Bioinformatics, 38(5), 2022, 1312–1319
https://doi.org/10.1093/bioinformatics/btab829
Advance Access Publication Date: 9 December 2021
Original Paper

Gene expression

EnGRaiN: a supervised ensemble learning method for recovery of large-scale gene regulatory networks

Maneesha Aluru[1,*], Harsh Shrivastava[2], Sriram P. Chockalingam [3], Shruti Shivakumar[4] and Srinivas Aluru[3,4,*]

[1]Department of Biology, Georgia Institute of Technology, Atlanta, GA 30308, USA, [2]Microsoft, Redmond, WA 98052, USA, [3]Institute for Data Engineering and Science, Georgia Institute of Technology, Atlanta, GA 30308, USA and [4]Department of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on July 15, 2021; revised on October 29, 2021; editorial decision on November 26, 2021; accepted on December 3, 2021

Abstract

Motivation: Reconstruction of genome-scale networks from gene expression data is an actively studied problem. A wide range of methods that differ between the types of interactions they uncover with varying trade-offs between sensitivity and specificity have been proposed. To leverage benefits of multiple such methods, ensemble network methods that combine predictions from resulting networks have been developed, promising results better than or as good as the individual networks. Perhaps owing to the difficulty in obtaining accurate training examples, these ensemble methods hitherto are unsupervised.

Results: In this article, we introduce *EnGRaiN*, the first supervised ensemble learning method to construct gene networks. The supervision for training is provided by small training datasets of true edge connections (positives) and edges known to be absent (negatives) among gene pairs. We demonstrate the effectiveness of *EnGRaiN* using simulated datasets as well as a curated collection of *Arabidopsis thaliana* datasets we created from microarray datasets available from public repositories. *EnGRaiN* shows better results not only in terms of receiver operating characteristic and PR characteristics for both real and simulated datasets compared with unsupervised methods for ensemble network construction, but also generates networks that can be mined for elucidating complex biological interactions.

Availability and implementation: *EnGRaiN* software and the datasets used in the study are publicly available at the github repository: https://github.com/AluruLab/EnGRaiN.

Contact: aluru@cc.gatech.edu or maneesha.aluru@biology.gatech.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Reverse-engineering gene regulatory networks (GRNs) from gene expression data are a grand challenge problem that facilitates numerous applications in biology including discovery of complex gene interactions and improving gene annotations. Owing to its importance, a wide range of mathematical techniques and computational methods have been proposed. This in turn spurred efforts to establish benchmark datasets and assess quality of the results (Marbach *et al.*, 2012; Pratapa *et al.*, 2020). Such surveys highlighted significant interaction biases, strengths and weakness of different inference methods and underscored the need for ensemble gene networks to improve overall prediction accuracy.

Ensemble networks are constructed by combining complementary gene–gene predictions inferred by several heterogeneous methods into 'community networks'. In a comprehensive review of over 30 such methods, Marbach *et al.*, (2012) show that rank averaging predictions from various methods tend to disadvantage lower ranked predictions inferred by one or a few methods, and consequently generates more robust ensemble networks that perform better than or as good as networks resulting from individual methods. Furthermore, weighted averaging, by assigning higher weights to methods with superior performance on simulated datasets, was shown to provide only marginal improvements.

A recent analysis of unsupervised ensemble methods by Bellot *et al.* (2019) provides a framework for such methods and evaluated eight different approaches using benchmark simulated datasets. In their study, *ScaleLSum* method generated the best results when combining networks built on heterogeneous datasets. *ScaleLSum* performs a neighborhood scaling via local z-score followed by an

aggregation via summation. Neighborhood scaling of an edge between two genes uses only the neighbors of the two genes in the network to normalize the predicted weight of the edge. For the homogeneous case of multiple networks constructed from the same dataset using different GRN methods, both *ScaleSum* and rank averaging performed comparably.

Although ensemble network methods aggregate and improve upon individual methods, current approaches suffer from the following deficiencies: (i) Results are just marginal improvement for real datasets, (ii) For the aggregated predictions to be superior, predictions by the constituent methods are expected to follow strict distribution constraints and (iii) Performance and scalability to construct GRNs by including more methods in the ensemble for larger datasets is often limited.

In this article, we present *EnGRaiN*, the first supervised method to construct ensemble GRNs from large datasets. *EnGRaiN* requires only a small training dataset of positives and negatives (presence/absence of an edge in the true network). Moreover, it does not require a specific distribution of predictions, and is able to produce improved results from fewer GRN methods. We demonstrate the effectiveness of our method with predictions from 15 different network inference methods using both simulated and real genome-scale datasets. Using *EnGRaiN*, we report the construction and analysis of a whole-genome ensemble network of the plant *Arabidopsis thaliana*, created from painstaking curation of heterogeneous microarray datasets from multiple public repositories.

2 Materials and methods

EnGRaiN integrates interaction/co-expression predictions from multiple gene network inference methods to generate a comprehensive ensemble network of gene interactions. The overall workflow we developed for constructing and evaluating ensemble networks is shown in Figure 1.

2.1 The *EnGRaiN* method

2.1.1 Input

Consider '*M*' GRN predictions generated by as many distinct GRN recovery methods, each run independently of the others. GRNs may have edge weights, denoting the confidence level in each predicted edge. The collection of predicted networks is represented by an $|E| \times M$ matrix, where E represents the set of edges such that each is present in at least one input network.

2.1.2 Task

Given the M input networks, the goal is to design an efficient method to combine them into an ensemble network with the following desirable properties:

1. Quality better than or equal to the best input method.
2. Is robust to low performing models and noise.
3. Exhibits runtime and data size scalability.
4. Quality does not degrade if more methods are added to the ensemble.

2.1.3 Supervision

To achieve these properties, we make use of the supervision available from known interactions. We gain access to some ground truth edges of the GRN under consideration or find a 'representative' GRN which can resemble the properties of the GRN at-hand. We collect both positive edges representing those that exist in the true network, and negative edges indicating otherwise. We demonstrate that very few ground truth edges are needed compared with

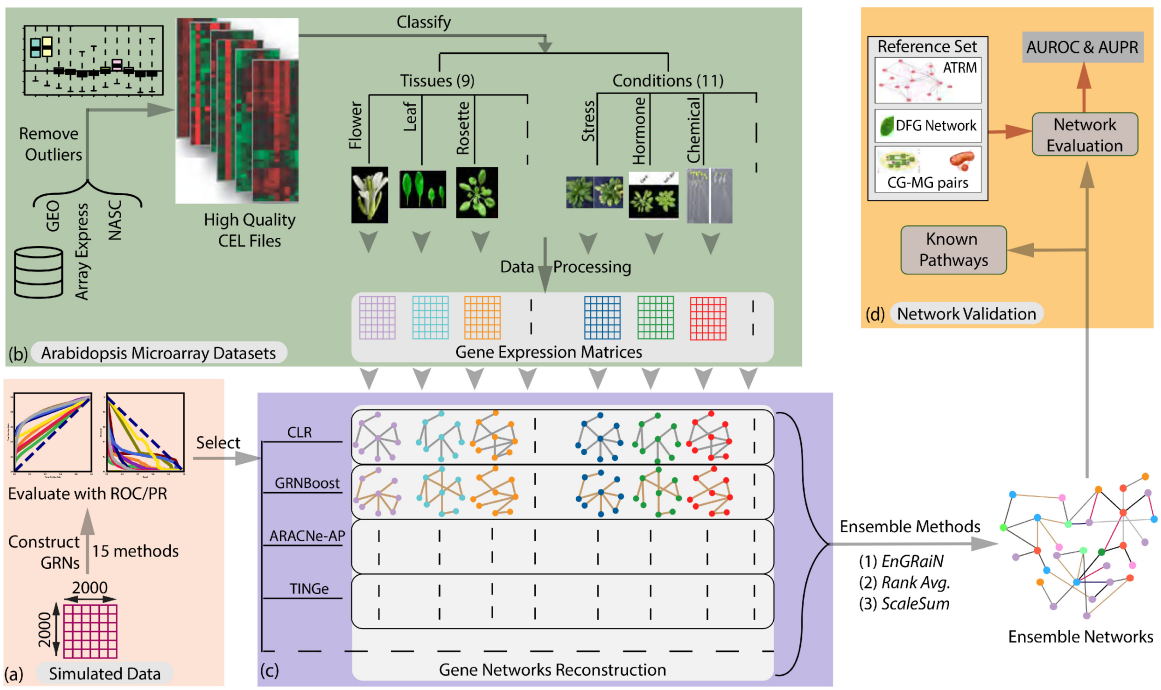

Fig. 1. Overall workflow for constructing and evaluating ensemble gene networks from simulated and real data: (**a**) Evaluation of individual network inference methods with simulated data. Seven different types of gene network inference methods and 15 corresponding software(s) were used in this study. Those that could generate large GRN's and performed better than random chance were used for generating ensemble networks. (**b**) Collection and processing of *A.thaliana* microarray datasets. Microarray data were downloaded from public repositories, subjected to quality control and categorized into tissue and conditions. The classified datasets were normalized, and genes/probesets were filtered using IQR (inter-quartile range) filter and annotated. Raw expression values were converted to gene expression values and log transformed. (**c**) Gene expression matrices from (b) were used to construct gene networks for each tissue and condition using 10 different network inference methods. These were then used as input to generate genome-scale ensemble networks using both unsupervised and supervised ensemble learning methods. (**d**) Network performance was assessed using standard AUROC and AUPR measures, and using experimentally validated biological networks (e.g. *Arabidopsis* ATRM and DFG (dyanamic factor graphs) networks)

the size of the GRNs, making it feasible to develop supervised techniques.

2.1.4 Our model

In *EnGRaiN*, we do not modify the individual methods participating in the ensemble as done in a 'joint training' process (Cheng *et al.*, 2016). Instead, all the networks in the input are created independently by the respective methods without the knowledge of each other. Based on this rationale, we can consider each row (weights of an individual edge) of the input matrix to be independent of the other rows. Using the knowledge of ground truth edges from a partially known GRN as training data, we construct a scalable and effective supervised ensemble model as follows.

Figure 2 gives an overview of our ensemble technique setup. Each row $e \in E$ represents the edge prediction scores for the M methods $\{m_1, m_2, \ldots, m_M\}$, upon which we want to fit a learning model $\mathcal{F}$, s.t.

$$\hat{y}^e = \mathcal{F}(m_1^e, m_2^e, \ldots, m_M^e) \tag{1}$$

Consider 'E_{Tr}', the set of known edges as our training data. We define the following L_2 loss function for learning our model:

$$L = \frac{1}{|E_{Tr}|} \sum_{e \in E_{Tr}} (y^e - \hat{y}^e)^2 \tag{2}$$

where $y^e \in \{0, 1\}$, representing the probability of the occurrence of an edge. In case of imbalanced training data where more negative edges than positive or vice-versa are given, an appropriate cost-sensitive class imbalance handling technique can be used while training, as addressed in some prior works (Bhattacharya *et al.*, 2017; Chawla *et al.*, 2002; Shrivastava *et al.*, 2015).

The supervised learning model $\mathcal{F}$ can be any desired traditional machine learning (ML) or deep learning-based model. Some examples of traditional ML models are support vector machines (SVM), decision tree-based methods like Random Forest and gradient boosting-based methods like XGBoost. A simple multilayer Neural network with input units equal to the number of methods 'M' and the output layer consisting of a single sigmoid unit representing the edge probability $\in [0, 1]$, can also be used in our case as an effective deep learning-based model.

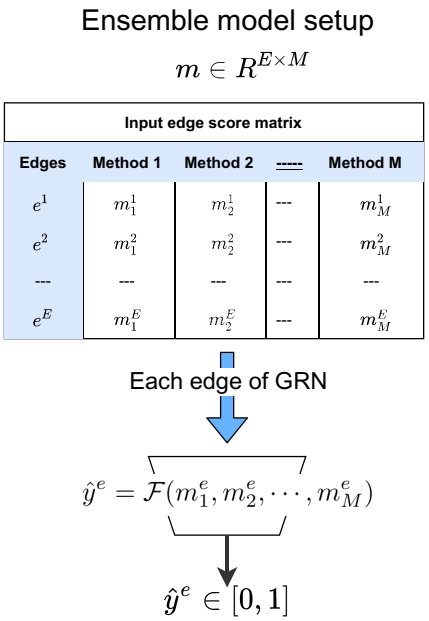

Fig. 2. The model $\mathcal{F}$ can be any appropriate ML- or DL-based model. The input is the row vector containing the edge scores for all the individual methods. The output $\hat{y}^e$ is the probability of the existence of the edge e

In our implementation, we use the sklearn Python pacakge to first normalize input data using the standard scaler, and then construct the ensemble models. In order to construct a robust model that avoids overfitting, we use 10-fold cross validation during training. There are three key benefits of our approach in terms of computational efficiency. First, our model needs significantly *less training data* as we are doing edge-wise predictions. Our ensemble model is able to learn a weighing function over the input methods by using the data of a few thousand edges. Second, this allows us to *scale to millions of edges* in an efficient manner, as each of these edge-wise predictions can be executed in parallel. Last, the edge-wise prediction approach also facilitates fast *inclusion of additional GRN* recovery methods into the *EnGRaiN* framework. Including a new GRN method will add a new column in the input score matrix (Fig. 2). Though retraining is needed, it is not burdensome as the size of our training data is significantly small and the number of methods is <100.

2.2 Datasets

2.2.1 Simulated datasets

Fifteen different network inference methods were evaluated (Table 1) using a yeast simulated dataset of 2000 samples and 2000 genes (Bellot *et al.*, 2015) to assess their quality and scalability. We added local noise and global noise to the dataset following the vignette available with NBM package (Balaji *et al.*, 2006). A brief description of each inference method is provided in Supplementary File S1. Our motivation for using simulated datasets from artificial networks is (i) to evaluate the accuracy of gene networks generated by each method, when a known reference network is available as the 'ground truth', and (ii) to assess whether a given method could scale to large GRNs. For the latter, we randomly selected 15 subsets of the yeast benchmark dataset with varying number of genes/samples in each set, and in increments of 250 genes/samples.

All software were run on a system with four 18 core 2.10 GHz Intel Xeon E7-8870 CPUs and 1TB of main memory. We used a maximum of 64 cores in the system, and a time limit of 24 h for all parallel methods (WGCNA, FastGGM, ARACNe-AP, CLR, TINGe, TIGRESS, Banjo, CATNET, GENIE3, GRNBoost and Inferelator) and 72 h for the sequential methods (PCC, GeneNet, MRNET and iRAFNet). We also developed docker containers for these methods in order to bundle together all the dependencies of the software, and to enable ease of reproducibility (Supplementary Table S1). For fair comparison, only default parameters were used as suggested in relevant papers (Table 1). For those methods that successfully completed their runs, the output of the edge weights of all the possible undirected edges (e.g. 1 999 000 edges for a dataset of 2000 genes) were provided as input for evaluating *EnGRaiN*.

We did not consider directionality of edges because many of the methods (e.g. MI-based methods, correlation-based methods) from which the supervised ensemble is constructed are not capable of predicting directionality of interactions. In order to include such methods in ensemble learning with ENGRAIN, we ignored directionality for methods that can predict it (e.g. GRN Boost, Inferelator, TIGRESS). However, the EnGRain model itself can generate a directional ensemble network provided such information is present in all of the individual networks. We would also need directionality information in the ground truth network for further performance assessment.

2.2.2 *Arabidopsis* microarray datasets

We collected ≈20 000 non-redundant *A.thaliana* microarray datasets from public repositories. These contain information from a whole gamut of tissues, treatments and environmental conditions and hence can be used collectively to generate networks at the whole-genome level. After removal of duplicate CEL files, data were classified into 9 different tissues and 11 different conditions (Table 2). We then processed microarray data according to Aluru *et al.* (2013) and Chockalingam *et al.* (2016). A total of 16 889 CEL files remained after this process (Supplementary Table S2). For matching Affymetrix probesets to corresponding gene identifiers, we

Table 1. List of gene network inference methods

Type of the method	Method/software	Notes/references
Correlation measures	PCC	Pearson correlation coefficient
	WGCNA	Langfelder and Horvath (2008)
Gaussian graphical models	GeneNet	Opgen-Rhein and Strimmer (2007)
	FastGGM	Wang *et al.* (2016)
Information theoretic measures	ARACNe-AP	Lachmann *et al.* (2016)
	CLR	Faith *et al.* (2007)
	MRNET	Meyer *et al.* (2008)
	TINGe	Aluru *et al.* (2013)
Regression models	TIGRESS	Haury *et al.* (2012)
Regression trees	GRNBoost	Aibar *et al.* (2017)
	GENIE3	Huynh-Thu *et al.* (2010)
	iRAFNet	Petralia *et al.* (2015)
Differential equations	Inferelator	Bonneau *et al.* (2006)
Bayesian networks	Banjo	Hartemink (2005)
	CATNET	Salzman and Almudevar (2006)

Table 2. Classification of *Arabidopsis* microarray datasets into tissues and conditions

Tissue/condition	No. CEL files	No. probesets
Flower	920	17 608
Leaf	3564	16 887
Root	2948	17 303
Rosette	1311	17 236
Seed	787	19 120
Seedling (1 week)	2822	16 298
Seedling (2 weeks)	1841	17 003
Shoot	1690	16 164
Whole plant	987	16 930
Chemical	605	17 486
Development	5102	18 373
Hormone (ABA, IAA, GA and BR)	1708	17 753
Hormone (JA, SA and ethylene)	1323	16 714
Light condition	1304	16 414
Nutrient condition	1375	17 766
Stress (light)	596	16 924
Stress (pathogen)	1636	17 483
Stress (salt drought)	935	17 611
Stress (temperature)	1514	16 004
Stress (other)	545	18 854

Note: The number of CEL files and genes/probesets remaining after data normalization and IQR filter are as given.

used the TAIR10 genome annotation. All probesets matching to more than one transcript ID were removed from consideration. Together, we obtained a total of 21 429 probesets, of which 20 463 remained for network construction after processing (Table 2 and Supplementary Table S3). We used the topGO module in R package to confirm tissue/condition-specificity of genes, and selected the top 50 gene ontology terms based on Fisher's exact test (Alexa *et al.*, 2006).

2.3 Reconstruction of *Arabidopsis* ensemble network

We initially reconstructed individual tissue and condition networks with each network inference method from microarray data assigned to that specific classification. Multiple ensemble networks were then constructed with three different methods—*EnGRaiN*, Rank Average and *ScaleLSum* using features (20 tissue/condition networks) from each of the individual methods as input. All software were run on a cluster of 64 nodes connected by EDR Infiniband, with each node having two 2.4 GHz 14-Core Intel E5-2680 V4 processors and 256 GB of main memory and running RedHat Enterprise Linux 7.0 operating system. In order to limit the total time required for reconstructing individual networks from 10 inference algorithms and 20 different datasets (200 networks in total), the runtime was constrained to 3 days for inferring networks with smaller data sizes (<1500 expression profiles, such as stress-light and flower etc.), and 8 days for networks with larger data sizes. For methods that could run in parallel, we used 128 distributed cores for TINGe, 28 shared-memory cores for WGCNA and ARACNe-AP and 8–16 shared memory cores for GRNBoost. We used fewer cores for GRNBoost to avoid 'Out of Memory' errors when run on larger number of cores.

2.4 Performance assessment of ensemble networks

The quality and performance of various network reconstruction methods were evaluated using the receiver operating characteristic (ROC) and precision-recall (PR) curves plotted by comparing reconstructed network(s) against the reference network. We report average and the SD of the 10-fold cross-validated AUROC (area under ROC curve) and AUPR (area under PR curve) measures for both simulated and real datasets. For supervised learning of networks from simulated data, the training and testing dataset consisted of 472 positives and 20 000 negatives and 4724 positives and 200 000 negatives, respectively. In case of the *A.thaliana* gene networks, the AUROC and AUPR measures reflect the average and the SD of values from ten different runs of *EnGRaiN*, each of which used a random subset of interactions from the reference network(s) as training dataset.

2.5 Reference set

Networks constructed from simulated data are compared against the known true network to compute the necessary performance metrics. To evaluate the accuracy of *Arabidopsis* network(s), we used the following two networks as 'ground truths': (i) Arabidopsis Transcriptional Regulatory Map (ATRM) constructed by Jin *et al.* (2015). This was generated from mining of published literature, and hence contains high confidence verified interactions mainly from developmental and stress response processes. It includes a total of 1359 non-redundant regulatory interactions between 388 transcription factors and target genes. (ii) N-response DFG network is a network constructed using dynamic factor graphs with time-series data from Nitrogen-treatment experiments (Brooks *et al.*, 2019). We included the top 295 of the high confidence edges predicted in this network for performance assessment. The interactions from these two given networks can be considered as true positives (TPs), i.e. interactions that are expected to be highly weighted edges in any predicted network. However, for assessing large networks reconstructed from real data, true negatives (TNs) indicating the absence of any interaction between a given pair of genes, are harder to

Table 3. Performance assessment of GRN methods using AUROC and AUPR measures

GRN method	AUROC	AUPR	Runtime(s)
CLR	0.8452	0.4387	535
TINGe	0.8359	0.3749	117
MRNET	0.8312	0.4657	70
PCC	0.8308	0.3552	41
WGCNA	0.8308	0.3552	38
ARACNe-AP	0.8184	0.3822	365
GRNBoost	0.8162	0.4038	423
GENIE3	0.7216	0.3842	3902
TIGRESS	0.6920	0.2999	2925
Inferelator	0.6530	0.3091	499
FastGGM	0.5258	0.0263	518
GeneNet	0.4973	0.0706	139

Note: Runtime reports the time (in seconds) to construct a GRN from a yeast simulated dataset of 2000 samples and 2000 genes.

know. Therefore, we generated a set of 4347 interaction pairs between chloroplast-encoded and mitochondria-encoded genes (Supplementary Table S4), based on the assumption that a direct interaction at the transcriptional level between such genes is expected to be highly unlikely in *A.thaliana* tissues (Woodson and Chory, 2008). We further assume that a robust GRN method would place these 'negatives' at the bottom of their ranked list of predictions.

2.6 Functional validation of *Arabidopsis* ensemble network

We collected a set of genes related to 'photosynthesis', 'cell-wall organization and biogenesis' and 'carbohydrate metabolism' from TAIR (arabidopsis.org) and Araport (araport.org) to assess biological significance of the *Arabidopsis* ensemble network (AEN; Supplementary Table S5). We also downloaded 14 genes related to 'heat stress' and used them as seed genes to run the network analysis tool GeNA (Aluru *et al.*, 2013). GeNA analyzes the ensemble network to rank other genes with respect to the seed genes, and outputs the minimum sized connected component containing the seed genes and the highest ranked genes in relation to these.

2.7 *EnGRaiN* for cross-tissue prediction

Our method is useful in applications beyond generating large ensemble networks. Specifically, our method can extract interactions that are present in a given tissue, based on training data/networks generated from a small collection of other biologically related tissues. For example, we expect a supervised method to predict most or all of the interactions that are present in the 'shoot' tissue based on the interactions learned only from the 'flower' and the 'leaf' tissues.

To setup our experiment, we chose 8 different tissues, namely 'Leaf', 'Flower', 'Root', 'Rosette', 'Shoot', 'Seed', 'Seedling (1 week)' and 'Seedling (2 weeks)'. From these, we prepare different combinations of training and testing subsets for our experiments. For evaluation of these ensembles, we constructed tissue-specific reference subnetworks of TPs based on the reference networks compiled in Section 2.5. Only the highest confidence edges as predicted by the construction methods are included in the tissue-specific reference subnetworks. Note that these reference subnetworks are unique with respect to the positives, while negatives remain as discussed in Section 2.5. We compute the AUROC/AUPR measures for the ensemble networks and compare against those of the individual methods tissue networks.

Table 4. Performance assessment of *EnGRaiN* with different learning models using yeast simulated data

Ensemble method	AUROC	AUPR
XGBoost	**0.8654 (0.0330)**	**0.5808 (0.0599)**
Neural network	0.8568 (0.0383)	0.5521 (0.0515)
Random forest	0.8583 (0.0462)	0.4728 (0.0655)
SVM	0.7845 (0.0273)	0.4703 (0.0609)
Logistic regression	0.7294 (0.1009)	0.4735 (0.0918)

Note: The average (and the SD in parentheses) AUROC and AUPR values of the 10-fold cross-validation are reported with the best performing method highlighted in bold.

Table 5. Performance assessment of ensemble gene networks reconstructed from yeast simulated data

Ensemble method	AUROC	AUPR
Rank average	0.8324	0.3050
ScaleLSum	0.7100	0.0176
ScaleSum	0.7940	0.1503
EnGRaiN	0.8654 (0.0330)	0.5808 (0.0599)
Rank average (top seven)	0.8361	0.4172
ScaleLSum (top seven)	0.7934	0.0265
ScaleSum (top seven)	0.8229	0.1231
EnGRaiN (top seven)	0.8532 (0.0339)	0.5556 (0.0653)

Note: AUROC and AUPR values for the EnGRaiN ensemble network report the average (and the SD in parentheses) of the 10-fold cross-validation. The SD is always zero for unsupervised methods as training data are not required for such methods, and hence they do not produce varying results in the 10-fold cross-validation experiments.

3 Results and discussion

3.1 *In silico* assessment of network inference methods

We selected varied network inference methods which have open-source implementations for quality evaluation with yeast simulated data (Table 1). As our goal is to reverse engineer genome-scale networks of higher organisms (e.g. humans, plants), we also assessed whether a given software could scale to large number of datasets and thousands of genes.

We included several heterogeneous methods in our study to exploit their strengths and diversity of predictions. Of the 15 different inference methods analyzed, 12 were able to infer the yeast network in a relatively short amount of time (Table 3). The two Bayesian methods BANJO and CATNET, as well as iRAFNet, failed to infer even a smaller network of 250 nodes from 2000 samples (Supplementary Table S6). Network quality evaluation measures show that both AUROC and AUPR values for all of the network models except GeneNet and FastGGM are markedly better than by random chance. Nonetheless, we used network predictions from all 12 inference methods to construct the yeast ensemble network with ENGRaiN. As mentioned previously in Section 2.1, the supervised learning model $\mathcal{F}$ can be any appropriate ML-based model. We compared *EnGRaiN*'s performance using XGBoost and four other classifying functions: (i) neural network, (ii) random forest, (ii) SVM and (iv) logistic regression, to assess the performance of these five different models in constructing robust ensemble gene networks. Results shown in Table 4 demonstrate that XGBoost performs better than other supervised learning models. We therefore performed further studies using XGBoost as the classifying function for constructing ensemble gene networks.

To further evaluate *EnGRaiN*'s performance, we generated three other ensemble networks using previously published unsupervised learning methods—the Rank Average, *ScaleLSum* and *ScaleSum* (Bellot *et al.*, 2019) and assessed these networks in comparison to *EnGRaiN*. *EnGRaiN* outperforms all previous individual as well as

Table 6. Comparison of the *A.thaliana* ensemble gene networks generated using unsupervised and supervised ensemble learning methods

Network	No. features	AUROC	AUPR
ARACNe-AP rank average	20	0.5465	0.4366
CLR rank average	20	0.5866	0.4660
GRNBoost rank average	20	0.5547	0.4570
MRNET rank average	20	0.4851	0.4346
TINGe rank average	20	0.5463	0.4364
WGCNA rank average	20	0.5249	0.4487
Rank average of all features	120	0.5334	0.4554
ARACNe-AP *ScaleSum*	20	0.5466	0.4386
CLR *ScaleSum*	20	0.5829	0.4786
GRNBoost *ScaleSum*	20	0.5554	0.4665
MRNET *ScaleSum*	20	0.5377	0.4968
TINGe *ScaleSum*	20	0.5464	0.4376
WGCNA *ScaleSum*	20	0.6553	0.5834
ScaleSum of all features	120	0.5878	0.50691
ARACNe-AP *EnGRaiN*	20	0.5365 (0.0073)	0.4230 (0.0070)
CLR *EnGRaiN*	20	0.9647 (0.0040)	0.9512 (0.0058)
GRNBoost *EnGRaiN*	20	0.5980 (0.0099)	0.4911 (0.0084)
MRNET *EnGRaiN*	20	0.9510 (0.0036)	0.9282 (0.0057)
TINGe *EnGRaiN*	20	0.5303 (0.0061)	0.4140 (0.0071)
WGCNA *EnGRaiN*	20	0.9676 (0.0030)	0.9551 (0.0030)
EnGRaiN with all features	120	0.9806 (0.0082)	0.9719 (0.0108)

Note: Features denote individual tissue and/or condition networks. For the *Arabidopsis* ensemble gene network constructed using *EnGRaiN*, the AUROC and AUPR values reflect the average and the SD of 10 runs. The SD is always zero for unsupervised methods as training data are not required for such methods, and hence they do not produce varying results in the 10-fold cross-validation experiments.

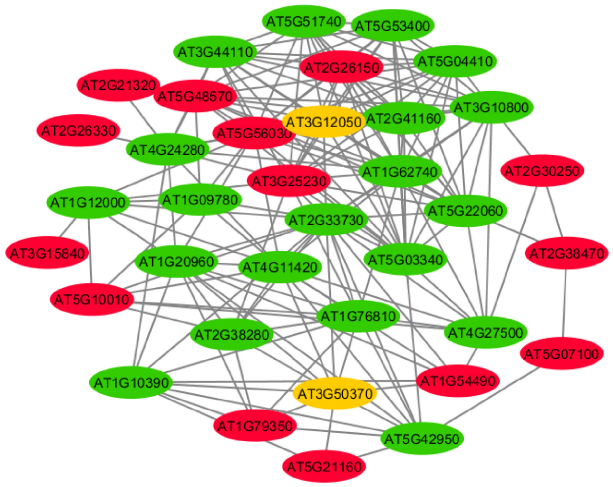

Fig. 3 Functional analysis of the *A.thaliana* ensemble network. A connected subnetwork of 36 genes extracted from the *EnGRaiN* generated AEN by GeNA software. The subnetwork contains 22 genes highly ranked with respect to potential interactions with the 14 seed genes (red) involved in 'heat stress'. Genes shown in green are known to be involved in stress response including temperature stress. Yellow represents genes with unknown function

ensemble methods with respect to both AUROC and AUPR measures (Tables 3 and 5 and Supplementary Table S7). This is also true for ensemble networks generated using predictions from only the top seven best performing methods (Table 3, CLR -> GRNBoost). The Rank averaging or other unsupervised ensemble methods make implicit assumption on the distribution (weights) of their individual methods which limits their performance. On the other hand, *EnGRaiN* uses small ground truth data to learn the underlying distribution over different individual methods and is thus able to identify positives and negatives with significantly better performance. Furthermore, while the AUROC values for networks generated by unsupervised methods are mostly on par with ENGRaiN generated network, their AUPR values are significantly poor. This is because aggregation of predictions from multiple networks for ensemble network generation without supervision can at best provide a mean approximation of the constituent GRNs. With only minimal supervision (given a few TPs and TNs), the EnGRaiN ensemble model shows significantly better AUPR compared with both the individual methods and other unsupervised ensemble methods. Our results also demonstrate that the EnGRaiN method works well even with smaller training data, and eliminates the need for a large number of ground truths for supervised learning. Note that for simulated datasets, AUPR provides a better measure of the network quality because of the imbalance in the underlying network (4724 TPs versus 200 000 TNs).

3.2 Evaluation of the *A.thaliana* ensemble network

We next applied *EnGRaiN* to reconstruct the genome-scale ensemble network of the model plant *A.thaliana*, and to determine its performance on real-world data. A standard practice when reverse-engineering gene networks is to apply network reconstruction methods to the entire compendium of microarray datasets, whether small or large. However, our studies (Chockalingam *et al.*, 2016), along with others (Hurley *et al.*, 2012) suggest that merely increasing the number of samples does not necessarily improve network performance. One key reason for this failure could be the spatiotemporal expression of genes. As the gene expression profiles for network construction are derived from many different tissues and conditions, those genes that are specifically expressed at certain times or cellular states are perhaps filtered from the network model as a result of the 'averaging out' effect which occurs during collective analysis of the entire dataset. Therefore, we first separated datasets based on different tissues and conditions and subsequently reconstructed individual networks from the corresponding datasets. Classification enabled us to capture more context-specific gene expression patterns and gene co-expressions (Supplementary Table S8). For instance, genes that were highly expressed in the flower dataset, but not in the root dataset, are enriched for the terms floral organ development (P-value = 8.8×10^{-3}), floral whorl development (P-value = 7.7×10^{-4}) and pigment biosynthesis (P-value = 1.8×10^{-7}), whereas in the case of root dataset enriched terms include root only nitrate transport (P-value = 2.4×10^{-14}), response to nitrate (P-value = 1.8×10^{-14}) and response to auxin (P-value = 1.9×10^{-4}) etc.

Table 7. Cross-tissue GRN comparison based on AUROC and AUPR

	Only flower	Only leaf	Only SD (1 week)	Flower and leaf	Flower and root	Flower and SD (1 week)
Leaf	✓	Tr	☒	Tr	☒	✓
Flower	Tr	☒	☒	Tr	Tr	Tr
Root	✓	☒	✓	✓	Tr	✓
Rosette	✓	☒	☒	✓	✓	✓
Shoot	☒	☒	✓	✓	✓	✓
Seed	✓	☒	☒	✓	✓	✓
SD (1 week)	✓	✓	Tr	✓	✓	Tr
SD (2 weeks)	✓	✓	✓	✓	✓	✓

Note: Rows list various tissues under consideration. 'Seedling' is abbreviated as 'Sd'. Each column describes the setting of an experiment. For instance, column 5 specifies that training (indicated by 'Tr') was done on the combined data of 'Flower' and 'Root' tissues and the remaining tissues were used for testing. A ✓ indicates that EnGRaiN outperformed every individual GRN recovery method, and the denotes an individual method performed better (e.g., 'Leaf' row in column 5).

Only six methods—ARACNe-AP, CLR, GRNBoost, MRNET, TINGe and WGCNA were able to successfully infer large networks within the time constraints. A total of 49.02, 7.23, 20.83, 4.79 days, 5.09 and 2.86 h, respectively, was required to construct all the 20 tissue/condition networks for each method. GeneNet was excluded due to its poor performance, and PCC although similar in performance to WGCNA, was also excluded as it takes a slightly longer runtime when compared with WGCNA. FastGGM, TIGRESS, GENIE3 and Inferlator were not able to complete their runs even for the smallest of the datasets.

We constructed multiple AENs using three ensemble learning methods—Rank Average, *ScaleSum* and *EnGRaiN* (Table 6). Due to the limited number of positives and negatives in the reference network, we evaluated the reference network with respect only to the induced subnetwork of the genome-scale ensemble networks. Since some of the edges included in the subnetwork have no other neighbors, the *ScaleLSum* computations are not applicable. These networks included predictions from either 20 tissue/condition networks, i.e. 20 features for each individual method, or from all 120 (6×20) networks generated by the six individual inference methods. Similar to the results from simulated data, *EnGRaiN* ensemble network(s) outperform both Rank Average- and *ScaleSum*-generated networks, with the exception of those constructed using ARACNe-AP and TINGe. Furthermore, a comparison of all ensemble networks shows that the *EnGRaiN* network generated from combining all 120 features is the best performing network. Note that for *A.thaliana* networks, AUROC and AUPR values are closer to each other in magnitude because the reference network is relatively balanced (1347 positives and 2233 negatives). *EnGRaiN* leverages the ground truth to learn optimal distribution over its various features, which explains its superior performance. Learning also helps it in extracting useful information from features which are not good predictors if considered as a standalone input.

Although for this study genome-scale ensemble gene networks were constructed from bulk microarray data, the *EnGRaiN* supervised learning model can also be applied to GRNs constructed from bulk RNA-seq data or single-cell data. With single-cell data however, known cell-type specific 'ground truth' interactions are not as readily available for supervised learning as for whole tissues or organisms. Therefore, the method might be less effective for constructing ensemble GRNs from single-cell data than doing so from bulk data.

3.3 AEN is functionally modular

Co-expression networks are built on the 'guilt by association' principle, and one feature of network architecture is that the strength of interactions vary considerably between individual gene pairs. Nevertheless, genes involved in similar biological processes are expected to have higher interaction strength and cluster closely in the network. Therefore, to assess biological relevance of the AEN (Supplementary Table S9), we collected a set of genes mediating three well-known biological processes/pathways from TAIR and determined the nature of their associations in the AEN (Supplementary Table S5 and Fig. S1). Of the 128 genes involved in the 'photosynthesis' process, 123 are closely connected in the AEN and form a single connected component. The remaining five genes are connected via six additional node(s), many of which also function in photosynthesis-related processes. Similarly, 238 of the 275 genes belonging to the 'cell-wall organization and biogenesis', and 305 of the 351 genes from 'carbohydrate metabolism' show direct connections forming coherent modules. We further applied GeNA using 14 seed genes related to 'heat stress' and extracted a connected subnetwork of 36 genes to determine which of the other genes in the subnetwork show higher interaction strength (ranked highly) with respect to the 14 seed genes (Fig. 3). Of the 22 additional genes in the subnetwork, 20 genes are involved in response to temperature stress, and other stress responses (Boyko *et al.*, 2010; Krishnakumar *et al.*, 2015; Omidbakhshfard *et al.*, 2012; Staiger and Brown, 2013; Zhang *et al.*, 2017). The remaining two (AT3G12050 and AT3G50370) genes have no known function, and thus form potential candidates for hypothesis testing. Taken together, our results show significant functional modularity of the AEN.

3.4 Cross-tissue comparison

Table 7 shows performance of the *EnGRaiN* ensemble model on different training/testing combinations of various tissues. This helps identify certain key tissues whose GRN can be used as training data for the ensemble model in order to recover the underlying GRNs for the other tissues. Additionally, these experiments are useful to evaluate the performance of the ensemble model. We consider the *EnGRaiN* ensemble network to be unsatisfactory (denoted by ☒) if it does not perform better or equivalent to the best performing model in its ensemble.

As expected, the *EnGRaiN* model constructed from only one tissue [either 'Flower' or 'Leaf' or 'Seedling (1 week)'] does not generalize well to the entire spectrum of tissues. However, ensemble models constructed from pairs of tissues such as 'Flower and Leaf' and 'Flower and Seedling (1 week)', are able to effectively recover the underlying GRNs of all the other tissues. Interestingly, the *EnGRaiN* model of 'Flower and Root' covers all the tissues except for the 'Leaf'. We speculate that this is due to significant differences between the physiological processes of the respective tissues.

4 Conclusions

The primary contributions of this work are the design of the supervised ML ensemble network method *EnGRaiN*, its validation against prior ensemble as well as standalone methods, end-to-end methodology to reconstruct robust genome-scale networks ranging from data classification to ensemble network generation with the plant model *A.thaliana* as case-study, and the usage of such

networks to predict gene function. *EnGRaiN* is the first supervised ensemble network learning method, to which we credit its quality advantages over the prevailing unsupervised methods. We presented techniques to generate training datasets through prior biological knowledge of known interactions (positive examples), estimating extremely unlikely interactions through domain knowledge (negative examples), and utilizing *EnGRaiN* predicted gene networks themselves as training data when appropriate. Taken together, we expect this work to facilitate improvement in overall accuracy of interaction predictions and analysis of large genome-scale gene networks.

Data Avaliability

The data underlying this article are available in Zenodo at https://doi.org/10.5281/zenodo.5772927.

Funding

This work was supported in part by the National Science Foundation under [IIS-1841351].

Conflict of Interest: none declared.

References

Aibar,S. *et al.* (2017) SCENIC: single-cell regulatory network inference and clustering. *Nat. Methods*, **14**, 1083–1086.

Alexa,A. *et al.* (2006) Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, **22**, 1600–1607.

Aluru,M. *et al.* (2013) Reverse engineering and analysis of large genome-scale gene networks. *Nucleic Acids Res.*, **41**, e24.

Balaji,S. *et al.* (2006) Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *J. Mol. Biol.*, **360**, 213–227.

Bellot,P. *et al.* (2015) NetBenchmark: a bioconductor package for reproducible benchmarks of gene regulatory network inference. *BMC Bioinformatics*, **16**, 1–15.

Bellot,P. *et al.* (2019). Unsupervised GRN ensemble. In: *Gene Regulatory Networks*.Edited by Sanguinetti, Guido and Huynh-Thu, Vân Anh. Springer, New York, NY, pp. 283–302.

Bhattacharya,S. *et al.* (2017). ICU mortality prediction: a classification algorithm for imbalanced datasets. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. AAAI Press, Palo Alto, California USA.

Bonneau,R. *et al.* (2006) The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.*, **7**, R36.

Boyko,A. *et al.* (2010) Transgenerational adaptation of *Arabidopsis* to stress requires DNA methylation and the function of Dicer-like proteins. *PLoS One*, **5**, e9514.

Brooks,M.D. *et al.* (2019) Network Walking charts transcriptional dynamics of nitrogen signaling by integrating validated and predicted genome-wide interactions. *Nat. Commun.*, **10**, 1–13.

Chawla,N.V. *et al.* (2002) SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, **16**, 321–357.

Cheng,H.-T. *et al.* (2016). Wide & deep learning for recommender systems. In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Association for Computing Machinery, New York, NY USA, pp. 7–10.

Chockalingam,S. *et al.* (2016) Microarray data processing techniques for genome-scale network inference from large public repositories. *Microarrays*, **5**, 23.

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.

Hartemink,A.J. (2005) Reverse engineering gene regulatory networks. *Nat. Biotechnol.*, **23**, 554–555.

Haury,A.-C. *et al.* (2012) TIGRESS: trustful inference of gene regulation using stability selection. *BMC Syst. Biol.*, **6**, 145.

Hurley,D. *et al.* (2012) Gene network inference and visualization tools for biologists: application to new human transcriptome datasets. *Nucleic Acids Res.*, **40**, 2377–2398.

Huynh-Thu,V.A. *et al.* (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**, e12776.

Jin,J. *et al.* (2015) An *Arabidopsis* Transcriptional Regulatory Map reveals distinct functional and evolutionary features of novel transcription factors. *Mol. Biol. Evol.*, **32**, 1767–1773.

Krishnakumar,V. *et al.* (2015) Araport: the *Arabidopsis* information portal. *Nucleic Acids Res.*, **43**, D1003–D1009.

Lachmann,A. *et al.* (2016) ARACNe-AP: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics*, **32**, 2233–2235.

Langfelder,P. and Horvath,S. (2008) WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, **9**, 1–13.

Marbach,D. *et al.*; DREAM5 Consortium. (2012) Wisdom of crowds for robust gene network inference. *Nat. Methods*, **9**, 796–804.

Meyer,P.E. *et al.* (2008) minet: AR/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, **9**, 1–10.

Omidbakhshfard,M.A. *et al.* (2012) Effect of salt stress on genes encoding translation-associated proteins in *Arabidopsis thaliana*. *Plant Signal. Behav.*, **7**, 1095–1102.

Opgen-Rhein,R. and Strimmer,K. (2007) From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Syst. Biol.*, **1**, 37–10.

Petralia,F. *et al.* (2015) Integrative random forest for gene regulatory network inference. *Bioinformatics*, **31**, i197–i205.

Pratapa,A. *et al.* (2020) Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods*, **17**, 147–154.

Salzman,P. and Almudevar,A. (2006) Using complexity for the estimation of Bayesian networks. *Stat. Appl. Genet. Mol. Biol.*, **5**,

Shrivastava,H. *et al.* (2015). Classification with imbalance: A similarity-based method for predicting respiratory failure. In: Proceedings of *the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE Computer Society, Washington, DC USA, pp. 707–714.

Staiger,D. and Brown,J.W. (2013) Alternative splicing at the intersection of biological timing, development, and stress responses. *Plant Cell*, **25**, 3640–3656.

Wang,T. *et al.* (2016) FastGGM: an efficient algorithm for the inference of Gaussian graphical model in biological networks. *PLoS Comput. Biol.*, **12**, e1004755.

Woodson,J.D., and Chory,J. (2008) Coordination of gene expression between organellar and nuclear genomes. *Nat. Rev. Genet.*, **9**, 383–395.

Zhang,L. *et al.* (2017) Mutations in eIF5B confer thermosensitive and pleiotropic phenotypes via translation defects in *Arabidopsis thaliana*. *Plant Cell*, **29**, 1952–1969.