

Systems biology

A generalized framework for controlling FDR in gene regulatory network inference

Daniel Morgan ¹, Andreas Tjärnberg², Torbjörn E. M. Nordling³ and Erik L. L. Sonnhammer^{1,*}

¹Department of Biochemistry and Biophysics, Stockholm Bioinformatics Center, Science for Life Laboratory, Stockholm University, Stockholm, Sweden, ²Department of Physics, Chemistry and Biology/Bioinformatics, Linköping University, Linköping, Sweden and ³Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan

*To whom correspondence should be addressed.

Associate Editor: Bonnie Berger

Received on May 21, 2018; revised on August 23, 2018; editorial decision on August 26, 2018; accepted on August 28, 2018

Abstract

Motivation: Inference of gene regulatory networks (GRNs) from perturbation data can give detailed mechanistic insights of a biological system. Many inference methods exist, but the resulting GRN is generally sensitive to the choice of method-specific parameters. Even though the inferred GRN is optimal given the parameters, many links may be wrong or missing if the data is not informative. To make GRN inference reliable, a method is needed to estimate the support of each predicted link as the method parameters are varied.

Results: To achieve this we have developed a method called nested bootstrapping, which applies a bootstrapping protocol to GRN inference, and by repeated bootstrap runs assesses the stability of the estimated support values. To translate bootstrap support values to false discovery rates we run the same pipeline with shuffled data as input. This provides a general method to control the false discovery rate of GRN inference that can be applied to any setting of inference parameters, noise level, or data properties. We evaluated nested bootstrapping on a simulated dataset spanning a range of such properties, using the LASSO, Least Squares, RNI, GENIE3 and CLR inference methods. An improved inference accuracy was observed in almost all situations. Nested bootstrapping was incorporated into the GeneSPIDER package, which was also used for generating the simulated networks and data, as well as running and analyzing the inferences.

Availability and implementation: <https://bitbucket.org/sonnhammergrni/genespider/src/NB/%2BMethods/NestBoot.m>

Contact: erik.sonnhammer@scilifelab.se

1 Introduction

A living cell responds to signals or changes in the environment according to the system encoded in its gene regulatory network (GRN). Understanding the GRN therefore offers valuable insights into how a cell functions. GRNs can be inferred based on measuring gene expression responses to system perturbations and applying mathematical modeling to find the GRN that best agrees with the data (Tegnér and Björkegren, 2007). Many GRN inference methods

exist that solve this problem. However, the solution usually depends on algorithmic parameters, such as an arbitrarily set regularization parameter which is needed to produce a sparse GRN (Tibshirani, 1996). Furthermore, as each method makes its own set of assumptions and finds the optimum in different ways, they will often yield different and inconsistent solutions. GRN inference accuracy has been shown to decrease with less informative data (Tjärnberg *et al.*, 2015, 2017). Unfortunately, experimental datasets are generally

very noisy and therefore uninformative, hence there is a great need to improve GRN inference accuracy, especially in uninformative data. One approach to solve this dilemma is to assess reproducibility of inferred links using repeated resampling of the data, i.e. bootstrapping.

Several recent GRN inference methodologies take this approach. *Stability selection* employs randomized parameterization of LASSO (Tibshirani, 1996) combined with the irrepresentable condition (Zhao and Yu, 2006). In conjunction with a stability criterion, which chooses links with subsampled frequencies above an expected upper false positive boundary, more reliable GRN inference was demonstrated (Meinshausen and Bühlmann, 2010). *Random LASSO* (Wang et al., 2011) improves link selection by averaging across bootstrapped distributions of randomly selected subsets of the data. *TIGRESS* combines stability selection with iterative LARS estimation (Haury et al., 2012). Although these methods address the reproducibility of results, they do not consider the number of false links expected to be included in an inferred network. A method that attempts to estimate this is *BINCO* which uses a U-shaped link frequency distribution to differentiate between true and false links (Li et al., 2013). Unfortunately, very often the inferred network does not produce the U-shaped frequency histogram necessary for BINCO (Tjärnberg et al., 2017), hence there is a need for a more general method.

We here present a new bootstrapping-based framework for inferring reproducible GRN links with controlled false discovery rate (FDR), that is generally applicable to any dataset and any GRN inference algorithm. By nesting multiple bootstrap we also get an estimate of how stable the bootstrap support values are. FDR values are derived from the distribution of support values for false links produced when shuffling the original data. A consensus GRN is formed from links exceeding the FDR cutoff in all runs of bootstraps. We call the method *nested bootstrapping* in analogy to similar approaches in other fields (Hinkley and Shi, 1989).

We present accuracy benchmarks of nested bootstrapping applied to five GRN inference methods and compare to their native performance using *in silico* datasets covering many data property combinations (Tjärnberg et al., 2015, 2017), and to three biological datasets that contain a gold standard network for accuracy measurement. The experimental design for the *in silico* datasets is perturbation-induced, steady-state measurements (Ideker et al., 1999). We also evaluated the benchmark performance of *GENIE3* (Huynh-Thu et al., 2010), a top performing bootstrap-based inference method (Greenfield et al., 2010).

2 Materials and methods

2.1 NestBoot algorithm

Applying any method within a nested loop, the inner bootstrap iteration Q samples individual experiments with replacement (Efron and Tibshirani, 1994) and infers a network, which process is iterated per run R in a procedure generally referred to as Nested Bootstrapping (Hinkley and Shi, 1989); we distinguish this algorithmic implementation as *NestBoot*, for its expressed purpose of limiting false links via parallel shuffled data inference. Link frequencies among individual networks inferred during R runs are collected to estimate stability. A version of this pipeline is run in parallel along data shuffled experiment-wise (Fig. 1). Comparison of link frequencies between this shuffled null and the measured data dictates the bootstrap support guaranteed for the desired False Discovery Rate

(FDR) cutoff level. Links which are present in R bootstrap runs above this cutoff are then extracted as a final consensus network.

The implementation of Nested Bootstrapping, the *NestBoot* algorithm (Fig. 1), requires the data $[P, Y]$ and proceeds as follows across any measured and shuffled datasets for any individual sparsity level:

Algorithm 1 Nested Bootstrap with FDR Cut-Off for Single Sparsity. Q and R are the number of inner and outer bootstrap runs, respectively. N is the number of genes.

```

1: procedure
2:   for  $j = [1..R]$  do
3:     generate shuffled data  $[Y^{(s)}]$ 
4:     for  $i = [1..Q]$  do
5:       while  $\text{rank}([P^{(b,i)}, Y^{(b,i)}]) < N$  do
6:         Generate bootstrap dataset from measured  $[P^{(b)}, Y^{(b)}]$  &
7:         shuffled data  $[P^{(b,s)}, Y^{(b,s)}]$ .
8:         Infer networks for measured & shuffled data.
9:         Collect frequency profiles.
10:      Calculate FDR cutoff based on measured & shuffled profiles.
11:      Form consensus network using links  $>$  cutoff (Fig. 2).
```

The utility of this procedure comes from the link persistence across bootstrap runs, comparing shuffled to measured data. A final consensus network includes links that in each run exceed a bootstrap support cutoff B selected by a user-defined FDR (Eq. 1), as follows

$$\frac{\#links_{shuffled}(i,j) > B}{\#links_{measured}(i,j) > B} \leq \text{desiredFDR}, \quad (1)$$

where the area under the population frequency curves operates over measured and shuffled bootstrapped data. Constraining consensus networks' links is entirely dependent on a valid null to estimate false links. This second dataset comprises our null model requiring data properties be maintained and thus is compiled by shuffling the measured dataset experiment-wise, $[Y^{(b,s)}]$. The data shuffling process allows for control of dataset properties while completely obfuscating underlying relationships, presenting a proper null hypothesis necessary for validation. Comparison between measured and shuffled bootstrap frequency support allows estimation of false links as part of a shared inferred link distribution. This process enables us to achieve the aforementioned limitation of FDR when inferring GRN, garnering reliability as well as gathering an additional level of confidence in the included links previously unavailable. In this way we are able to determine an ideal threshold of link inclusion, as illustrated by the support of both parallel process (Fig. 2). The ratio between these two frequency-support curves reflects the number of bootstraps where links were present and thus allows restriction of false link inclusion. This threshold ensures a minimal, acceptable amount of false links with minimal links missing (Fig. 6). Note that each bootstrapped dataset is checked for full rank experimental independence so not to perform inference using less informative datasets. Furthermore, since we require this level of bootstrap support in every run, the FDR level is defined very conservatively.

A known gold standard can then be used to measure true and false links and thereby estimate any metric of accuracy. We measure benchmark accuracy as an unbiased ratio in terms of all four link

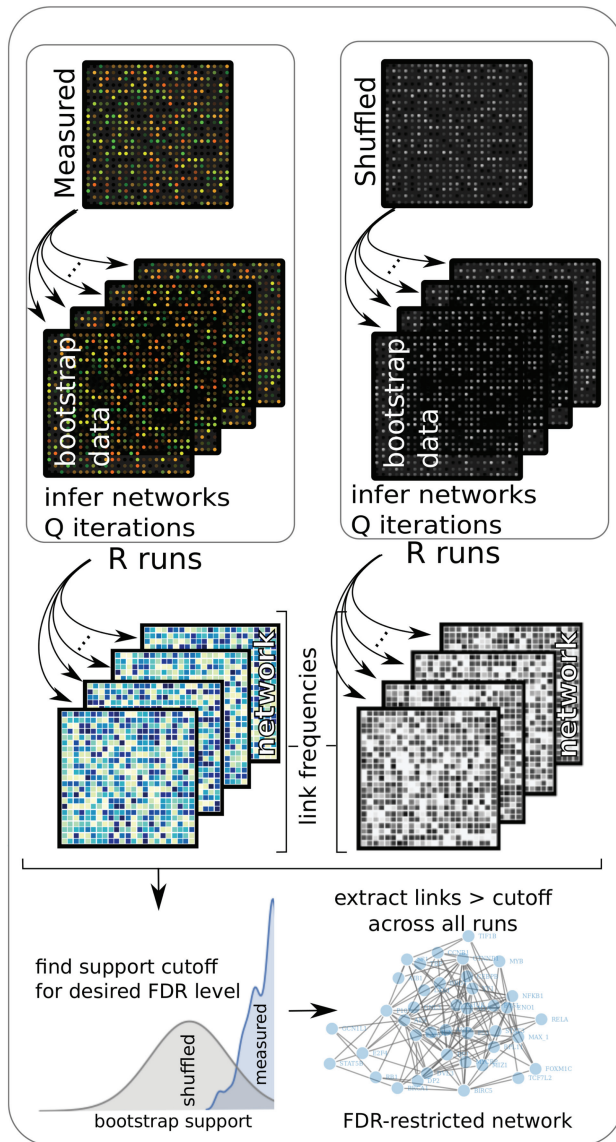


Fig. 1. FDR estimation via NestBoot algorithm for a given sparsity. Data is sampled at each bootstrap iteration and a bootstrap network is inferred. After Q iterations, bootstrap support for each link is saved as the frequency at which it was inferred, keeping direction and sign of the link as separate events. This process is repeated for R runs, which are later used to evaluate stability. The distribution of bootstrap support for all runs is compared with the distribution for shuffled expression data, and this is used to locate the bootstrap support cutoff at the desired FDR level. Links are then extracted which are present in each run above this cutoff

conditions, true and false, positive and negative, in Matthew's Correlation Coefficient (MCC) (Eq. 2) (Matthews, 1975). This represents an expected level of accuracy when inferring networks from datasets of various qualities.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (2)$$

where TP are links predicted correctly as existing, TN are correctly predicted as not existing, FP are nonexistent links predicted to exist, and FN are existing links predicted not to exist.

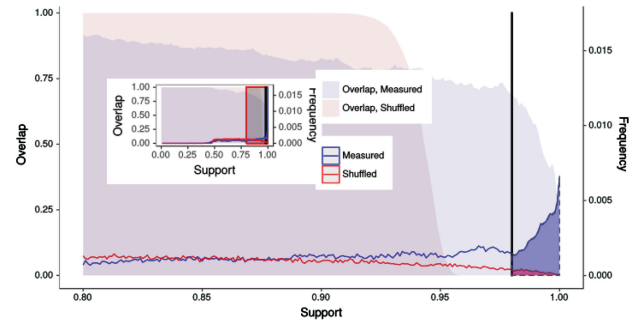


Fig. 2. Bootstrap support by overlap and frequency. Illustration for user-defined FDR cutoff (here 5%) for link inclusion. The blue line depicts link frequency as a function of bootstrap support when using measured data, and the red line when using shuffled data. The black vertical bar defines a bootstrap cutoff above which the measured data links are in the blue area and the shuffled data links are in the red area. In this illustration we set the FDR level for inferred links by finding the cutoff that makes the red area 5% of the total area under the measured data. The overlap is the intersection of links in the bootstrap runs divided by their union. This indicates what level of overlap is expected by chance, and what level is observed in the measured data (Color version of this figure is available at *Bioinformatics* online.)

2.2 Benchmark

40 synthetic 10-gene and 45-gene datasets each with SNR levels ranging from 0.001 to 10 with gold standard networks (Tjärnberg et al., 2015, 2017) are used. Given a dataset we assume that the gene expressions are collected at steady state, and are generated by a GRN $A \in \mathbb{R}^{N \times N}$ approximated by a linear model, where a_{ij} is an element of A , mapping the effect of the perturbation P and its noise F on gene j onto the gene expression of gene i , collected in Y with measurement noise E . $P, F, Y, E \in \mathbb{R}^{N \times M}$, where N = number of measured genes and M = number of samples collected, as follows

$$Y = -A^{-1}(P + F) + E. \quad (3)$$

Datasets were created in the *GeneSPIDER* (Tjärnberg et al., 2017) MATLAB (MATLAB, 2017) environment, where the perturbation information (P) enables transfer of the core network topology into the synthetic knockdown datasets (Eq. 3). We additionally repurposed three experimentally derived datasets (Arrieta-Ortiz et al., 2015; Gardner et al., 2003; Lorenz et al., 2009). The inclusion of their gold standard networks prove invaluable in allowing the direct calculation of inference accuracy. The Arrieta-Ortiz et al. dataset characterized more than this, and thus needed to be reduced from measures of several thousand genes to only those singly perturbed, a subset composed of some 28 genes and perturbation experiments. The Arrieta-Ortiz et al. gold standard network likewise accounts for thousands of genes, thus we select only those measured in the subset data as the network of comparison. We also felt it pertinent to compare to a method which utilizes bootstrapping for inference, though with a different aim and carried out in an entirely different manner, and include GENIE3 (Huynh-Thu et al., 2010) to this end with a cutoff to produce a range of sparsities. For the same effect, a similar cutoff was implemented in CLR.

Data properties can greatly affect the performance of any inference. We therefore implement a conservative estimate of one such particularly relevant property, the signal to noise for each dataset, to highlight the use of data properties as aids for choosing inference method or the tuning of parameters. We express the noise level of the system as the *Signal-to-Noise Ratio* (SNR), the smallest singular value of Y divided by the largest singular value of E (Eq. 4). In cases of biological data when E is unavailable, a different formulation is

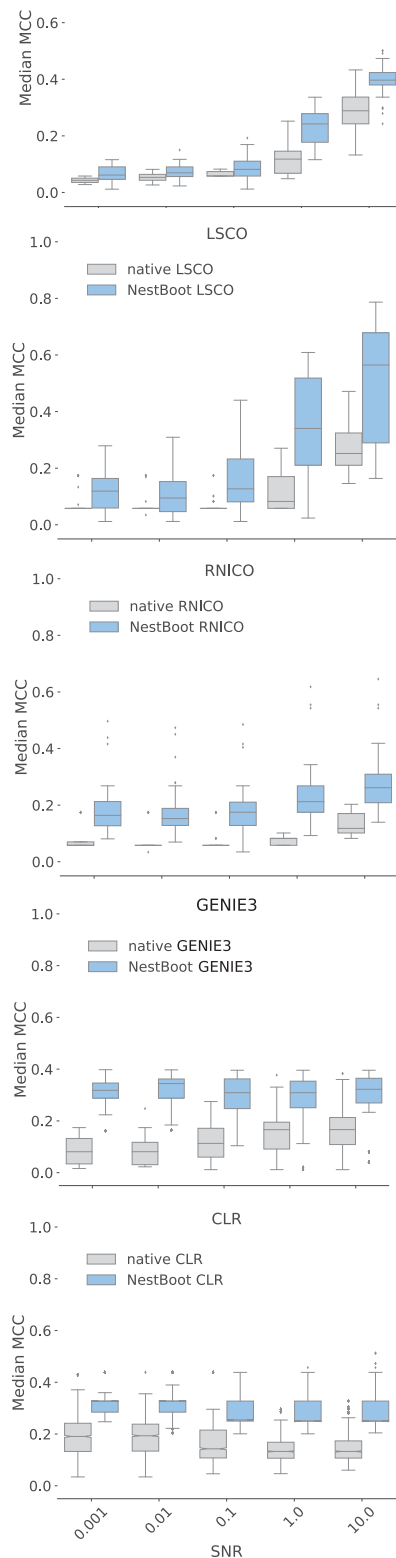


Fig. 3. GRN accuracies under native and NestBoot-enhanced inference on 10-gene synthetic data. The distribution of scores across all datasets is presented in grey where native inference configuration is used and in blue where NestBoot-enhanced inference is enabled for methods LASSO, LSCO, RNICO, GENIE3 and CLR. Accuracy was measured as median MCC across a range of sparsities for each of the 200 datasets of 10 genes. The box displays the median and interquartile range, and whiskers bound points maximally extending 1.5 times this range. Beyond this, outlier points are shown

used, assuming normally distributed noise (Tjärnberg *et al.*, 2017, Eq. S9)

$$\text{SNR}_{Y \text{ true}} \triangleq \frac{(Y)}{\bar{\sigma}(E)}. \quad (4)$$

The case study here carries 100 outer R runs, each to form a consensus from 1000 inner Q bootstrap iterations (Fig. 1). The inner iterations are checked for consistency among outer runs, informing accuracy of the inference, allowing estimation of link persistence.

In the attempt to reverse engineer networks of similar characteristics found throughout biological systems we build upon a foundational investigation of criterion for inferring networks of optimal sparsity (Tjärnberg *et al.*, 2013). We rely upon the long held notion that such elements rely upon specific regulators and thus are not near fully connected, but instead are sparse. We rely on inference methods incorporated into the *GeneSPIDER* (Tjärnberg *et al.*, 2017) MATLAB (MATLAB, 2017) package specifically capable of returning such networks, namely LASSO (Friedman *et al.*, 2009; Tjärnberg *et al.*, 2015, Eq. 4), Least Squares Cut Off (LSCO), Robust Network Inference Cut Off (RNICO) (Tjärnberg *et al.*, 2017), GENIE3 (Huynh-Thu *et al.*, 2010) and Context Likelihood of Relatedness (CLR) (Faith *et al.*, 2007). The three regression methods utilize known perturbation design (P) to map expression to network topologies, whereas GENIE3 and CLR have no such requirement. We utilize sparsity parameters native to the inference methods to retain only relevant interactions in the final network model. Among the sparsities possible to infer through these methods, we seek to estimate what has been found to be biologically relevant networks, resolving to around 3 links per node, or 10% connectivity (Tjärnberg *et al.*, 2013).

3 Results

To evaluate the NestBoot method we benchmarked its effect on five GRN inference methods, using a benchmark of simulated data with a range of properties, as well as three biological datasets. 40 10-gene and 45-gene datasets were created using the *GeneSPIDER* MATLAB package (Tjärnberg *et al.*, 2017). Each initial dataset was scaled to five SNR levels, logarithmically spaced from 0.001 to 10, resulting in 200 datasets. The datasets' gold standard networks are of random topologies with condition numbers 5–105.

Synthetic datasets offer the advantage of precisely measuring accuracy and the effect of properties such as SNR and condition number. Using biological data here is difficult because the true network is not known and the data properties cannot be varied. However, we found three datasets where the authors have proposed a true network and used that as a proxy.

We utilized five established methods of inference to reverse-engineer GRNs from these datasets, and it is around these that we built the nested bootstrapping enhancement. The methods are LASSO via the Glmmnet implementation, based on L1-regularized distance minimization, LSCO, the implementation of least squares with a cutoff to yield sparse networks, RNICO (Tjärnberg *et al.*, 2017), a derivative of *Robust Network Inference* (RNI) (Nordling, 2013) with a variable cutoff to yield multiple sparsities, GENIE3 and Context Likelihood of Relatedness (CLR). For each dataset, the methods were run at 30 different sparsity penalty settings to cover a wide range of sparsities.

Networks were inferred by each method in its native configuration as well as with nested bootstrapping. We measured performance in terms of median MCC, calculated across all initiated

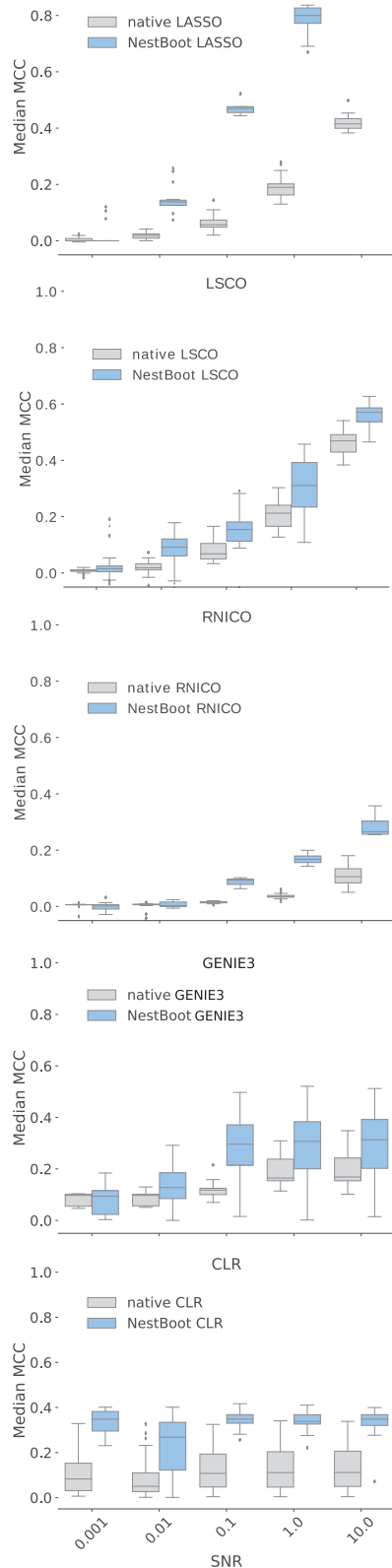


Fig. 4. GRN accuracies under native and NestBoot-enhanced inference on 45-gene synthetic data. The distribution of scores across all datasets is presented in grey where native inference configuration is used and in blue where NestBoot-enhanced inference is enabled for methods LASSO, LSCO, RNICO, GENIE3 and CLR. The accuracy was measured as median MCC across a range of sparsities for each of the 200 datasets of 45 genes. For a description of the plot see Figure 3

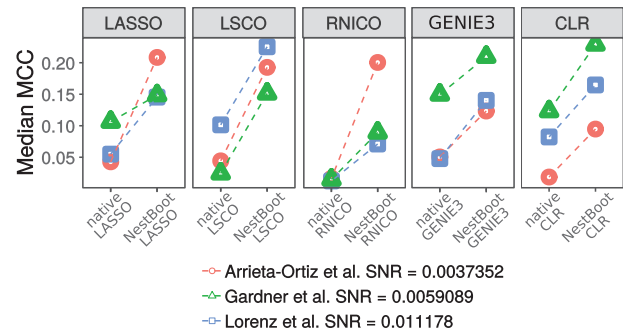


Fig. 5. Experimental NestBoot performance at 5% FDR cutoff. Comparison of native methods to those enhanced with NestBoot on three biological datasets. The measured SNR levels of the three datasets are shown below

sparsities, in order to present a global picture of the accuracies across variably sparse inferred networks. This is intended mainly as a relative accuracy measurement. As we use the same sparsity levels for one method under native and NestBoot-enhanced configurations, these are directly comparable, but the accuracies of different methods may be affected by different choices of sparsity levels. The median will normally be substantially lower than the maximum MCC value across all sparsities, but it is also a more realistic measure for a biological setting where the true GRN is unknown and hence no maximum MCC can be calculated. It would not be appropriate to use e.g. AUROC or AUPR measurements that require a full range of false positives and negatives, because NestBoot with its control of FDR only outputs a narrow range of false positives.

NestBoot improved the accuracy in nearly every case (Figs 3 and 4). The improvement tends to be greater for higher SNR levels. For the 10-gene datasets, the effect of NestBoot increased with SNR for LASSO and LSCO but was almost constant for the other methods. The greatest improvement was seen for LSCO at SNR10, where NestBoot boosted the accuracy from 0.24 to 0.56. For the 45-gene datasets there is a trend towards lower accuracies at low SNR but higher accuracies at high SNR, compared to the 10-gene datasets. Here NestBoot gave the greatest improvement with the LASSO method, boosting the accuracy from 0.19 to 0.80 at SNR 1. NestBoot gave greater improvement for 10-gene datasets than for 45-gene networks at SNR 0.001. However, in the 10-gene networks the overall trend of increasing accuracy with increasing SNR does not start until SNR 1, while 45-gene networks see significant improvement already at SNR 0.01 for LASSO and LSCO.

We are interested in improving performance when inferring networks using experimental datasets. To this end we gathered three biological datasets for which a ‘true network’ has been published. Even though this may not be the completely true network we here use it as a proxy for benchmarking GRN inference. Applied to real data we again see general accuracy improvement by NestBoot (Fig. 5). This benchmark contains data with typical properties of experimentally collected data, i.e. with high noise. The measured SNR levels for these datasets range from 0.004 to 0.01 (see Fig. 5). As demonstrated above, such levels will return low accuracies, but this can be improved substantially wrapping nested bootstrapping around the method, in particular LASSO and LSCO. All methods showed improvement under NestBoot across the biological datasets. The greatest increase in accuracy is seen for the Arrieta-Ortiz data with RNICO, which improved the median MCC from near zero to 0.2.

We note that the performance of both native and NestBoot modes of CLR and GENIE3 are essentially independent of SNR,

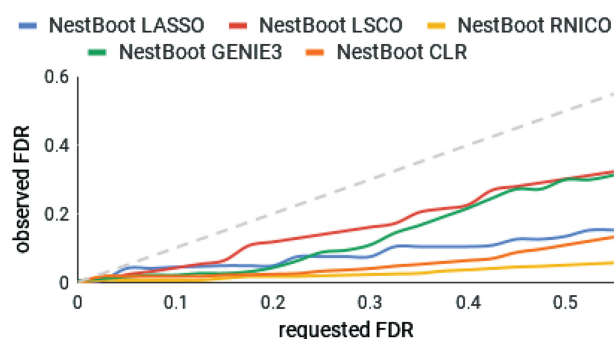


Fig. 6. Observed False Discovery Rate (FDR) at different requested NestBoot FDR settings. Because of the 100-fold nesting applied here, the NestBoot FDR levels very conservative

with the exception of GENIE3 on 45 genes, while the other methods reach far higher accuracies at high SNR. We attribute this to the fact that CLR and GENIE3 do not attempt to solve the inverse problem stated in Eq. 3 but instead use mutual information and tree ensemble building, respectively, without consideration for the experimental design.

Note that while we attempt to gather networks of similar final sparsities across methods, the sparsity is by necessity set differently. RNICO derives sparsity using γ probability cutoffs, CLR, LSCO and GENIE3 using a post-inference cutoff procedure, and LASSO using an L1 penalty. The comparison may therefore not be entirely free from such biases between methods that use different sparsity ladders, but the effect should be small.

In the implemented release of NestBoot the user can modify the FDR cutoff to trade false positives for false negatives. Note that the method is conservative as it demands a link to pass the bootstrap support cutoff of the set FDR in all bootstrap runs. Therefore, the overall false discovery rate in the configuration of 100 runs is considerably lower than the set FDR (see Fig. 6).

4 Discussion and conclusions

We have presented NestBoot, a bootstrap-based approach for GRN inference that is designed to improve the reliability of inferred networks. This is particularly advantageous when applying it to data that is biologically meaningful yet does not follow the U-shaped frequency distribution required by e.g. BINCO, which is often the case. NestBoot is able to control the rate of false positive links independent of regularization parameter choice, which reliably yields inferred networks of biologically realistic sparsity. We analyzed three biological datasets with gold standard GRNs and show that NestBoot improves the GRN inference accuracy when using it with any of the inference methods LASSO, LSCO, RNICO, GENIE3 and CLR. NestBoot was also shown to substantially improve GRN inference accuracy on synthetic datasets containing a varied assortment of data properties.

The improvement is clearly visible when looking at the median MCC (Figs 3, 4 and 5). Links in networks inferred by NestBoot can be relied upon to a high degree as we see the percentage of false positives increases minimally up to FDR 0.5 (Fig. 6). We therefore believe that modest median MCC values are due to false negatives in the NestBoot estimates, while without NestBoot it is more likely to be due to false positives.

Native GENIE3 uses a tree ensemble approach drawing on relationships between input gene expression patterns to predict those of

target genes, building trees based on bootstrapped samples to return one inferred network with ranked link strengths, whereas NestBoot infers an entire network for each bootstrap. Another distinguishing feature of GENIE3 is its design for data with multifactorial (i.e. unknown) perturbations, whereas the regression methods take into account which gene was perturbed explicitly via the perturbation design matrix (P). CLR applies normal distribution statistics to mutual information scores in order to identify network links. These may be reasons why GENIE3 and CLR are unable to capitalize on the more informative data at high SNR.

Most biologically derived datasets contain high noise with SNR typically below 0.01. We have previously demonstrated poor performance of inference methods when operating over such sufficiently uninformative datasets (Tjörnberg *et al.*, 2017), and indeed most native methods yielded a median MCC close to zero for SNR 0.001 and 0.01, especially for the 45-gene datasets. Nested bootstrapping was able to improve the accuracy at low SNR in some cases, but generally works better for high SNR data. There is thus clearly a great need to improve accuracy at ‘biologically relevant’ SNR levels.

The NestBoot algorithm is implemented in MATLAB, as part of the GeneSPIDER package, and as such provides functionality around any inference method incorporated. On 45 genes and 180 samples, Nestboot required roughly 1 core hour utilizing the fastest method tested, LSCO, on a Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, corresponding to 8 min when run as 8 parallel jobs. Taking advantage of this platform, the user can not only infer networks, but benchmark methods and estimate their performance on datasets with desired properties.

Funding

This work was partly supported by a startup grant by the National Cheng Kung University, and a grant from the Ministry of Science and Technology in Taiwan (105-2218-E-006-016-MY2).

Conflict of Interest: none declared.

References

- Arrieta-Ortiz, M.L. *et al.* (2015) An experimentally supported model of the *bacillus subtilis* global transcriptional regulatory network. *Mol. Syst. Biol.*, **11**, 839.
- Efron, B. and Tibshirani, R.J. (1994) *An Introduction to the Bootstrap*. Washington, DC: CRC Press.
- Faith, J.J. *et al.* (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.
- Friedman, J. *et al.* (2009) glmnet: lasso and elastic-net regularized generalized linear models. *R Package Version*, **1**.
- Gardner, T.S. *et al.* (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, **301**, 102–105.
- Greenfield, A. *et al.* (2010) Dream4: combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS One*, **5**, e13397.
- Haury, A.-C. *et al.* (2012) Tigrass: trustful inference of gene regulation using stability selection. *BMC Syst. Biol.*, **6**, 145.
- Hinkley, D. and Shi, S. (1989) Importance sampling and the nested bootstrap. *Biometrika*, **76**, 435–446.
- Huynh-Thu, V.A. *et al.* (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**, e12776–e12710.
- Ideker, T.E. *et al.* (1999). Discovery of regulatory interactions through perturbation: inference and experimental design. In: *Biocomputing 2000*. World Scientific, pp. 305–316.

- Li, S. *et al.* (2013) Bootstrap inference for network construction with an application to a breast cancer microarray study. *Ann. Appl. Stat.*, **7**, 391.
- Lorenz, D.R. *et al.* (2009) A network biology approach to aging in yeast. *Proc. Natl. Acad. Sci. USA*, **106**, 1145–1150.
- MATLAB (2017) 9.1 (r2016b). <https://mathworks.com/products/matlab.html>
- Matthews, B.W. (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta (BBA)-Protein Struct.*, **405**, 442–451.
- Meinshausen, N. and Bühlmann, P. (2010) Stability selection. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)*, **72**, 417–473.
- Nordling, T. E. M. (2013) Robust inference of gene regulatory networks: system properties, variable selection, subnetworks, and design of experiments. Ph.D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden.
- Tegnér, J. and Björkegren, J. (2007) Perturbations to uncover gene networks. *Trends Genet.*, **23**, 34–41.
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)*, **58**, 267–288.
- Tjärnberg, A. *et al.* (2013) Optimal sparsity criteria for network inference. *J. Comput. Biol. J. Comput. Mol. Cell Biol.*, **20**, 398–408.
- Tjärnberg, A. *et al.* (2015) Avoiding pitfalls in L1-regularised inference of gene networks. *Mol. Biosyst.*, **11**, 287–296.
- Tjärnberg, A. *et al.* (2017) Genespider—gene regulatory network inference benchmarking with controlled network and data properties. *Mol. Biosyst.*, **13**, 1304–1312.
- Wang, S. *et al.* (2011) Random lasso. *Ann. Appl. Stat.*, **5**, 468.
- Zhao, P. and Yu, B. (2006) On model selection consistency of lasso. *J. Mach. Learn. Res.*, **7**, 2541–2563.