently narrow and goal oriented benchmark is performed it can serve as a guide to the network inference community on what further improvements need be conducted to optimize network inference algorithms. It can also serve as a guide as to what methods are suitable and under what circumstances it can be used when an experimentalist would wish to investigate their own data.

1 *GeneSPIDER: A Continuing Focus on Data Properties*

The goal of GRN inference is to understand how genes influence each other in terms of their expression, i.e. to unravel the transcriptional regulation influences[1]. The primary objective in network inference is to obtain a network where each link corresponds to a real influence of importance in the biological system, i.e. to avoid false positives. The secondary objective is to have a link for each real influence of importance in the biological system, i.e. to avoid false negatives. To this end in GRN inference the following steps are typically carried out: (a) selection of the set of genes of interest, (b) design of perturbation experiments, (c) measurements of expression changes when performing the designed experiments *in vivo* or *in vitro*, (d) inference of a network model from the recorded data, (e) validation of the network model and possibly additional experiments if needed, (f) analysis of the inferred network. Each link in the inferred network represents a regulatory influence between two genes.

1.1 Future Direction

I aim to expand GeneSPIDER's native capabilities toward accurate network inference from the varying datasets publicly available; A properly maintained and updated package would become a segue for biological systems knowledge of the biological systems to the scientific community in a secondary manner. GeneSPIDER's ability to control network and data properties in simulation independently, combined with its analytic and quality control features, enables informative GRN inference where previously only sub-optimally possible, and its continued development will be a research-expanding resource as well as a homogenizing force in the field.

2 *NestBoot: Extending the Bootstrap Procedure*

We are interested not in individual response but in overall system regulation; not necessarily in direct interactions but in the response any individual gene's tuning elicits as part of the multitude of shared environmental resources. We recover genetic interactions by weighting the effect

of single perturbations across a dataset comprised of no more than 50 genes (currently computationally tractable). These biomolecular relationships are modeled using a linear ODE model and the GeneSPIDER MATLAB package to infer the regulatory network linking their cause and effects *en masse*. We seek to differentiate between the merely possible and highly probable elicited responses Parameter estimation schemes select for relevant variables with probabilities tending to one, while selecting irrelevant variables with positive probabilities. Simulating variously conditioned datasets using a bootstrap paradigm allowed the estimation of network inference accuracy through the use of L1-regularization methods and the building of statistical power. We reveal highly observed, relevant interactions for inclusion in the model across the disparate datasets, as well as a wash of random irrelevant interactions to disregard. It is critical to note that common GRN inference methods do not provide information of how accurate an inferred network is. We address this by analyzing key data properties laid out in [2], extending the bootstrap method to an iterated overlap analysis, and applying it to relevant public datasets.

It is important to introduce methods designed for evaluating dataset quality in order to guide inference, fitting, regularization, error balancing and optimization methods in the light of a recent publication detailing the shortcomings of regularization methods when operating over sufficiently informative data[2]. The present case study illustrates how these methods perform poorly for ill-conditioned data matrices, failing as a function of their sensitivity to input error[2]; specifically L1-regularization methods, *i.e.*, Elastic Net, CLS, LSCO, RNI, fail to reconstruct gold-standard (gs) networks even when the data are informative enough for network inference by other methods.

2.1 Goals

It is important to create a reliable method of deducing GRN because it is only through such a comprehensive understanding of disease that therapy beyond serendipitous discovery is likely to occur.

We utilize the *Bacillus subtilis* dataset obtained from R. Bonneau *et al.*[3], composed of some 28 genes each singly perturbed against the others. We will evaluate the capacity of bootstrap sampling to increase statistical support for an inferred network. The dataset comes a gold standard for comparison and evaluation of the inferred network. However, the inferred network also accounts for some thousands of other genes investigated by the team and is thus not directly comparable. We thus propose to implement the *Schur* complement of the gold standard to estimate MCC, our preferred metric of choice, a measure of correlation between observed and predicted outcomes, where

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{1}$$

Taking advantage of several gold standard datasets, we will test several inference methods, characterizing their strengths and weaknesses in order to provide a guide of best practices when there is no gold standard for validation. We seek to differentiate between what is highly probable and simply possible when reconstructing the interactions of gene products, when considering which of any dataset's perturbation effects are true. The methods of differentiation are contingent on a model capable of incorporating incorporating and accounting for random error as well as read or machine error, inherent to any scientific investigation and thus not exempt from computational modeling. The very nature of digitizing biology begets the need for error estimation and correction; estimation because if error could be accurately measured it could be measured against or corrected for, which by the limitations of our current "digitizing" methods we cannot. Here we use several systems approaches that attempt to accurately account for variation in biological measurements, as well as for the variability inherent to the systems. By exploring possible systemic and measurement variation in a sampling with replacement subspace, we distinguish our model and allow for more accurately reproducing the gold standard

networks, and thus produce networks of a higher statistical meaningfulness, in the hopes of informing biological hypothesis generation.

2.2 Methods

In order to control for certain variables, as well as to make the search space tractable, we assume the steady state for our model as follows,

$$\boldsymbol{Y} = -\boldsymbol{A}^{-1}\boldsymbol{P} + \boldsymbol{A}^{-1}\boldsymbol{F} + \boldsymbol{E} \tag{2}$$

because the expression data we base our inference was measured at one instant and thus is not observed to vary in time. Here, Y is the observed response gene expression matrix, and is equal to the final matrix from the combination of the perturbed network (A and P) in addition to the output (E) and input (F) error. This limit can surely be considered a relic of our time, our ability to estimate error within observed datasets, and thus the current case study considers these error terms to be empty matrices. Models that account for variation at multiple time points are of keen interest for continued research, will not be explored here.

2.2.1 Iterative Sampling with Replacement

Bootstrapping, specifically case resampling or resampling with replacement, is used here to estimate the accuracy of our use of L1-regularization methods to do network inference. Taking into account how each individual perturbation influences the expression of every other non-perturbed gene, we attempt to infer the as of yet unknown network of gene regulatory association by instead calculating en masse, upon what we do know, the samples and their replicates. We estimate variability through the use of iterative resampling of the samples' replicates, and in so doing, reduce the sampling bias/ increase the accuracy of the variance estimation, we encounter with our large datasets. This process of iterative error estimation gives a more representative picture of the variation within the data, and thus expands the possibility for accounting for the true nature of the underlying network. However, limitations are well known[4].

2.2.2 Least Squares Cut-Off

The Least-Squares method corrects for input error on the Y expression matrix with regard to the a one dimensional correction, along X.[4]. Here we seek to find the minimum frobenius norm of A along X, which can be rewritten in the linear form of Eq. 4

$$\{\hat{A}_{ls}, \Delta_{ls}\} := \arg\min_{X,\Delta A} ||\Delta A||_F \tag{3}$$

$$\hat{A}_{OLS} = (X^T X)^{-1} X^T y \tag{4}$$

The rationale behind this approximation method is to correct the right-hand side y as little as possible in the Frobenius norm sense, so that the corrected system of equations has an exact solution. When X is full rank, the LS solution is similar to that of the TLS.

2.2.3 Total Least Squares Cut-Off

The Total Least-Squares approach looks for the minimal correction when both input and output data have error to be accounted. The Total Least-Squares (TLS) approximation is a maximum likelihood estimator obtained by correcting both X and Y coordinates. The TLS approach looks for the minimal correction between matrices when both datasets are allowed to be perturbed. Augmenting the two separate matrices enables the simultaneous singular value decomposition,

which yields a real unitary matrix. One then takes the right eigenvector and normalizes over its terminal eigenvalue to complete the optimization.

$$\text{svd}[X\ Y] = USV^T \tag{5a}$$

$$\hat{A}_{TLS} = -\frac{VXY}{VYY} \tag{5b}$$

where XY = right most column of V, YY = bottom right most cell. A constrained weighted TLS algorithm could regularize and offers convergence in the case of ill-conditioned datasets.[5]. This project is beyond the scope of my current research plan, but is of certain interest to my further research and the groups in general.

2.2.4 Constrained or Structured Total Least Norm

The least and total least squares often stumble when trying to properly account for network characteristics such as special structure or disparity between error matrix and A matrix elements when optimizing their solutions. The problem seems to be on their reliance of SVD, which can change network topology, morphing zeros into non-zeros. STLN addresses this by allowing error minimization with any L norm, while also preserving structure in A and error matrices[6]. This is really the ideal way to account for errors in a systems space, only weighting those elements of importance with any potential error, leaving zero elements no room to become non-zero, thus preserving structure throughout.

2.2.5 LASSO (Least Absolute Shrinkage and Selection Operator) Cut-Off

The LASSO estimation selects relevant variables with probability tending to one (the signal) while selecting irrelevant variables merely with positive probability (the noise). LASSO is utilized here to distinguish between highly correlated predictors, gene expression knock down (perturbation), on the expression of other genes known to have some relation to the perturbed gene (comprising a related geneset). We implement it to fault deviation from our regression model describing the linear system. Sampling occurs in each bootstrap (procedure detailed next), upon which the following lasso estimate is computed:

$$\min \frac{1}{2m} \left\| \boldsymbol{A}_i \boldsymbol{Y}^T + \boldsymbol{P}_i^T \right\|_F^2 + \zeta \left\| \boldsymbol{A} \right\|_1 \tag{6}$$

The individual and overall support as well as the sparsity are then computed. The Bolasso algorithm[7] utilizes the advantages of the bootstrap procedure with the LASSO algorithm and samples from the data set uniformly to construct inferred networks.

2.2.6 Modeling random network overlap

We want to model the overlap of random networks to estimate the probability of a particular output at a particular support level for a bootstrap scheme using the estimated parameters from our real data set. Each link for a random network, l, is drawn with a specific probability, P, where p is estimated as the average bootstrap support, *i.e.* the average fraction of links for the bootstrapped networks. We can now model each link support as following a binomial distribution, $X \sim \text{Binomial(n, p)}$, such that

$$\mathsf{P}(X = s) = \binom{b}{s} p^s (1-p)^{b-s} \tag{7}$$

with s = the number of times a link has existed over the number of bootstraps b, and $\frac{s}{b} \cdot 100$ being the support for the link.

However this does not describe how the distribution for a specific support behaves. To achieve this, we first have to describe the distribution when $p_L = \mathsf{P}(X = s)$ for some specific s, that is, the probability of a link for a specific support $\mathsf{P}(X = s)$. This would suggest that we could model the distribution of frequencies of links for a specific support as another binomial,

$$\mathsf{P}(Y = L|X = s) = \binom{m}{L} \mathsf{P}(X = s)^L (1 - \mathsf{P}(X = s))^{m-L} \tag{8}$$

where m is the number of possible links and L the number of links in the network.

We should now be able to calculate if the estimated number of links for any support level is significant with the help of 8, where a two sided significant result at $\alpha \approx 0.05$ exists if the deviation from the expected mean $\mathbf{E}(Y) = mp_L$ is $> 2\times$ the standard deviation, $SD[Y] = \sqrt{mp_L \cdot (1 - p_L)}$. Consideration should also be taken to check equation 7 to ensure that $SD[X] = \sqrt{(bp \cdot (1 - p))}$ to estimate the number of bootstraps needed to ensure that no link with 100% support is deemed insignificant,

$$\left\lceil bp + 2 \cdot \sqrt{(bp \cdot (1 - p))} \right\rceil < b \tag{9}$$

We are uniquely interested when $\mathsf{P}(X \geq s)$ (or alternatively $\mathsf{P}(X \leq s)$), rather than when $\mathsf{P}(X = s)$, because we are looking at the probability that any large proportion of bootstraps gives the same result. In some cases we may want to look at underrepresented levels of support, but in general we want to know if we have more support than would be expected by chance. The proportion of links having a high overlap is restructured from 7 to be

$$\mathsf{P}(X \geq s) = \sum_{X=s}^{b} \binom{b}{s} p^s (1 - p)^{b-s} \tag{10}$$

and 8 becomes

$$\mathsf{P}(Y = L|X \geq s) = \binom{m}{L} \mathsf{P}(X \geq s)^L (1 - \mathsf{P}(X \geq s))^{m-L} \tag{11}$$

We have developed a randomization process that allows control of dataset properties while completely obfuscating data relationships within the underlying network, presenting a proper null hypothesis necessary for validation, which our initial results show a significant difference between the support for random and inferred networks. Discussion within the field has brought to question whether this is a proper null hypothesis, *i.e.* are we just claiming our methods to be better than random datasets with shared data properties, or rather are we claiming something more. This is explored more in the MYC section to follow, where shuffling of the data has lead to random networks composed of shuffled MYC data, maintaining network properties, namely intrampatness (IAA) and rank, but the relationships therewithin are necessarily obfuscated to allow for proper weighing of inference.

2.3 Preliminary Results

Fig. 1 shows the high correlation of observed MYC data to that predicted in the model. It is encouraging that we witness affirmation of our goal, ie recovering strong support for inferred network, through our nested bootstrapping method.

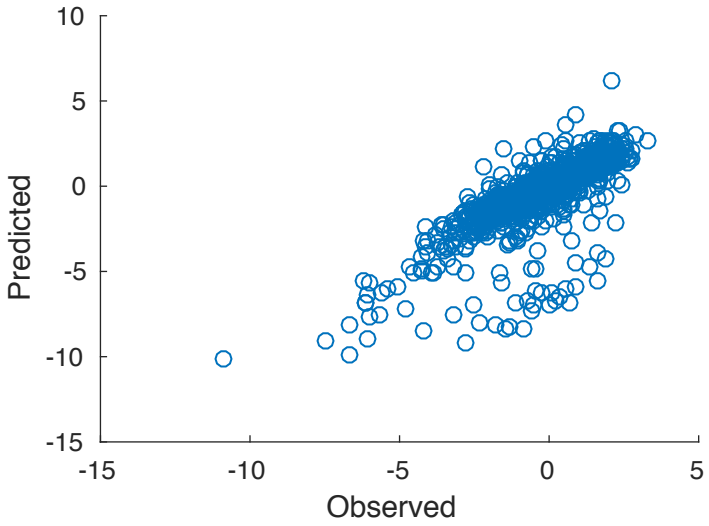
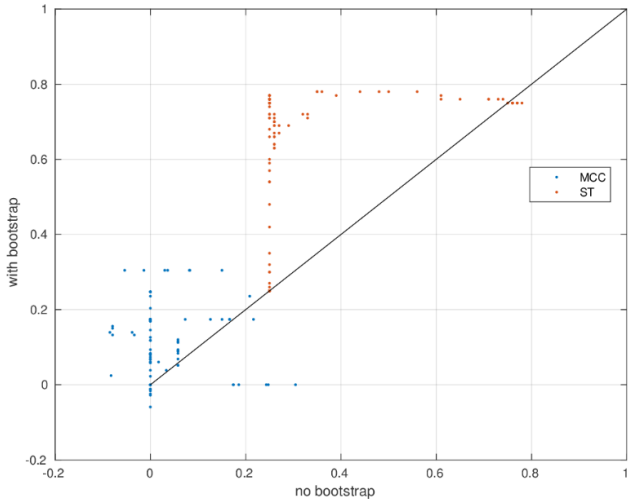
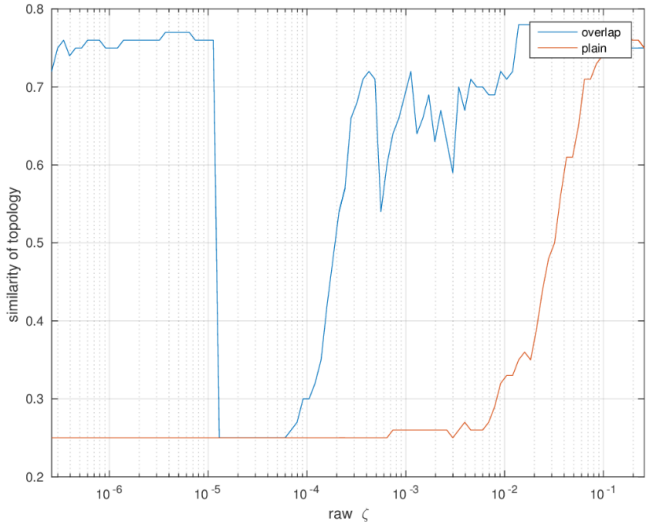


Figure 1: Correlation of Model to Data

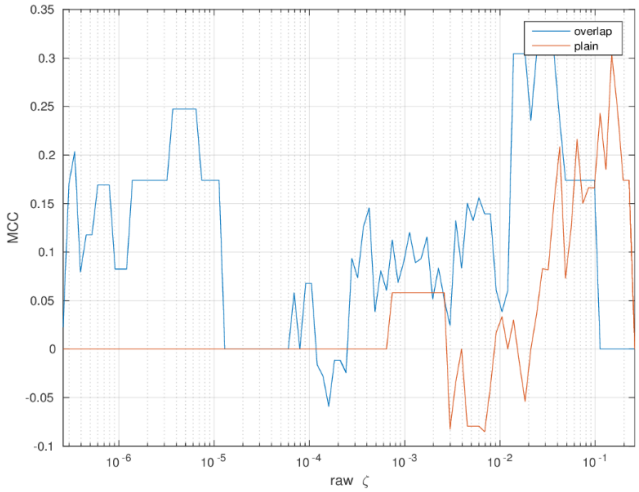
Fig. 2 depicts three different tests of performance on simulated data ten node data with condition number $\approx$ 107 and SNR 10 using 100 iterations of 1000 bootstraps over triplicate single perturbations. This displays the strength of our boostrapping procedure via correlation across network sparsity levels(a), by topological similarity (b) and MCC (c) across the zeta parameter (sparsity).



(a) Bootstrap Performance



(b) Similarity of Topology



(c) MCC Performance

Figure 2: Overall metrics of bootstrap performance: (a) comparing boostrap and non-boostrap LASSO across all sparsity values; while (b) compares the bootstrap and without boostrap methods by topological similarity and bootstrap largely outperforming other than an overlap, and (c) compares the same by MCC performance.

2.4 Future Direction

Our investigation will aim to deliver meaningful prediction, primarily based on IAA network evaluation, which defines biological network interaction strength and enables amplification and attenuation of different signals. Our focus will be on various signal-to-noise calculations as well as the support each network is given to attain significant IAA, and thus how representative each predicted network likely is of the true network. Only then can we estimate the error possible at each step of network formation and calculate a likelihood in the form of a positive support metric relative to support of known networks with predictions still made from their raw datasets.

Experimental datasets offer limited information with which to account for variance among any measured variables, and thus are inherently less informative than synthetic datasets created with known noise. Thus, synthetic datasets entered into the this linear dynamical model arrive at more frequent levels of high support quicker than comparable, experimentally gathered datasets. The nature of the result can be assessed relating the ratio of wanted characterization to inaccuracies native to the background state of the data. This ratio of signal to noise is represented in a variety of SNR calculations, factoring different interests each particular variation. The general calculation for *Signal-to-Noise Ratio (SNR)* and *Multivariable signal-to-noise ratio (MSNR)* are as follows

$$SNR = \frac{\min(\text{svd}(\boldsymbol{Y}))}{\max(\text{svd}(\boldsymbol{E}))} \tag{12}$$

and

$$\text{MSNR} \triangleq \frac{\min_{\|\boldsymbol{r}\|_2=1}, \|\mathbf{E}[\boldsymbol{Y}]\boldsymbol{r}\|_2}{\mathbf{E}\left[\|\boldsymbol{Y} - \mathbf{E}[\boldsymbol{Y}]\|_2\right]} \tag{13}$$

where E denotes the expectation operator that in practice should be replaced by an estimate. The MSNR provides a test of a data set suitability for invalidation of models of the network structure.

$$SNR_{matrix} = \frac{\delta_n(\Phi)}{\delta_1(\gamma)} ... \frac{+\delta_n(\Xi)}{+\delta_1(\Pi)} \tag{14}$$

3 *The Myc Dataset*

The Myc gene is a gene studied extensively in the context of human cancers. Here, we use it as an anchor or hub gene around which to assemble a small library of genes all known to play instrumental roles in the progression of human cancers. In this way, we test a collection no less than 40 genes which share similar functionality and should thus be expressed in some correlated manner [8]. This approach provides every component necessary to uncover a network of interconnected functionality and expression, and so we (previous postdoc M.Studham) introduce individual RNAi single and double gene perturbations. We are able to infer a network of interaction among the dataset genes by taking care to measure the remaining unperturbed genes' response to said perturbation. This will uncover not only any direct interactions stemming from a such singular knock-down experiments, but also the indirect and overall effect the perturbation has on the selected geneset.

3.1 Goals

It is important to create a GRN for Myc and the other genes constituting this genesets because thousands of genes have been show to be targeted offshoots, of which many have known affil-

iations to various types of mammalian cancer. Our Myc dataset is composed of 40 genes, all known to be related to human cancer in some manner, and thus hypothesized to have highly intertwined activity as a group. We attempt to model the interactions among an established geneset by relating the affect of single and double gene perturbations to the expression of the rest of the set.

3.2 Methods

The dataset is comprised of $M = 2 \cdot 40 - 5$ samples and $N = 40$ genes, mean value estimate of 3 replicates derived from RNA interference (RNAi) knockdown. Myc pathway genes were perturbed one at a time for the *singles* and in pairs for the *doubles* to bring out weak signals. These perturbations were then quantified using RNA-Seq and normalized using the ddCt method, normalizing for both experimental and technical replicates. The actual gene-specific RNAi knock-down was not measured, and instead the response in interacting genes is taken as a proxy, assuming "-1"s were counted along the diagonal axis (self perturbation).

The following signal to noise ratio definitions (15) clearly show that our confidence level α will scale the estimated SNR values, table 1 show the SNRs for two confidence levels $\alpha \in \{0.01, 0.05\}$.

$$\begin{aligned} \mathrm{SNR}_m &\equiv \frac{\sigma_N(\boldsymbol{Y})}{\sqrt{\chi^{-2}(\alpha, NM)\lambda}} \\ \mathrm{SNR}_v &\equiv \arg\min_i \frac{\|\boldsymbol{y}_i\|}{\sqrt{\chi^{-2}(\alpha, M)\lambda}} \\ \overline{\mathrm{SNR}}_v &\equiv \mathrm{mean} \frac{\|\boldsymbol{y}_i\|}{\sqrt{\chi^{-2}(\alpha, M)\lambda}} \end{aligned} \tag{15}$$

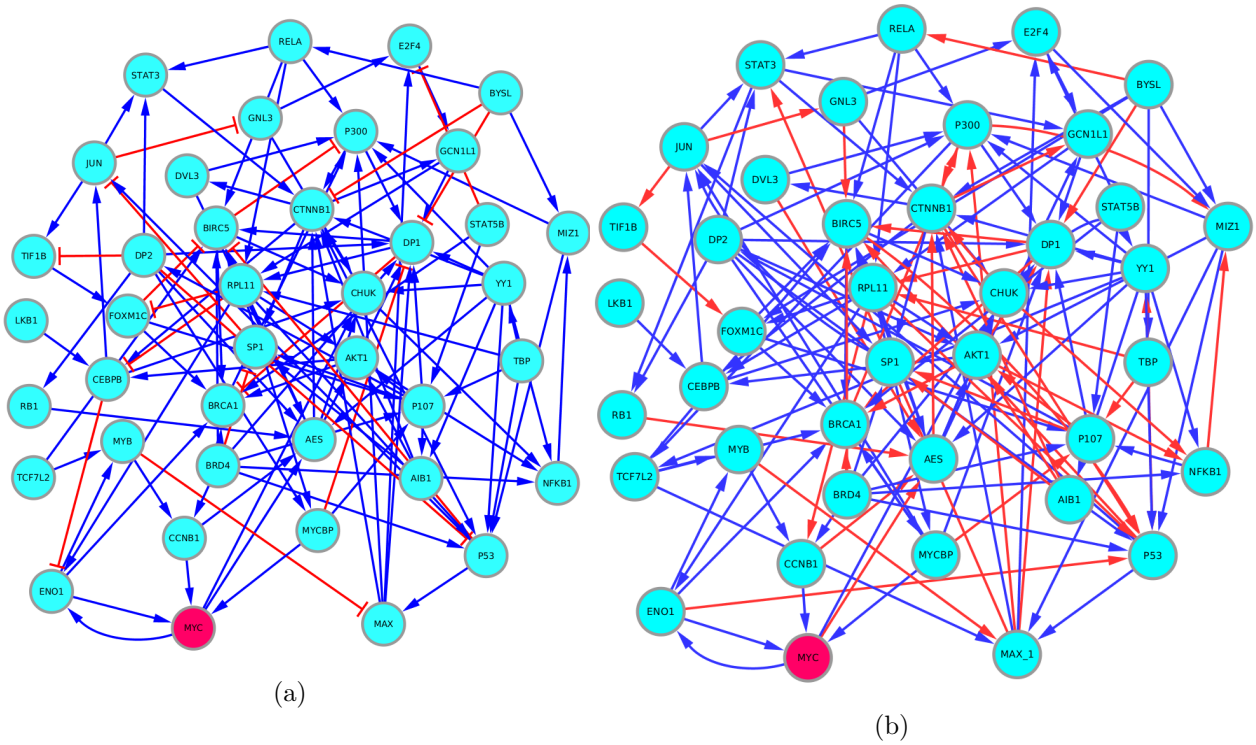
Table 1: Noise properties for Myc data for two confidence levels $\alpha \in \{0.01, 0.05\}$.

40 genes	$\alpha = 0.05$	$\alpha = 0.01$		
genes over $\mathrm{SNR}_v = 1$			$\alpha = 0.05$ (13 genes)	$\alpha = 0.01$ (11 genes)
SNR_m	0.0099367	0.0098511	0.21055	0.17683
SNR_v	0.63164	0.60068	1.2823	1.2714
$\overline{\mathrm{SNR}}_v$	0.94442	0.89812	1.6087	1.5915

where σ_N is the smallest singular value, χ^{-2} is the inverse chi square distribution with α confidence level and NM degrees of freedom, λ is the variance and $\boldsymbol{y}_i$ is the i^{th} gene's response over the set of samples in the $\boldsymbol{Y}$ expression matrix. The different SNR definition represent different aspects of how the noise is affecting the data. SNR_m represent how the SNR is affecting the complete expression dataset, SNR_v represent the noise level compared to each gene, by our convention we use SNR_v to represent the minimum SNR_v of the data set and $\overline{\mathrm{SNR}}_v$ to represent the mean SNR_v over the data set.

3.3 Preliminary Results

Preliminary visualizations of the network are presented based both on bootstrap weighted links and bootstrap signed links in Fig. 3. These networks were created sampling 1,000 nested iterations of 1,000 bootstraps from the MYC RNAi perturbation data, using link support levels .96 and .9 as inclusion criteria, respectively.



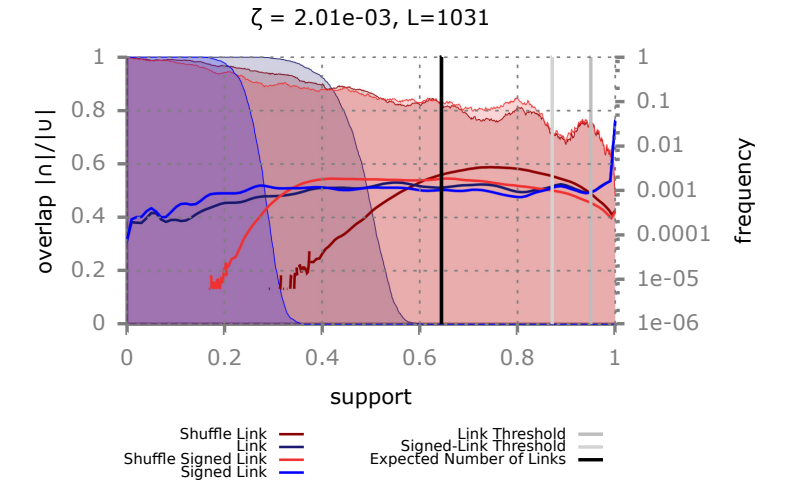
(a) (b)

Figure 3: Signed MYC networks based on bootstrap weight and bootstrap sign; (a) 157 link MYC network of minimum .965 support level, red arrows are repression regulatory interactions, blue are activation regulatory interactions. (b) Tentative 146 link MYC network of minimum .903 support level, red arrows are negatively signed regulatory interactions, blue are positively signed regulatory interactions.

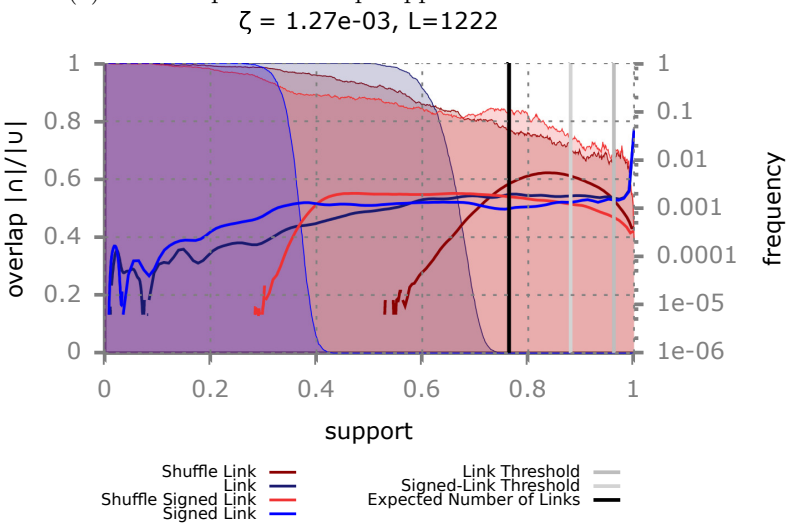
While random network overlap can be modeled by 11, we actually want to model the bootstrap procedure with the LASSO algorithm in order to arrive upon a specific network over all bootstraps. Now consider the linear mapping of Eq. 2. To simulate a random response we model a system without interactions, $\boldsymbol{A} = \boldsymbol{0}$, which reduces Eq. 2 to $\boldsymbol{Y} = \boldsymbol{E}$. Under these circumstances we would expect to only see noise in the data, with no input noise $\boldsymbol{F}$.

$$\text{relative overlap} = \frac{intersection}{union} = \frac{|\boldsymbol{A}_1(\alpha) \cap \boldsymbol{A}_2(\alpha) \cap ... \cap \boldsymbol{A}_{n-1}(\alpha) \cap \boldsymbol{A}_n(\alpha)|}{|\boldsymbol{A}_1(\alpha) \cup \boldsymbol{A}_2(\alpha) \cup ... \cup \boldsymbol{A}_{n-1}(\alpha) \cup \boldsymbol{A}_n(\alpha)|} \tag{16}$$

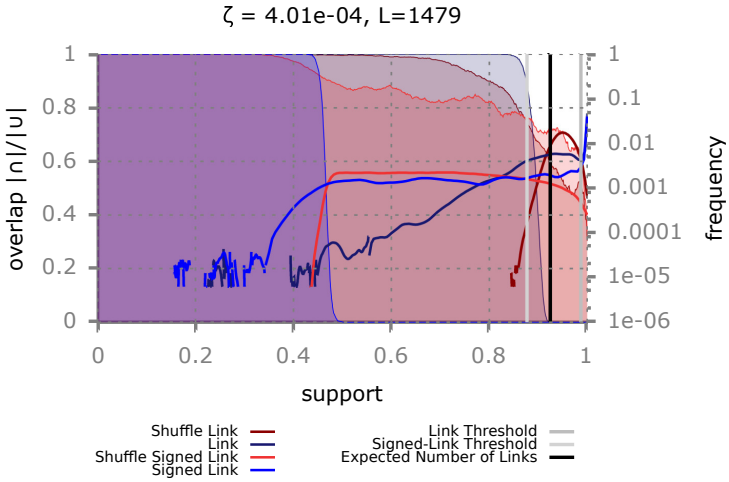
where $\boldsymbol{A}_i$ is network for run $i \in (1, .., n)$, with support $\geq \alpha \in [0, 1]$ and $|.|$ indicate cardinality of the set. Fig. 4a, Fig. 4b and Fig. 4c give an overview of the support gained for the MYC network signed links for networks of reasonable sparsity compared to overall capacity.



(a) Bootstrap and overlap support for 1031 link network



(b) Bootstrap and overlap support for 1222 link network



(c) Bootstrap and overlap support for 1479 link network

Figure 4: Bootstrap frequency for 100 bootstrap runs with 1000 bootstraps for signed MYC links (light blue), signed shuffled link (light red), unsigned MYC links (dark blue) and unsigned shuffled links (dark red) with the overlap (y-left) and frequency (y-right) of support for a network. Shaded areas depict network overlap, and vertical lines show sparsity (black, *i.e.* expected number of links) and first crossing of shuffle over non-shuffled frequency. Three highly supported networks of different link number, 1031, 1222 and 1479 out of 1600, respectively, show a good support threshold (b) with networks above (c) and below (a).

3.4 Future Direction

Building in the sign into the bootstrap procedure for LASSO seems to work, but the least squared and total least squared methods do no perform well; migrating them to constrained and/or weighted total least squares should solve this discrepancy in performance and bring it closer to supporting the LASSO output.

3.4.1 Vectorization of linear matrix equations

Assuming that $\boldsymbol{Y} \in \mathbb{R}^{p\times m}$, $\boldsymbol{Q} \in \mathbb{R}^{p\times o}$, and $\boldsymbol{P} \in \mathbb{R}^{n\times m}$ are given and $\boldsymbol{\Theta} \in \mathbb{R}^{o\times n}$ is unknown. The linear matrix equation

$$\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{\Theta}\boldsymbol{P} \tag{17}$$

is equivalent to the following standard system of pm linear equations[9] p. 254-255

$$\vec{\boldsymbol{Y}} = (\boldsymbol{P}^T \otimes \boldsymbol{Q})\vec{\boldsymbol{\Theta}}. \tag{18}$$

Here $\vec{\boldsymbol{Y}}$ denotes a column vector formed by stacking the columns of $\boldsymbol{Y}$ beneath each other and $\otimes$ the Kronecker product[9] p. 243

Data from steady-state experiments is often conveniently stored in matrix form such that the observed responses from m experiments are in $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ and the applied perturbations in $\boldsymbol{P} \in \mathbb{R}^{n\times m}$, while the static gain matrix $\boldsymbol{G} \in \mathbb{R}^{n\times n}$ of the system or its inverse the network matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is sought

$$\boldsymbol{Y} = \boldsymbol{G}\boldsymbol{P} = -\boldsymbol{A}^{-1}\boldsymbol{P}.^{10} \tag{19}$$

$$\vec{\boldsymbol{Y}} = (\boldsymbol{P}^T \otimes \boldsymbol{I})\vec{\boldsymbol{G}}^{10} \tag{20}$$

4 *Benchmarking GeneSPIDER*

While GeneSPIDER is certainly the first package to factor data properties into the GRN inference pipeline, it is not the only inference toolkit available. Among any highly qualified group of peers rank needs to be measured, and thus we set out to compare capabilities among the diverse assemblage of inference methods currently available. An accurate and thorough benchmark is crucial for continued development of inference methodologies, fore it will illuminate points of obvious shortcoming in the shadow of superior methods; thus not wanting to tread over another, novel techniques will be produced with the aim to surpass the present cutting edge.

4.1 Goals

The quality of the inferred network model, i.e. the number of false positives and negatives, depends both on the inference method and the recorded data. The benchmark would thus allow us to differentiate between performance of algorithms and quality of data sets, and demonstrate that both are essential for successful network inference. The GRN inference community has, e.g. in the DREAM challenge, benchmarked network inference methods by applying them to the same data and comparing their performance[11]. Benchmarking provides guidance on both (i) which inference method to use for a particular data set, and (ii) which errors to expect in the inferred network.

4.2 Future Direction

From the work done in[12] one can find a number of algorithms based on time series data, however, their performance and the experimental design needed to actually utilize these methods is not fully investigated. In this case steady state data models are easier to interpret and investigate. Steady state data posits to limits the complexity of the models by assuming a linear relationship

between the initial and end point of a response curve, regardless of the intermediary curve. This lowers what is required for inferring interactions among biomolecules, offering us the possibility to diversify and expand in other areas such as data set properties and give us a chance to thoroughly investigate this branch of algorithms. Gene Network Inference Methods/algorithms to be compared with those incorporated into GeneSPIDER follow. An ideal benchmarking package should be easily generate many networks and data sets with desired properties using different experiment designs. It should contain many network inference methods for easy testing and comparison, and be extendable and robust, for easy future extension with new methods. Most of the reviewed packages utilize complex multi-layer models of biochemical kinetics and focus on mimicking real data. The ability to fine tune specific properties is however limited in all current packages we reviewed. Their unnecessary complexity in general hampers any ability to gain insight, so the need for a package focusing on generating approximations with tuned properties, such as GeneSPIDER, is in our opinion a highly necessary implementation for the field's continuation. A common approach in the previous benchmarking papers is to pick "representatives" for each network inference category that is investigated. Some examples of different classes of algorithms that can be used to acquire causal mappings are shown in the list below.

Current packages which incorporate sundry assortment of this list include the following:

- *Gene Network Inference with Ensemble of Trees (GENIE3)* – operates in much the same fashion as MRNET, save using random forests and extra-trees for regression and feature selection.
- *SIRENE* - uses SVM to measure similarity between genes of different classes using the common Gaussian radial basis function kernel
- *GeneNetWeaver*[13] is capable of generating networks with the number of nodes and the in-degree specified by the user. The user can also choose among a variety of *in silico* perturbations when generating data. The generated nonlinear dynamical model is based on several types of omics data, in order to mimic a real biological system and give the simulated data all the properties of real data.
- SynTReN and *GeNGe*[14] distinguish themselves from GeneNetWeaver by use of Michaelis-Menten and Hill kinetics in the ODEs used for data creation.
  - *SynTReN*[15],like GeneNetWeaver, also utilizes selected dynamic structures and sub-networks from *Escherichia coli* and *Saccharomyces cerevisiae* to generate realistic networks described by a nonlinear ODE model. Both also support network generation by allowing user specified noise levels and numbers of network and latent nodes.
  - GeNGe uses a nonlinear model similar to both GeneNetWeaver and SynTReN to simulate data which combine linear kinetics with nonlinear kinetics for translation and transcription respectively and provides functionality similar to GeneNetWeaver with a number of standard perturbation designs and time-series data generation, as well as the same additional complexity. *GeNGe* contains functions for topological characterization, such as the in- and out-degree distributions, average path length, and clustering coefficients.

	Measure	Equation
Boolean	Pearson	$corr(X_i, X_j) = \frac{cov(X_i,X_j)}{\sigma(X_i)\sigma(X_j)}$
	Spearman's Rank	
	Z-score	$Z = \frac{x-\mu}{\sigma}$
	PCIT	$orr_{ij}(X_i, X_j, X_k) = \frac{corr(X_i,X_j)-corr(X_i,X_k)corr(X_j,X_k)}{\sqrt{(1-corr(X_i,X_k))^2(1-corr(X_j,X_k))^2}}$
Linear	*Glmnet*	
	LeastSquares	see eq 3, eq 4
	TotalLeastSquares	see eq 5b
	LASSO	see eq 6
Neural Net	Relevance Network (RN)	$I(X_i, X_j) = \sum_{x_i \epsilon X_i} \sum_{x_j \epsilon X_j} p(x_i, p_j) log \frac{p(x_i,p_j)}{p(x_i)p(x_j)}$
	MRNET	$X_i^{MRMR} = arg\max_{X_i \epsilon V/S}(u_i - r_i)$ $r_i = \frac{1}{\lvert S \rvert} \sum_{X_k \epsilon S} I(X_i, X_k)$
	CLR	$z_i = max_j\left(0, \frac{I(X_i,X_j)-\mu_i}{\sigma_i}\right)$ $z_{ij} = \sqrt{z_i^2 + z_j^2}$
	ARACNE2	$I(X_1, X_2) \leq \min(I(X_1, X_2), I(X_2, X_3))$

Table 2: **Class and Equation to be Benchmarked**: Measures a single dataset is to processed and compared against in this Benchmark. Correlation based upon the aforementioned assumption that correlated gene expression is indicative of similarity of function. Not able to infer direction. Least Squares aims to minimize the overall difference of the errors restricted to the y-axis. Total Least Squares expands upon the least squares regression by incorporating error of both x and y axes. Partial Correlation and Information Theory (PCIT), is similar to ARACNE2 but rather than calculating Mutual Information, it calculates partial correlation in order to nullify the third gene's effect. Relevance Networks (RN), are a measure of mutual information, and while they are able to identify non-linear gene relationships, they are not able to infer direction or activation/inhibition. Minimum Redundancy/ Maximum Relevance Networks (MRNET) treats all genes as terminal target genes, selecting the best regulator set. Context Likelihood Relatedness (CLR) is an extensions of RN in many respects, CLR takes into account the mutual information (MI) background distribution $I(X_i, X_j)$ into account, from where the most probable interactions deviate, ranked by the following z-score per gene,i, aiming to minimize the FDR. Like CLR, Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE2) is an extension of RN; it considers so called *interaction triangles*, indirect gene interactions involving a single additional node, calculating MI for each pair, which can then be discarded via user specified threshold.

References

[1] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren and R. Guthke, *Bio Systems*, 2009, **96**, 86–103.

[2] A. Tjarnberg, T. E. Nordling, M. Studham, S. Nelander and E. L. Sonnhammer, *Mol Biosyst*, 2015, **11**, 287–296.

[3] M. L. Arrieta-Ortiz, C. Hafemeister, A. R. Bate, T. Chu, A. Greenfield, B. Shuster, S. N. Barry, M. Gallitto, B. Liu, T. Kacmarczyk *et al.*, *Molecular Systems Biology*, 2015, **11**, 839.

[4] I. Markovsky and S. V. Huffel, *Signal Processing*, 2007, **87**, 2283 – 2302.

[5] V. Mahboub and M. Sharifi, *Journal of Geodesy*, 2013, **87**, 279–286.

[6] J. B. Rosen, H. Park and J. Glick, *SIAM Journal on Matrix Analysis and Applications*, 1996, **17**, 110–126.

[7] F. R. Bach, Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, 2008, pp. 33–40.

[8] M. Raff, B. Alberts, J. Lewis, A. Johnson and K. Roberts, *National Center for Biotechnology Information's Bookshelf*, 2002.

[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge university press, 2012.

[10] T. E. M. Nordling, *PhD thesis*, KTH School of Electrical Engineering, Automatic Control Lab, 2013.

[11] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins and G. Stolovitzky, *Nature Methods*, 2012, **9**, 796–804.

[12] C. A. Penfold and D. L. Wild, *Interface Focus*, 2011, **1**, 857–870.

[13] T. Schaffter, D. Marbach and D. Floreano, *Bioinformatics (Oxford, England)*, 2011, 2263–2270.

[14] H. Hache, C. Wierling, H. Lehrach and R. Herwig, *Bioinformatics*, 2009, **25**, 1205–1207.

[15] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor and K. Marchal, *BMC Bioinformatics*, 2006, **7**, 1–12.