공학박사학위논문

시계열 유전자 발현 데이터에서 스트레스 반응 유전자를 검출하기 위한 클러스터링 기반 통합 분석 기법

Clustering-Based Integrated Analysis of Time-Series Gene Expression Data to Identify Stress-Responsive Genes

2019년 2월

서울대학교 대학원
전기.컴퓨터공학부
안홍렬

Abstract

Clustering-Based Integrated Analysis of Time-Series Gene Expression Data to Identify Stress-Responsive Genes

Hongryul Ahn
Department of Electrical Engineering & Computer Science
College of Engineering
The Graduate School
Seoul National University

Microarray and RNA sequencing, highly parallel technologies for the measurement of intracellular RNA molecules, were developed in the 1990s and 2000s. They opened a new era of quantifying the amount of gene activation (expression) for every gene in a cell through a single experiment. Since then, gene expression data have been widely produced to investigate the change of the state of a cell, particularly in response to environmental stress, such as heat, drought, and cold, in plants. However, a cell is one of the most complicated systems in the universe. Understanding and modeling the system of a cell requires a huge amount of data, which we do not yet have. Thus, gene expression data analysis has to address the issue of the lack of data and the development of analytical procedures, models, and algorithms that work on small-sample-size data.

This doctoral study proposes computational methodologies that solve the problem of modeling a highly complex system with small-sample-size data based on clustering and integrated analysis. We can easily understand gene expression data in the format of machine learning data: genes as features and different conditions of samples as classes. In gene expression data, the number of features is generally much greater than the number of samples due to the high cost of measurement of a sample. Performing clustering analysis on gene expression data groups individual genes into several gene clusters, resulting in the reduction of the dimension of features. This doctoral study presents a method that uses clustering analysis to reduce the dimension of features. It shows the improvement of interpreting high-dimension and small-sample-size gene expression data.

In addition, the system of a cell consists of complicated interactions between genes, which leads to a computational problem known as high dependency between features. Introducing external information, domain data, and domain knowledge improves the modeling of relationships between genes to reflect real biological systems. This doctoral study proposes a method that introduces genetic data and knowledge into the analysis by constructing a template biological network. By combining the network with the condition-specific network derived from experimental data, it successfully explains the stress response mechanism of drought-resistant rice.

Moreover, gene expression data are measured at multiple time points along the time axis, which is called time-series data, to track the response of cells after drug or stress treatment. However, they often have a small number of time points, usually less than ten, and different intervals across different time-series samples because of the limitation that the cells die in the process of being measured. The sparsity and heterogeneity of time-domain data in gene expression data make it difficult to clarify the time-domain signals of genes. This study proposes a method to analyze time-

series gene expression data by using clustering analysis to extract the meaningful time-domain signal that is supported by many members of genes within the same cluster.

Lastly, clustering analysis is sensitive to the distribution of data objects. However, we do not yet know the distribution of genes in gene expression data. Thus, clustering algorithms for gene expression data are required to work on arbitrarily distributed data. The hierarchical clustering method has been the most widely used clustering method for gene expression data analysis, but it does not always work on arbitrarily distributed data. This study also proposes an improved version of the hierarchical clustering method to work on arbitrarily distributed data by combining effective recent clustering techniques, such as network representation, phase shifting, and cost-optimization-based tree integration.

In summary, this doctoral study proposes clustering-based computational methods for the analysis of gene expression data. Clustering analysis is used for dimension reduction, integration with biology-domain knowledge of genes, extraction of the time-domain signal, and development of clustering on arbitrarily distributed data. In addition, by applying it to actual stress data, this doctoral study explains the mechanism of drought-resistant rice, detects the cold-stress-responsive genes in Arabidopsis, and develops a new hierarchical clustering algorithm. The proposed methodology is expected to be useful for the analysis of other data with similar problems.

Keywords : Clustering, Network, Integrated Analysis, Time-series, Gene Expression Data, Stress-Responsive Genes

Student Number : 2012-23221

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A cell regulates the number of gene products (i.e., messenger RNAs and proteins) differentially depending on the external environment to maintain the system of the living organism. We call the process by which a cell generates a gene product from a gene "*gene expression*" and the values that measure how many gene products are in a cell "*gene expression data*". The gene products are the most important players that participate in all life activities of the cell, such as growth, differentiation, homeostasis, response to stimulation, and apoptosis. Gene expression data are a crucial resource for cell research, because they quantify the state of the cell and make the numerical analysis of the state of the cell possible. Figure 1.1 shows an example of gene expression data.

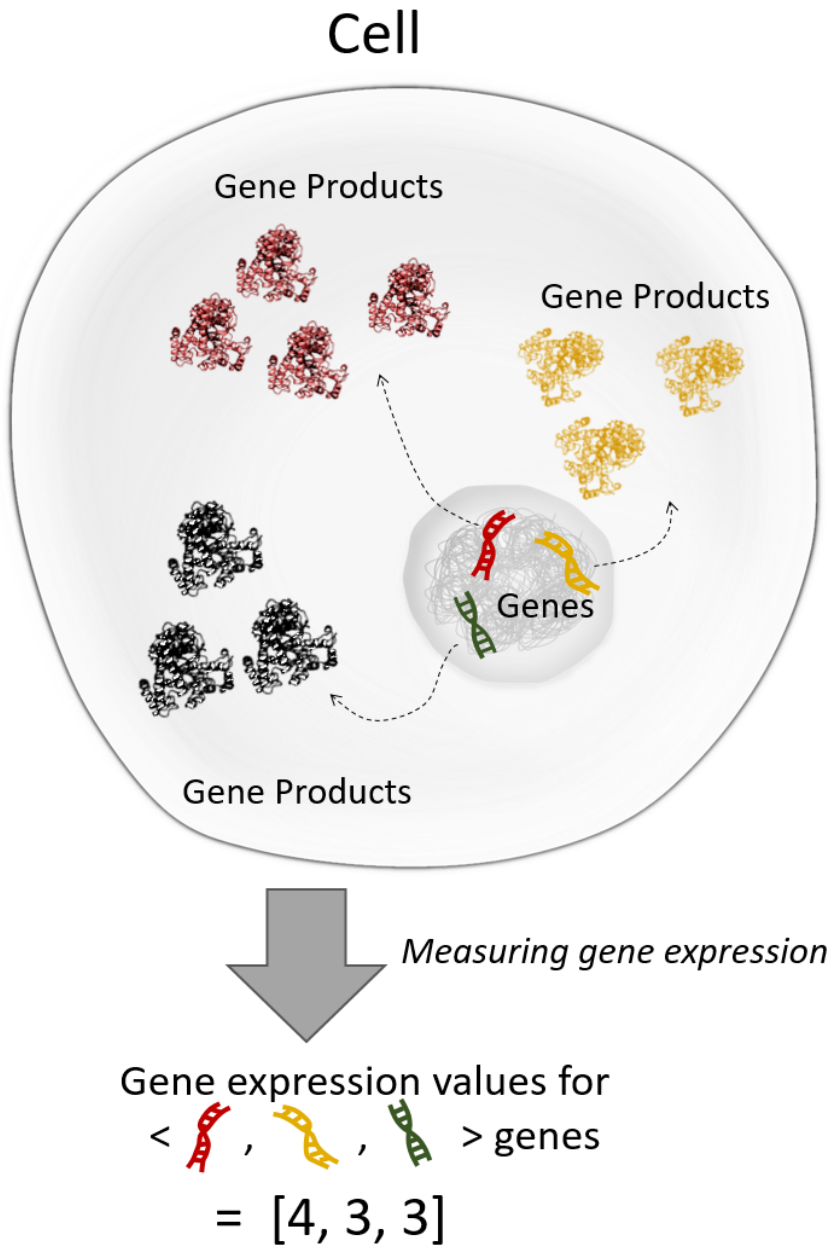

Figure 1.1. Example of measuring gene expression data. A nucleus inside a cell contains several kinds of genes (actually about 20,000). A cell produces gene products from genes differentially depending on the external environment. The process by which a cell generates a gene product from a gene is called "*gene expression*" and the values that measure how many gene products are in a cell is called "*gene expression data*". In this example, the gene expression values for the red, yellow, and green genes are [4, 3, 3]. Gene expression data numerically represent the state of the cell.

The advancement of technology to measure gene expression over the last 20 to 30 years has dramatically improved the volume and accuracy of gene expression data. Sanger sequencing [118] is an early technology to measure the gene expression level, but it has very low throughput. It allows the measurement of the gene expression level of one gene per experiment, resulting in unknown expression values for several genes. The later development of highly parallel technologies, such as microarrays [120] and RNA sequencing (RNA-Seq) [93], has made it possible to investigate the gene expression levels of all genes (about 20,000 in higher organisms) in a single experiment.

Since then, gene expression data have been widely produced in various fields of biological research, such as development, aging, disease diagnosis, drug response prediction, and genetic engineering, in different species, from humans to bacteria. Then, what are the problems in analyzing gene expression data?

Modeling a highly complex system with small-sample-size data is the fundamental problem of gene expression data analysis. Gene expression data are distinguished from other types of machine learning data because the data observe the system of a cell, which is one of the most complex systems in the universe. A cell includes huge amounts of information in the DNA sequence, and the agents in a cell, such as genes (about 20,000), their products, and chemical compounds, work together in a very organic way to perform the biological functions. Figure 1.2 shows how genes interact with others in very organic ways to perform programmed biological functions. Although many interactions between genes have been revealed, we have to uncover the unrecognized interactions between genes and understand how they are regulated differently according to external conditions. According to the empirical rule of machine learning, modeling and understanding a complex system requires a massive amount of data. However, we have not yet been able to obtain that

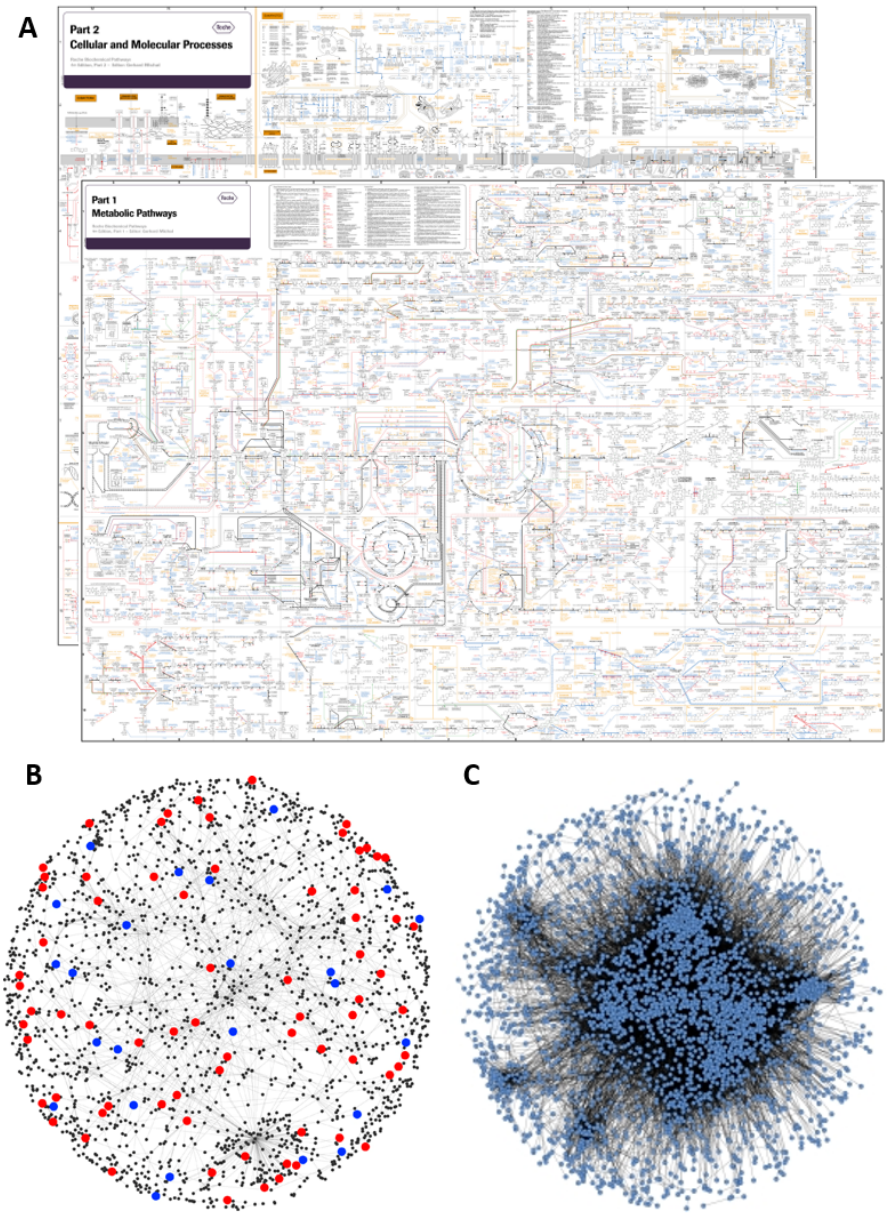

Figure 1.2. Complex system of a cell. (A) Biological pathways related to cellular and molecular process (http://biochemical-pathways.com/#/map/2), and metabolism (http://biochemical-pathways.com/#/map/1). (B) Mouse gene interaction network [79]. (C) Human coexpression network in B cells including 44,872 connections among 3056 genes [98]. These figures show how genes interact with others in very organic ways to perform programmed biological functions.

amount of data. The measurement of gene expression data is often constrained by ethical issues or uncontrollable events (e.g., death of samples). Most importantly, gene expression data are costly to produce.

On the other hand, the domain knowledge of genes has been accumulated over a long period and validated by very rigorous biological experiments conducted by professional biologists. The existence of domain knowledge provides gene expression data an advantage over other types of machine learning data. Introducing knowledge of genes can increase the reliability of cellular system modeling from gene expression data.

Figure 1.3 illustrates an example and the challenges of using gene expression data to investigate a cell's response to cold stress. Gene expression data can be easily understood in the format of machine learning data. Genes can be considered features, and the targeted trait of samples can be viewed as classes (e.g., stress-resistant vs. stress-sensitive). The challenges in gene expression analysis can be listed in detail as follows.

- **High-dimension and small-sample-size data**: Gene expression data in a single biological experiment generally have a large number of features ($>$ 20,000) and a much smaller number of samples (typically a few to tens).
- **High dependency between features**: The system of a cell consists of complicated interactions between genes, which leads to a computational problem known as high dependency between features.
- **Introducing domain knowledge and data**: Extensive biological knowledge and data about genes are available. Introducing them can improve the accuracy of the analytical model.

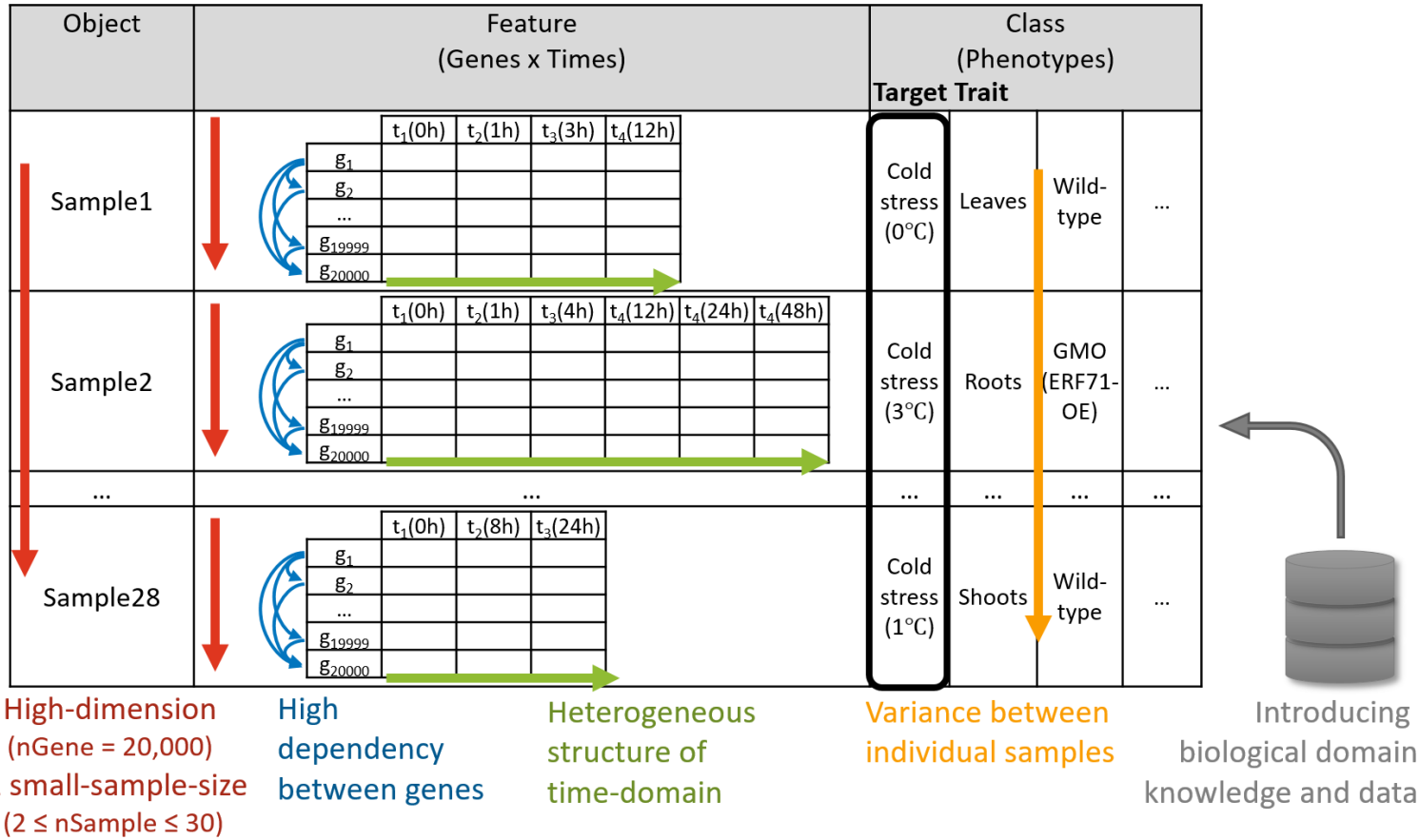

Figure 1.3. Challenges of using gene expression data. We can easily understand gene expression data in the format of machine learning data: genes as features and target condition of samples as classes. The main problem of analysis of gene expression data is modeling a highly complex system (i.e., a cell' state) with small-sample-size data (below few tens). It raises some subchallenges .

- **Heterogeneous structure of time-domain**: Gene expression data are often measured over time to trace the progress of an event (e.g., drought stress). However, the time points can be differentially selected in a dataset.

- **Variance between individual samples**: The samples in a dataset have different backgrounds (e.g., tissue, age, genetic background). Thus, the individual variance of the sample causes different responses from sample to sample, even if samples are exposed to the same stress.

- **Unknown distribution of data objects**: Knowing the distribution of data objects helps to establish a model. However, we do not know the distribution of genes.

- **Evaluation of results**: If we have external data with label information, we can evaluate the predicted results by comparing the predictions with labels. However, we generally do not have such labeled data for gene expression data analysis.

1.2 Dissertation Goal

The goal of this dissertation is to solve the recent problems in gene expression analysis. This dissertation addresses the challenges based on integrated analysis and clustering analysis, in detail, by (1) incorporating the public data followed by clustering for the high-dimension small-sample-size problem, (2) integrating heterogeneous time-series data with ordering of response time, and (3) combining advanced clustering techniques to develop a clustering algorithm that works on arbitrarily distributed data.

The scope of the dissertation is limited to analyzing gene expression data to investigate plant cellular responses under environmental stresses, such as heat, drought, or salt stress, because there are too many uses of gene expression data for this dissertation to cover. Understanding plant responses to environmental stress is becoming an increasingly important issue, as the climate has changed dramatically in recent years.

1.3 Dissertation Structure

This dissertation consists of six chapters. The first chapter presents an overview of this dissertation: data, challenges, and goals. The second chapter provides the background on clustering analysis and gene expression data analysis for a biological application that identifies stress-responsive genes. The last chapter provides a conclusion. The three intermediate chapters address specific studies as follows.

Chapter 3 describes clustering-based dimension reduction analysis for high-dimension and small-sample-size gene expression data. The data are extremely small samples (N=2) of two classes (drought sensitive and drought resistant). A novel computational framework called RiceTFnetwork [6] was developed to improve small-sample-size gene expression data analysis by incorporating large-scale public-domain gene expression data (N=1893) through a network-based clustering technique. It produced an explanation for the drought resistance mechanism.

Chapter 4 proposes a novel method to analyze multiple heterogeneous gene expression datasets. The input data are over 20 multiple-sample gene expression datasets where the structures of time-domain and phenotype-domain data are heterogeneous across multiple samples. The key element of integration is response time. A new method called HTRgene [7] was developed to determine the response time of genes based on clustering analysis of genes. The method successfully produced

stress-responsive genes.

Chapter 5 presents a novel hierarchical clustering method called IDEA [8]. We do not yet know the distribution of genes in gene expression data. Hierarchical clustering is a widely used clustering method for gene expression data, but it does not always work on arbitrarily distributed data. This chapter proposes an improved version of the hierarchical clustering method that works on arbitrarily distributed data by combining effective recent clustering techniques, such as network representation, phase shifting, and cost-optimization-based tree integration.

Chapter 2

Background

The previous chapter introduced technical challenges concerning the analysis of gene expression data and introduced clustering as one of the core technologies to address these issues. In this chapter, a survey of state-of-the-art clustering methods is presented to lay the groundwork for the main work in the following three chapters. This chapter also continues to explain the biological systems involved in measuring gene expression data.

2.1 Clustering Analysis

Clustering analysis is the most representative technique for the analysis of unlabeled data in various research fields such as information retrieval and text mining [32, 35, 125], geographic information systems (GIS) or astronomical data [38, 117, 138], sequence data analysis [21], web applications [29, 42, 53], and DNA analysis in computational biology [15]. The main idea of clustering analysis is to divide data into subgroups of similar instances. Clustering makes underlying characteristics of data recognizable.

The objective of clustering is to maximize intra-cluster similarity and minimize inter-cluster similarity [52].

- **Intra-cluster similarity**: similarity between objects in the same cluster
- **Inter-cluster similarity**: similarity between objects in the different clusters

With these two criteria, clustering analysis can be represented as an optimization problem (Definition 1).

Definition 1. *Clustering analysis* is a process to divide data objects X into k clusters for the given parameters with the following objective.

Input: X, a set of N data objects such as $X = \{x_1, \ldots, x_N\}$

Parameters: δ, a clossness measure defined on a pair of data objects (e.g. Euclidean distance) such as $\delta : X \times X \mapsto \Re$

k, the number of clusters

Alg, a clustering algorithm

Output: C, a membership function such as $C : X \mapsto \{1, \ldots, k\}$

The objective of clustering analysis:

$$\begin{cases} \text{maximization of intra-cluster similarity} \\ \text{minimization of inter-cluster similarity} \end{cases}$$

Figure 2.1 shows the process of clustering analysis. For given input X and parameters δ, k, Alg, the clustering algorithm Alg measures closeness between data objects X using δ. Then, it selects the edges with high closeness to construct a network. Network construction is an optional process. Then it groups divides data objects X in the network into k clusters for the objective of clustering analysis.

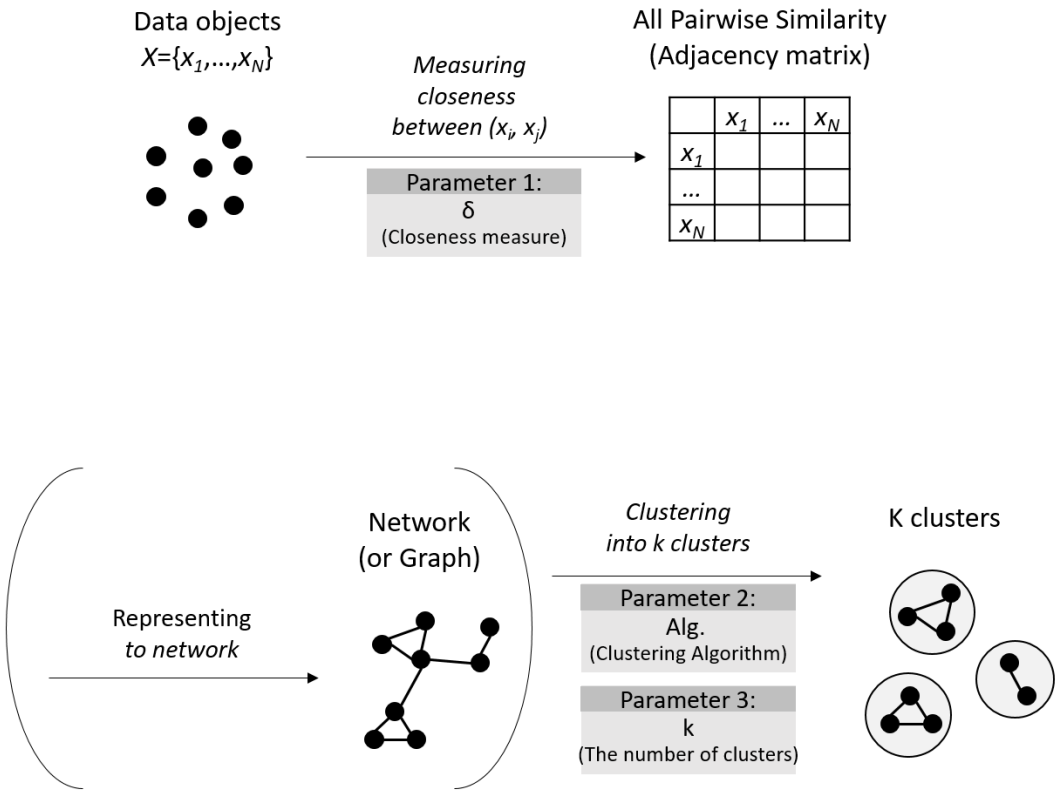

Figure 2.1. Process of clustering analysis. For given input X and three parameters δ, k, Alg, the clustering algorithm Alg measures closeness between data objects X using δ. Then, it selects the edges with high closeness to construct a network. Network construction is an optional process. Then it groups divides data objects X in the network into k clusters for the objective of clustering analysis.

2.2 Essential Elements of Clustering Analysis

There are three essential elements of clustering analysis: the closeness measure, number of clusters, and types of clustering algorithms.

2.2.1 Closeness Measure

There are two types of closeness measures: similarity and dissimilarity (distance). Distance has the same meaning as dissimilarity when used in general, but when used strictly (especially when specified as the "distance metric"), it is mathematically defined to satisfy the following conditions for $x, y, z \in X$ [25].

- $d(x,y) \geq 0$ (non-negativity)
- $d(x,y) = d(y,x)$ (symmetry)
- $d(x,y) \leq d(x,z) + d(z,y)$ (the triangle inequality)
- $d(x,y) = 0$ if and only if $x = y$ (identity of indiscernibles)

It is generally known that similarity and distance are preferred when dealing with qualitative and quantitative data features, respectively [137]. From this point of view, Xu and Tian [136] summarized widely used distances and similarities, as shown in Table 2.1.

The perspectives of the closeness measure are very different from each other. For example, the Euclidean distance measures the magnitude difference between two vectors. In contrast, the cosine distance measures the angular difference between two vectors. Thus, choosing an appropriate closeness measure is an important factor that results in meaningful or poor clustering results [60].

One of the issues for the closeness measure is the conversion between similarity and distance. When an algorithm is fixed for a similarity or distance as input, the conversion will be required (e.g., graph-based clustering requires similarity as input).

Table 2.1. Distance and similarity measures [136].

Name	Formula	Explanation
Minkowski distance	$\left(\sum_{l=1}^{d} \lvert x_{il} - x_{jl} \rvert^{n}\right)^{1/n}$	A set of definitions for distance: 1. City-block distance when $n = 1$ 2. Euclidean distance when $n = 2$ 3. Chebyshev distance when $n \to \inf$
Standardized Euclidean distance	$\left(\sum_{l=1}^{d} \lvert \frac{x_{il} - x_{jl}}{s_l} \rvert^{2}\right)^{1/2}$	1. S stands for the standard deviation 2. A weighted Euclidean distance based on the deviation
Cosine distance	$1 - \cos\alpha = \frac{x_i^T x_j}{\lVert x_i \rVert \lVert x_j \rVert}$	1. Stay the same in face of the rotation change of data 2. The most commonly used distance in document area
Pearson correlation distance	$1 - \frac{Cov(x_i, x_j)}{\sqrt{D(x_i)}\sqrt{D(x_j)}}$	1. *Cov* stands for the covariance and *D* stands for the variance 2. Measure the distance based on linear correlation
Mahalanobis distance	$\sqrt{(x_i - x_j)^T S^{-1} (x_i - x_j)}$	1. S is the covariance matrix inside the cluster 2. With high computation complexity
Hamming distance	The minimum number of substitutions needed to change one data point into the other	Especially for the data of string
Jaccard similarity	$J(A,B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$	1. Measure the similarity of two sets 2. $\lvert X \rvert$ stands for the number of elements of set X 3. Jaccard distance $= 1 -$ Jaccard similarity
For data of mixed type	Map the feature into (0, 1) Transform the feature into dichotomous one	[39, 49]

For the measure that is bounded $\leq M$ (M is the maximum value for the measure), e.g., cosine or Pearson correlation, "$f(x) = M - x$" is simply used for the conversion. However, for an unbounded measure, e.g., Euclidean distance, some decreasing kernel functions are used as follows [76].

- Cauchy function: $f(x) = \frac{1}{1+x}$
- Generalized Gaussian function: $f(x) = \exp(-x)$
- Fermi-Dirac function: $f(x) = \frac{1}{1+\exp(x)}$

2.2.2 Number of Clusters

Some clustering analysis methods pre-set the number of clusters to a fixed number and then generate clusters. Many methods have been developed to determine a proper number of clusters [28]. However, determining the optimal number of clusters is still an open problem. There are two ways to determine the number of clusters: user estimation and systemic estimation. The former leaves the responsibility to the user to determine the number of clusters. For example, k-means clustering requires the number of clusters k as input to produce k clusters as output. In the latter case, the number of clusters is deduced during the process, or the number of clusters is automatically determined by the condition that the execution is terminated. For example, in density-based clustering, the number of clusters is determined without input from the user. It merges the clusters above the density threshold and ends when they can no longer be merged ; in this way, the number of clusters is automatically determined.

2.2.3 Clustering Algorithm

The development of clustering algorithms is a historical research topic in data analysis. It is difficult to provide a clear classification of the clustering method, since various ideas have been put forth and combined to develop clustering algorithms [52].

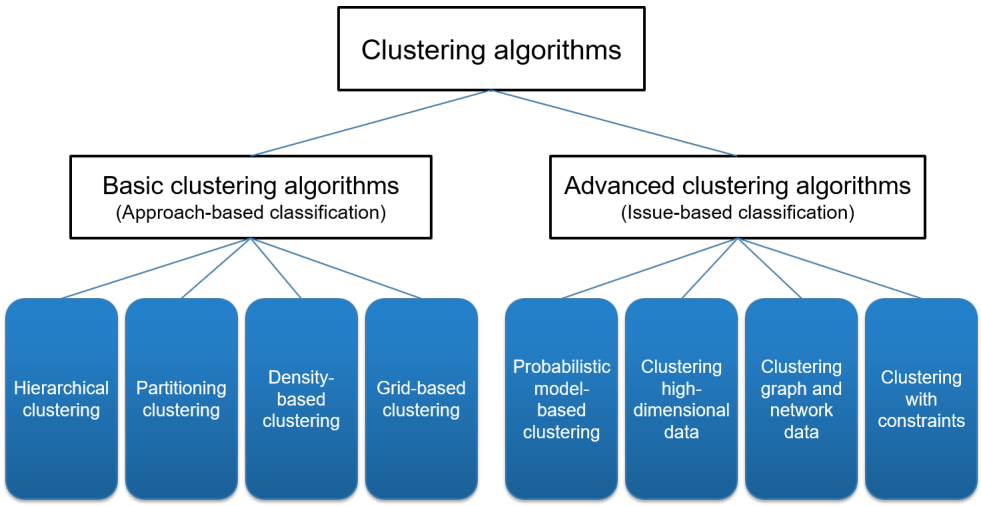

Figure 2.2. Classification of clustering algorithms.

Therefore, many reviews and books [17, 44, 50, 52, 58, 59, 119, 136, 137] that categorized clustering algorithms classified them according to their own criteria. However, they proposed two common criteria: basic or advanced clustering algorithms.

The categories of clustering algorithms are summarized in Figure 2.2

- **Basic clustering algorithms** are classified into four categories in terms of the "approach" used to solve the clustering problem: hierarchical, partitioning, density-based, grid-based methods. In most cases, the basic clustering algorithms assume the data points are defined in Euclidean space and the metric is Euclidean distance.
- **Advanced clustering algorithms** are classified according to "issues": probabilistic model-based clustering, clustering high-dimensional data, clustering graph and network data, clustering with constraints.

Because the criteria of subclass classification are different for basic and advanced clustering, one clustering algorithm can be included in both basic and advanced clustering at the same time.

Hierarchical clustering involves joining (or dividing) clusters successively. Bottom-up clusterings, called *agglomerate* clusterings, initially define clusters as individual data elements and successively merge the closest pair of clusters until only one cluster remains. On the other hand, top-down clusterings, called *divisive* clusterings, initially define one cluster including all data elements and successively split a cluster into two clusters until all clusters are partitioned into individual elements. It produces hierarchically structured output that is visualized as a tree structure called a hierarchical clustering tree or dendrogram. A posterior analysis called "cut tree" generates the k (or other possible numbers) flat clusters dynamically by cutting the internal nodes of the hierarchical clustering tree.

Partitioning clustering constructs k partitions of the objects, where each partition represents a cluster. The most representative partitioning clustering methods are k-means and k-medoids. K-means assign the objects to k clusters, identify the centroids (centers) of each cluster, re-assign the objects to the clusters whose centers are the closest to the object, and then repeat this process until convergence. K-medoids follows the same process as k-means, but it uses centers as medoids (they must be one of the data objects) rather than centroids (they can be new points out of the data objects).

Density-based clustering aims to discover clusters of arbitrary shape by considering the density of data objects. Partitioning clustering considers only the distance between objects, so it can effectively find spherical-shaped clusters but not arbitrarily shaped clusters. Density-based clustering selects densely gathered objects as an initial cluster, merges neighbor objects when the neighborhood of a given radius contains a minimum number of points, and then stops when there are no neighbor objects to be merged. Because it stops systemically, it generally does not need the number of clusters as input. Moreover, it produces unmerged objects (i.e., outliers), so it can be

used to filter out noise or outliers.

Grid-based clustering divides the object space into a finite number of grid-structured cells and then performs the entire clustering operation on the grid structure (i.e., on the quantized space). Although the time complexity clustering algorithms typically depend on the number of data objects, that of grid-based clustering depends only on the number of cells in each dimension in the quantized space, so grid-based clustering has the advantages of low time complexity and high scalability. STRING [132] and CLIQUE [3] are the representative algorithms of this kind of clustering. STRING divides the data space into many rectangular units that can be divided into the lower-level space hierarchically and recursively and uses statistical parameters of the data object in the structure spaces for clustering. CLIQUE is a grid-based and density-based clustering algorithm.

Probabilistic model-based clustering defines the memberships where a data object belongs to a cluster by a value within the continuous interval $[0,1]$. For example, Gaussian mixture clustering assumes that the data originated from k Gaussian distributions and defines the membership of each object to a Gaussian distribution as the probability that it belongs to the Gaussian distribution. The expectation maximization (EM) algorithm is used to estimate the parameters of the Gaussian distribution [80]. "Fuzzy" clustering is another name for probabilistic clustering. Fuzzy c-means (FCM) [18] is a fuzzy version of the k-means clustering algorithm. It probabilistically defines the membership as the weight of the distance of each center, while the k-means defines the membership in a black-and-white way by assigning the object to the cluster with the closest center. FCS [34], and MM [139] are other versions of fuzzy clustering algorithms.

High-dimensional data clustering addresses the issues that data of high dimensionality raises noise to conventional distance measures. Dimensionality reduction

approaches project data objects onto a much lower -dimensional space and conduct clustering analysis in the space. The feature selection approach projects data objects onto subdimensions by selecting a set of relevant features (variables, predictors), and the feature extraction approach projects data objects onto new dimensions by combining some dimensions with weighting from the original data [1]. The subspace clustering approach [103] is another way of performing high-dimensional data clustering. It focuses on subspaces of the given high-dimensional data space and searches for a cluster in the subspaces. CLIQUE [3] explores the clusters in lower subspaces and extends the search in higher subspaces. PROCLUS [2] searches for clusters by combining the k-medoid and top-down-style subspace approaches. Biclustering is a kind of subspace clustering widely used with biological data [81]. It takes an $N \times M$ matrix as input and tries to find a submatrix where the rows or columns share a specific similar pattern.

Graph and network data clustering divides a graph into several subgraphs, which are also called subnetworks, modules, or communities, where the connectivity within the subgraphs is high. Since graph or network representation is a powerful technique to model relationships of data objects, network data are increasingly popular in applications such as social networks, the World Wide Web, genetic regulatory networks, and citation networks. Community detection is an alternative name for network clustering. The `igraph` library [31] provides implements of recent community detection algorithms: optimal modularity [48], edge-betweenness [47], leading eigenvector [99], fast-greedy [26], multi-level [19], Walktrap [105], label propagation [107], and Infomap [111].

Constraint clustering gives some constraints in the process of clustering analysis. Background knowledge or spatial distribution of the objects can be applied to clustering analysis through these constraints. Constraints on instance determine how

a pair of instances has to be grouped, such as must-link or cannot-link constraints. Constraints on clusters regulate the attributes of clusters, such as the minimum number of objects in a cluster, the maximum diameter of a cluster, or the shape of a cluster (e.g., a convex). Constraints on similarity measurement specify a requirement that the similarity calculation must obey; for example, the distance of two points should be defined as the distance traveled through a specific point.

2.3 Biological System and Gene Expression Data

Biological systems are multi-scale and module-structured systems. The group of lower-scale systems composes the high-scale systems as cells compose tissues. One of the criteria of categorization of biology is based on the scale of interest: biochemistry, molecular biology, cell biology, physiology, ecology.

Molecular biology is a branch of biology that studies biomolecules and their reactions in a cell [9]. A cell contains genetic materials, such as deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and protein. The explanation of the flow of genetic information from DNA, to RNA, to protein is called "*central dogma*", which was first stated by Francis Crick in 1958 [30]. With the central dogma, a cell operates all life activities, such as growth, differentiation, homeostasis, response to environments, and so forth. Figure 2.3 illustrates the process of the central dogma.

Until the age of microscopy, humankind could not access genetic information. However, in 1977, when Sanger proposed a method for profiling DNA sequences [118], humanity became able to decode the genetic information of genetic molecules. Nevertheless, Sanger sequencing was limited by its low throughput, allowing the measurement of only one gene per experiment. The later development of highly parallel technologies, such as microarrays [120] and RNA-Seq [93] made it possible to inves-

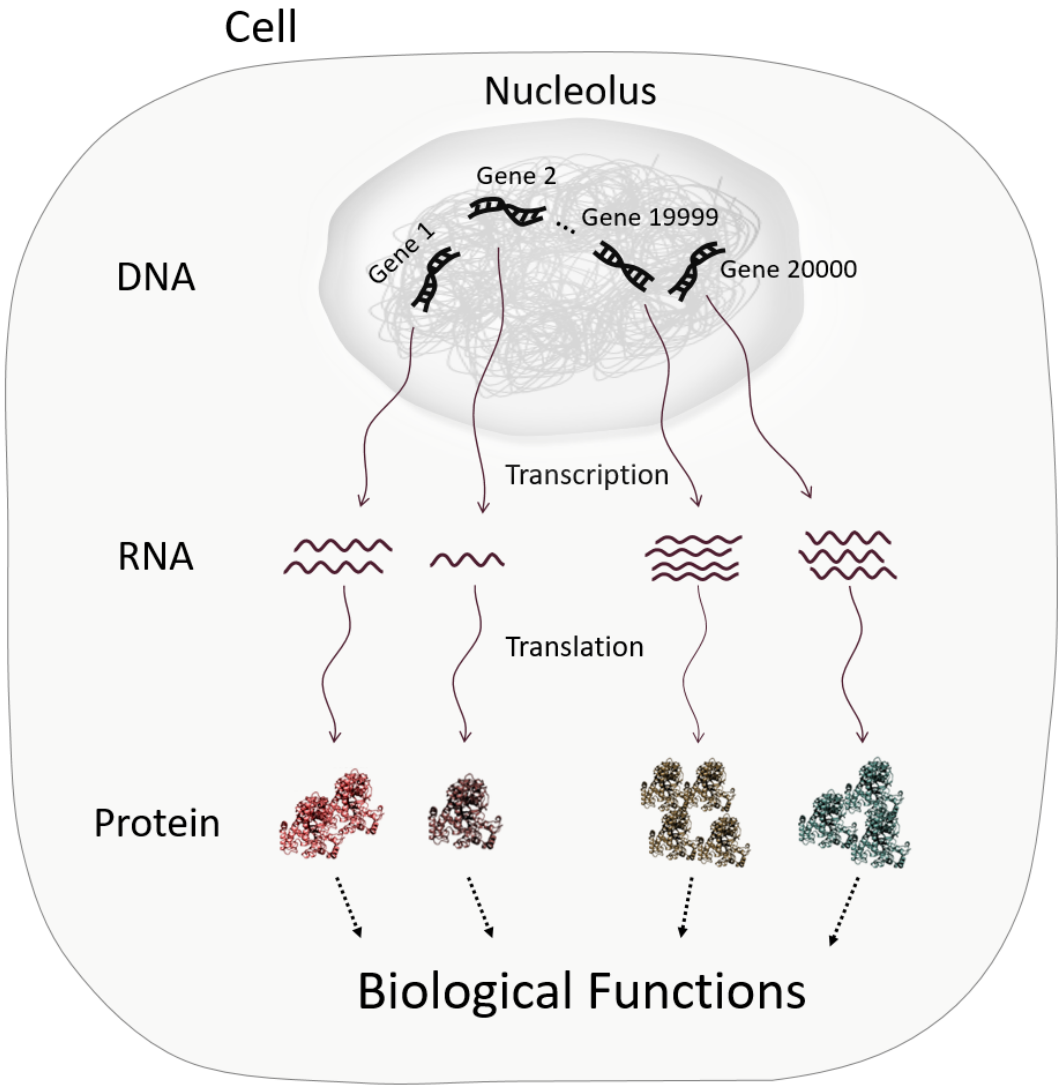

Figure 2.3. Central dogma. The explanation of the flow of genetic information from DNA, to RNA, to protein is called "*central dogma*". DNA is a source of genetic information that produces gene products : RNA and protein. RNA (messenger RNA) is a mediator between DNA and protein that transfers the genetic information of DNA to protein. Protein is the product of DNA, and it participates in all life activities of the cell, such as growth, differentiation, homeostasis, response to stimulation, and apoptosis.

tigate all genes in a single experiment. The development of genetic molecule measurement technology has highly facilitated the study of genetic molecules in cells and promoted the discovery of new biological knowledge.

Currently, two types of genetic data are dominantly generated: gene sequence data and gene expression data. Gene sequence data investigate the character information of genes, such as DNA and protein sequences consisting of four nucleotide characters (A, C, G, T) and 20 amino acid characters, respectively. Gene sequences contain deterministic information. They are determined and fixed at birth and do not actively change during life. Thus, most gene sequence studies do not measure the same person or organism multiple times. They mainly collect many positive and negative samples for a targeted trait (e.g., Huntington's disease) and conduct association analysis between the genomic sequence variants and the trait. In this type of analysis, gathering more samples increases the statistical accuracy of the prediction. The applications of gene sequence studies include population studies, genealogy analyses, and genome comparisons.

On the other hand, gene expression data contain dynamic information. Gene expression data quantify the state of a cell at a particular time point. Because the state of a cell changes depending on the external environment, gene expression data also change. Even in the same cell, gene expression varies over time. Applications of gene expression data include tracking the evolution of certain intracellular events, such as cell differentiation, development, or disease progression, and responses to drugs or stress.

There are two methodologies to measure gene expression levels. Proteome data are used to investigate the amount of protein, and transcriptome data are used to investigate the amount of messenger RNA. Whereas proteins are the final products of genes, proteins have a 3D structure and are difficult to measure. Thus, current

techniques investigate gene expression levels by measuring the amount of messenger RNA, which is the precursor of protein, instead of the amount of protein.

Figure 2.4 illustrates a typical biological research process that uses genetic data. Scientists design biological research projects to investigate mechanisms underlying specific phenotypes (e.g., cancer or obesity in medical applications or stress resistance in crop engineering) by preparing biological samples in positive vs. negative groups (e.g., cancer vs. control groups). For investigations at the molecular level, bulks of cells are extracted, and libraries for the cells are prepared. Then, genetic data are measured using microarray and sequencing technologies, and the data are analyzed using various computational tools. Based on the analysis results, scientists design follow-up biological experiments, which may result in new crops, vaccines, or drugs.

2.4 Identification of Stress-Responsive Genes Using Gene Expression Data

One important research topic that can be investigated by using gene expression data is the way in which plants respond to environmental stress. Characterizing environmental-stress-induced responses has historically been a major research topic in agricultural studies related to the productivity of crops [10, 62, 129]. Moreover, environmental stress has become a more serious issue in plant science due to recent dramatic climate changes. The Intergovernmental Panel on Climate Change (IPCC) estimates that the global mean temperature will increase somewhere between 1°C and 4°C until 2050 and that the drought stress increased by the climate change will significantly reduce crop productivity in Africa and Asia [14].

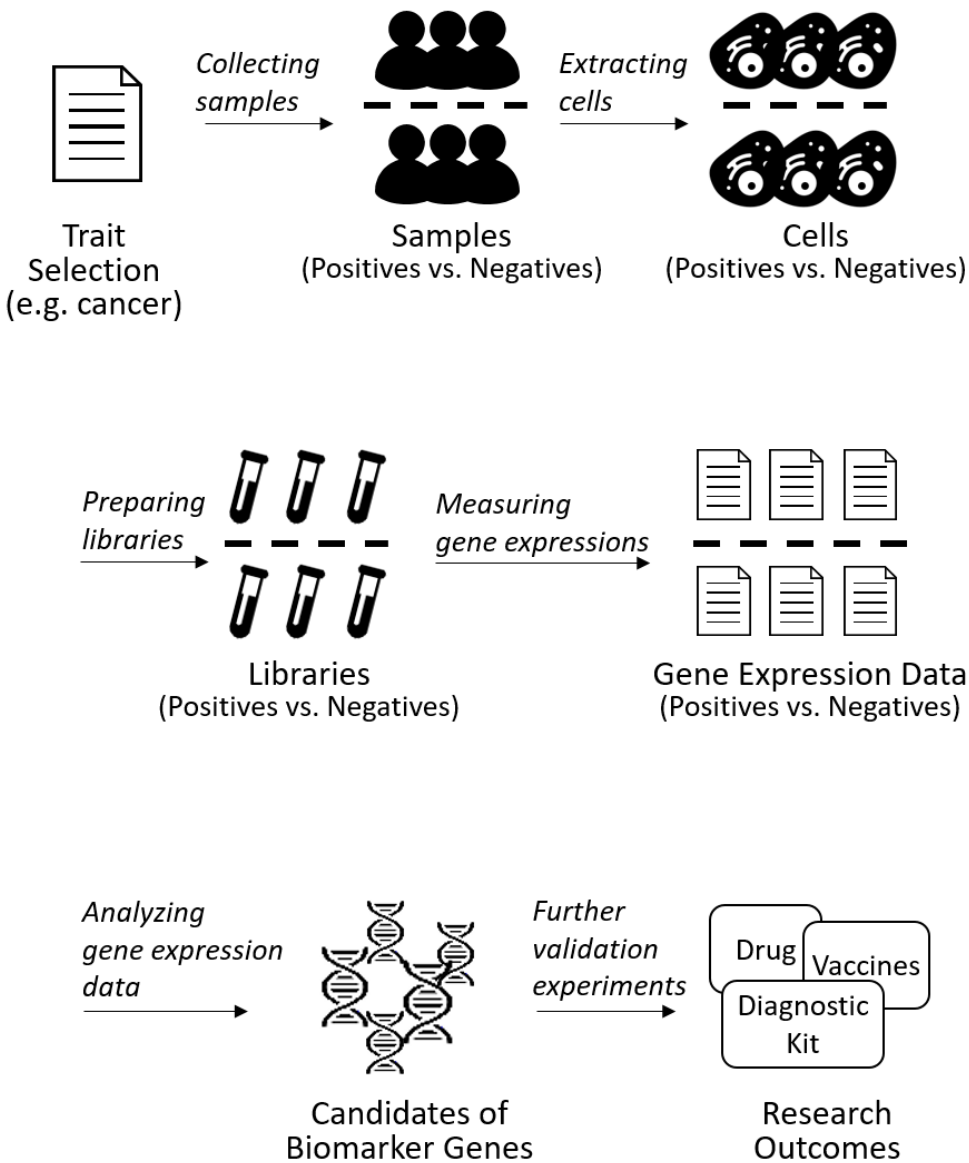

Figure 2.4. Process of biological research using gene expression data.

Figure 2.5 shows a typical signaling pathway for stimulation. When external stimuli, such as stress, hormone, drug, and intercellular signaling molecules, reach a cell, receptors on the cell membrane accept it and activate downstream proteins to deliver the signal. The activated protein then signals the downstream protein sequentially to the transcription factor (**TF**) protein. TF has the ability to enter the nucleus and bind to a specific DNA sequence called motif. When TF binds to the promoter or enhancer region of the target gene (**TG**), TF recruits RNA polymerase and promotes production (i.e., transcription) of messenger RNA of the TG. The resulting messenger RNA moves into the cytoplasm and the ribosome performs synthesis (i.e., translation) of the protein from the messenger RNA. The resulting protein then performs a biological function in response to the stimulus. Thus, understanding the cellular response to stress is the discovery of a stress signaling pathway in which the stress signaling genes deliver the stress signal from the receptor to the final protein product.

The use of gene expression data in plant stress studies has led to the realization of a new paradigm of research called "data-driven" research paradigm. The paradigm does not require sophisticated hypotheses derived from high-level biological knowledge to start a biological research project. After only selecting the stress of interest, for example, cold stress, experimental biologists collect samples before and after the stress and measure the gene expression data of the samples. Then, computational biologists, called bioinformaticians, analyze the gene expression data to produce the candidate stress-responsive genes. Then, they together select the interesting genes from the list and make a genetically modified (GM) crop where the expression level of the selected gene is artificially activated or depressed. This approach has been highly successful in the development of stress-resistant plants [54, 101, 106, 122, 131, 135, 143, 144]. In these data-driven studies, the analysis of gene expression data, which is the main topic of this doctoral study, is critical to the success of the

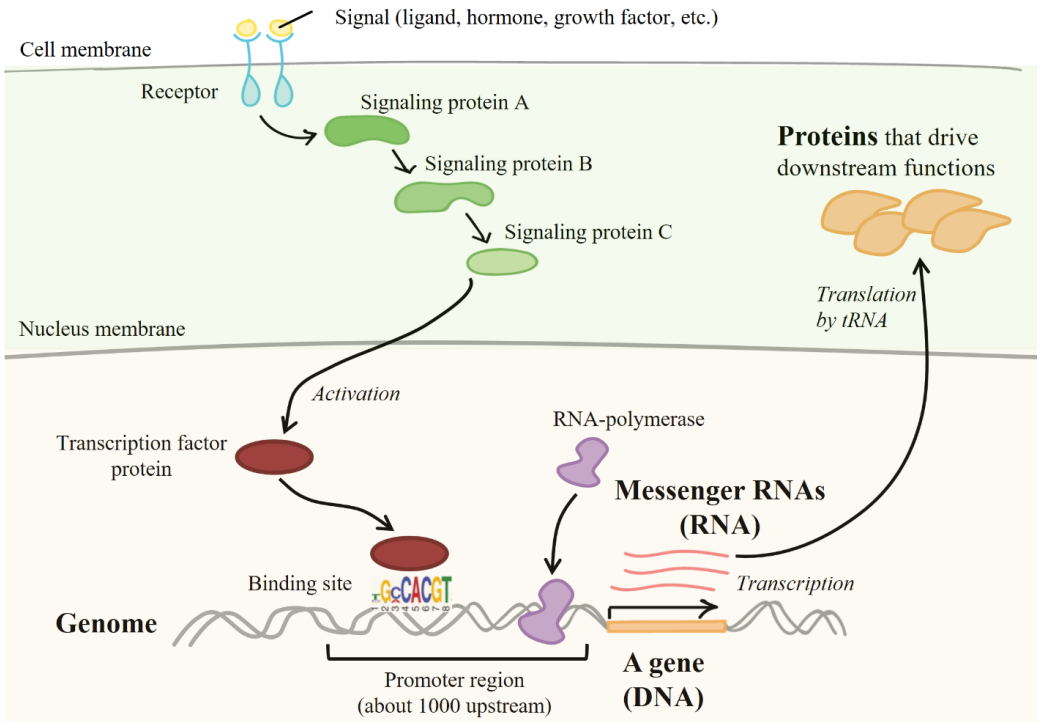

Figure 2.5. Typical signaling pathway in response to stress. When stress reaches a cell, receptors on the cell membrane accept it and activate downstream proteins to deliver the signal. The activated protein then signals the downstream protein sequentially to the TF protein. TF enters the nucleus, binds to the promoter or enhancer region of the TG, and recruits RNA polymerase to promote production (i.e., transcription) of messenger RNA of the TG. The resulting messenger RNA moves into the cytoplasm and the ribosome performs synthesis (i.e., translation) of the protein from the messenger RNA. The resulting protein performs a biological function in response to the stress.

project.

To find the stress signaling genes in gene expression data, gene expression levels are compared between sample groups before and after the stress for each gene. This produces the genes whose expression levels change significantly between the two groups called "*differentially expressed genes* (DEGs)". To analyze the data, pairwise comparison tools based on statistical models are used, such as limma [109], edgeR [110], and DESeq [11]. However, because the cost of producing the data is high, botanists cannot extensively investigate gene expression levels for many samples.

Subsequent technological advances have lowered the cost of measuring data and extended gene expression data in the time dimension. Gene expression data have been measured in plants at multiple time points after stress to create time-series data. Moreover, time-series analysis tools have been developed, such as maSigPro [100], Imms [126], splineTC [88], and ImpulseDE [116]. Gaussian process model also can be an effective technique to analyze time-series data [67]. However, since the cost of generating gene expression data is still high, those data include not only a small number of time points (usually under five) but also a small number of samples (usually under five).

Now, to analyze gene expression data for a large number of samples, field experts have been collecting data on specific research interests, building databases, and providing them to researchers in the public sector. For example, the OryzaExpress database [51] provides 624 gene expression datasets from 37 experimental series with their experimental conditions. Its improved version, PlantExpress [87], was published later, and it contains 3,884 and 10,940 gene expression datasets for rice and Arabidopsis species. A more recent database, the Rice Expression Database (RED) [134], provides 284 RNA-seq gene expression datasets that were measured under various

experimental conditions in rice species.

The integrated analysis of gene expression data is required to produce robust results because the deviation of the experimental conditions between samples makes a big difference between the DEG results of the samples. When comparing response genes reported in different experiments, we can observe only a small number of common response genes even though the experiments were conducted with the same stress on the same species. For example, Kreps [73] and Matsui [85] reported 2,086 and 996 DEGs for cold stress in Arabidopsis, respectively, and only 232 DEGs, about 16% of the union of the two DEG sets, were commonly determined. To overcome the effects of individual experimental conditions, a method that integrates multiple gene expression datasets is required. However, the multiple gene expression datasets collected from databases have heterogeneous data structures, which requires appropriate and sophisticated approaches and techniques such as network propagation [5] or deep learning [66]. The issue and method will be presented in detail in Chapter 5.

Chapter 3

RiceTFnetwork: Transcriptional Network Analysis for Revealing Drought Resistance Mechanisms of AP2/ERF Transgenic Rice

Recently, understanding how plants respond to environmental stress has been becoming more important because of the rapid change of the global environment. In this study, we compared the cellular response under dehydration stress for two rice species: the wild-type (WT) species of the drought-sensitive phenotype and the GM species of the drought-resistant phenotype. The GM was named *erf71* because it was made by overexpressing the *OsERF71* TF gene. It showed enhanced survival over the WT under drought stress at the vegetative stage of growth and a 23–42% increase in total weight gain over the WT under drought stress at the reproductive stage of growth. We examined the duration of drought stress by tracing the expression level of the *Dip1* (drought-induced protein 1) gene [141] and then measured time-series gene expression data at eight time points for the two rice species: 0, 0.5, 1, 3, and 6 hours for WT and 0, 1, and 6 hours for GM. The goal of analysis is to explain the drought resistance mechanism of the GM rice.

3.1 Computational Problems

To help the reader understand the data, Figure 3.1 illustrates the gene expression data in this study in the format of machine learning data. The data objects are two rice samples (WT and GM), the features are genes, and the classes are drought resistant and drought sensitive. The goal of the analysis is to explain the drought resistance mechanism of the GM rice. Computational challenges are (1) the analysis of high-dimension ($>$ 20,000 genes) and small-sample-size (only two samples) data, (2) the analysis of the high dependency between features due to the complex interactions between genes, and (3) the comparison of the time-series data of two samples.

3.2 Methods

To address the computational problems, this section developed a novel computational framework called **RiceTFnetwork** [6] to characterize the drought resistance mechanism of a GM rice species. The method uses network representation and clustering analysis as fundamental techniques. The proposed computational framework consists of two steps, as illustrated in Figure 3.2.

- **Step 1**: Constructing a dehydration TF network utilizing gene expression data from databases and a dehydration experiment.
- **Step 2**: Instantiating phenotype-differential dehydration networks and identifying DEG modules.

The first step profiles TFs first from domain knowledge. TFs are special-type genes that function as regulators for the other TGs. TF proteins bind to the DNA promoter region of a TG and activate or suppress the expression level of the TG. Considering only edges between TFs and the other TGs reduces the size of a gene

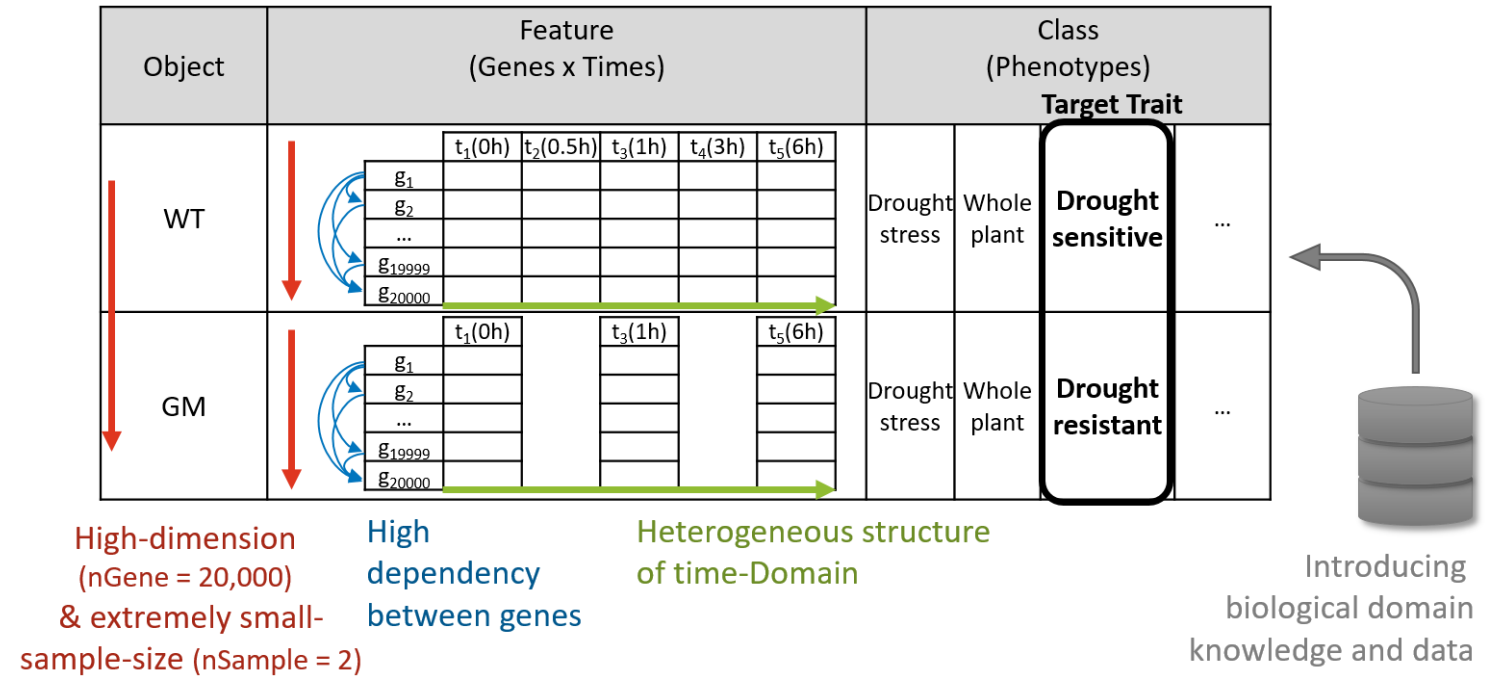

Figure 3.1. Extremely small-sample-size gene expression data. The data include two samples of rice: WT (drought sensitive) and GM (drought resistant). The goal of this research is to elucidate the drought resistance mechanism of GM rice. This chapter addresses two challenges for gene expression data analysis: high-dimension and small-sample-size data (orange arrows) and complex relationships between genes (green arrows).

network, thus reducing the problem space. After focusing on TFs, the method constructs a template TF network by inferring regulatory relationships between TFs and TGs. The network construction improves the inference power by using a large-scale public dataset including about 1,800 samples. Then, it instantiates a stress-condition-specific network by selecting edges guided by the experimental data. Integrating the large-scale public dataset with the small-sized experimental data increases the accuracy of the inferences of the network.

The second step applies clustering analysis to the stress-condition-specific network to divide it into several gene subnetworks (or clusters/modules). Clustering analysis reduces the dimension from the number of genes into the number of clusters, which addresses the issue of high-dimension and small-sized-sample data. The dimension reduction makes it possible to interpret the difference between two rice species at the gene expression level. Details on each step are described in the next sections.

3.2.1 Step 1: Constructing Dehydration TF Network Utilizing Gene Expression Data from Databases and Dehydration Experiment

To construct a reliable dehydration TF network, a template TF network was constructed by utilizing large-scale microarray data. At the beginning of this step, network construction factors, such as choice of network construction method, dataset size, and cutoff values, were thoroughly investigated. A recent study investigated many network construction methods and reported that mutual-information and correlation-based methods recovered feed-forward loops most reliably [83]. Since the goal of this study was to investigate the effect of *OsERF71* overexpression on other genes through relationships between the TFs and their TGs, which can be seen as feed-

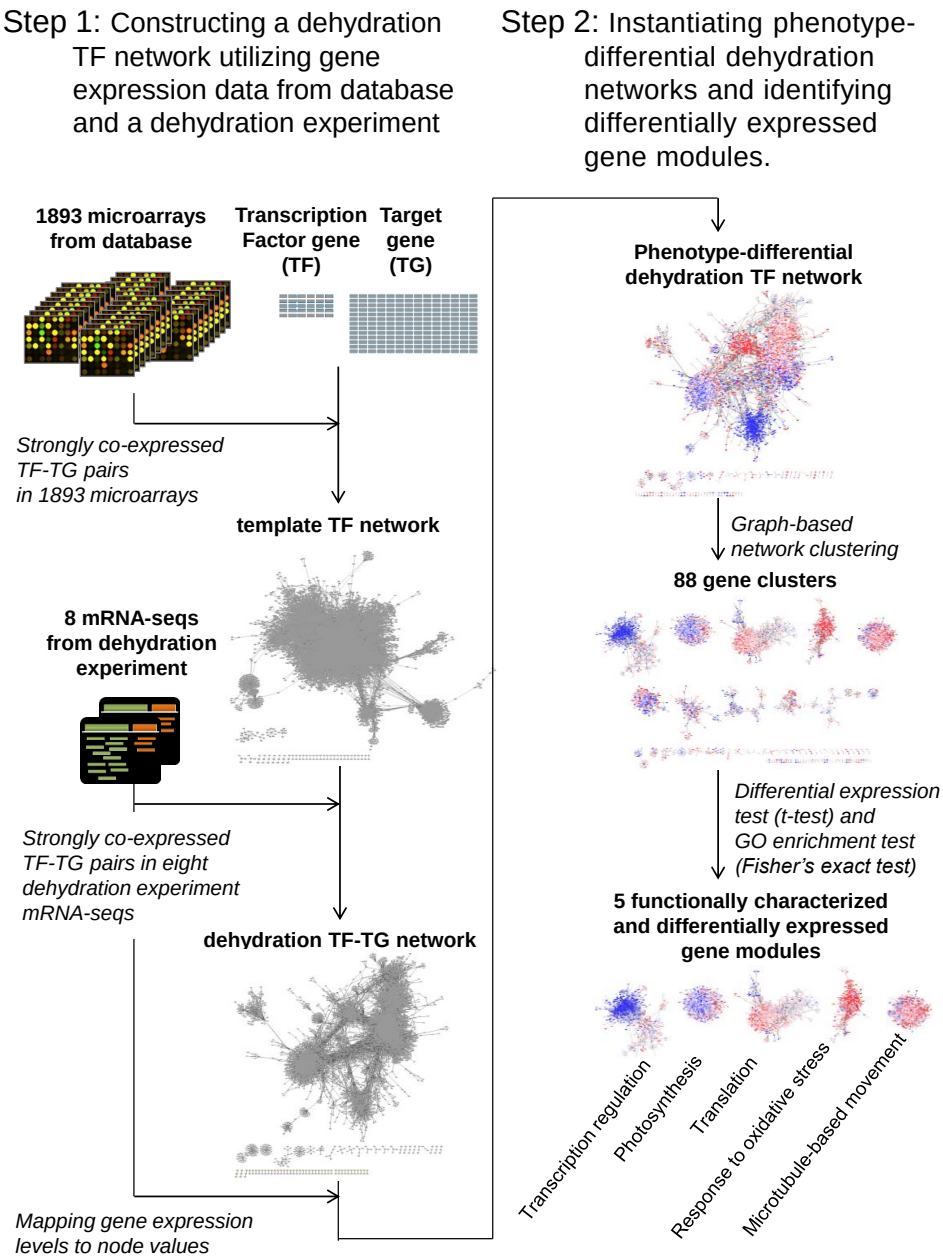

Figure 3.2. TF network analysis workflow. A template TF network was constructed by selecting strongly co-expressed TF–TG pairs in 1,893 sample public domain microarray datasets. Then, a dehydration TF network was constructed by selecting strongly co-expressed TF–TG pairs in eight dehydration experiment RNA-Seq datasets. Phenotype-differential dehydration networks were instantiated by mapping gene expression differences to node values. Then, clustering analysis was performed. Finally, DEG modules were selected by t-test, and the biological functions of gene modules were characterized by GO analysis.

forward propagation from *OsERF71* to other genes, Pearson's correlation coefficient (PCC) was used as the network construction method.

Large-scale gene expression datasets (1,893 microarray datasets) were downloaded from the OryzaExpress Gene Expression Network website (http://bioinf.mind.meiji.ac.jp/OryzaExpress/). Probe-IDs that were used for microarray experiments were converted to gene-IDs according to a previous study [89]. Since PCC is shown to converge as the sample size increases, an empirical study was performed to determine whether the dataset size was beyond the convergence threshold and was sufficient to construct a robust template TF network. By varying the number of samples, different sample-size subsets of the microarray data were produced by random sampling from the 1,893 microarray datasets. For each subset, PCCs between TFs and TGs were then computed and PCC density distributions and network topologies were investigated. The density distributions and the network topologies converged with a sample size greater than approximately 800. This observation showed that 1,893 microarray datasets were sufficient to produce a robust template TF network. Recent studies that used the PCC method for biological network construction detected modular structures of genes in Arabidopsis, rice, and maize networks [41, 82]. These studies reported that each of the modules had a specific biological function. Based on this result, the functionality score was defined as follows:

$$FunctionalityScore(G) = -\sum_{c_i \in C} \frac{|c_i|}{N} \log_{10}(p_{c_i}). \tag{3.1}$$

In the formula, G is a network and it is divided into a set of gene clusters, $C = [c_1, c_2, \ldots, c_n]$, using a graph-based clustering algorithm [19]. N is the number of genes in the network, and p_{c_i} is the p-value of the most significant gene ontology (GO) term in a GO enrichment test of the cluster c_i. The functionality score measures

how well a network is divided into functional gene modules. The functionality scores were investigated for each network constructed at different PCC cutoff values. The cutoff value of 0.67 was chosen because it was the cutoff value at which the functionality score was maximized. TF–TG pairs with strong associations ($|PCC| > 0.67$) in the 1,893 microarray datasets were then defined as edges in the template TF network. The template TF network consisted of 10,740 genes (898 TFs and 9,842 nonTFs) and 135,550 links (4,073 TF–TF links and 131,477 TF–nonTF links). A dehydration TF network was then constructed by selecting edges in the template TF network that had strong associations ($|PCC| > 0.67$) in eight dehydration experiment RNA-Seq datasets. The constructed dehydration TF network consisted of 7,319 genes (729 TFs and 6,590 nonTFs) and 50,672 links (1,375 TF–TF links and 49,297 TF–nonTF links). The topology of the network was visualized using Cytoscape [114].

3.2.2 Step 2: Instantiating Phenotype-Differential Dehydration Networks and Identifying DEG Modules

In this step, the goal was to identify DEG modules between WT and *erf71* from the dehydration TF networks by the following strategy.

- **Step 2-1**: Phenotype-differential dehydration TF networks were instantiated by mapping gene expression differences to node values of the dehydration TF network.
- **Step 2-2**: Graph-based network clustering broke down the phenotype-differential dehydration TF networks into several gene clusters according to connectivity.
- **Step 2-3**: Differential expression and GO enrichment tests were performed for each cluster.
- **Step 2-4**: DEG modules were selected and designated as “modules.”

In Step 2-1, phenotype-differential dehydration TF networks were instantiated

by mapping gene expression differences between time points for each plant (i.e., $\log_2(W1/W0)$ and $\log_2(W6/W0)$) as well as differences across plants (i.e., $\log_2(E1/A0) - \log_2(E1/W0)$) to nodes of the dehydration TF network, where "*W*" and "*E*" stand for WT and *erf71*, and "0", "1", and "6" stand for 0, 1, and 6 hours after treatment (HAT), respectively. In Step 2-2, the phenotype-differential dehydration TF networks were broken down into several gene clusters using a multi-level network clustering method [19] that groups highly connected nodes into a cluster of nodes. In Step 2-3, a paired sample t-test was performed on each cluster to determine whether each gene cluster was differentially expressed. In addition, the biological functions of each cluster were characterized by GO enrichment analysis based on Fisher's exact test. In Step 2-4, the clusters showing high-level significance ($p < 1.0e^{-9}$) in both tests were selected and designated as "modules."

3.3 Results and Discussion

3.3.1 Network and Clustering Analysis

The goal of RiceTFnetwork was to characterize the differences in gene expression response between WT and *erf71* utilizing a TF network. In a TF network, nodes are genes (TFs and nonTFs) and edges are connections between TFs and potential TGs including TFs. The TF network did not include edges between nonTF genes to focus on transcription-factor-centered regulation.

A template TF network was constructed by selecting TF–TG pairs with strong associations ($|PCCs| > 0.67$) in eight RNA-Seq datasets from the experiment and in 1,893 microarray datasets in the public domain. Differences in gene expression levels (i.e., $\log_2$-fold change)were then assigned to the genes in the network. Figures 3.3A and 3.3B show the constructed dehydration TF networks of WT and *erf71*, respec-

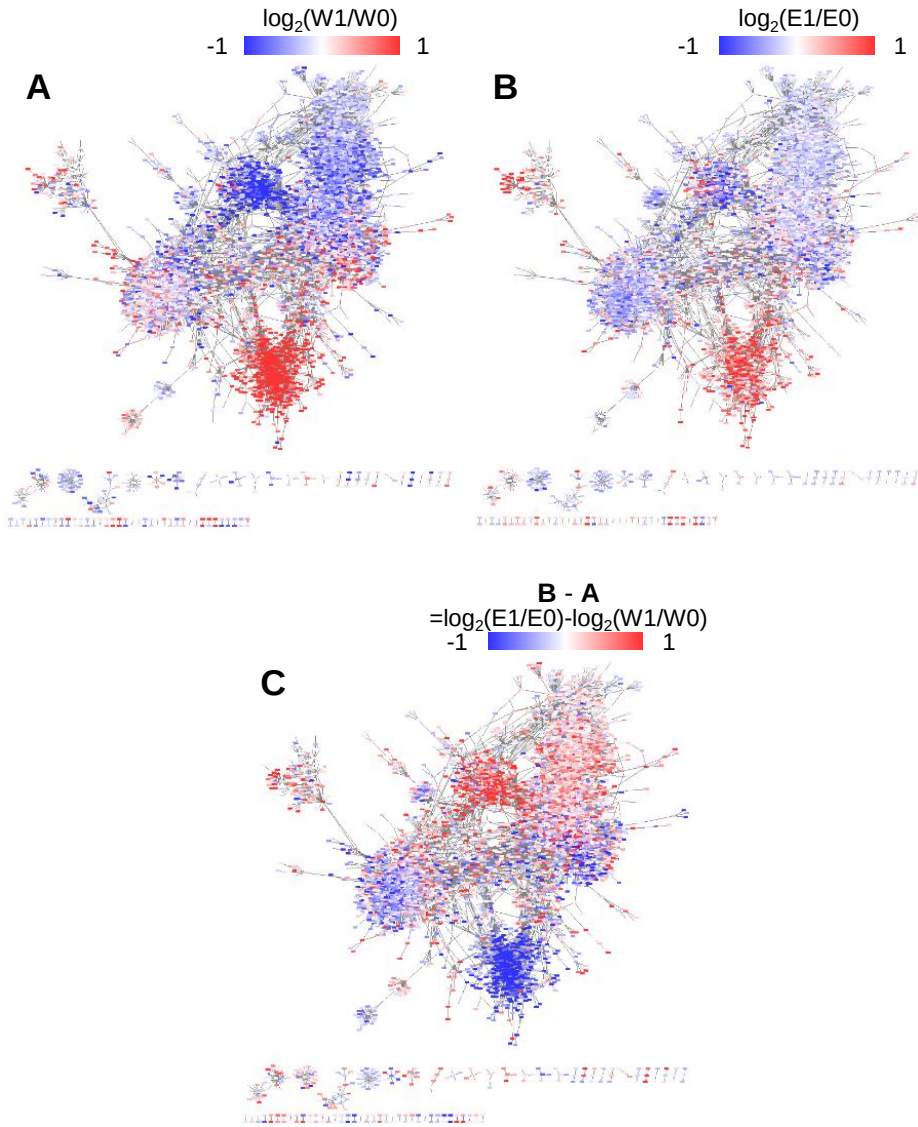

Figure 3.3. Phenotype-differential dehydration TF networks. The two time-point differential networks (A and B) were instantiated by mapping gene expression differences between time points such as $\log_2(W1/W0)$ and $\log_2(E1/E0)$, respectively. In these networks, red/blue dots denote the up/downregulation of gene expression under dehydration stress. A phenotype-differential network (C) was instantiated by mapping gene expression differences between two rice plants such as $\log_2(E1/E0) - \log_2(W1/W0)$. In this network, red/blue dots denote the relative up/downregulation of gene expression in *erf71* compared to WT.

tively, where red/blue dots denote up/downregulated genes before and after dehydration stress (i.e., 0 HAT vs. 1 HAT). Figure 3.3C shows a phenotype-differential TF network where red/blue dots denote relatively differentially regulated genes between WT and *erf71*. In other words, red dots indicate genes that are relatively upregulated (more upregulated or less downregulated) in *erf71* under dehydration stress.

The TF network was divided into 88 gene clusters by grouping highly connected genes into a cluster by a graph-based network clustering algorithm. Each cluster was characterized by differential gene expression and GO enrichment tests. Finally, five gene clusters were identified, with 713, 1,363, 537, 1,586, and 1,605 genes, showing a high level of significance ($p < 1.0e^{-9}$) in both tests. The five clusters were designated as "modules." Table 3.1 summarizes the results of the differential gene expression test and GO enrichment test for the five modules.

Figure 3.4 shows the characteristics of the five modules—the position in the TF–TG network, plots of the expression levels, and the biological functions. The mean expression levels of the five modules were changed in one direction (i.e., increased or decreased) as dehydration stress continued in both rice plants, but the degree of change was different between the two rice plants. Module 1 that included drought-response-related TFs was upregulated in both types of rice but less upregulated in *erf71*. Modules 2, 3, and 4 that were related survival-critical mechanisms, such as translation, response to oxidative stress, and cell division cycle, were downregulated in both types of rice but less downregulated in *erf71*. Module 5 that were related to photosynthesis was downregulated in both types of rice but more downregulated in *erf71*. These results suggests *erf71* diverted more energy to survival-critical mechanisms related to translation, oxidative response, and DNA replication while further suppressing energy-consuming mechanisms, such as photosynthesis. The next section presents a detailed analysis of each module.

Table 3.1. Results of differential expression test and GO enrichment test of five gene modules. The differential expression test was performed using t-test at 0-to-1 HAT between WT and *erf71*. The GO enrichment test was performed by Fisher's exact test. A p-value cutoff ($p < 1.0e^{-9}$) was used to decide differential expression and enriched GO terms.

Gene module	Differential expression at 0-to-1 HAT period between WT and erf71	P-value	Enriched GO terms	P-value
Module 1	Less up-regulated in *erf71*	1.2e-62	Regulation of transcription, DNA-dependent	3.3e-27
Module 2	Less down-regulated in *erf71*	3.0e-56	Translation	3.0e-108
			Ribosome biogenesis	2.0e-11
Module 3	Less down-regulated in *erf71*	4.0e-71	Response to oxidative stress	1.5e-12
Module 4	Less down-regulated in *erf71*	7.0e-71	Microtubule-based movement	3.4e-20
			DNA replication	2.3e-14
Module 5	More down-regulated in *erf71*	3.5e-80	Photosynthesis	3.0e-25
			Photosynthesis, light harvesting	9.4e-13

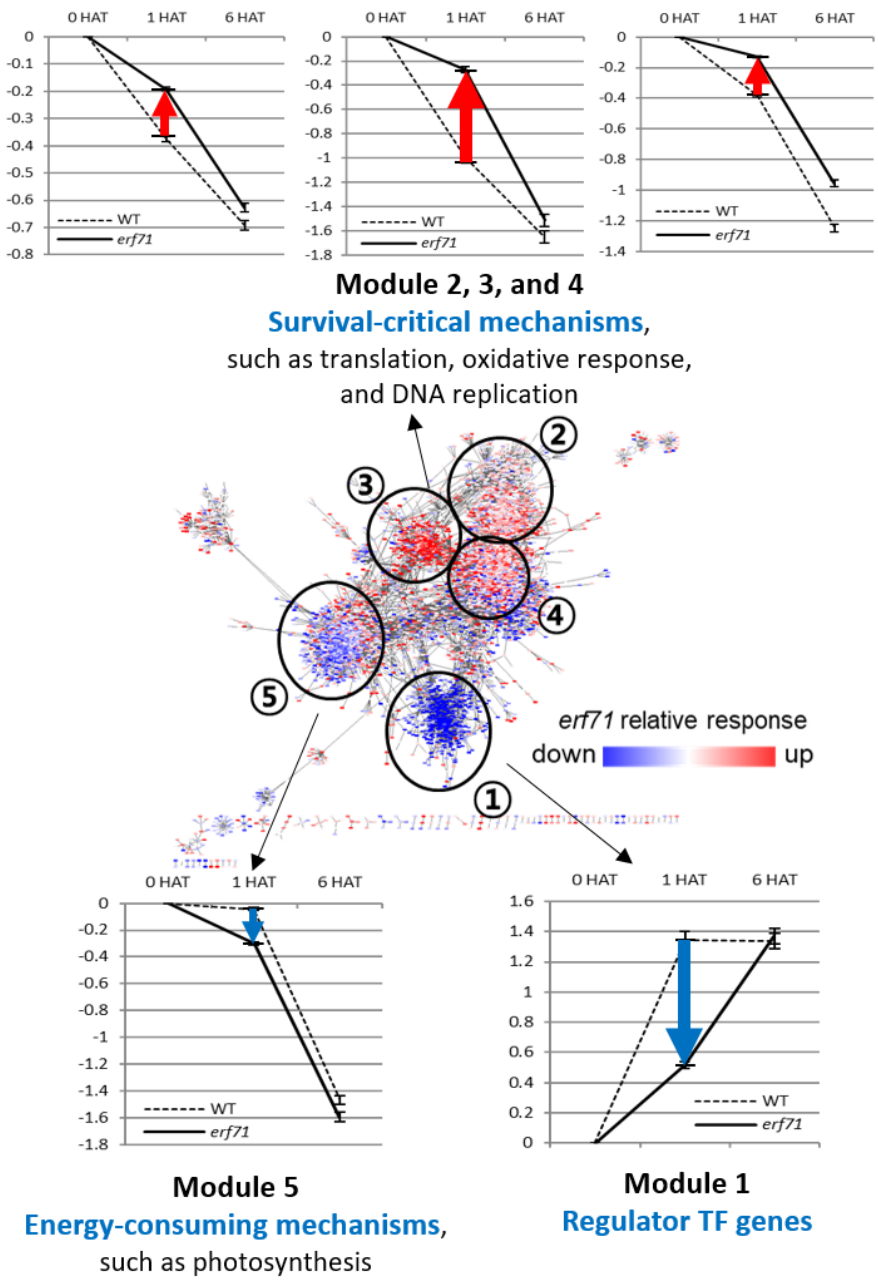

Figure 3.4. Characteristics of five gene modules. The figure in the center shows a dehydration TF network with five gene modules. The line plots around the network are gene expression levels of the five gene modules, where y-axis is the average of $\log_2$-fold changes of gene expression level with respect to the 0 HAT time point. The blue words of each module are the biological functions derived by GO enrichment tests. This figure suggests *erf71* diverted more energy to survival-critical mechanisms related to translation, oxidative response, and DNA replication while further suppressing energy-consuming mechanisms, such as photosynthesis.

3.3.2 Analysis of Drought-Response-Related Gene Module

The GO term highly enriched in Module 1 was DNA-dependent regulation of transcription (GO:0006355). According to the TF list obtained from the plant TF special database, PlantTFDB [63], about a fifth of the genes in Module 1 (143/713) consisted of TFs: WRKY (27), ERF (23), NAC (17), C2H2 (10), bZIP (9), bHLH (7), MYB (6), GRAS (6), HSF (5), MYB related (5), Trihelix (4), C3H (4), HD-ZIP (2), Dof (2), NF-YB (2), SBP (2), G2-like (2), ARR-B (2), NF-YC (1), CO-like (1), RAV (1), CAMTA (1), VOZ (1), ARF (1), CPP (1), and DBB (1), where the numbers in parentheses indicate the number of genes included in the module. Among them, TF families such as WRKY, ERF, NAC, C2H2, bZIP, bHLH, and MYB are well-known drought-stress-related TF families [24, 77, 92, 96, 97, 123]. Moreover, alterations in the expression of 10 TFs in Module 1, Os01g0797600 (*OsAP37*) [101], Os01g0968800 (*OsDREB1F*) [131], Os02g0654700 (*OsAP59*) [101], Os03g0741100 (*OsbHLH148*) [122], Os03g0815100 (*SNAC1*) [54], Os03g0820300 (*ZFP182*) [143], Os05g0322900 (*OsWRKY45*) [106], Os11g0127600 (*ONAC045*) [144], Os11g0184900 (*OsNAC5*) [124], and Os12g0583700 (*ZFP252*) [135], have already been reported to produce drought-resistant phenotypes.

Reverse-transcription-polymerase chain reaction (RT-PCR) experiments confirmed the expression levels of the TFs in Module 1 that have not yet been documented to affect drought resistance. Among them, eight TF genes, Os02g0764700 (*OsERF103*), Os03g0180900 (*TIFY11C*, *OsJAZ2*), Os03g0327100 (*ONAC039*, *OsCUC1*), Os03g0820400 (*ZFP15*), Os04g0671800 (*OsC3H32*), Os04g0676700, Os06g0670300, and Os12g0123800 (*ONAC132*, *ONAC300*), were shown to be upregulated in both plants but less upregulated in *erf71* during the 0-to-1 HAT period in response to drought stress (Figure 3.5). Os03g0180900 (*TIFY11C*, *OsJAZ2*) is a gene of the JAZ family that contains a well-conserved domain called ZIM or TIFY [127]. It was induced under drought

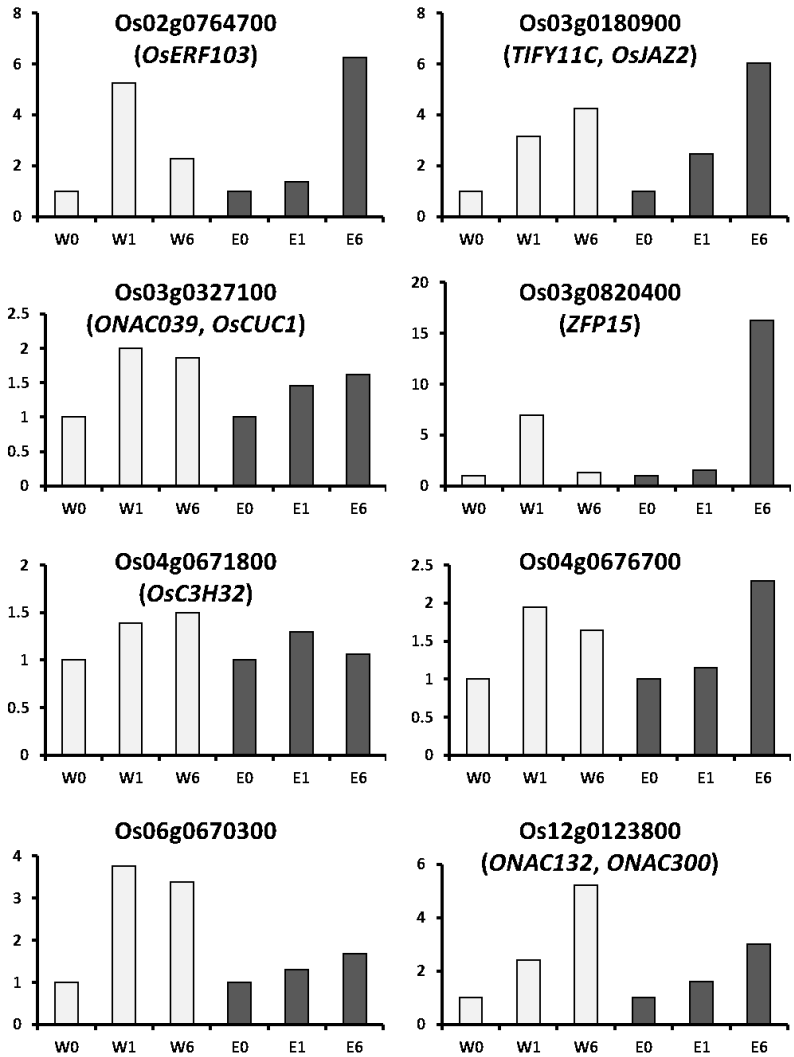

Figure 3.5. RT-PCR analysis of eight TF genes in Module 1. RT-PCR experiments measured the expression levels again for the TFs in Module 1 that have not yet been documented to affect drought resistance. Among them, eight TF genes (i.e., Os02g0764700 (*OsERF103*), Os03g0180900 (*TIFY11C*, *OsJAZ2*), Os03g0327100 (*ONAC039*, *OsCUC1*), Os03g0820400 (*ZFP15*), Os04g0671800 (*OsC3H32*), Os04g0676700, Os06g0670300, and Os12g0123800 (*ONAC132*, *ONAC300*)) were shown to be upregulated in both plants but less upregulated in *erf71* in the 0-to-1 HAT period in response to drought stress as in the RNA-Seq experiment.

stress and by the overexpression of *OsbHLH148*, a gene that causes drought tolerance when overexpressed. In addition, *OsJAZ2* exhibited a weak interaction with *OsbHLH148*, and it has been proposed to target the activation of *OsbHLH148* [122]. Os03g0327100 (*ONAC039*, *OsCUC1*) and Os12g0123800 (*ONAC132*, *ONAC300*) were reported to be responsive to drought, salt, and cold stress [40]. These results show that the eight TFs were differentially regulated in WT and *erf71* during the 0-to-1 HAT period and putatively related to the drought-resistance mechanism.

Genes in Module 1 were upregulated in both WT and *erf71*, and Module 1 was the only upregulated module among all five modules. As overall gene expression levels decreased with continued dehydration stress, indicating the suppression of various activities, upregulation was a relatively unexpected phenomenon. Module 1 contained many upregulated DEGs (31 up-DEGs among a total of 112 up-DEGs in both plants in the 0-to-1 HAT period) with a significance level of $p < 1.0e^{-23}$ by Fisher's exact test. In summary, the expression level of TF genes was increased in the dehydration response network, and those TF genes seemed to form a modular structure in the network clustering analysis, suggesting that the TF module might have particular biological functions. This observation needs further investigation, but this was beyond the scope of this study.

OsERF71, the overexpressed gene, was present in Module 1, and it had three direct neighbors (Os03g0701700, Os10g0346600, and Os11g0157200) in the module. The genes in Module 1 were directly connected to the transgene, unlike those in the other modules.

Although gene expression levels increased in Module 1, the degree of change differed between WT and *erf71*. Gene expression increased less in *erf71* in the early response phase (i.e., the 0-to-1 HAT period). This trend, relatively small gene expression changes in *erf71* in the early response phase, was observed consistently in the

results of other analyses: the number of DEGs was smaller in *erf71*, and the magnitude of change in expression was smaller for genes in Modules 2, 3, and 4 and globally during the 0-to-1 HAT period.

3.3.3 Analysis of Survival-Related Gene Modules

Three modules, Module 2, 3, and 4, included genes that were relatively upregulated in *erf71* compared with WT. The enriched GO terms were genetic information processing and translation for Module 2, response to oxidative stress for Module 3, and cell cycle for Module 4. All significantly enriched GO terms were commonly related to essential biological processes for sustaining life.

In Module 3, 54 genes were related to oxidative reduction (GO:0055114), while 23 were related to response to oxidative stress (GO:0006979). Oxidation is closely related to water deficiency tolerance in plants. In particular, reactive oxygen species (ROS) are known to be overproduced in response to abiotic stress. ROS are highly reactive and toxic, causing damage to proteins, lipids, carbohydrates, and DNA when they exceed the cell's antioxidant removal capacity [46, 91]. Since those genes in *erf71* were downregulated to a lesser extent than in WT, it is possible that *erf71* is more capable of detoxifying the rising level of oxidation, preventing severe damage to the plant.

3.3.4 Analysis of Photosynthesis-Related Gene Module

Module 5 consisted of genes that were downregulated more in *erf71* compared with WT. The significant GO terms enriched in the module were related to photosynthesis ($p < 1.0e^{-13}$). During photosynthesis, the plant synthesizes chemical compounds using energy from light. However, such photosynthetic metabolic processes require the plant to use energy. For example, toxic elements are generated as a sub-

sidiary product that must be detoxified, requiring the plant to produce anti-toxic elements. Thus, maintaining such photosynthetic metabolism during a critical situation, such as dehydration, hinders the survival of the plant [108]. In the analysis, *erf71* transgenic rice showed the strong downregulation of the expression levels of photosynthetic genes compared with WT, suggesting that *erf71* was possibly able to shut down photosynthesis mechanisms in response to dehydration stress.

3.3.5 Biological Validation Experiment

Smaller changes in gene expression in *erf71* in the early response phase (i.e., the 0-to-1 HAT period) were observed consistently. For instance, the number of DEGs was smaller in *erf71*, and the magnitude of the decrease in expression was smaller in *erf71* when considering all genes. The TF network analysis also showed that genes in Modules 2, 3, and 4 were related to survival-associated biological functions under stress conditions, such as microtubule-based movement, translation, and response to oxidative stress, and these were downregulated less in *erf71* compared with WT. This observation is intuitive, since maintaining the gene expression levels of survival-related genes promotes the dehydration-resistant phenotype. However, genes in Module 5 that were related to photosynthesis showed a greater response in *erf71* (i.e., the genes in Module 5 were downregulated more in *erf71*). Since this was a key observation in this study, the photosynthetic levels were measured for WT and *erf71* plants under dehydration stress through an experiment at the physiological level. The experiment confirmed that net photosynthesis levels decreased in both plants but with greater magnitude (two-fold) in *erf71* (Figure 3.6).

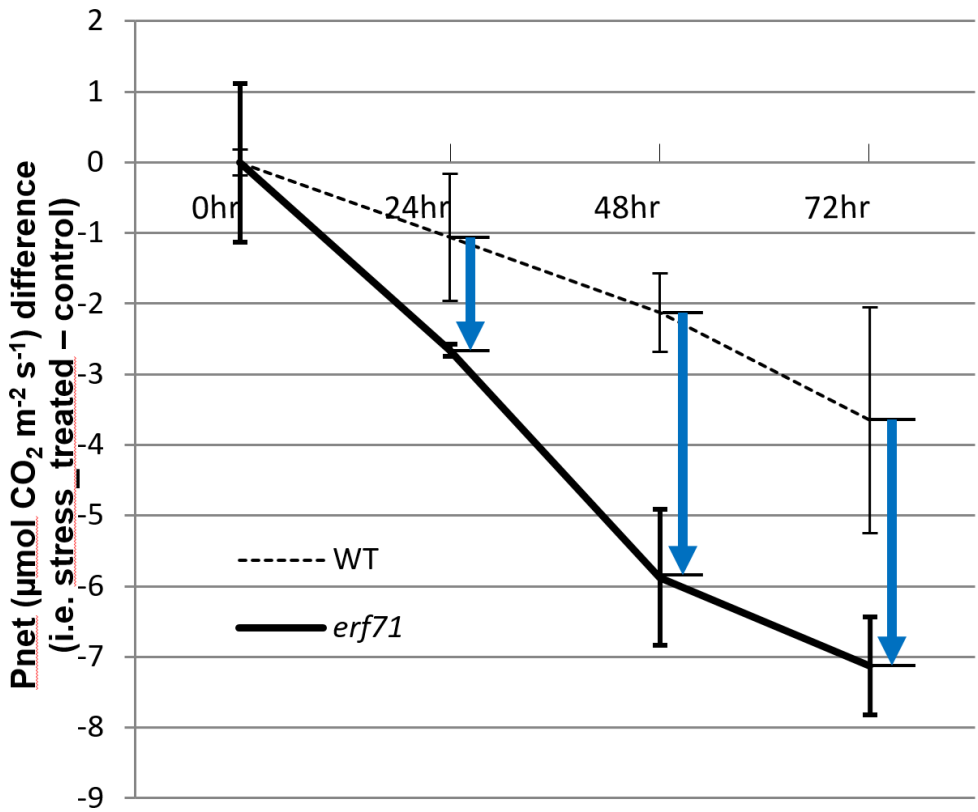

Figure 3.6. Differences of net photosynthesis levels in WT and *erf71* plants under drought stress treatment. The net photosynthesis levels were measured for WT and *erf71* at four time points under drought stress and then normalized with respect to the control sample (i.e., stress-treated sample – control sample). Error bars are pooled SEMs. The net photosynthesis level was downregulated in both types of rice but more in *erf71*, as denoted by blue arrows.

3.4 Summary

This chapter analyzed time-series gene expression data measured over time after drought stress treatment for two rice samples: WT and GM. The goal was to investigate the drought resistance mechanism of the GM rice. The computational problems of the analysis were the biased ratio between the number of features and the number of samples, the high dependency between dimensions induced by complex interactions between genes, and the small number of time points.

To address the problems, this study proposed a two-step comprehensive computational framework, RiceTFnetwork, involving the construction, integration, and clustering of gene networks. The first step used network representation for the integrated analysis of gene expression data and external domain data. The second step conducted clustering analysis for dimension reduction from the number of genes to the number of subnetworks. Characterization of gene subnetworks suggested that GM diverted more energy to survival-critical mechanisms related to translation, oxidative response, and DNA replication while further suppressing energy-consuming mechanisms, such as photosynthesis. The follow-up biological experiments confirmed the further suppression of the photosynthesis of GM at the physiological level.

Chapter 4

HTRgene: Integrating Multiple Heterogeneous Time-series Data to Investigate Cold and Heat Stress Response Signaling Genes in Arabidopsis

Different background conditions of samples lead to large variance of responses to the stress across the samples, making it difficult to recognize the signal. It is well known that when more data are used for the analysis, the signal-to-noise ratio is increased, and the accuracy of the results improves. Fortunately, the time-series gene expression data under the same stress with different experimental conditions are available in databases (see Section 2.4). However, none of the databases above provides an integrated analysis of the collected data. This study collects multiple time-series datasets from public databases measured under the same stress and conducts an integrated analysis of them. The goal of analysis in this chapter is an integration of heterogeneous time-series gene expression data to investigate how Arabidopsis plant responses to stress.

4.1 Computational Problems

Figure 4.1 illustrates the input data of this chapter, time-series gene expression datasets of 28 samples of Arabidopsis under cold stress treatment, and the challenges. The heterogeneous structure of time-domain (i.e., different time points) and the variance of experimental conditions between individual samples (i.e., different tissue, age, background, etc.) arise the computational challenges. The heterogeneous time-domain and variance of experimental conditions between individual samples cause the response timing to vary from sample to sample, making the integration of data difficult [4, 65]. The analysis of heterogeneous time-domain data is a new problem of time-series data analysis as far as the author is aware.

4.2 Methods

This section presents a method called **HTRgene** [7] that determines stress-responsive genes by integrating and analyzing the heterogeneous gene expression data under the same stress. The key ideas of integration are "response time (RT)" and "response order". It works on the assumption that the response order of genes will be preserved even if the response time of genes is advanced or delayed across multiple samples. In addition, HTRgene uses clustering analysis to reduce the complexity of computation.

The following are definitions of the concepts used in heterogeneous time-series data integration analysis.

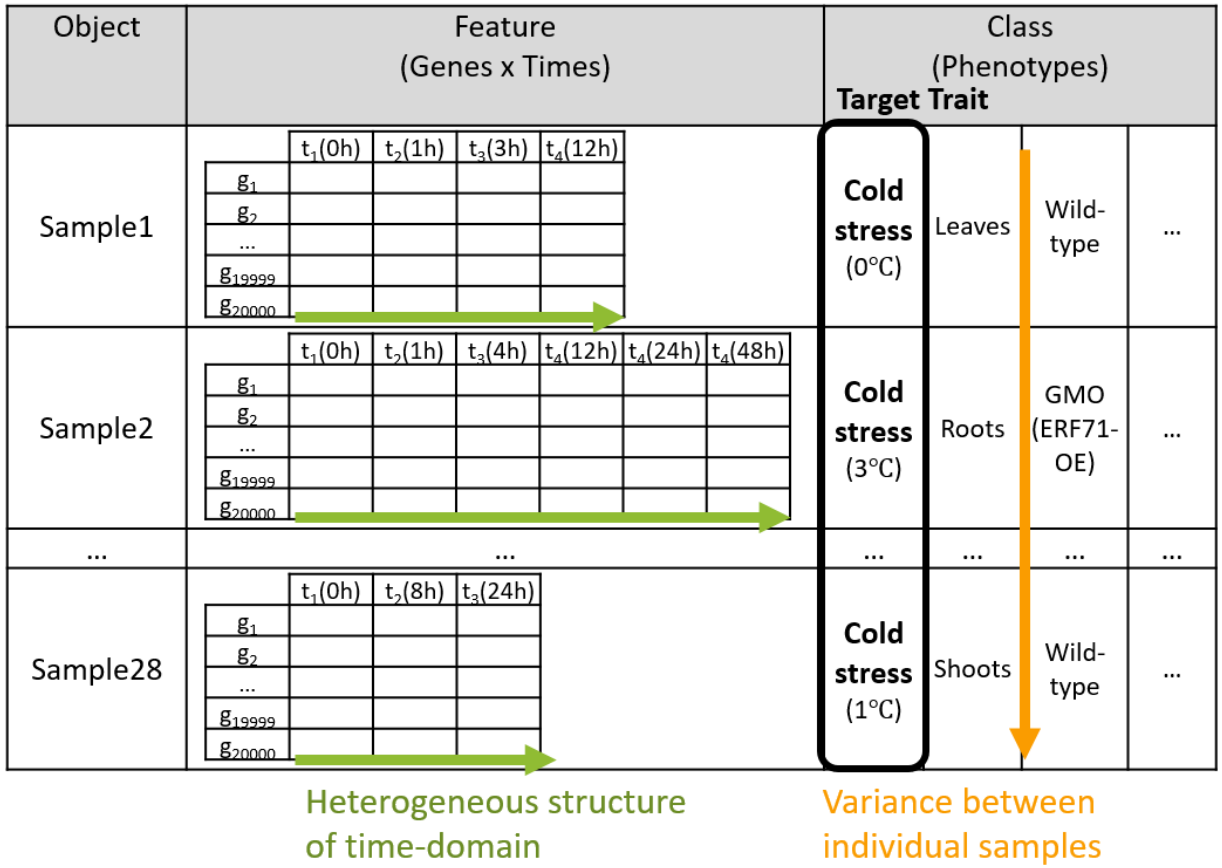

Figure 4.1. Heterogeneous time-domain and phenotype-domain gene expression data. The data include 28 samples of Arabidopsis under cold stress treatment. The goal of this research is to understand the cold stress response mechanism by integrating multiple time-series datasets.

Definition 2. Let l_i be the number of time points in time-series sample i and $e_{g,i,j}$ be the expression level of a gene g in sample i at time point j. Let also $B_{g,i,j}$ be a set of expression levels of a gene g in sample i before time point j excluding j, i.e., $\{e_{g,i,1}, \ldots, e_{g,i,j-1}\}$ and $A_{g,i,j}$ be a set of expression levels of a gene g in sample i after time point j including j, i.e., $\{e_{g,i,j}, \ldots, e_{g,i,l_i}\}$.

- A **response time** t_g^i of a gene g in sample i is a time point where a statistical test of significance of expression level difference is maximized between B_{g,i,t_g^i} and A_{g,i,t_g^i}.
- A **response time vector** $\vec{R_g}$ of a gene g is a vector of response time $< t_g^1, \ldots, t_g^m >$ for m samples.
- **The order of two response time vectors**, $\vec{R_{g_1}}$ and $\vec{R_{g_2}}$, is determined as $\vec{R_{g_1}} \preceq \vec{R_{g_2}}$ if $\dot{t_{g_1}} < \dot{t_{g_2}}$ for at least one sample and $\dot{t_{g_1}} \leq \dot{t_{g_2}}$ for all samples.
- A **response schedule** is a longest consistent ordering of genes for a set of binary ordering of two genes based on response time vectors.
- A **response phase** is the position of response in the response schedule.
- A **candidate response genes** are defined as genes that are differentially expressed significantly in multiple samples and whose response phases can be determined.

There are two complexity-related challenges for determining the response time and order of HTRgene.

- **Computational challenge 1:** The complexity of determining response time vectors depends on the number of samples and time points. Given n time points, there are potentially $n-1$ possible response time points, excluding one before the first time point and one after the last time point, for one sample. The integrated analysis of m samples has to consider $(n-1)^m$ candidates to determine a response time vector.
- **Computational challenge 2:** The complexity of ordering the response time vectors depends on the number of genes. In Arabidopsis, there are 27,416 cod-

ing genes [74], which is too many to be ordered.

HTRgene reduces complexity by determining and ordering the response times at the gene cluster level, not at the gene level, which reduces the computation depending on the number of clusters. The process of HTRgene consists of four steps, as illustrated in Figure 4.2.

4.2.1 Step 1: Normalization and Detection of Consensus DEGs

HTRgene takes a set of time-series gene expression data from a single platform, either microarray or RNA-Seq, as input. Different scale normalization methods are used depending on the data platform.Quantile normalization using the `affy` R package [45] is used for microarray data, and variance stabilization transformation using the DESeq package [11] is used for RNA-Seq data. After scale normalization, HTRgene performs base normalization to initialize expression values at the initial time point (i.e., the time point before the stress $T = 0$) to zero. Different basement normalization methods are used depending on the shape of data distribution. For instance, when plotting expression levels of a gene, the plot follows a normal distribution, so substitution-based normalization (eq. 4.1) is used for normal-shaped data. However, log-fold-change-based normalization (eq. 4.2) is used for log-scale-shape distribution data, which is the standard practice for RNA-Seq data.

The expression level $e_{g,i,j,k}$ of gene g measured in time-series sample i at time point j in a replicate k is adjusted as follows for microarray data:

$$e_{g,i,j,k} - \frac{1}{|R|} \sum_{k}^{|R|} e_{g,i,0,k} \tag{4.1}$$

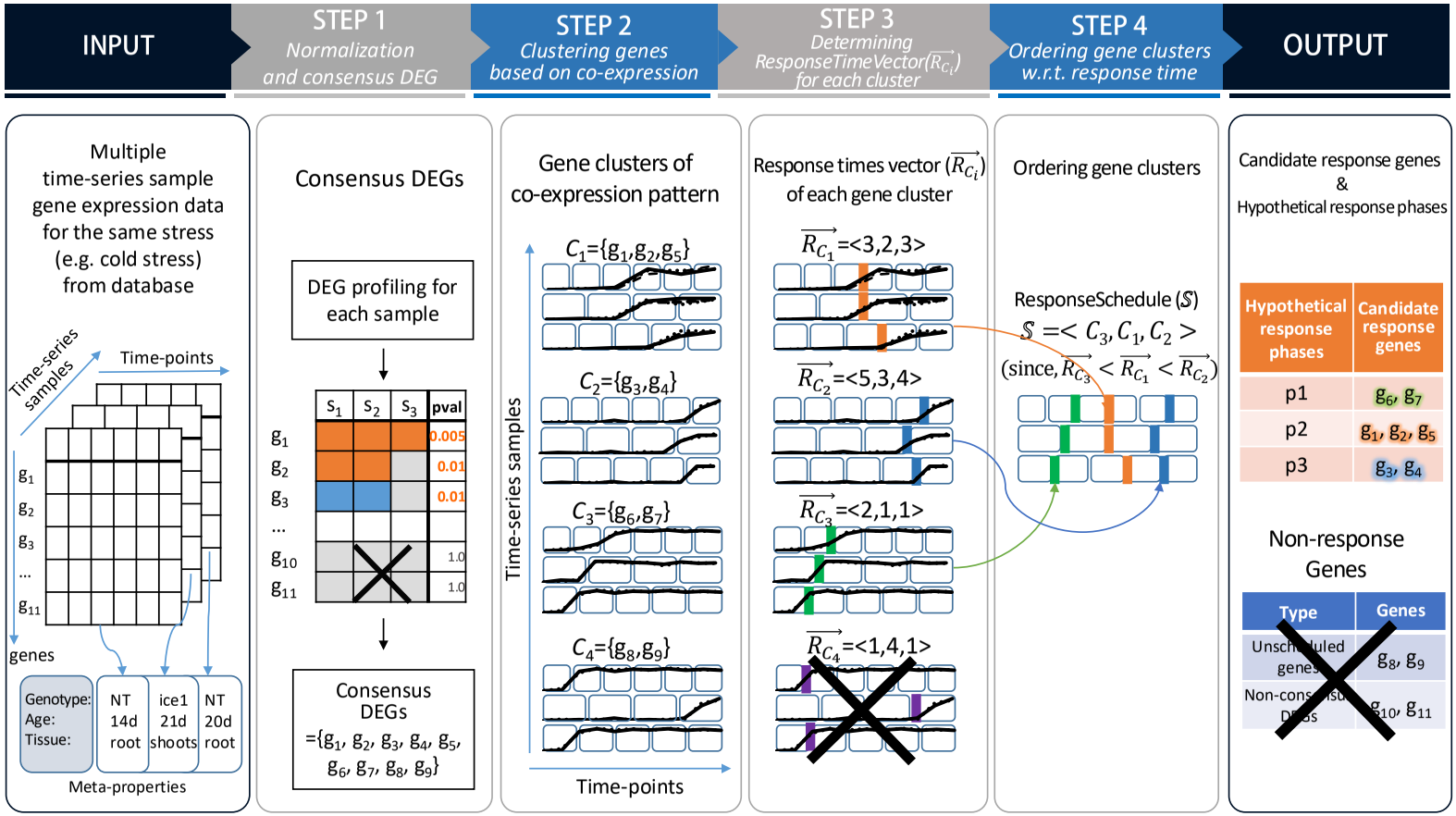

Figure 4.2. Overview of HTRgene algorithm. In Step 1, multiple time-series gene expression datasets of the same stress (e.g., cold stress) are collected from databases and normalized, and then consensus DEGs are selected. In Step 2, genes with high co-expression patterns are clustered. In Step 3, a response time vector $\vec{R}_{C_i}$ is determined for each gene cluster. In Step 4, gene clusters are ordered and aligned in a response schedule according to their response time. Finally, response genes and their response phases (i.e., early or late response) are provided.

and as follows for RNA-Seq data:

$$log(e_{g,i,j,k}+1)-\frac{1}{|R|}\sum_{k}^{|R|}log(e_{g,i,0,k}+1). \tag{4.2}$$

After normalization, HTRgene determines consensus DEGs across multiple time-series samples. First, single-sample DEGs on the time domain were determined (i.e., DEGs in each sample). Differential expression tests were performed for each time point with respect to the time point before the stress ($T = 0$) using the limma [109] tool. A gene is considered a DEG in a single time-series sample if it is differentially expressed in at least one time domain in the sample.

To evaluate the significance of a DEG across multiple time-series samples, a statistical test was performed on the number of samples in which a gene could be a DEG. A matrix of genes vs. time-series samples was constructed for the genes that were determined as DEGs in samples. Then, the matrix elements were randomly shuffled and a random distribution was generated by counting the number of samples where a particular gene was a DEG. In this way, the p-value of DEG frequencies was measured, and Benjamini-Hochberg multiple correction [16] was performed. Then, the genes were selected whose DEG frequencies were significant ($adj.p < 0.05$) for samples, and they were considered consensus DEGs.

The top 10% significant genes among consensus DEGs were selected as pseudo-reference genes since they were highly likely true-response genes for stress. These pseudo-reference genes were used to determine parameters for further analysis steps.

4.2.2 Step 2: Gene Clustering Based on Co-Expression Patterns

To determine the response time points of the multiple time-series samples, clustering of genes was performed across different samples. Since there are three dimensions, genes $\times$ samples $\times$ time points, clustering is very difficult. A recent three-dimensional clustering algorithm To determine the response time points of the multiple time-series samples, clustering of genes was performed across different samples. Since there were three dimensions, genes $\times$ samples $\times$ time points, clustering was very difficult. A recent 3D clustering algorithm [64] was used. The basic idea is to generate a single vector for each gene by concatenating the time and the sample dimensions. Then a spherical K-means (skmeans) clustering algorithm [20] is used to generate clusters based on the cosine distance between two vectors. The 3D clustering produced a set of gene clusters, $\{C_1, \ldots, C_K\}$. Among them, small-sized clusters with less than three members were excluded. The method to determine the number of clusters K is described in the "Determining the number of gene clusters" section.

4.2.3 Step 3: Response Time Vector Detection for Each Gene Cluster

This step determines the response time vector $\vec{R_{C_i}}$ for each gene cluster C_i using a t-test on the difference in gene expression values before and after response vectors. Determining an optimal response time vector is a computationally complex problem because of its exponentially increased search space. To handle the big search space issue, a hill-climbing approach was used to determine the optimal RT solution suggested in [140]. Step 3 consists of three substeps: 1) initializing an RT, 2) generating a new candidate RT, and 3) selecting an RT that improves the separation score. Steps

2) and 3) are repeated until no candidate RT improves the separation score.

Initializing $\vec{R_{C_i}}$ using a hierarchical clustering

The hierarchical clustering of genes is used to generate the initial $\vec{R_{C_i}}$. Since the goal is to determine a time point as a stress response time, hierarchical clustering is performed on the time dimension, progressively merging adjacent time points based on gene expression values. To set the initial $\vec{R_{C_i}}$, a response time r_i is determined for each sample i for all genes in C_i and then $\vec{R_{C_i}}$ is a vector $< t^1_{C_i}, \ldots, t^s_{C_i}, \ldots, t^m_{C_i} >$ where $t^s_{C_i}$ is a response time for each sample s.

Generating and selecting a new candidate $\vec{R_{C_i}}$

The next step is to generate a candidate $\vec{R_{C_i}}$ by moving an element of $\vec{R_{C_i}}$ to a nearby time point. The testing is done by performing a t-test on the gene expression difference before and after a $\vec{R_{C_i}}$ vector.

$Tstat^R{}_{g_j}$, the quality score of response point R based on a statistical model of the individual gene g_j, is calculated according to the following procedure. Suppose that $EXP^{pre}_{g_j}$ and $EXP^{post}_{g_j}$ are the sets of expression values of gene g_j where the expression value of gene g_j of sample s_i is assigned to $EXP^{pre}_{g_j}$ or $EXP^{post}_{g_j}$ depending on whether s_i belongs to the pre-response or post-response group. $Tstat^R{}_{g_j}$ is defined as the absolute value of t-statistics with an assumption of two-sample equal variance. Then, $Tstat^R{}_{C_i}$, the quality score of a cluster C_i, is an average of quality scores of all genes in C_i.

4.2.4 Step 4: Ordering Gene Clusters

Among all clusters, the goal is to select and order a set of clusters that are consistent in terms of response times. To do this, the concept of *ResponseSchedule* was defined. Informally, a response schedule $\mathbb{S}$ is the longest consistent ordering of response time vectors without conflict. The "conflict" between $\vec{R_{C_i}}$ of two clusters means that no ordering of the two clusters can be determined. For instance, when the current schedule $\mathbb{S}$ is $\{<1,1,1,1>,<3,3,4,5>\}$, a candidate $\vec{R_i} =<3,3,5,4>$ has conflict with $<3,3,4,5>$ because the third and fourth elements have disagreeing orders.

Each cluster was considered in the order of quality scores $Tstat^R{}_{C_\bullet}$. Initially, $\mathbb{S}$ is empty, so the cluster with the highest quality score is added to $\mathbb{S}$. Then, the cluster C_i with the next best quality score is considered. If C_i does not have conflicts with any of the clusters that are already included in $\mathbb{S}$, then C_i is added to $\mathbb{S}$. Otherwise, C_i is rejected. This process continues until there is no cluster to be considered. Then, the "response phases" are defined as the positions of the clusters remaining in ResponseSchedule $\mathbb{S}$.

4.2.5 Determining the Number of Gene Clusters

Once consensus DEGs and pseudo-reference genes were determined in Step 1, sets of candidate response genes were generated by increasing K in steps of 50 starting from 50 to half of the number of consensus DEGs. Then, K, maximizing the association between pseudo-reference genes and candidate response genes by measuring F1 score, was chosen.

4.3 Results and Discussion

4.3.1 Cold and Heat Stress Datasets

HTRgene was applied to two stress type (i.e., heat and cold) time-series data in Arabidopsis. Affymetrix microarray platform raw data were collected from GEO [13] and ArrayExpress [72]. This study focused on detecting genes and aligning them according to their response time to a single stress factor. Thus, the recovery phase data were excluded from the dataset. The collected raw data were processed and quantile normalized using the `affy` R package [45]. The 28 and 24 time-series sample datasets for cold and heat stress showed heterogeneous meta-properties, as shown in Tables 4.1 and 4.2, respectively. HTRgene produced 425 and 272 candidate response genes that were assigned to 12 and 8 response phases as a result of the integrated analysis of cold and heat stress datasets, respectively. The heat map of 425 candidate genes to cold stress in Figure 4.3 shows that the response time defined by the HTRgene method propagates along the time points as the response phases proceed from p1 to p12. This result shows that HTRgene successfully determined the gene clusters and their orders that are consistent with signaling propagation along the cold stress pathway. The next section investigates whether the results are consistent with actual biological mechanisms found through the literature review.

4.3.2 Reproduction of Cold Stress Pathway

The integrated analysis for cold stress data produced 425 candidate response genes that were assigned to 12 response phases. The results were compared to known cold stress pathway genes reported in review papers [61, 90, 145]. As shown in Figure 4.4, the cold stress pathway can be organized into a three-level pathway: signal transmission, TF cascade, and downstream gene level pathways.

Table 4.1. Heterogeneous meta-properties of 28 time-series gene expression datasets for cold stress treatment.

No.	Dataset ID	Eco-type	Geno-type (NT: non-transgenic)	Age (days)	Tissue (Extra condition)	Temperature treatment (°C)	Time points (minutes (m) or hours (h) after treatment)
1	E-MTAB-375	Columbia	NT	14	rosette leaf (low light)	4	0h, 5m, 10m, 20m, 40m, 1h, 80m, 100m, 2h, 140m, 160m, 3h, 200m, 220m, 4h, 260m, 280m, 5h, 320m, 340m, 6h, 10h40m, 21h20m
2	E-MTAB-375	Columbia	NT	14	rosette leaf (dark)	4	0h, 5m, 10m, 20m, 40m, 1h, 80m, 100m, 2h, 140m, 160m, 3h, 200m, 220m, 4h, 260m, 280m, 5h, 320m, 340m, 6h, 10h40m, 21h20m
3	GSE5621	Columbia	NT	14	shoot	4	0h, 30m, 1h, 3h, 6h, 12h, 24h
4	GSE5621	Columbia	NT	14	root	4	0h, 30m, 1h, 3h, 6h, 12h, 24h
5	GSE3326	Columbia	NT	14	seedlings	0	0h, 3h, 6h, 24h
6	GSE3326	Columbia	ice1	1	seedlings	0	0h, 3h, 6h, 24h
7	GSE55835	Columbia	NT	42	leaves	-3	0h, 8h, 24h, 72h
8	GSE55835	Rschew	NT	42	leaves	-3	0h, 8h, 24h, 72h
9	GSE55835	Tenela	NT	42	leaves	-3	0h, 8h, 24h, 72h
10	GSE5534	Columbia	NT	10	seedlings (plate)	4	0h, 1h, 24h, 168h
11	GSE5535	Columbia	NT	10	seedlings (soil)	4	0h, 1h, 24h, 168h
12	GSE53990	Columbia	NT	28	9-11th adult leaves	4	0h, 48h, 120h
13	GSE53990	Columbia	rcf	28	9-11th adult leaves	4	0h, 48h, 120h
14	GSE39090	Columbia	NT	14	seedlings	4	0h, 12h, 24h
15	GSE39090	Columbia	rcf	14	seedlings	4	0h, 12h, 24h
16	GSE37130	C24	NT	20	seedlings	4	0h, 3h, 24h
17	GSE37130	Columbia	NT	20	seedlings	4	0h, 24h
18	GSE43818	Columbia	NT	21	entire aerial part	4	0h, 24h
19	GSE43818	Columbia	camta1/2/3	21	entire aerial part	4	0h, 24h
20	GSE55906	WS-2	NT	11	entire aerial part	4	0h, 24h
21	GSE55906	WS-3	CBF2DN	11	entire aerial part	4	0h, 24h
22	GSE55907	Columbia	NT	12	seedlings	4	0h, 24h
23	GSE64575	Columbia	NT	10	entire aerial part	4	0h, 24h
24	E-MEXP-1345	Columbia	NT	45	leaf tip	4	0h, 24h
25	GSE19254	Columbia	NT	38	aerial tissues	4	0h, 48h
26	GSE19254	Columbia	sfr3	38	aerial tissues	4	0h, 48h
27	E-MEXP-3714	Columbia	NT	11	aerial tissues	1	0h, 2h
28	E-MEXP-3714	Columbia	ahk2ahk3	11	aerial tissues	1	0h, 2h

Table 4.2. Heterogeneous meta-properties of 24 time-series gene expression datasets for heat stress treatment.

No.	Dataset ID	Eco-type	Geno-type (NT: non-transgenic)	Age (days)	Tissue (Extra condition)	Temperature treatment (°C)	Time points (minutes (m) or hours (h) after treatment)
1	E-MTAB-375	Columbia	NT	14	rosette leaf (normal light)	32	0h, 5m, 10m, 20m, 40m, 1h, 80m, 100m, 2h, 140m, 160m, 3h, 200m, 220m, 4h, 260m, 280m, 5h, 320m, 340m, 6h, 10h40m, 21h20m
2	E-MTAB-375	Columbia	NT	14	rosette leaf (dark)	32	0h, 5m, 10m, 20m, 40m, 1h, 80m, 100m, 2h, 140m, 160m, 3h, 200m, 220m, 4h, 260m, 280m, 5h, 320m, 340m, 6h, 10h40m, 21h20m
3	GSE5628	Columbia	NT	16	shoots	38	0h, 15m, 30m, 1h, 3h
4	GSE5628	Columbia	NT	16	roots	38	0h, 15m, 30m, 1h, 3h
5	GSE62163	Columbia	NT	21	shoots (EBR)	43	0h, 1h, 3h
6	GSE62163	Columbia	NT	21	shoots (no EBR)	43	0h, 1h, 3h
7	GSE63128	Columbia	NT	18	leaves	38	0h, 8h, 24h
8	E-MEXP-2760	Columbia	NT	35	shoot	40	0h, 20m, 1h
9	E-MEXP-2760	Columbia	mbf1c	35	shoot	40	0h, 20m, 1h
10	E-MEXP-3754	Columbia	NT	21	meristem	40	0h, 15m, 45m
11	GSE19603	Columbia	NT	56	above-ground	37	0h, 24h
12	GSE19603	Columbia	msh1/recA3	56	above-ground	37	0h, 24h
13	GSE43937	Columbia	NT	14	leaves	40	0h, 6h
14	GSE43937	Columbia	er-105	14	leaves	40	0h, 6h
15	E-MEXP-1725	Columbia	NT	35	leaves	37	0h, 2h
16	E-MEXP-1725	Columbia	hsf4-7	35	leaves	37	0h, 2h
17	GSE16222	Columbia	NT	4	seedlings	38	0h, 1h30m
18	GSE63372	Columbia	WT	7	seedlings	37	0h, 1h
19	GSE63372	Columbia	HSFA6b-OE	7	seedlings	37	0h, 1h
20	GSE63372	Columbia	HSFA6b-RD	7	seedlings	37	0h, 1h
21	GSE12619	Columbia	NT	7	seedlings	37	0h, 1h
22	GSE12619	Columbia	til1-1	7	seedlings	37	0h, 1h
23	GSE44053	Columbia	NT	7	seedlings	38	0h, 45m
24	GSE44053	Columbia	NT	7	seedlings	38	0h, 45m

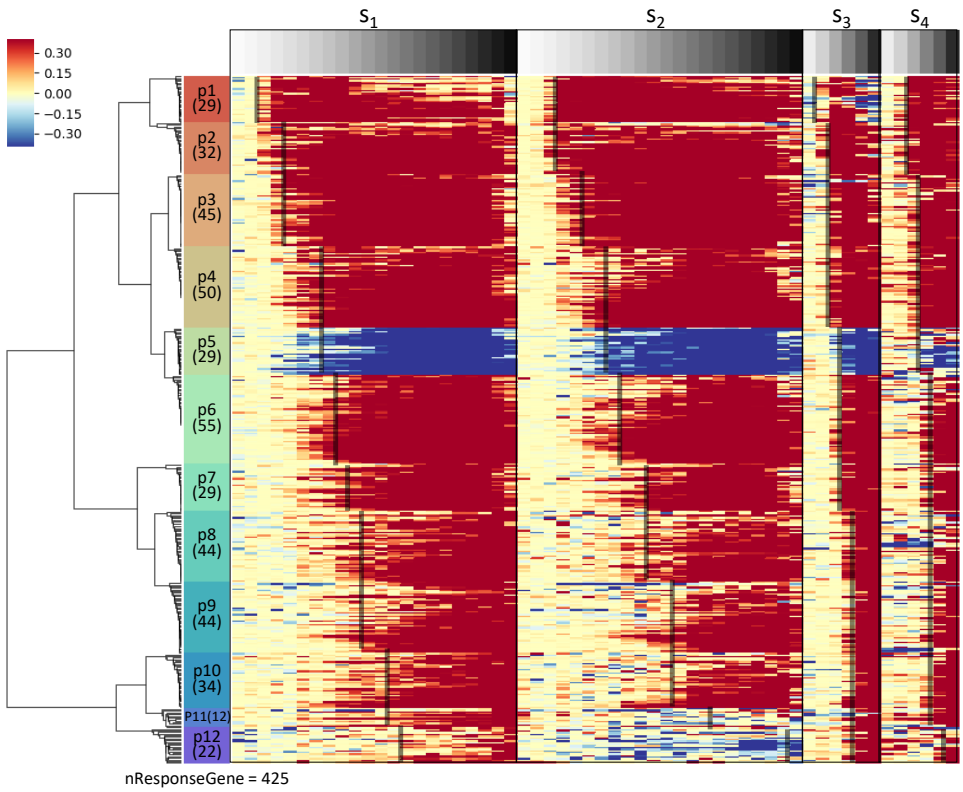

Figure 4.3. Heat map of 425 candidate response genes to cold stress during response phases. HTRgene produced 425 candidate response genes that were assigned to 12 response phases as a result of an integrated analysis of 28 cold stress time-series sample datasets. The columns of the heat map are four time-series samples with more than five time points: S_1 to S_4. The rows of the heat map are 12 response phase gene clusters, and the numbers in parentheses indicate the number of genes in a cluster. The colors of the heat map indicate up (red) or down (blue) gene expression changes in comparison to the non-stressed time point ($T = 0$). The black lines represent the response time point of a cluster in each sample. The heat map shows that the response time (the black line) defined by the HTRgene method propagates along the time points as the response phases proceed from p1 to p12. This result shows that HTRgene successfully determined the gene clusters and their orders that are consistent with signaling propagation along the cold stress pathway.

In the signal transmission level pathway, the cold stress signal first alters membrane rigidity and the concentration level of Ca^{2+}. The signal is transmitted by changing the activation status of proteins sequentially, such as *CLRKs*, *CPKs*, *CBL-CIPKs*, *MEKK1*, *MKK2*, *MPK3/4/6*, *ICE1*, *HOS1*, *CAMTA3* genes [90, 145]. These genes in the signal transmission level pathway were not assigned to any cluster in the results. This result is biologically interpretable since the actions in the signal transmission level pathway, such as phosphorylation, SUMOylation, and ubiquitination [61, 90, 145], affect the proteins' structures but not their expression levels.

In the TF cascade level pathway, *ICE1* and *CAMTA3* genes, which are the last activated genes in the signal transmission level pathway, initiate gene expression regulation [37]. However, the two genes were not assigned to any cluster since they are activated by protein-structure-modifying actions. They bind to CG1 and ICE1-box DNA cis-elements, which induces *CBF2* (*DREB2A*) and *CBF3* (*DREB1A*), respectively [61]. The *CBF2* and *CBF3* genes are assigned to the second response phases "p2" in the result. *CBF2* and *CBF3* are known to bind to CRT/DRE elements, ACCGACNN and [A/G]CCGACNT, respectively, and regulate the downstream genes [84, 115]. In addition, since they share the common CRT/DRE element ACCGACNT, they regulate some of the same downstream genes.

In the downstream level of the signal transduction pathway, among *CBF2* downstream genes [115], eight genes were found in the late response phases "p4," "p6," "p7," and "p9." Moreover, among *CBF3* downstream genes [84], 19 genes were found in the late response phases "p2," "p3," "p4," "p6," "p8," "p9," and "p10." Among them, five genes are common. Collectively, it is shown that the analysis results correspond to the known cold stress pathway.

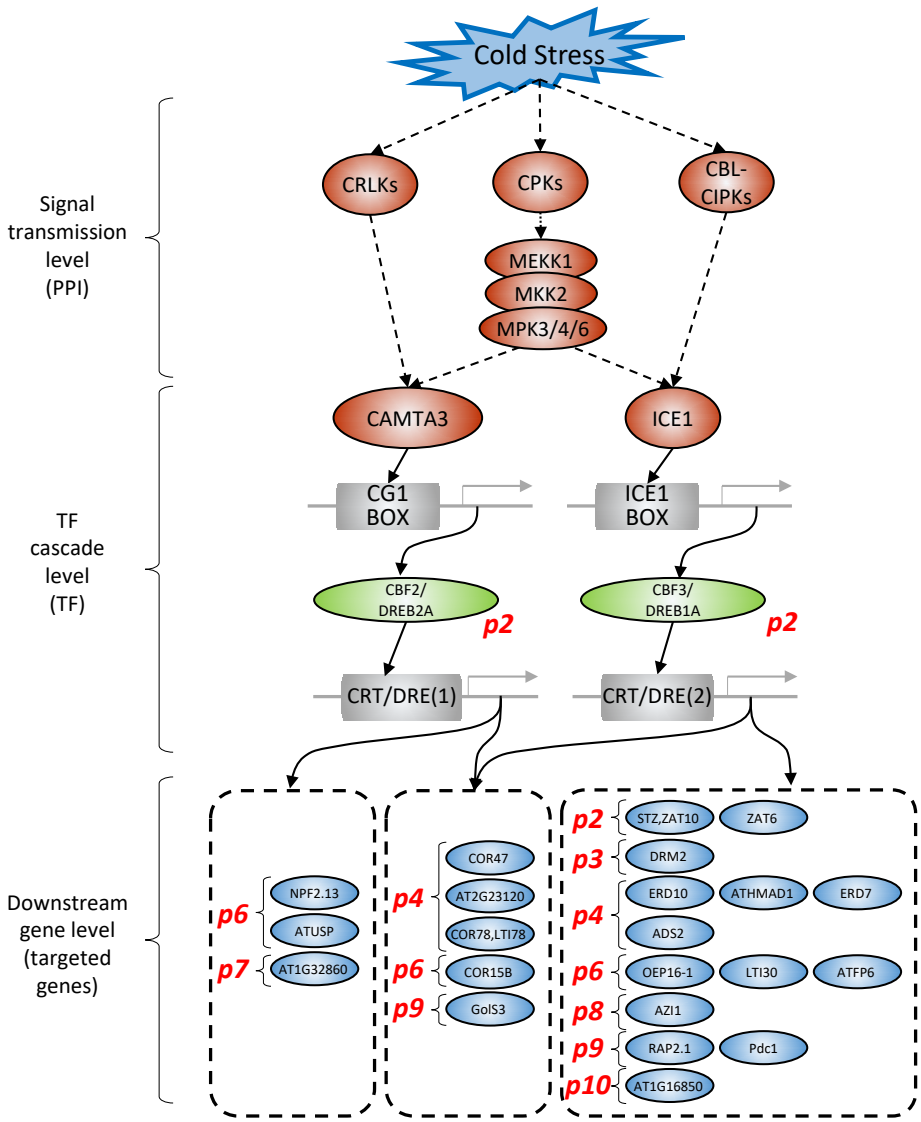

Figure 4.4. Cold stress pathway and cluster results. The cold stress pathway can be organized into a three-level pathway: signal transmission, TF cascade, and downstream gene level pathways. In the signal transmission level pathway, the cold stress signal is transmitted by changing the activation status of some proteins sequentially, such as *CLRKs*, *CPKs*, *CBL-CIPKs*, *MEKK1*, *MKK2*, *MPK3/4/6*, *ICE1*, *CAMTA3* genes. In the TF cascade level pathway, *ICE1* and *CAMTA3* genes initiate gene expression regulation. They bind to CG1 and *ICE1* DNA elements and then upregulate *CBF2* (*DREB2A*) and *CBF3* (*DREB1A*) genes, respectively. The *CBF2* and *CBF3* genes were assigned to the second response phase "p2" in the result. They are known to bind to CRT/DRE elements regulating downstream genes. In the downstream-level pathway, eight of the *CBF2* downstream genes were found in the late response phases "p4," "p6," "p7," and "p9." In addition, 19 *CBF3* downstream genes were found in the late response phases "p2," "p3," "p4," "p6," "p8," "p9," and "p10." Collectively, it is shown that the analysis results correspond to the known cold stress pathway.

4.3.3 Reproduction of Heat Stress Pathway

The integrated analysis for heat stress data produced 272 candidate response genes that were assigned to seven response phases. The results were also compared to the heat stress pathway summarized in a review [102]. As shown in Figure 4.5, the heat stress pathway can also be organized into a three-level pathway: signal transmission, TF cascade, and downstream gene level pathways.

In the signal transmission level pathway, the heat stress signal first changes membrane rigidity and the concentration level of Ca^{2+} and ROS. The signal is then transmitted by changing the activation status of some proteins sequentially, such as *CBK3*, *PP7*, *CDKA1*, *CPKs*, *CBL-CIPKs*, and *HSFA1s* genes [102]. These genes in the signal transmission-level pathway were not assigned to any cluster in the results, which is biologically interpretable since the actions in the signal transmission level pathway, such as phosphorylation, dephosphorylation, SUMOylation, and protein–protein interaction [102], affect the proteins' structures but not their gene expression levels, as in the cold stress results.

In the TF cascade level pathway of heat stress, *HSFA1* genes are the major regulator of TF cascade [78]. However, they did not show gene expression changes in the result. The result was possible because they are activated by protein-structure modifying actions. *HSFA1s* bind to HSE elements that target directly some TFs: *HSFA2*, *HSFA7A*, and *HSFBs* (*HSFB1A*, *HSFB2A*, and *HSFB2B*) [102]. Interestingly, *HSFBs*, *HSFA7A*, and *HSFA2* bind to HSE elements then activating themselves again. Thus, transcriptional upregulation is accelerated by this feed-forward signaling [56]. Among these TFs, *HSFA2*, *HSFA7A*, and *HSFBs*, the direct target of *HSFA1* were assigned to the second response phase "p2."

In the downstream level pathway, among the downstream genes of heat shock factor genes [55, 121, 142], 52 genes were found in late response phases "p2," "p3,"

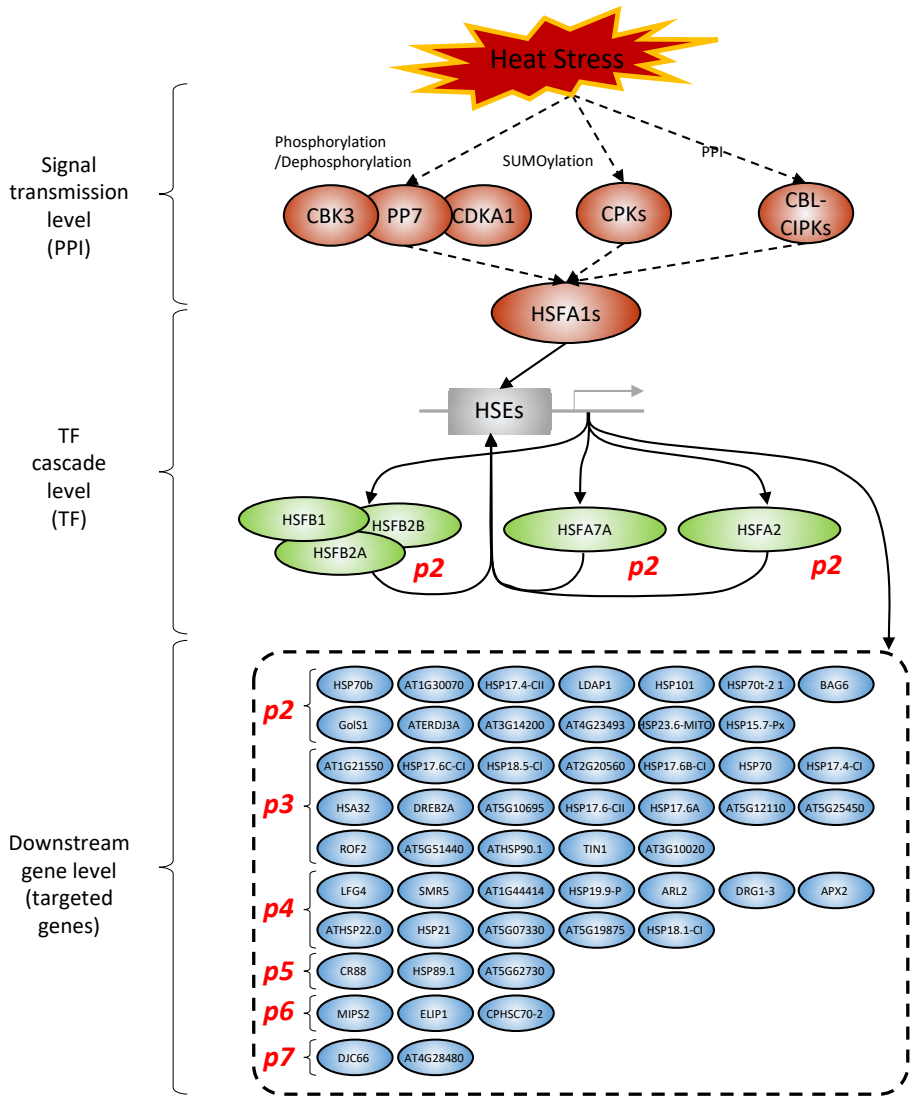

Figure 4.5. Heat stress pathway and cluster results. The heat stress pathway also can be organized into a three-level pathway: signal transmission, TF cascade, downstream gene level pathways. In the signal transmission-level pathway, heat stress signal is transmitted by changing the activation status of some proteins sequentially, such as *CBK3*, *PP7*, *CDKA1*, *CPKs*, *CBL-CIPKs*, and *HSFA1s* genes. In the TF cascade-level pathway, *HSFA1s* genes are the major regulator of TF cascade. *HSFA1s* bind to HSE elements then target directly some TFs such as *HSFA2*, *HSFA7A*, and *HSFBs* (*HSFB1A*, *HSFB2A*, and *HSFB2B*). Then, the targeted TF genes bind to HSE elements, then activate themselves again, inducing heat stress signaling acceleration. Among these TFs, *HSFA2*, *HSFA7A*, and *HSFBs* were assigned to the second response phase "p2." In the downstream-level pathway of *HSFA2*, *HSFA7A* and *HSFBs*, 52 genes among heat shock factor downstream genes were found in late response phases "p2," "p3," "p4," "p5," "p6," and "p7." Collectively, it is shown that the analysis results correspond to the known cold stress pathway.

"p4," "p5," "p6" and "p7." Collectively, it is shown that the analysis results correspond to the known heat stress pathway.

4.3.4 Comparison with Existing Methods

HTRgene was evaluated by comparison with existing tools. Qualitatively, HTRgene is more informative than other tools since it provides not only candidate response genes but also response phases. For instance, DEG detection tools such as limma [109], edgeR [110], and DESeq [11] output DEGs by comparing the control samples with the case samples but do not produce response phases. Other pattern-based tools, such as ImpulseDE [116], are designed to report differentially patterned genes between control and case time-series samples but do not report response phases.

Since the existing tools do not provide response phases, the performance comparison is done in terms of accuracy of determining candidate response genes only. For performance comparison, HTRgene without ordering and with ordering were considered separately in order to trace how much improvement was made by ordering the reference genes (i.e., ground truth) to estimate accuracy. 330 and 158 genes with GO annotation "response to cold" and "response to heat" from the TAIR database [74], as limma, ImpulseDE, HTRgene without ordering, and HTRgene with ordering, produced 3449, 7840, 3602, and 425 candidate response genes for cold stress analysis; and 5091, 8193, 2957, and 272 candidate response genes for heat stress analysis, respectively, as shown in Figure 4.6A. Among the genes, 41, 56, 124, and 41 were ground truth genes for cold stress; and 73, 83, 69 and 49 ground truth genes for heat stress, respectively. To measure the performance quantitatively, F1 score was computed (a widely used accuracy metric to consider both precision and recall). HTRgene outperformed the other tools in terms of the F1 score about three-fold as shown in Figure 4.6B. In addition, ordering was effective to reduce the number

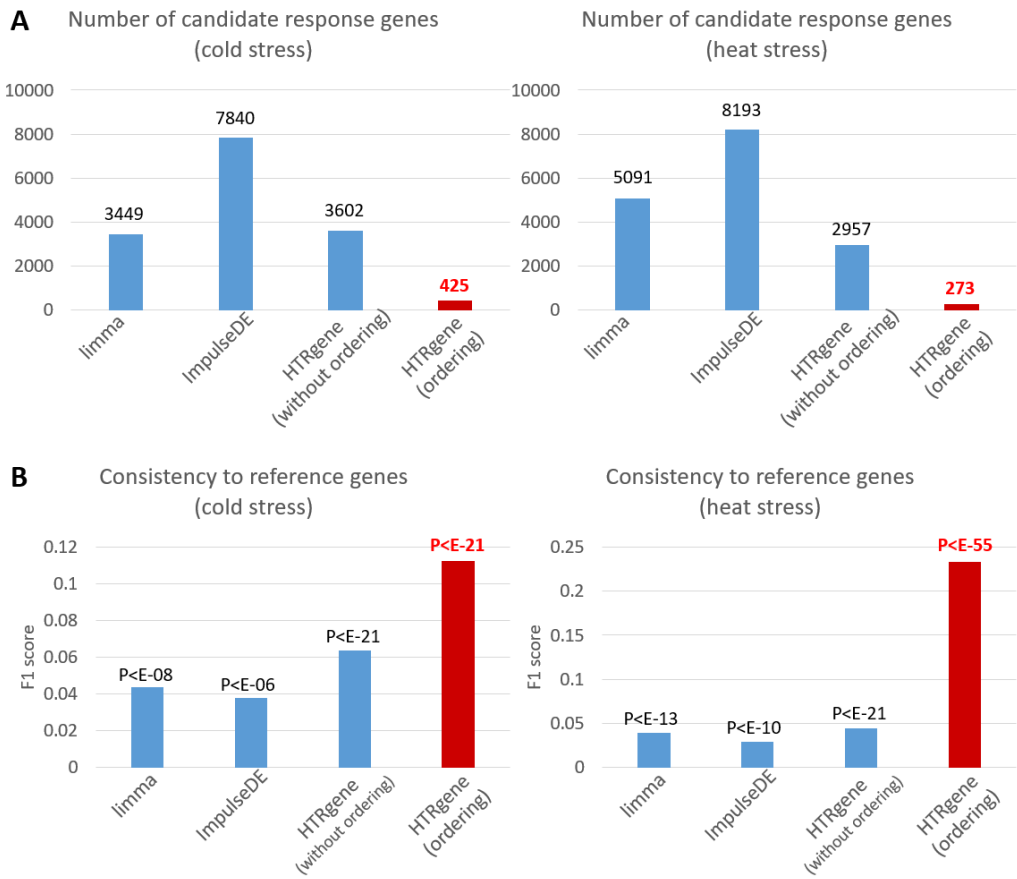

Figure 4.6. Candidate response gene detection results of limma, ImpulseDE, and HTRgene (without and with ordering). A) The number of candidate response genes. B) F1 scores as an estimate of the accuracy of detecting candidate response genes. F1 scores were calculated by measuring consistencies between the outputted candidate response genes and reference genes that are labeled as "response to cold" and "response to heat" in the TAIR database. The p-values were calculated by Fisher's exact test.

of DEGs and improve F1 score of DEG detection.

4.4 Summary

In this chapter, HTRgene was developed, which is a method to integrate multiple heterogeneous time-series gene expression data for the investigation of stress response signaling mechanisms in plant. Collecting all available datasets in the public domain is a way to increase the power of analysis in investigating the signaling mechanisms. The challenge is that the datasets are heterogeneous in terms of the time-domain (the number of time points and intervals are different) and phenotype-domain (the tissue of samples and the age of samples are different). In this study, response times are defined to integrate different datasets. With respect to response time points, genes are ordered to predict stress-responsive genes. In the process, clustering analysis is used to extract the meaningful time-domain characteristic, which is supported by many members of genes within the same cluster. In experiments using 28 and 24 time-series sample gene expression datasets under cold and heat stress, HTRgene successfully reproduced biological mechanisms of cold and heat stress in Arabidopsis.

Chapter 5

IDEA: Integrating Divisive and Ensemble-Agglomerate Hierarchical Clustering Framework with Density-Based Tree Search for Arbitrary Shape Data

Hierarchical clustering is one of the most widely used clustering methods for gene expression data analysis. Hierarchical clustering is easily understood because of its simple and intuitive framework. Hierarchical clustering successively divides clusters, which is called bottom-up or agglomerate clustering, or successively merges clusters, which is called top-down or divisive clustering. Although many new algorithms and strategies have been proposed, hierarchical clustering has remained based on the simple successive process as first developed. Then, it shows weakness for recent clustering issues such as distribution of data or a large number of data objects. The goal of this chapter is to develop a new hierarchical clustering algorithm that works on clustering analysis of genes.

5.1 Computational Problems

The input data in this chapter is a similarity matrix of genes, and the goal is to cluster the genes meaningfully. Figure 5.1 shows the challenges of this chapter. Although the distribution of data objects strongly affects the clustering analysis, the distribution of genes remains unknown. In addition, the size of input data is the square of number of genes, about 20,000 $\times$ 20,000, so the clustering on genes has to be efficient for computation. This chapter proposes an improved version of hierarchical clustering method to work on the arbitrarily distributed data with computational efficiency by combining effective recent clustering techniques such as network representation, phase shifting, and cost-optimization-based tree integration.

5.2 Evaluation Metric of Hierarchical Clustering Tree

This section introduces the cost function of hierarchical clustering tree, which operates theoretical fundamentals for the ensemble clustering algorithm of this study. Dasgupta [33], in 2016, proposed a cost function, which is the first evaluation function of the hierarchical clustering tree. The invention of evaluation function started a new paradigm in the development of hierarchical clustering methodologies [23, 27, 94, 113]. This is because researchers since have since been able to develop new hierarchical clustering methods as an optimization problem that is to minimize the cost function of Dasgupta. As preliminary knowledge, the cost function that Dasgupta proposed is as follows:

$$cost_G(T) = \sum_{ij \in E} w_{ij} |\mathbf{leaves}(T[i \vee j])|, \tag{5.1}$$

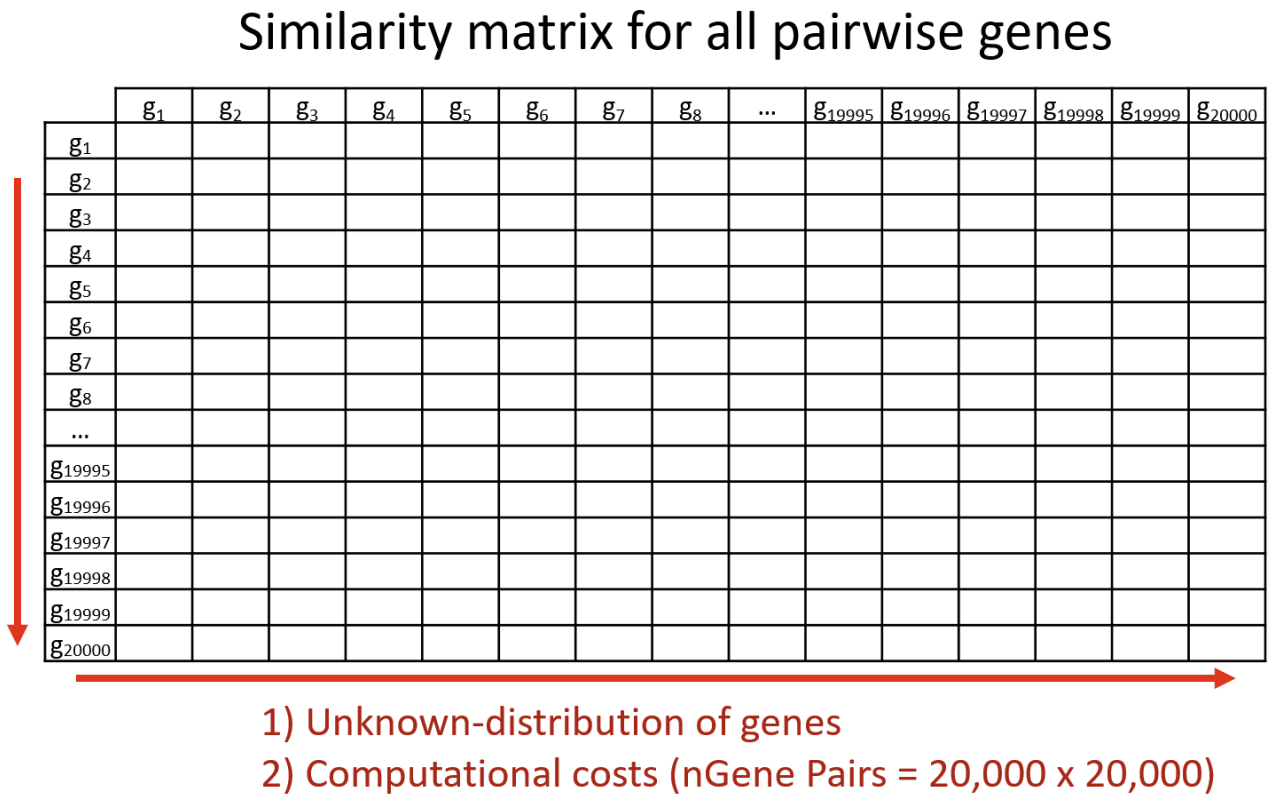

Figure 5.1. Clustering analysis on gene expression data. The input data is a similarity matrix between all pairwise genes (nGene Pairs = 20,000 × 20,000). We do not know the distribution of genes in the gene expression data, and the development of a clustering method working on arbitrarily distributed data is one of the challenges in clustering analysis on gene expression data. The development of a clustering method that is efficient for computation is another challenge.

where $G = (V, E, w)$ denotes a graph whose vertexes, edges, and weights of edges are V, E, and w, respectively. T denotes a tree (not necessarily binary). For a node u, $T[u]$ denotes the subtree rooted at u. For leaves $i, j \in V$, the expression $i \vee j$ denotes their lowest common ancestor in T. Then, $T[i \vee j]$ is the smallest subtree whose leaves include both i and j. $\textbf{leaves}(T[i \vee j])$ denotes the leaves of this subtree. Figure 5.2 shows an example of computing Dasgupta's cost function. Dasgupta also proposed a generalized version of cost function that uses a monotonically increasing function, f, to $|\textbf{leaves}(T[i \vee j])|$ in the basic cost function as follows:

$$cost_G(T) = \sum_{ij \in E} w_{ij} f(|\textbf{leaves}(T[i \vee j])|). \tag{5.2}$$

In the same paper, he also showed that computing the optimal cost function is *NP*-hard, but a top-down recursive partitioning hierarchical clustering heuristic using α_n-approximation sparsest cut has $O(\alpha_n \log_n) \times c(T^*)$, where $c(T^*)$ denotes a cost of optimal tree T^* (hereafter, $c(T^*)$ will be skipped for convenience). Then, he showed that Leighton-Rao sparsest cut [75]-based algorithm has $O(\log_n \log_n)$. Subsequently, the cost was reduced to $O(\sqrt{\log_n} \log_n)$ in a paper [23] by using Azora-Rao-Vazirani sparsest cut [12], and then to $O(\log_n)$ via spreading metrics method [113]. Then, another paper [27] proposed an algorithm with $O(1)$-approximation in special condition when the input graph G is generated from hierarchical stochastic block model (HSBM).

In addition, Moseley and Wang [94] proposed a revenue function $rev_G(T)$ that is a counterpart of Dasgupta's cost function:

$$rev_G(T) = \sum_{ij \in E} w_{ij} |\textbf{non-leaves}(T[i \vee j])|, \tag{5.3}$$

where $\textbf{non-leaves}(T[i \vee j])$ denotes leaves that do not belong to the subtree $T[i \vee j]$. Because $cost_G(T) + rev_G(T) = n \sum_{ij \in E} w_{ij}$, the cost function and revenue function

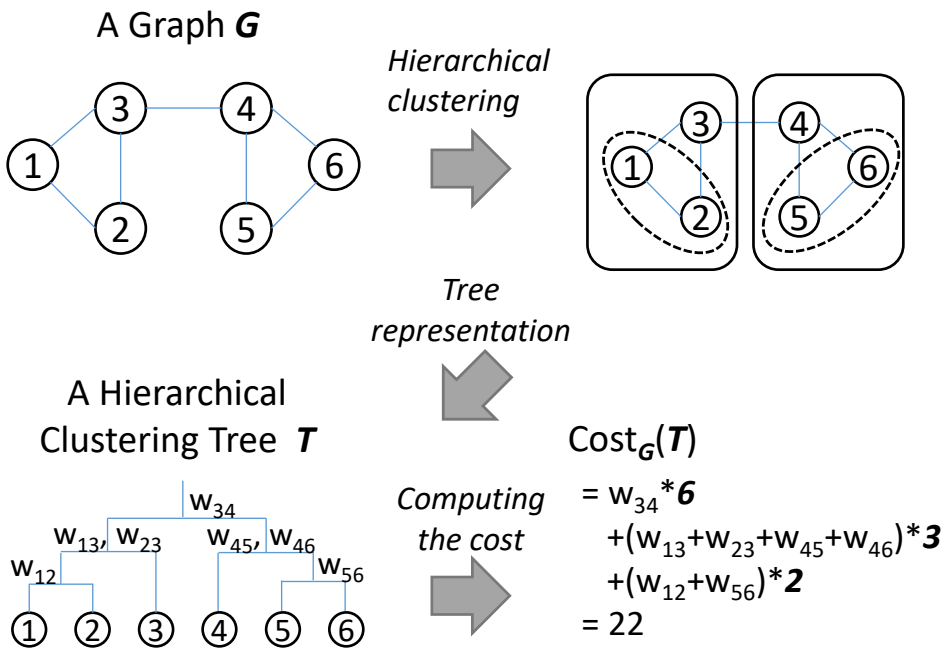

Figure 5.2. Example of Dasgupta's cost function. Let a hierarchical clustering method successively divides graph G, where all weights of connected edges are 1 ($w_{ij} = 1$), to generate a hierarchical tree. It divides the graph into $\{1,2,3\},\{4,5,6\}$ by disconnecting edge $\{w_{34}\}$ in the first round. Because $|\textbf{leaves}(T[\bullet])|$ is 6 in the first round, the cost increases by 1×6. In this way, the total cost becomes 22. If $|\textbf{leaves}(T[\bullet])|$ is thought of as a penalty term, then we can see that the first round has the biggest penalty, and the right strategy to reduce the cost is to cut off the least weighted edges in the first round.

have a dual relationship and the optimal solution to minimizing $cost_G(T)$ is the same as the optimal solution to maximizing $rev_G(T)$. They performed theoretical analysis for three clustering algorithms: average linkage, bisecting k-means, and divisive local search. Then, they analyzed the upper/lower bounds of approximation:

- for average linkage algorithm:

  $\frac{1}{3} rev_G(T^*) \geq rev_G(T_{\text{average linkage}}) \leq (\frac{1}{2} + \varepsilon) rev_G(T^*)$,

- for bisecting k-means algorithm:

  $rev_G(T_{\text{bisecting k-means}}) \leq \frac{1}{\Omega(\sqrt{n})} rev_G(T^*)$,

- for divisive local search algorithm:

  $\frac{n-6}{n-2} \frac{1}{3} rev_G(T^*) \leq rev_G(T_{\text{divisive local search}})$,

where T^* is the optimal tree.

5.3 Methods

This section first presents an ensemble clustering algorithm that integrates multiple trees by minimizing a cost function. Since the ensemble approach increases time complexity, this section continuously suggests a computationally efficient clustering framework, *Integrating Divisive and Ensemble-Agglomerate hierarchical clustering framework (IDEA)* [8], which uses advanced clustering techniques such as divisive-agglomerate hybridization and nearest neighbor-based graph construction.

5.3.1 Ensemble of Hierarchical Clustering Methods

Figure 5.3 illustrates an ensemble integration of hierarchical clustering method. Assume that there are L hierarchical clustering methods, each of which builds individual hierarchical clustering trees from a graph. Let $G = (V, E, w)$ be an undirected

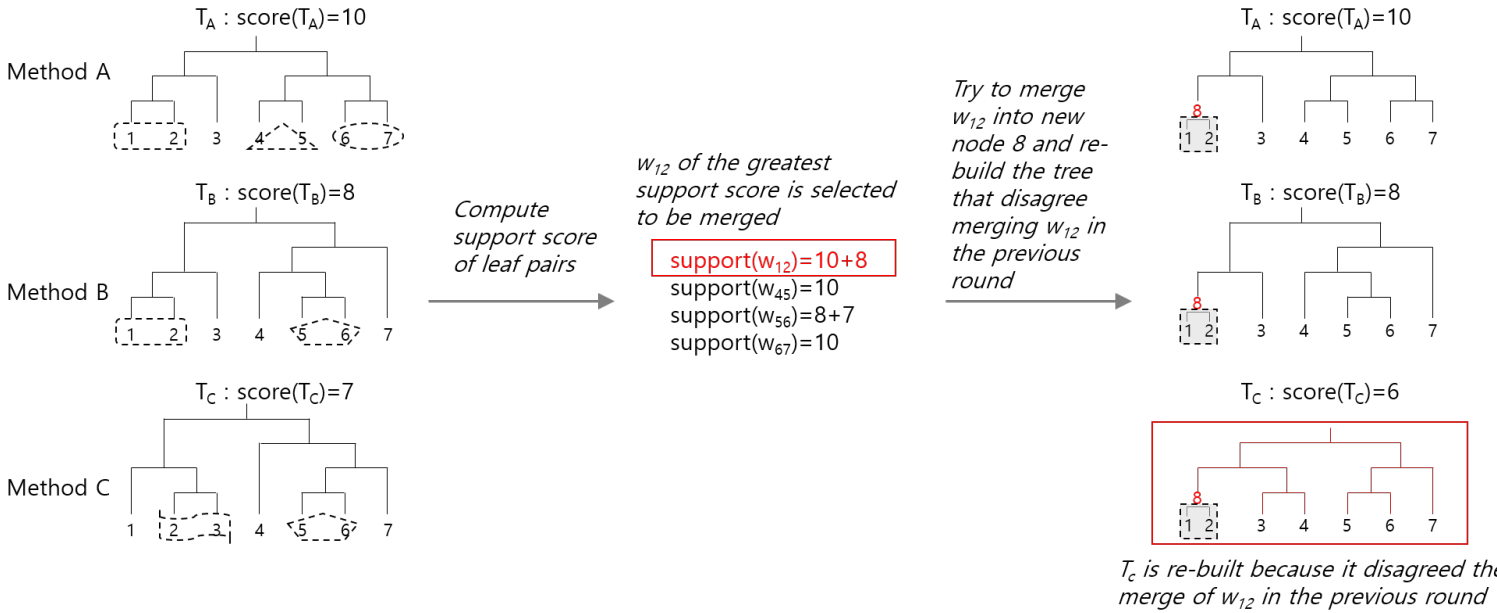

Figure 5.3. Example of ensemble tree integration. Let three divisive hierarchical clustering methods, A, B, and C, generate hierarchical trees T_A, T_B and T_C with scores 10, 8, and 7, where score of T_i is defined as $e^{-cost(T_i)}$. Then, the method collects all pairs of leaves that are merged in at least one tree and defines a support score for each pair as $\sum_{T_i \in T^\dagger} s(T_i)$ where $T^\dagger$ is a set of trees that merge the pair. It selects w_{12}, the pair with the greatest support score, forces to merge the pair, and updates all trees. Because T_A and T_B are already merged, there is no change to the tree. On the contrary, because T_C did not merge the pair in the previous round, T_C is re-built. In this way, all trees are integrated step-by step until one ensemble tree is produced.

and weighted graph, $T_1, T_2, \cdots, T_L$ be hierarchical clustering trees that are generated by L hierarchical clustering methods, and $s(T_\bullet) = e^{-c(T_\bullet)}$ be a score of tree (larger is better), where $c(T_\bullet)$ is Dasgupta's cost function. Then, the method collects all pairs of leaves that are merged in at least one tree and defines *a support score for each pair* as $\sum_{T_i \in T^\dagger} s(T_i)$, where $T^\dagger$ is a set of trees that merged the pair. It selects the pair with the biggest support score, and updates all trees by forcing the trees to merge the pair. If a tree already merged the pair, there is no change to those trees. On the contrary, if a tree under the current consideration does not merge the pair, it is re-built by the corresponding hierarchical clustering method from the graph where the pairs are forced to be merged. In this way, all trees are integrated step-by step, and only one ensemble tree is produced finally.

5.3.2 IDEA Hierarchical Clustering Framework

The designed ensemble method is not computationally efficient, since it is based on an adapting divisive-agglomerate hybrid approach and repeats weighted α-nearest neighbor graph construction. Thus, a hierarchical clustering framework, IDEA, was designed. IDEA takes a set of data points, X, with a metric of the user's choice (e.g., Euclidean distance or cosine similarity) defined on pairs of data points $\delta : X \times X \to \Re$ and three parameters, α, β, and k, as input and produces a hierarchical clustering tree, T, and a set of flat clusters, C. IDEA consists of pre-processing, main, and post-processing steps. IDEA is described in detail in Algorithm 18 and illustrated in Figure 5.4.

Algorithm 1: IDEA clustering framework

Input: a set of datapoints X and a metric (e.g. Euclidean distance) defined on pairs of datapoints $\delta : X \times X \to \Re$.

Parameters : α, the number of nearest neighbor,
β, the number of chunk,
k, the number of flat clusters.

Output: A hierarchical clustering tree T and a set of clusters C.

`// Pre-processing`

Construct a weighted α-nearest neighbor graph G from X and w;

Initialize a tree $T \leftarrow$ a tree where all nodes in G are direct children of $T.root$;

`// Main-step`

Set a set of *treeGraph* $S \leftarrow \{(T,G)\}$;

Set a target *treeGraph* $(\widehat{T},\widehat{G}) \leftarrow S.selectOne()$;

while $|S| < k$ **do**

  Partition $\widehat{G}$ into β chunks by recursive min-cut algorithm ; `/* i.e. Divisive stage */`

  Generate multiple hierarchical clustering trees of β chunks by applying multiple graph abstractions and linkages ; `/* i.e. Agglomerate stage */`

  Build a integrated tree of chunks by ensemble assembly of trees where each chunk has member nodes as direct children ; `/* i.e. Ensemble stage */`

  Replace $\widehat{T} \leftarrow$ the ensemble assembled tree and update the change of tree structure on T;

  Divide $\widehat{T}$ into two subtrees $\widehat{T}_L$ and $\widehat{T}_R$ and $\widehat{G}$ into two sub graphs $\widehat{G}_L$ and $\widehat{G}_R$ whose nodes are leaves of $\widehat{T}_L$ and $\widehat{T}_R$;

  Try to change memberships of the leaf nodes of $\widehat{T}_L$ and $\widehat{T}_R$ to the opposite subtree and accept the change if the cost is reduced;

  Compute separation scores of $\widehat{T}_L.score$ and $\widehat{T}_R.score$;

  Update $S \leftarrow S \cup \{(\widehat{T}_L,\widehat{G}_L),(\widehat{T}_R,\widehat{G}_R)\} \backslash (\widehat{T},\widehat{G})$;

  Set a target *treeGraph* $(\widehat{T},\widehat{G}) \leftarrow \arg\max_{(\widehat{T},\widehat{G}) \in S} \widehat{T}.score$;

end

`// Post-processing`

Complete a full binary tree T by performing average linkage hierarchical clustering on leaf nodes under the chunks;

set a set of k clusters $C \leftarrow \{\text{leavesOf}(\widehat{T})\}_{(\widehat{T},\widehat{G}) \in S}$;

return T and C

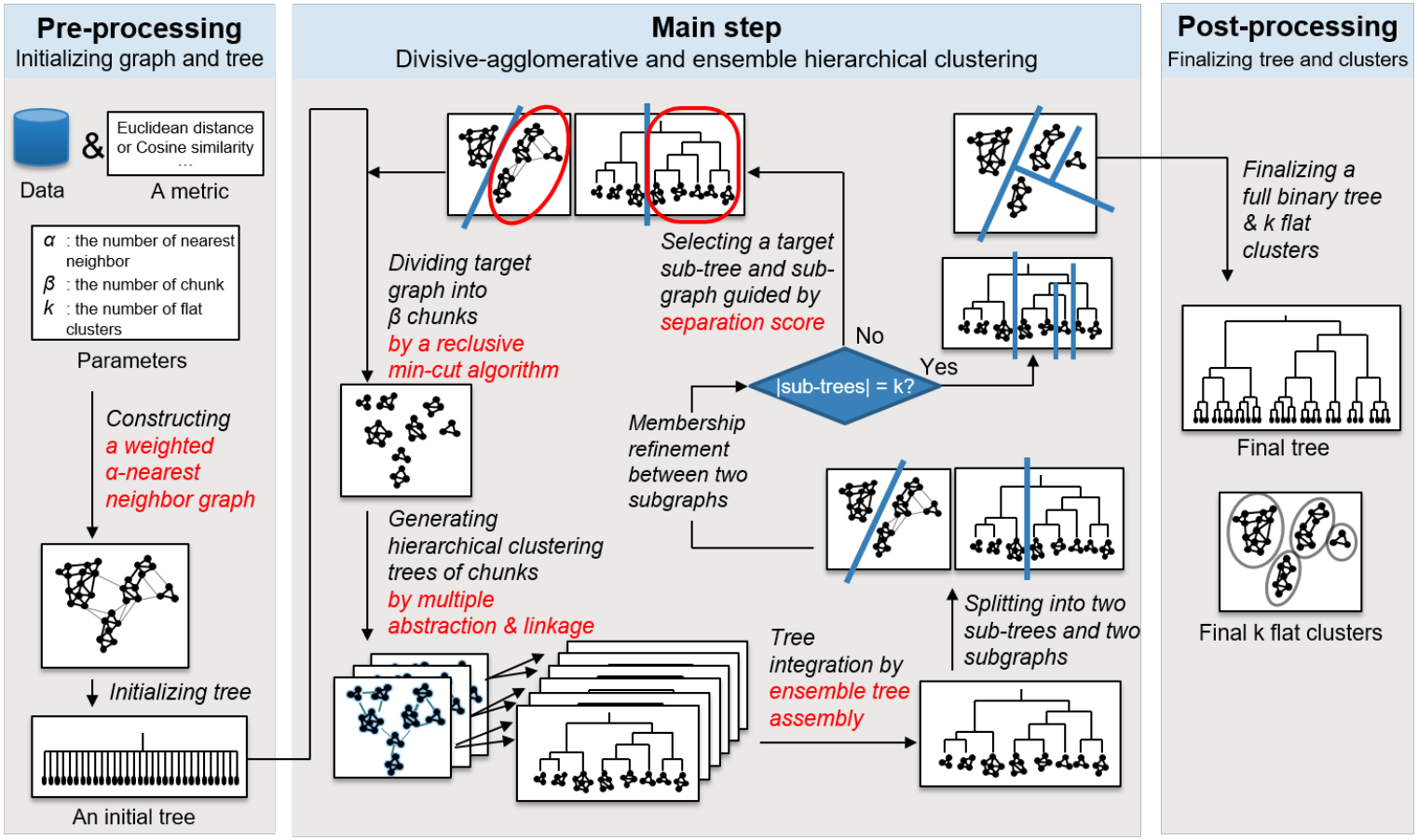

Figure 5.4. IDEA hierarchical clustering framework. From a set of data points, X, with a metric defined on pairs of data points $\delta : X \times X \rightarrow \Re$ and three parameters, α, β, and k, IDEA produces a hierarchical clustering tree, T, and a set of flat clusters, C, using nearest neighbor graph construction, divisive-agglomerate hybridization, and ensemble hierarchical clustering techniques.

Pre-processing: initializing input graph by constructing a weighted α-nearest neighbor graph

In the pre-processing step (LINE 1, 2), IDEA constructs a weighted α-nearest neighbor graph, $G = (V, E, w)$, from X and δ. It collects a set of α-nearest neighbor edges $E = \{e_{ij}\}$ s.t. $N = |E|$, the number of edges, $\leq |X| \times \alpha$. Then, $w(e_{ij})$, a weight of edge e_{ij}, is defined as follows:

$$\begin{aligned} w(e_{ij}) =& \lambda \times rank(e_{ij})/N \\ &+ (1-\lambda) \times e^{-(neighborness(i,j)+neighborness(j,i))}, \end{aligned} \tag{5.4}$$

where $rank : E \rightarrow \{N, N-1, \cdots, 1\}$ denotes a function to map the closest edge to N and the farthest edge to 1 according to the metric δ and $neighborness : \{i, j\} \rightarrow \{0, 1, \cdots, \alpha - 1\}$ denotes a function to map to 0 if node j is the first neighbor or to $\alpha - 1$ if the α-th neighbor with respect to node i. λ is the weight between *rank* and *neighborness* ($\lambda = 0.9$ by default). Converting a dataset into a weighted α-nearest neighbor graph has practical advantages in real-world clustering problems. It is because it reduces the number of edges from $|X|^2$ to $|X| \times \alpha$, which also reduces time complexity of the clustering algorithm so that it helps to speed up and complements increment of time complexity induced by an ensemble approach.

Main-step: integrating divisive and ensemble agglomerate hierarchical clustering

The main step (LINE 3∼14) employs an ensemble hierarchical clustering strategy to build trees with good Dasgupta's cost. To perform ensemble clustering on datasets with lower complexity, it uses a divisive-agglomerate hybridization method. It first performs division (dividing the graph G into β chunks), then agglomeration (merging the chunks successively). In the agglomerative stage, multiple clustering

methods are used to generate independent clustering trees; then, the ensemble method is used to integrate them and provide a single integrated tree. Then, it splits the tree into two subtrees and tunes membership of boundary nodes. Among the subtrees, it chooses one target subtree to be split next according to a separation score. It repeats this process recursively until the tree is separated into k subtrees. Below are the details on the division method, the ensemble agglomerate clustering method, the membership refinement method, and computation of a separation score.

Divisive stage: Clustering experiments to investigate how well graph partition methods divide a graph into a set of β chunk graphs showed that clustering accuracy improves as the size of chunks becomes even. In addition, although the data globally has an irregular shape, if it is separated into a suitable number of small chunks, the chunks have convex-shapes. These observations motivated a recursive graph partitioning algorithm that divides into a similar number of partial graphs of convex-shape using a fast min-cut based graph partitioning software `hMETIS` [71]. First, the algorithm divides a graph into two subgraphs using `hMETIS`. Then, it selects the largest subgraph and divides it into two subgraphs again. It repeats this process until it obtains β subgraphs (i.e., chunks). For each chunk, it divides the chunk into two half-chunks and tests if half chunks are closer to another chunk than to the opposite half chunk. If so, it shatters the chunk into single nodes. In addition, it tears off boundary single nodes from the chunks. Then, it reassigns the single nodes into the remaining chunks. It repeats this process τ times ($\tau = 5$ by default).

To determine the chunks to be shattered and the boundary single nodes to be reassigned, a modified metric of silhouette score [112] was developed. Let a chunk A be divided into two half-chunks, A_L, and A_R, and a node, $i \in A_L$, is closest to another chunk B among the set of chunks except A. Also, let $a(i)$ be the sum of weights from node i to the chunk A_R and $b(i)$ be the sum of weights from node i to the chunk B.

Then, the score of i, $s(i)$, is defined as below:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{5.5}$$

Note that $s(i)$ has the range $-1 \leq s(i) \leq 1$, where $s(i) > 0$ means that a node i in the chunk A_L is closer to a chunk A_R than chunk B. In addition, the score of node in chunk A_R can be computed the same way. If an average score of A, i.e., $\frac{\sum_{i \in A} s(i)}{|A|}$, is less than 0, the chunk is considered to be closer to the other chunk; thus, it is shattered. If a score of node $s(i)$ is less than -0.5, then the node is considered boundary single nodes then separated out from the chunk. Single nodes are reassigned to the chunks for which the sum of weights is greatest.

Agglomerate stage: Once a graph is divided into β chunk graphs, IDEA generates multiple clustering trees of the chunks. To generate multiple trees, a combinatorial method using different graph abstraction and linkage methods was developed. Graph abstraction means a process to construct a new chunk, graph from the original graph. In this process, the number of vertexes is reduced from the number of nodes $|V|$, to the number of chunks, β. There are several possible edge abstraction methods that define a new weight of edge for pairs of chunks. For instance, the two chunks, A and B, contain $|A|$ and $|B|$ nodes, then there are several possible weights of edges between the chunks (i.e., $\{w_{ij}\}_{i \in A, j \in B}$), and then a "*sum*" abstraction method reduces the weights of edges into a single representative weight value, i.e., $w_{AB} = \sum_{i \in A, j \in B} w_{ij}$. To do this, it uses six abstraction methods: sum, min, max, edge count, edge average (sum of weights divided by the number of edges), and all pair average (sum of weights divided by the number of all possible pairs).

With the abstracted graph, it applies four traditional agglomerate clustering methods: single, complete, weighted, and average linkage methods. When two chunks, A_L

and A_B, are merged into new chunks A, the weight of an edge between A and B is defined as follow:

$$\text{single linkage: } w_{AB} = \max\{w_{A_LB}, w_{A_RB}\} \tag{5.6}$$

$$\text{complete linkage: } w_{AB} = \min\{w_{A_LB}, w_{A_RB}\} \tag{5.7}$$

$$\text{average linkage: } w_{AB} = \frac{w_{A_LB}|A_L||B| + w_{A_RB}|A_R||B|}{(|A_L| + |A_R|) \times |B|} \tag{5.8}$$

$$\text{weighted linkage: } w_{AB} = \frac{w_{A_LB} + w_{A_RB}}{2} \tag{5.9}$$

Thus, it uses six graph abstraction methods and four linkage methods, and thus 24 different clustering methods total. Then, it assembles the 24 trees into a single tree by the ensemble integration of multiple trees that is presented before in this article.

Membership refinement: After divisive and ensemble agglomerative clustering, a constructed tree, T, is divided into two subtrees, T_L and T_R. For each subtree (for convenience, choose T_L) and each leaf node, $i \in T_L$, of the subtree, IDEA computes the sum of weights from the node to the leaves of subtrees. If node i is closer to the opposite subtree (i.e., $\sum_{j \in T_R} w_{ij} > \sum_{j \in T_L} w_{ij}$), it keeps the boundary nodes. For the boundary nodes, it tries changing memberships to the chunks of the opposite tree and accepts the change if the cost decreases.

Target tree search: After membership refinement, it searches for a set of subtrees and select a target subtree with a maximum separation score, and it is clustered

and divided into two subtrees. A separation score of subtree T is defined as follows:

$$separationScore(T) = \frac{\frac{\sum_{i \in A_L, j \in A_R} w_{ij}}{|A_L||A_R|}}{\frac{\sum_{i,j \in A_L} w_{ij}}{\frac{|A_L|(|A_L|-1)}{2}} \times \frac{\sum_{i,j \in A_R} w_{ij}}{\frac{|A_R|(|A_R|-1)}{2}}}, \tag{5.10}$$

where A_L and A_R denote the leaves of two subtrees divided from T.

Post-processing: finalizing a full binary tree and k flat clusters

In the post-processing step (LINE 16, 17), it produces final tree and k flat clusters as output. The final tree is a full binary tree constructed by performing average linkage for leaves under chunks. The final k flat clusters, C, are defined as the decedent nodes of k subtrees.

5.4 Experiments and Results

For the evaluation of performance, IDEA was compared with other clustering algorithms for 20 datasets. These datasets are generated and collected from https://github.com/deric/clustering-benchmark, and they are grouped into four types of subsets: convex-overlapped, non-convex, non-convex-noise, and complex biological datasets. The numbers of class and data points are summarized in Table 5.1.

The compared clustering algorithms consist of four baseline (single, complete, weighted, and average linkage), one cost optimization-based (linkage++ [27]), and two density-based hierarchical clustering (densityCut [36] and HDBSCAN [86]) methods. Among them, linkage++ is designed given a graph as input, and baseline methods can be performed on a graph or an adjacency distance matrix. HDBSCAN, on the other hand, accepts only adjacency distance matrix, and densityCut accepts data points as input. A weighted nearest neighbor graph is given as input to linkage++ and

Table 5.1. The statistics of clustering analysis datasets.

No.	The number of cluster	The number of data points	Source
A01	15	600	[128]
A02	31	3100	[128]
A03	35	5250	[68]
A04	15	5000	[43]
A05	15	5000	[43]
A06	15	5000	[43]
B01	3	312	[22]
B02	6	7236	[69]
B03	9	9208	[70]
B04	8	7677	[70]
B05	10	7676	self-generated
B06	9	7675	self-generated
C01	3	3673	[57]
C02	6	8000	[69]
C03	9	10000	[70]
C04	8	8000	[70]
C05	10	8000	self-generated
C06	9	8000	self-generated
D01	11	300	[104]
D02	5	5804	[6]

baseline methods. Baseline methods were also tested for adjacency distance matrix by using the `fastcluster` package [95]. Dasgupta's cost was measured to evaluate the result of hierarchical clustering trees, and adjusted rand index (ARI) [130] was measured to evaluate the result of k flat clusters for IDEA and other clustering methods. To facilitate comparison, the cost was normalized by dividing the cost of IDEA method. That is, a normalized cost of "2" means twice the cost of IDEA clustering. In addition, since densityCut software does not produce hierarchical clustering tree, it was excluded in the cost comparison.

5.4.1 Experiment on Convex and Overlapped Datasets

Dataset A consists of six datasets (A01 to A06) of convex-shape, but they overlap between clusters. A recent clustering comparison review study [133] showed that overlapping among clusters makes the cluster analysis challenging. Figure 5.5A showed IDEA had the minimum cost (1.000), followed by average-graph (1.0002), average-dist (1.204), complete-dist (1.830), and weighted-dist (1.989). In addition, IDEA produced the result of maximum ARI (0.857) as shown in Figure 5.5B, followed by densityCut-dist (0.855), average-graph (0.836), average-dist (0.769), and complete-dist (0.697). Note that the values in parentheses are the means of normalized cost, and the words "graph" and "dist" behind the names of algorithms represent a weighted nearest neighbor graph and an adjacency distance matrix that are given as input to the algorithms. Figure 5.5C is a visualization of results of seven representative clustering methods for dataset A.

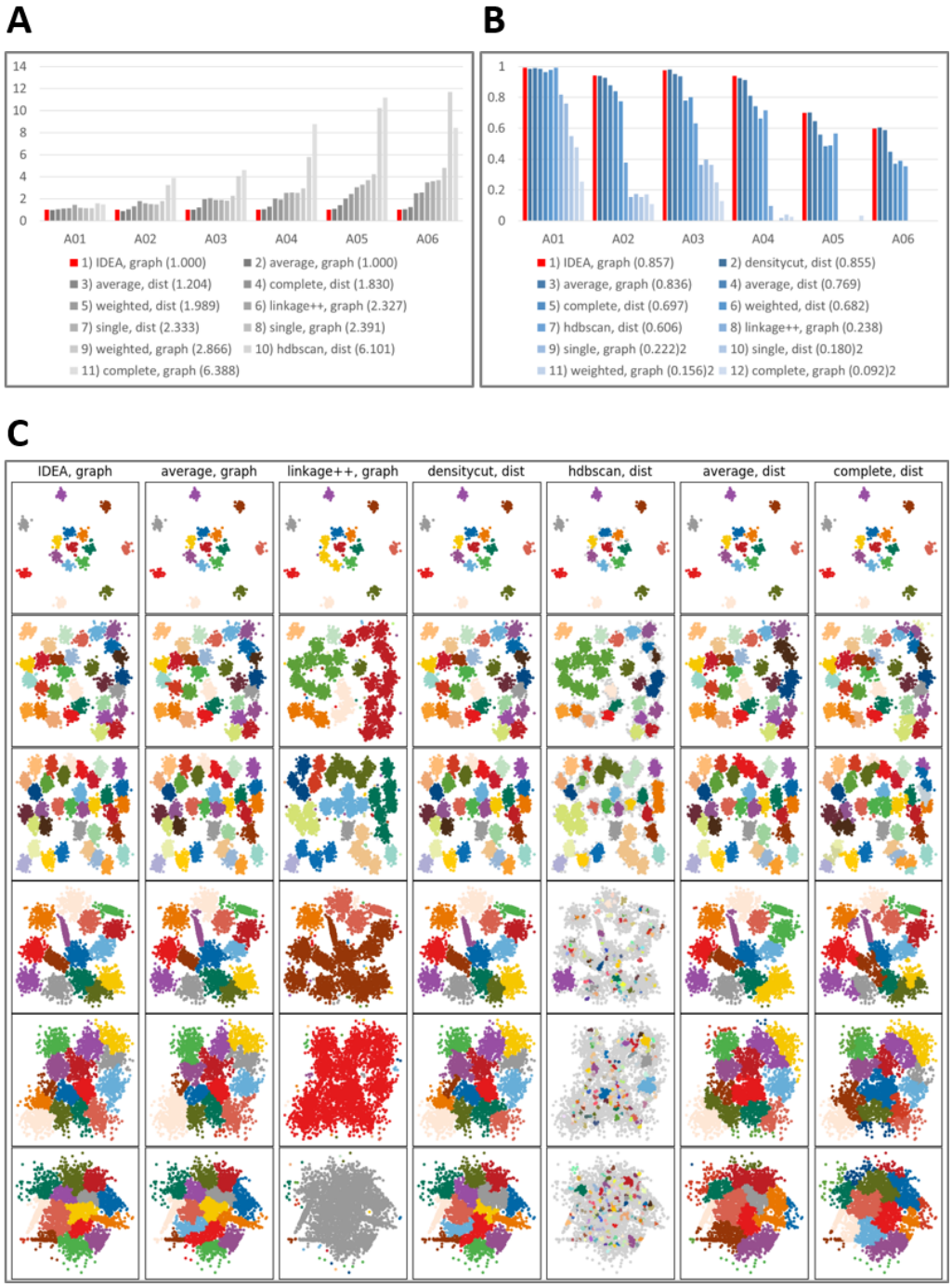

Figure 5.5. Cluster performance comparison on convex and overlapped dataset. Dasgupta's costs (A), ARI (B), and clustering result plots for six datasets and clustering algorithms. The words "graph" and "dist" behind the names of algorithms represent a weighted nearest neighbor graph and an adjacency distance matrix, which are given as input to the algorithms.

5.4.2 Experiment on Non-Convex-Shape Datasets

The second type of dataset consisted of six datasets (B01 to B06). The challenge is that clusters are of non-convex-shapes, as shown in a recent clustering comparison review [133]. In an experiment with these datasets, average-graph (0.976) and IDEA (1.000) were the first and second best in minimizing cost, as shown in Figure 5.6A, followed by average-dist (1.587), single-graph (2.040), and single-dist (2.152). In addition, IDEA showed the maximum ARI (1.000), as shown in Figure 5.6B, followed by average-graph (0.9999), densityCut-dist (0.973), HDBSCAN-dist (0.960), and single-graph (0.956). Figure 5.6C is a visualization of results of seven representative clustering methods for dataset B.

5.4.3 Experiment on Non-Convex-Shape and Noisy Datasets

Another experiment was performed with six datasets (C01 to C06) of non-convex-shape with noise data points. Because of the noise data points, this dataset is more difficult than those used in the previous experiments. In an experiment with these datasets, IDEA outperformed all competing clustering methods in minimizing cost function (1.000), as shown in Figure 5.7A, followed by average-graph (1.021), average-dist (1.521), single-graph (2.855), and complete-dist (2.919). IDEA produced the result of maximum ARI (0.980), as shown in Figure 5.7B, followed by average-graph (0.920), densityCut-dist (0.812), HDBSCAN-dist (0.738), and average-dist (0.537). Figure 5.7C is an visualization of results of seven representative clustering methods for dataset C.

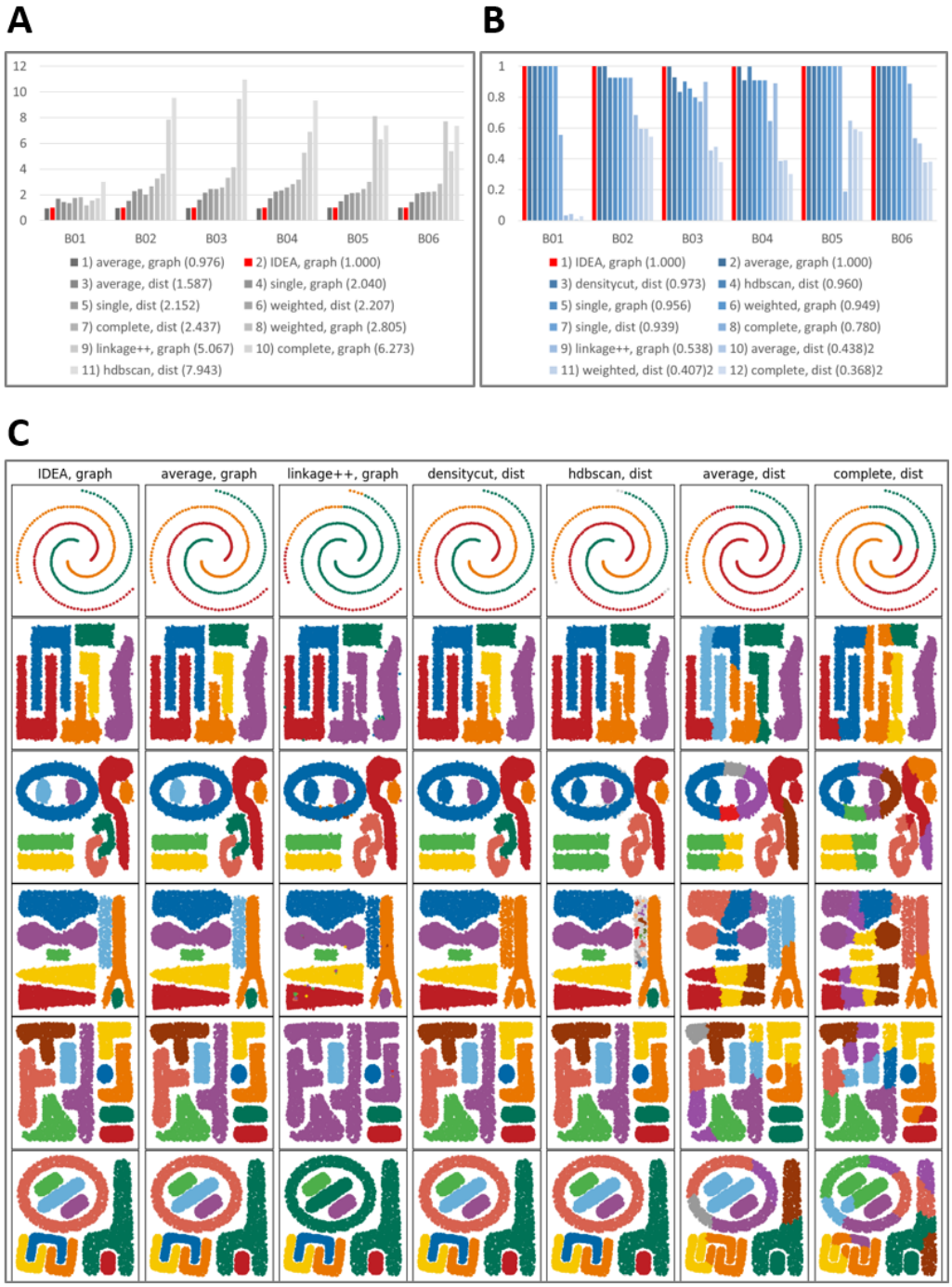

Figure 5.6. Cluster performance comparison on non-convex dataset. Dasgupta's costs (A), ARI (B), and clustering result plots for six datasets and clustering algorithms. The words "graph" and "dist" behind the names of algorithms represent a weighted nearest neighbor graph and an adjacency distance matrix, which are given as input to the algorithms.

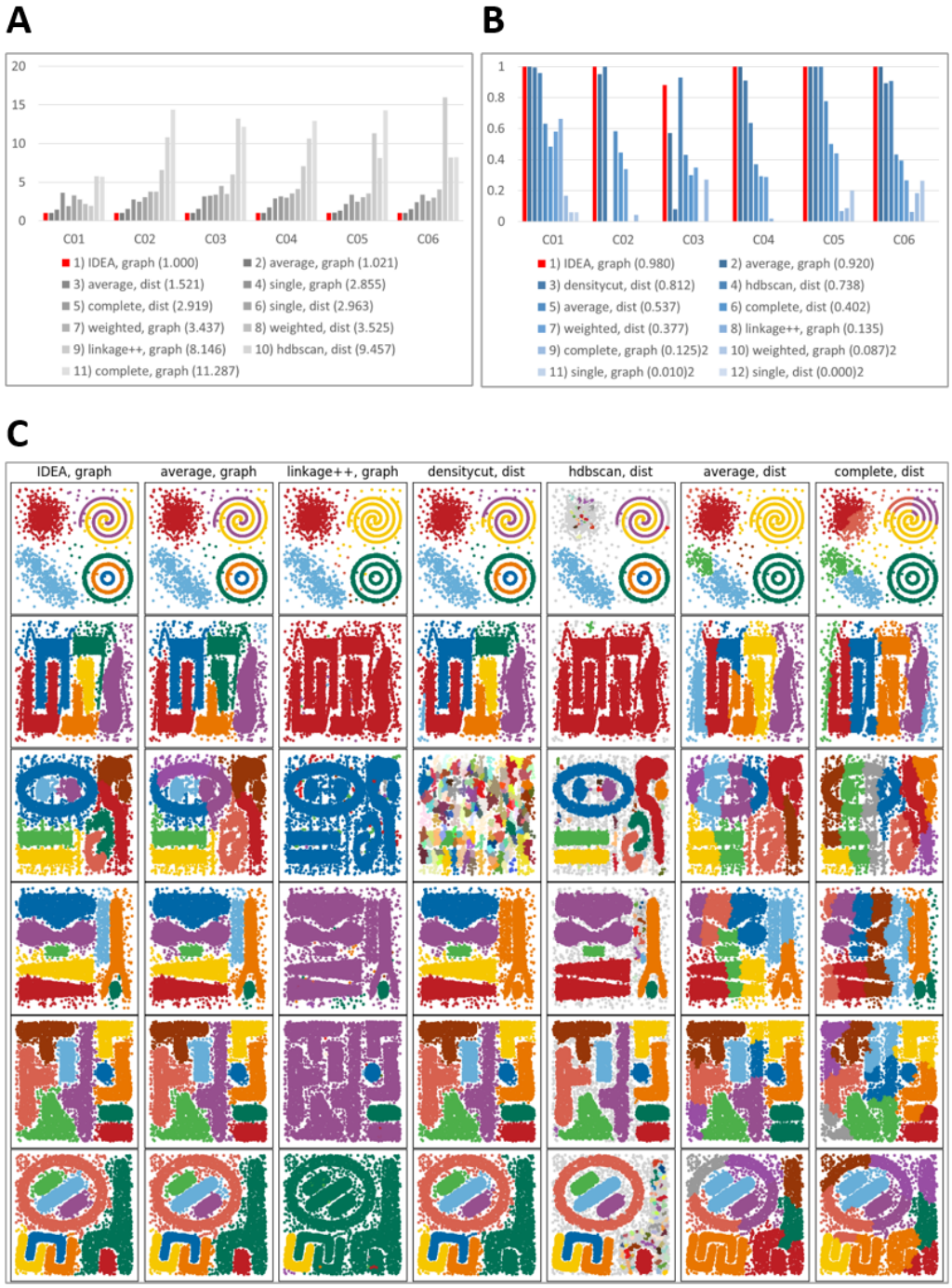

Figure 5.7. Cluster performance comparison on non-convex and noise dataset. Dasgupta's costs (A) and ARI (B) for six datasets and clustering algorithms. Dasgupta's costs (A), ARI (B), and clustering result plots for six datasets and clustering algorithms. The words "graph" and "dist" behind the names of algorithms represent a weighted nearest neighbor graph and an adjacency distance matrix, which are given as input to the algorithms.

Table 5.2. Evaluation of clustering methods for dataset D (complex biological dataset).

Rank	D01 Cost		D01 ARI	
1	**IDEA, graph**	1.000	HDBSCAN, dist	0.934
2	average, graph	1.005	**IDEA, graph**	0.926
3	single, dist	1.118	average, graph	0.888
4	single, graph	1.126	densityCut, dist	0.885
5	weighted, graph	1.157	single, dist	0.879
6	complete, graph	1.286	average, dist	0.777
7	average, dist	1.505	weighted, dist	0.745
8	weighted, dist	1.988	single, graph	0.603
9	HDBSCAN, dist	2.343	weighted, graph	0.597
10	complete, dist	2.528	complete, graph	0.557
11			complete, dist	0.535

Rank	D02 Cost		D02 ARI	
1	**IDEA, graph**	1.000	**IDEA, graph**	0.582
2	average, graph	1.017	average, graph	0.579
3	single, graph	1.937	complete, graph	0.041
4	weighted, graph	6.825	HDBSCAN, dist	0.001
5	complete, graph	7.439	single, graph	0.000
6	HDBSCAN, dist	9.217	weighted, graph	0.000
7	complete, dist	13.310	average, dist	0.000
8	weighted, dist	13.409	single, dist	0.000
9	average, dist	13.515	weighted, dist	0.000
10	single, dist	13.610	complete, dist	-0.001

5.4.4 Experiment on Complex Biological Datasets

IDEA was additionally evaluated using two complex biological datasets (D01 and D02). IDEA shows the best performance on cost for D01, cost and ARI for D02, and the second best on ARI for D01, as shown in Table 5.2. These results show IDEA performs well on real-world data as long as 2D datasets.

5.5 Summary

This chapter presented a computationally efficient hierarchical clustering algorithm, called integrating divisive and ensemble-agglomerate (IDEA), for clustering on arbitrarily distributed data. Hierarchical clustering can generate numerous clustering trees, but it is difficult to determine that a tree is better than other trees. Recently, Dasgupta' cost function was developed as a metric to evaluate clustering trees. IDEA uses the cost function to integrate multiple trees generated by several hierarchical clustering methods and produces an integrated tree with reduced cost. IDEA also includes a top-down and bottom-up strategy to reduce the complexity of clustering tree enumeration. In experiments using arbitrary shape datasets on 2D dimension, IDEA performed better in minimizing Dasgupta's cost and improving accuracy (adjusted rand index) over existing cost-minimization-based, and density-based hierarchical clustering methods in experiments using arbitrary shape datasets. It also showed better performance in complex genetic datasets where the distribution of data was not yet known. This chapter presented a computationally efficient hierarchical clustering algorithm, called integrating divisive and ensemble-agglomerate (IDEA), for clustering on arbitrarily distributed data. Hierarchical clustering can generate numerous clustering trees, but it is difficult to determine that a tree is better than other trees. Recently, Dasgupta' cost function was developed as a metric to evaluate clustering trees.

IDEA uses the cost function to integrate multiple trees generated by several hierarchical clustering methods and produces an integrated tree with reduced cost. IDEA also includes a top-down and bottom-up strategy to reduce the complexity of clustering tree enumeration. In experiments using arbitrary shape datasets on 2D, IDEA performed better in minimizing Dasgupta's cost and improving accuracy (ARI) over existing cost-minimization-based and density-based hierarchical clustering methods in experiments using arbitrary shape datasets. It also showed better performance in complex genetic datasets where the distribution of data was not yet known.

Chapter 6

Conclusion

The goal of this dissertation was to solve the recent computational problems in gene expression analysis. The fundamental problem of gene expression data analysis is modeling a highly complex system with small-sample-size data. This dissertation addressed the challenges based on clustering and integrated analysis. This dissertation developed three practical methods or algorithms for analyzing gene expression data to investigate the cellular response of plants under environmental stress:

- **RiceTFnetwork**: a computational framework to integrate a large-scale repository dataset based on network representation for the analysis of extremely small-sample-size gene expression data
- **HTRgene**: a computational framework to integrate a multiple time-series gene expression data where time-domain and phenotype-domain are heterogeneous
- **IDEA**: a hierarchical clustering algorithm that works on arbitrarily distributed data.

This dissertation, in Chapter 3, proposed a comprehensive analysis framework, RiceTFnetwork, that involves construction, integration, and clustering of gene networks, for the analysis of two sample gene expression datasets. By constructing a template network using a large-scale public gene expression dataset and then integrating with the network that was generated from experimental data, the computational framework successively analyzed the gene expression data and revealed the

underlying drought-resistance mechanism of a GM rice species.

HTRgene was proposed in Chapter 4, which is a method to integrate multiple heterogeneous time-series gene expression data to investigate stress response signaling genes. HTRgene used a response-time-based approach to integrating multiple heterogeneous data. HTRgene clusters genes based on co-expression patterns, detects the response times of the gene clusters, and then determines a response order of the clusters across multiple samples to produce response order preserving DEGs. HTRgene successfully reproduced biological mechanisms of cold and heat stress in Arabidopsis in the analysis of 28 and 24 time-series sample gene expression datasets under cold and heat stress.

IDEA was developed in Chapter 5, which is a computationally efficient hierarchical clustering algorithm that works on arbitrarily distributed data. IDEA constructed a weighted nearest neighbor graph and divided the graph into chunks. Then, it performed an ensemble hierarchical clustering by integrating multiple trees generated by several hierarchical clustering methods and produced an integrated tree with reduced cost. The IDEA clustering method showed better performance in minimizing Dasgupta's cost and improving accuracy (ARI) over existing cost-minimization-based, and density-based hierarchical clustering methods in experiments using arbitrarily distributed datasets and complex genetic datasets.

In conclusion, this study has studied an effective method to model a highly complex system with small-sample-size data. Clustering genes into subgroups reduced the number of features, then also the complexity of modeling. It thus increased the reliability of modeling and the interpretability of gene expression data analysis. Network representation was an effective method to integrate domain knowledge/data and the experimental data. Constructing a template network using biological knowledge and the public domain data followed by incorporating the template network with the

experimental data produced a well-established network that revealed the underlying characteristics of data. Integrating heterogeneous time-series data with a novel concept, response time, was a solution of the heterogeneous structure of time domain and the individual variance of samples. Nearest neighbor graph, phase shifting, and ensemble tree integration were also effective for clustering on arbitrarily distributed data.

With the method, this dissertation successfully analyzed time-series gene expression datasets, producing results that were helpful in characterizing underlying drought resistance mechanism of a GM rice species, and identified stress response signaling genes for cold and heat stress in Arabidopsis, which was partially consistent with current biological knowledge.

Bibliography

[1] Shigeo Abe. "Feature selection and extraction". In *Support Vector Machines for Pattern Classification*, pages 331–341. Springer, 2010.

[2] Charu C. Aggarwal, Joel L. Wolf, et al. "Fast algorithms for projected clustering". In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 61–72, 1999.

[3] Rakesh Agrawal, Johannes Gehrke, et al. "Automatic subspace clustering of high dimensional data for data mining applications". In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[4] Hongryul Ahn, Heejoon Chae, et al. "Integration of heterogeneous time series gene expression data by clustering on time dimension". In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 332–335. IEEE, 2017.

[5] Hongryul Ahn, Kyuri Jo, et al. "PropaNet: time-varying condition-specific transcriptional network construction by network propagation". *Frontiers in Plant Science*, 2019. In revision.

[6] Hongryul Ahn, Inuk Jung, et al. "Transcriptional network analysis reveals drought resistance mechanisms of AP2/ERF transgenic rice". *Frontiers in Plant Science*, 8:1044, 2017.

[7] Hongryul Ahn, Inuk Jung, et al. "HTRgene: integrating multiple heterogeneous time-series data to investigate cold and heat stress response signaling genes in Arabidopsis". In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018.

[8] Hongryul Ahn, Inuk Jung, et al. "IDEA: Integrating Divisive and Ensemble-Agglomerate hierarchical clustering framework with density-based tree search for arbitrary shape data". *In preparation*, 2019.

[9] Bruce Alberts, Alexander Johnson, et al. *"Molecular Biology of the Cell"*. W. W. Norton & Company, 6th edition, 2014.

[10] Madana M. R. Ambavaram, Supratim Basu, et al. "Coordinated regulation of photosynthesis in rice increases yield and tolerance to environmental stress". *Nature Communications*, 5:5302, 2014.

[11] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data". *Genome Biology*, 11(10):R106, 2010.

[12] Sanjeev Arora, Satish Rao, and Umesh Vazirani. "Expander flows, geometric embeddings and graph partitioning". *Journal of the ACM*, 56(2):5, 2009.

[13] Tanya Barrett, Stephen E. Wilhite, et al. "NCBI GEO: archive for functional genomics data sets–update". *Nucleic Acids Research*, 41(D1):D991–D995, 2013.

[14] Vicente R. Barros, Christopher B. Field, et al. "Climate change 2014: impacts, adaptation, and vulnerability. Part B: regional aspects", 2014.

[15] Amir Ben-Dor and Zohar Yakhini. "Clustering gene expression patterns". In *Proceedings of the Third Annual International Conference on Computational Molecular Biology*, pages 33–42, 1999.

[16] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

[17] Pavel Berkhin. "A survey of clustering data mining techniques". In *Grouping Multidimensional Data*, pages 25–71. Springer, 2006.

[18] James C Bezdek, Robert Ehrlich, and William Full. "FCM: the fuzzy c-means clustering algorithm". *Computers & Geosciences*, 10(2-3):191–203, 1984.

[19] Vincent D. Blondel, Jean-Loup Guillaume, et al. "Fast unfolding of communities in large networks". *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[20] Christian Buchta, Martin Kober, et al. "Spherical k-means clustering". *Journal of Statistical Software*, 50(10):1–22, 2012.

[21] Igor V Cadez, Padhraic Smyth, and Heikki Mannila. "Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction". In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 37–46, 2001.

[22] Hong Chang and Dit-Yan Yeung. "Robust path-based spectral clustering". *Pattern Recognition*, 41(1):191–203, 2008.

[23] Moses Charikar and Vaggos Chatziafratis. "Approximate hierarchical clustering via sparsest cut and spreading metrics". In *Proceedings of the Twenty-Eighth Annual ACM SIAM Symposium on Discrete Algorithms*, pages 841–854, 2017.

[24] Ligang Chen, Yu Song, et al. "The role of WRKY transcription factors in plant abiotic stresses". *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1819(2):120–128, 2012.

[25] Shihyen Chen, Bin Ma, and Kaizhong Zhang. "On the similarity metric and the distance metric". *Theoretical Computer Science*, 410(24-25):2365–2376, 2009.

[26] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks". *Physical Review E*, 70(6):066111, 2004.

[27] Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. "Hierarchical clustering beyond the worst-case". In *Advances in Neural Information Processing Systems*, pages 6202–6210, 2017.

[28] Wikipedia contributors. "Wikipedia, the free encyclopedia", 2011. `https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set`.

[29] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. "Data preparation for mining world wide web browsing patterns". *Knowledge and Information Systems*, 1(1):5–32, 1999.

[30] Francis H. C. Crick. "On protein synthesis". In *Symposia of the Society for Experimental Biology*, volume 12, page 8, 1958.

[31] Gabor Csardi and Tamas Nepusz. “The igraph software package for complex network research”. *InterJournal*, Complex Systems:1695, 2006.

[32] Cuthbert Daniel and Fred S. Wood. *“Fitting equations to data: computer analysis of multifactor data”*. Wiley, 2nd edition, 1980.

[33] Sanjoy Dasgupta. “A cost function for similarity-based hierarchical clustering”. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 118–127, 2016.

[34] Rajesh N. Dave and Kurra Bhaswan. “Adaptive fuzzy c-shells clustering and detection of ellipses”. *IEEE Transactions on Neural Networks*, 3(5):643–662, 1992.

[35] Inderjit S. Dhillon, James Fan, and Yuqiang Guan. “Efficient clustering of very large document collections”. In *Data Mining for Scientific and Engineering Applications*, pages 357–381. Springer, 2001.

[36] Jiarui Ding, Sohrab Shah, and Anne Condon. “densityCut: an efficient and versatile topological approach for automatic clustering of biological data”. *Bioinformatics*, 32(17):2567–2576, 2016.

[37] Maria E. Eriksson and Alex A. R. Webb. “Plant cell responses to cold are all about timing”. *Current Opinion in Plant Biology*, 14(6):731–737, 2011.

[38] Martin Ester, Alexander Frommelt, et al. “Spatial data mining: database primitives, algorithms and efficient DBMS support”. *Data Mining and Knowledge Discovery*, 4(2-3):193–216, 2000.

[39] Brian S. Everitt, Sabine Landau, and Morven Leese. *“Cluster analysis”*. Arnold, 4th edition, 2001.

[40] Yujie Fang, Jun You, et al. “Systematic sequence analysis and identification of tissue-specific or stress-responsive genes of NAC transcription factor family in rice”. *Molecular Genetics and Genomics*, 280(6):547–563, 2008.

[41] Stephen P. Ficklin and Frank A. Feltus. “Gene co-expression network alignment and conservation of gene modules between two grass species: maize and rice”. *Plant Physiology*, pages pp–111, 2011.

[42] Andrew Foss, Weinan Wang, and Osmar R. Zaïane. "A non-parametric approach to web log analysis". In *Proceedings of the Workshop on Web Mining, the First SIAM Conference on Data Mining*, pages 41–50, 2001.

[43] Pasi Fränti and Olli Virmajoki. "Iterative shrinking method for clustering problems". *Pattern Recognition*, 39(5):761–775, 2006.

[44] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *"Data clustering: theory, algorithms, and applications"*, volume 20. Society for Industrial and Applied Mathematics, 2007.

[45] Laurent Gautier, Leslie Cope, et al. "affy—analysis of Affymetrix GeneChip data at the probe level". *Bioinformatics*, 20(3):307–315, 2004.

[46] Sarvajeet Singh Gill and Narendra Tuteja. "Reactive oxygen species and antioxidant machinery in abiotic stress tolerance in crop plants". *Plant Physiology and Biochemistry*, 48(12):909–930, 2010.

[47] Michelle Girvan and Mark E. J. Newman. "Community structure in social and biological networks". *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[48] Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset. "Performance of modularity maximization in practical contexts". *Physical Review E*, 81(4):046106, 2010.

[49] John C. Gower. "A general coefficient of similarity and some of its properties". *Biometrics*, pages 857–871, 1971.

[50] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. "On clustering validation techniques". *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[51] Kazuki Hamada, Kohei Hongo, et al. "OryzaExpress: an integrated database of gene expression networks and omics annotations in rice". *Plant and Cell Physiology*, 52(2):220–229, 2010.

[52] Jiawei Han, Micheline Kamber, and Jian Pei. *"Data mining: concepts and techniques"*. Elsevier, 3rd edition, 2011.

[53] Jeffrey Heer and Ed H. Chi. "Identification of web user traffic composition using multi-modal clustering and information scent". In *Proceedings of the Workshop on Web Mining, the First SIAM Conference on Data Mining*, pages 51–58, 2001.

[54] Honghong Hu, Mingqiu Dai, et al. "Overexpressing a NAM, ATAF, and CUC (NAC) transcription factor enhances drought resistance and salt tolerance in rice". *Proceedings of the National Academy of Sciences*, 103(35):12987–12992, 2006.

[55] Miho Ikeda, Nobutaka Mitsuda, and Masaru Ohme-Takagi. "Arabidopsis HsfB1 and HsfB2b act as repressors of the expression of heat-inducible *Hsfs* but positively regulate the acquired thermotolerance". *Plant Physiology*, 157(3):1243–1254, 2011.

[56] Pierre Jacob, Heribert Hirt, and Abdelhafid Bendahmane. "The heat shock protein/chaperone network and multiple stress resistance". *Plant Biotechnology Journal*, 2016.

[57] Anil K. Jain. "Data clustering: 50 years beyond k-means". *Pattern Recognition Letters*, 31(8):651–666, 2010.

[58] Anil K. Jain and Richard C. Dubes. *"Algorithms for clustering data"*. Prentice-Hall, Inc., 1988.

[59] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review". *ACM Computing Surveys*, 31(3):264–323, 1999.

[60] Pablo A. Jaskowiak, Ricardo JGB Campello, and Ivan G. Costa. "On the selection of appropriate distances for gene expression data clustering". *BMC bioinformatics*, 15(2), 2014.

[61] Jin Jeon and Jungmook Kim. "Cold stress signaling networks in Arabidopsis". *Journal of Plant Biology*, 56(2):69–76, 2013.

[62] Jin Seo Jeong, Youn Shic Kim, et al. "Root-specific expression of *OsNAC10* improves drought tolerance and grain yield in rice under field drought conditions". *Plant Physiology*, 153(1):185–197, 2010.

[63] Jinpu Jin, He Zhang, et al. "PlantTFDB 3.0: a portal for the functional and evolutionary study of plant transcription factors". *Nucleic Acids Research*, 42(D1):D1182–D1187, 2013.

[64] Inuk Jung, Kyuri Jo, et al. "TimesVector: a vectorized clustering approach to the analysis of time series transcriptome data from multiple phenotypes". *Bioinformatics*, page btw780, 2017.

[65] Dongwon Kang, Hongryul Ahn, et al. "Formulation of a problem for the integrated analysis of heterogeneous time-series gene expression data and cold stress response gene set analysis in arabidopsis". In *Proceedings of the Korean Information Science Society Conference*, pages 648–650, 2015.

[66] Dongwon Kang, Hongryul Ahn, et al. "Identifying stress-related genes and predicting stress types in Arabidopsis using logical correlation layer and CMCL loss through time-series data". In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018.

[67] Hyejin Kang, Hongryul Ahn, et al. "mirTime: identifying condition-specific targets of microRNA in time-series transcript data using Gaussian process model and spherical vector clustering". *Bioinformatics*, 2019. In revision.

[68] Ismo Kärkkäinen and Pasi Fränti. *"Dynamic local search algorithm for the clustering problem"*. University of Joensuu, 2002.

[69] George Karypis. "CLUTO-a clustering toolkit". Technical report, Minnesota University Minneapolis Department of Computer Science, 2002.

[70] George Karypis, Eui-Hong Han, and Vipin Kumar. "Chameleon: hierarchical clustering using dynamic modeling". *Computer*, 32(8):68–75, 1999.

[71] George Karypis and Vipin Kumar. "A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices". *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 1998.

[72] Nikolay Kolesnikov, Emma Hastings, et al. "Arrayexpress update–simplifying data submissions". *Nucleic Acids Research*, page gku1057, 2014.

[73] Joel A. Kreps, Yajun Wu, et al. "Transcriptome changes for Arabidopsis in response to salt, osmotic, and cold stress". *Plant Physiology*, 130(4):2129–2141, 2002.

[74] Philippe Lamesch, Tanya Z Berardini, et al. "The Arabidopsis information resource (TAIR): improved gene annotation and new tools". *Nucleic Acids Research*, 40(D1):D1202–D1210, 2011.

[75] Tom Leighton and Satish Rao. "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms". *Journal of the ACM*, 46(6):787–832, 1999.

[76] Marie-Jeanne Lesot, Maria Rifqi, and Hamid Benhadda. "Similarity measures for binary and numerical data: a survey". *International Journal of Knowledge Engineering and Soft Data Paradigms*, 1(1):63, 2009.

[77] Søren Lindemose, Charlotte O'Shea, et al. "Structure, function and networks of transcription factors involved in abiotic stress responses". *International Journal of Molecular Sciences*, 14(3):5842–5878, 2013.

[78] Hsiang-chin Liu and Yee-yung Charng. "Acquired thermotolerance independent of heat shock factor A1 (HsfA1), the master regulator of the heat stress response". *Plant Signaling & Behavior*, 7(5):547–550, 2012.

[79] Xin Lu, Vipul V. Jain, et al. "Hubs in biological interaction networks exhibit low changes in expression in experimental asthma". *Molecular systems biology*, 3(1):98, 2007.

[80] Christopher Bishop M. *"Pattern recognition and machine learning"*. Springer-Verlag, 2006.

[81] Sara C. Madeira and Arlindo L. Oliveira. "Biclustering algorithms for biological data analysis: a survey". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[82] Linyong Mao, John L. Van Hemert, et al. "Arabidopsis gene co-expression network and its functional modules". *BMC Bioinformatics*, 10(1):346, 2009.

[83] Daniel Marbach, James C. Costello, et al. "Wisdom of crowds for robust gene network inference". *Nature Methods*, 9(8):796, 2012.

[84] Kyonoshin Maruyama, Yoh Sakuma, et al. "Identification of cold-inducible downstream genes of the Arabidopsis DREB1A/CBF3 transcriptional factor using two microarray systems]". *The Plant Journal*, 38(6):982–993, 2004.

[85] Akihiro Matsui, Junko Ishida, et al. "Arabidopsis transcriptome analysis under drought, cold, high-salinity and ABA treatment conditions using a tiling array". *Plant and Cell Physiology*, 49(8):1135–1149, 2008.

[86] Leland McInnes, John Healy, and Steve Astels. "Hdbscan: Hierarchical density based clustering". *The Journal of Open Source Software*, 2(11):205, 2017.

[87] Agata Michna, Herbert Braselmann, et al. "Plantexpress: a database integrating oryzaexpress and arthaexpress for single-species and cross-species gene expression network analyses with microarray-based transcriptome data". *Plant and Cell Physiology*, 58(1):e1–e1, 2017.

[88] Agata Michna et al. "Natural cubic spline regression modeling followed by dynamic network reconstruction for the identification of radiation-sensitivity gene association networks from time-course transcriptome data". *PLoS One*, 11(8):e0160791, 2016.

[89] Jeremy A. Miller, Chaochao Cai, et al. "Strategies for aggregating gene expression data: the collapseRows R function". *BMC Bioinformatics*, 12(1):322, 2011.

[90] Kenji Miura and Tsuyoshi Furumoto. "Cold signaling and cold response in plants". *International Journal of Molecular Sciences*, 14(3):5312–5337, 2013.

[91] Yasuhide Miyamoto, Young Ho Koh, et al. "Oxidative stress caused by inactivation of glutathione peroxidase and adaptive responses". *Biological Chemistry*, 384(4):567–574, 2003.

[92] Junya Mizoi, Kazuo Shinozaki, and Kazuko Yamaguchi-Shinozaki. "AP2/ERF family transcription factors in plant abiotic stress responses". *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1819(2):86–96, 2012.

[93] Ali Mortazavi, Brian A. Williams, et al. "Mapping and quantifying mammalian transcriptomes by RNA-Seq". *Nature Methods*, 5(7):621, 2008.

[94] Benjamin Moseley and Joshua Wang. "Approximation bounds for hierarchical clustering: average linkage, nisecting k-means, and local search". In *Advances in Neural Information Processing Systems*, pages 3097–3106, 2017.

[95] Daniel Müllner et al. "fastcluster: fast hierarchical, agglomerative clustering routines for R and python". *Journal of Statistical Software*, 53(9):1–18, 2013.

[96] Kazuo Nakashima, Hironori Takasaki, et al. "NAC transcription factors in plant abiotic stress responses". *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1819(2):97–103, 2012.

[97] Kazuo Nakashima, Kazuko Yamaguchi-Shinozaki, and Kazuo Shinozaki. "The transcriptional regulatory network in the drought response and its crosstalk in abiotic stress responses including drought, cold, and heat". *Frontiers in Plant Science*, 5:170, 2014.

[98] Renuka R. Nayak, Michael Kearns, et al. "Coexpression network based on natural variation in human gene expression reveals gene interactions and functions". *Genome research*, 2009.

[99] Mark E. J. Newman. "Finding community structure in networks using the eigenvectors of matrices". *Physical Review E*, 74(3):036104, 2006.

[100] María José Nueda, Sonia Tarazona, and Ana Conesa. "Next masigpro: updating masigpro bioconductor package for rna-seq time series". *Bioinformatics*, 30(18):2598–2602, 2014.

[101] Se-Jun Oh, Youn Shic Kim, et al. "Overexpression of the transcription factor AP37 in rice improves grain yield under drought conditions". *Plant Physiology*, 150(3):1368–1379, 2009.

[102] Naohiko Ohama, Hikaru Sato, et al. "Transcriptional regulatory network of plant heat stress response". *Trends in Plant Science*, 22(1):53–65, 2017.

[103] Lance Parsons, Ehtesham Haque, and Huan Liu. "Subspace clustering for high dimensional data: A review". *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, June 2004.

[104] Alex A. Pollen, Tomasz J. Nowakowski, et al. "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex". *Nature Biotechnology*, 32(10):1053, 2014.

[105] Pascal Pons and Matthieu Latapy. "Computing communities in large networks using random walks". In *Proceedings of the 20th International Conference on Computer and Information Sciences*, pages 284–293, 2005.

[106] Yuping Qiu and Diqiu Yu. "Over-expression of the stress-induced OsWRKY45 enhances disease resistance and drought tolerance in Arabidopsis". *Environmental and Experimental Botany*, 65(1):35–47, 2009.

[107] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". *Physical Review E*, 76(3):036106, 2007.

[108] Attipalli Ramachandra Reddy, Kolluru Viswanatha Chaitanya, and Munusamy Vivekanandan. "Drought-induced responses of photosynthesis and antioxidant metabolism in higher plants". *Journal of Plant Physiology*, 161(11):1189–1202, 2004.

[109] Matthew E Ritchie, Belinda Phipson, et al. "limma powers differential expression analyses for RNA-sequencing and microarray studies". *Nucleic Acids Research*, 43(7):e47–e47, 2015.

[110] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". *Bioinformatics*, 26(1):139–140, 2010.

[111] Martin Rosvall, Daniel Axelsson, and Carl T. Bergstrom. "The map equation". *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[112] Peter J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[113] Aurko Roy and Sebastian Pokutta. "Hierarchical clustering via spreading metrics". In *Advances in Neural Information Processing Systems*, pages 2316–2324, 2016.

[114] Rintaro Saito, Michael E. Smoot, et al. “A travel guide to Cytoscape plugins”. *Nature Methods*, 9(11):1069, 2012.

[115] Yoh Sakuma, Kyonoshin Maruyama, et al. “Functional analysis of an Arabidopsis transcription factor, DREB2A, involved in drought-responsive gene expression”. *The Plant Cell*, 18(5):1292–1309, 2006.

[116] Jil Sander, Joachim L. Schultze, and Nir Yosef. “ImpulseDE: detection of differentially expressed genes in time series data using impulse models”. *Bioinformatics*, 33(5):757–759, 2017.

[117] Jörg Sander, Martin Ester, et al. “Density-based clustering in spatial databases: The algorithm gdbscan and its applications”. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.

[118] Frederick Sanger, Steven Nicklen, and Alan R. Coulson. “DNA sequencing with chain-terminating inhibitors”. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

[119] Amit Saxena, Mukesh Prasad, et al. “A review of clustering techniques and developments”. *Neurocomputing*, 267:664–681, 2017.

[120] Mark Schena, Dari Shalon, et al. “Quantitative monitoring of gene expression patterns with a complementary DNA microarray”. *Science*, 270(5235):467–470, 1995.

[121] Franziska Schramm, Arnab Ganguli, et al. “The heat stress transcription factor HsfA2 serves as a regulatory amplifier of a subset of genes in the heat stress response in Arabidopsis”. *Plant Molecular Biology*, 60(5):759–772, 2006.

[122] Ju-Seok Seo, Joungsu Joo, et al. “OsbHLH148, a basic helix-loop-helix protein, interacts with OsJAZ proteins in a jasmonate signaling pathway leading to drought tolerance in rice”. *The Plant Journal*, 65(6):907–921, 2011.

[123] Dhriti Singh and Ashverya Laxmi. “Transcriptional regulation of drought response: a tortuous network of transcriptional factors”. *Frontiers in Plant Science*, 6:895, 2015.

[124] Shi-Yong Song, Ying Chen, et al. "Physiological mechanisms underlying OsNAC5-dependent tolerance of rice plants to abiotic stress". *Planta*, 234(2):331–345, 2011.

[125] Michael Steinbach, George Karypis, et al. "A comparison of document clustering techniques". In *Proceedings of the Workshop on Text Mining, the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 525–526, 2000.

[126] Jasmin Straube, Alain-Dominique Gorse, et al. "A linear mixed model spline framework for analysing time course 'omics' data". *PLoS One*, 10(8):e0134540, 2015.

[127] Bartel Vanholme, Wim Grunewald, et al. "The tify family previously known as ZIM". *Trends in Plant Science*, 12(6):239–244, 2007.

[128] Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. "A maximum variance cluster algorithm". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, 2002.

[129] Ramaiah Venuprasad, Honor Renee Lafitte, and Gary N. Atlin. "Response to direct selection for grain yield under drought stress in rice". *Crop Science*, 47(1):285–293, 2007.

[130] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance". *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[131] Qiuyun Wang, Yucheng Guan, et al. "Overexpression of a rice OsDREB1F gene increases salt, drought, and low temperature tolerance in both Arabidopsis and rice". *Plant Molecular Biology*, 67(6):589–602, 2008.

[132] Wei Wang, Jiong Yang, et al. "STING: a statistical information grid approach to spatial data mining". In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 186–195, 1997.

[133] Christian Wiwie, Jan Baumbach, and Richard Röttger. "Comparing the performance of biomedical clustering methods". *Nature Methods*, 12(11):1033, 2015.

[134] Lin Xia, Dong Zou, et al. "Rice expression database (RED): an integrated RNA-Seq-derived gene expression database for rice". *Journal of Genetics and Genomics*, 44(5):235–241, 2017.

[135] Dong-Qing Xu, Ji Huang, et al. "Overexpression of a TFIIIA-type zinc finger protein gene ZFP252 enhances drought and salt tolerance in rice (*Oryza sativa* L.)". *FEBS Letters*, 582(7):1037–1043, 2008.

[136] Dongkuan Xu and Yingjie Tian. "A comprehensive survey of clustering algorithms". *Annals of Data Science*, 2(2):165–193, 2015.

[137] Rui Xu and Donald Wunsch. "Survey of clustering algorithms". *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[138] Xiaowei Xu, Martin Ester, et al. "A distribution-based clustering algorithm for mining in large spatial databases". In *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 324–331, 1998.

[139] Ronald R. Yager and Dimitar P. Filev. "Approximate clustering via the mountain method". *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8):1279–1284, 1994.

[140] Youngik Yang, Kenneth Nephew, and Sun Kim. "A novel k-mer mixture logistic regression for methylation susceptibility modeling of CpG dinucleotides in human gene promoters". *BMC Bioinformatics*, 13(3):S15, 2012.

[141] Nari Yi, Youn Shic Kim, et al. "Functional analysis of six drought-inducible promoters in transgenic rice plants throughout all stages of plant growth". *Planta*, 232(3):743–754, 2010.

[142] Takumi Yoshida, Yoh Sakuma, et al. "Functional analysis of an Arabidopsis heat-shock transcription factor HsfA3 in the transcriptional cascade downstream of the DREB2A stress-regulatory system". *Biochemical and Biophysical Research Communications*, 368(3):515–521, 2008.

[143] Hong Zhang, Lan Ni, et al. "The C2H2-type zinc finger protein ZFP182 is involved in abscisic acid-induced antioxidant defense in rice". *Journal of Integrative Plant Biology*, 54(7):500–510, 2012.

[144] Xingnan Zheng, Bo Chen, et al. “Overexpression of a NAC transcription factor enhances rice drought and salt tolerance”. *Biochemical and Biophysical Research Communications*, 379(4):985–989, 2009.

[145] Jian-Kang Zhu. “Abiotic stress signaling and responses in plants”. *Cell*, 167(2):313–324, 2016.

초 록

본 논문은 유전자 발현 데이터를 분석할 때의 문제들을 정리하고 그 문제들을 해결하는 방법을 제시한다. 유전자 발현 데이터는 세포 내에 유전자가 활성화된 양을 수치화한 데이터이며 세포의 상태를 모델화하기 위하여 이 데이터를 사용한다. 하지만 세포는 이만 개 이상의 유전자, RNA, 단백질, 기타 화학 물질 등이 유기적으로 작용하여 구성되는 매우 복잡한 시스템이며, 이러한 세포를 모델화하기 위해서는 많은 수의 데이터가 필요하다. 그런데 현재 기술 및 자원적 한계에 의해 충분한 수의 데이터를 확보할 수 없으며, 적은 수의 데이터로 이 복잡한 세포를 모델화해야 하는 것이 유전자 발현 데이터 분석의 핵심적인 문제이다.

본 논문은 적은 수의 데이터로 세포를 효과적으로 모델화하기 위하여 클러스터링과 네트워크 기법을 사용하여 기존의 생물 지식과 공개된 데이터를 통합적으로 이용하는 방법론을 제시한다. 그 구체적인 방법은 다음과 같다. 클러스터링 분석을 통해 개별 유전자를 적은 수의 클러스터로 묶음으로써 특성 차원을 축소하고 모델화의 복잡성을 줄임으로써, 적은 수의 발현량 데이터로 세포의 상태를 모델화하고 해석하는 방법을 제시한다. 대량의 외부 데이터로부터 유전자 네트워크를 구성하고 실험 데이터로 구성한 네트워크와 통합함으로써 생물학적 도메인 데이터와 지식을 네트워크를 형태로 분석 과정에 도입하여 모델의 정확성을 향상하는 방법을 제시한다. 이질적 시간 구조를 가지는 다수의 시계열 데이터를 통합하는 분석에서, 클러스터링 방법으로 유전자의 반응 순서가 보존되는 유전자들을 찾는 방법을 제시한다. 아직 그 분포를 알지 못하는 유전자의 집합을 클러스터링하기 위해, 앙상블 기법 및 비용 최소화 기법 등 최신 클러스터링 기술을 사용하여 계층적 클러스터링 방법을 향상한다.

정리하면, 이 논문은 복잡한 시스템이면서 데이터 개수가 적어 모델화가 어렵고, 시계열 구조가 비균질한 유전자 발현 데이터 분석의 문제를 클러스터링과 네

트워크를 기반으로 통합 분석하여 해결하는 방법을 제시한다. 또한 이러한 개발한 방법들을 실제 스트레스 실험 데이터에 적용하여, 가뭄 저항성 벼의 메커니즘을 설명하고, 저온 스트레스에 대해 반응하는 유전자를 검출한다. 제시된 방법론은 컴퓨터 공학의 데이터 분석 분야에서 비슷한 문제를 가진 문제들을 해결하는데 활용될 수 있을 것으로 기대된다.

주요어 : 클러스터링, 네트워크, 통합 분석, 시계열, 유전자 발현 데이터, 스트레스 반응 유전자

학번 : 2012-23221