Developing machine learning tools to understand transcriptional regulation in plants

Qi Song

Dissertation submitted to the faculty of the
Virginia Polytechnic and State University
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Genetics, Bioinformatics and Computational Biology

Song Li, Chair
Ruth Grene
Lenwood S. Heath
David Haak
Liqing Zhang

June 30, 2019
Blacksburg, Virginia

Keywords: Arabidopsis, regulatory network, machine learning, plant genomics

Developing machine learning tools to understand transcriptional regulation in plants

Qi Song

ABSTRACT

Abiotic stresses constitute a major category of stresses that negatively impact plant growth and development. It is important to understand how plants cope with environmental stresses and reprogram gene responses which in turn confers stress tolerance. Recent advances of genomic technologies have led to the generation of much genomic data for the model plant, Arabidopsis. To understand gene responses activated by specific external stress signals, these large-scale data sets need to be analyzed to generate new insight of gene functions in stress responses. This poses new computational challenges of mining gene associations and reconstructing regulatory interactions from large-scale data sets. In this dissertation, several computational tools were developed to address the challenges. In Chapter 2, ConSReg was developed to infer condition-specific regulatory interactions and prioritize transcription factors (TFs) that are likely to play condition specific regulatory roles. Comprehensive investigation was performed to optimize the performance of ConSReg and a systematic recovery of nitrogen response TFs was performed to evaluate ConSReg. In Chapter 3, CoReg was developed to infer co-regulation between genes, using only regulatory networks as input. CoReg was compared to other computational methods and the results showed that CoReg outperformed other methods. CoReg was further applied to identified modules in regulatory network generated from DAP-seq (DNA affinity purification sequencing). Using a large expression dataset generated under many abiotic stress treatments, many regulatory modules with common regulatory edges were found to be highly co-expressed, suggesting that target modules are structurally stable modules under abiotic stress conditions. In Chapter 4, exploratory analysis was performed to classify cell types for Arabidopsis root single cell RNA-seq data. This is a first step towards construction of a cell-type-specific regulatory network for Arabidopsis root cells, which is important for improving current understanding of stress response.

Developing machine learning tools to understand transcriptional regulation in plants

Qi Song

GENERAL ABSTRACT

Abiotic stresses constitute a major category of stresses that negatively impact plant growth and development. It is important to understand how plants cope with environmental stresses and reprogram gene responses which in turn confers stress tolerance to plants. Genomics technology has been used in past decade to generate gene expression data under different abiotic stresses for the model plant, Arabidopsis. Recent new genomic technologies, such as DAP-seq, have generated large scale regulatory maps that provide information regarding which gene has the potential to regulate other genes in the genome. However, this technology does not provide context specific interactions. It is unknown which transcription factor can regulate which gene under a specific abiotic stress condition. To address this challenge, several computational tools were developed to identify regulatory interactions and co-regulating genes for stress response. In addition, using single cell RNA-seq data generated from the model plant organism Arabidopsis, preliminary analysis was performed to build model that classifies Arabidopsis root cell types. This analysis is the first step towards the ultimate goal of constructing cell-type-specific regulatory network for Arabidopsis, which is important for improving current understanding of stress response in plants.

To my parents, who have always been supportive of my decisions to pursue science and everything else in my life

To my friends at Virginia Tech, with whom I share many moments of happiness that helped me go through hard times in life

Acknowledgements

This dissertation cannot be completed without the support from many people at Virginia Tech. I started my Ph.D. program in 2015 under the guidance of my Ph.D. advisor, Dr. Song Li, who has kindly supported me throughout four years of Ph.D. training. Dr. Song Li has extensive experiences in Plant genomics and bioinformatics, as well as insightful thoughts in biology which usually lead to brilliant research ideas. Our meetings and conversations often inspired me a lot and resulted in exciting new ideas. As a young researcher in his early career as principal investigator, Dr. Song Li is always willing to share his thoughts about transformation from a graduate student to an independent researcher, which benefited me in many ways. I would like to thank Dr. Song Li for all his supports and guidance in four years of my Ph.D. training.

I am grateful to Dr. Ruth Grene and Dr. Lenwood Heath, who have provided many valuable suggestions not only for basic scientific discussions of the manuscripts, but also for general skills such as research presentation and English writing. In particular, with her great expertise and broad knowledge in Plant biology, Dr. Ruth Grene always guided me to improve my skills to clearly explain bioinformatics to experimental biologists and think critically about results generated from computational tools. Dr. Lenwood Heath, as an experienced computer scientist, has guided me in the process of formulating biology question as computational problem. Both Dr. Grene and Dr. Heath have corrected many of my mistakes in English writing, which is of great help to a non-native speaker for English and a graduate student who needs to interact and communicate with many other researchers in United States.

I would like also thank Dr. David Haak and Dr. Liqing Zhang, who have provided many helpful feedbacks for my projects and research presentations.

I would like to thank my lab mate Jiyoung Lee and my roomates Amogh Jalihal and Brittany Boribong. All of us joined GBCB program and became good friends.

I would like to thank GBCB program liaison Dennie Munson, who has provided great support for all graduate students in GBCB program. With her efforts, GBCB program has become a big family where people all support each other and share wonderful ideas.

Qi Song
June, 18
Blacksburg, VA

List of abbreviations

ABA	**Ab**scisic **A**cid
ABRE	**Ab**scisic Acid **R**esponsive **E**lement
ACC	Accuracy
ampDAP-seq	PCR-**amp**lified **DAP-seq**
AREB	**A**B**RE** (abscisic acid-responsive element)-**B**inding Protein
ATAC-seq	Assay for Transposase-Accessible Chromatin using sequencing
ATHB	*Arabidopsis Thaliana* Homeobox
ATWRKY	Arabidopsis Thaliana WRKY
AUC	Area Under Curve
AUC-PRC	Area Under Curve of Precision Recall Curve
AUC-ROC	Area Under Curve of Receiver Operating Characteristic curve
BBX	B-Box
BED	Browser Extensible Data
BLH	BEL1 Like Homeobox
bZIP	Basic Leucine Zipper
CC	Companion Cell
CCA	Canonical Correlation Analysis
CDPKs	Calcium-Dependent Protein Kinase
ChIP-seq	Chromatin Immunoprecipitation Sequencing
CHX	Cycloheximide
ConSReg	**Con**dition **S**pecific **Reg**ulation
CoReg	**Co**-regulatory gene **Reg**ulation
DAE	Denoising Autoencoder
DAP-seq	DNA Affinity Purification Sequencing
DEG	Differentially Expressed Gene
DHS	DNaseI hypersensitive sites
DIV	DIVARICATA

DNase-seq	Deoxyribonuclease I hypersensitive sites sequencing
DNN	Deep Neural Network
DR	Down-Regulated
DREB	Dehydration-Responsive Element-Binding
E. coli	Escherichia coli
EB	Edge Betweenness
EBI	European Bioinformatics Institute
EN	Elastic Net
ENCODE	Encyclopedia of DNA Elements
ERF	Ethylene Reponse Factor
EV	Empty Vector
eY1H	ehanced Yeast-one-Hybrid
GEO	Gene Expression Omnibus
GRN	Genetic Regulatory Network
GRRF	Guided Regularized Random Forest
H. sapiens	*Homo sapiens*
HB	Homeobox
HD-ZIP	Homeodomain-leucine Zipper
ICI	Index of Cell Identity
JA	Jasmonic Acid
KNN	K-Nearest Neighbors
LARS	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LEGs	Low-Expressed Genes
LGLASSO	Logistic regression with Group LASSO penalty
LP	Label Propagation
LREN	Logistic Regression with Elastic Net penalty
LRLASSO	Logistic Regression with LASSO penalty
LRPCC	Logistic Regression with Pearson Correlation Coefficient
LSVM	Linear Support Vector Machine
MAP	Mean Average Precision

MAPK	Mitogen Activated Protein Kinase
MI	Mutual Information
MNase-seq	Micrococcal Nuclease sequencing
MYB	**My**elo**b**lastosis
N	Nitrogen
NAC	Petunia No Apical Meristem (**N**AM), Arabidopsis transcription activation factors (**A**TAF1 and **A**TAF2) and **C**up-shaped cotyledon 2 (CUC2)
NDEGs	Non-Significantly Differentially Expressed Genes
NF-YC	Nuclear Factor Y, subunit C
NMI	Normalized Mutual Information
NN	Neural Network
PBM	protein binding microarray
PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
PGSIP	Plant Glycogenin-like Starch Initiation Protein
PHB	Polyhydroxybutyrate
PRC	Precision Recall Curve
RF	Random Forest
RN	Relevance Network
ROC	Receiver Operating Characteristic
RRS	rewiring recall scores
SA	Salicylic Acid
SCR	SCARECROW
scRNA-seq	single cell RNA-seq
SVM	Support Vector Machine
TARGET	Transient Assay Reporting Genome-wide Effects of Transcription factors
TF	Transcription Factor
TGA	TGACGTCA cis-element-binding protein
T-spm	Thermospermine oxidase

TSS	Transcription Start Site
UDGs	Undetected Genes
UR	Up-Regulated
UV-B	Ultraviolet-B
VRN	Vernalization gene
WRKY	WRKY is named by its core motif sequence 'WRKYGQK'
WT	Walk Trap
Y1H	Yeast-One Hybrid
ZAT	Zinc finger of *Arabidopsis Thaliana*
ZFHD	Zinc Finger-Homeodomain

Table of contents

1. Chapter 1. Introduction

1.1 Significance of study for plant abiotic stress

Plants could suffer from many abiotic stresses during development and reproduction [1]. Common abiotic stresses such as drought, heat, cold, and high salinity can significantly impact plant growth [1,2]. As a result, abiotic stresses are estimated to account for more than 50 % of annual yield loss for major crop species [2]. Improving stress tolerance for plants can potentially improve crop yield and address current challenges of food shortage to meet the need of a growing population [3]. Understanding how plants cope with environmental stresses is a crucial first step towards reprograming gene responses to stress, which in turn confers stress tolerance to plants.

Abiotic stress responses in plants are mediated through various signal transduction pathways [2]. External stress signals were first captured by sensor molecules located in cell wall or membrane and then will be passed to second messengers (e.g. calcium ions) [2]. Next, second messengers can initiate various signaling pathways including mitogen activated protein kinases (MAPKs) pathway and calcium-dependent protein kinases (CDPKs) pathway [2]. This eventually leads to activation of transcription factors (TFs) that can either activate or suppress expressions of stress response genes [2].

1.2 Transcriptional regulation for plant abiotic stress response

Gene expression is regulated through binding of TFs to cis-elements located close to the transcription start sites (TSS) of each individual gene. This binding event is also generally referred to as gene regulation, which has been shown to play an essential role in modulating complex pathways of stress responses in plants [4–6]. Several TF families have been well characterized due to their prominent role in regulating plant stress responses. These TF families include AREB/ABRE (ABRE (abscisic acid – responsive element)-binding protein / abscisic acid responsive element), DREB (dehydration-responsive element-binding protein), MYB (**my**elo**b**lastosis), NAC (Petunia No Apical Meristem (**N**AM), Arabidopsis transcription activation factors (**A**TAF1 and **A**TAF2), and **C**up-shaped cotyledon 2 (CUC2) [7 8–10]. Notably, abscisic acid (ABA) plays a central role in integrating stress responses mediated by multiple TF families. ABA

accumulates rapidly in response to an adverse environment, which induces expressions of stress related genes [5,11]. Signal transduction pathways orchestrated by TFs can initiate stress responses in either an ABA-dependent or ABA-independent manner [8]. For example, AREBs/ABREs [8–10,12], MYBs [12], and NACs [8–10,12] are engaged in ABA-mediated stress responses while DREBs function in an ABA-independent manner [8–10,13]. However, condition-specific activation of the stress response TFs has not yet been systematically explored. Such study can contribute to an improved understanding of condition-specific signaling pathways for abiotic stress response.

1.3 Computational challenges

To address the challenge of identifying gene regulation, large-scale genomic data sets are needed to build computational tools. In recent years, efforts have been made to investigate plant gene regulation using different experimental platforms, including yeast-one hybrid (Y1H) [14], Chromatin Immunoprecipitation Sequencing (ChIP-seq) [15], DNA affinity purification sequencing (DAP-seq) [16] and Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq) [17]. These studies have provided a plethora of data sets to characterize stress response gene regulation.

Moreover, mining associations among multiple genes is another important step towards an improved understanding of stress specific regulatory mechanisms. Using published large-scale data sets, one can easily construct a regulatory network, in which each interaction between TF and target gene is represented as a directed edge. A Network can be decomposed into different modules and a module is defined as a group of genes that share similar properties [18]. These properties should reflect the associations among the genes. The computational challenge is that, given a regulatory network, how to identify regulatory modules of genes that share similar properties.

1.4 Overall objective of this dissertation and organization of chapters

The overall objective of this dissertation is to develop computational tools to systematically characterize gene regulation, gene associations using different types of plant genomic data. In particular, three research questions were investigated and the

corresponding computational tools were developed. These research questions include: (1) How to characterize gene regulation and prioritize important regulators for specific stress? (2) How genes can be associated with each other in stress responses? (3) How can gene expressions be used to distinguish plant cell types and how to infer cell-type-and-stress-specific gene regulation? To answer these research questions, this dissertation is structured into five chapters, in which Chapter 1 (this chapter) briefly introduce the background and Chapter 2 through Chapter 4 present and discuss results of each research project and Chapter 5 summarizes the conclusions of this dissertation. The main focuses of each chapter are briefly summarized below.

1.4.1 Chapter 2. Prediction of condition specific gene regulation using integrated genomic data

DAP-seq is a recently invented experimental technique that performs genome-wide detection of binding sites for TFs [19]. There have been several successful applications of DAP-seq in the field of plant genomics [20–24]. Compared to ChIP-seq, which is another commonly used *in vivo* assay, DAP-seq is an *in vitro* technique that allows for fast generation of genome-wide binding sites for hundreds of TFs. The first published study utilizing DAP-seq has generated over two million binding sites for 387 Arabidopsis TFs. This large dataset has provided a comprehensive regulatory network of potential TF-target interactions. However, this huge regulatory network can have very high number of false positive interactions for any specific environmental perturbation, since interactions are detected in vitro. Particularly, the authors mentioned in the published study that a lack of information from chromatin accessibility and histone modification may have limited the accuracy of the DAP-seq assay [19]. To address this issue, integrating DAP-seq binding sites with other data types that encode condition-specific information is recommended [19]. Chapter 2 discusses integration of DAP-seq data with ATAC-seq and RNA-seq/microarray data. A computational tool, ConSReg (**Con**dition **S**pecific **Reg**ulation) has been developed to infer interactions between transcription factors and target genes. ConSReg was evaluated using expression data sets collected from 22 publications and was compared to other computational tools including TF2Network, PlantPAN 3.0, and Cistome in a study of nitrogen (N) response TFs.

1.4.2 Chapter 3. Identification of co-regulatory modules in genome scale transcription regulatory networks

This chapter explores computational methods to mine gene-gene association from regulatory networks. As discussed previously, recent publications have contributed many large-scale data sets for TF-target interactions. One challenge is how to discover associations among the genes and how can this information be used to understand regulatory processes for stress response. This chapter focuses on the discovery of co-regulating genes from regulatory network since co-regulation has been shown to be one important gene regulatory mechanism [25]. Chapter 3 presents a computational tool, CoReg (**Co**-regulatory gene **Reg**ulation), which was developed for this task. CoReg was evaluated and compared to other module-finding approaches including walk trap (WT), edge betweenness (EB), and label propagation (LP), using simulated networks and real networks. CoReg was applied to large-scale regulatory network of Arabidopsis and many co-regulatory modules were identified. The results showed expression levels of genes in some of the modules are highly correlated under abiotic stress conditions.

1.4.3 Chapter 4. Identification of cell types for plant single cell RNA-seq data

Single cell RNA-seq (scRNA-seq) is one of the latest technological advances in plant genomics. Compared to conventional bulk RNA-seq, this technique can reveal gene expressions in a single cell and enable visualization of time-course trajectory of cell type differentiation in a population of single cells. Compared to bulk RNA-seq data, scRNA-seq data can better characterize heterogenous cell populations, which may be difficult to be detected using bulk RNA-seq [26,27]. Another important feature of scRNA-seq is the ability to profile transcriptomes of cells in transitional states [26]. Identification of cell types for scRNA-seq data is critical to better understand cell-type-specific abiotic stress responses [28]. This also highlights the importance of constructing cell type specific regulatory networks from scRNA-seq data. Application of several types of deep neural network (DNN) to cell type identification has been discussed in a published study, using mouse scRNA-seq dataset [29]. In chapter 4, exploratory analyses were performed to build DNN models for identification of cell types using plant scRNA-seq data. This is a first

step towards the goal of constructing single cell regulatory network, which will be explored in future work.

2. Chapter 2. Prediction of regulatory maps in Arabidopsis using integrated genomic data

Qi Song, Jiyoung Lee, Shamima Akter, Ruth Grene, Song Li.

Abstract

Recent advances in genomic technologies have generated data for large-scale protein-DNA interactions and open chromatic regions for multiple plant species. Defining condition specific functions of transcription factors using these genome wide data has become a major challenge in plant genomic research. To prioritize candidate regulatory genes under any experimental conditions, we have developed a **Con**dition **S**pecific **Reg**ulatory network inference engine (**ConSReg**), which combines genomic data using several machine learning methods followed by feature selection and stability selection to select key regulatory genes. Using Arabidopsis as a model system, we constructed maps of gene regulation for over 50 experimental conditions including abiotic stresses, cell type-specific expression, and stress responses in individual cell types. ConSReg accurately predicted gene expressions (average auROC of 0.84) across multiple testing data sets. We found that including open chromatin information from ATAC-seq data significantly improves the performance of ConSReg across all tested data sets. We also found that the choice of negative training samples and length of promoter regions are two key factors that affect model performance. To validate the performance of our prioritized candidate genes, we analyzed an independent dataset related to plant nitrogen (N) responses. ConSReg provided better rankings for 17 nitrogen response TFs compared to published enrichment-based approach. We applied ConSReg to Arabidopsis single cell RNA-seq data of two root cell types (endodermis and cortex) and identified five regulators in two root cell types. Four out of the five regulators have additional experimental evidence to support their roles in regulating gene expression in Arabidopsis roots.

Keywords

Regulatory network inference; DAP-seq; ATAC-seq; integration

2.1 Introduction

Over the past decades, thousands of expression profiles have been generated using either RNA-seq or microarray hybridizations to investigate how environmental perturbations and developmental cues regulate gene expression in plants[30]. Understanding the regulation of transcription in plants is crucial to improving crop productivity under changing environmental conditions[31,32]. However, computational tools in plant transcriptome analysis are largely based on co-expression analysis [33] that associate genes by computing correlation of expression. However, This correlation of expression may not be able to identify transcription factors (TFs) that regulate specific biological processes. Recent advancements in both *in vivo* and *in vitro* genomic experimental techniques have in generation of new data as tools to study transcriptional regulation in plants. Large-scale ChIP-seq experiments [15], protein binding microarrays [34], and DAP-seq experiments [16] have produced millions of candidate TF-target interactions. ATAC-seq and DNase hypersensitive assays have enabled profiling of active chromatin regions under specific conditions and in specific tissue types [17,35–37]. A current major challenge is how to integrate protein-DNA interaction data, active chromatin region data and expression data to reveal regulatory mechanisms of gene expression under specific conditions in plants.

Conventionally, regulatory mechanisms were characterized by constructing a genetic regulatory network (GRN) that typically consists of thousands of TF-target interactions. Many network inference approaches have been developed to construct GRNs by combining different types of genomic data. One notable example is mutual information (MI), which is a type of unsupervised machine learning approach that does not rely on any known interactions. Relevance Network (RN) is one of the first MI-based approaches to infer interactions [38]. To improve predictions, other MI-based inference methods were also developed [39–41]. Other unsupervised methods have also been developed including those based on partial correlation [42] and weighted co-expression networks [43]. By contrast, supervised machine learning approaches which take known interactions as prior knowledge, have also been applied to the inference of network interactions. Several commonly used supervised models can infer GRNs from expression data, including Support Vector Machine (SVM) [44,45], least angle regression (LARS) [46], least absolute shrinkage and selection operator (LASSO) [47,48], and elastic net (EN) [49]. Despite the success of these approaches in predicting gold standard interactions that were collected from a large compendium of publications, predictions solely based on expression profiles are far from perfect. Some approaches need gene expression data from multiple samples such as those from a time course experiment [47–51] or multiple tissue or cell types[38–41]. Such experiments are typically time consuming and are still not available for many plant species (See **supplementary table 2.1** for a summary of published methods). Each inferred interaction

represents global association between TF and target genes spanning many observations. However, regulatory interactions are often characterized by transient binding of TFs to cis-regulatory elements [52].

Other methods for the inference of interactions focus on data types that present direct evidence of binding events. Binding site data has received much attention in recent years in plants as evidenced by databases such as PlantTFDB [53], AGRIS [54], and Grassius [55] which have accumulated substantial amounts of data documenting experimentally identified binding sites. Previous studies have also identified a considerable number of binding sites from data obtained *in vivo* related to different environmental perturbations in plants. For example, binding sites were screened to construct regulatory networks in response to far red light [56,57], hormone [58–60], and fungal infection [61] in *Arabidopsis thaliana*. In contrast to expression data, binding site data present direct evidences of TF-target interactions. Based on available binding site data, several web-based tools have been developed to prioritize the targets of specific TFs for a group of genes using enrichment analysis. Some examples include TF2Network [62] and Cistome [63], which compute enrichment of binding sites for corresponding TFs based on large collection of binding sites in Arabidopsis; PlantPAN 3.0 [64], which identified enriched combination of TFs for multiple plant species, and g:Profiler [65], a tool designed to support binding site enrichment analysis of 213 species including 38 plant species.

However, direct evidence for binding site identification also has their limitations. For example, due to the cost of ChIP-seq experiments, only a few TFs were typically screened under any specific condition. Compared to ChIP-seq, DAP-seq can identify the possible targets of thousands of TFs efficiently [16]. However, DAP-seq is an *in vitro* technique [19], and some binding sites detected by DAP-seq may not be available for binding under a given environmental perturbation. Therefore, integration of binding site and expression data is key to improving prediction accuracy under specific conditions or cell types.

In this study, we developed the **Con**dition **S**pecific **Reg**ulatory network inference engine (**ConsReg**), a machine learning approach that infers regulatory networks from heterogeneous genomic data including expression data, DAP-seq data, and ATAC-seq data. Training data were supplied to machine learning models to perform binary classification with regularization-based feature selection. This procedure can prioritize and select the most relevant TFs for a specific environmental perturbation. We performed cross-validation for ConSReg using a compendium of expression data sets from 22 publications related to different environmental perturbations. The

evaluation result shows that the features of the integrated representation can accurately predict expression of target genes (average AUC-ROC = 0.84).

Our results highlight several important discoveries that provide new insights into the regulation of gene expression in plants. First, the appropriate selection of negative training data sets is crucial for the improvement of model performance, specifically, undetected genes are better negative training data than non-differentially expressed genes. Second, we demonstrated that including ATAC-seq data significantly improved model performance regardless of the experimental conditions, whereas prior publications in plant only demonstrated enrichment of binding sites or regulatory motifs in ATAC-seq peaks [20,22–24]. Third, we found that the length of promoter regions contributes to the model performance. Although published studies show that stress regulated motifs are enriched in 500bp upstream of the TSS of target genes [66], our analysis showed that using 3KB upstream of TSS +0.5KB downstream of TSS as promoter provides better performance across all data sets [62]. When ConSReg was applied to data sets generated from drought, cold, and heat perturbations, it successfully identified a pair of TFs, MYB44 and MYB77, that play active roles in co-regulating target genes in all three stresses. This result is consistent with recent findings that suggested that MYB44 and MYB77 are co-regulating TFs [35].

Despite the good performance reported by AUC-ROC (area under curve for receiver operating characteristic curve) values, one of the key challenges is the interpretability of evaluation results. While machine learning approaches used in this study are usually evaluated by metrics such as AUC-ROC, AUC-PRC (area under curve of precision-recall curve), these metrics are less interpretable compared to direct experimental evidence. Therefore, we reported a systematic recovery of N response TFs from a recently published study that used TARGET (Transient Assay Reporting Genome-wide Effects of Transcription factors*)* to screen target genes in Arabidopsis root [67]. TARGET is a novel *in vivo* technique that uses inducible translocation of a selected TF to induce gene expressions, with a focus on detecting genes with differential expression under a target TF [67,68]. We compared ConSReg to TF2Network [62], PlantPAN3.0 [64], and Cistome [63]. These tools can also infer and prioritize TFs from a list of target genes. We evaluated their ability to recover N response TFs using differentially expressed genes (DEGs) reported by TARGET assay as input. The idea is to use these DEGs as potential targets for N response TFs and infer N response TFs using different computational tools. ConSReg correctly selected 16 TFs from 17 TFs tested in the independently published data sets and provide an overall better ranking for N response TFs as compared to other methods.

We expanded ConSReg to a published single cell sequencing dataset from plants to infer regulatory networks at single cell level. We tested ConSReg to *Arabidopsis* single cell RNA-seq (scRNA-seq) data of two root cell types (endodermis and cortex) and successfully identified five key regulators in two root cell types. Four out of the five regulators are supported by additional evidence from existing publications or cell type-specific expression data (See **2.2 Results**). Finally, we demonstrated that ConSReg has the potential to transform any published gene expression data into condition specific gene regulatory networks which will provide a system level overview of transcriptional regulation in plants. ConSReg is provided as a Python package and is available for download from GitHub (https://github.com/LiLabAtVT/ConSReg).

2.2 Results

2.2.1 Analysis overview

In this work, we focused on using protein-DNA interaction data and open chromatin data to predict the combinations of TFs that can best explain observed differential gene expression under different environmental perturbations or in different cell types. To achieve this goal, we have tested multiple machine learning methods in combination with different feature selection techniques to determine the optimal parameters and training strategies.

Our pipeline consists of two major steps (See **Figure 2.1 A**). The first step is to integrate heterogeneous genomic data sets including interaction data generated from DAP-seq, open chromatin region data from ATAC-seq and expression data from RNA-seq/microarray experiments. This step produced training, validation and testing data sets for machine learning models. The second step is to perform binary classification with sparse feature selection methods. The input feature matrix for classification was constructed from binding site information (DAP-seq) and activated chromatin regions (ATAC-seq) for a list of differentially expressed genes. These genes were obtained by standard statistical analysis approach using a contrast between a replicate group of treated samples and a replicate group of control samples [69]. (See **Methods** and **supplementary Table 2.2** for more details). This contrast is hereinafter referred to as differential contrast.

The goal of the machine learning method is to identify a minimum set of TFs that can best explain the observed gene expression data. Specially, we were using differentially expressed genes as positive training set to perform binary classification. For each gene, the feature matrix consists of all interactions between TFs and their target genes. In our analysis, results from DAP-seq experiments were used to construct a feature matrix. This framework can be easily extended to include other interaction data. Open chromatin regions were used to set weights on the feature matrix (See **Figure 2.1B**). Up- and down-regulated genes were analyzed separately to train up-regulated (UR) and down-regulated (DR) models.

The performance of machine learning models was evaluated by AUC-ROC and AUC-PRC values computed from cross-validation. To prioritize important TFs for each condition, we assigned an importance score to each TF by performing stability selection [70] (see **2.5 Methods** for more details). We performed comprehensive analysis to identify optimal settings for machine learning models. We tested different machine learning approaches, different methods of selecting negative training samples (see 2.5 **Methods**), different lengths of promoter region, different combinations of data types, and different types of DAP-seq data.

The output results of ConSReg include a list of selected TFs inferred to be important for the given condition and the corresponding target genes for each selected TF. The selected TFs are ranked by importance scores which were computed from stability selection with LRLASSO (logistic regression with LASSO penalty). Briefly, input feature matrix is randomly scaled and a subset of training examples are randomly selected. Then a LASSO model is trained using the scaled subset of original training data set. This process will then be repeated multiple times. Importance score was computed as the number of times a TF was selected divided by the total number of times of randomization runs (See **2.5 Methods** for more details about computation of importance scores).

We performed various analyses to compare ConSReg with other computational approaches. We compared ConSReg to a basic enrichment test, which is a method commonly used by several computational tools to prioritize important TFs [62–64]. We reported results of recovery of N response TFs generated from ConSReg, TF2Network, plantPAN 3.0, and Cistome, as a case study of systematic reconstruction of regulatory events specific to an environmental perturbation.

Finally, we demonstrated application of ConSReg to single cell RNA-seq (scRNA-seq) data and constructed single cell GRNs for Arabidopsis root cell types. Although this application is exploratory, the results showed that ConSReg can successfully recovered known cell type specific TFs.

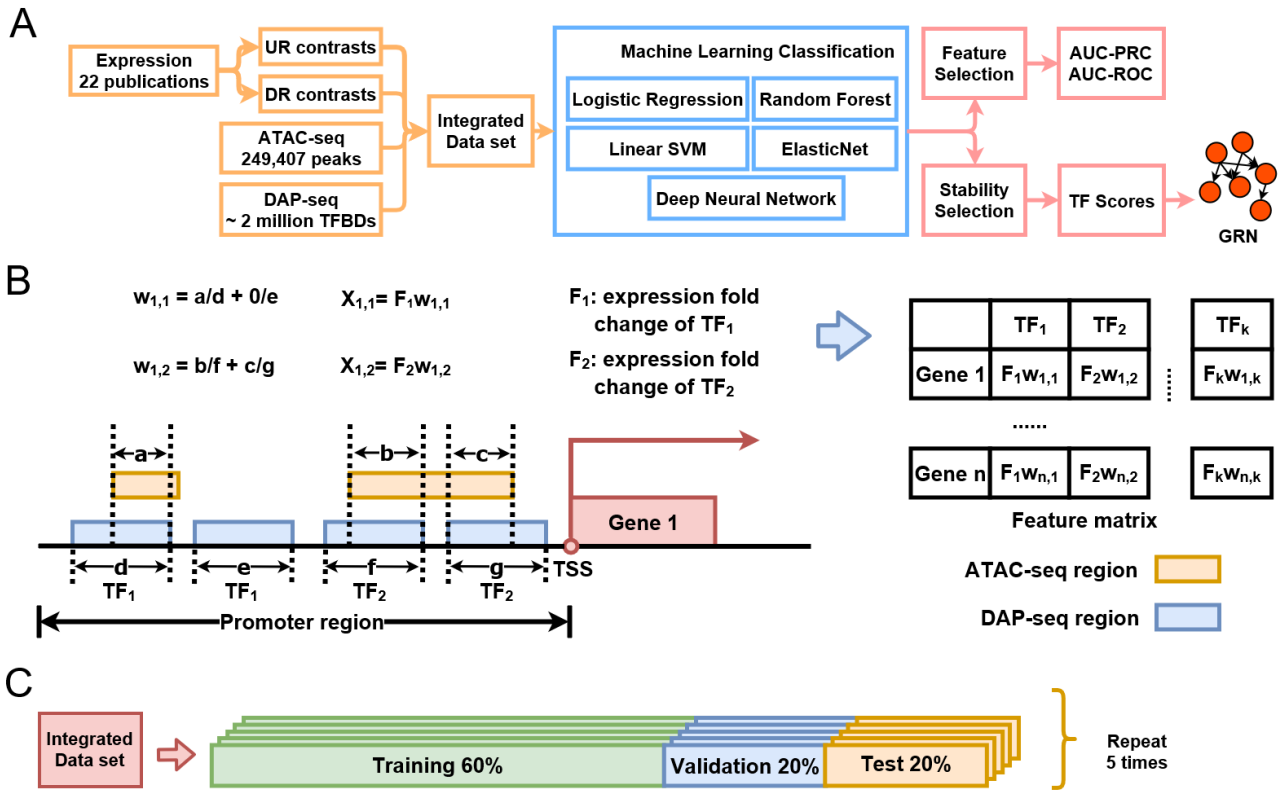


Figure 1.1 Flowchart of ConSReg pipeline. (A) Analysis workflow for this study. (B) Genomic data integration strategy. DAP-seq and ATAC-seq regions were intersected and a weight for each intersected region was computed then summed up as final weight for each TF-gene pair. The product of TF fold change and final weight is then filled into the corresponding entry of the feature matrix (See Methods for more details). a,b,c,d,e,f,g represent lengths of corresponding regions. (C) Cross-validation strategy. For each integrated data set (UR or DR), genes were split into 60% for training, 20% for validation (hyper-parameter tuning) and 20% for testing. Final AUC-ROC values were computed from the 20% test data. We repeated this for five times for each integrated data set and calculated average and standard deviation of AUC-ROC values.

2.2.2 Evaluation of different negative training data sets and different machine learning approaches.

As shown in a previous study, the choice of negative training data sets can significantly impact the performance of machine learning models [71]. We systematically evaluated three different methods for selecting negative training genes and evaluated the model performance for each method. These methods include: 1) non-significantly differentially expressed genes (**NDEGs**), which have p-value > 0.05; 2) low-expressed genes (**LEGs**), which have mean expression between 0 and 0.5; and 3) undetected genes (**UDGs**), which have a mean expression value equal to zero. The three methods were tested using evaluation dataset A (See **Methods**), where we constructed both an up-regulated (**UR**) feature matrix and a down-regulated (**DR**) feature matrix for each differential contrast. Machine learning models tested in this analysis include 1) logistic regression with lasso penalty (**LRLASSO**), 2) logistic regression with group lasso penalty (**LGLASSO**), 3) logistic

regression with elastic net penalty (**LREN**), 4) logistic regression with Pearson Correlation Coefficient (**LRPCC**), 5) guided regularized random forest (**GRRF**), 6) linear support vector machine (**LSVM**). See **Methods** for more details about the machine learning models.

For both UR and DR feature matrices, UDGs show consistently higher AUC-ROC than NDEGs and LEGs (**Figure 2.2A**). AUC-ROC values of UDGs are significantly higher than NDEGs (Wilcoxon signed-rank test, p-value < 0.001) and LEGs (Wilcoxon signed-rank test, p-value < 0.001). Shown in **Figure 2.2A** are all AUC-ROC values computed from the six machine learning approaches. The demonstrated results suggest that for both UR feature matrices and DR feature matrices, machine learning classifiers are better at classifying positive training genes vs UDGs as compared to positive training genes vs other types of negative training genes. However, we do not find obvious difference for number of selected TFs among the three types of negative training genes (embedded plot in **Figure 2.2A**).

We further compared the performance of different machine learning approaches and found that the six machine learning approaches achieved similar AUC-ROC values (**Figure 2.2B** and **Figure 2.2C**). However, the numbers of selected TFs by different machine learning models are quite different (**Figure 2.2B** and **Figure 2.2C**). LRLASSO selected fewer TFs than other methods, and standard deviation of the number of selected TFs is smaller than other models. Based on this observation, we performed the following analyses using LRLASSO as the best prediction method.

In recent years, deep neural network (DNN) has been extensively applied in the field of genomics to model gene regulations [72–74]. We further explored whether DNN can bring better performance than LRLASSO. A previous study has introduced a DNN-based feature selection method [75]. We used a similar strategy in our analysis (See **Methods**) to prioritize TFs and compare the result to LRLASSO. A DNN usually needs a large number of training samples to estimate thousands of parameters. However, most of the expression data sets used in this study have fewer than 2000 genes for training (See **Supplementary table 2.2**). Therefore, a comparison using multiple data sets in evaluation dataset A or evaluation dataset B (See **Methods**) only demonstrates poorly fitted DNN model and can be biased. We selected a differential contrast that has the largest number of training samples (8,948 genes for UR and DR feature matrices, respectively). Shown in **Figure 2.2D** are AUC-ROC values and selected number of features from five rounds of cross-validation. For UR feature matrix, the performance of LRLASSO is significantly better than DNN (**Figure 2.2D**, Wilcoxon rank-sum test, p-value < 0.01), whereas the performance does not show significant difference for DR feature matrix (**Figure 2.2D**, Wilcoxon rank-sum test, p-value > 0.05). LRLASSO selected fewer TFs than DNN (embedded plot in **Figure 2.2D**) and has more robust

performance compared to high variation of number of selected TFs from DNN. Given these findings, we concluded that DNN-based feature selection does not perform better than LRLASSO in our analysis.

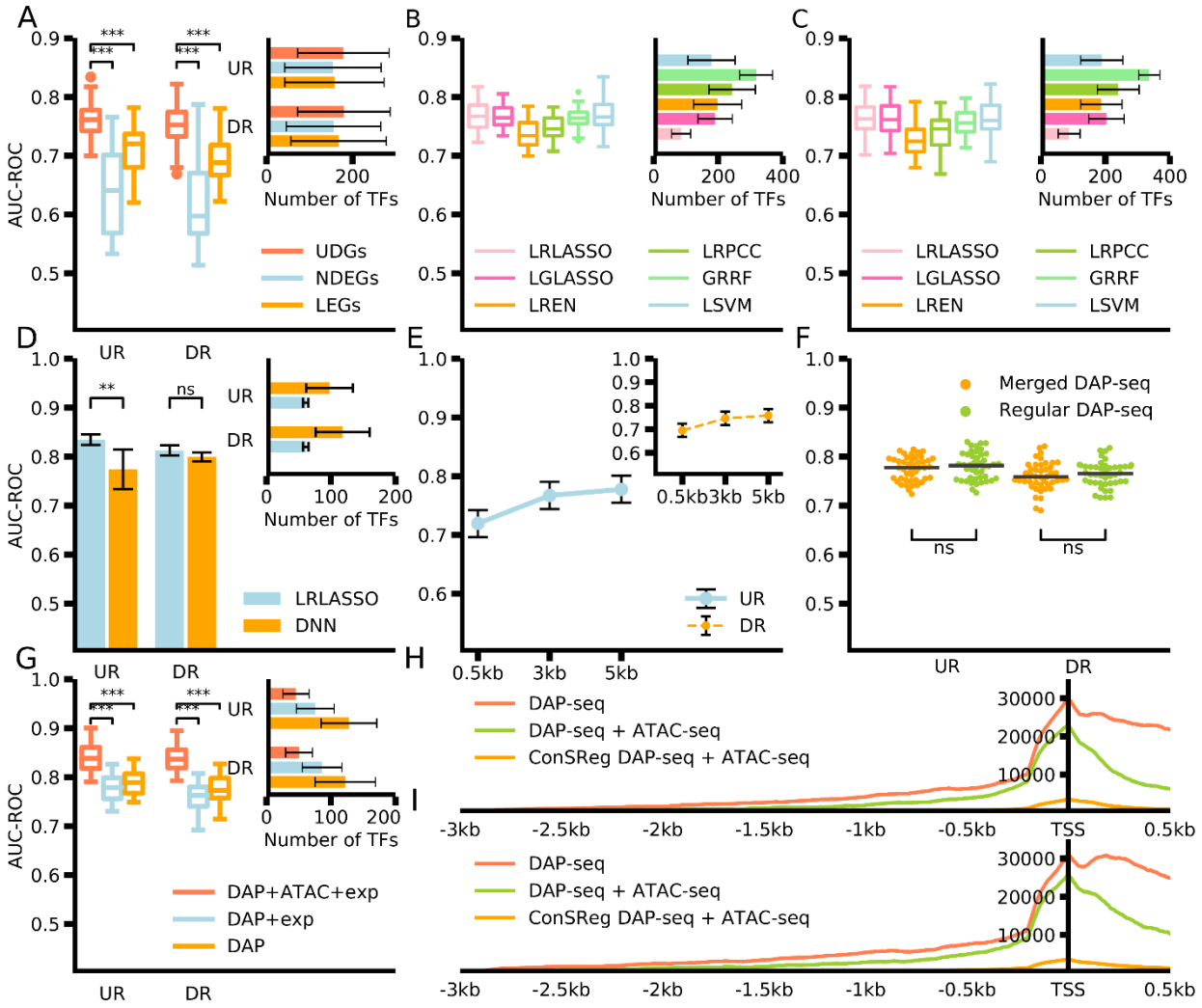


Figure 1.2 Comprehensive evaluation of model performance under different conditions. (A) Evaluation for different negative data sets. **UDGs**: undetected genes, **NDEGs**: non-differentially expressed genes, **LEGs**: low-expression genes. Box plot demonstrates AUC-ROC for different negative data sets (Three boxes on the left: UR models, three boxes on the right: DR models). The embedded bar chart shows number of TFs obtained using different negative data set.s Other conditions used for this sub-figure: 1) Results of all classifiers were mixed 2) Promoter region: 3kb upstream and 0.5 kb downstream. 3) Merged DAP -seq + expression data were used. 4) AUC-ROC values were computed using evaluation data set A. **(B, C)** Evaluation for different classifiers. **LRLASSO**: logistic regression with LASSO penalty, **LGLASSO**: logistic group LASSO, **LREN**: logistic regression with elastic net penalty. **LRPCC**: logistic regression with Pearson correlation coefficient, **GRRF**: Guided regularized random forest, **LSVM:** linear support vector machine. **B** shows the AUC-ROC values for UR model and **C** shows the AUC-ROC values for DR model. In both **B** and **C**, box plot demonstrates AUC-ROC values and embedded bar chart shows number of selected TFs. Other conditions used for the two sub-figures: 1) UDGs were used as negative data set. 2) Promoter region: 3kb upstream and 0.5 kb downstream. 3) Merged DAP -seq + expression data were used. 4) AUC-ROC values were computed using evaluation data set A. **(D)** Comparison between LRLASSO and DNN. Bar chart in the major plot area

shows comparison of AUC-ROC values (two bars on the left: UR model, two bars on the right: DR model). The embedded bar chart shows comparison of number of selected TFs (two bars on the top: UR model, two bars on the bottom: DR model). Other conditions used for this sub-figure: 1) UDGs was used as negative data set. 2) Promoter region: 3kb upstream and 0.5 kb downstream. 3) Merged DAP -seq + expression data was used. 4) AUC-ROC values were computed using ABA response data set. **(E)** evaluation for different promoter region lengths. Curve in the major plot area shows AUC-ROC values for UR model computed from evaluation data set A and curve in the embedded plot area shows AUC-ROC values for DR model computed from evaluation data set A. Other conditions used for this sub-figure: 1) UDGs was used as negative data set. 2) Merged DAP-seq + expression data was used. **(F)** comparison between merged DAP-seq and regular DAP-seq. Medians were marked by black bars. Two clusters on the left: UR model, two clusters on the right: DR model. Other conditions used for this sub-figure: 1) UDGs was used as negative data set. 2) Promoter region: 3kb upstream + 0.5kb downstream. 3) Merged DAP-seq/regular DAP-seq + expression data was used. 4) AUC-ROC values were computed using evaluation data set B. **(G)** Evaluation for different integration strategies. Box plot shows comparison of AUC-ROC values and embedded bar chart shows comparison of number of selected TFs. Other conditions used for this sub-figure: 1) UDGs was used as negative data set. 2) Promoter region: 3kb upstream + 0.5kb downstream. 3) Merged DAP-seq + expression data was used. 4) AUC-ROC values were computed using evaluation data set B. **(H, I)** Peak pileup plot for TGA1 TARGET perturbation experiment. X axis represents upstream and downstream promoter region positions relative TSS (upstream positions were marked by negative numbers). Y axis represents number of peaks mapped to the corresponding position. Both **H** and **I** are pileup of peaks computed from promoter regions of positive genes. **H** shows the peak pileup for UR model and **I** showed the peak pileup for DR model. Red curve is the pileup for DAP-seq binding sites. Green curve is the pileup for overlapped regions between DAP-seq binding sites and ATAC-seq open chromatin regions. Orange curve is the pileup of ConSReg predicted binding sites selected from regions presented by green curve.

2.2.3 Condition specificity of negative training genes.

Although positive training genes in this study reflect condition-specific activities, it is unclear whether negative training genes are also condition specific. One possibility is that all negative training genes are not detected under any of the conditions tested. We checked whether UDGs are different in different environmental perturbations. For each differential contrast in each environmental perturbation, we computed the percentage of UDGs that are detected (fpkm > 0) in other perturbations. Then the percentages were averaged for each environmental perturbation. We found that this average percentage ranges from 72.54% to 91.76%, suggesting that UDGs in one condition are typically expressed in other environmental perturbation(s). Therefore, a large portion of UDGs remain inactive specific to one or multiple environmental perturbations (**Supplementary Figure 2.1**).

2.2.4 Choice of promoter region length affects model performance.

TFs regulate expression of target genes by binding to regulatory elements located in the promoter regions of these genes. Combinatorial regulations by multiple TFs have been studied in plants [76]. It has been shown that binding sites located within a 5kb upstream region of transcription start sites (TSS) can better explain regulatory effect on the target genes than shorter regions [62]. However, the optimal promoter length has not been thoroughly investigated using. We therefore set the promoter region length up to 5kb upstream of TSS and 1kb downstream of TSS in feature construction step (See **Methods**) and tested different lengths under various environmental perturbations. We tested three types of promoter regions: 1) 5kb upstream of TSS to 1kb downstream of TSS; 2) 3kb upstream of TSS to 0.5kb downstream of TSS; and 3) 0.5 kb upstream of TSS to TSS. **Figure 2.2E** shows AUC-ROC values for these three types of promoter regions evaluated on evaluation dataset B (See **Methods**). We observed consistent improvements when the promoter region length was extended from 0.5 kb upstream to 3kb upstream + 0.5kb downstream. When the promoter region was further extended to 5kb upstream + 1kb downstream, no significant improvement was found. As shown in **Figure 2.2E**, these results are consistent for UR models and DR models. In summary, our findings suggest that most of the binding sites predictive of gene expressions were successfully captured within 3kb upstream + 0.5kb downstream region.

2.2.5 Methylation events do not significantly affect model performance.

As described previously [16], DAP-seq can be performed in two ways: 1) sequence regular genomic DNA (gDNA), 2) sequence gDNA libraries in which methyl-cytosines were removed by PCR. The former is regular DAP-seq and the latter is called ‘ampDAP-seq’ [16]. We tested the performance of two sets of binding sites: 1) using all available DAP-seq binding sites, which is the

merged set of regular DAP-seq binding sites and ampDAP-seq binding sites 2) using only regular DAP-seq binding sites. Although it was reported that many DAP-seq binding sites (~180,000) were occluded by DNA methylation, our result shows that, compared to using regular DAP-seq binding sites, the merged set of DAP-seq binding sites does not provide better prediction result (**Figure 2.2F**). Therefore, methylation events in DAP-seq data do not significantly affect model performance.

2.2.6 ATAC-seq data significantly improves model performance.

Although DAP-seq has shown higher throughput than earlier TF-target screening assay such as ChIP-seq [16], all DAP-seq binding sites are detected *in vitro*, and some of the binding sites in vitro might not be accessible in living cells. As suggested in a previous publication describing the DAP-seq assay, this limitation can be overcome by integrating DAP-seq with data on open chromatin regions [19]. To improve prediction performance, we encoded open chromatin information from ATAC-seq data into the feature matrices (See **Figure 2.1B** and **2.5 Methods**). To assess the impact of chromatin accessibility, the feature matrices were constructed either with or without integrating ATAC-seq data. In this analysis, ATAC-seq data was downloaded from one published data set [20] and open chromatin regions of all tissues were merged into a single ATAC-seq data set. We compared the model performance of ATAC-seq included feature matrices to ATAC-seq free feature matrices using evaluation dataset B (See **2.5 Methods**). As shown in **Figure 2.2G**, for both UR and DR genes, there are consistent improvements when ATAC-seq data were included in the feature matrices. The other noticeable advantage of including ATAC-seq data is that it helped the model to select fewer TFs, producing a more interpretable result (embedded plot in **Figure 2.2G**). We further investigated whether including expression and ATAC-seq data can better predict expressions than using DAP-seq binding site information alone. To answer this question, we constructed feature matrices only by DAP-seq data (see **2.5 Methods**) and compared prediction results to feature matrices constructed with expression, ATAC-seq peaks, and DAP-seq binding sites. The results show that including all three types of data has consistently improved performance (**Figure 2.2G**).

2.2.7 ConSReg outperforms simple enrichment test

An enrichment test has been applied in recent studies to prioritize TFs given a set of input genes [62,63,65,77] (See **2.5 Methods** for details about enrichment test). However, enrichment-based prediction does not take into consideration combinations of multiple TFs. We compared our prediction pipeline to an enrichment-test-based method (See **2.5 Methods** for details). We computed AUC-ROC values using evaluation dataset B (See **2.5 Methods** for details). As shown in **Figure 2.3A**, ConSReg outperforms enrichment test in all tested differential contrasts. AUC-ROC

values for ConSReg are significantly higher than enrichment test (Wilcoxon rank-sum test, p-value < 0.001 for both UR and DR feature matrices).

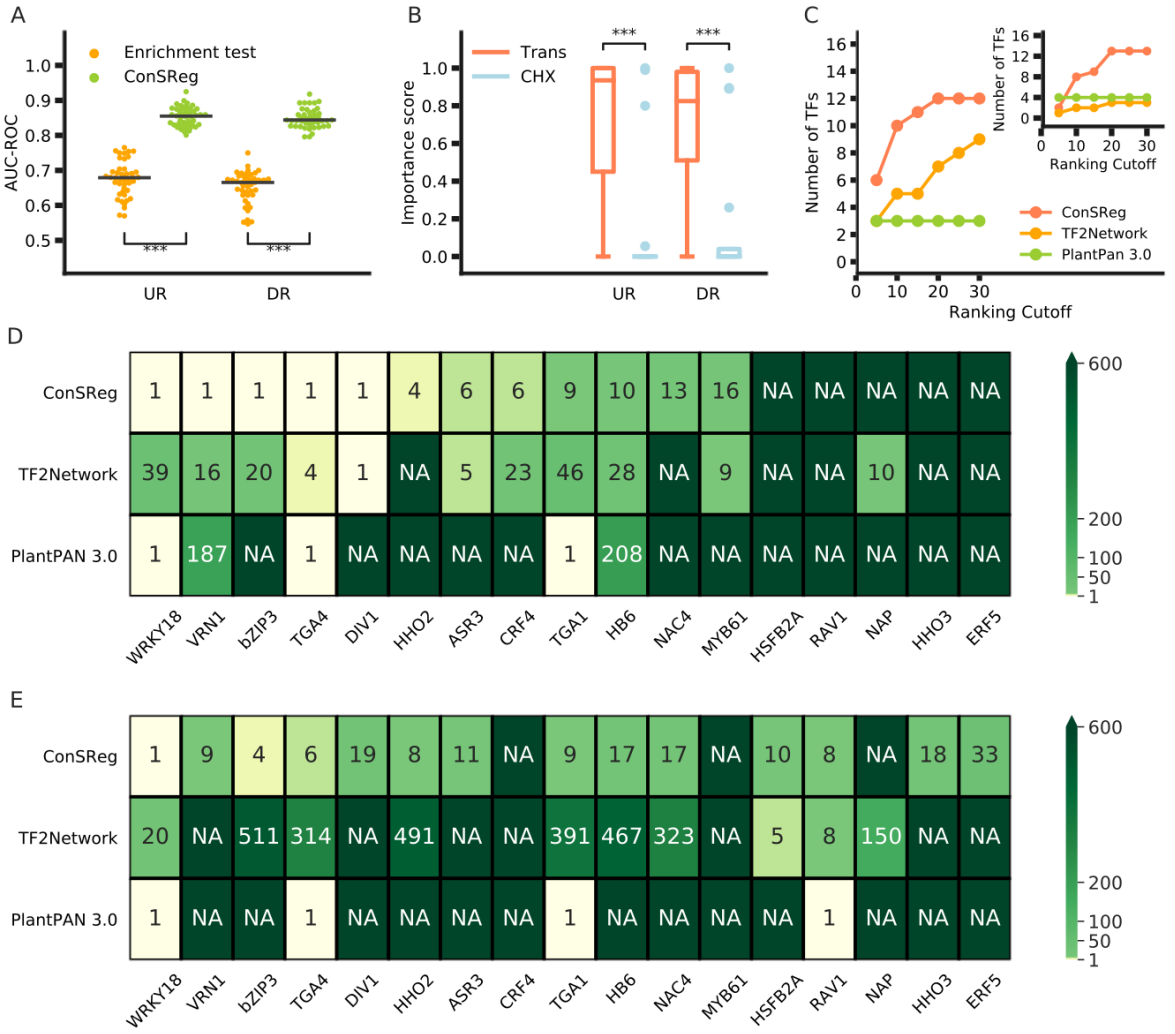


Figure 1.3 Comparison of different computational methods. (A) AUC-ROC for enrichment test and ConSReg. Two clusters on the left represent AUC-ROC values for UR models and two clusters on the right represent AUC-ROC values for DR models. **(B)** Importance scores of the 17 TFs for TF transfected root cells and CHX treated root cells. Two boxes on the left represent importance scores for UR models and two boxes on the right represent importance scores for DR models. **(C)** Number of recovered TFs for ConSReg, TF2Network, PlantPAN 3.0 and Cistome in different ranking cutoffs. Results predicted from UR models are plotted in the major plot area and results predicted from DR models are plotted in the embedded plot. **(D-E)** Ranking for each of the 17 N responsive TFs predicted by ConSReg, TF2Network, PlantPAN 3.0 and Cistome. Ranking for each TF was mapped to a color scale represented by a color bar on the right. Lighter color indicates better ranking. **D** shows the results predicted by UR model and E shows the results predicted by DR model

2.2.8 ConSReg recovered TFs known to be involved in nitrogen response

To evaluate whether ConSReg can systematically recover known TFs involved in a specific environmental perturbation, we applied ConSReg to an Arabidopsis root RNA-seq dataset from a recently published study [67] and compared our result to TF2Network, PlantPAN 3.0, and Cistome. The authors of this study used modified TARGET assay to evaluate how N responsive TFs can regulate gene expressions of their targets. In this study, 33 TFs were selected based on their transcriptional response to N, identified by a previously published time-course study [52]. The selected 33 TFs were transfected into each root protoplast cell pool one at a time and cells were treated with N. Entry of TF into nuclear is controlled by a subdomain of the glucocorticoid receptor (GR) fused to the TF. HSP90-GR binding holds the GR-TF fusion protein in cytoplasm. Expression of TFs were then induced by disrupting this binding with dexamethasone (DEX) treatment after N treatment. The last step is to perform RNA-seq for each cell pool transfected with the corresponding TF to obtain expressions of target genes [67]. We applied ConSReg to this RNA-seq dataset in combination with root ATAC-seq data and DAP-seq data we collected, in an effort to evaluate how many of the induced TFs can be recovered by ConSReg. The idea is to use expression data of target genes detected by TARGET assay and infer potential TFs that regulate them. We found there is an overlap of 17 TFs between 387 DAP-seq tested TFs and the 33 N responsive TFs. We therefore used this set of 17 N responsive TFs for our evaluation (See **Figure 2.3D** and **Figure 2.3E** for TF gene names). We re-analyzed the RAN-seq data using DESeq2 and generated differential contrasts (See **Supplementary table 2.2**) as input for ConSReg.

We first compared importance scores of the 17 N response TFs in two test conditions: 1) cycloheximide (CHX) and N treated TF transfected root cells versus (VS) CHX and N treated empty vector (EV) transfected root cells and 2) CHX and N treated EV transfected root cells VS N treated EV transfected root cells. CHX was used to block downstream regulation of secondary TF targets [67]. For the first condition, we generated a differential contrast between each TF transfected group and EV transfected control group. This resulted in 17 differential contrasts respectively for the 17 N response TFs. For the second condition, we generated a single differential contrast between CHX treated EV transfected samples VS EV transfected samples (See **Supplementary table 2.2**) and obtained importance scores for all 17 N response TFs from this single contrast. For both UR and DR predicted TFs, the importance scores from the first condition were significantly higher than the second condition (**Figure 2.3 B**, Wilcoxon signed-rank test, p-value < 0.001 for both), suggesting that ConSReg can successfully reconstruct the 17 induced N response TFs in first condition while assigning very low importance scores to the same but non-induced 17 TFs in second condition.

We then compared the result of 17 N responsive TFs to the results generated from each of TF2Network, plantPAN 3.0, and Cistome. These published tools can infer regulators for a given list of target genes. For ConSReg, the ranking of each N response TF was retrieved using the following two steps 1) For each of the 17 differential contrast described previously, we generated importance scores for 387 DAP-seq TFs and sorted TFs by descending order using importance scores. Only TFs having importance scores > 0.5 were kept. 2) The previous step can generate a set of ranked TFs for each of the 17 differential contrasts. For each differential contrast, we obtained the ranking of the corresponding TF used to transfect the protoplast cell pool. For other tools, we used UR DEGs and DR DEGs generated from DESeq2 as input gene lists to infer their active TFs. Therefore, same UR and DR input of 17 sets of DESeq2 results were used for all the tools. Instead of directly returning a list of TFs, Cistome returns enriched motif IDs collected in the CIS-BP database [78]. We first ranked these motifs by their q-values and mapped motif IDs to their corresponding TF IDs. Ranking of the best-ranked motif for the TF was used as the final ranking for that TF. Similarly, we ranked the TF2Network-predicted TFs by the p-value of best-ranked motif of each TF

We compared ranking results for the 17 N TFs. We set different cutoffs for ranking and only considered TFs that obtained better ranking than the cutoff as true positive predictions (**Figure 2.3C**). The original outputs and final ranking results of all the tools can found in **Supplementary file 1**. We found ConSReg consistently outperformed other tools under different cutoffs. As an example, when ranking cutoff is set to the top 30 predicted TFs, ConSReg can recover 12/17 N response TFs (70.59%) from UR models, compared to the recovery rate of TF2Network (11/17, 64.71%), PlantPAN 3.0 (9/17), and Cistome (0/17, 0%). For DR models, ConSReg was able to recover 14 of 17 N response TFs (82.35%) from the top 30 predicted TFs, compared to the recovery rate of TF2Network (3/17, 17.65%), PlantPAN3.0 (4/17, 23.53%), and Cistome (0/17, 0%). Cistome failed to recover any of the 17 N response TFs. This possibly is because Cistome uses version 1.01 motif IDs from CIS-BP database, which was collected before Arabidopsis DAP-seq data was published. Many DAP-seq tested TFs can be missing in this version. Therefore, we only demonstrated the comparison among ConSReg, TF2Network, and PlantPAN 3.0 in **Figure 2.3** and hereafter omit Cistome in our discussion. As shown in **Figure 2.3D** and **Figure 2.3E**, there is a considerable overlap of TFs (10 TFs) between UR and DR models predicted by ConSReg and this number is higher than TF2Network (6 TFs) and PlantPAN 3.0 (3 TFs). This observation is consistent with the results reported in the previous study that all of the 33 assayed N response TFs can act as both an inducer and a repressor of target genes [67]. Detailed ranking results showed that for many recovered TFs, ConSReg assigned better rankings compared to other tools. For example,

five recovered UR model TFs (WRKY18, VRN1, bZIP3, TGA4, and DIV1) were ranked as top 1 by ConSReg and these rankings are better than the other two tools (See **Figure 2.3D**). Notably, a few TFs predicted by PlantPAN 3.0 achieved ranking of top 1, while others predicted by PlantPAN 3.0 were assigned very low rankings (187 for VRN1, 208 for HB6, see **Figure 2.3D**). This is not surprising since many TFs predicted by PlantPAN 3.0 have identical support value, these TFs will share identical ranking. For example, although WRKY18 was ranked as top 1 by PlantPAN 3.0 in UR models, there are 187 other TFs which were assigned the same ranking (See **supplementary file 2.1**). In contrast, ConSReg only showed one other TF that shared the same ranking as WRKY18. Compared to PlantPAN 3.0, this result can better capture the known truth that WRKY18 involved in regulating transcriptional N response in this experiment. Taken together, we believe ConSReg generated better and more interpretable ranking results than TF2Network and PlantPAN 3.0.

2.2.9 Importance score can indicate predictability of TF

To evaluate the predictive power of highly ranked TFs and to verify whether these TFs are more predictive of gene expressions than other TFs, we performed simulation of perturbation to TFs with high importance scores (importance score > 0.5). We first computed importance score for each TF using differential contrasts compiled in evaluation dataset B. Then we removed all TFs with importance scores = 0. For the remaining TFs, we compiled following three sets of TFs in each differential contrast: **1)** all TFs with importance scores > 0.5; **2)** replace the top five TFs in 1) using five lowest ranked TFs; **3)** replace the top ten TFs in 1) using ten lowest ranked TFs. We evaluated the performance of the three sets of TFs using the same cross-validation strategy shown in **Figure 2.1 B**. The results are shown in **Figure 2.4**. The reported AUC-ROC and AUC-PRC values for **1)** are significantly higher for the other two sets of TFs (wilcoxon signed-rank test, p-value < 0.01 and p-value < 0.001). This can be observed clearly in **Figure 2.4 A** and **Figure 2.4 B**, where performance of UR models was evaluated. Similar pattern were not apparent for DR feature matrices (**Figure 2.4 C** and **Figure 2.4 D**), suggesting that DR regulatory processes are more difficult to be modeled than UR. Taken together, we concluded that for modeling UR genes, TFs with higher importance scores can be more predictive of gene expressions.

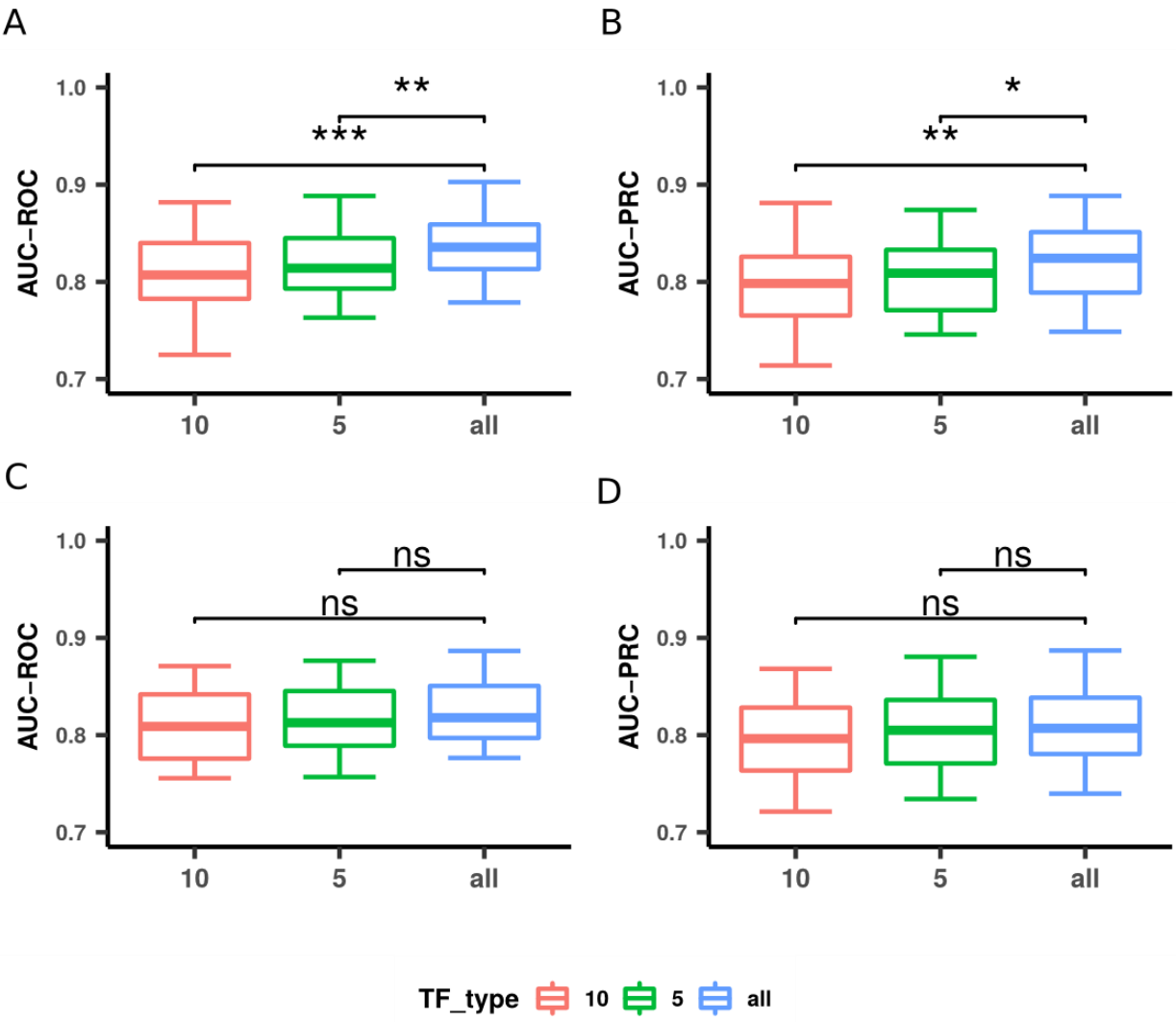


Figure 1.4 Simulation of Perturbation for TFs. We performed simulation to perturb TFs with high importance scores (importance score > 0.5). Results shown here were generated from evaluation dataset B. For each differential contrast in evaluation dataset B, we used TFs with importance scores > 0.5 to construct three sets of TFs for testing: **1)** all TFs with importance scores > 0.5 (marked by 'all' in the figure); **2)** replace the top five TFs in 1) using five lowest ranked TFs (marked by '5' in the figure); **3)** replace the top ten TFs in 1) using ten lowest ranked TFs (marked by '10' in the figure). **(A B)**, AUC-ROC and AUC-PRC for UR models. **(C D),** AUC-ROC and AUC-PRC for DR feature matrices. Significance level was marked by stars over the boxes. *: p-value < 0.05; **: p-value < 0.01; ***: p-value < 0.001; ns: not significant. P-value was computed from wilcoxon signed-rank test.

2.2.10 ConSReg successfully identified known TFs that regulate abiotic stress

We performed a comprehensive investigation of potential TFs active under multiple abiotic environmental perturbations using our prediction pipeline. ConSReg was run for all differential contrasts included in evaluation dataset B, which encompasses nine common environmental perturbations: cold, heat, drought, salt, wounding, osmotic stress, red light, blue light, and high light. For each differential contrast in each environmental condition, we assigned importance score to each TF by LRLASSO + stability selection (See **Methods**). The highest importance score was

selected as a representative importance score for each TF in each environmental condition. We then inferred a GRN for each environmental condition (See **Methods**). We computed the basic network properties of these GRNs (**supplementary table 2.3**). We counted how many times each TF achieved a score higher than 0.5 across nine environmental conditions. This number is hereafter referred to as 'condition count' which was then used to rank all TFs.

MYB and ERF protein families are known for regulating many abiotic stress responsive genes [79,80]. We found that presence of many ERF and MYB/MYB related TFs were present among our top 20 candidates generated from UR feature matrices, including five TFs from MYB/MYB related family (AT1G18330, AT3G50060, AT1G49010, AT5G67300, and AT1G74650) and two TFs from the ERF family (AT2G31230 and AT4G16750).

Some of the known stress related TFs were ranked as top candidates by condition count (See **supplementary table 2.4**). For example, AT1G27730 (ZAT10), which was reported to be involved in defense response of plants to abiotic stresses such as heat, cold, drought, and salt [81–83], was assigned high condition count in both UR and DR feature matrices (condition count = 8 for both UR and DR feature matrices). Our result is consistent with a previous study that suggests that ZAT10 may function as both a positive and a negative regulator of defense response to abiotic stresses [82]. More interestingly, our result indicates that ZAT10 is likely to be an active regulator during multiple light environment perturbations including high light, blue light, and red light. However, the involvement of ZAT10 in multiple light stresses has not yet been well characterized. Although there are evidences suggesting that ZAT10 mediates a response to high light [84,85], little evidence has been reported about the involvement of ZAT10 in red light and blue light response. A previous study identified ZAT10 as the substrate of Mitogen-Activated Protein Kinases (MAPK) and showed that ZAT10 can directly interact with two MAPKs: MPK3 and MPK6 [86] through protein-protein interaction. It has been reported that MAPKs can be activated by blue light to modulate the response [87]. We computed Pearson Correlation Coefficient (PCC) to quantify co-expressions of ZAT10 with a gene that encodes MPK3 protein (AT3G45640), and ZAT10 with another gene encoding MPK6 protein (AT2G43790). Significance of co-expression was computed by Fisher's Z-transformation as described in [88] (See **2.5.6 Computation of p-values for co-expression analysis** for more details). Expression data used were from a GSE data set (GSE59699) generated under blue light treatment (See **supplementary table 2.2** for sample SRR IDs and **supplementary file** 2.2 for the expression matrix of blue light treatment). Our result shows that ZAT10 has exhibited a significantly high co-expression with MPK3 (PCC = 0.943, p-value = 4.232×10^{-12}). However, ZAT10 was not significantly co-expressed with MPK6 (PCC = 0.190, p-value = 0.228). This result

indicates MPK3 may impact ZAT10 through protein-protein interaction under blue light treatment. Given the evidences above, we hypothesize blue light response is likely to be regulated by MPK3-ZAT10 interaction, and ZAT10-initiated gene regulation.

The other notable example of a gene with high condition count (condition count = 8) is AT2G46680 (ATHB7), a member of the homeodomain-leucine zipper family (HD-ZIP). ATHB7 was already shown to confer salt and drought tolerance to plants [89–91] and act as a negative regulator for plant growth [89]. In contrast, its functional role for other stress responses is less well characterized. Our result shows that ATHB7 is an active regulator for both UR genes and DR genes, suggesting it may function as both positive and negative regulator for multiple environmental stresses (See **supplementary table 2.4**).

2.2.11 ConSReg uncovers combinatorial regulation patterns

TFs are known to modulate expression of target genes by combinatorial regulation in plants [76,92]. Combinatorial regulations among TFs can be established either by forming protein complexes between TFs, or through indirect interactions between TFs [76]. We explored possible combinatorial regulations between TFs with high importance score (> 0.5) for three environmental perturbations (cold, heat, and drought). For each environmental perturbation, we computed importance scores and inferred GRN (See **2.5.6 Network Inference** in **Methods**). Due to the large size of each GRN, we selected the TFs from each GRN and visualized only the subnetworks formed by these TFs. Shown in **Supplementary figure 2.2** and **Supplementary figure 2.3** are subnetworks of TFs for cold, heat, and drought condition. Detailed information regarding all interaction edges in these subnetworks has been put into **Supplementary table 2.5** and **Supplementary table 2.6**. To better understand this hierarchy of regulation, we clustered the TFs using a simulation-annealing-based algorithm [93]. We enforced the algorithm to cluster TFs in three clusters: 1) top TF cluster, in which TFs have many out-going edges and fewer in-coming edges from other TFs, 2) bottom TF cluster, in which TFs have many in-coming edges and fewer out-going edges from other TFs and 3) intermediate TF cluster, in which TFs have balanced out-going edges and in-coming edges from other TFs. The visualization result shows that many TFs were put into the top TF cluster, suggesting that many of these TFs act as master regulators regulating fewer intermediate TFs and bottom TFs (**Supplementary figure 2.2** and **Supplementary figure 2.3**).

An important pair of TFs, MYB44 and MYB77, was predicted by ConSReg to be involved in combinatorial regulation of multiple stresses. We identified co-regulating modules of TFs from each GRN using a previously published tool CoReg [94]. Then we identified maximum common co-regulating TFs (See **Methods**) across the three conditions. While no common co-regulating TFs can

be found for UR GRNs, we found a pair of common co-regulating TFs for DR GRNs: MYB77 (AT3G50060) and MYB44 (AT5G67300). We plotted the network of MYB77 and MYB44 with their respective top 20 DEG targets in each condition sorted by ascending order of p-value. (**Figure 2.5)**. Our analysis result indicated that combinatorial regulation exists between MYB77 and MYB44 across different abiotic stresses. Despite that the two TFs are not differentially expressed in the three abiotic stresses tested (cold, heat, and drought), they both have high importance scores in different abiotic stresses (See **Supplementary table 2.4**). Although MYB77 and MYB44 are known to be functionally redundant and can both regulate auxin signaling [95], the coordination between the two TFs across multiple abiotic stresses remains poorly understood. In a recent publication, MYB77 and MYB44 were identified as a regulatory module that co-target and regulate many root hair cell specific genes [35]. An enrichment of stress response functional annotations also indicated their role of co-regulating stress responses [35]. These findings lend further support to our prediction result. MYB44 has been reported to be involved in enhancing abiotic stress tolerance of plants [96,97]. As for MYB77, it is a known regulator for auxin-responsive genes [95]. Previous study suggests that auxin content can induce GH3 genes, which in turn can suppress auxin signaling [98]. This reduces plant growth and thus increases resistance to abiotic stresses [98]. In summary, both MYB44 and MYB77 are related to enhancing abiotic stress tolerance. Although this is difficult to be directly observed based solely on any single type of data, importance score generated by ConSReg were able to provide insight into regulatory roles of TFs. Taken together, we concluded that combinatorial regulation between MYB77 and MYB44 confers abiotic stress tolerance to plants.

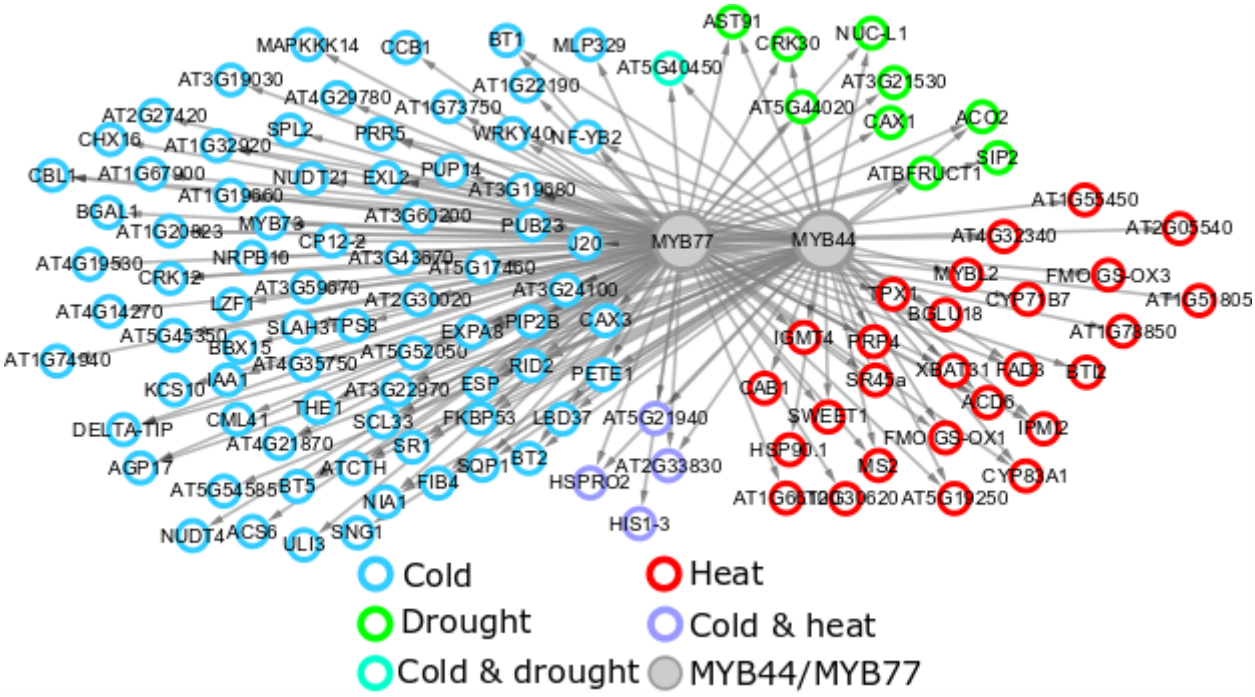


Figure 1.5 Combinatorial regulation between MYB44 and MYB77. Plotted in the center are MYB44 and MYB77 which regulate many common target genes in response to different abiotic stresses. For the targets of MYB44 and MYB77, target genes of 20 top DEGs in each differential contrast were selected to be plotted in the figure. Edge list of this network can be found in **supplementary table 2.7**.

2.2.12 Inferred regulatory genes from single cell RNA-seq data agree with bulk sequencing results.

We applied ConSReg to Arabidopsis scRNA-seq data of two root cell types (endodermis and cortex). scRNA-seq data was generated by drop-seq assay in a recent study [99]. UR and DR feature matrices were generated by comparing cortex cell type to endodermis cell type and importance scores were computed using these feature matrices (See **Methods** for details). TFs with importance score > 0.5 were considered to be predicted regulator. Among the predicted regulators of single cell gene expression from cortex cluster and endodermis cluster, we found 4 and 5 genes are consistently predicted (importance score > 0.5) as regulators for genes preferentially expressed in cortex and endodermis clusters respectively. Among these genes, we identified two genes (ATWRKY27 and ATHB34) that are commonly active in both cortex and endodermis clusters. The two commonly predicted factors (AtWRKY27 and ATHB34) could represent common regulators for both cortex and endodermis cells. The cortex cluster contains a unique regulator gene, ERF115, whereas the endodermis cluster contains BBX31 and TGA6. Surprisingly, among more than 20 ethylene response factor (ERF) genes included in our input dataset, our model selected ERF115, which is known to regulate cell cycle in quiescent center (QC) cells [100]. Although QC cells are not cortical or endodermal cells, this discrepancy could be attributed to the noise in assigning single cell sequencing data to specific cell types. Some QC cells may have similar expression profile as young

cortex or endodermis cells. For example, the commonly used SCR marker include cells from both QC and endodermis. The BBX31 gene is not only predicted as a regulator for single cell gene expression data, but is also found to be regulator in bulk RNA-seq (importance score = 0.98 in E30 cell type, see **supplementary table 2.8**), strongly supporting the role of this gene in controlling gene expression in endodermis. The other predicted regulator for endodermis, TGA6, is known as a regulator that mediates Phytoprostanes inhibition of root growth [101]. Finally, based on bulk RNA-seq data, AtWRKY27 is predicted as a regulator only in developing cortex (importance score = 1, **supplementary table 2.8**), and has minor role in developing endodermis or maturing endodermis (importance score ~0.2, **supplementary table 2.8**). In contrast, AtWRKY27 was found to be active in both cortex and endodermis (importance score > 0.5) by single cell expression data. This result suggests the cortex population from single cell experiment could be more similar to developing cortex cell in bulk RNAseq results, whereas the endodermis population in single cell experiment is a mix of both developing and maturing endodermis cells.

2.3 Discussion

Including other genomic features in ConSReg. Apart from the genomic features we tested in this work, our prediction pipeline can be easily extended and applied to other types of expression data. The limitation of DAP-seq data is that some of the interactions detected by DAP-seq may not be active under specific *in vivo* conditions [19]. One solution to address this issue is to integrate DAP-seq data with more genomic features that confer *in vivo* binding specificity of TFs. Such genomic features may encompass 1) the activities of TFs and their target genes and 2) the chromatin accessibility for TF binding sites. The first criterion can be satisfied by including expression of TFs and targets into prediction pipeline. There are abundant expression data sets generated for model plant species Arabidopsis under different environmental perturbations, including drought [102], heat [103,104], cold [104–106], and salt stresses [104,107]. For the second criterion, ATAC-seq is used in the current analysis [17]. However, this ATAC-seq data is generated for seedling and roots. If additional tissue or condition specific data become available, our method can integrate these new genomic features. Other experimental approaches such DNase-seq and MNase-seq [19] can also be used in addition to ATAC-seq experiments.

Application to single cell expression data. The recent advancement of sequencing technology has enabled the investigation of gene expressions at single cell level. We have demonstrated application to scRNA-seq data in our analysis. To improve our current method, an important issue to be addressed is the sparsity of single cell data, which is usually characterized by zero-inflated read counts for the majority of genes [108]. A large portion of zero read counts may arise from technical noise or biological variability between single cells [108]. These phenomena are known as 'dropout' events. These dropout events give rise to a sparsity of gene expression values and thus results in limited number of DEGs for training, which can compromise model performance In previous work, attempts were made to address stochastic dropout by modeling it as a three component mixture model [109], two-component mixture linear model [110] or exponential function of expected expression [111]. To include more genes for training, our current method uses variable genes generated by Seurat [112] as positive genes, which were selected by finding the outliers on a mean variability plot. However, this selection process is based on normalized expressions which are not generated with an error model that explains dropout events. In future work, one potential improvement is to integrate error model into the process of selecting positive and negative genes.

Potential future improvement. While ConsReg achieved good performance (average ROC-AUC = 0.84), the tool can be further improved by either enhancing the model performance or by including data types that indicate dynamic regulation. Open chromatin regions have been reported to be both cell-type-specific [113,114] and condition-specific [115] as revealed by distribution of DNaseI

hypersensitive sites (DHSs). In our analysis, expression data and ATAC-seq data were not generated under the same conditions nor from the same tissue type. This is because only data from roots and seedlings are currently available for Arabidopsis [17]. We merged all open chromatin regions detected in two tissue types to maximize the discovery of potential interactions. This could introduce false positives and compromise the ability of the model to predict condition-specific interactions. In the future, such false positives can be reduced by integrating open chromatin data and expression data generated under the same conditions and same tissue type, as more data accumulate.

Another improvement could be made by altering the assumption of LRLASSO. Currently, our LRLASSO model assumes that the combinatorial regulations among TFs are identical for all DEGs (trained coefficients are identical for all DEGs). However, compared to real regulations, this is a simplistic assumption. To address this issue, we can either use information from other data types or cluster genes and adaptively fit local linear model to each cluster. In this case, the inferred combinatorial regulations may better represent the truth. We will leave this to future exploration in our follow-up work.

Additional possible functionalities. To better understand how condition- or cell-type-specific regulation changes across different condition/cell type, networks inferred by ConsReg can be compared. For example, when applied to single cell expression data, or bulk expression data with many time points, network comparisons can identify different regulation at different time points and how a given network dynamically changes over a time series. This would allow us to capture transient and dynamic combinatorial regulations. For cell-type-specific expression data, an effective strategy might be to investigate the specificity of network module(s) for each cell type or a group of cell types. Modules found in many cell types may characterize fundamental pathways and modules highly specific to few cell types may play unique functional roles.

2.4 Conclusions

In this study, we developed a novel computational tool, ConSReg. We performed comprehensive analyses to identify the factors that affect the performance of machine learning models and the optimal settings for constructing feature matrix. We performed a systematic recovery of N response TFs using ConSReg, TF2Network, PlantPAN 3.0, and Cistome. We showed that ConSReg generated better ranking results and recovered more N response TFs compared to other computational tools. We performed simulation of perturbation to TFs with high importance scores (importance score > 0.5) and found TFs with higher importance scores can be more predictive of gene expressions. Network analysis for the GRNs inferred by ConSReg revealed a novel combinatorial regulation between MYB44 and MYB77 in response to cold, heat, and drought stresses. We applied ConSReg to *Arabidopsis* scRNA-seq data of two root cell types (endodermis and cortex) and successfully identified regulators supported by existing publications. ConSReg provides a useful way of integrating currently available genomic features with published gene expression data to infer regulatory networks and to better understand mechanisms of gene regulation.

2.5 Methods

2.5.1 Preprocessing of genomic data sets

Bulk RNA-seq and microarray expression data. We re-analyzed published RNA-seq and microarray expression data for Arabidopsis from 20 experiments that were generated under different environmental and hormonal perturbations including cold, heat, drought, wounding, salt, osmosis, blue light, high light, far red light, abscisic acid (ABA), salicylic acid (SA), jasmonic acid (JA), auxin, and thermospermine oxidase (T-Spm) (See **supplementary table 2.2**). To validate the ConSReg predicted results and compare them to results generated from the TF2Network, we re-analyzed RNA-seq data for N treated Arabidopsis root [67] which is accessible under two GEO accession numbers: GSE117857 and GSE128209 (See **supplementary table 2.2** for more details). We also re-analyzed cell-type-specific gene expression in Arabidopsis root generated in our previous publication [116], and gene expression of cell-type-specific responses to salt stress in Arabidopsis root [28] (See **supplementary table 2.2**). The latter data set was generated using microarrays. We downloaded the pre-analyzed data files from the EBI expression atlas [117]. Processed data is available in the ArrayExpress database (http://www.ebi.ac.uk/arrayexpress) under accession number "E-GEOD-7641". We compiled **differential contrasts** for each experiment. Each differential contrast was defined as the contrast between a replicate group of treated samples and a replicate group of control samples (See **supplementary table 2.2** for all contrasts used in this study). We used whole root samples as the control group for the cell-type-specific expression experiment because no environmental perturbations were used in the experiment. For RNA-seq data, we used a published protocol to identify differentially expressed genes. In brief, we used STAR for read mapping, featureCounts for read counting and DESeq2 for differential expression analysis. Differentially expressed genes (DEGs) were identified as genes with FDR < 0.05. Microarray data were analyzed using protocol established at the EBI expression atlas. For both RNA-seq and microarray data, we used the average gene expression level across all the samples within one replicate group.

scRNA-seq expression data. We re-analyzed published Arabidopsis scRNA-seq data for two root cell types (endodermis and cortext) [99]. Expression data was downloaded from Gene Expression Omnibus (GEO) with accession number GSE116614 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE116614). The downloaded expression matrix contains read counts of each gene for each cell. Expression matrix was filtered by selecting cells that have minimally two expressed genes and genes that are expressed in more than one cell. Read counts were normalized using log normalization method in the Seurat package [112]. The normalized expressions were used to cluster cells by a graph-based clustering approach in the

Seurat package. The identified clusters were assigned with known Arabidopsis root cell types by computing index of cell identity (ICI) scores between each cluster and the profile of marker genes generated by a previous study [118]. ICI score characterizes the probability that each cluster represents a known root cell type. We assign the cell type of highest probability to the cluster. Next, we used clusters identified as endodermis and cortex cells to compute fold change of each gene. The SCDE package [109] was used to fit an error model of drop-out events and compute fold change of each gene with respect to the comparison of cortex versus endodermis.

TF-target interaction and open chromatin data. We downloaded BED files of peak regions for 387 TFs from a published Arabidopsis DAP-seq dataset [16]. All BED files are available at Plant Cistrome Database (http://neomorph.salk.edu/dap_web/pages/index.php). Interactions for the DAP-seq dataset were generated by assigning corresponding genes to each peak using R package ChIPseeker [119]. When the promoter region was set as 5kb upstream and 1kb downstream region of TSS, a total of 1,812,475 interactions were identified from DAP-seq peaks. Among these interactions, 1,540,984 interactions were generated from regular DAP-seq, where methylations were not removed from genomic DNA and 1,280,138 interactions were generated from amp-DAP-seq, where methylations were removed. These two types were merged, which resulted in a total of 1,812,475 non-redundant interactions. We also compiled a list of published TF-target interactions generated by literature curation and ChIP-seq [53], eY1H [60,120,121], and other methods. All interactions were merged and duplicates were removed, which resulted in a set of 1,866,371 interactions. The total number of interactions provided by DAP-seq account for 97.11% of the interactions (See **table 2.1**), therefore we only used the DAP-seq data for simplicity of data integration and feature construction. For the open chromatin region data, we downloaded Arabidopsis ATAC-seq peaks from a published study that identified open chromatin regions for whole seedlings and roots [17]. All BED file for the peaks are downloaded from GEO website (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE85203).

Table 2.1 Number of interactions collected from each platform.

Platfrom type	Number of interactions	Percentage
DAP-seq	1,812,475	97.11%
ChIP-seq	50,513	2.71%
eY1H	1,925	0.10%
Literature	1,431	0.08%
Total	1,866,344	100.00%

Evaluation dataset A and evaluation dataset B. We constructed different evaluation data sets. The reason for using different evaluation data sets is to provide sufficient positive/negative training

genes for machine learning and feature selection methods. For all expression experiments, we selected differential contrasts which can provide more than 500 positive and 500 negative genes for all three types of negative genes (NDEGs, LEGs, UDGs). We compiled these differential contrasts into evaluation dataset A (**Supplementary table 2.2**). After we determined that UDGs are better negative training sets than other two types of training sets, we selected differential contrasts that provide more than 500 positive and 500 negative genes for only UDGs and compiled them into evaluation dataset B. This data set was then used to evaluate the performance of integrating ATAC-seq data and performance of different types of DAP-seq data.

2.5.2 Feature construction

Based on expression data, we constructed differential contrasts between replicate group of control and treatment samples. Each replicate group typically includes expression data from multiple samples and each differential contrast produced a list of genes with fold change, mean expression value and p-value of differential expression. **Supplementary table 2.2** provides more details regarding the replicate group for each sample, and treatment and control information for each differential contrast. Next, we generated a feature matrix for each differential contrast. In our analysis, feature matrix was constructed by two steps. First, for each differential contrast, we generated a list of DEGs as positive training examples and sampled equal number of negative examples from the genome. The feature matrix X is a n by m matrix where n is the sum of number of positive examples and negative examples, and m is the number of TFs. In the second step, information from expression data, DAP-seq data and ATAC-seq data were integrated to construct X. Each entry X_{ij} in the feature matrix is computed by the following equations:

$$X_{ij} = F_j w(i,j) \quad (1)$$

$$w(i,j) = \sum_{k=1}^{l} len(O(D_{jik}, A_i))/len(D_{jik}) \quad (2)$$

Where j denotes jth TF and i denotes ith gene (either positive or negative gene). F_j is the log2 fold change value of TF j. In equation (1), $w(i,j)$ is the weight for each X_{ij}. In equation (2), D_{jik} denotes the kth DAP-seq peak region of TF j found in the promoter region of gene i. The weight $w(i,j)$ was computed by summing all l DAP-seq peak regions of TF j found in the promoter region of gene i. We evaluated each DAP-seq peak region by information from ATAC-seq, which was done by searching overlapping regions between each DAP-seq peak on gene i and all open chromatin regions on gene i (denoted by A_i), and the sum of length of overlapping regions was divided by the length of DAP-seq peak. This integration method will give higher weight $w(i,j)$ if

DAP-seq peaks for a TF j have more overlapping regions with the open chromatin regions found on gene i. $w(i,j)$ equals zero if no DAP-seq peaks of TF j can be found on gene i or no ATAC-seq peaks can be found on gene i or no overlapping regions were detected between them.

To efficiently search for all overlaps, we constructed an interval tree for ATAC-seq peaks in each chromosome then iterated over each DAP-seq peak to find all overlaps between DAP-seq peak and ATAC-seq peaks. Python package Intervaltree (https://github.com/chaimleib/intervaltree) was used to perform the search. While our current analysis only explored the use of DAP-seq interaction data and ATAC-seq open chromatin region data, other types of interaction data and chromatin feature data can be easily integrated into equation 1 and equation 2. We will leave this to future exploration.

To construct feature matrices with only DAP-seq data, we marked each entry X_{ij} by '1' if binding site(s) of TF j are found in promoter region of gene i and '0' if not.

We normalized the feature matrices by min-max normalization. Each X_{ij} was normalized by:

$$X'_{ij} = \frac{X_{ij} - \min(|X|)}{\max(|X|) - \min(|X|)}$$

Where $\min(|X|)$ is the smallest absolute value in feature matrix X and $\max(|X|)$ is the largest absolute value in feature matrix X. During cross-validation, we computed $\min(|X|)$ and $\max(|X|)$ from training feature matrix and used them to normalize validation feature matrix and testing feature matrix.

2.5.3 Machine learning models and feature selection

We tested several machine learning methods for classification, including logistic regression (LR), support vector machine (SVM), random forest (RF) and deep neural network (DNN). To perform feature selection, we applied different regularization techniques to each classifier. The details of classification and feature selection methods are described below.

LRLASSO. This method is logistic regression with lasso penalty, which uses L1 regularization for feature selection [122]. LRLASSO minimizes the following loss function:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} -L(y_i, \hat{y}_i) + \lambda \sum_{j=1}^{m} |\beta_j|$$

Where y_i and $\widehat{y_i}$ are the true label and predicted label for each training example, respectively. $\widehat{y_i}$ is estimated by the logistic function:

$$\widehat{y_i} = \frac{1}{1 + e^{\sum_j^m x_{ij}\beta_j}}$$

$L(y_i, \widehat{y_i})$ is the log likelihood function and $\lambda \sum_{j=1}^{m} |\beta_j|$ is the L1 penalty term. β_j is the coefficient for feature j (In our analysis, TF j). $L(y_i, \widehat{y_i})$ is usually calculated by the cross-entropy loss function:

$$L(y_i, \widehat{y_i}) = y_i \log(\widehat{y_i}) + (1 - y_i)\log(1 - \widehat{y_i})$$

To perform feature selection, we tuned the L1 penalty parameter λ for this model using the R package gglasso [123]. Given a sequence of ordered λ values, gglasso computes the solution for each λ iteratively. The computed solution for the current λ will be used as the initial value for next λ in the sequence. For each round of cross-validation, we used a sequence of 100 λ values that ranged from min(λ) to max(λ) and were spaced evenly on a log scale. max(λ) is the smallest λ value that shrinks all coefficients to zero for a given feature matrix. And min(λ) = η * max(λ), where η is a factor specified by user. For more details, please see the publication of gglasso [123] and the online documentation of the R pacakge (https://cran.r-project.org/web/packages/gglasso/gglasso.pdf). In this way, each λ generates a LRLASSO model by training on training data set and the model was evaluated using validation data set to determine which λ gave the best prediction accuracy. Then the λ and the model with best prediction accuracy was again evaluated by the test data set.

LGLASSO. This method is logistic regression with group lasso penalty. The group lasso penalty regularizes the coefficients $\beta_1, \beta_2 \ldots \beta_m$ by grouping and summing them using L2 norm [123,124]. If the m features can be grouped into l groups, the loss function of LGLASSO can therefore be written as:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} -L(y_i, \widehat{y_i}) + \lambda \sum_{k=1}^{l} w_k ||\beta^{(k)}||_2$$

Where $||\beta^{(k)}||_2$ is the L2 norm of all βs in group k and w_k is the weight for group k. We followed the default choice in gglasso paper [123]. We set $w_k = \sqrt{p_k}$, where p_k is the number of features in group k. To obtain the prior grouping information for TFs, we ran CoReg [94] on DAP-seq interaction network to identify co-regulator groups and used these groups in LGLASSO. For hyperparameter tuning and search, we used the same approach with LRLASSO.

LREN. This method is logistic regression with elastic net penalty, which uses L1 + L2 regularization for feature selection. The loss function for LREN can be written as:

$$\min_{\beta} \frac{1}{n}\sum_{i=1}^{n} -L(y_i, \hat{y}_i) + \alpha\rho \sum_{j=1}^{m}\left|\beta_j\right| + \frac{\alpha(1-\rho)}{2}\sqrt{\sum_{j=1}^{m}|\beta_j|^2}$$

Where $\alpha\rho\sum_{j}^{m}\left|\beta_j\right|$ is the L1 penalty term and $\frac{\alpha(1-\rho)}{2}\sqrt{\sum_{j=1}^{m}|\beta_j|^2}$ is the L2 penalty term. The parameter α and ρ control the strength of L1 penalty and the ratio of L1 penalty, respectively. Therefore, these two parameters are the only hyperparameters for LREN. For α, we used a sequence of five α values which range from 10^{-3} to 10^{3} and are evenly spaced on a log scale. For ρ, we used 0, 0.25, 0.5, 0.75 and 1 for tuning. Then grid search was performed to find the best combination of α and ρ. We used the function SGDClassifier() from scikit-learn package to perform training for LREN.

LRPCC. This method is logistic regression with Pearson correlation coefficient (PCC). We computed the PCC between each feature and the true labels across all training examples. These features were then ranked using the PCC values by descending order. Similar to the process of stepwise linear regression, we iteratively added top *k* features at a time to train the model [125] and the best set of features were determined by calculating accuracy of the trained model on validation data set. We set *k* as 50 to efficiently train the model. We used the function LogisticRegression() from scikit-learn package to perform training for LRPCC.

GRRF. This method is guided regularized random forest. GRRF first trains a RF model and then uses the importance scores computed from this model to guide the feature selection process [126]. Briefly, GRRF computes a normalized importance score for each feature based on the original importance score from RF model. GRRF also assigns a penalty coefficient λ_i ($\lambda_i \in (0,1]$) to each newly introduced feature when Gini information gain is calculated to split tree nodes. This is to penalize the use of new feature if it has not been used for splitting the node. λ_i is calculated by:

$$\lambda_i = 1 - \gamma(1 - Imp'_i)$$

Where γ is the only hyperparameter that controls the strength of regularization. For more details about the algorithm of GRRF, please see the reference [126]. Since GRRF is more computationally expensive than other linear model-based methods and SVM-based method, we only explored γ = 0, 0.5 and 1 for hyperparameter tuning.

LSVM. This method is linear support vector machine with L1 regularization. The only hyperparameter for this model is C, which controls the strength of L1 regularization. Smaller C will

apply stronger regularization to the model, leading to a sparser model with many coefficients shrunk to zero. We used the function LinearSVC() from scikit-learn to perform training for LSVM. In our analysis, we used a sequence of C values which ranges from min(C) to 10^3 and are evenly spaced on a log scale. min(C) is the minimum value of C which ensures all coefficients will be non-zero. min(C) is computed by calling the function l1_min_c() from scikit-learn library. Then the best C was determined by training the model on training data set and evaluating on validation data set.

DNN. This method is deep neural network with L1 regularization for feature selection. The use of regularized DNN for genomic feature selection has been investigated in a previous publication [75]. The authors added a one-to-one layer between the input layer and hidden layers. L1 and L2 regularization were applied to the one-to-one layer to select features. Due to the high computational cost of tuning hyperparameters of DNN, we chose to use only L1 regularization in the one-to-one layer and hidden layers. We used a similar DNN architecture with the previous publication [75]. In the input layer, there are 387 neurons and this number is equal to the number of input features (TFs). In the second layer (one-to-one layer), the same number of neurons are used and each is connected to one neuron from the input layer. Then we added two hidden layers which have 32 and 16 neurons after the one-to-one layer. The first hidden layer is fully connected with one-to-one layer and second hidden layer is fully connected with the first hidden layer. The last layer is an output layer which only has one neuron. Batch normalization was applied to one-to-one layer and each hidden layer to accelerate the training process. See [75] for more details about using DNN to select features.

For hyperparameter tuning, we tuned L1 regularization parameter λ for DNN model. We used a sequence of 10 λ values which range from 10^{-6} to 10^3 and are evenly spaced on a log scale. Adam optimizer was used to train the DNN model and learning rate α was fixed as 0.1. We compiled and trained DNN model using Keras library (https://keras.io/) with CUDA GPU acceleration. Training, hyperparameter tuning and testing was performed in the same way as described in other methods.

We provide ConSReg as a Python library for model training, tuning and testing using the models described in this study (github link https://github.com/LiLabAtVT/ConSReg).

2.5.4 Evaluation strategy

Evaluating different conditions. To evaluate the effect of selecting negative training examples, we tested three different methods: 1) non-significantly differentially expressed genes (NDEGs), which have p-value > 0.05 2) low-expressed genes (LEGs), which have mean expression between 0 and 0.5. 3) undetected genes (UDGs), which have mean expression value equal to zero. To evaluate

the effect of promoter region length, we constructed feature matrices using three different promoter lengths, which are 1) 5kb upstream of TSS to 1kb downstream of TSS; 2) 3kb upstream of TSS to 0.5kb downstream of TSS; and 3) 0.5kb upstream of TSS to TSS. The promoter region length is passed as input argument to ChIPseeker package to search for corresponding genes for each DAP-seq peak. To evaluate the effect of regular DAP-seq peaks VS merged DAP-seq peaks, we constructed the feature matrices using the DAP-seq peaks from regular DAP-seq (methylated DAP-seq peaks) and the DAP-seq peaks from merged DAP-seq peaks. The performance of two methods were then compared.

Cross-validation for the models. For each feature matrix, we randomly split the matrix into three subsets: 60% for training, 20% for validation (hyperparameter tuning) and 20% for testing. We trained the machine learning models on training data set and found the optimal set of hyperparameters by evaluating the trained model on validation data set (**Figure 2.1B**). Then the final performance of model with optimal hyperparameters was evaluated using the test data set. We used AUC-ROC and AUC-PRC as the metrics for evaluation. This process was repeated five times for each feature matrix to obtain the mean and standard deviation of AUC-ROC and AUC-PRC.

Compare performance to enrichment-based method. We compared our methods to enrichment-based method. Similar to the approach used in TF2Network [62], we computed the statistical significance of enrichment for each individual TF by hypergeometric test. The probability mass function is defined as:

$$P(x = i) = \frac{\binom{N}{i}\binom{M-N}{n-i}}{\binom{M}{N}}$$

Where each parameter is explained below:

i is the number of DEGs that have DAP-seq peak(s) of the current TF.

N is the total number of DEGs in the current differential contrast.

n is the total number of protein-coding genes that have DAP-seq peak(s) of the current TF.

M is the total number of protein-coding genes.

p-values for all TFs were then computed by hypergeometric test and corrected by Benjamini-Hochberg correction [127]. We used the same training set of positive genes and negative genes to compare LRLASSO with enrichment-based method. For each condition, this training set is the same feature matrix we used to evaluate machine learning models

To calculate AUC-ROC value for the enrichment-based method, we first ranked all TFs by ascending order using corrected p-values. Then we iterated over the ranked list of TFs. In each iteration, we used top kTFs as predictors and k is increased by one in next iteration until all 387 available TFs were included as predictors. Gene is considered as predicted positive if it has any predictors' peak regions in its promoter region and predicted negative if not. Therefore, false positive genes are those predicted as positive but are negative in the training set and false negative genes are those predicted as negative but are positive in the training set. We calculated false positive rate $\frac{FP}{N}$ and false negative rate $\frac{FN}{P}$ in each iteration and then all points of $(\frac{FP}{N}, \frac{FN}{P})$ were put together to construct ROC curve for computing AUC-ROC value.

Since hold-out test will not be applicable for enrichment-based method, for both LRLASSO and enrichment-based method, training and testing were performed using the same training set to have fair comparison.

2.5.5 Stability selection and computation of importance score

Since coefficients generated by LRLASSO model do not reflect the importance of each TF and the selected set of TFs would be slightly different when coefficients are initialized randomly. We applied stability selection [70] to generate robust feature selection result from LRLASSO.

Randomized lasso was proposed as an implementation of stability selection for lasso method. The difference between randomized lasso and regular lasso is that subsampling of training examples and random perturbations for features are introduced into the feature selection process [70]. Briefly, a subset of training examples were selected and their features were randomly perturbed. Then a lasso model was trained using the perturbed subset of original training data set. This process was then repeated multiple times. The idea is that important features will be selected more often than the unimportant ones during this randomized process. When used with LRLASSO, the objective function of randomized lasso can be written as:

$$\min_{\beta} -L(y_i, \hat{y_i}) + \lambda \sum_{j=1}^{n} \frac{|\beta_j|}{w_j}$$

Where $L(y_i, \hat{y_i})$ is the log likelihood function as described previously in **Methods** section. β_j is the coefficient for feature j. $\lambda \sum_{j=1}^{n} \frac{|\beta_j|}{w_j}$ can be considered as the penalty term for randomized lasso, similar to L1 penalty term for regular lasso model. The only difference here is that random

perturbation is introduced by w_j, a scaling factor sampled from the range (0,1]. For simplicity of implementation, features can be rescaled to have the same effect with rescaling the coefficients [70].

In our analysis, we randomly sampled half of the training examples from a feature matrix. Features were randomly perturbed by a scaling factor randomly sampled from (0,1]. Randomized lasso was performed *n* times for each feature matrix. For each feature, the final importance score was calculated as number of times the feature gets non-zero coefficient divided by *n*. In our analysis, we set $n = 200$.

2.5.6 Network inference

GRN was inferred by computing the score for each TF using LRLASSO + stability selection and connecting the selected TFs to corresponding target genes. The detailed steps are described in this section.

First, we computed the importance score for each TF in each differential contrast. There could be multiple feature matrices available for a single condition and each feature matrix corresponds to a differential contrast between treatment group and control group. For each of these differential contrasts, LRLASSO + stability selection was run and TFs with importance score > 0.5 were selected as regulators in GRN.

Second, with the selected regulators for each differential contrast, we constructed the GRN. If the selected TF j has a non-zero entry X_{ij} in feature matrix X, TF j is considered as having impact on target gene i. Then TF j will be linked to target gene i. To capture the most active regulatory interactions, we limit target genes to be only DEGs in each differential contrast. This process would generate multiple networks and each of them corresponds to a differential contrast. We then merged all networks generated in a single condition to build a final GRN for that condition.

2.5.7 Computation of p-values for co-expression analysis

p-values were computed using Fisher's Z-transformation as described in [88]. This method first computes a Z-score for each expression correlation by the equation below:

$$\mathrm{Z}(X,Y) = \frac{\sqrt{N-3}}{2}\ln\left(\frac{1+\mathrm{r}(X,Y)}{1-\mathrm{r}(X,Y)}\right)$$

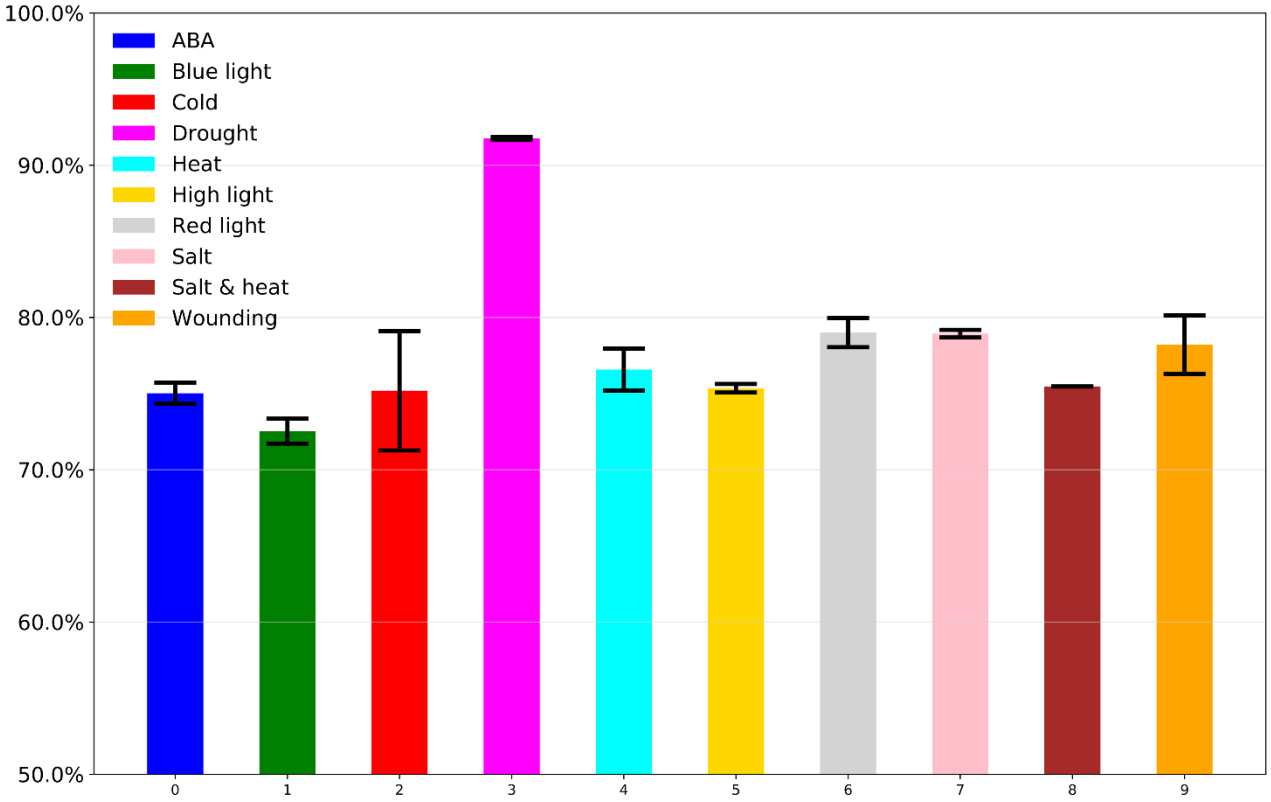
Where X, Y are two vectors of gene expressions and *N* is the number of samples used in co-expression analysis (dimensions of the vectors). $\mathrm{r}(X,Y)$ is the expression correlation between two genes. In this work, Pearson Correlation Coefficient (PCC) was used to calculate $\mathrm{r}(X,Y)$.

The distribution of Z(X, Y) approximately follows a standard normal distribution [88]. Therefore, p-value for the corresponding Z-score can be calculated using standard normal distribution.

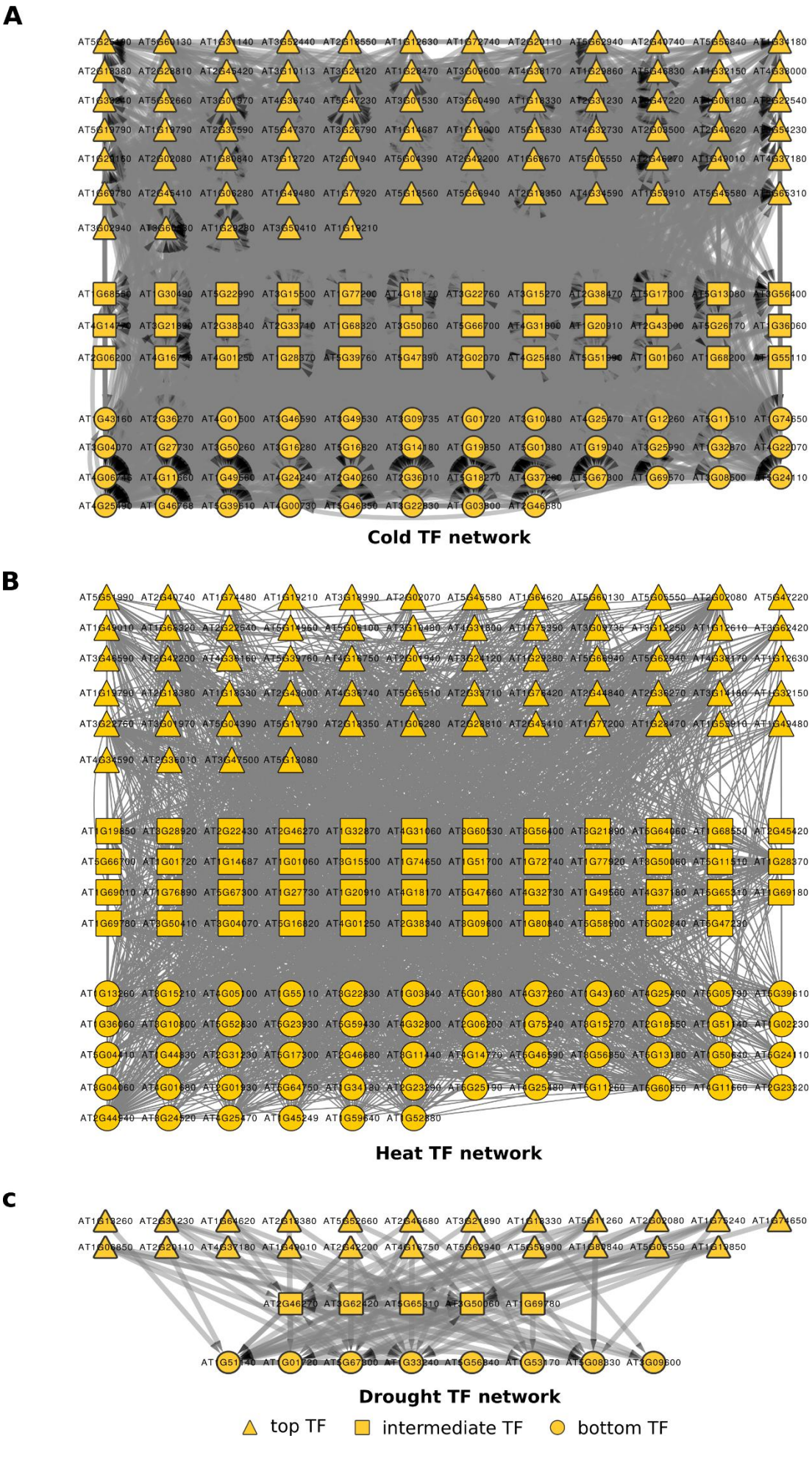
2.6 Authors' contribution

SL conceived the idea. SL and QS designed the experiments. JL and SA prepared the expression data sets. QS developed the ConSReg package and performed all the analysis. SL, QS and RG wrote the manuscript.

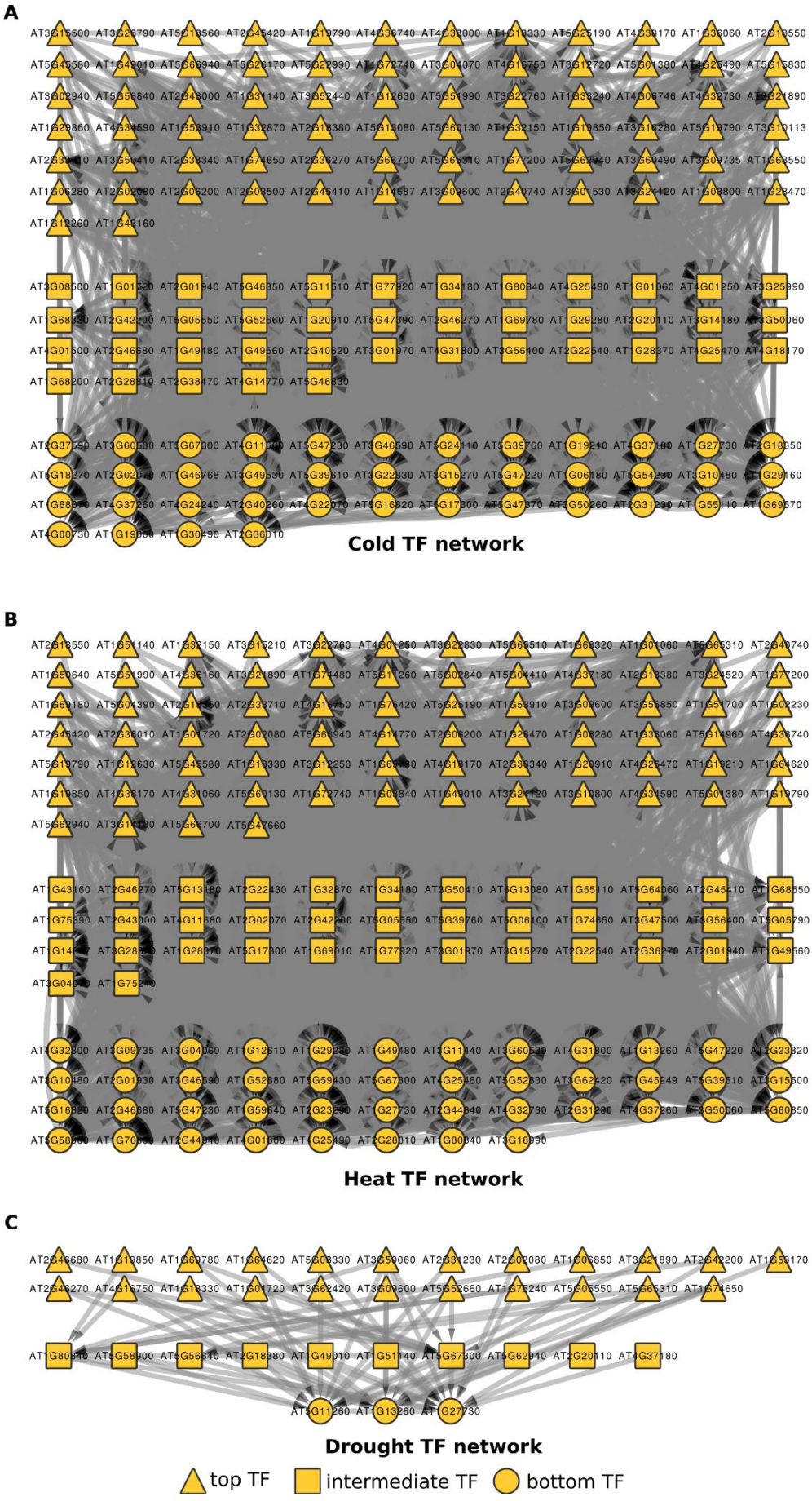
2.7 Supplementary figures



Supplementary figure 2.1 Condition specificity of negative training genes. Data sets generated from environmental perturbations were marked by different colors. For each differential contrast in each environmental perturbation, we computed the percentage of UDGs that are detected (fpkm > 0) in other perturbations. Then the percentages were averaged for each environmental perturbation. Error bars indicate the standard deviations.


A
Cold TF network
B
Heat TF network
C
Drought TF network
top TF
intermediate TF
bottom TF

Supplementary figure 2.2 Subnetworks formed by UR TFs extracted from GRN generated for heat, cold and drought environmental perturbations. **Triangles**: top TF cluster, in which TFs have many out-going edges and less in-coming edges from other TFs. **Squares**: bottom TF cluster, in which TFs have many in-coming edges and less out-going edges from other TFs. **Circles**: intermediate TF cluster, in which TFs have balanced out-going edges and in-coming edges from other TFs.



Supplementary figure 2.3 Subnetworks formed by DR TFs extracted from GRN generated for heat, cold and drought environmental perturbations. **Triangles**: top TF cluster, in which TFs have many out-going edges and less in-coming

edges from other TFs. **Squares**: bottom TF cluster, in which TFs have many in-coming edges and less out-going edges from other TFs. **Circles**: intermediate TF cluster, in which TFs have balanced out-going edges and in-coming edges from other TFs.

3. Chapter 3. Identification of regulatory modules in genome scale transcription regulatory networks

This chapter is reproduced from a published study [128]

Qi Song, Ruth Grene, Lenwood Heath, Song Li

Abstract

Transcription factors function as co-regulators to regulate expression of their target genes. Existing module-finding algorithms can identify densely connected genes but not co-regulators in regulatory networks. Here, we presented a computational tool, CoReg, to identify co-regulators in large-scale regulatory networks. Using simulated and real networks, we found CoReg outperforms other published methods in identifying co-regulatory genes. We applied CoReg to a large-scale network of *Arabidopsis* with more than 2.8 million edges and found that many regulatory modules with common in-coming edges tend to be highly co-expressed, suggesting that target modules are structurally stable module in abiotic stress conditions.

Keywords

Co-regulation; network module; regulatory network; systems biology

3.1 Introduction

Characterization of the structures of gene regulatory networks is an essential step towards understanding the transcription regulation in living organisms. In recent years, genome scale regulatory networks have become available for many species [14,25,93,121,129,130]. In the human Encyclopedia of DNA Elements (ENCODE) project, Transcription Factors (TF)-target interactions for 119 human TFs have been identified using Chromatin Immunoprecipitation followed by sequencing (ChIP-seq) [93]. In the model plant species *Arabidopsis thaliana*, cell type-specific regulatory networks in xylem and ground tissues were generated using enhanced yeast one-hybrid (eY1H) for 267 TFs [14,121]. Genome-scale TF-target interactions can also be inferred from direct sequencing of TF binding site *in vitro* [16] and measuring TF binding specificity [131] More recently, integration of interaction data sources has provided convenient access to TF-target interaction data for researchers [132]. Co-expression and function association based prediction is another approach to infer TF-target interactions, which has been successfully applied in a recent online tool, TF2Network [62]. These experimentally identified or predicted gene regulatory networks typically contain thousands of nodes and thousands to millions of edges (**Figure 3.1A**) [16,93], which provide enormous amount of information regarding the regulatory targets of each TF and the putative regulators of each gene in the genome. The key challenge is how to use these large-scale networks to identify functional information for both TFs and their target genes.

One way to approach this problem is to find cluster of genes with similar regulatory properties [133–135]. In *Arabidopsis*, it has been shown that identification of co-regulatory modules can provide insight into biological functions [121]. For example, analysis of regulatory network showed that two key transcription factors (SHORTROOT and SCARECROW) that determine cell fates in ground tissues are controlled by both activators and repressors [121]. Co-regulatory targets of stress-responsive transcription factors were found to be key regulators of ABA responses [15]. In general, a regulatory network is represented by a directed graph and the process of identifying clusters of nodes (regulatory modules) with similar network properties is called network module finding [18] or modular decomposition [136]. Many computational approaches have been developed for module finding, and these approaches are based on various ways to calculate node similarities followed by graph partitioning methods. For example, Walk Trap (WT) [137] calculates the distance between the nodes and group nodes based on pairwise similarity matrices; Edge Betweenness (EB) [138] builds hierarchical relationship between the nodes, and partitions the network into modules; Label Propagation (LP) [139] performs simulation on the network by propagating cluster labels. Other examples include leading eigenvectors [140] and spin-glass [138]. These algorithms can be applied to either undirected networks [137–140] or directed networks [18] and in many cases, performed well in

finding group of densely connected nodes. However, densely connected group may not reflect biologically meaningful clusters in regulatory networks. For example, in **Figure 3.1B**, gene A and gene B are biologically related because they regulate the same target genes and will be expected to form one network module, which is not the typical module that most clustering approaches are designed to identify (**Figure 3.1B**).

Here, we propose a new computational tool, CoReg, to identify co-regulatory modules in genome scale regulatory networks. CoReg calculates the similarity of genes based on their common targets and regulators and groups highly similar genes into co-regulatory modules. We compared several similarity indices, including the Jaccard index, the geometric index, and the inverse log-weighted similarity index using simulated and real networks (**Figure 3.1C and D**). We tested CoReg on simulated networks with different parameters. We performed extensive rewiring-simulation and tested CoReg on plant, human, and bacterial networks. CoReg outperformed other commonly used module finding methods in identifying co-regulatory modules in all data sets tested. We identified many co-regulatory modules in *Arabidopsis* genome and demonstrated that the expression levels of genes in some of the modules are also highly correlated. Finally, we applied CoReg to published gene expression data in *Arabidopsis* roots and found that genes co-regulatory modules tend to be highly co-expressed in abiotic stress conditions. CoReg is implemented as an R package, which can be used to analyze any regulatory network. Sample network data used in this paper and CoReg package can be downloaded from GitHub (https://lilabatvt.github.io/CoReg/).

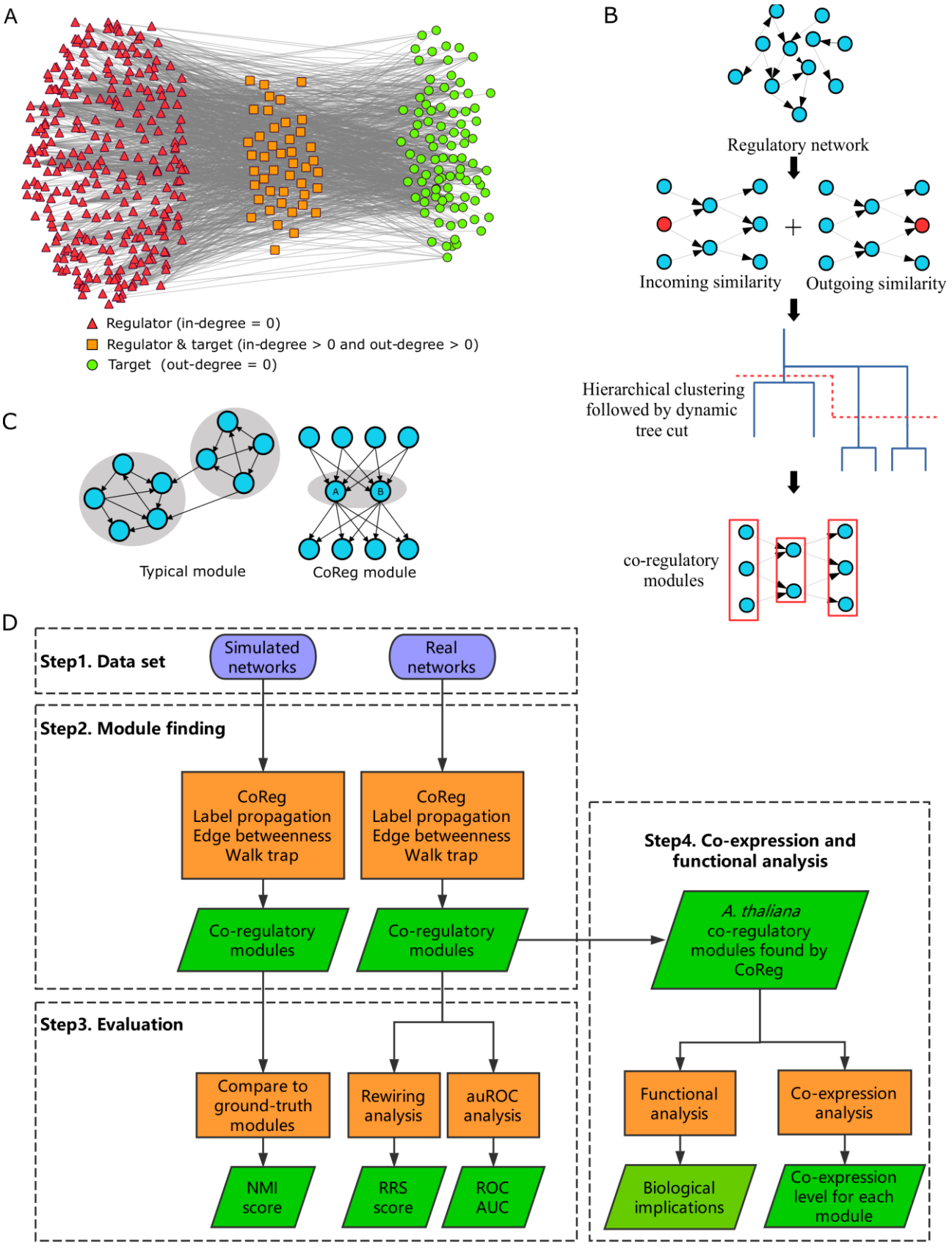


Figure 3.1 The complexity of the *A. thaliana* regulatory network, two clustering strategies and the work flow of CoReg. **(A)** The complexity of regulatory *A. Thaliana* network. Each node represents one gene in the network and each edge represents an interaction between one TF and its target. We classified the nodes into three categories based on the degree: 1) triangle, in-degree = 0; 2) rectangle, in-degree > 0 and out-degree > 0; 3) circle, out-degree = 0. **(B)** The brief work flow of CoReg starting from input (a regulatory network). Red nodes in the second step represent

common target (for out-similarity) or regulator (for in-similarity) for the pair of nodes in the middle. CoReg adds up the incoming similarity and outgoing similarity and then calculates a distance matrix. Next, distance matrix is used as the input to hierarchical clustering. In the last step, dynamic tree cut is performed to obtain final module assignment for each node. **(C)** CoReg uses a clustering strategy different from existing clustering method. Typically, the network modules that normal clustering algorithm identifies are shown on the left. However, if there are two genes which share many targets and regulators in common, they are most likely to be the actual co-regulators (shown on the right, gene A and gene B) CoReg is designed to work on the clustering problem on the right. **(D)** Flowchart of analysis in this study.

3.2 Results

3.2.1 Assessment of Different Module Finding Methods

There are typically two approaches to evaluate a computational method: using either existing biological knowledge or using computational simulations as a "gold standard". Since there has not been a systematic study that summarizes known co-regulatory modules in any species, we performed computational evaluations in two ways: 1) we generated simulated networks with pre-specified module assignment for each node and evaluated different methods using mutual information; 2) We performed duplication-rewiring simulation on real networks and evaluate using using receiver operating characteristic (ROC) curves and rewiring recall score.

3.2.2 Performance assessment using simulated networks

We generated the simulated networks using a method described in a previous publication [141]. We modified this approach to generate co-regulatory modules for directed networks (See **3.5 Methods**). Simulated networks were generated using different combination of parameters to explore the performance of algorithms in varying module size and number of targets (see **3.5 Methods** for details). Briefly, each regulator node was assigned to predefined modules. A pool of candidate targets was selected for each module and each regulator node can link to a node either in the pool or a node outside the pool. This procedure was repeated until target nodes for each module are assigned. One of the key parameters is "*prob*". With higher *prob*, the generated modules will have a stronger co-regulation pattern, characterized by nodes in co-regulatory modules preferably connecting to a small group of nodes rather than random targets in the network (**Supplementary Figure 3.1**). We then tested different module-finding algorithms on the simulated network. The performance was evaluated by comparing the algorithm identified modules to the pre-specified "ground truth" using Normalized Mutual Information (NMI) score [142].

The NMI score (see **3.5 Methods**) between the pre-specified modules and algorithm identified modules was plotted against the co-regulation probabilities. We plotted NMI score curve for each similarity index used by CoReg: 1) CoReg with inverse log weighted similarity index,

(CoReg+inv); 2) CoReg with jaccard index (CoReg+jaccard) and 3) CoReg with geometry similarity index (CoReg+geo). The three methods were compared to the result generated by Walk Trap (WT), Label Propagation (LP) and Edge Betweenness (EB). **Figure 3.2** shows the NMI score curves under different parameters. In all the simulations, NMI score increases as the co-regulation probability raises, indicating that algorithms perform better when network shows stronger co-regulation pattern. This trend is apparent for all the methods tested using simulated networks except for CoReg+inv. The NMI scores for CoReg+inv decrease when co-regulation probability is greater than 0.6 (**Supplementary Figure 3.2**), which is due to the high similarity between target genes (see **Discussion**). These results showed that CoReg+jaccard and CoReg+geo consistently outperformed the other methods (**Figure 3.2**).

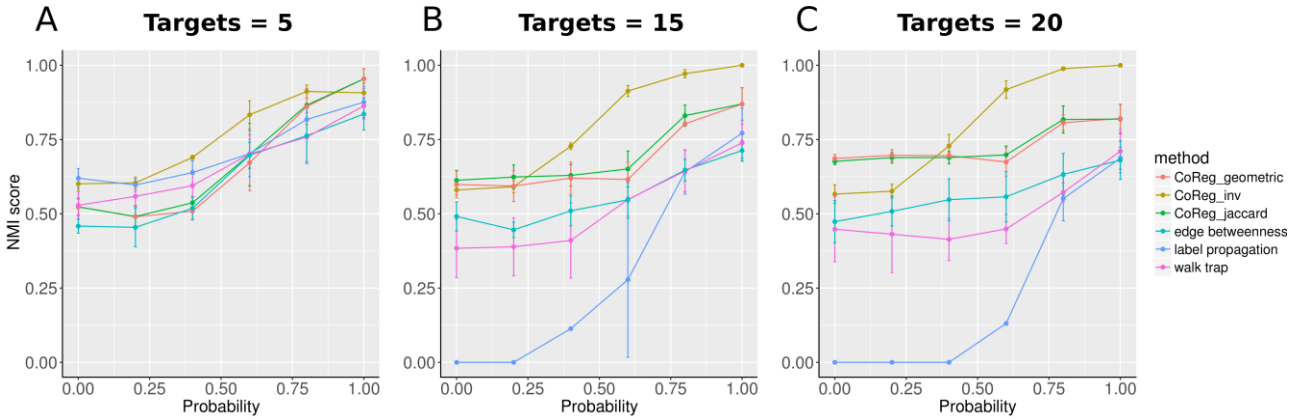


Figure 3.2 Evaluation of different module-finding methods using simulated networks. Each data point is the average score of five runs. We constructed simulated networks with module size of 5. Simulation result of other tested parameters are in **Supplementary Figure 3.2**. **(A)** Number of targets is equal to 5. **(B)** Number of targets is equal to 15 **C)** Number of target is equal to 20.

3.2.3 Performance assessment using real networks

For real networks, we designed our simulation such that the simulated networks are based on known topology of biological networks. We used published regulatory networks from human, *Arabidopsis* and *Escherichia coli* (*E. coli*) as the starting point for our simulations (**see 3.5 Methods**). In each simulation, we selected a subset of regulators and duplicated those genes, while preserving their neighbors in the network. We then rewired the network with a pre-specified probability to introduce noise to the network (for more details, see **3.5 Methods** and **Supplementary Figure 3.5**). For each species, we tested three rewiring probabilities (0.1, 0.3 and 0.5). In the simulated networks, a gene and its duplicated counterpart belong to the same co-regulatory module, and these genes are used to evaluate algorithm performance using receiver operating characteristic (ROC) and area under the ROC curves (auROC).

For each species, we plotted the ROC curves for CoReg+inv, CoReg+jaccard and CoReg+geo. These similarity indices were compared to the similarity index computed from Walk Trap (**WT**). **LP** and **EB** are not used here because these two methods do not calculate similarity matrix and cannot be directly compared. WT allows the user to specify the length of random walks. We tested WT with steps = 2 and steps = 4. **Figure 3.3A**, **B** and **C** show the ROC curves generated from the three species with rewiring probability = 0.5 where CoReg outperforms WT in all simulations. ROC curves for the three similarity indices have very similar performance. The AUC values of CoReg+jaccard and CoReg+geo are always slightly higher than that of CoReg+inv (**Table 3.1**).

Table 3.1 AUC for CoReg with different similarity index and Walk Trap.

Species	Rewiring probability	CoReg_inv	CoReg_jaccard	CoReg_geometric	WT(4 steps)	WT (2 steps)
A. thaliana	0.1	0.859 ± 0.035	0.957 ± 0.017	0.972 ± 0.014	0.597 ± 0.049	0.556 ± 0.033
A. thaliana	0.3	0.823 ± 0.031	0.939 ± 0.010	0.910 ± 0.036	0.548 ± 0.048	0.526 ± 0.027
A. thaliana	0.5	0.794 ± 0.035	0.872 ± 0.034	0.866 ± 0.031	0.531 ± 0.027	0.505 ± 0.038
E. coli	0.1	0.849 ± 0.048	0.968 ± 0.022	0.963 ± 0.013	**0.611** ± 0.030	**0.971** ± 0.020
E. coli	0.3	0.813 ± 0.037	0.930 ± 0.023	0.919 ± 0.027	0.557 ± 0.035	0.827 ± 0.034
E. coli	0.5	0.819 ± 0.033	0.865 ± 0.024	0.878 ± 0.039	0.555 ± 0.028	0.683 ± 0.060
H. sapiens	0.1	0.914 ± 0.037	**0.999** ± 0.001	**0.999** ± 0.002	0.500 ± 0.033	0.512 ± 0.025
H. sapiens	0.3	**0.919** ± 0.035	0.994 ± 0.007	0.991 ± 0.008	0.516 ± 0.016	0.489 ± 0.025
H. sapiens	0.5	0.918 ± 0.032	0.983 ± 0.010	0.983 ± 0.008	0.512 ± 0.027	0.503 ± 0.025

* CoReg_inv: CoReg+inverse log weighted similarity index
  CoReg_jaccard: CoReg+jaccard similarity index
  CoReg_geometric: CoReg+geometry similarity index
  WT: Walk Trap

3.2.4 Rewiring recall score

The second step in finding co-regulatory modules is node clustering. To assess and compare the performance of different clustering methods, we calculated rewiring recall scores (**RRS**) for all clustering methods and compared the results obtained using different methods. The rewiring recall score is a normalized measure of the accuracy of the method. For an ideal clustering method, each

duplicated node and its original node should belong to the same module with only these two nodes in this module. The RRS is designed to equal to 1 under such ideal cluster assignment (see **3.5 Methods**). If a method can find a module containing both the duplicated node and its original node, but also includes other nodes, the score will be smaller than 1 (see **3.5 Methods**). In our simulations, RRS rarely equals 1, because if two genes are regulating the same set of targets in the original network, the duplicated simulation will introduce another gene that is highly similar to both genes. In this situation, the RRS cannot equal to 1 for the correct clustering. Despite this limitation, RRS can be used to compare relative performance between different methods.

In the *A. thaliana* network, CoReg+geo index and CoReg+jaccard index outperformed all other clustering methods and their performances are similar to each other (**Figure 3.3D**). In the *E. coli* network, both the CoReg+geo index and the CoReg+jaccard index have better performance than other methods when rewiring probability is equal to 0 and 0.1. However, as the rewiring probability increases, performance of CoReg+jaccard and CoReg+geo drops much faster than that of CoReg+inv **(Figure 3.3E**) and CoReg+inv started to outperform CoReg+jaccard and CoReg+geo when rewiring probability equals 0.2. In the *H. sapiens* network, the decreasing trend of performance is not very obvious as compared to the other two species, presumably due to the large size of the human network. Although no any single similarity index performed better than all others in all species, in our simulations, CoReg+jaccard and CoReg+geo outperformed CoReg+inv more often than CoReg+inv outperformed CoReg+jaccard and CoReg+geo. In the *A. thaliana* and *E. coli* networks, CoReg+jaccard outperformed CoReg+geo.

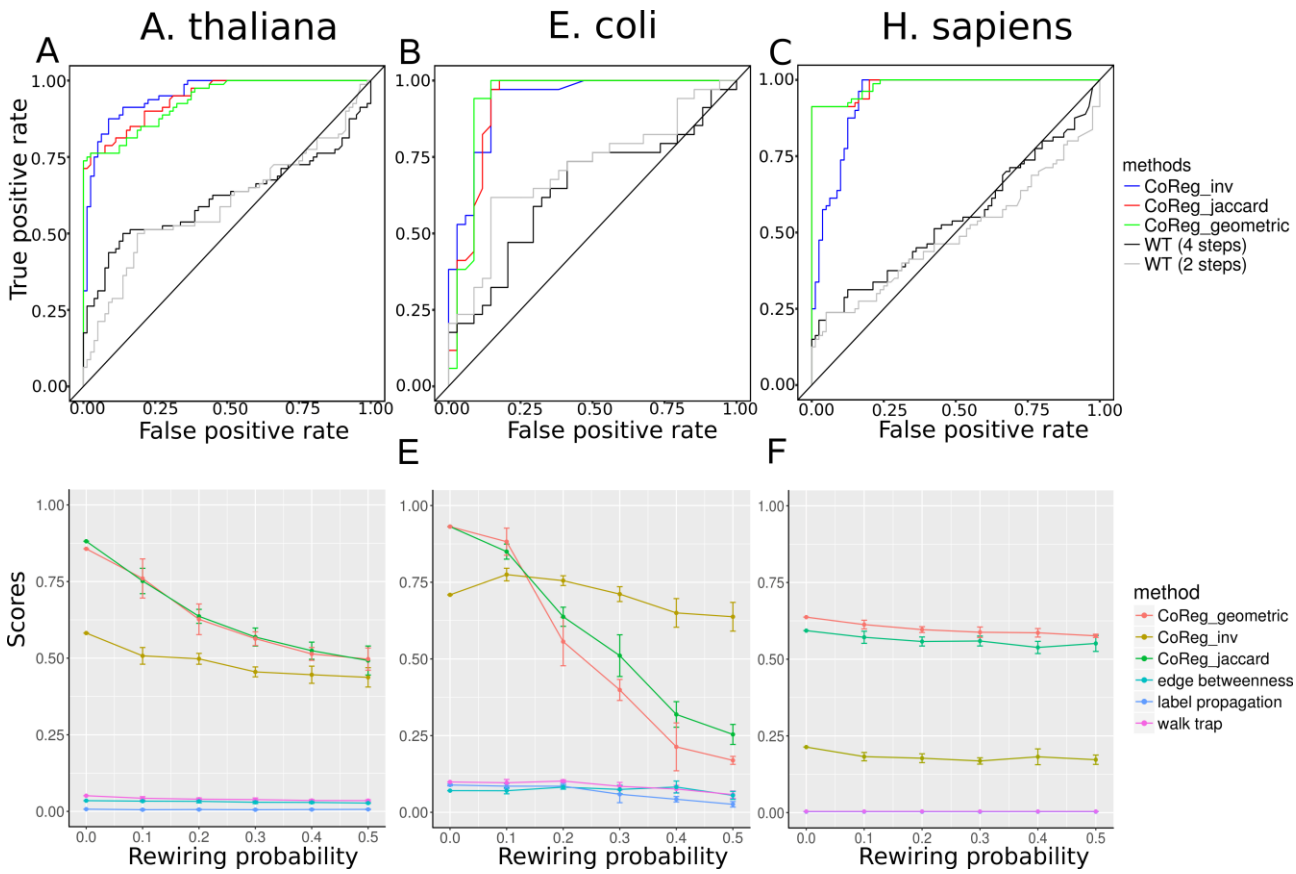


Figure 3.3 Evaluation of the different module-finding methods using real networks. We used different similarity indices for CoReg, CoReg_inv: CoReg+inverse log weighted similarity index; CoReg_jaccard: CoReg+jaccard similarity index; CoReg_geometric: CoReg+geometry index. CoReg was also compared to other three clustering algorithms, namely, Label Propagation (LP), Edge Betweenness (EB), Walk Trap (WT). We performed the evaluation on *A. thalina*, *E. coli* and *H. sapiens* network, respectively (From left to right, species was indicated on the top of the figure). **(A, B, C)** The ROC curve for co-regulators pairs based on the ranking result from CoReg and WT. **(D, E, F)** Rewiring recall score for all the methods. We calculated rewiring recall score under rewiring probability from 0 to 0.5. Each data point is the average score of five runs. Error bar was added to show the standard error. For the human network, EB algorithm was not tested because computation cannot be finished within a reasonable amount of time on large-scale network such as human network.

3.2.5 Different Tree Cut Strategies: Dynamic Tree Cut and Static Tree Cut

A proper strategy to cut the hierarchical tree is necessary because 1) there is no prior knowledge available for the expected number of modules and 2) it is really difficult to decide an optimal cutting height that works for all the branches of the hierarchical tree. The parameters provided by dynamic tree cut algorithm gives more parameters to adjust module size (**see 3.5 Methods**), providing flexibility to tree cutting. Here, we explored the performance of both static tree cut and dynamic tree cut strategies for cutting a hierarchical tree. The performance of each method was shown in **Figure 3.4**. For all three species, dynamic tree cut has outperformed the static tree cut in most of the rewiring probabilities. Thus, in the case of co-regulatory modules finding, dynamic tree cut works better than static tree cut.

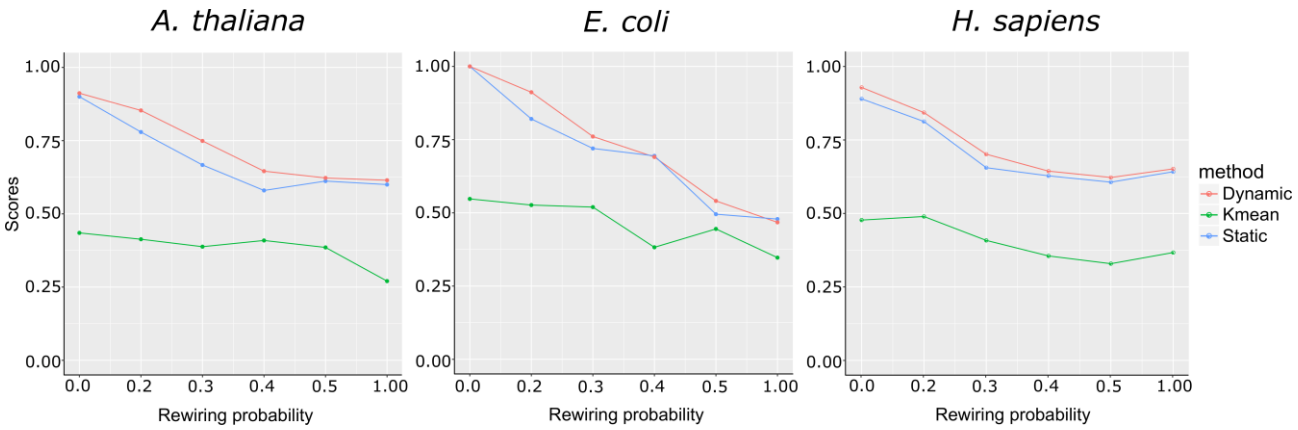


Figure 3.4. **Comparison of dynamic tree cut and static tree cut**. Different tree cut method was applied after hierarchical clustering. We evaluated each tree cut method by RRS.

3.2.6 CoReg identified three types of co-regulatory modules

After computational simulation and comparisons, we decided to use CoReg+jaccard in the following analysis because of the consistent performance of this similarity measurement. CoReg identified 87, 141 and 1208 co-regulatory modules in *A. thaliana*, *E. coli* and *H. sapiens* networks, respectively. We focused on the *A. thaliana* network for further explorative analysis, because we are interested in the roles of co-regulatory genes in plant development and abiotic stress responses, and regulatory connections between those two processes. For the *A. thaliana* network, the largest co-regulatory module contains 13 nodes while the smallest module contains only two nodes. To annotate the transcription factors in this network, we obtained the transcription factor annotation from the Plant Transcription Factor Database (PlantTFDB) [53]. The co-regulatory module assignment and protein family assignment for each transcription factor are provided as **supplementary table 3.1**. For each co-regulatory module, we identified all the genes within the module and their first neighbors in the network. All the interactions between these genes and gene annotations are presented in **supplementary table 3.2**.

Based on the in-degree and out-degree of the genes in the co-regulatory modules, co-regulatory modules can be classified into three types: 1) Regulator modules, which include genes with more than 90% edges are as outgoing edges; 2) Target modules, which include genes with more than 90%

edges are incoming edges; 3) Other modules are classified as intermediate modules. Regulator

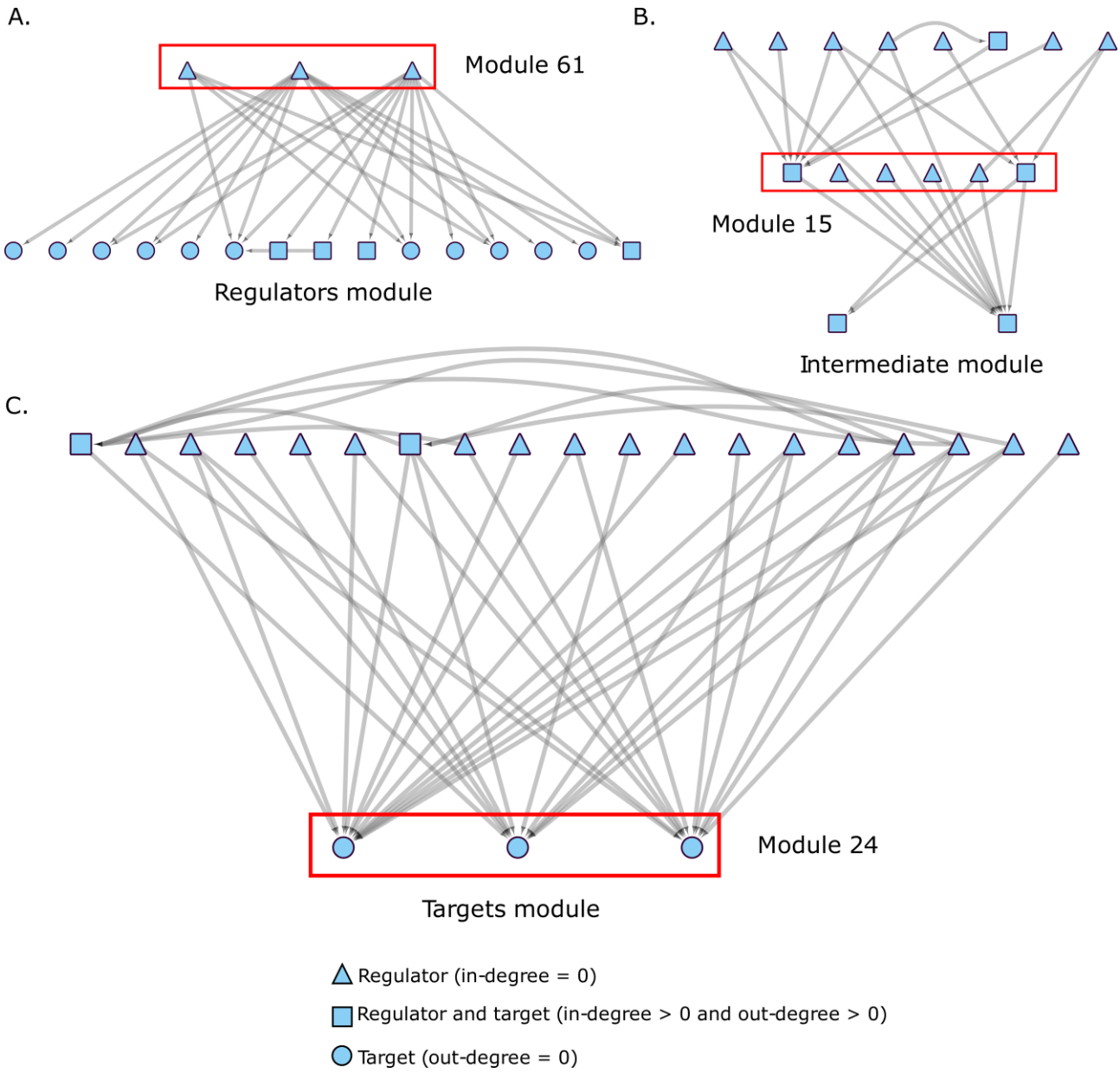


Figure 3.5. **Three types of modules identified by CoReg**. Based on the in-degree and out-degree of the genes in each module, modules were classified into three categories. Red boxes indicate nodes in the same module. In each of the panels, only first neighbors of the nodes in the modules are included. **(A)** Regulator modules. **(B)** Intermediate module. **(C)** Target modules. Please see Additional file 5: **Supplementary table 3.2** for the gene names for each module.

module consists of mostly regulators, which are likely to initialize transcriptional regulation, whereas target module contains mostly target genes of transcriptional regulation. The intermediate module serves as the mediator for the regulation activities. **Figure 3.5** shows examples for each type of module. The regulator module in **Figure 3.5A** contains three regulators from module 61 (AT2G38340, AT5G15210, AT1G24625). The intermediate module in **Figure 3.5B** consists of 6 genes from module 15 (AT5G44080, AT3G49930, AT2G31370, AT1G32150, AT1G09540 and AT2G22850), which connect to 8 regulators and 2 targets. The three regulators connect to 16 target

genes in total. The target module shown in **Figure 3.5C** includes three target genes from module 24 (AT5G17420, AT5G13180 and AT5G44030), which are targeted by 19 TFs in total. **Supplementary table 3.2** shows module ID for each gene. The presence of many common targets and common regulators demonstrates the co-regulation detected by CoReg in complex directed networks.

3.2.7 CoReg identified both known and novel co-regulatory modules

Because the co-regulatory modules are solely based on their network connections, we investigated the expression patterns of genes in the same co-regulatory modules using published microarray expression data from the AtGenExpress database [104,143,144]. The expression of over 22,000 *Arabidopsis* genes was analyzed using microarray hybridization and provided expression patterns in three data sets: a developmental tissue series, hormone treatment, and abiotic stress responses. In the case of the developmental tissue series, various tissue types including leaves, roots, flowers, and stems were sampled at different developmental stages. In the case of the hormone treatment dataset, plant hormones--auxin, cytokinin, gibberellin, brassinosteroid, abscisic acid, jasmonate and ethylene--were used to treat growing seedlings and expression levels were monitored in time course experiments. In the abiotic stress data set, time course experiments were performed under abiotic stress conditions including heat, cold drought salt, high osmolarity UV-B (Ultraviolet-B) light and wounding (see **3.5 Methods**). To measure the correlation between the genes in co-regulatory modules in the *A. thaliana* network, we calculated the Pearson Correlation Coefficient (PCC) between genes and estimated the significance of PCC for each co-regulatory module using these three different data sets. Twenty-one out of 87 modules identified by CoReg show significant co-expression (estimated p-value<0.05) in at least one of the three data sets. More specifically, there are 6, 13, and 4 modules showing significant co-expression in the developmental, stress, and hormone expression data sets, respectively. These results suggest that genes in co-regulatory modules are co-expressed across various conditions and play roles in transcription co-regulation.

Among all modules identified by CoReg, module 70 (**Figure 3.6B**) contains two transcription factors from the nuclear factor YC (NF-YC) gene family: AT1G54830 (NF-YC3) and AT1G56170 (NF-YC2). NF-Y is a transcription factor complex which includes subunit A, B and C. The three subunits form a NF-Y transcription factor complex which binds to promoters containing a CCAAT-box [145,146]. NF-YC2 and NF-YC3 were found to participate in the control of floral induction in *A. thaliana* [146]. Our expression analysis shows that the two NF-YC TFs are significantly highly correlated with each other in the developmental data set (PCC = 0.800, p-value < 0.05) and in the

stress data set (PCC = 0.862, *p*-value < 0.05). The developmental data set covers a broad range of developmental stages from embryogenesis to senescence, which suggests that co-regulation of NF-YC2 and NF-YC3 may also participate in stress response and other developmental processes.

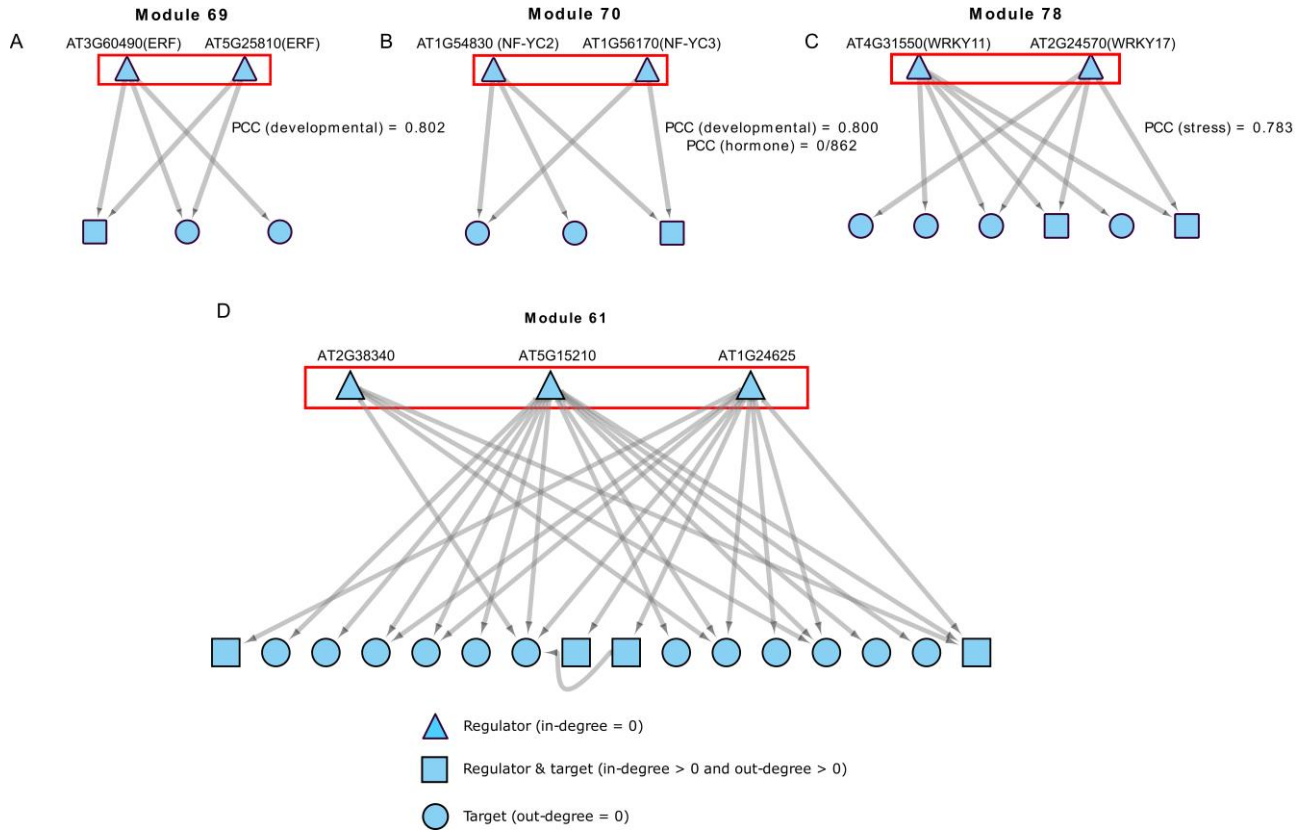


Figure 3.6 Visualization of module 69,70 and 78, along with their first neighbors in the network. **(A)** Module 69 and all of its first neighbors. **(B)** Module 70 and all of its first neighbors. **(C)** Module 78 and all of its first neighbors. **(D)** Module 61 and all of its first neighbors.

CoReg also identified module 78, which contains two WRKY transcription factors (**Figure 3.6C**). The members of the WRKY transcription factor family are involved in diverse biological processes, such as response to biotic/abiotic stresses, seed development and seed germination [147]. The two transcription factors, AT4G31550 (WRKY11) and AT2G24570 (WRKY17) are previously reported to be involved in the regulation of basal defense against *Pseudomonas syringae* pv tomato [148]. It was concluded that both WRKY11 and WRKY17 act as negative regulators in this defense process and WRKY11 and WRKY17 double mutant plants showed stronger defense than WRKY11 single mutant plants [148]. Expression analysis shows that this co-regulatory module has a high expression correlation (average PCC = 0.783, *p*-value < 0.05,) in the stress data set, suggesting both WRKY11 and WRKY17 are active during a biotic stress response process. The results from module 70 and 78 indicate that our method can identify known co-regulatory modules through mining

large-scale gene regulatory networks. We therefore analyzed other CoReg modules to identify significantly co-expressed modules.

Among other modules identified by CoReg, module 69 includes two ethylene response factor (ERF) transcription factors: AT3G60490 and AT5G25810 (**Figure 3.6A**). Genes in the ERF family play important role in various developmental and physiological processes in plants [149], such as leaf petiole development [150], shoot formation [151], resistance to pathogen attack [152] and various abiotic stresses [153] . Co-regulation between AT3G60490 and AT5G25810 was not previously reported. AT3G60490 and AT5G25810 show a significantly high correlation with each other in the developmental data set (PCC = 0.802, *p*-value < 0.05), suggesting that module 69 is a co-regulatory module involved in developmental processes.

Module 61 contains three TFs (**Figure 3.6D**). Two of them, At5g15210, encoding DREB19, which is active both in development and in stress responses [154] and At2g38340, encoding ZFHD3/HB30, a homeodomain protein with a role in the regulation of floral development [155]. DREB19 had twelve targets in the module, while ZFHD3 had only five targets, four of which were co-regulated by DREB19. The four co-regulated targets are each associated with growth and development. BLH3, a TF that regulates the transition from vegetative to reproductive development [156], PGSIP1, a protein involved in secondary wall biosynthesis. AT4G28370, encoding an E3 ligase associated with plant cell wall modification, and AT2G34710, encoding an HD-ZIP TF, PHB, which regulates leaf vascular development through auxin responses.

3.2.8 CoReg reveals roles of co-regulatory modules in *Arabidopsis* abiotic stress responses

To test whether genes in CoReg modules are also co-expressed in genome-scale network, we applied CoReg to a large scale regulatory network generated by DAP-seq with more than 2.8 million interactions and more than 2,300 gene expression profiles (**Figure 3.7**). We applied CoReg on the DAP-seq network to identify co-regulatory modules. Then for each co-regulatory module, we calculated the pairwise PCCs for all the genes in the module and obtained a module average PCC.

The gene expression data (2327 samples) were generated from 62 experiments with each experiment containing multiple replicated samples. These 62 experiments fell into multiple categories including biotic stresses, abiotic stresses, hormone treatments, developmental series and mutant experiments. One experiment can be assigned to multiple categories. Experiments do not fall into the categories listed above are classified as “other types”. The information of experimental

conditions for each experiment data set is shown in **Supplementary table 3.3** and each experiment has a unique GSE id from gene expression omnibus (GEO) database. We selected the 108 modules whose co-expression is significant (p-value<0.05) in at least 1/3 of all 62 experiments and plotted the PCC values for these 108 modules across different treatments (**Figure 3.7**). The co-expression values for all 108 modules in different GSE accession and the corresponding p-value for these co-expression values are provided as **Supplementary table 3.4** and **3.5** respectively. Among these modules, we found two major groups of coReg modules. Group 1 contains 15 target modules, 1 intermediate module and 6 regulator modules. Group 2 contains 30 target modules, 2 intermediate modules and 4 regulator modules. In both groups, genes within the modules are highly co-expressed in abiotic stress conditions, suggesting that modules identified by coReg are likely to be co-expressed under abiotic stresses in *Arabidopsis*.

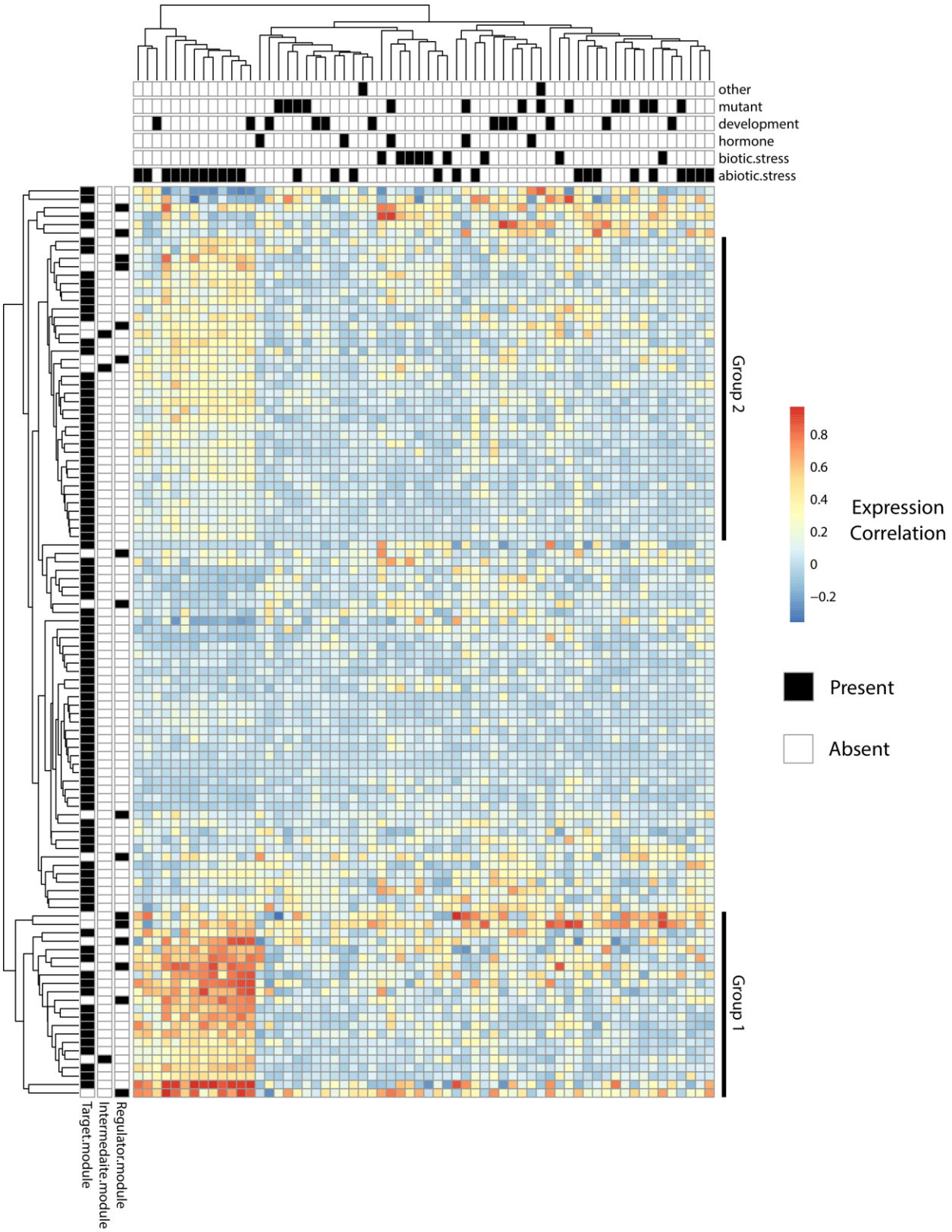


Figure 3.7 Heat map for co-expression levels of CoReg-identified modules in DAP-Seq network. We first ran CoReg on the DAP-seq network to identify the modules. Then for each co-regulatory module, we calculated the pairwise co-expression values for all the genes in the module and averaged the pairwise co-expression to get a single co-expression value for the module. We selected 108 modules which are significantly co-expressed (p-value < 0.05) in at least 20 out of 62 experiments (See **estimate *p*-value in large-scale network** in **Methods** section) to plot the heat map for co-expression. The conditions for each experiment are marked in the top panel with black squares. The module

type is marked on the left panel with black square.

3.3 Discussion

Two recently published online tools, TF2Network [62] and ePlant [157], have integrated DAP-seq data for interaction prediction and visualization. TF2Network is an online tool which allows the user to infer candidate regulators from a list of genes. ePlant provides user-friendly interface for query and visualization of regulators predicted by DAP-seq experiments. We compared the result produced by CoReg to TF2Network and ePlant. We selected all target genes of module 61 to infer their regulators in TF2Network. Totally, TF2Network identified 49 regulators for the given list of targets while CoReg identified 3 co-regulators for the same set of targets. We did not find any overlap between TF2Network identified regulators and CoReg identified co-regulators. For ePlant, we submitted three co-regulators in module 61 to search for their targets. ePlant identified 9620 targets in total while there are only 17 targets for module 61 in our network. 11 out of 17 targets can be found in ePlant identified targets. This result shows that, while eplant gives an overview of all potential targets of the TFs, CoReg selected the subset of targets to give users specific targets that are commonly regulated by the TFs. Although the goal of CoReg is different from TF2Network and ePlant, we think CoReg can be used as a complementary method to search for specific targets of interest.

The three similarity indices calculated the similarity score by measuring the proportion of overlap of first neighbors. The difference is that inverse log-weighted index takes into account the degrees of shared first neighbors while the other two do not. The idea is that two nodes may be more similar if they share some common low-degree neighbors, because high-degree neighbors are more likely to connect to the nodes by pure chance [158]. However, from our simulation result, this strategy lead to decreased performance when co-regulation probability is high (*prob > 0.6*, **Supplementray Figure 3.2)**. This decrease in performance happened when the module size is larger or equals to the number of targets (e.g. module size = 15 or 20 and target = 5). In these cases, regulators in the same module share few targets, whereas targets have higher degrees than the regulators. The inverse log-weighted similarities between targets are thus higher than the similarities between regulators, which causes CoReg to fail to identify co-regulators in the same module. In contrast, Jaccard index and Geometric index are normalized by total number of common neighbors and the product of common neighbors respectively. These methods avoid the problem found in the inverse log-weighted similarity. Our results suggest that CoReg+geo index and CoReg+jaccard index are better choices when the number of regulators is larger than the number of targets. However, such situation is unlikely to happen in transcription regulatory

networks because number of transcription factors are usually much smaller than number of target genes.

Combinatorial regulation by transcription factors (TFs) of target genes underlies the functioning of gene regulatory networks and determines gene expression levels during development and under both biotic and abiotic stresses in many organisms. In *Arabidopsis* and rice, WRKY transcription factors are found to form four and nine co-regulatory clusters respectively [159]. These clusters are involved in diverse signal transduction pathways and in pathogen responses [159]. In a prokaryotic organism such as *E. coli*, transcriptional co-regulation is a key mechanism, for example, in regulating cellular responses to changes in amino acid pools [160]. In human studies, co-regulation was found to be significantly enriched in gene regulatory networks and to be important for maintaining the robustness of gene regulation [161]. Co-regulation is also involved in specific disorders in human, for example, mis-regulation of two co-regulators were shown to be related to the onset of autism [162]. Transcriptional co-regulators are also mis-regulated in breast and ovarian cancer [163]. These studies show that co-regulators occur ubiquitously in all living organisms and are involved in many biological processes, and our computational method represents one important step towards identifying all tissue- and condition- specific co-regulatory modules.

In recent years, high throughput experimental techniques, such as yeast one hybrid [14,121], ChIP-seq [15,93] protein binding microarray (PBM) [131] and DNA affinity purification sequencing (DAP-seq) [16], have significantly increased the amount of known transcriptional regulatory networks for many organisms. We have tested CoReg in three different organisms and on networks generated by four technologies including ChIP-seq, DAP-seq, Yeast-1-hybrid and literature database. Our results showed that CoReg performed better than existing approach in all these species and methods used. These powerful experimental techniques will provide CoReg with abundant data sets to mine co-regulation information in the future.

Besides direct TF-gene interactions analyzed in this study, other genomic data such as chromatin modification data and DNA methylation data are also available for many species. Both chromatin modifications and DNA methylations are known to regulate gene expression. However, the challenge is that both chromatin modification and DNA methylation data are condition- and cell type-specific. Therefore, we did not include chromatin modification or DNA methylation in our analysis. There are multiple methods to incorporate other types of regulatory information in CoReg. For a pair of regulatory nodes in the network, the information of chromatin modification or DNA methylation of the target genes can be directed added in the similarity measurement used by CoReg.

Alternatively, other regulatory information can be analyzed in a post-hoc fashion: for co-regulatory modules identified by CoReg, one can perform enrichment analysis to identify which type of chromatin modification or DNA methylations are enriched in the co-regulatory modules.

Using *Arabidopsis* as a model system, we found that CoReg not only detected known co-regulatory genes such as WRKY transcription factors, but also uncovered unknown co-regulatory genes. By integrating gene expression data with regulatory network information, we identified co-regulatory modules that are highly co-expressed under abiotic stresses, hormone treatments and during plant development. These results suggest that CoReg can be used to mine existing network data and gene expression data to identify key co-regulatory genes in many other organisms.

We identified three types of co-regulatory modules: regulator modules, target modules and intermediate modules. By combining CoReg modules with gene expression data from almost all published studies in *Arabidopsis* [164], we found that target modules tend to be highly co-expressed under abiotic stress conditions. For example, in module 24 (**Figure 3.5C**), there are three target genes co-regulated by 19 regulators, with each of the three genes is regulated by more than 10 regulators. This observation could represent one type of structural stability in gene regulatory networks where the network structure is robust against perturbations because removal of each individual edge or regulatory nodes has small impact on the total number of regulators for each target genes. In contrast, in regulator modules (for example, module 61 in **Figure 3.5A**), each regulator is regulating many genes and mutations in any of the regulators can have strong effect on expression of the target genes. Genes in module 24 and module 61 are highly connected hub genes. However, in directed gene regulatory networks, perturbation of regulators of target modules has small impact on expression regulation. This may explain why target modules are widely used in abiotic stress responses in *Arabidopsis*.

3.4 Conclusions

In this study, we developed a computational tool, CoReg, for identifying co-regulators in gene regulatory network. We performed the simulation-based analysis to evaluate CoReg and other module-finding algorithms. The result shows that CoReg outperforms other algorithms in identifying co-regulators. We applied CoReg to a genome-scale regulatory network for *Arabidopsis*. The subsequent co-expression analysis using a large expression data set indicates that many highly co-expressed modules in this network are associated with abiotic stress, suggesting that target modules are more robust against random perturbation of regulatory networks.

3.5 Methods

3.5.1 Regulatory network data sets

A regulatory network involved in secondary cell wall synthesis [14] and a SHORTROOT-SCARECROW regulatory network [121] for *A. thaliana* was downloaded from online supplementary materials. These two regulatory networks were mered and duplicated interactions were removed. To test CoReg in a larger network of *Arabidopsis*, we downloaded a recent large-scale regulatory network generated by DAP-seq for *A. thaliana* [16]. The *E. coli* regulatory network data was downloaded from http://www.mrc-lmb.cam.ac.uk/genomes/madanm/ec_tf/ [130], which integrated TF-DNA interactions for *E. coli* from multiple publications. A *H. sapiens* network generated by ChIP-seq was downloaded from http://encodenets.gersteinlab.org/ [93]. For *H. sapiens*, we used only the interactions between the TFs and proximal promoters. Self-loops and duplicated edges were removed using the igraph R package [165]. However, self-loops and duplicated edges could be integrated into our computational tool. In summary, there are 412 genes and 1490 edges in the *A. thaliana* yeast-1-hybrid network, 32606 genes and 2,848,929 edges in the *A. thaliana* DAP-seq generated network, 889 genes and 1405 edges in the *E. coli* network, 9057 genes and 26043 edges in the *H. sapiens* network.

3.5.2 CoReg method

Definition of directed networks. A regulatory network can be represented by a directed graph $G = (V, E)$, where V (vertices) is the set of genes in the network and E (edges) is the set of TF-gene interactions. In a directed graph, each edge is represented by a pair of ordered nodes including a head node and a tail node. The head node is a TF (a regulator), whereas the tail node (a target) is the gene regulated by the head node and can be either a TF gene or a non-TF gene (See **supplementary figure 3.4** for an schematic example of directed network). Every edge $e = (i, j), (e \in E)$ in G links a head node v_i to a tail node v_j $(v_i, v_j \in V)$. v_i is the in-neighbor of v_j and v_j is the out-neighbor of v_i.

Define Problem. In a directed network with $|V|$ nodes, a module m is a group of n nodes which is represented by $m = \{v_1, v_2, \ldots v_n | v \in V, n \leq |V|\}$. When there are M modules in the graph, a partition P of the graph is a set of modules that divides all the nodes V into M modules. The goal of CoReg, is to find a partition P such that in each module m, for any two genes v_i and v_j $(v_i, v_j \in m)$, the similarity $S(v_i, v_j)$ is greater than $S(v_i, v_k)$ and $S(v_j, v_k)$, $(v_k \notin m)$. Here, we tested several similarity scores and clustering approaches to find this partition. We also validated our methods by analyzing gene co-expression data.

CoReg takes a regulatory network (a directed graph) as input and generates a module assignment for each gene. CoReg first calculates a pairwise similarity score for all the nodes in G to generate a similarity score matrix *S*. Then *S* is transformed into a dissimilarity matrix *S'*. CoReg applies hierarchical clustering followed by the dynamic tree cut algorithm [166] to identify the modules. **Figure 3.1B** shows the brief workflow for CoReg. The detailed description for each step is given below. We evaluated CoReg using simulation, ROC curve and rewiring recall score (See **3.5.3 Performance assessment** in **3.5 Methods** section).

Similarity indices. We explored different similarity indices for calculating the pairwise similarity score. Three similarity indices were compared in this study: the Jaccard similarity index [167], the geometric similarity index [167], and the inverse log-weighted similarity index [158]. Given node v_i, we define the set of in-neighbors and out-neighbors of v_i as $\boldsymbol{N}^{(in)}(v_i)$ and $\boldsymbol{N}^{(out)}(v_i)$. For every node pair v_i and v_j, we computed in-similarity and out-similarity separately. For each pair of nodes, the **Jaccard index** is calculated by dividing the number of common neighbors by the total number of neighbors for both nodes:

$$J_{v_i,v_j}^{(in)} = \frac{|N^{(in)}(v_i) \cap N^{(in)}(v_j)|}{|N^{(in)}(v_i) \cup N^{(in)}(v_j)|}$$

$$J_{v_i,v_j}^{(out)} = \frac{|N^{(out)}(v_i) \cap N^{(out)}(v_j)|}{|N^{(out)}(v_i) \cup N^{(out)}(v_j)|}$$

The **geometric index** is the square of the number of common neighbors of v_i and v_j divided by the product of the number of neighbors of v_i and v_j:

$$G_{v_i,v_j}^{(in)} = \frac{|N^{(in)}(v_i) \cap N^{(in)}(v_j)|^2}{|N^{(in)}(v_i)| \cdot |N^{(in)}(v_j)|}$$

$$G_{v_i,v_j}^{(out)} = \frac{|N^{(out)}(v_i) \cap N^{(out)}(v_j)|^2}{|N^{(out)}(v_i)| \cdot |N^{(out)}(v_j)|}$$

For Jaccard similarity index, if both v_i and v_j have no common in-neighbors/out-neighbors, the corresponding in-similarity/out-similarity score is set to zero. For geometric similarity index, if either v_i or v_j has no in-neighbors/out-neighbors, the corresponding in-similarity/out-similarity score is set to zero.

The **inverse log weighted similarity index** is the inverse log weighted sum of the degree of all the common neighbors for v_i and v_j. The idea is that two nodes may be more similar if they share some common low-degree neighbors, because high-degree neighbors are more likely to connect to the nodes by pure chance [158]. The degree of the node c is represented by d(c).

$$I_{v_i,v_j}^{(\text{in})} = \sum_{c \in \{N^{(in)}(v_i) \cap N^{(in)}(v_j)\}} \frac{1}{\log(d(c))}$$

$$I_{v_i,v_j}^{(\text{out})} = \sum_{c \in \{N^{(out)}(v_i) \cap N^{(out)}(v_j)\}} \frac{1}{\log(d(c))}$$

For weighted networks, the degree of each node can be replaced by the sum of edge weights.

Similarity and dissimilarity matrix. CoReg calculates the similarity score between every pair of v_i and v_j to get in- and out-similarity matrices, respectively. Similarity matrix $\boldsymbol{S}$ is the sum of two matrices. A dissimilarity matrix $\boldsymbol{S'}$ was computed based on $\boldsymbol{S}$:

$$\boldsymbol{S}_{ij} = \boldsymbol{w_1} \boldsymbol{S}_{ij}^{(\text{in})} + \boldsymbol{w_2} \boldsymbol{S}_{ij}^{(\text{out})}$$

$$\boldsymbol{S}'_{ij} = \frac{max(\boldsymbol{S}) - \boldsymbol{S}_{ij}}{max(\boldsymbol{S})}$$

where $\boldsymbol{S}_{ij}^{(\text{in})}$ is the incoming similarity between gene v_i *and* v_j and $\boldsymbol{S}_{ij}^{(\text{out})}$ the outgoing similarity between v_i and v_j. The maximum value in matrix *S* is denoted by max(**S**). $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ represent weights assigned to the incoming and outgoing similarity matrices. In this study, we set $\boldsymbol{w_1}$=$\boldsymbol{w_2}$=1.

Hierarchical clustering and dynamic tree cut. Hierarchical clustering and dynamic tree cut is implemented using R built-in function 'hclust' and a R package DynamicTreeCut [166]. Hierarchical clustering was performed on the dissimilarity matrix with complete linkage as agglomeration method. R package DynamicTreeCut [166] was then used to cut the tree from hierarchical clustering with a "hybrid" cutting method. The advantage of the dynamic tree cut algorithm over the fixed height tree cut is that the dynamic tree cut method takes into account the shape of the hierarchical tree and cuts the tree adaptively. In Brief, the core of a cluster consists of nodes that have low dissimilarity with each other in the cluster. The core scatter is the average of pairwise dissimilarity in the core and the cluster gap is the difference between core scatter and the joining height where the cluster joins the dendrogram. Dynamic tree cut algorithm merges the clusters in the dendrogram from bottom to the top. When the criterion of score scatter and cluster gap is met, the cluster will stop merging with other clusters. This process will continue until all the clusters stop merging. We chose dynamic tree cut to process the dendrogram because when no prior knowledge except for the

network itself is provided, it is difficult to find a single unique cutting height that works best. Here, we set the minimum size of a cluster to 2, since the co-regulation activity requires at least two genes to collaborate. The R package DynamicTreeCut provides a parameter 'deepSplit' to conveniently set up the threshold for cluster shape. deepSplit takes only four values, 0, 1, 2, 3. The higher the value is, the smaller the clusters tend to be. For the analysis in this paper, we used the default value for deepSplit (deepSplit = 1). However, other values for deepSplit can be set in our CoReg package. Please refer to the supplementary materials of [166] for more details about the algorithm.

3.5.3 Performance assessment

Overview of Performance Assessment. In the following subsections, we described the bipartite transformation, generating simulated co-regulatory network, rewiring simulation, estimation of AUC and ROC, and comparison between different tree cutting methods. Bipartite transformation was performed to transform directed network to undirected network so that LP, EB and WT can be performed on this transformed network. Simulated co-regulatory networks were used to assess the ability of different methods to identify pre-specified co-regulatory modules. Rewiring simulation and estimation of AUC and ROC were designed to assess the performance of different module finding methods. Performance of different tree cutting methods was compared to find best strategy for co-regulatory module finding. The network is first randomly rewired and different module finding methods were applied to the rewired network to generate modules. A Performance score is computed based on the module finding result. AUC and ROC were estimated using the similarity score calculated from the rewired network. **Figure 3.1D** illustrates the flow chart of the analyses.

Bipartite Transformation for LP, EB and WT. To compare the performance of CoReg to other module finding algorithms, we applied CoReg and LP, EB, WT algorithms on the same data. However, LP, EB and WT were initially designed for undirected network and cannot be directly applied to directed networks. This could be solved by transforming the directed network into a bipartite network [18]. Such undirected network preserves the direction information [18]. In this paper, we applied a transformation process as described previously [18]. A bipartite network is defined as $G_B = (V_h, V_t, E_b)$ and G_B is transformed from a directed network $G = (V, E)$ according to:

$$V_h = \{v_h | v \in V, k_v^{out} > 0\}$$

$$V_t = \{v_t | v \in V, k_v^{in} > 0\}$$

Where V_h is the set of nodes transformed from source nodes in G and V_t is the set of nodes transformed from target nodes in G and E_b is the set of edges in G_B. k_v^{in} and k_v^{out} are the in-degree and out degree for node v_h and node v_t, respectively. For each directed edge $u \to v$ in G, we

transformed u into a head node u_h and transformed v into tail node v_t. Then u_h and v_t will be added into V_h and V_t, respectively. Then the undirected edge between u_h and v_t is created. **Supplementary Figure 3.4** shows an example of transforming the directed network into a bipartite network. Next, we applied LP, EB and WT algorithms on the bipartite networks, assigning the module to V_h and V_t. If the head node and the tail node were transformed from the same node but were given different modules, we assign to this node all the modules that were assigned to both the head node and the tail node. We implemented the bipartite transformation and LP, EB and WT using igraph R package [165].

Generating simulated co-regulatory network. To assess the performance of different module-finding methods, we generate ground-truth modules by constructing simulated network from the scratch. The process is similar to a published method [141]. The original method was proposed to generate modules for bipartite network. Here, we modified this approach to generate co-regulatory modules in directed network. The simulation requires five parameters:

1. *mSize*: size of each module
2. *mNum*: total number of modules
3. *targetNum*: number of targets for each regulator node
4. *auxNum*: number of auxiliary nodes
5. *prob*: co-regulation probability.

We constructed simulated networks by following steps:

1. Generate *mSize*× *mNum* regulator nodes. Each node is assigned to one module and each module will have equal number of nodes (specified by *mSize*).
2. Generate auxiliary nodes as specified by "*auxNum*". These nodes are targets of regulators and will not have any outgoing edges.
3. Select a pool of target candidates for each module. The size of pool is equal to *targetNum*. Each pool is considered as a set of potential co-regulated targets for the corresponding module.
4. For each regulator node, with probability equals "*prob*", randomly select a target from the

pool of target genes. Otherwise, randomly select a target not in the pool. With higher *prob*, regulator nodes in same module will tend to select nodes from the pool of target genes, showing stronger co-regulating modular structure (**Supplementary Figure 3.1**). The numbers of targets for regulators are the same (specified by *targetNum*). This step will be repeated until all regulator nodes have been assigned given number of targets.

The simulated network is generated as an edge list. The pseudo code implementation is provided in Additional file 6. In our simulation experiment, we explored the different settings of parameters. We set *mSize* = (2, 5, 15, 20), *mNum* = 10, *targetNum* = (5, 15, 20), *auxNum* = 200, *prob* = (0, 0.2, 0.4, 0.6, 0.8, 1). We set *mSize* = (2, 5, 15, 20), because the number of known transcription co-regulators are usually small. We set *targetNum* = (5, 15, 20), *mNum* = 10, and *auxNum* = 200, such that the total number of genes in the network is similar to the number observed in the *Arabidopsis* Y1H network. We set *targetNum, mNum* and *auxNum* also because we want to reduce the total time of computation. Higher *targetNum* or *auxNum* are likely to lead to better performance, because the calculation of similarity will be more stable with more nodes. Higher *mNum* is unlikely to alter the performance. We used NMI score to quantify the correlation between pre-specified modules and algorithm identified modules [142].

Network duplication and rewiring. As no published genome scale studies of true co-regulators are available, we constructed the true co-regulators by duplicating nodes in the network. We also introduce noise to the network by rewiring simulation. For a given network $\boldsymbol{G}$, we randomly selected **n** nodes from the network and duplicated these nodes. If we duplicate small number of nodes, there will not be sufficient nodes for performance analysis. If we duplicate too many nodes, the duplicated network will drastically alter the topology of the original network. Because of these restrictions, we chose to set n=80 nodes in each of the networks. We define the set of randomly selected original nodes as $\boldsymbol{U}$ and the set of their duplicated nodes as $\boldsymbol{U}'$. The corresponding nodes in $\boldsymbol{U}$ and $\boldsymbol{U}'$ were denoted as u and u'. For each duplicated node u_i', the neighbors of u_i were set as neighbors for u_i'. For these duplicated nodes, each node and its original node are a pair of "true" co-regulators (i.e. u_i->u_i'). Negative co-regulators were defined by randomly selecting one from the duplicated nodes and another from duplicated nodes of other nodes.

We rewired the edges connecting u_i' to its neighbors with a selected probability. One edge starting from u_i' will be randomly selected with given probability, then another edge starting from another duplicated node in $\boldsymbol{U}'$ will be randomly selected as well. The target nodes of the two edges

will be swapped. **Supplementary Figure 3.5** shows the steps of rewiring in detail. The rewiring operations do not change the total number of interactions and node degree distributions in the network. In our rewiring simulation, we also preserved the topology of the original networks by rewiring only duplicated nodes. We did not perform rewiring simulation in the whole network because if we rewire the whole network with a given probability, the chance of rewiring the newly duplicated nodes is very low.

Calculate rewiring recall score. To compare the module finding result of CoReg to WT, LP and EB, we calculated rewiring recall scores according to following equations:

$$score = \frac{\sum^{u_i \in U} s_{u_i} w_{u_i}}{\frac{|V||U|}{2}} \quad (7)$$

$$w_{u_i} = \frac{|V|}{m_{u_i}} \quad (8)$$

where s_{u_i} = 1 if u_i and u_i' are in the same module otherwise s_{u_i} = 0. $|U|$ is total number of nodes in U. $|V|$ is total number of nodes in the network. m_{u_i} is number of nodes of the module of u_i . Here, w_{u_i} is a weight to reduce the effect of false positive brought by module size, since the larger the module size is, the more probable that the module will include both u_i and u_i' by pure chance. The denominator of equation (7) is the theoretical maximum value of the numerator. The rewiring recall score is a number between zero and one, and the score will be equal to one when every pair of u_i and u_i' is in the same module and the module contains only u_i and u_i' .

Similarity between co-regulators for Walk Trap method. For Walk Trap, we calculated the similarity by two steps. First, we converted regulatory network to a directed adjacency matrix $\boldsymbol{A}$ which preserved directions information. If node v_i points to node v_j, we mark the entry $\boldsymbol{A}_{ij}$ as '1', otherwise '0'. All the diagonal entries were set to '1' to add self-loops to avoid being divided by 0 in walk trap algorithm. Second, we calculated a transition matrix T according to the equation below:

$$\boldsymbol{T}_{ij} = \frac{\boldsymbol{A}_{ij}}{\sum_{k=1}^{|V|} \boldsymbol{A}_{ik}}$$

Where the denominator is the out degree of node v_i. We calculated a probability matrix similar to what was described in [137]. Here, probability matrix $\boldsymbol{P} = \boldsymbol{T}^m$ and we set m=4 to reduce computational cost. Similar to [137], we calculated distance between the co-regulators v_i and v_j using the following equation:

$$D(v_i, v_j) = \sqrt{\sum_{k=1}^{|V|} \frac{\left(\boldsymbol{P}_{ik} - \boldsymbol{P}_{jk}\right)^2}{d(v_i)}}$$

where $d(v_i)$ denotes the degree of node i and n is the total number of nodes. Here, $D(v_i, v_j)$ defines the pairwise distance between node v_i and v_j. Similarity for Walk Trap is therefore defined as $\boldsymbol{S}_{ij} = 1 - D(v_i, v_j)$. We ranked the co-regulators pairs using the above-mentioned similarity measurement then computed ROC curves and AUC values.

Comparison Between Dynamic Tree Cut and Static Tree Cut. We first performed the duplication and rewiring process as described in **3.2.4 Rewiring recall score**. Then we calculated a distance matrix and applied hierarchical clustering. In the next step, we applied both a static tree cut method and a dynamic tree cut method. For the static tree cut, we sampled the cutting height from 0 to 1 (1 is the maximum tree height since distance ranges from 0 to 1) and increased the cutting height by 0.1 each time (11 sampled points in total). We evaluated each cutting height using RRS (rewiring recall score) and then picked the highest RRS for each rewiring probability. For the dynamic tree cut, two parameters were used to find the optimal RRS: maximum cutting height and deepSplit. The sampling process for maximum cutting height is the same as the sampling process for cutting height in static tree cut. deepSplit only has five possible values (0,1,2,3,4). Therefore, a grid of 11×4 combinations was searched to find the optimal RRS. K-means clustering was added to compare to these two strategies. The parameter k ranges from 1 to number of nodes in the network and increased by (number of nodes)/10 each time. This produced 11 RRS for each rewiring probability. The highest RRS was then picked as the optimal score

3.5.4 Co-expression analysis

Expression data sets. We downloaded expression data sets for *A. thaliana* under stress [104], hormone [143] and developmental condition [144]. Stress expression data set was generated using *A. thaliana* exposed to various abiotic stresses including heat, cold drought salt, high osmolarity UV-B light and wounding [104]. Hormone expression data set was produced from *A. thaliana* samples treated with auxin, cytokinin, gibberellin, brassinosteroid, abscisic acid, jasmonate and ethylene [143]. Gene expression in developmental data set was detected from *A. thaliana* in a series of developmental stages [144]. To compute the co-expression level of CoReg-identified modules in genome-scale network, we downloaded over 2,300 gene expression samples from a recent

publication which collected over 6000 expression samples in total for *A. thaliana* [164]. We selected the experiments which contains more than 10 conditions for co-expression analysis.

Co-expression and *p*-value calculation. We used expression data to estimate the correlations between co-regulators. For each pair of the co-regulators, we calculated the Pearson Correlation Coefficient (PCC) of expression. To estimate the module-level significance of correlation, for each co-regulator module, we calculated the average PCC over all the co-regulators pairs and randomly selected the same number of genes from the whole genome and calculated average PCC. This step was repeated for 1000 times. Thus, *p*-value of PCC is defined as how many times the random PCC is higher than the actual PCC of a given module.

Estimate *p*-value in large-scale network. Since the number of expression data sets and number of modules are large for DAP-seq network, it is computationally expensive to compute *p*-value by random permutation as described previously. Here, we computed the *p*-value using Fisher's combined probability test. Briefly, we first randomly selected two genes from the genome and calculated their co-expression value. This was repeated for 10000 times to generate an empirical null distribution of pairwise co-expression. Then for each module, we calculated the pairwise *p*-value for all the genes in the module. A module level statistic combing these *p*-values is calculated using the following equation:

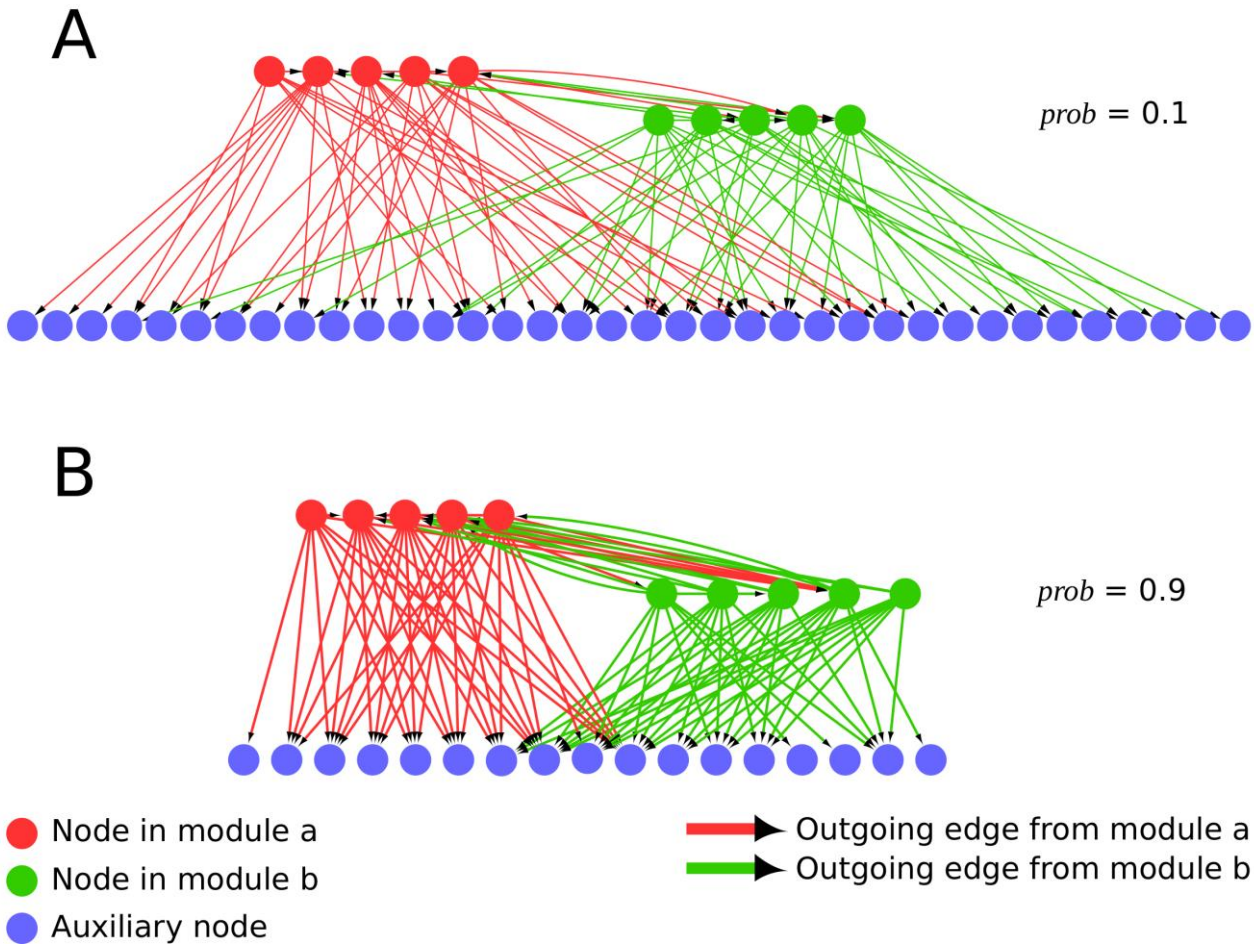
$$X_{2k}^2 = -2\sum_{i=1}^{k} \ln(p_i)$$

Where k is the total number of *p*-values and p_i is the ith *p*-value for the module. The statistic X_{2k}^2 follows χ^2 distribution with 2k degrees of freedom. The module level *p*-value was then computed using the statistic X_{2k}^2 and χ^2 distribution.

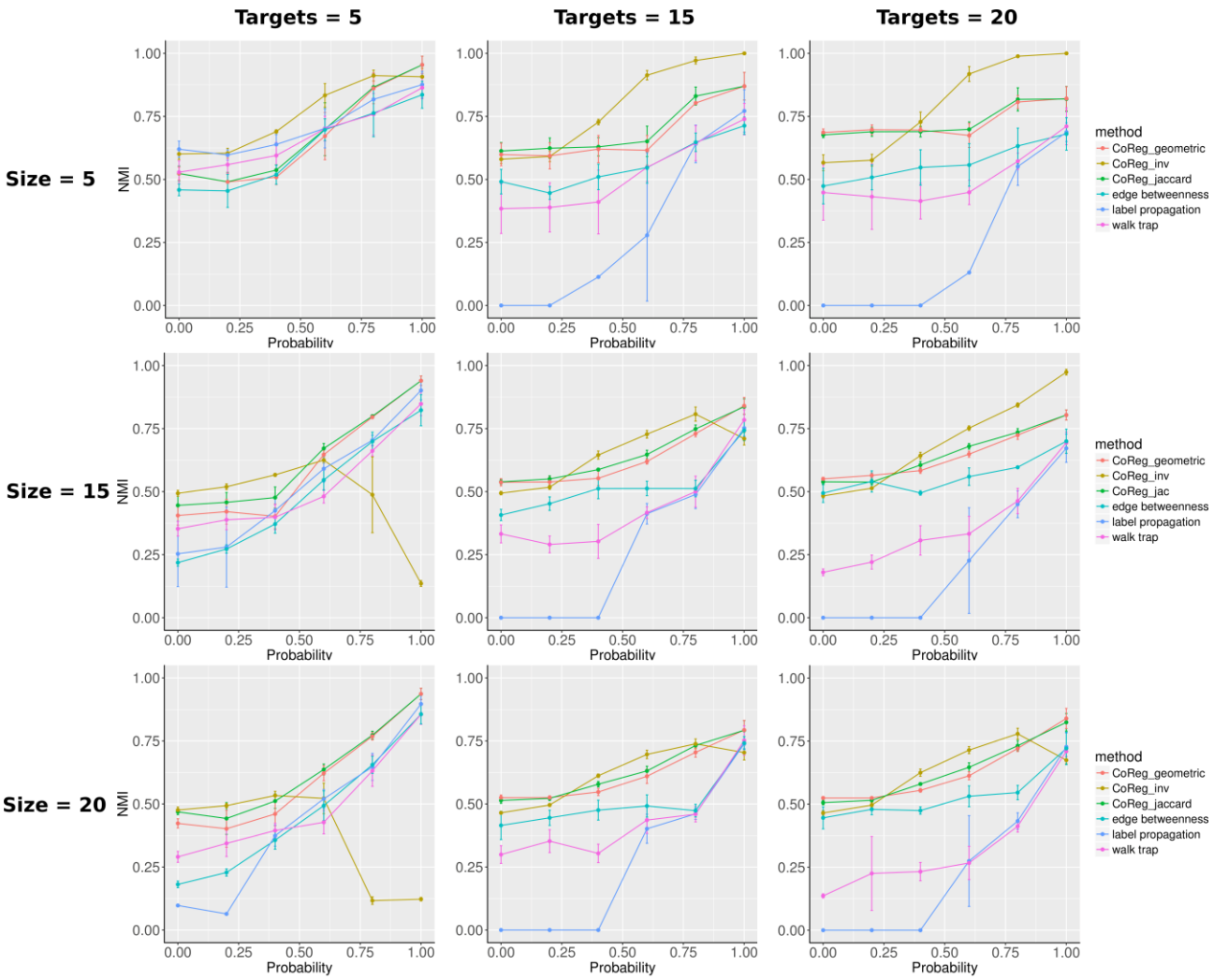
3.6 Authors' contribution

SL conceived the idea. SL and QS designed the experiments. QS developed the R package and performed all the analysis. SL, QS, RG and LH wrote the manuscript.

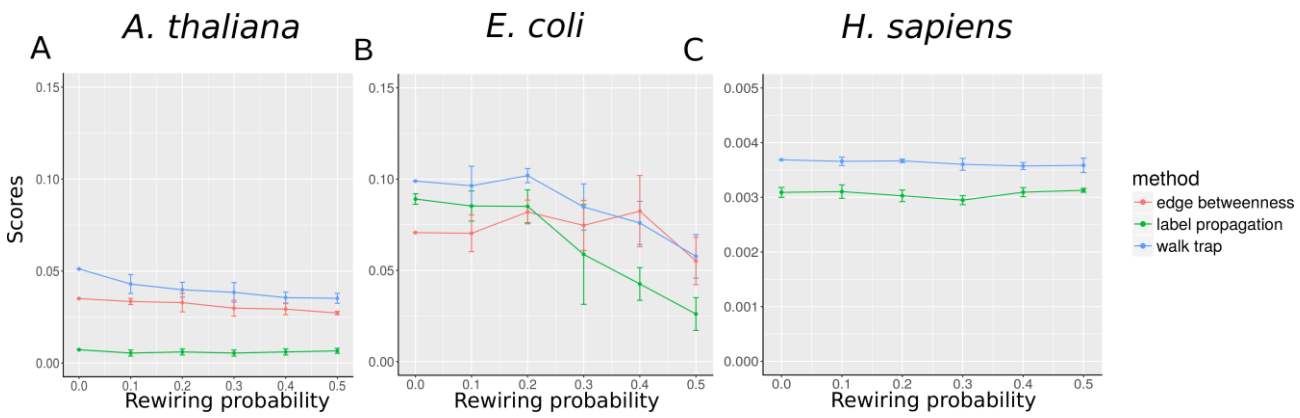
3.7 Supplementary figures



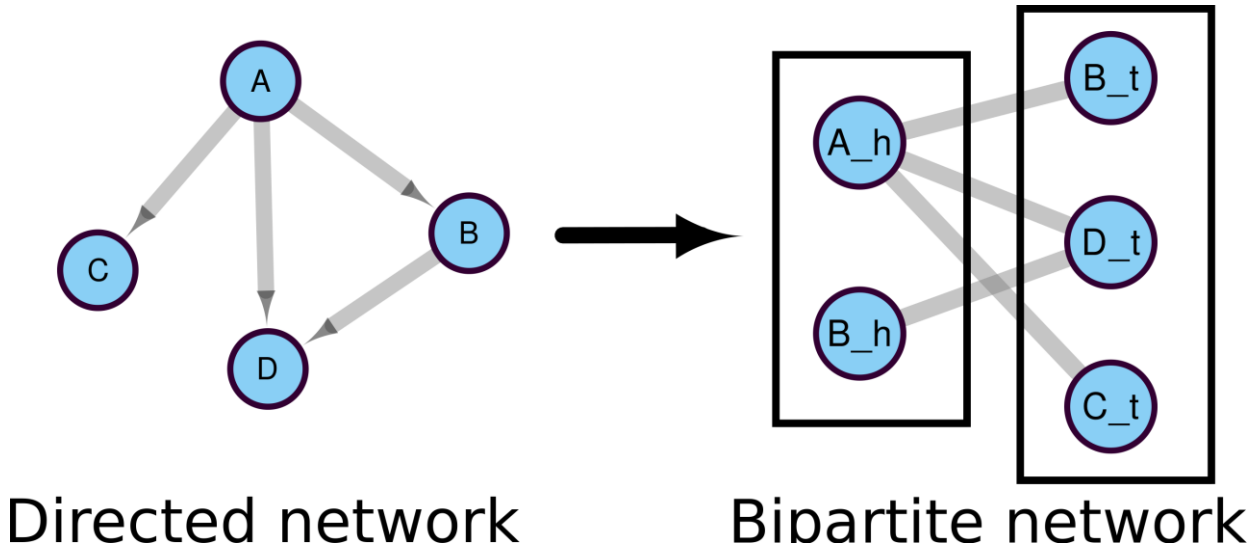
Supplementary Figure 3.1 Co-regulation pattern in networks with different co-regulation probability (specified by the parameter *prob*). Network with higher *prob* is expected to have stronger co-regulation pattern. We generated two modules a and b in this example network. Modules are marked by different color. Zero-degree auxiliary nodes were not shown in the figure. A) network generated with *prob* = 0.1 B) network generated with *prob* = 0.9



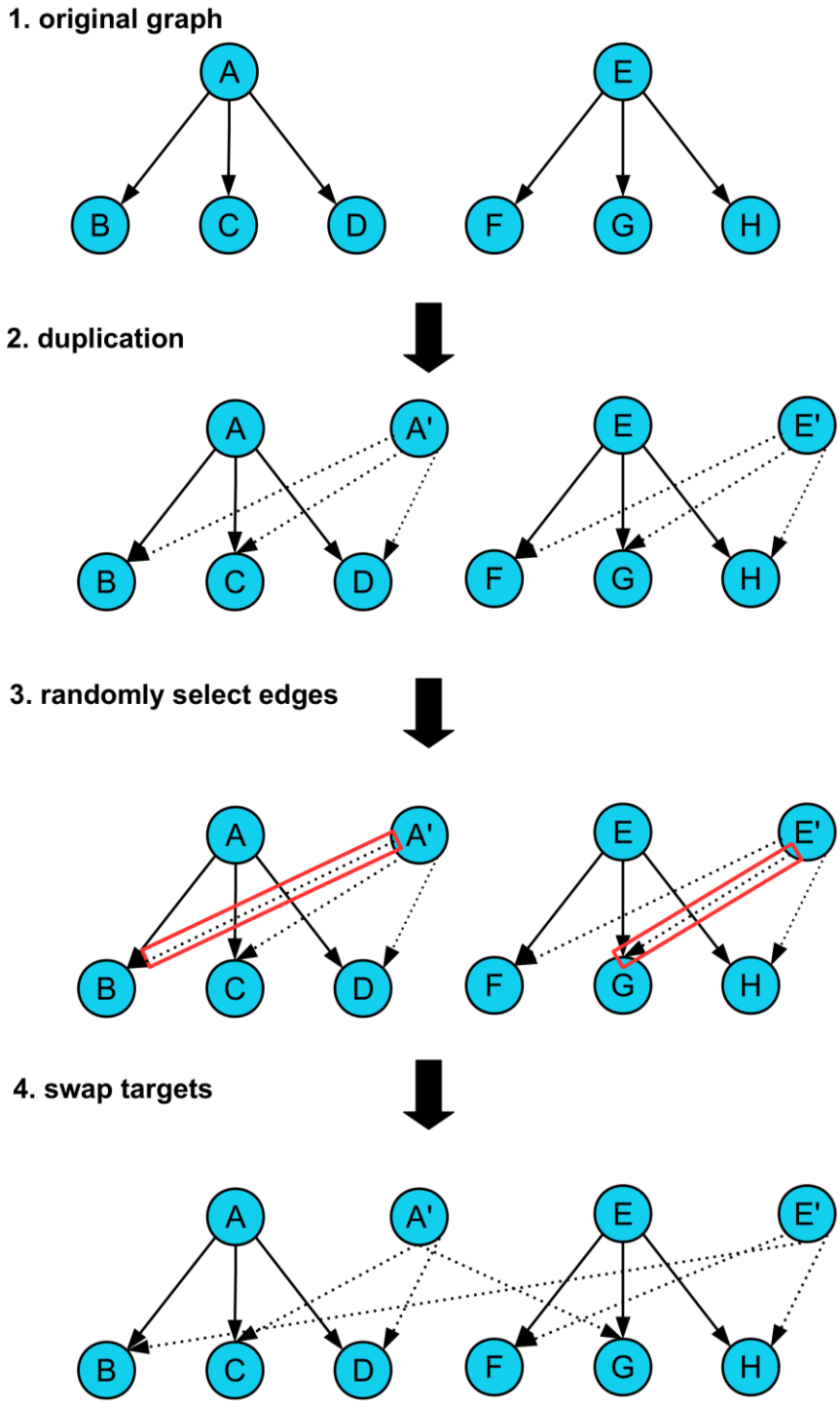
Supplementary figure 3.2 Evaluation of different module-finding methods using simulated networks with different parameters. From top row to bottom row: mSize = 5, mSize = 15, mSize = 20. From left most column to right most column: targetNum = 5, targetNum = 15, targetNum = 20.



Supplementary Figure 3.3 Rewiring recall score for LP, WT and EB in real networks. We rescaled the y-axis to examine the curves for LP, WT and EB.



Supplementary Figure 3.4 The example of bipartite transformation. Network on the left is a directed network, which could be transformed into a bipartite network on the right. The suffix '_h' represents the head node and '_t' means the tail node.



Supplementary Figure 3.5 Network duplication and rewiring. We randomly selected a subset of nodes from the whole network then duplicated them. New nodes that were duplicated from the original nodes are referred as 'pseudo node'. In the figure, A' and E' are the pseudo nodes of A and E, respectively. This means before rewiring occurs, A' and E' duplicated all the edges from A and E (These duplicated edges are the dashed edges in the figure). For rewiring, CoReg first goes through every edge connecting to A' and attempts to rewire the edge with given probability. Once CoReg decides to rewire that edge, another edge in the network will be randomly selected. Then the target nodes for these two edges will be exchanged. In the case shown above, the two edges marked by red box have their target nodes swapped. Therefore, rewiring only applies on pseudo nodes and the original graph remains unchanged during the process.

4. Chapter 4. Summary

In this dissertation, computational tools were developed to investigate 1) regulatory events of plants in response to environmental stresses 2) co-regulatory gene modules for abiotic stresses and developmental response and 3) cell-type-specific classification for plant scRNA-seq data. Various evaluations were performed to validate results generated from computational methods either developed in this dissertation or developed by previous studies. Performances were compared using common metrics such AUC-ROC, AUC-PRC, accuracy (ACC) and mean average precision (MAP). To further validate prediction results, in Chapter 2, experimental results from TARGET assay were utilized to test recovery rate of N response TFs. Admittedly, there is no gold standard dataset to evaluate co-regulation between genes. In Chapter 3, the rewiring analysis of real networks and construction of simulated networks were therefore designed to address this issue, despite that such evaluations do not present any direct evidences from experiments. This is the current limitation of the research conducted in Chapter 3, which we hope can be properly addressed as more plant genomic data is published in the future.

The developed computational tools have addressed several existing issues in the field of plant genomics. First, the DAP-seq assay currently provides most binding sites for Arabidopsis. However, DAP-seq is an *in vitro* assay, which does not provide condition-specific regulatory interactions. By introducing information of chromatin accessibility and gene expressions to DAP-seq binding sites, ConSReg was able to address this issue. Second, as a follow-up step, the generated regulatory networks needs to be better interpreted using computational tools. CoReg was developed in this dissertation, which detects co-regulating genes from large-scale regulatory networks. We have shown that by taking ConSReg-constructed regulatory network as input, CoReg can successfully identify coordination between MYB44 and MYB77, two previously reported TFs involved in modulating hair cell development and integrating hormone response pathways in Arabidopsis root [35]. Third, assigning cell types for Arabidopsis root scRNA data is laborious and time-consuming. Based on a previously published study that explored the use of Siamese neural network for mouse scRNA-seq data [29], a neural network based classifier was described in Appendix A, which showed a way to automate assignment of cell types for Arabidopsis root scRNA-seq data.

In the long run, the ultimate goal for bioinformatics is to generate new hypothesis that can be further validated experimentally and eventually contribute to advancing existing knowledge of biology. Specifically, for regulatory genomics, the first step towards this goal is to develop computational tools to generate regulatory networks and tools to better interpret basic properties. In the subsequent step, as future extension to the framework outlined in this dissertation, functional

information such as pathway or ontology annotations can be integrated into the prediction pipeline. This integration of multi-omics data holds the promise of generating interpretable results that present sufficient functional information for the predicted regulatory associations, which depicts a regulatory map with more insights into biology.

Appendix A: 6. Identification of cell types for plant single cell RNA-seq data

Abstract

Recent applications of single cell RNA-seq (scRNA-seq) techniques have provided several large-scale expression data sets that profile the transcriptome of Arabidopsis root. An important step for plant scRNA-seq analysis pipeline is to assign cell types to the different cell populations. However, current approaches either rely on generating an index of cell identity (ICI) scores using marker genes or inspecting expression patterns of marker genes. In this chapter, we test a neural-network-based pipeline to automatically identify cell types in plant root scRNA-seq data. This pipeline was compared to other machine learning classification approaches including k-nearest neighbors, random forest and support vector machine. The results showed random forest has achieved the best performance. However, a more complete investigation of optimal hyperparameters will be conducted in the future, which may lead to a different conclusion. Detailed results for each cell type suggested that Endodermis, Phloem and Phloem CC are distinctively different from other cell types in respect to transcriptomic profiles.

Keywords

Single cell RNA-seq; Arabidopsis root cell types; Deep neural network

6.1 Introduction

Single cell RNA-seq (scRNA-seq) has recently emerged as a powerful approach to investigate gene expressions at single-cell resolution. Compared to bulk RNA-seq, scRNA-seq can identify rare cell populations and reveal transitions of cell states at different developmental stages, which are difficult to capture using bulk RNA-seq [112,168,169]. scRNA-seq assay has been applied in a number of studies to profile transcriptomes of the model plant, Arabidopsis [170–175]. These applications examined transcriptional dynamics of Arabidopsis root cells in different aspects, including expression pattern of a rare cell type in root quiescent center (QC) [170,172], developmental trajectory of root cells [170,172,174,175], and differential expression of stress responsive genes at single cell level [170,171,174].

An important step in the scRNA-seq analysis pipeline is the identification of cell types. Currently, identification of Arabidopsis root cell types falls into three major categories: (1) Calculate index of cell identity (ICI) [118] using marker genes and assign cell type with highest ICI score [171,173]; (2) Compute correlation coefficient with published root RNA-seq data sets [171,174]; (3) Assign cell types by visualizing expression patterns using known marker genes [170,174,175]. The three methods are often combined to obtain a more precise mapping of known cell types. However, this procedure involves identification of marker genes, manual inspection of expression patterns, and many steps of data processing, which are laborious and time-consuming. Besides, other limitations include 1) complex heterogeneity of cell populations is often characterized by multiple sub-population of cells within one cluster, which can compromise the performance of these approaches; and 2) Known marker genes can be used to identify distinctive cell types but it is difficult to capture rare or novel cell types using these known markers. All these limitations represent computational challenges for automated identification of cell types for scRNA-seq data.

In recent years, a growing number of published studies have utilized machine learning approaches to automate the discovery of cell types. Depending on the training data, these methods can be generally considered as unsupervised and supervised approaches. While unsupervised methods cluster cells without prior knowledge of the cell types, supervised methods typically train cell type classifier with known cell type labels. Unsupervised approaches usually perform dimension reduction for original high-dimensional input data, followed by clustering on the reduced dimensions. For example, pcaReduce is an agglomerative clustering algorithm that clusters cells based on principal components [176]. pcaReduce clusters cells by merging cell groups iteratively in each step using data projected by top principal components. ZIFA is another dimensionality reduction method that takes into account the drop-out events of scRNA-seq data [111]. SIMLR uses

kernel based learning to perform dimensionality reduction, clustering and visualization [177]. Apart from direct application of cell type classification, an unsupervised learning technique, denoising autoencoder (DAE) was also used to denoise expression data [178], and initialize training parameters for neural-network-based classifier [179]. For supervised methods, a broad range of machine learning methods were explored, including support vector machine (SVM) with PCA-transformed data [180], XGBoost [181], generalized linear model with elastic net [182], multi-task neural network [179,183] and Siamese neural network [29]. In particular, when trained on mouse scRNA-seq data sets, Siamese neural network has shown an improved performance compared to PCA based approach and multi-task neural network [29]. This successful application suggests that Siamese neural network can be broadly applied to other scRNA-seq data of other species. Therefore, this study aims at exploring the use of Siamese neural network to train cell type classifier for plant scRNA-seq data and discussing possible extensions for the original framework [29].

In this study, scRAN-seq data from four published studies were collected [170–172,174,175], which produced a merged single-cell expression dataset with 12,000 cells and 15 cell types for Arabidopsis root. Canonical correlation analysis (CCA) was performed to align cell types across different studies. Multiple types of cell type classifiers were evaluated using mean average precision. The results showed that random forest outperformed other classifiers. This work is the first comparative study of application of Siamese neural network and various machine learning classifiers to plant scRNA-seq data. Although the analyses presented here are only tentative exploration of this application, potential future directions of this study are discussed in **Discussion** section. As an example, this framework can be extended to identify novel cell types by performing resampling of distances among the cells from the same cell types.

6.2 Results

6.2.1 Overview of the dataset

Five recently published scRNA-seq data sets [170–172,174,175] of Arabidopsis roots were downloaded from GEO database (accessions: GSE123013, GSE121619, GSE122687 and GSE123818) and a webserver (http://wanglab.sippe.ac.cn/rootatlas/). Five data sets were merged into a single dataset which has 86,338 cells in total. Cells having no genes expressed and genes not expressing in any cells were removed. Expressions were normalized using log normalization method provided by Seurat R package [112]. Index of cell identity (ICI) was used to label the cell type for each cell (See section **6.4 Methods**). Number of cells for each cell type was summarized in **Table 6.1**. To balance the number of cells for each cell type, cells in each cell type were ranked by ICI scores and only top 1000 cells were selected. Only cell types with more than 1000 cells mapped were used and others

were removed. These filtering steps resulted in a final dataset of 12,000 cells, 31,819 genes for 12 cell types. To train machine learning models, this final dataset was split into a training dataset of 10,800 cells and a testing dataset of 1200 cells using a stratified sampling strategy that keeps equal number of cells in 12 cell types in both training and testing dataset. Then different machine learning approaches were trained using the training dataset and evaluated with testing dataset.

Table 6.1 Number of cells for each ICI assigned cell type. Numbers in bold font are selected cell types which have more than 1000 cells.

Cell type	Number of cells	Cell type	Number of cells
Endodermis	**24,791**	Protophleom	**1,696**
Trichoblast	**17,018**	Pericycle	**1,403**
Cortex	**12,705**	Phloem	**1,376**
Atrichoblast	**9,765**	Phloem CC	**1,028**
Late PPP	**6,889**	Quiescent center	822
Protoxylem	**2,813**	LRM	458
Columella	**2,697**	Late XPP	97
Meri Xylem	**2,339**		

* Late PPP: Late Phloem-Pole Pericycle
Meri Xylem: Meristematic Xylem
Phloem CC: Phloem Companion Cell
LRM: Lateral Root Meristem
Late XPP: Late Xylem-Pole Pericycle

6.2.2 Evaluation of cell type classification

Three types of deep neural network (DNN), triplet NN, contrastive NN and multi-task NN were tested for classifying Arabidopsis root cell types. To compare with performance of DNN models, support vector machine (SVM), k-nearest neighbors (KNN) and principal component analysis (PCA) were also used for classification. Although PCA is primarily designed as a dimension reduction technique, classification can be performed using KNN method taking PCA-generated lower dimensional embeddings as inputs (See **6.4 Methods**). Denoising autoencoder (DAE) was used to pre-train the weights of DNN models (See **6.4 Methods**). Mean average precision (MAP) was used as an evaluation metric for all classification approaches. As shown in **Table 6.2**, RF outperformed other classification methods on testing dataset. Triplet NN and contrastive NN achieved better overall MAP than multi-task NN but performed worse than simpler models SVM and RF. However, fine-tuning of hyperparameters usually can lead to an improved classification

result for DNN models. A more complete investigation of optimal hyperparameters will be conducted in the future. It has been reported that DNN models can be improved by using DAE to pretrain the weights [29]. However, we do not observe improved performance for DNN models when DAE was used to initialize their weights (**Table 6.3**). MAP for each specific cell type was shown in **Table 6.3**. Notably, Endodermis, Phloem, and Phloem CC (Companion Cell) had the best average cell-type-specific MAPs, suggesting that transcriptomic profiles of these cell types are distinctively different from other cell types.

Table 6.2 Performance of different classification performance. Classifiers were evaluated using MAP. Color scale indicates the MAP score (MAP ranges from 0 to 1)

Method name	Overall MAP
Triplet NN	0.729
Contrastive NN	0.737
Multi-task NN	0.141
DAE + triplet NN	0.575
DAE + contrastive NN	0.511
DAE + multi-task NN	0.242
SVM	0.827
RF	0.859
KNN	0.708
PCA	0.414

Table 6.3 Cell-type-specific MAP. MAP score for each cell type evaluated from each classifier was presented in this table. Color scale indicates the MAP score (MAP ranges from 0 to 1)

	Cell type					
Classification method	**Endodermis**	**Meristematic Xylem**	**Protophloem**	**Phloem**	**Pericycle**	**Cortext**
Triplet NN	0.358	0.950	0.743	0.315	0.125	0.576
Contrastive NN	0.401	0.942	0.723	0.370	0.189	0.589
Multi-task NN	0.000	0.000	0.000	0.000	0.000	0.000
PT triplet NN	0.896	0.422	0.170	0.861	0.456	0.163
PT contrastive NN	0.889	0.242	0.187	0.783	0.251	0.268
PT multi-task NN	0.493	0.117	0.071	0.754	0.002	0.000

SVM	0.973	0.748	0.743	0.979	0.753	0.662
RF	0.982	0.814	0.799	0.988	0.806	0.705
KNN	0.984	0.622	0.511	0.931	0.550	0.257
PCA	0.688	0.234	0.679	0.505	0.711	0.120
Average MAP	0.666	0.509	0.463	0.649	0.384	0.334

	Cell type					
Classification method	**Late PPP**	**Phloem CC**	**Trichoblast**	**Protoxylem**	**Atrichoblast**	**Columella**
Triplet NN	0.317	0.721	0.956	0.979	0.548	0.754
Contrastive NN	0.290	0.743	0.955	0.978	0.582	0.733
Multi-task NN	0.000	0.000	1.000	0.000	0.000	0.000
PT triplet NN	0.353	0.848	0.185	0.598	0.477	0.568
PT contrastive NN	0.239	0.835	0.184	0.581	0.226	0.589
PT multi-task NN	0.239	0.360	0.000	0.015	0.012	0.204
SVM	0.953	0.917	0.263	0.378	0.736	0.830
RF	0.944	0.932	0.310	0.568	0.828	0.868
KNN	0.363	0.834	0.281	0.417	0.614	0.764
PCA	0.472	0.556	0.076	0.257	0.154	0.163
Average MAP	0.417	0.675	0.421	0.477	0.418	0.547

6.3 Discussion

Although Amir et al. has comprehensively tested triplet NN and contrastive NN [29], there are other important research questions to be investigated related to identification of cell types. In this section, several future directions are discussed as extensions to the prediction pipeline. While there exist many possibilities, this section only focuses on identification of novel marker genes, discovery of novel cell types and construction of cell-type-specific regulatory network.

6.3.1 Identification of novel marker genes

Marker genes are genes showing disparate expression patterns across different cell types. In the published studies, marker genes have been broadly used to identify distinctive cell types [170,171,173–175]. However, identification of marker genes is often laborious and needs manual inspection. To automate this process, regularization-based feature selection can be applied on top of the trained NN model. As discussed in Chapter 2, this can be done by adding a one-to-one layer between the input layer and hidden layers (see **Methods** section in Chapter 2). As a validation step, the

identified marker genes can then be compared with existing genes to find common genes. Novel marker genes not identified by previous publications can be examined by visualization results of expression pattern.

6.3.2 Discovery of novel cell types

Cell populations of Arabidopsis roots are characterized by a high level of heterogeneity. Even within a cell population, composition of cell types usually is not homogeneous because sub-populations may exist [26]. Besides, it is not clear whether all cell types have been discovered for Arabidopsis root [175]. This highlights the importance of identifying novel cell types. However, such process usually involves manual inspection of the cell population structure, which is somewhat arbitrary [184,185]. This suggests that automating the identification of novel cell types is needed. This NN based cell prediction pipeline can be easily extended to discover novel cell types, with only a few extra steps of resampling analysis. Based on the trained triplet NN and contrastive NN, neural embeddings of all cell types can be easily generated. For each cell type in training dataset, a distribution of distance can be estimated by bootstrap sampling, which randomly samples large number of pairwise distances among cells with replacement and repeats for many times to approximate the true distribution of distance. For each new input vector, its closest cell type is determined as described in previous section. Distance between each new input vector and the centroid of its closest cell type is computed and p-value is calculated based on the estimated distribution and adjusted by Bonferroni correction. New input vector will be assigned with novel cell type if the final adjusted p-values is smaller than 0.05. This resampling-based method can discover the embeddings that stay distantly away from any known cell groups. To evaluate this method, cell types of interest can be held out from the training dataset and added into testing dataset. In this case, cell types of interest are unseen to the trained triplet NN and contrastive.

6.3.3 Construction of cell-type-specific regulatory network

An important question to be answered is how genes respond differently to abiotic stresses across different cell types. With scRNA-seq data, cell-type-specific response can be mapped at higher resolution than bulk RNA-seq data. The idea here is to use ConSReg to construct single cell regulatory networks. To infer cell-type-specific regulatory network, each root cell type can be compared to QC cells to generate a list of DEGs. Fold changes of DEGs, along with DAP-seq and ATAC-seq data, will be used as input data for ConSReg to construct cell-type-specific regulatory networks for root scRNA-seq data (See **Chapter 2** for more details). Important regulators are identified as those TFs with importance scores greater than 0.5. This combination of cell type

classification and regulatory network inference can potentially provide new insights into cell-type-specific stress regulation.

6.4 Methods

6.4.1 Dataset preprocessing

scRNA-seq dataset. scRNA-seq data of root cells from four publications were downloaded from the GEO website [186] and a web server (http://wanglab.sippe.ac.cn/rootatlas/). For each dataset, raw counts were used as input data. The five data sets were merged. Cells having no genes expressed and genes not expressing in any cells were removed. These steps resulted in a total of 86,338 cells and 37, 624 genes.

Normalization using standard workflow. Raw count matrices were normalized using Seurat R package [112]. Seurat package provides a standard workflow for normalizing the data by global-scaling log normalization. This normalization procedure was performed on training, testing, and validation data sets separately with default settings.

Assigning cell type labels using ICI. ICI score for each cell was computed using R scripts provided by a previous publication [118]. The predefined marker genes and their specificity (Spec) scores for the 15 root cell types were provided as supplementary file of the publication. The provided 15 root cell types include Trichoblast, Cortext, LRM (Lateral Root Meristem), Late PPP (Late Phloem-Pole Pericycle), Protophloem, Meristematic Xylem, Phloem CC (Companion Cell), Protoxylem, Phloem, Pericycle, Endodermis, Atrichoblast, Columella, QC (Quiescent Center), and Late XPP (Xylem-Pole Pericycle). The scripts compute ICI score by averaging expression of all genes in the predefined set of marker genes and weight each gene by its Spec score for the specific cell type. For each cell, ICI score was computed for each cell type, representing a similarity to each cell type. Cell type with the highest ICI score was assigned to the cell as final cell type label.

6.4.2 Machine leaning classification

Several common machine learning approaches were evaluated for the task of cell type classification, including support vector machine (SVM), K nearest neighbors (KNN), random forest (RF), Multi-task simple neural network, Siamese neural network with triple loss (triplet DNN), Siamese neural network with contrastive loss (contrastive NN). Python library scikit-learn was used to perform classification with KNN, SVM, and RF. Python library Keras was used to perform classification with multi-task NN, Triplet NN, and Contrastive NN. All implementations of neural networks were modified based on the GitHub repository provided by the published study [29]. The

entire dataset was split into 90% for training (10800 cells) and 10% for testing (1200 cells). Machine learning models trained with training dataset were evaluated with testing dataset. Each machine learning approach is briefly described below.

KNN. KNN is a commonly used simple classifier that does not have explicit training process. KNN makes new prediction by first computing Euclidean distance between the new input vector and every feature vector in the training dataset. Then the top K nearest neighbors are used for new prediction. In the last step, class label of the new input vector is determined by majority vote among the K nearest neighbors. The only hyperparameter for KNN is K, the number of top nearest neighbors. K is set to be 50 in the analysis. KNN is fast and simple, which makes it a first choice of machine learning classifier in many cases when computation resource is limited.

RF. RF is a tree-based machine learning approach built on a collection of decision trees. For each decision tree, a subset of training examples is randomly sampled as inputs and a subset of features are randomly sampled to split each tree node. The final class label is determined by majority vote. Number of trees (N) for RF is an important hyperparameter that can impact the model performance. Here, N was set as 50.

SVM. SVM is a machine learning classifier that maximizes the margin between different classes in a high dimensional space transformed by kernel function. Depending on kernel function, SVM can be a linear classifier (linear kernel) or a non-linear classifier (e.g., Gaussian kernel). To best capture the complex gene-gene relationships that characterize the cell type, Gaussian kernel was used to train SVM classifier.

Multi-task NN. Multi-task NN refers to a basic type of neural network that uses densely connected layer as input layer and hidden layers. The output layer has number of neurons equal to number of cell types (12 cell types). Architecture of multi-task NN is demonstrated in **Figure 6.1 A**. Briefly, input layer has number of neurons equal to number of genes used for classification (37, 624 genes) and three hidden layers were used, of which each has 586, 256, and 100 neurons. The last layer is an output layer to which a softmax is applied to ensure output scores are summed to 1.

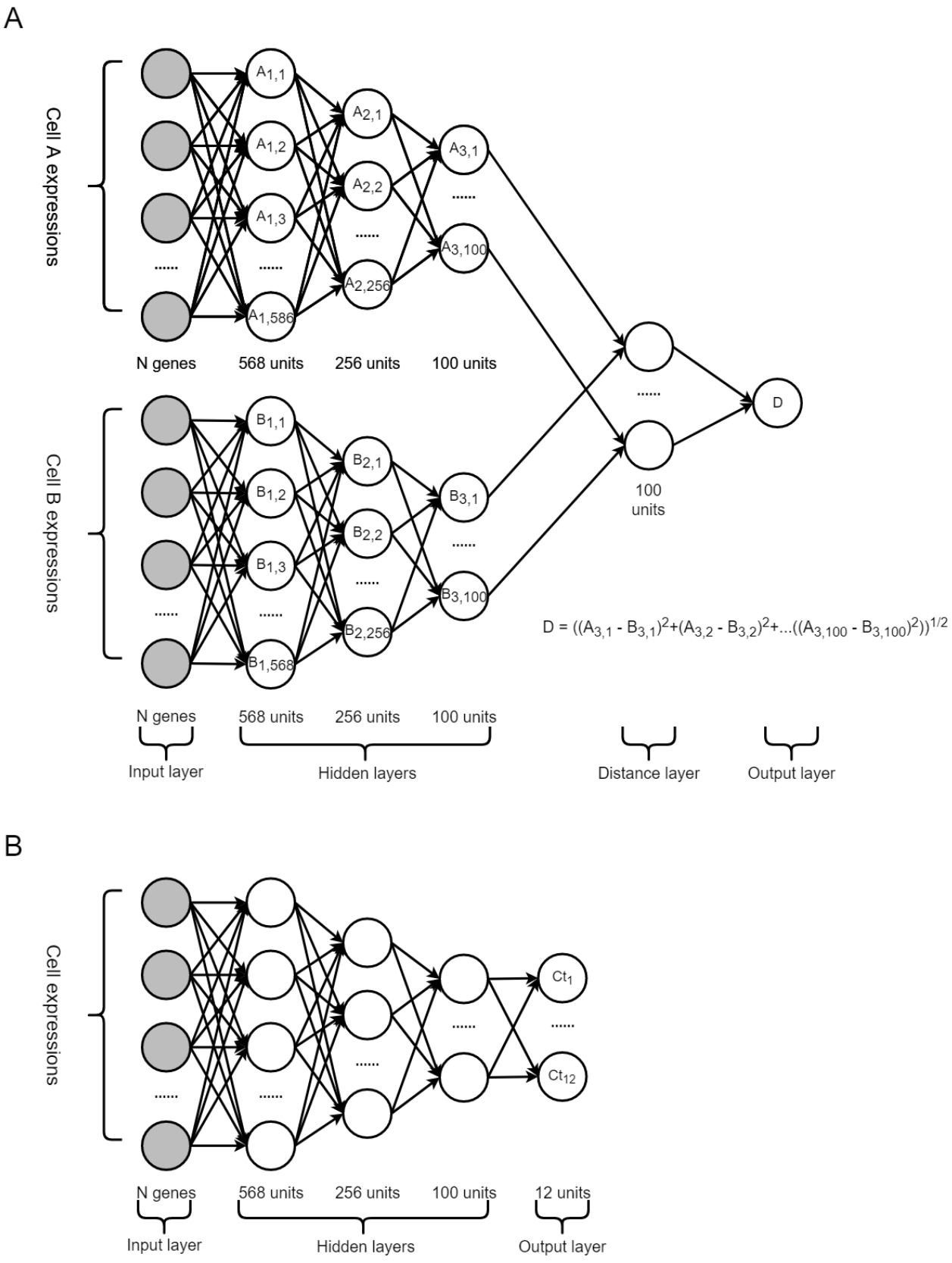


Figure 6.1 Schematic demonstration of architecture for each type of neural network. (A) Architecture of multi-task NN. Ct represents a cell type (12 cell type were used in total for classification) **(B)** Architecture of Siamese NN, which was used for both triplet NN and contrastive NN. The distance layer computes a vector of distance between the last two hidden layers A_3 and B_3. This distance was then used in the objective function of triplet NN and contrastive NN to train cell type classifier.

Triplet NN. Triplet NN is the implementation of Siamese neural network with triplet loss function. The use of triplet loss function was discussed in a published study [29]. Briefly, Siamese DNN consists of two subnetworks which have identical architecture and weights. The two neural networks connect to the same distance layer which computes a vector of distance between the last two hidden layers in the two subnetworks. The last two hidden layers are lower dimensional embeddings of original feature vectors. Architecture of Siamese NN is demonstrated in **Figure 6.1 B.** In this work, number of neurons in input layer is equal to number of genes used for classification (37, 624). Numbers of neurons used in three hidden layers are 586, 256, and 100. In training dataset, each scRNA-seq expression profile is an "anchor" that can be paired with positive example and negative example. Positive examples are those labeled with the same cell type with anchor and negative examples are those with different cell type. For each anchor, it will be paired with a positive example and a negative example, which forms a group of triplets. Then for each group of triplets, anchor-positive and anchor-negative pairs will be respectively fed into triplet NN. Based on the discussion in [187] and [29], the loss function of triplet NN can be written as:

$$L(D)max\left\{,\left(\sum_{i=1}^{T}\left(D_{a,p}^{i}\right)^{2}-\left(D_{a,n}^{i}\right)^{2}+m\right)\right\}$$

Where T is the number of groups of triplets. $D_{a,p}^{i}$ is the Euclidean distance between anchor and positive samples and $D_{a,n}^{i}$ is the Euclidean distance between anchor and negative samples. m is a hyperparameter that represents the margin between $\left(D_{a,p}^{i}\right)^{2}$ and $\left(D_{a,n}^{i}\right)^{2}$.

To ensure that triplet NN can be effectively trained, the groups of triplets need to include anchor-positive pairs with large distances and anchor-negative pairs with small distances. These are the hard training examples that enforce the model to learn effectively. As discussed in [29], batch hard loss function is used to generate hard training examples. In each iteration of optimization, *M* cell types which have *K* cells in each are sampled to generate a mini-batch. In this mini-batch, losses of hard training examples are selected and summed up as final loss value for the mini-batch. A slight modification of batch hard loss function was made in this study to include more training samples in each mini-batch. Instead of using one pair of hardest anchor-positive and anchor-negative respectively for each anchor, top *k* pairs of hardest pairs are selected for each each anchor. The batch hard loss function therefore can be written as:

$$L'(D)=\left\{0,\sum_{i=1}^{M}\sum_{j=1}^{K}\left[topmax(k,P_{j}^{i})-\ topmin(k,N_{j}^{i})+m\ \right]\right\}$$

Where P_{j}^{i} is the set of distances between *j*th cell from *i*th cell type and all other cells in *i*th cell type (anchor-positive pairs) and N_{j}^{i} is the set of distances between *j*th cell from *i*th cell type and all

other cells not from *i*th cell type (anchor-negative pairs). $topmax(k, P_j^i)$ selects the top k pairs with largest distances in P_j^i and sums the selected distances. $topmin(k, N_j^i)$ selects the top k pairs with smallest distances in N_j^i and sums the selected distances. This gives k pairs of anchor-positive sample pairs and k pairs of anchor-negative sample pairs for each anchor. In our analysis k was set as 10.

Contrastive NN. Contrastive NN is an implementation of Siamese neural network with contrastive loss function. Its use for cell type classification has been discussed in a published study [29]. In this work, contrastive NN was constructed using the same neural network architecture as triplet NN (See **Figure 6.1 B**). The difference here is that contrastive NN uses paired samples which pair the cell assigned with same/different cell types. The idea is to penalize large distances between samples of same cell type and small distances between samples of different cell types. The loss function of Contrastive NN can be written as:

$$L(Y, D) = \sum_{i=1}^{P} (Y^i) \frac{1}{2} (D)^2 + (1 - Y^i) \frac{1}{2} (\{0, m - D\})^2$$

Where P represent number of pairs of training samples. $Y^i = 1$ if two samples in the *i*th pair are assigned with same cell type and $Y^i = 0$ if not. D is the Euclidean distance between the two samples in each pair, computed using the last hidden layers of the two sub-networks. m is a hyperparameter representing the margin between two samples assigned with different cell types, usually set to 1. Please see [29] for more details about contrastive loss function.

6.4.3 Cell type prediction

Identification of existing cell types. For KNN, SVM, RF and multi-task NN, training dataset was used to train the models that can directly predict cell type label. The trained model was then used to predict cell type labels for testing dataset. For triplet NN and contrastive NN, training dataset was used to trained models that predict neural embeddings of the original feature vectors of in training dataset. For each new input vector from testing dataset, the trained model was first used to predict a neural embedding and this embedding was compared to all neural embeddings of the training dataset. The final cell type label was determined by majority vote of *m* nearest neighbors. Here we set $m = 50$.

6.5 Conclusions

In this study, a compendium of Arabidopsis root scRNA-seq data was compiled. Various machine approaches were tested to classify cell types for Arabidopsis root scRNA-seq data,

including NN-based prediction methods framed in a recent publication [29]. The classification results showed random forest has outperformed other approaches. However, a more complete investigation of optimal hyperparameters will be conducted in the future, which may lead to a different conclusion. Detailed results for each cell type suggested that Endodermis, Phloem and Phloem CC are distinctively different from other cell types in respect to transcriptomic profiles. Finally, several future directions for this framework were also discussed to extend the existing framework.

Appendix B: List of supplementary files

All supplementary files can be found in zipped file named “all_supp_file.zip”

Table/file name in this dissertation	File name in external file
supplementary file 2.1	supp_file_2_1.zip
supplementary file 2.2	supp_file_2_2.zip
supplementary table 2.1	supp_table_2_1.xlsx
supplementary table 2.2	supp_table_2_2.xlsx
supplementary table 2.3	supp_table_2_3.xlsx
supplementary table 2.4	supp_table_2_4.xlsx
supplementary table 2.5	supp_table_2_5.xlsx
supplementary table 2.6	supp_table_2_6.xlsx
supplementary table 2.7	supp_table_2_7.xlsx
supplementary table 2.8	supp_table_2_8.xlsx
supplementary table 3.1	supp_table_3_1.xlsx
supplementary table 3.2	supp_table_3_2.xlsx
supplementary table 3.3	supp_table_3_3.xlsx
supplementary table 3.4	supp_table_3_4.xlsx
supplementary table 3.5	supp_table_3_5.xlsx

References

1. Cramer, G. R., Urano, K., Delrot, S., Pezzotti, M. & Shinozaki, K. Effects of abiotic stress on plants: a systems biology perspective. *BMC Plant Biol.* **11**, 163 (2011).
2. Wang, H., Wang, H., Shao, H. & Tang, X. Recent Advances in Utilizing Transcription Factors to Improve Plant Abiotic Stress Tolerance by Transgenic Technology. *Front. Plant Sci.* (2016). doi:10.3389/fpls.2016.00067
3. Wang, Y. & Frei, M. Stressed food - The impact of abiotic environmental stresses on crop quality. *Agriculture, Ecosystems and Environment* (2011). doi:10.1016/j.agee.2011.03.017
4. Asensi-Fabado, M. A., Amtmann, A. & Perrella, G. Plant responses to abiotic stress: The chromatin context of transcriptional regulation. *Biochimica et Biophysica Acta - Gene Regulatory Mechanisms* (2017). doi:10.1016/j.bbagrm.2016.07.015
5. Mickelbart, M. V., Hasegawa, P. M. & Bailey-Serres, J. Genetic mechanisms of abiotic stress tolerance that translate to crop yield stability. *Nat. Rev. Genet.* **16**, 237–251 (2015).
6. Verma, V., Ravindran, P. & Kumar, P. P. Plant hormone-mediated regulation of stress responses. *BMC Plant Biol.* (2016). doi:10.1186/s12870-016-0771-y
7. Hussain, R. M., Ali, M., Feng, X. & Li, X. The essence of NAC gene family to the cultivation of drought-resistant soybean (Glycine max L. Merr.) cultivars. *BMC Plant Biol.* (2017). doi:10.1186/s12870-017-1001-y
8. Nakashima, K., Ito, Y. & Yamaguchi-Shinozaki, K. Transcriptional regulatory networks in response to abiotic stresses in Arabidopsis and grasses. *Plant Physiol.* **149**, 88–95 (2009).
9. Haak, D. C. *et al.* Multilevel Regulation of Abiotic Stress Responses in Plants. *Front. Plant Sci.* **8**, 1564 (2017).
10. Nakashima, K., Yamaguchi-Shinozaki, K. & Shinozaki, K. The transcriptional regulatory network in the drought response and its crosstalk in abiotic stress responses including drought, cold, and heat. *Front. Plant Sci.* (2014). doi:10.3389/fpls.2014.00170
11. Zhang, J., Jia, W., Yang, J. & Ismail, A. M. Role of ABA in integrating plant responses to drought and salt stresses. in *Field Crops Research* **97**, 111–119 (2006).
12. Fujita, Y., Fujita, M., Shinozaki, K. & Yamaguchi-Shinozaki, K. ABA-mediated transcriptional regulation in response to osmotic stress in plants. *Journal of Plant Research* (2011). doi:10.1007/s10265-011-0412-3
13. Lata, C. & Prasad, M. Role of DREBs in regulation of abiotic stress responses in plants. *Journal of Experimental Botany* (2011). doi:10.1093/jxb/err210
14. Taylor-Teeples, M. *et al.* An Arabidopsis gene regulatory network for secondary cell wall synthesis. *Nature* **517**, 571–575 (2014).
15. Song, L. *et al.* A transcription factor hierarchy defines an environmental stress response

network. *Science* **354**, 598+ (2016).

16. O'Malley, R. C. *et al.* Cistrome and Epicistrome Features Shape the Regulatory DNA Landscape. *Cell* **166**, 1598 (2016).
17. Lu, Z., Hofmeister, B. T., Vollmers, C., DuBois, R. M. & Schmitz, R. J. Combining ATAC-seq with nuclei sorting for discovery of cis-regulatory regions in plant genomes. *Nucleic Acids Res.* **45**, (2016).
18. Malliaros, F. D. & Vazirgiannis, M. Clustering and community detection in directed networks: A survey. *Phys. Rep.* **533**, 95–142 (2013).
19. Bartlett, A. *et al.* Mapping genome-wide transcription-factor binding sites using DAP-seq. *Nat. Protoc.* **12**, 1659–1672 (2017).
20. Lu, Z., Hofmeister, B. T., Vollmers, C., DuBois, R. M. & Schmitz, R. J. Combining ATAC-seq with nuclei sorting for discovery of cis-regulatory regions in plant genomes. *Nucleic Acids Res.* **45**, e41 (2017).
21. Wilkins, O. *et al.* EGRINs (Environmental Gene Regulatory Influence Networks) in Rice That Function in the Response to Water Deficit, High Temperature, and Agricultural Environments. *Plant Cell* **28**, 2365–2384 (2016).
22. Tannenbaum, M. *et al.* Regulatory chromatin landscape in Arabidopsis thaliana roots uncovered by coupling INTACT and ATAC-seq. *Plant Methods* (2018). doi:10.1186/s13007-018-0381-9
23. Maher, K. A. *et al.* Profiling of Accessible Chromatin Regions across Multiple Plant Species and Cell Types Reveals Common Gene Regulatory Principles and New Control Modules. *Plant Cell* **30**, 15–36 (2018).
24. Sijacic, P., Bajic, M., McKinney, E. C., Meagher, R. B. & Deal, R. B. Changes in chromatin accessibility between Arabidopsis stem cells and mesophyll cells illuminate cell type-specific transcription factor networks. *Plant J.* **94**, 215–231 (2018).
25. Balaji, S., Babu, M. M., Iyer, L. M., Luscombe, N. M. & Aravind, L. Comprehensive Analysis of Combinatorial Regulation using the Transcriptional Regulatory Network of Yeast. *J. Mol. Biol.* **360**, 213–227 (2006).
26. Liu, S. & Trapnell, C. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research* (2016). doi:10.12688/f1000research.7223.1
27. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental and Molecular Medicine* (2018). doi:10.1038/s12276-018-0071-8
28. Dinneny, J. R. *et al.* Cell identity mediates the response of Arabidopsis roots to abiotic stress. *Science* **320**, 942–5 (2008).

29. Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z. & Bar-Joseph, Z. A web server for comparative analysis of single-cell RNA-seq data. *Nat. Commun.* (2018). doi:10.1038/s41467-018-07165-2
30. Athar, A. *et al.* ArrayExpress update - From bulk to single-cell expression data. *Nucleic Acids Res.* (2019). doi:10.1093/nar/gky964
31. Krasensky, J. & Jonak, C. Drought, salt, and temperature stress-induced metabolic rearrangements and regulatory networks. *Journal of Experimental Botany* (2012). doi:10.1093/jxb/err460
32. Golldack, D., Lüking, I. & Yang, O. Plant tolerance to drought and salinity: Stress regulating transcription factors and their functional significance in the cellular transcriptional network. *Plant Cell Reports* (2011). doi:10.1007/s00299-011-1068-0
33. Serin, E. A. R., Nijveen, H., Hilhorst, H. W. M. & Ligterink, W. Learning from Co-expression Networks: Possibilities and Challenges. *Front. Plant Sci.* **7**, (2016).
34. Franco-Zorrilla, J. M. *et al.* DNA-binding specificities of plant transcription factors and their potential to define target genes. *Proc. Natl. Acad. Sci.* **111**, 2367–2372 (2014).
35. Maher, K. A. *et al.* Profiling of Accessible Chromatin Regions across Multiple Plant Species and Cell Types Reveals Common Gene Regulatory Principles and New Control Modules. *Plant Cell* **30**, 15–36 (2018).
36. Cumbie, J. S., Filichkin, S. A. & Megraw, M. Improved DNase-seq protocol facilitates high resolution mapping of DNase I hypersensitive sites in roots in Arabidopsis thaliana. *Plant Methods* **11**, 42 (2015).
37. Zhang, W., Zhang, T., Wu, Y. & Jiang, J. Genome-Wide Identification of Regulatory DNA Elements and Protein-Binding Footprints Using Signatures of Open Chromatin in Arabidopsis. *Plant Cell* **24**, 2719–2731 (2012).
38. Butte, a J. & Kohane, I. S. Mutual Information Relevance Networks:Functional Genomic Clustering Using Pairwise Entropy Measurements. *Pac. Symp. Biocomput.* **426**, 418–29 (2000).
39. Margolin, A. A. *et al.* ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics* **7**, S7 (2006).
40. Faith, J. J. *et al.* Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol.* **5**, 0054–0066 (2007).
41. Meyer, P. E., Kontos, K., Lafitte, F. & Bontempi, G. Information-theoretic inference of large transcriptional regulatory networks. *Eurasip J. Bioinforma. Syst. Biol.* **2007**, (2007).
42. Yuan, Y., Li, C. T. & Windram, O. Directed Partial Correlation: Inferring Large-Scale Gene Regulatory Network through Induced Topology Disruptions. *PLoS One* (2011).

doi:10.1371/journal.pone.0016835

43. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* (2008). doi:10.1186/1471-2105-9-559
44. Mordelet, F. & Vert, J. P. SIRENE: Supervised inference of regulatory networks. in *Bioinformatics* **24**, (2008).
45. Ni, Y. *et al.* A Machine Learning Approach to Predict Gene Regulatory Networks in Seed Development in Arabidopsis. *Front. Plant Sci.* **7**, (2016).
46. Haury, A.-C., Mordelet, F., Vera-Licona, P. & Vert, J.-P. TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. *BMC Syst. Biol.* **6**, 145 (2012).
47. Liu, L. Z., Wu, F. X. & Zhang, W. J. A group LASSO-based method for robustly inferring gene regulatory networks from multiple time-course datasets. *BMC Syst. Biol.* **8**, (2014).
48. Omranian, N., Eloundou-Mbebi, J. M. O., Mueller-Roeber, B. & Nikoloski, Z. Gene regulatory network inference using fused LASSO on multiple data sets. *Sci Rep* **6**, 20533 (2016).
49. Altarawy, D., Eid, F.-E. & Heath, L. S. PEAK: Integrating Curated and Noisy Prior Knowledge in Gene Regulatory Network Inference. *J. Comput. Biol.* **24**, 863–873 (2017).
50. de Luis Balaguer, M. A. *et al.* Predicting gene regulatory networks by combining spatial and temporal gene expression data in Arabidopsis root stem cells . *Proc. Natl. Acad. Sci.* (2017). doi:10.1073/pnas.1707566114
51. Desai, J. S., Sartor, R. C., Lawas, L. M., Jagadish, S. V. K. & Doherty, C. J. Improving Gene Regulatory Network Inference by Incorporating Rates of Transcriptional Changes. *Sci. Rep.* (2017). doi:10.1038/s41598-017-17143-1
52. Varala, K. *et al.* Temporal transcriptional logic of dynamic regulatory networks underlying nitrogen signaling and use in plants. *Proc. Natl. Acad. Sci.* (2018). doi:10.1073/pnas.1721487115
53. Jin, J. *et al.* PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. *Nucleic Acids Res.* gkw982 (2016). doi:10.1093/nar/gkw982
54. Davuluri, R. V *et al.* AGRIS: Arabidopsis Gene Regulatory Information Server, an information resource of Arabidopsis cis -regulatory elements and transcription factors. *BMC Bioinformatics* **4**, 25 (2003).
55. Yilmaz, A. *et al.* GRASSIUS: a platform for comparative regulatory genomics across the grasses. *Plant Physiol.* **149**, 171–80 (2009).
56. Chen, F. *et al.* Arabidopsis Phytochrome A Directly Targets Numerous Promoters for Individualized Modulation of Genes in a Wide Range of Pathways. *Plant Cell* **26**, 1949–1966 (2014).

57. Chen, F. *et al.* Photoreceptor partner FHY1 has an independent role in gene modulation and plant development under far-red light. *Proc. Natl. Acad. Sci.* **111**, 11888–11893 (2014).
58. Fan, M. *et al.* The bHLH Transcription Factor HBI1 Mediates the Trade-Off between Growth and Pathogen-Associated Molecular Pattern-Triggered Immunity in Arabidopsis. *Plant Cell* **26**, 828–841 (2014).
59. Song, L. *et al.* A transcription factor hierarchy defines an environmental stress response network. *Science (80-.).* **354**, aag1550–aag1550 (2016).
60. Shani, E. *et al.* Plant Stress Tolerance Requires Auxin-Sensitive Aux/IAA Transcriptional Repressors. *Curr. Biol.* **27**, 437–444 (2017).
61. Liu, S., Kracher, B., Ziegler, J., Birkenbihl, R. P. & Somssich, I. E. Negative regulation of ABA Signaling By WRKY33 is critical for Arabidopsis immunity towards Botrytis cinerea 2100. *Elife* **4**, (2015).
62. Kulkarni, S. R., Vaneechoutte, D., Van de Velde, J. & Vandepoele, K. TF2Network: predicting transcription factor regulators and gene regulatory networks in Arabidopsis using publicly available binding site information. *Nucleic Acids Res.* (2017). doi:10.1093/nar/gkx1279
63. Austin, R. S. *et al.* New BAR tools for mining expression data and exploring Cis-elements in Arabidopsis thaliana. *Plant J.* **88**, 490–504 (2016).
64. Chow, C. N. *et al.* PlantPAN3.0: a new and updated resource for reconstructing transcriptional regulatory networks from ChIP-seq experiments in plants. *Nucleic Acids Res.* (2019). doi:10.1093/nar/gky1081
65. Reimand, J. *et al.* g:Profiler-a web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Res.* **44**, W83–W89 (2016).
66. Wang, D., Rendon, A., Ouwehand, W. & Wernisch, L. Transcription factor co-localization patterns affect human cell type-specific gene expression. 1–12 (2012).
67. Brooks, M. D. *et al.* Network Walking charts transcriptional dynamics of nitrogen signaling by integrating validated and predicted genome-wide interactions. *Nat. Commun.* **10**, 1569 (2019).
68. Bargmann, B. O. R. *et al.* TARGET: A Transient Transformation System for Genome-Wide Transcription Factor Target Discovery. *Mol. Plant* (2013). doi:10.1093/mp/sst010
69. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
70. Meinshausen, N. & Bühlmann, P. Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **72**, 417–473 (2010).
71. Natarajan, A., Yardimci, G. G., Sheffield, N. C., Crawford, G. E. & Ohler, U. Predicting cell-

type-specific gene expression from regions of open chromatin. *Genome Res.* **22**, 1711–1722 (2012).

72. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
73. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–4 (2015).
74. Singh, R., Lanchantin, J., Robins, G. & Qi, Y. DeepChrome: Deep-learning for predicting gene expression from histone modifications. in *Bioinformatics* **32**, i639–i648 (2016).
75. Li, Y., Chen, C. Y. & Wasserman, W. W. Deep feature selection: Theory and application to identify enhancers and promoters. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9029**, 205–217 (2015).
76. Kaufmann, K., Pajoro, A. & Angenent, G. C. Regulation of transcription in plants: Mechanisms controlling developmental switches. *Nature Reviews Genetics* **11**, 830–842 (2010).
77. Chow, C. N. *et al.* PlantPAN 2.0: An update of Plant Promoter Analysis Navigator for reconstructing transcriptional regulatory networks in plants. *Nucleic Acids Res.* **44**, D1154–D1164 (2016).
78. Weirauch, M. T. *et al.* Determination and Inference of Eukaryotic Transcription Factor Sequence Specificity. *Cell* **158**, 1431–1443 (2014).
79. Fujita, M. *et al.* Crosstalk between abiotic and biotic stress responses: a current view from the points of convergence in the stress signaling networks. *Current Opinion in Plant Biology* **9**, 436–442 (2006).
80. Müller, M. & Munné-Bosch, S. Ethylene Response Factors: A Key Regulatory Hub in Hormone and Stress Signaling. *Plant Physiol.* **169**, 32–41 (2015).
81. Sakamoto, H. Arabidopsis Cys2/His2-Type Zinc-Finger Proteins Function as Transcription Repressors under Drought, Cold, and High-Salinity Stress Conditions. *PLANT Physiol.* **136**, 2734–2746 (2004).
82. Mittler, R. *et al.* Gain- and loss-of-function mutations in Zat10 enhance the tolerance of plants to abiotic stress. *FEBS Lett.* **580**, 6537–6542 (2006).
83. Xie, Y., Mao, Y., Lai, D., Zhang, W. & Shen, W. H2 Enhances Arabidopsis Salt Tolerance by Manipulating ZAT10/12-Mediated Antioxidant Defence and Controlling Sodium Exclusion. *PLoS One* **7**, (2012).
84. Rossel, J. B. *et al.* Systemic and Intracellular Responses to Photooxidative Stress in

Arabidopsis. *PLANT CELL ONLINE* **19**, 4091–4110 (2007).

85. Gordon, M. J., Carmody, M., Albrecht, V. & Pogson, B. Systemic and Local Responses to Repeated HL Stress-Induced Retrograde Signaling in Arabidopsis. *Front. Plant Sci.* **3**, 303 (2012).
86. Nguyen, X. C. *et al.* Identification of a C2H2-type zinc finger transcription factor (ZAT10) from Arabidopsis as a substrate of MAP kinase. *Plant Cell Rep.* **31**, 737–745 (2012).
87. Johnson, G. L. & Lapadat, R. Mitogen-activated protein kinase pathways mediated by ERK, JNK, and p38 protein kinases. *Science* **298**, 1911–2 (2002).
88. Weirauch, M. T. Gene Coexpression Networks for the Analysis of DNA Microarray Data. in *Applied Statistics for Network Biology: Methods in Systems Biology* (2011). doi:10.1002/9783527638079.ch11
89. Olsson, A. S. B., Engström, P. & Söderman, E. The homeobox genes ATHB12 and ATHB7 encode potential regulators of growth in response to water deficit in Arabidopsis. *Plant Mol. Biol.* **55**, 663–677 (2004).
90. Mishra, K. B. *et al.* Engineered drought tolerance in tomato plants is reflected in chlorophyll fluorescence emission. *Plant Sci.* **182**, 79–86 (2012).
91. Pruthvi, V., Narasimhan, R. & Nataraja, K. N. Simultaneous expression of abiotic stress responsive transcription factors, AtDREB2A, AtHB7 and AtABF3 improves salinity and drought tolerance in peanut (Arachis hypogaea L.). *PLoS One* **9**, (2014).
92. Singh, K. B. Transcriptional Regulation in Plants: The Importance of Combinatorial Control. *Plant Physiol.* **118**, 1111 LP – 1120 (1998).
93. Gerstein, M. B. *et al.* Architecture of the human regulatory network derived from ENCODE data. *Nature* **489**, 91–100 (2012).
94. Song, Q., Grene, R., Heath, L. S. & Li, S. Identification of regulatory modules in genome scale transcription regulatory networks. *BMC Syst. Biol.* **11**, 140 (2017).
95. Zhao, Y. *et al.* The ABA receptor PYL8 promotes lateral root growth by enhancing MYB77-dependent transcription of auxin-responsive genes. *Sci. Signal.* **7**, (2014).
96. Jung, C. *et al.* Overexpression of AtMYB44 Enhances Stomatal Closure to Confer Abiotic Stress Tolerance in Transgenic Arabidopsis. *PLANT Physiol.* (2007). doi:10.1104/pp.107.110981
97. Persak, H. & Pitzschke, A. Dominant repression by Arabidopsis transcription factor MYB44 causes oxidative damage and hypersensitivity to abiotic stress. *Int. J. Mol. Sci.* (2014). doi:10.3390/ijms15022517
98. Park, C.-M. Auxin Homeostasis in Plant Stress Adaptation Response. *Plant Signal. Behav.* **2**, 306–307 (2007).

99. Shulse, C. N. *et al.* High-throughput single-cell transcriptome profiling of plant cell types. *bioRxiv* (2018).
100. Heyman, J. *et al.* ERF115 Controls Root Quiescent Center Cell Division and Stem Cell Replenishment. *Science (80-.).* **342**, 860–863 (2013).
101. Mueller, S. *et al.* General Detoxification and Stress Responses Are Mediated by Oxidized Lipids through TGA Transcription Factors in Arabidopsis. *PLANT CELL ONLINE* **20**, 768–785 (2008).
102. Dubois, M., Claeys, H., Van den Broeck, L. & Inzï¿½, D. Time of day determines Arabidopsis transcriptome and growth dynamics under mild drought. *Plant Cell Environ.* **40**, 180–189 (2017).
103. Rawat, V. *et al.* Improving the annotation of Arabidopsis Lyrata using RNA-Seq data. *PLoS One* **10**, (2015).
104. Kilian, J. *et al.* The AtGenExpress global stress expression data set: Protocols, evaluation and model data analysis of UV-B light, drought and cold stress responses. *Plant J.* **50**, 347–363 (2007).
105. Schlaen, R. G. *et al.* The spliceosome assembly factor GEMIN2 attenuates the effects of temperature on alternative splicing and circadian rhythms. *Proc. Natl. Acad. Sci.* **112**, 9382–9387 (2015).
106. Gehan, M. A. *et al.* Natural variation in the C-repeat binding factor cold response pathway correlates with local adaptation of Arabidopsis ecotypes. *Plant J.* **84**, 682–693 (2015).
107. Suzuki, N. *et al.* ABA is required for plant acclimation to a combination of salt and heat stress. *PLoS One* **11**, (2016).
108. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: Challenges and opportunities. *Nature Methods* **14**, 565–571 (2017).
109. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat. Methods* **11**, 740–742 (2014).
110. Finak, G. *et al.* MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16**, (2015).
111. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16**, (2015).
112. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).
113. Boyle, A. P. *et al.* High-Resolution Mapping and Characterization of Open Chromatin across

the Genome. *Cell* **132**, 311–322 (2008).

114. Thurman, R. E. *et al.* The accessible chromatin landscape of the human genome. *Nature* **489**, 75–82 (2012).
115. Hesselberth, J. R. *et al.* Global mapping of protein-DNA interactions in vivo by digital genomic footprinting. *Nat. Methods* **6**, 283–289 (2009).
116. Li, S., Yamada, M., Han, X., Ohler, U. & Benfey, P. N. High-Resolution Expression Map of the Arabidopsis Root Reveals Alternative Splicing and lincRNA Regulation. *Dev. Cell* **39**, 508–522 (2016).
117. Kolesnikov, N. *et al.* ArrayExpress update-simplifying data submissions. *Nucleic Acids Res.* **43**, D1113–D1116 (2015).
118. Efroni, I., Ip, P. L., Nawy, T., Mello, A. & Birnbaum, K. D. Quantification of cell identity from single-cell gene expression profiles. *Genome Biol.* **16**, (2015).
119. Yu, G., Wang, L. G. & He, Q. Y. ChIP seeker: An R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics* **31**, 2382–2383 (2015).
120. Taylor-Teeples, M. *et al.* An Arabidopsis gene regulatory network for secondary cell wall synthesis. *Nature* **517**, 571–575 (2014).
121. Sparks, E. E. *et al.* Establishment of Expression in the SHORTROOT-SCARECROW Transcriptional Cascade through Opposing Activities of Both Activators and Repressors. *Dev. Cell* 1–12 (2016). doi:10.1016/j.devcel.2016.09.031
122. Lee, S. S., Lee, H., Abbeel, P. & Ng, A. Y. A. Efficient L1 regularized logistic regression. in *The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference* **21**, 401 (2006).
123. Yang, Y. & Zou, H. A fast unified algorithm for solving group-lasso penalize learning problems. *Stat. Comput.* **25**, 1129–1141 (2015).
124. Meier, L., Van De Geer, S. & Bühlmann, P. The group lasso for logistic regression. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **70**, 53–71 (2008).
125. Draper, N. R. & Smith, H. Applied regression analysis. in *Applied regression analysis* 709 (1981).
126. Deng, H. & Runger, G. Gene selection with guided regularized random forest. *Pattern Recognit.* **46**, 3483–3489 (2013).
127. Y.Benjamini and Y.Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B* **57**, 289–300 (1995).
128. Song, Q., Grene, R., Heath, L. S. & Li, S. Identification of regulatory modules in genome scale transcription regulatory networks. *BMC Syst. Biol.* **11**, (2017).
129. Yue, F. *et al.* A comparative encyclopedia of DNA elements in the mouse genome. *Nature*

515, 355–64 (2014).

130. Babu, M. M. & Teichmann, S. A. Evolution of transcription factors and the gene regulatory network in Escherichia coli. *Nucleic Acids Res* **31**, 1234–1244 (2003).
131. Franco-Zorrilla, J. M. *et al.* DNA-binding specificities of plant transcription factors and their potential to define target genes. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 2367–72 (2014).
132. Kulkarni, S. R., Vaneechoutte, D., Van de Velde, J. & Vandepoele, K. TF2Network: predicting transcription factor regulators and gene regulatory networks in Arabidopsis using publicly available binding site information. *bioRxiv* 1–28 (2017). doi:10.1101/173559
133. Dobrin, R., Beg, Q. K., Barabási, A.-L. & Oltvai, Z. N. Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network. *BMC Bioinformatics* **5**, 10 (2004).
134. Guelzim, N., Bottani, S., Bourgine, P. & Képès, F. Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet.* **31**, 60–63 (2002).
135. Shalgi, R., Lieber, D., Oren, M. & Pilpel, Y. Global and local architecture of the mammalian microRNA-transcription factor regulatory network. *PLoS Comput. Biol.* **3**, 1291–1304 (2007).
136. Bui-Xuan, B. M., Habib, M., Limouzy, V. & de Montgolfier, F. Algorithmic aspects of a general modular decomposition theory. *Discret. Appl. Math.* **157**, 1993–2009 (2009).
137. Pons, P. & Latapy, M. Computing communities in large networks using random walks. *Comput. Inf. Sci.* **3733**, 284–293 (2005).
138. Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004).
139. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **76**, (2007).
140. Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **74**, (2006).
141. Guimerà, R., Sales-Pardo, M. & Amaral, L. A. N. Module identification in bipartite and directed networks. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **76**, (2007).
142. Danon, L., Díaz-Guilera, A., Duch, J. & Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, P09008–P09008 (2005).
143. Goda, H. *et al.* The AtGenExpress hormone and chemical treatment data set: experimental design, data evaluation, model data analysis and data access. *Plant J.* **55**, 526–542 (2008).
144. Schmid, M. *et al.* A gene expression map of Arabidopsis thaliana development. *Nat. Genet.* **37**, 501–6 (2005).
145. Kumimoto, R. W., Zhang, Y., Siefers, N. & Holt, B. F. NF-YC3, NF-YC4 and NF-YC9 are

required for CONSTANS-mediated, photoperiod-dependent flowering in Arabidopsis thaliana. *Plant J.* **63**, 379–391 (2010).

146. Hackenberg, D., Keetman, U. & Grimm, B. Homologous NF-YC2 subunit from arabidopsis and tobacco is activated by photooxidative stress and induces flowering. *Int. J. Mol. Sci.* **13**, 3458–3477 (2012).
147. Rushton, P. J., Somssich, I. E., Ringler, P. & Shen, Q. J. WRKY transcription factors. *Trends in Plant Science* **15**, 247–258 (2010).
148. Journot-Catalino, N., Somssich, I. E., Roby, D. & Kroj, T. The transcription factors WRKY11 and WRKY17 act as negative regulators of basal resistance in Arabidopsis thaliana. *Plant Cell* **18**, 3289–3302 (2006).
149. Nakano, T., Suzuki, K., Fujimura, T. & Shinshi, H. Genome-Wide Analysis of the ERF Gene Family. *Plant Physiol.* **140**, 411–432 (2006).
150. van der Graaff, E., Dulk-Ras, a D., Hooykaas, P. J. & Keller, B. Activation tagging of the LEAFY PETIOLE gene affects leaf petiole development in Arabidopsis thaliana. *Development* **127**, 4971–4980 (2000).
151. Banno, H., Ikeda, Y., Niu, Q. W. & Chua, N. H. Overexpression of Arabidopsis ESR1 induces initiation of shoot regeneration. *Plant Cell* **13**, 2609–18 (2001).
152. Gu, Y. Q., Yang, C., Thara, V. K., Zhou, J. & Martin, G. B. Pti4 is induced by ethylene and salicylic acid, and its product is phosphorylated by the Pto kinase. *Plant Cell* **12**, 771–86 (2000).
153. Dubouzet, J. G. *et al.* OsDREB genes in rice, Oryza sativa L., encode transcription activators that function in drought-, high-salt- and cold-responsive gene expression. *Plant J.* **33**, 751–763 (2003).
154. Krishnaswamy, S., Verma, S., Rahman, M. H. & Kav, N. N. V. Functional characterization of four APETALA2-family genes (RAP2.6, RAP2.6L, DREB19 and DREB26) in Arabidopsis. *Plant Mol. Biol.* **75**, 107–127 (2011).
155. Tan, Q. K.-G. & Irish, V. F. The Arabidopsis zinc finger-homeodomain genes encode proteins with unique biochemical properties that are coordinately expressed during floral development. *Plant Physiol.* **140**, 1095–1108 (2006).
156. Zhang, L. *et al.* Ovate family protein1 interaction with BLH3 regulates transition timing from vegetative to reproductive phase in Arabidopsis. *Biochem. Biophys. Res. Commun.* **470**, 492–497 (2016).
157. Waese, J. *et al.* ePlant: Visualizing and Exploring Multiple Levels of Data for Hypothesis Generation in Plant Biology. *Plant Cell* tpc.00073.2017 (2017). doi:10.1105/tpc.17.00073
158. Adamic, L. A. & Adar, E. Friends and neighbors on the Web. *Soc. Networks* **25**, 211–230

(2003).

159. Berri, S. *et al.* Characterization of WRKY co-regulatory networks in rice and Arabidopsis. *BMC Plant Biol.* **9**, 1–22 (2009).
160. Hart, B. R. & Blumenthal, R. M. Unexpected coregulator range for the global regulator Lrp of Escherichia coli and Proteus mirabilis. *J. Bacteriol.* **193**, 1054–1064 (2011).
161. Kim, J. *et al.* The co-regulation mechanism of transcription factors in the human gene regulatory network. *Nucleic Acids Res.* **40**, 8849–61 (2012).
162. Sarachana, T. & Hu, V. W. Differential recruitment of coregulators to the RORA promoter adds another layer of complexity to gene (dys) regulation by sex hormones in autism. *Mol. Autism* **4**, 39 (2013).
163. Yang, M. Q., Koehly, L. M. & Elnitski, L. L. Comprehensive annotation of bidirectional promoters identifies co-regulation among breast and ovarian cancer genes. *PLoS Comput. Biol.* **3**, 733–742 (2007).
164. He, F. *et al.* Large-scale atlas of microarray data reveals the distinct expression landscape of different tissues in Arabidopsis. *Plant J.* **86**, 472–480 (2016).
165. Csárdi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Syst.* **1695**, 1695 (2006).
166. Langfelder, P., Zhang, B. & Horvath, S. Defining clusters from a hierarchical cluster tree: The Dynamic Tree Cut package for R. *Bioinformatics* **24**, 719–720 (2008).
167. Bass, J. I. F. *et al.* Using networks to measure similarity between genes: association index selection. *Nat. Methods* **10**, 1169–1176 (2013).
168. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Research* **25**, 1491–1498 (2015).
169. Wang, Y. & Navin, N. E. Advances and Applications of Single-Cell Sequencing Technologies. *Molecular Cell* (2015). doi:10.1016/j.molcel.2015.05.005
170. Ryu, K. H., Huang, L., Kang, H. M. & Schiefelbein, J. Single-Cell RNA Sequencing Resolves Molecular Relationships Among Individual Plant Cells. *Plant Physiol.* (2019). doi:10.1104/pp.18.01482
171. Shulse, C. N. *et al.* High-Throughput Single-Cell Transcriptome Profiling of Plant Cell Types. *Cell Rep.* (2019). doi:10.1016/j.celrep.2019.04.054
172. Denyer, T. *et al.* Spatiotemporal Developmental Trajectories in the Arabidopsis Root Revealed Using High-Throughput Single-Cell RNA Sequencing. *Dev. Cell* (2019). doi:10.1016/j.devcel.2019.02.022
173. Turco, G. M. *et al.* Molecular Mechanisms Driving Bistable Switch Behavior in Xylem Cell Differentiation. *bioRxiv* (2019). doi:10.1101/543983

174. Jean-Baptiste, K. *et al.* Dynamics of Gene Expression in Single Root Cells of Arabidopsis thaliana . *Plant Cell* (2019). doi:10.1105/tpc.18.00785
175. Zhang, T. Q., Xu, Z. G., Shang, G. D. & Wang, J. W. A Single-Cell RNA Sequencing Profiles the Developmental Landscape of Arabidopsis Root. *Mol. Plant* (2019). doi:10.1016/j.molp.2019.04.004
176. žurauskiene, J. & Yau, C. pcaReduce: Hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics* (2016). doi:10.1186/s12859-016-0984-y
177. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nat. Methods* (2017). doi:10.1038/nMeth.4207
178. Tan, J., Hammond, J. H., Hogan, D. A. & Greene, C. S. ADAGE-Based Integration of Publicly Available Pseudomonas aeruginosa Gene Expression Data with Denoising Autoencoders Illuminates Microbe-Host Interactions . *mSystems* (2016). doi:10.1128/msystems.00025-15
179. Lin, C., Jain, S., Kim, H. & Bar-Joseph, Z. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Res.* (2017). doi:10.1093/nar/gkx681
180. Wagner, F. & Yanai, I. Moana: A robust and scalable cell type classification framework for single-cell RNA-Seq data. *bioRxiv* (2018). doi:10.1101/456129
181. Lieberman, Y., Rokach, L. & Shay, T. CaSTLe - Classification of single cells by transfer learning: Harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments. *PLoS One* (2018). doi:10.1371/journal.pone.0205499
182. Pliner, H. A., Shendure, J. & Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *bioRxiv* (2019). doi:10.1101/538652
183. Xie, P. *et al.* SuperCT: a supervised-learning framework for enhanced characterization of single-cell transcriptomic profiles. *Nucleic Acids Res.* (2019). doi:10.1093/nar/gkz116
184. Usoskin, D. *et al.* Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat. Neurosci.* (2015). doi:10.1038/nn.3881
185. Luo, Y. *et al.* Single-cell transcriptome analyses reveal signals to activate dormant neural stem cells. *Cell* (2015). doi:10.1016/j.cell.2015.04.001
186. Barrett, T. *et al.* NCBI GEO: Archive for functional genomics data sets - Update. *Nucleic Acids Res.* **41**, (2013).
187. Schroff, F., Kalenichenko, D. & Philbin, J. FaceNet: A unified embedding for face recognition and clustering. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2015). doi:10.1109/CVPR.2015.7298682