# Extended Methods and Biological Networks using Explainable AI Algorithm Iterative Random Forest

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Ashley Cliff

May 2022

# Acknowledgments

I would like to thank my family and friends for being supportive of my work, even if they had no idea what I was doing. I would like to thank the many instructors I have had through the years who instilled in me a deep drive to be curious and to always keep learning. And I would like to thank Jonathon Romero, who joined me on this journey and kept me sane as we followed this path together.

# Abstract

The rapid pace of increasing biological data may require a combination of high performance computing, machine learning, data analytics, graph theory, and systems biology-based processes to best further scientific research. Almost all biological functions operate in conjunction with the system they are in and to best understand them, systems biology studies the function in the context of its system. To handle the highly interacting complexities within these systems, new high performance and human understandable algorithms should be utilized. To leverage the explainable AI capabilities of machine learning model iterative Random Forest (iRF), a high performance computing capable implementation is introduced. This new implementation enables iRF-LOOP and iRF Cross-Layer, new network methods for omics interaction analysis for within- and across-omic relationships. By using these methods together, iRF-based heterogeneous networks with omics layers provide a systems biology view. Using these methods with public data for model organism *Arabidopsis thaliana*, a large-scale network resource is produced for hypothesis generation and research utility.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Background

The field of Biology is seeing rapid growth in the amount of available data. The quantity of genomic data being generated is increasing at an exponential rate, and is quickly catching up to big data fields such as astronomy[123]. Due to the complex nature of biological systems[175, 185, 191], new methods for analyzing data are constantly being developed and existing methods are being refined and expanded[113]. This complexity expresses itself often as combinatoric interactions within and across biological data layers, leading to a necessity for a full systems biology approach to study these interactions and algorithms designed with this expected level of complexity, large data, and cross-layer interaction.

As new methods are being developed, more emphasis is being placed on usage of high performance computing (HPC), an area biology has been slow to enter[25, 8]. While computationally intense, biological research has historically lagged behind fields like astronomy and molecular chemistry in the need for fast calculations for research, as with simulations of cosmic events or molecular interactions. As more data has become available, research has reached a point where questions that previously could not be answered due to lack of data are now reaching the limit of currently used methods and systems. In contrast, large computational power allows us to think in totally new and novel ways that traditional computing would consider infeasible. Problems that would take months or years of compute time using a basic statistics approach on a laptop can now be completed in hours with appropriate algorithms on high performance systems.

Concurrent with a drive for larger computational resource use is a surge in machine learning and artificial intelligence (ML/AI) interest in biology[158, 30, 154, 137, 128]. AI is often considered as a means for prediction of some 'future' event or data by learning intrinsic information in the data at hand, however that is not the only use for ML/AI in biology. For prediction, many ML/AI and deep learning methods are appropriate. However, when asking *how* a method is reaching its conclusions, many of these same methods are considered a 'black box' and are very difficult to interpret[2]. Due to biology's high quantity of data, and general need to mine the data for information on associations and interactions, rather

than the need for large quantities of calculations or predictions of future events, different approaches and techniques are being explored.

Explainable machine learning, or X-AI, methods are superior for these tasks[2]. These methods provide prediction and feature selection abilities with easily interpretable algorithmic processes. *Classification and Regression Trees (CART)* was published by Leo Breiman in 1984[20], and is now considered an early machine learning method. Following in 2001 was Breiman's *Random Forest*[19], as both a prediction method that handles over-fitting, as well as a feature selection method, where the decision making process is human-understandable. Due to the straightforward and transparent analysis process used in CART and then Random Forest, these methods are both well regarded X-AI methods for interpretation of biological data[135, 81, 18, 59, 82, 112, 107, 33, 193, 97], with many available offshoots and specialties[78, 63].

As with any method, specific considerations should be taken to properly set up data and analyze results[30]. As ML methods have become more commonplace and easy to implement, these intricacies and caveats are getting swept aside by promises of high prediction accuracies and popular models. Random Forest methods require their own considerations and data cleaning practices for appropriate use and results interpretation. These generally include variance analyses, batch effect analysis caused by biological sampling processes and outlier detection.

Complementary to X-AI methods is Graph Theory and the network topology-based concepts and algorithms it provides. Often, diagrams of biological processes are visualized and interpreted as pathways[22, 129], implying a set of routes and a methodical step-by-step process. With the ability to concurrently analyze more and more aspects of biological systems, networks are becoming a more useful visualization and analysis tool[4], as they allow more flexibility for high levels of interaction between components of a biological system. Networks can be used to view interactions between genes, such as with co-expression networks[74], interactions between proteins as in protein-protein interaction (PPI) networks[110], and many types of interactions that involve multiple types of data[183].

To approach the ever-increasing complex questions in biology will require some combination of all of these areas. Systems biology level analysis is important to start understanding

the many interacting components of biological systems. New high performance-capable and human understandable algorithms will need to be designed and implemented that can handle the combinatoric complexities within these systems. Appropriate consideration must be given to data set up and analysis. Visualization techniques capable of showing these interactions and providing a structure to work within for biological discovery must be tested and expanded. Together, these tools and processes open the doors to biological analysis on a scale generally considered infeasible.

## 1.2 Iterative Random Forest

The Random Forest (RF) algorithm was introduced by Leo Breiman in the early 2000s[19], drawing on his earlier works defining Categorical and Regression Trees (CART)[20]. The basic algorithmic unit is a decision tree, whose main goal is to subdivide a set of samples into 'like' subgroups based on a dependent variable, using features shared by all the samples. The tree starts at a root node and repeatedly makes decisions, splitting into two children nodes. At each decision (node split), the best feature is selected based on which feature and value 'best' subdivides the dependent variable. This process is repeated recursively until appropriate stopping parameters are reached, which could be tree depth, number of samples in a terminal node, or 'purity' of the dependent variable for the samples in that terminal node. A feature importance score is calculated for each feature based on how influential it was in the creation of the model.

A Random Forest expands on this process by creating many decision trees in tandem, with the main difference being that only a random subset of the features are checked at each decision. The feature importance scores are then average across all of the trees in the forest. This process of randomly looking at features across many splits and many trees allows for the analysis of highly-combinatoric features without the brute force analysis of all possible combinations of features, allowing for the analysis of much larger and more complex systems. Figure 1.1 below provides some simple definitions and concepts of a Random Forest. See *A Random Forest Guided Tour*[15] by Biau and Scornet (2016) for a more in-depth look into the history and underlying mathematics of the algorithm.

The expansion of RF, iterative Random Forest (iRF)[13], takes this process another step farther. iRF creates iterative sets of random forests, where a weighting scheme, calculated based on the prior RF's importance scores, are used to determine the feature sampling at each split. This weighted random sampling provides two main benefits. First, the weighting process provides the forest with the chance to slowly consolidate importance from features that may be providing the same information. Secondly, the iterative process allows for a more ordered path through the decision trees themselves, providing conditionalities for splitting. iRF also provides the use of Random Intersection Trees (RIT) as a method to mine these paths for repeating patterns, which may be able to find conditional feature sets of varying sizes. For biological data sets and features, this ability to condense features and find combinatoric relationships is of significant use.

### 1.2.1 Splitting Criteria

Determining which feature and feature value best splits the data is a main algorithmic inclusion of the method[192]. The splitting criteria used for this calculation can differ depending on the tree type - classification for discrete and non-ordered classes or regression for continuous values, as well as simply the implementation of the algorithm in use. Of the different options, the most frequently used methods are Gini impurity - as used by Brieman in the original CART for classification, information gain[136], and decrease in variance[188].

Gini impurity is the measure of how often a randomly selected sample would be labeled incorrectly if it was randomly classified according to the distribution of classes in the given set of samples. Mathematically, it can be viewed as the probability $p_x$ of sample $x$ being chosen times the probability $\sum_{z \neq x} p_z = 1 - p_x$ of a mis-catogorization of sample $x$. For use in a decision tree, at each split, the feature and feature value that best minimizes the impurity in the children nodes is chosen.

Information gain is based on entropy and information content. Entropy is defined as:

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, ..., p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

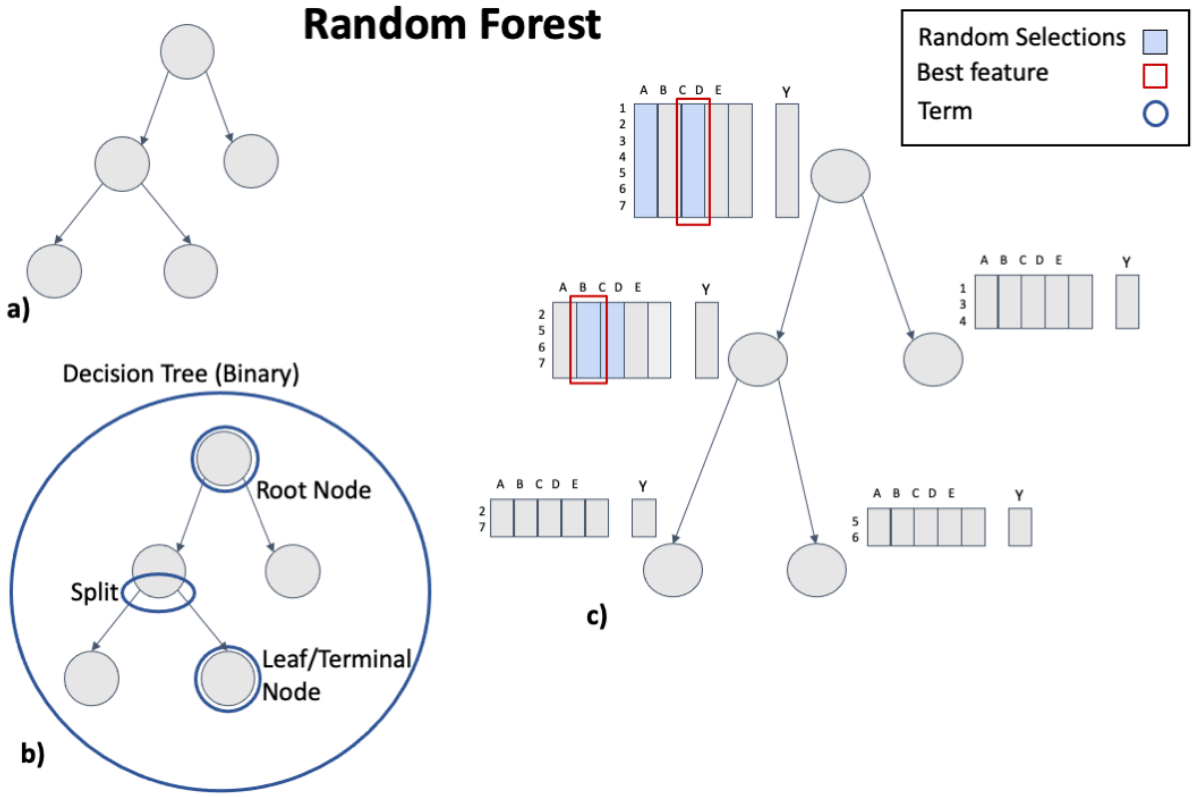



**Figure 1.1:** A visual diagram of the different aspects of a Random Forest - the three trees shown would together be considered a small Random Forest. a) A basic binary decision tree. Each grey node represents a set of data and each arrow represents a decision that split the data in two. b) Some common terms used with decision trees and Random Forests. A decision tree, as shown in a) is the whole entity. A root node is the starting node of a tree that contains all data. A split is the final step in a decision. A leaf, or terminal, node is the final node a data sample when propagate into. c) A visual diagram of the samples (rows) and features (columns) used in a decision tree within a Random Forest, where a subset of features are randomly analyzed prior to each split and one is chosen. In a Random Forest, all features propagate through the entire tree, but the samples only flow through specific segments from root to a given leaf node.

where $p_1, p_2, ..., p_J$ are fractions that add up to 1 and represent the percentage of each class present in the child node of the current node. Information gain is then:

$$\mathrm{IG}(T, a) = \mathrm{H}(T) - \mathrm{H}(T|a) = -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{a} p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

where, $a$ represents the children nodes. Simply, information gain is the entropy of the parent node minus the entropy of the children nodes. Within a decision split, the goal is to minimize the entropy in the children, which maximizes the information gain, by choosing the best feature and feature value to split upon.

Decrease in variance is often used for regression trees, where the dependent variable is continuous. The *change* in variance of parent node $p$, split into left child $l$ and right child $r$, with corresponding sample counts $n$, $n_l$, and $n_r$ respectively, is defined as:

$$\Delta\sigma^2(p) = \Sigma(x_p - \hat{x_p})^2/n_p - \Sigma(x_l - \hat{x_l})^2/n_l - \Sigma(x_r - \hat{x_r})^2/n_r$$

By maximizing this equation, the amount of variance in the children is lower than in the parent, thus decreasing overall variance in the tree. The feature and feature value that best maximizes the equation is chosen at each split.

### 1.2.2 Comparison to Other Methods

Appropriate comparison of iterative Random Forest to other methods requires consideration of its two main use cases independently - feature selection and prediction. As this dissertation focuses solely on the feature selection aspect, this section will focus on comparing iRF to other feature selection methods. Comparisons will include how each method handles large data, how efficiently the methods search the solution space, the computational intensity, and how well the method handles combinatoric or non-linear feature interactions.

Feature selection is an often used method in biological research[44, 99, 147, 141] and there exist many tools and packages that provide suites of options for feature selection[144]. However, some of these tools may actually provide *feature extraction*, rather than *selection*, where the resulting features have been transformed in some way[87]. As these features are

no longer the originals, these tools are less directly useful for processes where the importance of input features themselves is the desired outcome, and as such, feature extraction methods will not be discussed here.

A mainstay of feature selection analyses is Pearson's Correlation[23]. For each feature $x$ and dependent variable $y$, for $n$ samples, Pearson's Correlation is defined as:

$$r_{xy} = \frac{\sum x_i y_i - n \sum \overline{xy}}{\sqrt{\sum x_i^2 - n\overline{x}^2}\sqrt{\sum y_i^2 - n\overline{y}^2}}$$

As with iRF's feature importance scores, a cut off value must be determined to provide feature selection choices, rather than simply a score for every feature. The method is relatively straightforward and simple to calculate, but does not scale well with large data, as a direct comparison is required for each feature. Because the process is exhaustive, the entire solution space for pair-wise relationships is searched. However, the computational cost grows exponentially when searching for higher-order associations. Unlike iRF, Pearson's Correlation does not inherently search the higher-order combinatoric space and the requirements to do so limit its other benefits.

Forward Feature Selection (FFS) and Backward Feature Elimination (BFE) are two simple feature selection processes that have many versions, but with the same basic steps. Given a set of features and a given criteria for model accuracy, a process is repeated until the accuracy has been optimized[23]. For forward selection, one feature at a time is added, and for elimination, one feature at a time is removed, both until the model accuracy maximum is reached. The processes are greedy in the sense that every feature is checked in conjunction with the others, providing some accounting for higher-order interactions. However, the order of the features heavily influences the process and predetermining that order can be nontrivial. Determining a model accuracy measure that is appropriate may also be nontrivial. If the model used provides a feature importance value, backward feature elimination becomes easier by simply removing the feature with the lowest importance value. Compared to iRF, FFS and BFE are highly dependent on their meta-parameters for efficiency and complexity, and still may not appropriately capture interaction between features.

Mutual information (MI) provides a measure of dependency between two variables[23, 14, 161]. For discrete distributions, or samples from continuous distributions, for feature $x$ and dependent variable $y$, mutual information is defined as:

$$\mathrm{I}(X;Y) = \sum_{y \in \mathbb{Y}} \sum_{x \in \mathbb{X}} p_{(\mathrm{X,Y})}(x,y) \log \left(\frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)}\right) dxdy$$

where $p_{(\mathrm{X,Y})}(x,y)$ is the probability of the specific combination of $x$ and $y$ in the probability space of (X,Y). Conceptually, MI is very similar to information gain, as described in section 1.2.1. As a splitting criteria, information gain is only looking at how much information regarding the dependent variable can be gained from an individual feature and feature split value, whereas MI, as a feature selection method, is used to look at the overlap in information between the dependent and an independent variable. As a feature selection method, MI has the benefit of good precision if the number of samples is high, but can be severely impacted by small sample sizes, especially as the number of independent variables grows. Along with the same issues associated with Pearson's Correlation, MI also does not scale well for biological data, which is often high in feature number and low in sample number.

Support Vector Machines (SVMs) are a marginal classifier which maximizes the margin between the data samples in the two classes[23, 17]. The SVM builds a hyperplane in n-dimensions using a binary linear classifier, where n is the number of features, based on the best possible splitting of the samples that maximizes the distance of the two resulting groups from the hyperplane[70]. The dimensions most represented in the hyperplane parameters use the most important features. SVMs can perform non-linear classification, but the process requires conversion to higher dimension feature spaces, thus losing the the ability to attribute importances to any original features. Because SVMs are inherently designed for binary classification, applying them for other types of data, SVMs start to lose their benefits - becoming much more expensive to calculate. Compared to iRF, SVMs are very useful for specific tasks, but do not generalize as well.

LASSO (Least Absolute Shrinkage and Selection Operator) is a linear model and a 'particular case of the penalized least squares regression with L1-penalty function'[122, 159]. Least squares regression determines the regression coefficient that minimizes the squared

error. LASSO implements a weighting function to determine which features optimally reduce the error and a tuning parameter determines the number of features allowed to be used in the weighting. Reducing the tuning parameter in turn reduces the total number of features selected. As a linear model, LASSO does not account for interacting, or conditional, features. Ridge regression is a similar method, with potential use when the features are highly correlated and uses an L2-penalty on the weighting function, rather than the L1 used with LASSO[73]. Because L1 and L2 prioritize different information, Elastic Net is a method that combines LASSO and Ridge regression by optimizing the combination of L1 and L2 penalties[203]. These processes can be extremely computationally efficient and they do search the full space, however they are limited by the tuning parameters and they do not account for non-linear feature relationships, as iRF does.

### 1.2.3 Versions and Off-Shoots of Random Forest

Random Forests are a very useful tool for biologist, as they not only have the ability to provide feature selection and prediction - as a Machine Learning method, but they also provide the results in a human-understandable method - deeming them an Explainable AI (X-AI) method and provide analysis with interaction data. There have been many spin offs of the Random Forest for different uses in biology; provided here are a few of the most commonly used and recently released.

Published in 2010, the RF-based method GENIE3[78] (GEne Network Inference with Ensemble of trees) is designed to create Gene Regulatory Networks (GRNs). As with RFs, the method requires very few assumptions about the relationships between the features, allowing for high-order conditional dependencies. GENIE3 performed well on the DREAM4 challenge in its ability to match an existing gold standard network for *E. coli*. GENIE3 differs from a RF mainly in the creation of multiple tree ensembles (RFs and/or Extra-Trees) - one for each gene, and the creation of a network using the feature importance scores as edge weights. In 2018, the authors introduced dynGENIE3 - dynamical GENIE3 for gene networks from time series data[77].

Published in 2015 as a dedicated R package, VSURF (Variable Selection Using Random Forest), returns two subsets of features, one for prediction and the other for model

interpretation or feature selection[63]. The multi-step process starts with average rankings of each feature based on multiple RF runs (50) and immediately eliminates the features with small importance - based on the standard deviation of each feature's importance across the RFs. For model interpretation, a nested set of RFs are created using the first $k$ features, where $k$ ranges from 1 to the total number of features remaining. The features from the resulting RF with the smallest out-of-bag error are the final features for model interpretation. For prediction, starting with the remaining features from the original pruning, sequential RF models are created 'in a stepwise manner' and the variables from the model that reduce the out of bag error the most are maintained. For many analyses, such as those done for gene importance for a physical trait, a different set of genes may be reported for a full set of 'relevant', i.e. model interpretation, and 'necessary' for prediction. Being able to obtain both from the same model allows for the confirmation that the features were all considered appropriately, and saves time.

Published in 2018, COMPACT+FV[54] introduces a new type of feature importance for Random Forest. This new measure focuses only on positive feature values (ignoring negative values). Positive features are those where the biological property is present, requiring the expectation of a binary dependent variable. This new method provides a novel algorithm 'that measures the predictive accuracy of a positive feature value by its overall predictive accuracy across all rules (root-to-leaf paths) in the RF where that feature value occurs.' The authors justify the metric with the logic that a positive feature indicates stronger evidence for a property than a negative feature indicates a lack - ie negatives can simply be 'unknowns'. This new method allows for the analysis of data sets with incomplete dependent variables or unbalanced sample sets.

Published in 2019, TSLRF is a two-stage algorithm based on least angle regression and random forest[156], designed for the identification of SNPs related to a specific trait. The algorithm first considers population structure and polygenic effects, then selects SNPs using least angle regression, followed by RF analysis of the selected SNPs to detect quantitative trait nucleotides. The authors tested the method on both simulated and real data for validation and showed that TSLRF had better prediction accuracy and lower compute time than comparable methods, including RF.

## 1.3 Big Data and High Performance Computing

A foundational understanding of the different architectures and processes used with big data and high performance computing is extremely beneficial when designing new algorithms. Specifically for biology, where for many years and many projects, the data sets were no larger than a spreadsheet and the analysis required nothing more than a laptop, the shift to appropriate management of giga- and tera-bytes of results and large computational systems has not yet permeated the field. However, the changes are coming and the need for algorithms and implementations that can handle these larger data sets in a timely fashion is growing.

### 1.3.1 System Architecture

High performance computers (HPCs), or 'supercomputers', differ from general purpose systems in the amount of calculations that can be done at any one time, generally in a massively parallel manner. The basic computational/processing units consist of cores - the computation pieces, CPUs (central processing units) which house a set of cores, sockets which house multiple CPUs, and compute nodes which house one or multiple sockets. With the recent increase in GPU (graphical processing unit) computing, a socket may contain a number of GPUs as well as CPUs or instead of CPUs.

Within the category of HPCs themselves, there are many different designs and architectures used. The structure of the memory architecture, shared versus distributed, is one of the main divisions, as well as one of the most relevant for algorithm development. A shared memory system, as the name suggests, contains a single memory unit that is shared by all of the processing units. Alternatively, distributed systems generally have separate memory units for each compute node. This memory unit is then shared by the CPUs within that node. Due to the amount of data analysis and data processing done with many algorithms, especially for biological analysis, it is necessary to know the memory architecture of the system the algorithm will be used on prior to development of the implementation.

Parallel computing describes any computing where multiple cores/CPUs/nodes are used together. Within a single core, parallelism may be achieved with threads, in a process called multithreading[167, 166]. When multiple CPUs are used together, the process

is multiprocessing. Within a software implementation, both or either types of parallel computing can be used.

With shared memory systems, the main consideration for multiprocessing implementations of algorithm is maintaining a structured system for how memory objects are changed[67]. If one thread accesses a memory location and changes a value and then checks it later, it is imperative that all threads operating on that location are expecting the change. Memory locks, where only one thread can access a memory address at a time[168, 131], help to alleviate this problem, but can cause processes to run slower if not implemented in an efficient way.

For distributed memory systems, the process of accessing and sharing memory objects becomes more complicated. To share, the data needs to be passed from one memory object to one or more other memory objects. This process became known as message passing or the Message Passing Interface (MPI), and multiple conferences were held to set standards for this messaging system[Walker and Dongarra, 31]. Many implementations of these standards exist, with a few becoming very popular, such as MPICH[68] and Open-MPI[61].

### 1.3.2 Algorithm Design

Separate from the memory architecture of the system to be used is the guiding plan for the algorithm and implementation. Namely, will many computations and analyses be done to one set of data or will one set of computations be done to multiple sets of data, or potentially both. These concepts encapsulate the ideas of task and data parallelism. If a single program spawns multiple threads to complete the same process on multiple pieces of data, the software uses a 'single-program multi-data' design [67]. However if multiple processes are run on a single set of data, the software uses a 'multi-program single-data' design. If a piece of software uses multiple process on multiple pieces of data, then the design would be 'multi-program multi-data'. Determining which of these paradigms is the most appropriate for the algorithm and implementation may point towards or away from different languages, resources, wrappers, or designs.

### 1.3.3 Importance in Biology

High performance computing has been used in beneficial ways as early as the mid 1980s, specifically in molecular biology[116]. High performance computing has often been used for studies that involve atomic or molecular simulations[148, 111] and protein modeling[40], statistical phylogenetics[9], and high-throughput model testing in fields such as therapeutics[100].

## 1.4 Complex Biological Systems

Specific considerations for computational analysis may differ depending on the type of data to be analyzed, as they come from different measurement processes and have different inherent characteristics. Because of this, a solid understanding of both the organism and the type of data measurement is necessary to appropriately apply a model, be it statistics or machine learning.

### 1.4.1 Plant Model Organisms

Model organisms are example organisms which are heavily studied for specific biological processes or phenomena. This group generally includes bacterium *Escherichia coli*, baker's yeast (*Saccharomyces cerevisiae*), nematode worm (*Caenorhabditis elegans*), fruit fly (*Drosophila melanogaster*), and the mouse (*Mus musculus*), and dependent on context may also include the mustard plant (*Arabidopsis thaliana*), fission yeast (*Schizosaccharomyces pombe*), zebrafish (*Danio rerio*), the frog (*Xenopus laevis*)[55] and for food crops, rice (*Oryza sativa*), maize (*Zea mays mays*), and wheat (*Triticum aestivum*)[16]. Due to the common ancestry of most biological entities, studies of a few model organisms is often able to provide insights into similar systems in other organisms.

#### *Arabidopsis thaliana*

Due to *Arabidopsis thaliana*'s fast growth rate, easy transformation ability, small size, and close relationship to several hundred thousand plant species[153], it developed as a model

organism for the study of plants. The first complete genome assembly was released in 2000, after approximately sixty years of slow adoption as an organism worth extensive study. In 1943 Dr. Friedrich Laibach outlined the suitability of *Arabidopsis thaliana* as a model organism[95]. In the 1950s, it was used to study mutants in Germany and biochemical genetics in the Netherlands in the 1960s[153]. The *Arabidopsis* Information Service (AIS) newsletter was established in 1964. In 1975, *Arabidopsis thaliana* was suggested as an ideal plant for genetic experiments by George Rédei[139]. More justifications for using *Arabidopsis thaliana* were published in science and genetics journals in the mid 1980s, with the first gene sequences published in 1986.

Due to its eventual widespread study and use as a model organism, *Arabidopsis thaliana* has a variety of database resources for public data access. One of these, TAIR (The Arabidopsis Information Resource) contains a large quantity of genetic and molecular data specific to *Arabidopsis thaliana*, including genome sequences, gene structure, gene products, gene expression and more. In 2021, a new database, AtMAD ( *Arabidopsis thaliana* multi-omics association database)[98] was introduced for multi-omic data and associations, emphasising *Arabidopsis thaliana*'s use as model organism for systems biology research[177].

### *Populus trichocarpa*

As climate change has become more pronounced, interest in sustainable, non-fossil fuels has become of increasing interest, leading to a program funded put by the Department of Energy to create Bioenergy Research Centers (BRCs), starting in 2007. These BRCs were to engage in research to forward the knowledge of bioenergy and biofuel understanding and 'to provide the fundamental science to underpin a cost-effective, advanced cellulosic biofuels industry' (https://genomicscience.energy.gov/centers/centers2007.shtml). One of the three grand challeneges of this goal was to develope next-gen biocrops, with a strong understanding of the underlying plant biology.

*Populus trichocarpa* is specifically useful for bioenergy and biofuels[169, 184], as it has a fast growth rate, moderate lignin content, and high cellulose content ideal for cellulosic biomass-based processing. As such, it has come to be thought of as a model organism for woody plants and trees by many[190, 80, 48, 50]. Due to its position as a potential bioenergy

feedstock, with backing by the DOE, much research has been done to provide a wide variety of data across a multitude of genotypes.

The first poplar genome (from the Nisqually-1 genotype) was sequenced in 2006[172] and was the first tree to have its genome sequenced. Since then, over 1300 *Populus trichocarpa* genotypes have been resequenced for population and association studies[170, 53]. Seven clonal replicate common gardens have been designed and initiated, providing over 1000 natural accessions from across the US and Canada, with periodic phenotypic measurements and genetic sequences for each accession[171]. Multiple Genome Wide Association Studies (GWAS, see section 1.4.4) have been done to find associations between genetic and phenotypic variance, with specific studies focusing on various biomass traits[120], wood composition[69], and climate adaption[119]. Research has also been done with *Populus trichocarpa* for plant-microbe interactions[37]. Due to the drive for sustainable biocrops and the quality and quantity of data that has been provided, *Populus trichocarpa* is a prime candidate for multi-omic analyses and whole-system studies.

### 1.4.2 Omics Layers

Here, I use 'omics' layers to refer to a specific type of biological unit or process and its set of measurements. The most often studied include the genome - or DNA, the methylome - methylation of DNA, the transcriptome - the transcription of DNA as RNA, and the metabolome - metabolite production, and more recently, the microbiome - microbial organisms that interact with the organism studied, symbiotically or parasitically.

#### Genome

The genome refers to the set of chromosomes and genetic material in an organism, generally DNA. The four nucleobases that make up the DNA - adenine (A), guanine (G), cytosine (C), and thymine (T) - encode for genes or provide non-gene structural aspects. In the most basic view, variation in physical traits and biological function can be attributed to differences in the nucleotide pattern of the genome. These differences, single nucleotide polymorphisms (SNPs), are a quantifiable measure of how different one genome is from

another, generally a reference. However, other differences, such as copy number variation (CNVs), where parts of the genome are duplicated any number of times, removal or addition of one or more nucleotides, or other structural changes can also occur and may effect traits and functions[1].

These differences in genetic patterns provide the basis for phenotypic differences, although they are not necessarily the sole cause. The full relationship between any phenotype and genome is often very complex, involving many different genes, as well as structural differences and varying levels of biological compounds. To better understand these relationships, studies are often done on large, diverse populations to find associations between specific genotypic and phenotypic characteristics. Because of linkage disequalibrium (LD), where different segments of a genome are frequently co-inherited with each other[140, 58], if the population studied is not diverse, genetically, it may be hard to differentiate between causal genetic markers and those simply 'in LD' with those markers.

The field of genomics has become possible due to the increase in genetic resources, from identifying individual genes to assembling entire organism genomes. Identifying the nucleotide sequences of the genome and common patterns in genetic variation are now only one aspect of a vast array of data allowing researchers to better understand phenotypes.

### Methylome

An important mechanism contributing both to DNA structure and gene regulation is DNA methylation - the addition of methyl groups to cytosine bases. Methylation of DNA in gene body and promoter regions plays an important role in gene regulation. In general, methylation has been found to repress gene expression by blocking the binding sites of transcriptional activators. However, there are known cases where methylation has been found to activate gene expression. DNA methylation has been a topic of extensive research in recent years; Zhang, et al.[197] provides a detailed review on DNA methylation for further information.

The two main approaches to generating methylation data for study are methylation arrays and whole genome sequencing (WGS). Methylation arrays are cheaper and easier to use and allow for the same methylation sites to be measured in many samples, however, they are

limited by the identification of mehtylation levels at pre-determined regions of the genome, defined by probes in the array[96]. Whole genome sequencing, on the other hand, assays the whole genome, allowing for greater novel research potential. WGS is, however, a more complicated and expensive process and requires more thorough data processing. Methylation levels can differ wildly across species[125], as well as tissues[180], requiring careful analysis of results.

Methylation analyses in *Arabidopsis thaliana* have shown that methylation is heavily related to transcription - moderately transcribed genes being more likely to maintain specific methylation patterns than either highly or lowly transcribed genes[202] and that methylation disruption can cause developmental abnormalities such as smaller plant size and abnormal leaf shape[56]. Research has also shown that there is a complex relationship between the three types of sequence contexts in plant methylation, CG, CHG, and CHH (where CG indicates a cytosine followed by a guanine and H indicates A, T, or C), within *Arabidopsis thaliana*, with both mechanism and pattern contributing to varying influence on gene expression and regulation.

### Transcriptome

Transcription is the process of transcribing a segment of DNA into RNA, often referred to as gene expression. The amount of transcription of a segment section of DNA creates multiple RNA molecules. As such, the transcriptome is defined by the quantification of these pieces of RNA within an organism, or levels of gene expression. RNA molecules can be either messenger RNA (mRNA) that encode for protein structures or non-coding RNA (ncRNA). RNA contains an ordered set of nucleic acids, where the specific arrangement of bases determines the process that the molecule will enact. The comparison of transcriptomes across differing conditions or tissues is referred to as differential expression[138]. Levels may vary due to transcriptional regulation[108], the process of limiting which sections of DNA are able to be transcribed into RNA.

One of the most common types of data generated for the analysis of transcription is RNA-seq data. This method has greatly increased the ability to compare expression levels across samples and datasets, which was challenging with hybridization techniques such as

microarrays. Using high-throughput sequencing techniques, the amount of RNA in a sample is quantified by way of measuring cDNA(complementary DNA)[132], which is synthesized from an RNA template by reverse transcriptase. Initial sequencing methods included the use of Sanger sequencing to directly determine the cDNA sequence. In recent years, RNA-seq has become a much more high throughput method, utilizing RNA which is converted to a cDNA library, amplified, sequenced, and mapped back to a reference genome. Through this process the transcription of annotated genes in quantified by the depth of sequence reads mapped to that region of the genome.

### Metabolome

Metabolomics is the analysis of metabolites, such as 'small molecules such as sugars, organic acids, amino acids, and nucleotides, that are produced through cellular metabolism'[86]. The metabolome of an organism is known to change based on external factors[163, 164]. By studying the differences in organisms and those factors, novel functions of metabolites can be determined and utilized.

Until the last decade, nuclear magnetic resonance (NMR) spectroscopy was the most widely used technique to analyze metabolites in a high throughput fashion. Recently, mass spectrometry (MS) based methods have become more popular because they provide increased sensitivity[86, 46]. Gas chromatography- mass spectrometry (GC-MS) separates compounds by GC then analyzes them via MS, which reduces the complexity of the mass spectra and gives added information about physio-chemical properties about the metabolites[46].

The analysis process and methodologies for metabolomics generally fall into two categories: untargeted metabolomics, which analyzes all the measurable chemical quantities in a sample including those that are still unknown, and targeted metabolomics, which measures 'defined groups of chemically characterized and biochemically annotated metabolites'[142]. As such, metabolic research may focus on analysis of metabolites to understand their function or interactive analysis to understand their influence on other aspects of the organism, or often both.

#### Microbiome

As the majority of plants exist in some partnership with fungi[21] and/or bacteria[155], the microbiome can be considered an omics layer and studied alongside other omics layers[36]. Research has shown that the fungi and bacteria microbiome on plant roots, specifically *Populus trumula* × *alba*, correlates with metabolite production[117]. Analysis has shown commonalities within root microbiomes across plant species, indicating structure to the microbiome[36]. Due to the localized, and potentially functional, nature of microbial interaction, microbiomes may differ within a single plant from root to xylem to leaf[71].

The gathering of microbial data is inherently tied to the analysis that may be preformed with the results. The two main microbial analysis processes, in the context of colonized organisms, are 16s rRNA sequencing and metagenomics. 16s ribosomal RNA is an extremely conserved sequence that is able to distinguish between microbial species when used in PCR primers (16s Surveys), whereas metagenomics involves shotgun sequencing of all of the DNA content in a sample, as opposed to just the 16S gene[162]. 16s rRNA can provide relatively accurate counts of different microbes within the microbiome, while metagenomics may have more false matching of sequences to microbe but is able to provide 'functional capacity'. Kraken2 is a metagenomics tool, able to quickly assign taxanomic labels to metagenomic (often microbial) DNA sequences[187]. Small sections of genome sequence, k-mers, are used to match input data sequences to reference sequences to one of many potential organisms across kingdoms and species.

### 1.4.3 Central Dogma of Molecular Biology

The central dogma of molecular biology[38], originally put forward by Francis Crick in the 1950s and then updated in the 1970s, describes the association between the three main biological units: DNA, RNA, and protein. Namely, the dogma indicates the possibility of a two-way transfer 'of sequential information' between DNA and RNA, but that once either DNA or RNA transfer to protein there is no return transfer. This view of how the different omics layers relate to each other provided the foundation for cross-omic associations. However, as research has progressed, the understanding that this dogma may be limited in

scope has grown. Cross-omic relationships and patterns are varied and potentially cyclic, rather than the strict one-way process proposed. They also include many other types of biological entities, such as metabolites and even microbes, that are not included in the original conceptual model.

### 1.4.4 Cross-omic Associations

At the foundational level of biology, the goal is to understand how organisms function. To do this, one must have a thorough understanding of how the many biological subsystems and layers within an organism relate, as biological systems are highly complex and interactive. As such, there are many methods and processes to determine associations across omics layers.

Quantitative trait locus (QTL) analyses are broadly the methods and analyses used to find associations between a quantifiable biological unit (locus) and some phenotype (trait)[92]. Until the 1980s, quantitative analysis focused on statistical processes of traits without a direct tie to the underlying genetic patterns, instead using 'polygenes'[93]. QTL analysis uses DNA markers - restriction fragment length polymorphisms (RFLPs) followed by polymerase chain reaction (PCR) markers, that differed across populations to provide biological locus associations with the traits. The different types of QTL analyses are often named based on the pattern of [trait]QTL, such as eQTL for expression to locus (generally SNPs), methQTL for methylation to locus, etc[57]. With the development of better genome sequencing tools, many basic QTL processes, using markers, have been coupled with whole-genome analysis steps[121].

Genome Wide Association Studies (GWAS), one of the most often used cross-omic analysis method, finds statistical associations, often calculated as an odds ratio using a chi-squared or mixed model test, between SNPs and a phenotype - be it a separate biological layer, physical trait, or clinical outcome[181]. Transcription Wide Association Studies (TWAS) provide the same analysis process with gene expression rather than SNPs, as Epigenome Wide Association Studies (EWAS) do with methylation[98]. *Multi-phenotype association decomposition: unraveling complex gene-phenotype relationships* by Weighill, et al.(2019) [182] provides methods for GWAS analysis with multiple phenotypes, such as potential pleiotropy. GWAS studies have shown genomic influence on *Populus*

*trichocarpa* wood structure[28], callus formation[174], and leaf morphology[29]. microbeQTL analysis has shown that microbial communities effect metabolite production, also in *Populus trichocarpa*[164]. TWAS studies have shown that metabolites are transcriptionally regulated[45].

The 2021 publication *AtMAD: Arabidopsis thaliana multi-omics association database* by Lan, et al.[98] provides a collection of consolidated cross-omics analyses on a single multi-layer, many-sample data set in *Arabidopsis thaliana*, including eQTL, emQTL, Pathway-mQTL, Phenotype-pathway, GWAS, TWAS and EWAS. eQTL analysis was done with an additive linear model. A linear mixed model (GEMMA) was used to find associations between environment and eQTLs for eQTL-GWAS association. GWAS data was obtained from AraGWAS and LD used to map between eQTLs and GWAS results. Methylation sites were 'identified by root mean square test, using 1000 permutations', and emQTL was calculated as Pearson correlation between each methylation site and all genes. For TWAS, Pearson and Spearman correlation were calculated between each phenotype and all gene expression values.

## 1.5 Pre-processing Considerations

Pre-processing is an easily-overlooked, but extremely important step in model building[30]. Pre-processing includes any analysis prior to the main model analysis, up to and including other models in a pipeline. A solid understanding of how the model will use the data set is necessary to apply the concepts appropriately. Below are a few of the most relevant pre-processing methods for Random Forest (RF) use.

### 1.5.1 Outlier Detection

Outliers, values that significantly differ from the rest of the data, can heavily influence statistical and machine learning model results. This could be an error in recording or calculating the data point, or it could simply be a correct value that does not follow the pattern set by other values. If it is correct and the work being done with the data is not highly susceptible to outliers, then they should not be removed, as this would remove potentially

valuable information. However, if the methods used on the data are susceptible to outlier errors handling of outliers is a necessary step.

There are multiple statistical processes to determine which values are outliers, and multiple ways to handle the outliers once they have been identified. If the data roughly follows a normal distribution, the common process is to label any point farther out than two standard deviations from the mean as an outlier[102], indicating that it lies outside the grouping of 95% of the data. If a smaller number of outliers is needed, three standard deviations is used, this being outside 99.87% of the data grouping.

### MAD Median

Unfortunately, many data sets cannot be assumed to be normal, and outliers can influence both the mean and standard deviation. Methods such as Median Absolute Deviation about the Median (MAD Median) are more useful in many of these cases. The median is not affected by outliers like the mean[102, 145], making it a more reliable metric. The median absolute deviation (MAD) is indifferent to sample size and has a breakdown point (see, e.g., Donoho & Huber, 1983) of 0.5, meaning that the median absolute deviation results in a false value (infinity or null) only if more than half the data has been 'contaminated', set to infinity[102] - the highest possible outlier value that could distort calculations.

### Robust PCA

Robust principal component analysis (rPCA), unlike PCAs, are designed to appropriately handle outliers within the data, as well as identifying the outliers themselves[27]. There are four main methods for rPCA, *PcaCov*, *PcaGrid*, *PcaHubert*, and *PcaLocantore*, two of which have been specifically tested for sample outlier detection with RNASeq data, *PcaHubert* and *PcaGrid*[27].

### Outlier Replacement

Once a value is determined to be an outlier, a decision must be made on how it is handled. For many models, including RF, simply removing the value completely and leaving a 'gap' in the data is not a viable solution. First, it should be determined that the value is an

incorrect measure and cannot be simply replaced by the 'intended value', i.e. a mis-typed entry. Following that process, if the data the value is associated with follows a pattern, imputation can be used to replace the value with an appropriate maximum or minimum. If the data follows a somewhat-normal distribution, the value of two standard deviations above the mean (or median if using MAD) may be used for outliers higher than the distribution and two standard deviations below the mean for outliers below the distribution. Care should be taken when replacing an outlier with a value of zero as zero values can have highly varying meanings in different data sets.

### 1.5.2 Variance Testing

Variance measures how far data points are spread out. Formally, variance is the square of the standard deviation from the mean. The formula is written below, where $\sigma^2$ is the variance, N is the population size, and $\mu$ is the population mean[6].

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Low variance would be indicative of little fluctuation across the data. For methods such as Random Forests that rely on variance for calculations, testing for and removing features with low variance can help trim out unnecessary data and avoid spurious associations. By calculating the variance for each independent vector of data, some can be marked as unimportant for future calculations. Testing variance should be done after outliers are removed. If done in the reverse order, the outliers could influence the calculations, potentially increasing the variance enough to include the vector when it should be removed.

### 1.5.3 Correlated Features/Predictors

Correlation amongst features in a model can severely hinder the model's ability to function, specifically with machine learning algorithms like Random Forests[124]. If two features are highly correlated, the process of determining which feature is best for a split could result in a divided feature importance score - giving each of the features a partial score. This concept is

especially relevant in any analyses where linkage disequilibrium may cause single nucleotide polymorphisms (SNPs) to have lower importance than they should for a specific gene.

Some specialized methods do exist, such as RF-Recursive Feature Elimination[66], however they do not always handle high dimensional data well[41]. The most basic process, completing an all-to-all correlation analysis, such as with Pearson's correlation, may still be the most useful, by removal of one feature from each pair of highly correlated features prior to use with the RF.

### 1.5.4 Batch Effects

Batch effects are non-biological differences between batches of samples, especially data such as microarray experiments[115]. These effects can influence the models used to analyze the data, leading to associations and predictions that are not indicative of the underlying data but of the data processing. *Why Batch Effects Matter in Omics Data, and How to Avoid Them* by Goh, et al (2017)[64] provides a detailed breakdown of the different types of batch effects and how they influence different types of analyses.

A variety of methods exist to combat batch effects, however they tend to be very specific to the data type, or even the specific technology/machine. For data such as RNA-seq, which are well known to include batch effects, there is ComBat[84], ComBat-seq[199], BARA[65], SVASeq[101], and countless others. Special care should be taken to understand the caveats of each method, as not all batch correction methods work in a similar fashion and could fail due to specific structures within the data[127].

### 1.5.5 Sampling

For models where prediction is a factor, care must be taken when sample splitting for building and testing the model. For random data, simple random sampling may be sufficient, however, biological samples are often not independent, making this process unreliable[85]. These dependencies could be the result of population effects, which can be caused by experimental sampling biases, environmental similarities/differences, or many other confounding factors.

While it may be impossible to account for all of the possible factors when building a model, the more that are appropriately managed, the stronger the model.

For a basic prediction model, sets of samples are designated as 'training' or 'testing', and sometimes a separate set for 'validation'[30]. These sets are groups of samples that are used for only one part of the prediction process - either the building of the model (training) or the predication accuracy calculations (testing). If cross-contamination occurs, then the model creation becomes compromised - having what could be thought of as 'insider information' when scoring the prediction capacity on what should be an unknown set of testing data.

There a multiple methods to help balance the over-fitting that is possible with train/test splitting, when the model learns the intricacies of the training data so well that it is no longer generalizable. A well established method is k-fold cross validation[30]. For a k-fold method the entire data set is broken into $k$ subsets and a unique model is built using each of the $k$ subsets as the test set, leaving the rest to train the model. Randomly redistributing the samples into the sets and repeating the process can further alleviate biases that may exist. Once each model is built and tested, model accuracy measures can be implemented to score the overall accuracy of the model.

## 1.6 Network Biology

Conceptually, biological processes are often referred to as *pathways*. However, these pathways often intersect in many locations and include branching and repeating paths. Due to this nature, networks are often a more generalizable format for displaying the associations within these paths. Graph theory has a rich mathematical history that can provide many tools and insights into analyses that use these visualization structures.

### 1.6.1 Networks, Nodes, and Edges

A graph, or network, consists of vertices (nodes) and edges, where each edge is a relationship between at least two vertices[24]. These relationships are often described with an adjacency matrix, provided a grid layout of every edge between every pair of nodes. Mathematically,

for a graph $G = (V, E)$, the adjacency matrix is an $nxn$ matrix where:

$$a_{v,u}^{G} = \begin{cases} 1, \text{if} v, u \in E, \\ 0, \text{otherwise} \end{cases}$$

where $v$ and $u$ are vertices in $G$ and $n$ is the size of $v$[149].

For graphs where the strength of the relationship between the nodes may change, weighted edges provide a numeric score the connection between the connected vertices. In the adjacency matrix for a weighted graph, the 1 value is replaced with the corresponding weight for each edge. For graphs where the relationship between $e_i$ and $e_j$ is not symmetric, i.e. edge $e_i - e_j \neq e_j - e_i$, directed edges, symbolized with arrows, indicate that the relationship is not mutual. Often with predictive models, or independent and dependent variables, use of directed edges will point *from* an independent variable/vertex *to* a dependent variable/vertex.

Simple graph metrics can provide quick views of patterns and outliers. The degree of each node shows the number of edges it is connected to and exceptionally high or low degrees may highlight nodes of interest. Betweenness centrality of each node provides a score for how often each node is included in the shortest path between any two nodes in the network. While nodes with high degrees may have a high betweenness centrality, for weighted edges, the weight could be used as a 'cost' for each edge, pushing paths away from nodes with high edge weights, or inversely, a higher weight is a better edge.

### 1.6.2 Multiplex and Heterogeneous Networks

Networks are not limited to a single node or edge type. A network that consists of multiple types of edges between the same nodes is known as a multiplex network, whereas a network with multiple types of nodes is a heterogeneous network. A network may contain both multiple types of nodes and edges, and would be both a multiplex network and heterogeneous network.

Multiplex networks are useful when trying to provide multiple types of associations between entities. For a network of genes, one set of edges may represent co-expression, while

a second could represent regulation. By providing both edge types within the same network, patterns that exist across both types may become visible. Alternatively, heterogeneous networks are useful for providing associations between multiple types of entities (nodes). This could be a network where one set of nodes is genes and another is proteins, with edges between them representing gene inclusion in a process that produces the connecting protein. Networks that are both multiplex and heterogeneous could then have all of the above attributes, such as multiple edge types between the genes as well as edges from genes to proteins. In figure 1.2, network a) is a multiplex network with a single type of node and multiple types of edges and, network b) is a heterogeneous network with one (directed) edge type and two types of nodes, and network c) is both a multiplex and heterogeneous network, containing both multiple node and edges types.

While extremely useful for displaying multitudes of information in a cohesive manner, both multiplex and heterogeneous networks are less often used due to their complexity. As such, many of the following methods and processes assume basic network topology and may or may not function appropriately with more complicated topological interactions and entities.

### 1.6.3 Clusters and Cliques

Graph clustering is the process of separating the vertices of a graph into 'related' subgraphs or groups[149]. Clustering is a useful tool when an analysis is looking to find related nodes based on their shared associations (edges). Cliques are independent, complete subgraphs[149], where there is an edge between every pair of vertices. Within clustered networks, cliques are indicative of strong shared associations or processes within a set of nodes. There are many different types of clustering techniques, often designed with very specific network structure expectations and varying computational requirements.

#### *k*-means Clustering

$k$-means clustering[106] is a popular method for clustering. The data points are divided into $k$ clusters based on their distance to the nearest mean or cluster center. The main drawback

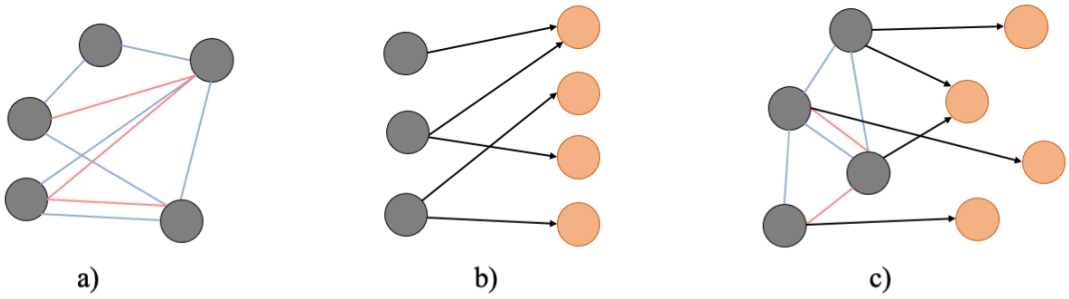



**Figure 1.2:** Three different networks showing different patterns of node and edge types, with each unique color representing a type of node or edge. Network a) is a multiplex network, with multiple edges types but a single node type. Network b) is a heterogeneous network, with multiple node types and a single (directed) edge type. Network c) is both a multiplex and heterogeneous network, with both multiple types of nodes and multiple types of edges.

for this method is the necessity for the user to provide $k$, when many clustering problems do not have a predefined number of clusters to expect. This process can be somewhat alleviated by repeating the process for a set of $k$s and choosing the clustering that best fits the needs of the user, be it a minimum or maximum number of clusters, points per cluster, or distance between points within each cluster. Note that this clustering works with weighted networks, but not directed. Directed edges must be assumed to be undirected for this algorithm.

#### Markov Clustering

An unsupervised clustering method, Markov Clustering[176, 51] (MCL), simulates stochastic flow within the network, with the understanding that higher weights between edges indicate a higher chance of flow between the nodes that share those edges. The method uses matrix algebra on the adjacency matrix to 'grow' strong edges and 'shrink' weak edges repeatedly. While MCL does avoid some of the pitfalls of $k$-means clustering due to the lack of a required cluster number input, MCL does provide multiple tuning parameters, such as the inflation factor, that can drastically change the resulting clusters. Nor does it appropriately handle directed edges. However, the author provides some guiding steps on how best to use the algorithm with directed edges - namely some processes to determine if certain directed edge weights are appropriate for mapping to comparable undirected weights.

### 1.6.4 Network Walking

While useful for determining subsections of networks that are highly related, clustering is limited in usefulness outside of this one aspect. An alternative to clustering is network walks, where edges can be thought of as paths or walkways between nodes, with weights as either scores to accumulate, or tolls to avoid. For exploratory analysis and finding associations to a specific node or set of nodes of interest, network walks can be very useful at dispersing through a network and providing a likelihood for other nodes to be visited. Mathematically, network walks are a specific type of markov chain (MC), where a markov chain is defined as a set of states with probabilities of transition from one state to another. The nodes in a network are the states and the edge weights can be either used as, or transformed into,

probabilities of transition, with no edge between two nodes being interpreted generally as a zero weight and zero chance of transition[5]. Often, the transition matrix is a direct mapping from the adjacency matrix used to define the network.

### Random Walk

A random walker takes the markov chain process a step further, and conceptually sends a 'walker' out through the network, randomly moving based on the transition probabilities, to see where it ends up[114]. However, to reach a steady distribution of end locations and probabilities of different nodes visited would require an infinite number of walks. Instead, Random Walks use a set of probability states to simulate an infinite number of simultaneous walkers, where the probabilities indicate the chance of the walkers being in any one state, or at a specific node.

A new probability state, representing the many walkers taking a single step, is calculated by multiplying the current probability state by the transition matrix. After a set of steps, the walkers may end up in a *sink*, where there are many paths with high probability to reach a node, but few or no probabilities of leaving the node, leaving the walkers stuck and unable to travel farther.

### Random Walk with Restart

Random Walk with Restart (RWR) uses this process with an additional probability of the walker returning to the starting node at any step. By using seeds with restart probabilities, the end scores for each node in the network can be viewed as a proximity to the seeds in the network space[130]. This is included in the model by updating the transition probabilities to include a nonzero chance of returning to a given set of starter seeds; the updated transition matrix can then be used as described previously. This process helps the walker avoid sink states where further movement is impeded. Using the restart probability, the random walk process will converge to a steady state after a number of runs, providing a final probability state for a given set of seeds.

### 1.6.5 Thresholding

Thresholding is a process of removing edges from a graph to introduce sparsity to reveal meaningful network topologies, and may or may not result in clusters. The intent is to prune out edges that may be irrelevant for further analysis, which could be due to low confidence, low score, etc. This is especially important when further computational analysis will be done with the network, as this limits the computations necessary in the next steps[133]. As with clustering, there are many different options for graph thresholding, all of which have pros and cons for their use.

#### Hard Thresholding

For basic hard thresholding, a cut-off value is determined through some means, and every edge with a value at or above the threshold is changed to a one and every value below the cut-off is changed to a zero. A new unweighted network is created by using only the edges with values of one. This method can be reversed to include only values below a cut-off depending on what the network's intended use. The formal equation is as follows[75], where $\tau$ is the cut-off value, $a_{ij}$ is the binary adjacency matrix for the network, and $s_{ij}$ is the original network:

$$a_{ij} = \begin{cases} 1 & \text{if } s_{ij} \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Hard thresholding's advantage is the creation of a simple, unweighted network. The disadvantage is that any value just below the cut-off, no matter how close, becomes zero and means nothing in the new network. For data that has a potential for large amounts of noise, such as genomic data, this process could remove potentially valuable information and connections.

### Soft Thresholding

With soft thresholding, rather than change edge values to ones or zeros, every value is taken to a power higher than one, $\beta$, set by the researcher. This process punishes weak connections and strengthens strong connections. The formal equation is as follows:

$$a_{ij} = s_{ij}^{\beta}$$

Soft thresholding overcomes the problem of missing 'just below the cut-off' values that hard thresholding faces. However, soft thresholding itself does not result in fewer edge values, and generally must be followed with a hard thresholding pass to remove those deemed too low. Both of these methods anticipate boundaries or rules from the domain science or other methods to determine the cut-off value or $\beta$ value. An arbitrary choice may remove potentially important connections or increase false positives[133].

### Spectral Thresholding

Spectral thresholding focuses on spectral graph theory based clustering, and uses this to determine a suitable threshold. Spectral graph theory revolves around graphs in regards to their matrix properties, such as eigenvalues and eigenvectors in adjacency matrices and Laplacian matrices[133]. This method specifically focuses on calculating the number of 'nearly-disconnected' sections of the network. The correct cut-off will maximize the number of these components and minimize the number of edges connecting these components.

## 1.6.6 Current Uses in Biology

Many graph theory methods are available for use provided the biological data can be encoded as adjacency matrices and nodes and edges. As such, networks are used in many different ways in biological research. Any type of broad association or interaction between biological entities could be visualized in a network structure; the process for how the network is analyzed for biological insight will depend on the type of associations, the edge weighting scheme, and multitude of topological aspects for the network.

A well-cited use of graph theory in biology is the Weighted Gene Correlation Network Analysis (WGCNA) method[194]. WGCNA, as the name implies, creates and uses correlation networks of genes. The correlation metric used can vary, but is often simply Pearson's correlation, sometimes with a soft-thresholding effect applied afterward[196]. Many gene co-expression studies have been published, either explicitly using this method, or simply using a very similar process for analysis[195, 76, 201, 186, 146], where the correlation aspect between the genes is the co-expression level, also often calculated using correlation metrics such as Pearson's Correlation.

Gene Regulatory Networks, as used for the GENIE3 Random Forest method described in section 1.2.3, are another biological network concept used often[42, 88, 43, 52, 178, 118]. Where the edges in a co-expression network would be some value of co-expression or correlation between two genes, a GRN edge may simply be a binary, directed edge indicating one gene's regulatory effect on another. Weights may be added to indicate the strength of the relationship or the reliability of the edge itself. Networks can also be used to visualize protein-protein interactions[126, 79, 10, 3], where each node is a protein and the edge weights are a confidence score for interaction, with edges being directed or undirected as needed. STRING[157] provides a database of experimentally measured and predicted protein-protein interactions across organisms.

As machine learning techniques have become more broad-reaching and have entered the field of biology, ML-based biological network methods are being developed. The GRN method used by GENIE3[78], with Random Forests, is one such method. Instead of a regulatory aspect used as the edge weights, the genes are connected by feature importance scores produced from the Random Forest. iRF-LOOP[32], similarly to GENIE3, applies an RF-based method to create networks of feature importance scores. New deep learning methods are also being developed or expanded for biological networks, such as GNE - a deep learning method to learn topological and gene expression properties from gene interaction networks[91].

## 1.7 Conclusion

Biological system analysis is seeing huge growth in available data types and quantities that require new methods, new systems, and comprehensive analyses to distill the biological patterns that they may explain. Explainable-AI and Random Forest methods provide an avenue for analysis of these data that can account for the complex nature of the many different interactions, as well as providing human interpretable results. As these quantities of data continue to grow, intentional method design that takes advantage of large memory and high performance computing systems will become increasingly necessary. As results require larger amounts of time to process due to growing sizes, specific considerations for sampling biases, outlier removal, and overall data preprocessing will also be more relevant. Systems biology analysis using multi-omic methods will continue to see growth as more omics layers see high throughput data collection developments and improvements, necessitating visualization and analysis capabilities that can handle these heterogeneous data sets. Together these diverse aspects provide a guiding framework for future systems biology analysis.

# Chapter 2

# A High-Performance Computing Implementation of Iterative Random Forest for the Creation of Predictive Expression Networks

This chapter is published as Cliff A, Romero J, Kainer D, Walker A, Furches A, Jacobson D. A High-Performance Computing Implementation of Iterative Random Forest for the Creation of Predictive Expression Networks. Genes. 2019; 10(12):996. https://doi.org/10.3390/genes10120996, distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/). Conceptualization was completed by AC, JR, DK, and DJ. Methodology was completed by AC, JR, and DK. Software was completed by AC and JR. Validation was completed by AC, JR, DK, and AW. Formal analysis was completed by AC and JR. Investigation was completed by AC. Resources were provided by DJ. Data curation was completed by AC, JR, AW, and AF. Writing - original draft was completed by AC. Writing - review and editing was completed by AC, JR, DK, AW, and DJ. Visualization was completed by AC. Supervision, project administration, and funding acquisition was provided by DJ.

## 2.1 Abstract

As time progresses and technology improves, biological data sets are continuously increasing in size. New methods and new implementations of existing methods are needed to keep pace with this increase. Presented here is a high-performance computing (HPC)-capable implementation of Iterative Random Forest (iRF). This new implementation enables the explainable-AI eQTL analysis of SNP sets with over a million SNPs. Using this implementation, we also present a new method, iRF Leave One Out Prediction (iRF-LOOP), for the creation of Predictive Expression Networks on the order of 40,000 genes or more. Also shown is a comparison of the new implementation of iRF with the previous R version and an analysis of its time to completion on two of the world's fastest supercomputers, Summit and Titan. Also shown is iRF-LOOP's ability to capture biologically significant results when creating Predictive Expression Networks. This new implementation of iRF will enable the analysis of biological data sets at scales that were previously not possible.

## 2.2 Introduction

Due to innovation in the areas of genome sequencing and 'omics analysis, biological data is entering the age of big data. As opposed to other fields of research, biological data sets tend to have large feature quantity, but much smaller sample counts, such as in a GWAS population where there are typically hundreds to thousands of genotypes, but millions of SNPs. The number of independent features of biological systems is large, and would require many lifetimes of the entire scientific community to sufficiently study [72]. The ability to determine which features are influential to a particular phenotype, be it SNPs, gene expression, or interactions between multiple molecular pathways, is essential in reducing the full feature space to a subset that is feasible to analyze.

Many methods exist to determine feature importance and feature selection, such as Pearson Correlation, Mutual Information (MI), Sequential Feature Selection (SFS) [23], Lasso, and Ridge Regression. Random Forest [19] is a commonly used machine learning method for making predictions, and while not classically defined as a feature selection method, it is useful in scoring feature importance. During the training phase, decision trees are built, where a subset of features is examined at each decision point and the one that best divides the data is chosen. Feature importance is calculated for each feature based on its location and effectiveness within the tree structures. By using random subsets of the training data for each tree and considering random features for each decision point, Random Forest prevents over fitting. Because of the nature of decision trees, the importance of any chosen feature is inherently conditional on the features that were chosen previously. In this way the Random Forest can account for some of the interconnected dependencies that occur in biological systems. As a non-linear model, Random Forest has been applied to a range of biological data problems, including GWAS, genomic prediction, gene expression networks and SNP imputation [26].

Iterative Random Forest [13] (iRF) is an algorithmic advancement of Random Forest (RF), which takes advantage of Random Forests ability to produce feature importance and produces a more accurate model by iteratively creating weighted forests. In each iteration, the importance scores from the previous iteration are used to weight features

for the current forest. Until now, iRF was implemented solely as an R package [12]. While useful for small projects, it was not designed for big data analysis. This paper describes the process of implementing a high-performance computing-enabled iRF, using MPI (Message Passing Interface) [Walker and Dongarra]. This new implementation enabled the creation of Predictive Gene Expression Networks with 40,000 genes and quickly completed the feature importance calculations for 1.7 million *Arabidopsis thaliana* SNPs in relation to a gene expression profile, as part of a genome-wide explainable-AI-based eQTL analysis.

## 2.3 Materials and Methods

### 2.3.1 Random Forest and Iterative Random Forest Methods

The base learner for the Random Forest (RF) and Iterative Random Forest (iRF) methods is the decision tree, also known as a binary tree. A decision tree starts with a set of data: samples, features, and a dependent variable. The goal is to divide the samples, through decisions based on the features, into subsets containing homogeneous dependent variable values, generally following the CART (Classification and Regression Trees) method [20], where each decision divides one node into two child nodes. This is a greedy algorithm which will continue to divide the samples into child nodes, based on a scoring criterion, until a stopping criteria is met. Decision trees are weak learners and tend to over-fit to the data provided.

A Random Forest is an ensemble of decision trees. However, the trees in a Random Forest differ from standard decision trees in that each tree starts with a subset of the samples, chosen via random sampling with replacement. Also differing from standard decision trees, the features being considered at each node are a random subset, with the number of features provided by a parameter. Once a forest has been generated, the importance of each feature can be calculated from node impurity, such as the Gini index for classification or variance explained for regression, or permutation importance. Unlike a single decision tree, a Random Forest is a strong learner, because it averages many weak learners and avoids putting too

much weight on outlier decisions. The number of trees in a forest is a parameter that is chosen by the user and influences the accuracy of the model.

Iterative Random Forest expands on the Random Forest method by adding an iterative boosting process, producing a similar effect to Lasso in a linear model framework. First, a Random Forest is created where features are unweighted and have an equal chance of being randomly sampled at any given node. The resulting importance scores for the features are then used to weight the features in the next forest, thus increasing the chance that important features are evaluated at any given node. This process of weighting and creating a new Random Forest is repeated $i$ times, where $i$ is set by the user. Due to the ability to easily follow the decisions that these models make, they have been deemed explainable-AI (X-AI) [72], which differs from many standard machine and deep learning methods.

For both Random Forest and Iterative Random Forest, the total number of features, samples, trees, and iterations (for iRF) all influence run time to differing degrees. Most of the computation time is spent creating the decision trees, so run time scales with the number of trees. The number of samples influences the number of decisions within a tree needed to divide the data into homogeneous subsets. This influences the number of nodes created in a tree, and thus the run time per tree. Furthermore, the number of features influences the amount of time required to find the feature that best divides the samples at any given node. Finally, for iRF, a whole new forest must be generated for each iteration, though the run time for subsequent forests tends to diminish as many feature weights are set to zero. The number of iterations allows the user to find a balance between over- and under-parameterization of the model, by progressively eliminating features.

### 2.3.2 Implementation of iRF in C++

We used Ranger [188], an open source Random Forest implementation in C++, as the core of our iRF implementation. Ranger already implements the decision tree and forest creation aspects, but implements neither the communication necessary for running on multiple compute nodes of a distributed HPC system, nor the iterative aspect of iRF.

Within our implementation, each decision tree is initialized with a random subset of the data sample vectors and is then built in an independent process. Groups of trees (sub-forests)

are built on compute nodes and then sent to a master compute node that aggregates them into a full Random Forest. This is done by giving each sub-forest a randomly generated seed number which determines the random subset of data each tree in a sub-forest uses, allowing for a higher likelihood of unique random data subsets on each tree. Once in the form of a single Random Forest, Ranger's functions for forest analysis are used, including feature importance aggregation.

On a distributed system, where parts of the forest are created on different compute nodes, the aggregation of results requires most of the inter-node communication. This process relies on MPI (the Message Passing Interface), an internode-communication standard for parallel programming, and Open-MPI [61], an open source C library containing functions that follow the MPI standard.

To implement $i$ iterations, the forest creation and aggregation is performed $i$ times. After the completion of each iteration, the feature weights are written to a file. At the beginning of the next iteration, this file is read into an array that is used to create the weighted distribution from which features are sampled during the decision process at each node on each tree. This process, while potentially slow due to the file I/O, uses the preexisting functionality of weighted sampling from Ranger.

### 2.3.3 iRF-LOOP: iRF Leave One Out Prediction

Given a data set of $n$ features and $m$ samples, iRF Leave One Out Prediction (iRF-LOOP) starts by treating one feature as the dependent variable (Y) and the remaining $n - 1$ features as predictor matrix (X) of size $m\ x\ n - 1$. Using an iRF model, the importance of each feature in X, for predicting Y, is calculated. The result is a vector, of size $n$, of importance scores (the importance score of Y, for predicting itself, has been set to zero). This process is repeated for each of the $n$ features, requiring $n$ iRF runs. The $n$ vectors of importance scores are concatenated into an $n\ x\ n$ importance matrix. To keep importance scores on the same scale across the importance matrix, each column is normalized relative to the sum of the column. The normalized importance matrix can be viewed as a directional adjacency matrix, where values are edge weights between features. See Figure 2.1 for a diagram of this

process. Due to the nature of iRF, the adjacency matrix is not symmetric as feature A may predict feature B with a different importance than feature B predicts feature A.

### 2.3.4 Big Data: Showing the scale of iRF with *Arabidopsis thaliana* SNP Data

A typical use case of iRF, with a matrix of features and a single target vector of outcomes, becomes comparable to an X-AI-based eQTL analysis, when the matrix of features is a set of single nucleotide polymorphisms (SNPs) and the dependent variable vector is a gene's expression measured across samples. This analysis determines which set(s) of SNPs are important to variation in the gene's expression.

We obtained *Arabidopsis thaliana* SNP data from the Weigel laboratory at the Max Planck Institute for Developmental Biology, available at https://1001genomes.org/data/GMI-MPI/releases/v3.1/. We filtered the SNPs using bcftools 1.9 [103] to keep only those that were biallelic, had a minor allele frequency greater than 0.01, and had less than ten percent missing data across the population. This resulted in a set of 1.71 million SNPs for 1135 samples, from the original 11.7 million SNPs.

We obtained *Arabidopsis thaliana* expression data [90] from 727 samples and 24,175 genes. Of these 727 samples, 666 samples were also present in the SNP data set. The vector of gene expression values for gene AT1G01010 for those 666 samples was used as the dependent variable for iRF. The feature set was the full set of 1.71 million SNPs, for the same 666 samples. While this is not many samples, this is clearly many features.

The C++ iRF code was run using these data as input with five iterations, each generating a forest containing 1000 trees. The number of trees was chosen as a value close to the square root of the number of features (a common setting for this parameter), where each feature has a 95% chance of being included in the feature subset within the first two layers in at least one tree. This helps to guarantee that all features are considered at least once across an iteration. HPC node quantities of 1, 2, 5, and 10 were used to show run time changes as the amount of resources increases.

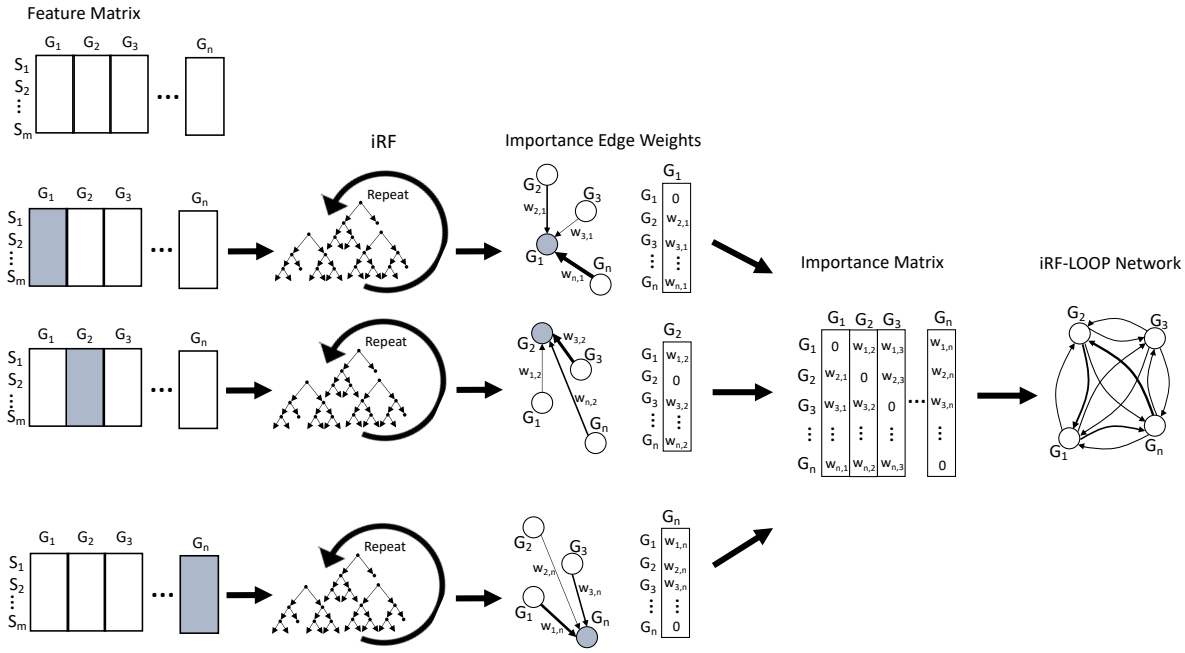



**Figure 2.1:** The diagram shows the process of iRF-LOOP for a set of Expression profiles, creating a Predictive Expression Network. Each gene is independently treated as the target for an iRF run, with all other genes as predictors. iRF provides importance scores of each predictor gene, and creates network edge weights between target and predictors. These importance scores are then combined into an edge matrix, providing a value for each possible connection, from which a network can be generated. Generally, the weights are thresholded at some value, determined through other means, and only edges with large enough weights are included in the final network. Due to the inherent directionality of a prediction, the edges are weighted, and not likely to be symmetric.

Due to the large number of SNPs, the full data could not fit in memory on a standard laptop, for use in the R iRF program. Instead, small subsets of features of sizes 2,000, 1,000, 500, 100 and 50 features were run, each using the full 666 samples and the same AT1G01010 gene expression dependent variable. Each feature set was run three times and averaged to account for the inherent stochasticity in the algorithm. Only one run was performed for each parameter set on Summit with the full data set due to limited compute time availability.

### 2.3.5 Using iRF-LOOP to Create Predictive Expression Networks

Given a matrix of gene expression data, there are a multitude of approaches for inferring which genes potentially regulate the expression of other genes, ranging in complexity from pairwise Pearson Correlations to advanced methods such as Aracne [118], GENIE3 [78], and dynamic Bayesian networks (DBNs) [134]. Due to the large number of features and the complexity of the interactions between them, Random Forest-type approaches are well suited to this task.

We applied iRF-LOOP to a matrix of gene expression data measured in 41,335 genes across 720 genotypes of *Populus trichocarpa* (Black cottonwood). The RNAseq data [198] from were obtained from the NCBI SRA database (SRA numbers: SRP097016–SRP097036; www.ncbi.nlm.nih.gov/sra). Reads were aligned to the *Populus trichocarpa* v.3.0 reference [173]. Transcript per million (TPM) counts were then obtained for each genotype, resulting in a genotype–transcript matrix, as referenced in [60]. The adjacency matrix resulting from iRF-LOOP represents a Predictive Expression Network (PEN) where a directed edge (AB) between and two genes (A and B) is weighted according to the importance of gene A's expression in predicting gene B's expression, conditional on all other genes in the iRF model. We removed zeros and produced four thresholded networks, keeping the top 10%, 5%, 1%, and 0.1% of edge scores, respectively.

To determine the biological significance of the PEN produced by the iRF-LOOP, we compared each of the four thresholded networks to a network of known biological function, created from Gene Ontology (GO) annotations. GO is a standardized hierarchy of gene descriptions that captures the current knowledge of gene function. It is accepted that there is both missing information and some error, nevertheless it is useful as a broad truth set

for large sets of genes. We calculated scores for each network by intersecting with the GO network. We then evaluated the probability of achieving such scores relative to random chance by creating null distributions of scores produced by random permutation of the iRF-LOOP networks and calculated $t$-statistics for each threshold.

We created a Populus trichocarpa GO network for the Biological Process (BP) GO terms, using annotations from [83]. Genes are connected if they share one or more GO term. Due to the hierarchical nature of the Gene Ontology, a term shared by only a few genes, indicates a very detailed and specific association, where the associations shared by many genes are generally broad categorizations. The edge weight between two genes equals $1/(n-1)$, where $n$ is the number of genes with the shared term. This weighting attempts to balance the scoring metric so that correctly identifying rare edges is not out weighted by simply capturing many generic GO terms. Genes that share a GO term create connected sets of genes, where each gene has an edge with each other gene in the set. A predicted network's intersect score is a summation of all GO edge weights from edges that appear in both the GO and predicted networks. For example, if the predicted network has an edge between gene A and gene B, and the GO network also has an edge between gene A and B with a weight of X, then the intersect score would increase by X. If two genes share more than one GO term, then the largest weight is used for the edge. To avoid edges between very loosely associated genes, only GO terms with less than 1000 genes were used for this analysis. The resulting network provides the relationship between genes that share some level of known functionality.

To generate the null distribution of intersect scores for each thresholded PEN, the node labels of each network were randomly permuted 1000 times, and each random network was scored against the GO network. From these null distributions, $t$-statistics were calculated for the score of each thresholded Predictive Expression Network.

For comparison purposes, the noted GO scoring process was also done on Pearson's Correlation-generated networks, creating co-expression scores. The input data was the same expression data, and Pearson's Correlation was calculated for each pair of genes. The top percent of correlation values were then analyzed with the GO scoring process, and compared to the each of the edge scores from the corresponding PENs.

### 2.3.6 Comparison of R to C++ Code

To compare the original iRF R code to the new implementation, both were run on a single node of Summit with a variety of running parameters. These parameters included all combinations of 100, 1000, and 5000 trees and 1, 2, 3, and 4 threads for 1000 features. All combinations were run three times and the scores were averaged. Due to the R code's doParallel [35] back-end not being designed for an HPC system, the R code was limited to a single CPU on a node with up to four independent threads. For consistency, the new implementation was limited to the same resources. A subset of the *Populus trichocarpa* expression data mentioned above was used as the feature set.

### 2.3.7 Computational Resources

The computational resources used in this work were Summit, Titan, and a 2015 MacBook Pro laptop. Summit is an Oak Ridge Leadership Computing Facility (OLCF) supercomputer located at Oak Ridge National Laboratory (ORNL). It is an IBM system with approximately 4,600 nodes, each with two IBM POWER9 processors, each with 22 cores (176 hardware threads) and six NVIDIA VOLTAV100 GPUs, and 512 GB of DDR4 memory. Titan was a former OLCF supercomputer, recently decommission, and was a Cray system that had approximately 18,688 nodes, each with a 16-core AMD Opteron 6274 processor and 32 GB of DDR3 ECC memory. The 2015 MacBook Pro has a 3.1 GHz Intel Core i7 Processor and 16 GB of DDR3 memory.

## 2.4 Results

### 2.4.1 Comparison of the R to C++ Code

Previously, the only published iRF code existed as an R library. This library uses R's 'doParallel' functionality, generally allowing for multi-core thread parallelism on shared memory CPUs. However, this system did not function on Summit, so our analysis was limited to running differences on a single Summit CPU.

To compare the R code to the C++ code, both programs were run on a single CPU on Summit. While this is a small resource set, it was sufficient in showing trends and making comparisons between the two implementations. Figure 2.2 shows the time to completion as the number of threads increases. As the number of threads that the runs are spread over increases, both implementations decrease in run time. However, for the 5000 tree runs for 1, 2, and 3 threads the R code was unable to complete in the 2 h time limit set by the Summit system. This is a good indication that these runs were not efficient enough, as the C++ implementation runs were able to complete. Even though the R implementation uses C++ and Fortran functions internally, it is likely that the overhead of using R and the associated I/O bottlenecks significantly impacts total run time. Similarly, as seen in Figure 2.3, as the number of trees increases, the R code takes significantly longer than the C++ code to complete. Together, these figures show that in a one-on-one comparison using appropriate resources, the C++ implementation is more efficient than the R implementation, and can handle more computations per unit of time.

### 2.4.2 Scaling Results for Big Data: *Arabidopsis thaliana* SNPs to Gene Expression

To show how well our implementation of iRF handles large feature sets, a set of 1.7 million SNPs from *Arabidopsis thaliana* was used to predict the expression of gene AT1G01010. Figure 2.4 shows the times to completion for the four thread quantities tested on Summit (160 threads per compute node). Our implementation was easily able to handle the data set for all thread quantities and finished in reasonable amounts of time. It is worth noting that there were diminishing returns as the number of nodes gets close to 10 (1600 threads) since the number of trees per node at this scale would only be 100 (1000 trees total spread over 10 nodes) and did not use the resources to its fullest potential. For a larger feature set, a larger number of trees would be advised, and a larger number of nodes could be used more efficiently.

To try to determine approximately how long this calculation would take on a standard laptop, multiple smaller runs were completed. Figure 2.5 shows the run times for multiple

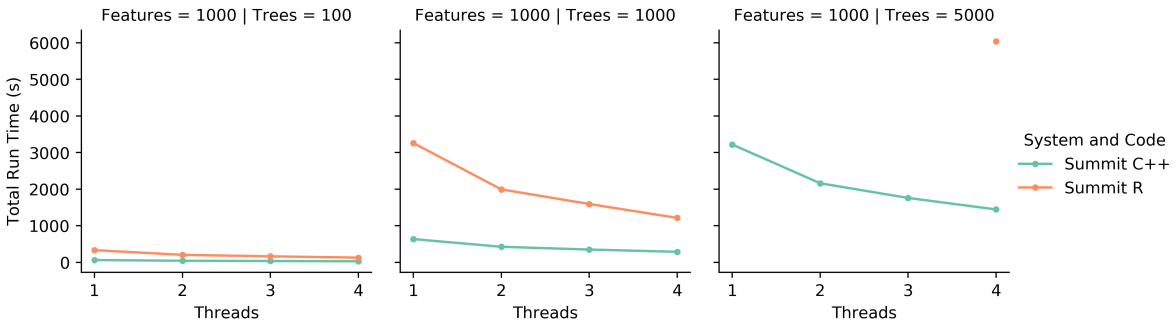



**Figure 2.2:** Each of these graphs shows the total run time as the number of threads increases. Both the R code and C++ code were run on Summit. Note for 5000 trees, the R implementation failed to complete using less than 4 threads.

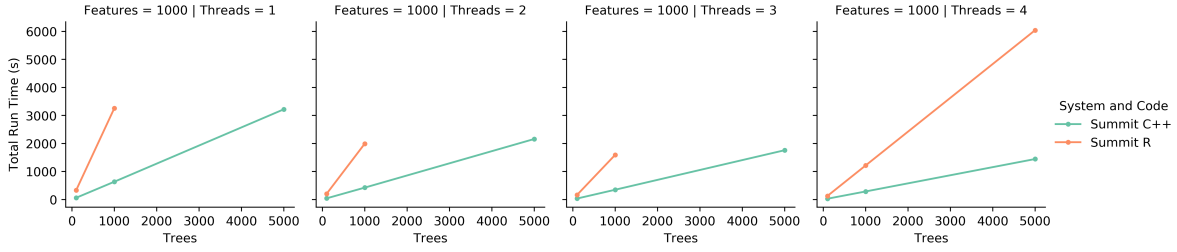



**Figure 2.3:** These graphs show a different orientation of the data from Figure 2.2. Each graph shows the total run time as the number of trees increases, while the number of features and number of threads stays constant. Due to the 5000 tree runs not completing with the R code for 1, 2, or 3 threads, those graphs are missing points.

feature amounts for the R iRF code on one CPU of a 2015 MacBook Pro laptop. The linear fit, while not perfect, is accurate enough for a rough estimate for larger feature sizes. Using the provided equation, the 1.7 million feature set run on Summit would take approximately 33 days to complete on a laptop, given that the system had enough memory to contain the data set and results, which most standard laptops do not have. When compared to the approximately 40 min required on 5 nodes of Summit, it is easy to see what a difference these resources, and programs that can use them, can make.

### 2.4.3 Predictive Expression Networks

We used iRF-LOOP to produce Predictive Expression Networks for *Populus trichocarpa*. Figure 2.6a shows the run time results for the C++ code for all varying numbers of threads and trees, for one of the approximately 40,000 iRF runs within an iRF-LOOP. Figure 2.6b shows the total run time as the number of threads increases on Summit and Titan for the C++ code, showing that iRF works comparably on different system architectures. Due to the architecture differences, Summit nodes can independently run 160 threads simultaneously while Titan nodes could only run 16 threads. Summit (in red) had a harder time with larger data on a single thread, but both systems function well as the number of threads increases to appropriate numbers for general uses cases. The full graph of all parameters comparing time to completion on Summit and Titan is available in Appendix figure 1.

Table 2.1 shows the number of edges and nodes in the four resulting thresholded PENs and the co-expression comparison networks. The GO network that was generated to analyze the PENs contained 16,836 nodes and 3,274,574 edges. Figure 2.7 shows a small example of the intersected networks with a calculated intersect score.The iRF-LOOP edges that do not have corresponding GO network edges are important for biological discovery. These edges do not necessarily represent 'wrong' results, but rather interactions found using iRF that are not listed in the GO network. These edges can be used for hypothesis generation of gene interactions, such as regulation, as well as putative gene functions.

Also shown in Table 2.1 are the mean and standard deviation for the null distribution of the random permutations for each thresholded PEN and co-expression comparison network. The null distribution and intersect score for the top 0.1% PEN network is shown in Figure 2.8

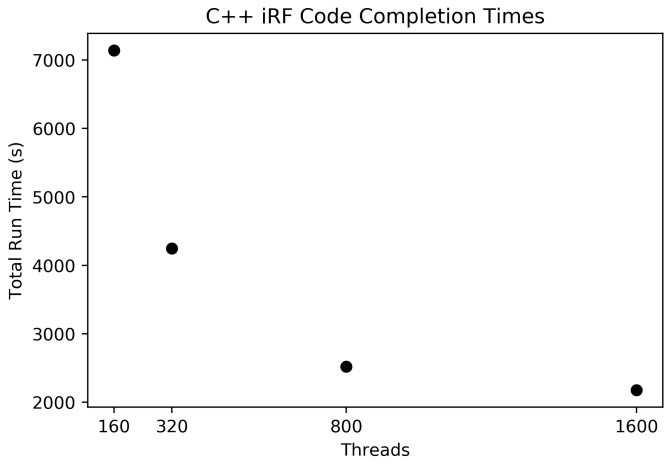



**Figure 2.4:** The graph shows the run times for four different compute node quantities, each completing 1000 trees for the 1.7 million SNPs.

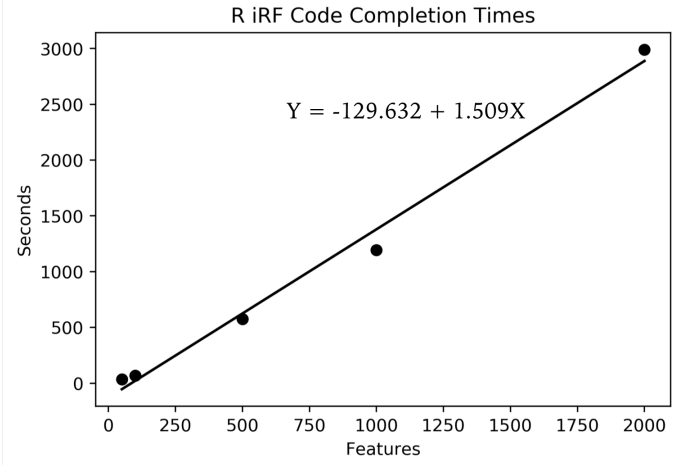
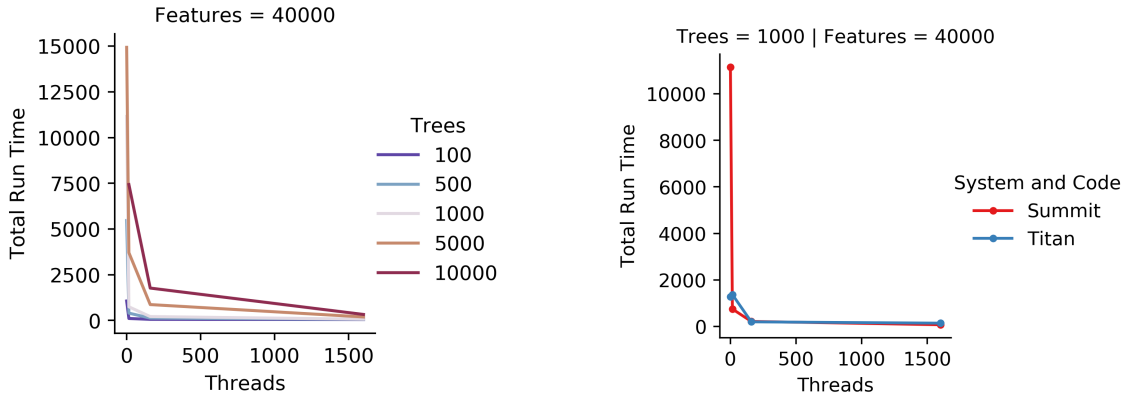



**Figure 2.5:** The graph shows the run time for five different feature sizes, on a single CPU of a standard MacBook Pro laptop. Each point represents the average of three runs. A linear regression was fit, with the equation shown. The fit is not perfect, but is enough to indicate that the run time increase approximately linearly in comparison to the number of features.




(**a**) (**b**)

**Figure 2.6:** Graph (**a**) provides the total run time for the C++ code on Summit, with various tree and thread counts, for 40,000 features. Graph (**b**) provides a comparison of the C++ code on Summit and Titan, two HPC systems. For both graphs, run time is in seconds.

**Table 2.1:** The table provides the graph results for the 4 thresholded Predictive Expression Networks for xylem, as well as the co-expression comparison networks. The listed mean and standard deviation are for the corresponding null distributions. The $p$-values for the listed $t$-statistics were effectively zero.

| **Network** | **Nodes** | **Edges** | **Intersect Score** | **Null Dist Mean** | **Null Dist s.d.** | ***t*-statistic** |
|---|---|---|---|---|---|---|
| 10% PEN | 39,349 | 14,663,754 | 981.66 | 252.82 | 4.78 | 152.36 |
| 5% PEN | 39,349 | 7,331,877 | 678.59 | 126.28 | 3.09 | 178.48 |
| 1% PEN | 39,349 | 1,466,375 | 295.80 | 25.28 | 1.38 | 196.05 |
| 0.1% PEN | 38,872 | 146,637 | 103.44 | 2.53 | 0.41 | 245.85 |
| 10% P.C. | 34269 | 31,182,857 | 1287.97 | 535.84 | 19.11 | 39.36 |
| 5% P.C. | 29,372 | 15,591,428 | 824.31 | 268.24 | 11.37 | 48.89 |
| 1% P.C. | 19,982 | 3,118,285 | 282.40 | 53.65 | 3.57 | 64.12 |
| 0.1% P.C. | 6,258 | 311,828 | 43.49 | 5.37 | 1.06 | 35.89 |

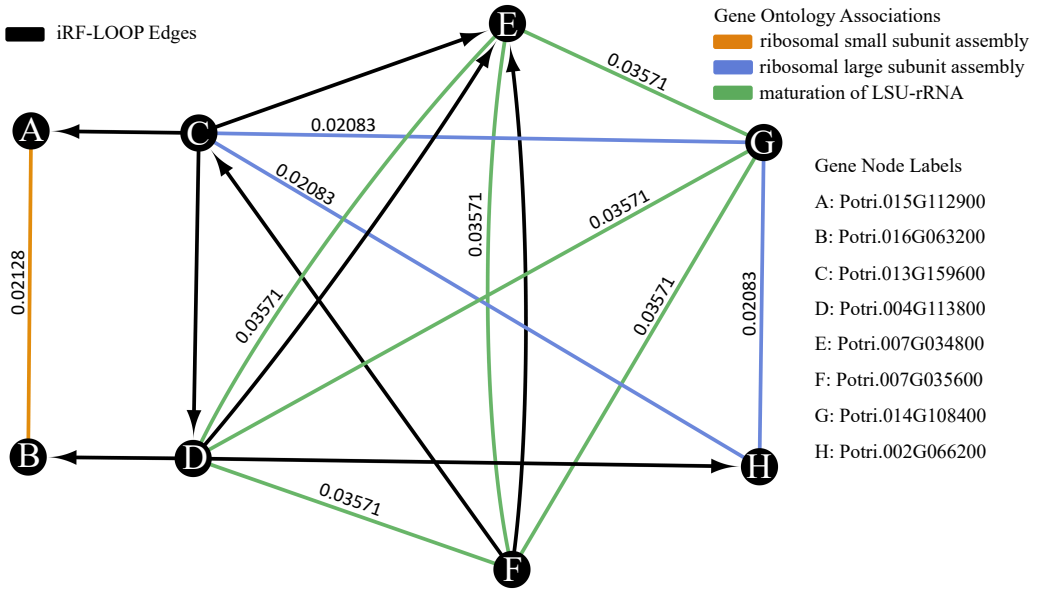



**Figure 2.7:** The network shown is a small example from the iRF prediction expression network overlaid with the GO process network. The nodes represent the genes. The black edges represent the iRF edges, which are directed *from* the feature *to* the predicted target. The colored edges represent different GO associations between genes, meaning that they share a GO term. Using the provided GO edge weights, this network has an intersect score of 0.0714, from connections DE and FE with both iRF edges and GO edge.

in blue, with the null distribution and intersect score for the top 0.1% co-expression network is shown in orange. The other three PEN and co-expression comparison networks are available in Appendix figure 2 through 4. The $t$-statistic for each network was calculated from these values, giving the values shown. All PEN $t$-statistic values had a $p$-value of effectively zero, as the iRF intersect scores were all significantly larger than the null distributions. The $t$-statistic for the co-expression comparison networks also had small $p$-values, but it is worth noting that even with more edges than the comparable PEN network, they generally had a lower GO score. This result confirms that the PENs created using iRF-LOOP are biologically significant, with respect to known GO annotations.

## 2.5 Discussion

We have presented a high-performance computing-capable implementation of Iterative Random Forest. This implementation uses Ranger, C++, and MPI to use the resources available on multi-node computation resources. We have shown that our implementation can perform X-AI-based eQTL-type analyses with millions of SNPs and have shown its ability to scale with multiple parameters. Using iRF to complete a whole analysis of the 24,175 gene expression profiles for *Arabidopsis thaliana*, assuming each run took approximately the same time as the shown above, would take approximately 705 days using 5 nodes, or 84,680 compute node hours, or 18 h using 4,600 Summit nodes. To complete the analysis for all 24,175 gene expression predictions on a laptop using the R code would take approximately 2,191 years. While this comparison seems impressive, it should also be noted that for a larger feature set and larger number of trees the large resources will be even more appropriate as the scaling factor will be even less effected by overhead. For cases where the number of SNPs is 10 million and higher, the code should be even more efficient on HPC systems. However, as not everyone has access to the fastest computers in the world this code could still run efficiently on a smaller system using a smaller set of SNPs.

Using this new implementation, we developed iRF-LOOP and used it to produce Predictive Expression Networks which were shown to have biologically relevant information. The process of iRF-LOOP has the potential to be used for a wide variety of data analysis

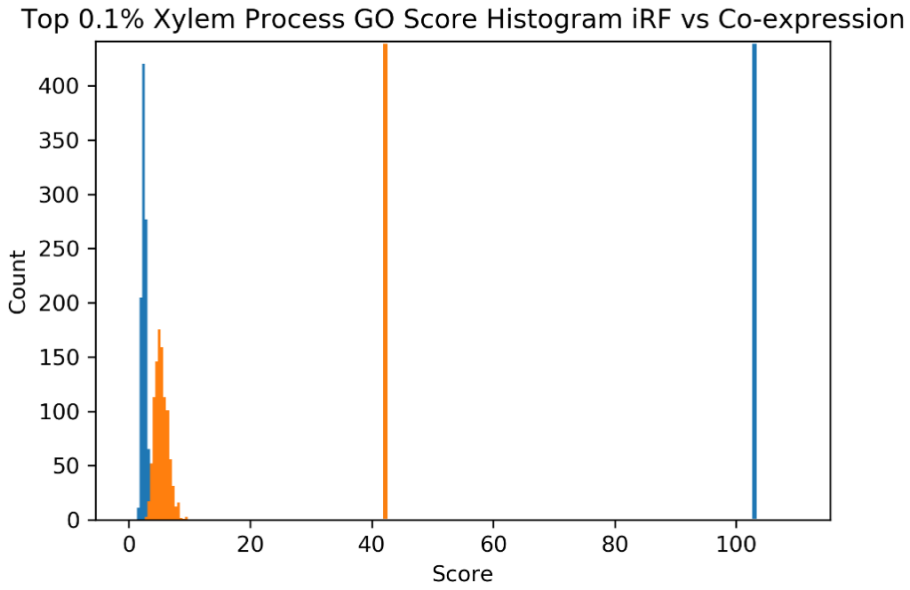



**Figure 2.8:** The null distribution histogram of the iRF network is shown in blue, with the network score in red. The co-expression null distribution is shown in orange, with the corresponding network score also in orange.

problems. With an appropriate amount of compute resources, it would be possible to build connected networks for each level of 'omics data available for a given species. The same concept could also be used to connect 'omics layers to each other as shown by using the SNPs in the X-AI-based eQTL analysis. This machine learning method is not limited to genetics or biology and has uses in other fields where systems can be represented as a matrix.

Downstream of any iRF analysis, there is the possibility of finding epistatic interactions among features from the resulting forests, using Random Intersection Trees (RIT) [150]. This method works regardless of the data type for the features, where it can find sets of SNPs that influence gene expression or sets of genes that influence other gene's expression, adding another set of nodes and groups to a Predictive Expression Network.

# Chapter 3

# Efficient Association Analysis in *Populus trichocarpa* using X-AI Heterogeneous Networks

## 3.1 Abstract

By considering biological systems as a whole, rather than a discrete set of parts, we can better determine functionality. Heterogeneous networks, which provide associations within and across multiple omics layers, allow for streamlined, robust hypothesis generation. Here, we introduce iterative Random Forest-based heterogeneous networks that provide an explainable-AI relationship between gene expression and metabolomics that considers the background information provided by the entire omics layer for each connection. We show that iRF-based heterogeneous networks are better than iRF-LOOP alone and Pearson's Correlation-based heterogeneous networks. This new method will enable confident and streamlined hypothesis generation for biological systems analyses.

## 3.2 Introduction

One of the many uses for computational biology analysis is the process of hypothesis generation. By taking existing data and known interactions, the goal of a computational method or model is to provide meaningful results worth further research. The confidence of the results directly impacts the downstream validation processes, with higher confidence results being more worthy of the effort, time, and money required by experimentation. Additionally, due to the complicated process, time, and money needed for experimental validation, the fewer results (with very high confidence), the easier it is to determine which experiments are worth pursuing.

For systems biology, the focus is not on any one data type, function, or outcome, but on how the many disparate parts of an organism interact, both within itself and with its surroundings. The goal for computational systems biology is then to appropriately capture all aspects of an organism, which requires gathering and analysis of tremendous amounts of biological data, as well as data on how those biological entities effect each other. One of the many ways to portray this information is with biological networks, where the nodes of these networks may show genes or other biological objects and the edges are some relationship among them.

One such network type is the heterogeneous network, which is a network created from relationships within and across multiple layers of data. Biological relationships and associations across omics layers, i.e. genes to metabolites, genes to proteins, or metabolites to proteins, are an intrinsic aspect of understanding biological systems. Someone using this network as a resource would be able to follow edges within one omic layer, into another, and potentially back along a different path. These new avenues provide additional lines of evidence to support various relationships, with stronger lines leading to lower false positives and fewer unnecessary experiments.

Many different methods could be used to build these networks. Any analysis process that can generate associations within a data layer could provide the within-layer associations, while any analysis process that can generate associations across data layers could provide the cross-layer associations. Pearson's Correlation is often used for associations both within a single omics layer, such as correlation of gene expression profiles, as well as across multiple omics layers, as done with gene expression to phenotype correlations. Weighted Gene Correlation Network Analysis (WGCNA)[194], a well-cited network analysis process in biology, often employs Pearson's Correlation as the method used to create the networks.

As shown in Cliff, et al. (2019)[32], iRF-LOOP is able to provide network associations for within omics layer analyses better than comparable Pearson's Correlation analyses. iRF-LOOP utilizes the X-AI method iterative Random Forest (iRF) to build predictive associations between features within a layer of data, such as the Predictive Expression Networks (PENs) described in Cliff, et al. These associations are better able to account for the background, i.e. the other features, when making associations and can provide more accurate feature importances via the iterative weighting of Random Forests[13]. This process also circumvents the combinatorics inherent in brute-force methods when trying to complete a higher order (i.e. n-way) 'all-to-all' process, as a model is needed once for each feature rather than once for each feature pair or set.

To use iRF as the basis for heterogeneous network creation requires an additional model for cross-layer analyses. Shown here is a new data-layer-agnostic cross-layer association method, iRF Cross-Layer. Together, these two methods provide a fully X-AI-based method suite for the creation of heterogeneous networks. Also provided is a comparison of iRF-LOOP

*Populus trichicarpa* gene expression networks to iRF heterogeneous networks using *Populus trichicarpa* gene expression and metabolite data, to determine whether or not the additional associations provided by the metabolites are able to provide any additional strength to the gene to gene relationships. A comparison of iRF heterogeneous networks to Pearson's Correlation heterogeneous networks, using both the *Populus trichicarpa* gene expression and metabolome data is also included.

## 3.3 Materials and Methods

*Gene expression data processing done by Joao Gabriel Felipe Machado Gazolla, Manesh Shah, Doug Hyatt, David Kainer, and Piet Jones. Metabolomics data generated and provided by Tim Tschaplinski.*

### 3.3.1 Gene Expression Data

The transcriptomics data is a matrix of gene expression data measured in 41,335 genes of leaf tissue, across 470 genotypes of *Populus trichocarpa* (Black cottonwood). The RNAseq data[198] was obtained from the NCBI SRA database (SRA numbers: SRP097016–SRP097036; (www.ncbi.nlm.nih.gov/sra). Reads were aligned to the *Populus trichocarpa* v.3.0 reference[173], generating a count of the number of reads per gene. Genes with less than 50 reads for at least 10% of the samples were removed. Gene length corrected Trimmed Mean of M-value[152, 143] (geTMM) counts were then obtained for each genotype, resulting in a genotype–transcript matrix of size 15,204 genes by 470 samples.

### 3.3.2 Metabolomics Data

As described in *Pleiotropic and Epistatic Network-Based Discovery: Integrated Networks for Target Gene Discovery* by Weighill, et al[184], 850 unique *Populus trichocarpa* leaf samples were collected over three consecutive sunny days in July 2012. Leaves (leaf plastocron index 9 plus or minus 1) were collect on a south facing branch from the upper canopy of each tree. The leaves were then wiped clean and fast frozen with dry ice. Leaves were stored at

−80 until processed for analyses. "Metabolites from leaf samples were lyophilized and then ground in a micro-Wiley mill (1 mm mesh size). Approximately 25 mg of each sample was twice extracted in 2.5 mL 80% ethanol (aqueous) for 24 h with the extracts combined, and 0.5 mL dried in a helium stream. Sorbitol [(75 $\mu$l of a 1 mg/mL aqueous solution)] was added, before extraction as an internal standard to correct for differences in extraction efficiency, subsequent differences in derivatization efficiency and changes in sample volume during heating[200]. Metabolites in the dried sample extracts were converted to their trimethylsilyl (TMS) derivatives, and analyzed by gas chromatography-mass spectrometry, as described in[105, 165, 160]. "[The peak areas were] normalized to the quantity of the internal standard (sorbitol) [injected, and the] amount of sample extracted... A large user-created database (>2,400 spectra) of mass spectral electron impact ionization (EI) fragmentation patterns of TMS-derivatized metabolites, as well as the Wiley Registry [10th] Edition combined with NIST [2014] mass spectral database, were used to identify the metabolites of interest to be quantified", Zhao et al[200]. Data creation was completed and provided by Tim Tschaplinski. The final processing resulted in 429 retention time and mass-to-charge ratio pairs across 739 *Populus trichocarpa* genotypes.

### 3.3.3 Iterative Random Forest

Iterative Random Forest (iRF)[13] is a an X-AI algorithm based on the Random Forest[19] method introduced by Leo Breiman in the early 2000s. The constructed model is an assemblage of unique decision trees, where each decision tree is designed to subdivide a set of samples into subgroups based on the similarity of the dependent variable. The tree makes decisions based on feature values and splits the samples into two children nodes. At each split, the best feature is selected based on the best feature and value that best splits the samples into like groups. This process is repeated recursively until appropriate stopping parameters are reached, which could be tree depth, number of samples in a terminal node, or 'purity' of the dependent variable for the samples in that terminal node. A feature importance score is calculated for each feature based on how relevant they were for the decisions made in the splits.

A Random Forest expands on this process by creating many decision trees, with each tree using a random subset of samples and a random subset of the features at each decision. The importance scores are averaged across the trees to produce scores for each feature from the entire forest. This process of randomly looking at different feature subsets across many splits and many trees allows for detection of highly interacting features without the brute force analysis of all possible combinations of features. Starting from the basis of Random Forest, iRF uses an iterative pattern of forests where a weighting scheme, calculated based on the prior forest's importance scores, replace the flat random sampling to determine the feature sampling at each split. This weighted sampling provides a weighting process which allows the forest to consolidate importance from features that likely contain the redundant information, and it allows for a more ordered path through the trees which provides human-interpretable conditional relationships between features.

### 3.3.4 iRF-LOOP

iRF-Leave One Out Prediction[32] (iRF-LOOP) uses iRF as the base association method to create networks. For a given feature set, a feature is used as the dependent variable for an iRF model and importance scores for all other features are calculated. This is repeated in a leave-one-out fashion for each feature in the set, providing associations from each feature to every other feature using the importance scores. These predictive associations can then be compared to correlation-type networks, such as Pearson's Correlation. In Cliff, et al.(2019) we were able to show that these new iRF-LOOP networks were able to capture more of a gold standard network than comparably generated Pearson's Correlation networks, even when the Pearson networks contained significantly more edges. This showed that iRF's ability to distill the features after multiple iterations was able to provide fewer and higher fidelity associations.

### 3.3.5 iRF Cross-Layer

iRF Cross-Layer uses a similar conceptual process to iRF-LOOP, however instead of providing associations between features within a single set, iRF Cross-Layer provides

associations between two sets of features, such as two distinct omics layers. Given a data set $G$ of $n$ features and $m$ samples and a data set $P$ of $s$ features and $m$ samples, iRF Cross-Layer starts by treating one feature from data set $P$ as the dependent variable (Y) and the $n$ features of data set $G$ as a predictor matrix (X) of size $m$ x $n$. Using an iRF model, the importance of each feature in X, for predicting Y, is calculated. The result is a vector, of size $n$, of importance scores. The scores are then normalized by the sum of the vector, producing importance score percentages for each feature. This process is repeated for each of the $s$ features in data set $P$, requiring $s$ iRF runs. The $s$ vectors of importance score percentages are concatenated into an $n$ x $s$ importance matrix. A diagram of the process can be seen in figure 3.1.

The resulting importance scores can then be used as an adjacency matrix, which is used to describe a network. The features from the first layer are represented along one axis and the features from the second layer are along the other, and together represent the total set of nodes in the network. The cells in the matrix correspond to the edge weights and are the different importance scores between each pair of features from layer one and layer two. Because edges exist solely *between* the two layers, the resulting network would look as it does in far right of figure 3.1, which is known as a bipartite network.

Because the process is agnostic to the types of data used for the layers, any data provided with the same set of samples in both is viable. This also means that one iRF Cross-Layer network could be built with layer X as the features and layer Y as the dependent variables and a separate network could be built with layer Y as the features and layer X as the dependent variables. Due to iRF's ability to take in background information when making the decision tree splits and eventually calculating the importance score, the edge weight from feature A in layer X to feature B in layer Y may be quite different from the reverse. This relationship means that the direction of the relationship is significant to interpreting the results, and that the edge in a network should have a corresponding direction - pointing *from* a given feature in the adjacency matrix *to* the given dependent variable.

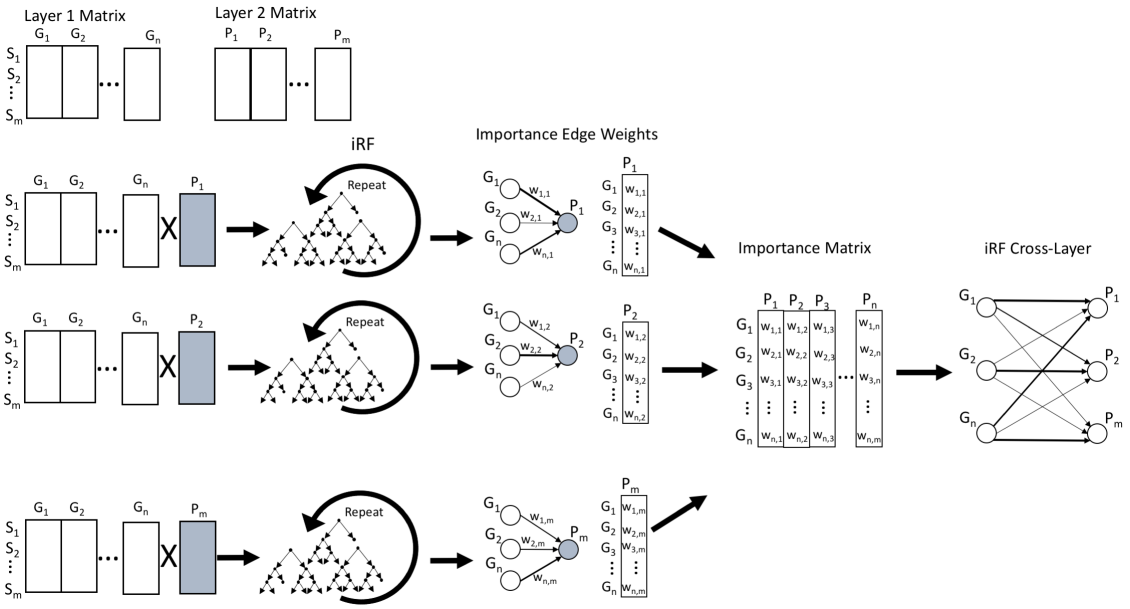



**Figure 3.1:** The process of iRF Cross-layer. Starting with two data layer matrices with feature columns, complete an iRF model for each feature in the 'layer 2' matrix. Together, the results of these models can be viewed as an adjacency matrix and bipartite graph, where all edges are directed and point *from* 'layer 1' *to* 'layer 2'.

### 3.3.6 iRF-based Heterogeneous Network Creation

Heterogeneous networks provide a consistent process for integrating multiple types of nodes and edges into a cohesive resulting network. For a two-layer network, a heterogeneous network requires four discrete adjacency matrices - a within-layer association matrix for each layer, and a cross-layer association matrix from layer one to layer two and another from layer two to layer one. These four pieces are then merged into one larger adjacency matrix that provides the relationship between all types of nodes in the final network. For larger sets of layers, the corresponding within-layer matrix and cross-layer matrices for every pair of layers would need to be created.

To create a 2-layer heterogeneous network using iRF-LOOP and iRF Cross-Layer requires four separate models - iRF-LOOP with layer 1, iRF-LOOP with layer 2, iRF Cross-Layer from layer 1 to layer 2, and iRF Cross-Layer from layer 2 to layer 1. All importance score vectors are normalized after each model is built. To create the overall network, the normalized edge files can be concatenated or the weight matrices for each can be combined as shown in Figure 3.2. Here, this process is completed with layer 1 as the *Populus trichocarpa* expression of genes and layer 2 as *Populus trichocarpa* metabolites.

### 3.3.7 Comparison to iRF-LOOP

To evaluate the usefulness of the multi-layer associations provided by iRF heterogeneous networks, a comparable iRF-LOOP network can be created using a single layer from the heterogeneous layers. For this analysis, an iRF-LOOP network containing solely the gene expression layer was created. Using the pathways and scoring described in sections 3.3.9 through 3.3.11 below, the ability of the two methods - iRF heterogeneous versus iRF-LOOP - to recover relevant genes for pathways can be compared. Effectively, this analysis shows whether or not metabolite associations (omics layer 2) are able to provide novel information about gene relationships not provided in the gene-gene associations (omics layer 1) alone.

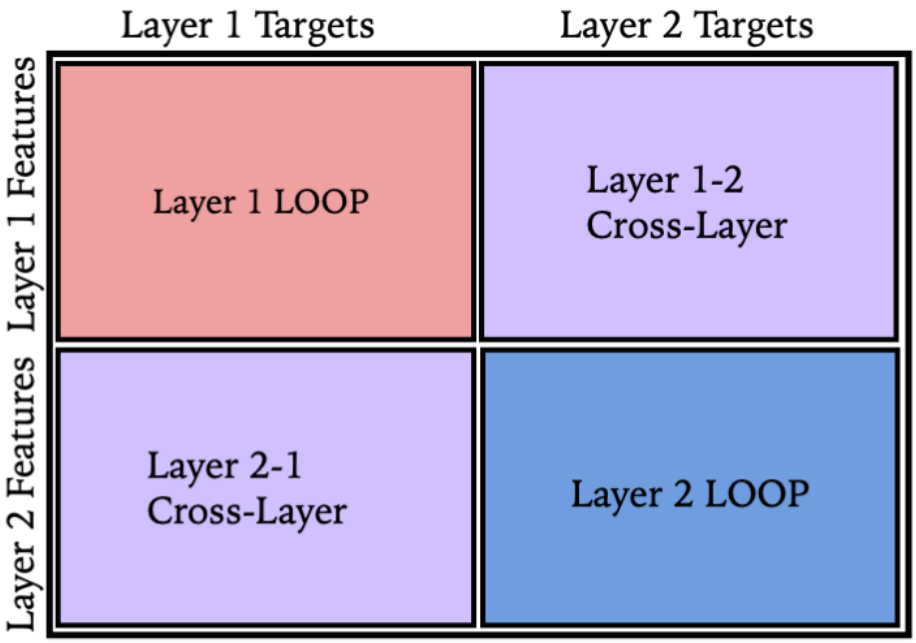



**Figure 3.2:** For each iRF-based model used for a 2-layer heterogeneous network, a independent adjacency matrix is created, using the normalized importance scores. The layer 1 and layer 2 iRF-LOOP matrices will have completely separate sets of features and targets. Each Cross-Layer will contain the features of one layer and the targets (dependent variables, y-vectors, etc.) of the other layer.

### 3.3.8 Comparison to Pearson's Correlation

To appropriately consider the usefulness of iRF-derived associations for heterogeneous networks, it is compared to a comparably generated heterogeneous network using Pearson's Correlation - a commonly used variation of Weighted Gene Correlation Network Association (WGCNA). To keep the analysis as similar as possible, the same subsets of data used to create the iRF-based networks are used for the Pearson's Correlation versions. Pearson's Correlation is completed between each pair of gene expression profiles to correspond to the iRF-LOOP expression model, between metabolite pairs to correspond to the iRF-LOOP metabolite model, and from each gene expression profile to each metabolite to correspond to the iRF Cross-Layer model. As Pearson's Correlation is bi-directional, due to its solely one-to-one nature, a separate analysis from metabolite to gene is not necessary, as the results would match the first cross-layer Pearson analysis.

As with the iRF-LOOP comparison analysis, the two model types are analyzed using the methods described in sections 3.3.9 through 3.3.11 below. The analyses will show whether iRF-derived or Pearson Correlation-derived associations are better able to capture gene relationships using gene and metabolite heterogeneous networks.

### 3.3.9 Thresholds

To appropriately compare the various networks requires a comparison of how well the networks perform at different thresholding levels. The ability to provide strong, useful associations with fewer edges would be very advantageous for hypothesis generation and the downstream experiments. All percentage comparisons are done against the original data and the total possible edges from those data, guaranteeing that the various thresholds are consistent.

To determine a threshold, the edges for each sub-network, i.e. iRF-LOOP on gene expression or Pearson's Correlation from expression and metabolite, are sorted with the largest values at the top. Edges are then kept, moving down the list of edges, until the number of predetermined edges is retained. Three raw count cut-offs were chosen, 5,000, 10,000, and 25,000 edges *per sub-network*, as well as two percentage counts, 0.1% and 1%

of the total possible edges for the respective sub-networks. These edge-based thresholds provide the ability to compare how strong the associations are for the different heterogeneous networks at different scales.

### 3.3.10 Pathway Selection

To determine which biological data to to use for the testing, all *Populus trichocarpa* pathways were downloaded from plantCyc.org (https://pmn.plantcyc.org/group?id=:ALL-PATHWAYS&org-id=POPLAR). These pathways provide gene sets associated with a given metabolite pathway, providing ideal genes for gene and metabolite associations. Pathways were kept if they contained at least five overlapping genes with our original expression data, prior to any thresholding. A minimum of five genes allows for five sets of at least four genes in the training set and one gene in the test set for the kfold process. This resulted in 253 pathways to use for further analysis of the various networks. Seed and test gene sets were then created using an 80/20 kfold process, producing 5 sets of seeds and corresponding test genes for each pathway.

### 3.3.11 Random Walk with Restart (RWR)

To compare the different models, the Random Walk with Restart[114] (RWR) method was implemented. A network 'walk' represents steps from node to node along the edges. Often, the edge weights are converted to probabilities of walking along that edge. This process facilitates finding associations and relationships in the network that may not be a direct connection, but could be strong indirect associations. For hypothesis generation using biological networks, a network walk may show strong backbones for a wider set of branching associations, as well as indirect biological relationships.

#### Network Walking

A basic walk builds probabilities of moving from one location (node) to another. If there are multiple locations to move to, each has their own probability. Given an example walker at

location $A$ and two other locations $B$ and $C$, the walker could could have a 10% chance of staying at $A$, a 30% chance of moving to $B$, and 60% chance of moving to $C$, written as:

$$\begin{array}{c} \\ A \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 0.1 & 0.3 & 0.6 \end{pmatrix} \end{array}$$

Where the row indicates the starting location and the column indicates the location being transitioned to. The probabilities of a row always sum to one and represent the total possible options for the walker. To expand to the probabilities of moving from any of the three locations to any other would require a matrix, such as:

$$T = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.2 & 0 & 0.8 \\ 0.4 & 0.5 & 0.1 \end{pmatrix} \end{array}$$

This matrix is referred to as transition matrix. For networks, where adjacency matrices are used to define the inherent relationships between the nodes, conversions can be used to create transition matrices for corresponding adjacency matrices. This conversion generally involves dividing each row value by the sum of the row, effectively using the weights as probability strengths.

### Random Walks

A random walker can use these created transition matrices to visit nodes in a network. The walker has a weighted random chance of moving from any location to another, based on the transition matrix's probabilities. To determine the probability of visiting a given node, from a certain starting location, the walker would need to make infinite walks. Instead, a Random Walk uses a probability state to simulate an infinite number of walkers, with the probabilities indicating the chance of the walkers being in any one state, i.e. at a given location. This probability state starts with a set of 'seeds' or starting locations, such as:

$$S = \begin{matrix} A & B & C \end{matrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

indicating that the starting location is solely location $A$, or:

$$S = \begin{matrix} A & B & C \end{matrix} \begin{pmatrix} 0.5 & 0.5 & 0 \end{pmatrix}$$

indicating that the walkers being at location $A$ or $B$. The vector should always sum to one, which allows the probability state vector to always sum to one and represent probabilities after the next set of calculations. A new probability state, representing the many walkers taking a single step, is calculated by multiplying the current probability state by the transition matrix:

$$\begin{matrix} A & B & C \end{matrix} \begin{pmatrix} 0.5 & 0.5 & 0 \end{pmatrix} * \begin{matrix} A \\ B \\ C \end{matrix} \begin{matrix} A & B & C \\ \end{matrix} \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.2 & 0 & 0.8 \\ 0.4 & 0.5 & 0.1 \end{pmatrix} = \begin{matrix} A & B & C \end{matrix} \begin{pmatrix} 0.15 & 0.15 & 0.7 \end{pmatrix}$$

As seen here, the probabilities of a walker residing in any of the three states after taking a step are heavily in favor of being in location $C$. This process can be repeated any number of times, indicating more steps taken, and is often used for network diffusion, as the steps flow out from the seeds and diffuse into the network. The probability state may eventually reach a point where another step does not change the states, indicating that a steady-state has been reached. This is often due to locations within the network that have high probabilities of being entered, but very low or no probability of leaving, indicating *sinks* in the network.

### Random Walk Restarting

For some types of analyses, finding the sinks of a network are less informative than a thorough understanding of the relationship between the nodes close to the starting seeds. For these types of walks, the transition probabilities adds an additional chance for the walker to return

to the original location and traverse outward once more, known as a Random Walk with Restart[114] (RWR). Given a restart probability, $\frac{\delta}{100}$, indicating that for every step taken, there's a $\delta\%$ chance the walkers return to their starting locations, the transition matrix is updated as follows:

$$\hat{T} = T + S * \frac{\delta}{1+\delta}$$

where $\hat{T}$ indicates the resulting transition matrix prior to re-normalization of the rows. After the rows are updated to sum to one, the new transition matrix can be used comparably to its prior use, it now simply has updated probabilities that would periodically send the walkers back to their starting states. Using the restart probability, the random walk process will converge to a steady state after a number of runs, providing a final probability state for a given set of seeds. For heterogeneous networks, RWR provides the ability to 'walk' across multiple layers of data using the provided associations. This can strengthen the hypothesis generation using these types of tools on these networks. The full transition matrix calculation for heterogeneous networks is described in [104].

### RWR Result Ranking

To accurately compare the models, as well as the different thresholds, requires a quantitative score for each. Because RWR provides resulting probabilities for every node in the network, ordering the gene ranks from largest to smallest provides a basic ranking of relevance for each gene. By using genes from the pathways selected, seeds from a set would be expected to provide high rankings for the test genes in the set. The higher these genes appear in the sorted probabilities, the higher their rank, and the more value given to that pathway's overall score. Scores across a variety of pathways could then be accumulated for a given network, providing a total score for how well a given network was able to provide associations, the larger the better. While the metabolites themselves do not appear in the ranking, they may influence the ranking of genes by providing better (or worse) avenues for the walker to traverse.

Discounted cumulative gain[49] (DCG) is a rank scoring process that provides large values for the top rank and diminishing values the farther a rank is from the top. Classically, a rank is simply the distance from the top of the ordering. For these analyses, we added one adjustment; the rank distance of each gene is defined as:

$$\text{rank distance} = \begin{cases} 1, & \text{if original rank} \leq X \\ \text{original rank} - (X-1), & \text{otherwise} \end{cases}$$

where the original rank is each gene's location in the sorted probabilities and $X$ is the total number of genes in the pathway. This converts the original rank into a distance measurement where a distance of 1 indicates the group of outputs that correctly align the expected set (i.e. genes in the pathway) and each value above 1 represents the 'distance' a gene is away from that expected set. This process is done because the genes in the pathways have no inherent rank amongst themselves. The score for a given pathway (for a given seed set, gene set, threshold, and model creation method) is then defined as:

$$\text{Score} = \sum_{i=1}^{n} \frac{1}{\log_2(\text{rank distance}_i + 1)}$$

where $i$ goes through the list of non-seed genes for a given fold. The higher the score for a given pathway and network, the better that network was able to capture that pathway's associations. Because of the rank distance adjustment, all pathway genes that rank in the expected set (within the top $X$) provide the same boost to the total score, as these genes are unordered in their pathways.

### 3.3.12 Computing Resources

The analyses described here used the supercomputers Summit and Andes, both of which are Oak Ridge Leadership Computing Facility (OLCF) supercomputers located at Oak Ridge National Laboratory (ORNL). Summit is an IBM system with approximately 4,600 nodes, each node with two IBM POWER9 processors and each processor with 22 cores (176 hardware threads) and three NVIDIA VOLTAV100 GPUs. Each node also contains 512 GB of DDR4 memory. Andes is a smaller cluster, with 704 nodes, each node with 2 AMD

16-core processors and 256GB of RAM, used for post-processing the larger runs completed on Summit, as well as smaller analytical processes such as the DCG calculations.

## 3.4 Results and Discussion

### 3.4.1 *Populus trichocarpa* iRF-based Heterogeneous Network

To create the data used for these networks, an overlap of genotypes between the expression and metabolite data, as described in sections 3.3.1 and 3.3.2, was produced, resulting in 385 genotypes. Each data layer was then trimmed to contain only this set of genotypes. The iRF-LOOP models for expression and metabolite, respectively, were generated using the previously published steps described in *A High-Performance Computing Implementation of Iterative Random Forest for the Creation of Predictive Expression Networks* by Cliff, et al[32]. The iRF Cross-Layer models were generated from expression to metabolite, and from metabolite to expression. The four sub-matrices, as shown in Figure 3.2, were created from the normalized importance scores of each model, and combined to generate the full iRF-based heterogeneous network.

### 3.4.2 Comparison to iRF-LOOP Networks

A comparison model using the same gene-gene iRF-LOOP model as used in section 3.4.1 was created. Using the thresholding method described in section 3.3.9, the full iRF-based heterogeneous network and the gene-gene iRF-LOOP network were thresholded at different edge cut-offs. Since these thresholds were calculated per-layer, the gene-gene layer for the iRF heterogeneous network and the full gene-gene iRF-LOOP model for each threshold level contained the same edges. The full iRF Heterogeneous network thresholds also contained the corresponding gene-metabolite, metabolite-metabolite, and metabolite-gene thresholded edges.

Transition matrices were calculated for each thresholded network and RWR was run for each seed set for each pathway. The corresponding test gene set was then ranked and scored using the DCG scoring process described previously. These gene scores were then

summed together for each pathway and this total was added to the cumulative score for the thresholded network.

Figure 3.3 provides a multi-step demonstration of how the RWR and DCG process would differ for the iRF-LOOP versus iRF heterogeneous networks. The gene-gene sub-network of the iRF heterogeneous network exactly matches the iRF-LOOP network, but also contains associations between genes and metabolites. Starting from node A, the seed for this RWR walker, the walker steps out into the network with probabilities based on the edge weights (shown as line thickness here). The higher the probability, the higher the chance a node is visited in a given step. At each iteration, the walker takes another probability-based step out from each location already visited. At the first step, for the iRF heterogeneous network, the X metabolite had the same chance of being visited as the B gene, whereas only the B gene was visited in the iRF-LOOP network. At the next iteration, the walker can now use the X metabolite and B gene to move further, while the iRF-LOOP uses just the B gene. For this RWR process, the test gene for the pathway was gene D. The iRF heterogeneous network, using the X metabolite, was able to reach this gene in three iterations of the walker, while the iRF-LOOP network took five iterations. The order at which the genes are visited provide simplified DCG ranks.

For other pathways, such as if Gene E were the test gene, the additional metabolite edges do not help the rankings. Notably, while the iRF heterogeneous network contains more edges overall than the comparable iRF-LOOP network, the only edges that differ between the two are metabolite-associate edges, which do not directly provide better scores for the DCG rankings. The metabolite associations can, however, provide better avenues for the walker than some direct gene to gene edges, if the metabolite-associations, and thus the heterogeneous network as whole, are able to capture cross-omic layer associations of relevance for gene-gene associations.

Table 3.1 provides the node and edge counts for each of the iRF-LOOP and iRF heterogeneous thresholded networks, as well as the cumulative DCG scores for each. Figure 3.4 shows the comparison of total DCG scores for each thresholding level for the two network types, as well as the total number of gene and metabolite edges provided in each. Of note, the

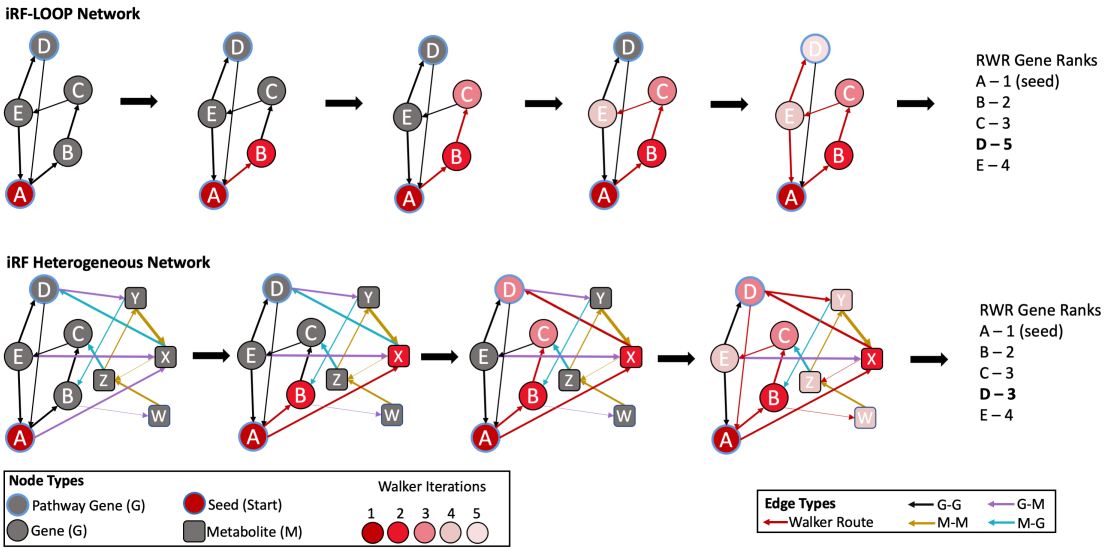



**Figure 3.3:** Shown here is a comparison of an iRF-LOOP network to an iRF Heterogeneous network. The iRF-LOOP network is a sub-network within the iRF Heterogeneous network. Starting from gene A, the seed gene, the walker makes steps out into the network in iterations, walking along edges based on probabilities of being visited. Here, the edge weights and the corresponding probabilities are shown with the edge thickness. The pathway test gene, D, is visited in three iterations of the walker in the iRF Heterogeneous network due to a high edge weight to metabolite X. Without this association, the iRF-LOOP network takes five iterations to reach gene D.

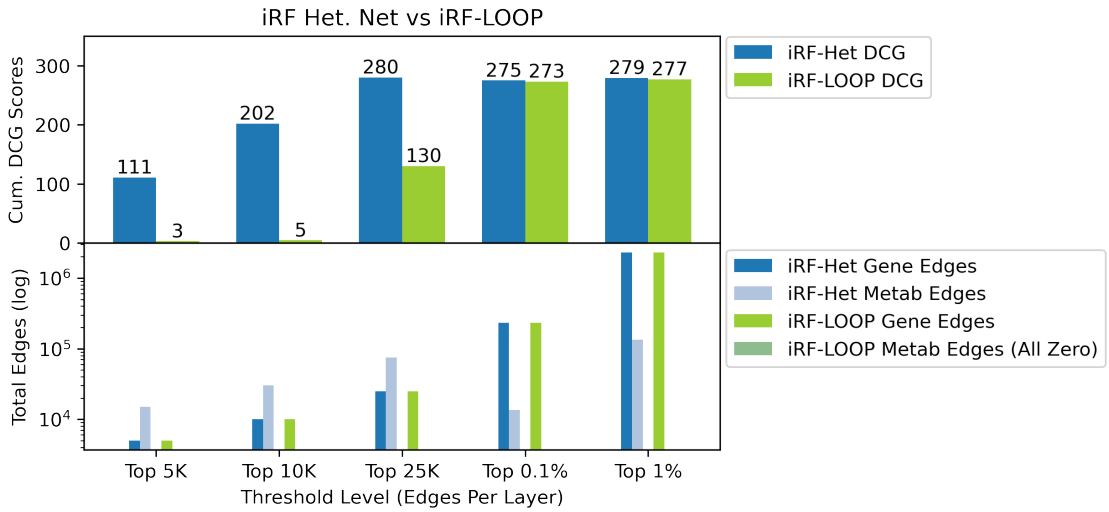



**Figure 3.4:** Comparison of iRF Heterogeneous network to iRF-LOOP network, for edge-based thresholding, using Random Walk with Restart and DCG. The y-axis for the top set of bars shows the cumulative scores across all 253 pathways and 5 seed sets for each thresholded network. The y-axis for the bottom set of bars shows the total number of gene-gene edges (Gene Edges) and gene-metabolite, metabolite-gene, and metabolite-metabolite (Metab Edges), in log-scale. Since the iRF-LOOP network contained no metabolite associations, the Metab Edges bar for those networks are zero. The cumulative DCG score the iRF-LOOP Top 5k is 3.01 and the iRF-LOOP top 10K DCG score is 5.04, both too low to be visible.

iRF heterogeneous networks were able to score much higher than the their corresponding iRF-LOOP networks for the 5K, 10K, and 25K thresholds, indicating that while the two network types had the exact same gene-gene edges, the addition of the metabolite associations was able to provide better avenues for the RWR walker and thus relevant relationships. For the Top 0.1% and Top 1% networks, the number of gene-gene edges seems to reach a level where the metabolite associations no longer provide any additional benefit and the two network types contain comparable information. Given the log-scale of the y-axis, the total number of edges for the 25K threshold is less than half of the total number of edges for both the iRF-LOOP and iRF heterogeneous Top 0.1% networks, but is able to provide a comparable cumulative DCG score. This appears to show that the metabolite associations are particularly useful when the total number of edges is limited to the very best edges, providing more information with fewer edges. This high signal to noise ratio is very useful for hypothesis generation and experimental design, where the very strongest associations are often the first to be analyzed.

### 3.4.3 Comparison to Pearson's Correlation

Heterogeneous models using Pearson's correlation were created. Comparable to the iRF-LOOP comparison analyses, networks were created for the various thresholds and from these transition matrices were calculated. Using the same seed and test gene sets, these models were also processed with RWR and the corresponding test genes were ranked and scored with the DCG calculations. The scores for each pathway were then accumulated to provide a cumulative score for each thresholded network. Table 3.2 provides the total node and edge counts for each model and each threshold level, as well as the cumulative DCG score for each.

Figure 3.5 shows the resulting cumulative DCG scores across all pathways for iRF and Pearson, for each threshold level. Across all thresholds, the iRF-based networks scored higher than the comparable Pearson-based networks. For the most stringent edge thresholds, indicating the very strongest edges in both networks, the iRF-based version were considerably better. As seen in the top panel of the graph, the iRF-based networks also had considerably more nodes than the corresponding Pearson-based networks. Nodes were retained and

**Table 3.1:** The table provides the node and edge counts, as well as cumulative DCG score for all pathways, for the different edge-based threshold levels of the iRF-LOOP and iRF Heterogeneous networks. Provided at the bottom of the table are the total possible node and edge counts for an all-to-all analysis of the entire data set, expression and metabolites.

| **Method** | **Threshold** | **Nodes** | **Edges** | **Total DCG Score** |
|---|---|---|---|---|
| Het. | top 5k Edges | 8,734 | 20,000 | 111.112 |
| LOOP | top 5k Edges | 8,298 | 5,000 | 3.010 |
| Het. | top 10k Edges | 13,597 | 40,000 | 201.529 |
| LOOP | top 10k Edges | 13,161 | 10,000 | 5.038 |
| Het. | top 25k Edges | 15,619 | 100,000 | 280.128 |
| LOOP | top 25k Edges | 15,183 | 25,000 | 130.412 |
| Het. | top 0.1% Edges | 15,641 | 244,644 | 275.530 |
| LOOP | top 0.1% Edges | 15,205 | 231,193 | 273.223 |
| Het. | top 1% Edges | 15,641 | 2,446,410 | 279.051 |
| LOOP | top 1% Edges | 15,205 | 2,311,921 | 277.768 |
| Total | — | 15,641 | 244,640,881 | — |

**Table 3.2:** The table provides the node and edge counts, as well as cumulative DCG score for all pathways, for the different edge-based threshold levels of the iRF-based and Pearson-based heterogeneous networks. Provided at the bottom of the table are the total possible node and edge counts for an all-to-all analysis of the entire data set, expression and metabolites.

| **Method** | **Threshold** | **Nodes** | **Edges** | **Total DCG Score** |
|---|---|---|---|---|
| iRF | top 5k Edges | 8,734 | 20,000 | 111.112 |
| Pearson | top 5k Edges | 1,447 | 20,000 | 27.146 |
| iRF | top 10k Edges | 13,597 | 40,000 | 201.529 |
| Pearson | top 10k Edges | 2,027 | 40,000 | 47.795 |
| iRF | top 25k Edges | 15,619 | 100,000 | 280.128 |
| Pearson | top 25k Edges | 3,764 | 100,000 | 81.473 |
| iRF | top 0.1% Edges | 15,641 | 244,644 | 275.530 |
| Pearson | top 0.1% Edges | 2,475 | 244,644 | 45.456 |
| iRF | top 1% Edges | 15,641 | 2,446,410 | 279.051 |
| Pearson | top 1% Edges | 14,088 | 2,446,410 | 259.460 |
| Total | — | 15,641 | 244,640,881 | — |

counted if they had at least one edge to another node in the network; all nodes with no edges at a given threshold were removed. Given that the total number of edges were consistent for each iRF-based and Pearson-based network comparison, this would indicate that the Pearson-based networks had more edges interconnecting a small subset of nodes and the iRF-based networks had more even distribution of edges across the full set of nodes. Figure 3.6 illustrates this difference, where networks a) and b) both contain eight nodes and the top ten edges. However, three nodes in network a) are connected to no other nodes due to the thresholding and would be removed from the 'node count', while network b) was able to retain associations for all nodes with the same thresholding, due to the differences in associations provided by the metrics used to derive them. Given the apparent relationship between number of nodes retained and cumulative DCG score, iRF's ability to provide strong associations across a diverse set of nodes is more beneficial for general analyses and potentially for hypothesis generation, whereas Pearson's Correlation can provide value for only a small cluster of nodes and their relationships to each other.

Figure 3.7 provides a diagram illustrating the individual DCG scores for genes from both the iRF and Pearson results, for the 25,000 Edge threshold networks, for the *Superpathway of geranylgeranyldiphosphate biosynthesis I (via Mevalonate)*. The subtext beneath each metabolite indicates the associated gene function, for which there were gene paralogs, denoted by the number in parenthesis. For each gene, if the gene was within the corresponding thresholded network, i.e. iRF-based top 25,000 Edges, a colored circle sized to the gene's resulting RWR DCG score is provided, blue for the iRF-based scores and orange for the Pearson-based. The cumulative DCG scores are indicated by the large circles in the bottom left. Notably, while some of the Pearson-based genes provided slightly larger individual DCG scores, the cumulative score for the Pearson-based network was approximately half that of the iRF-based network, due to the broad set of genes the iRF-based network was able to maintain.

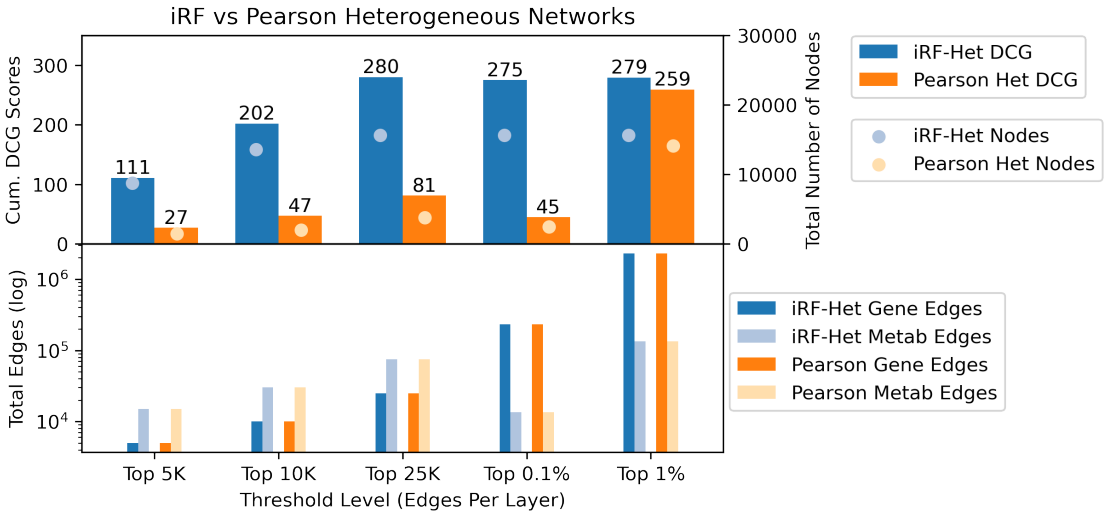



**Figure 3.5:** Comparison of iRF-based heterogeneous networks to Pearson-based heterogeneous networks, for edge-based thresholding, using Random Walk with Restart and DCG. The top y-axis shows the cumulative scores across all 253 pathways and 5 seed sets for each thresholded network. The y-axis for the bottom set of bars shows the total number of gene-gene edges (Gene Edges) and gene-metabolite, metabolite-gene, and metabolite-metabolite (Metab Edges), in log-scale.

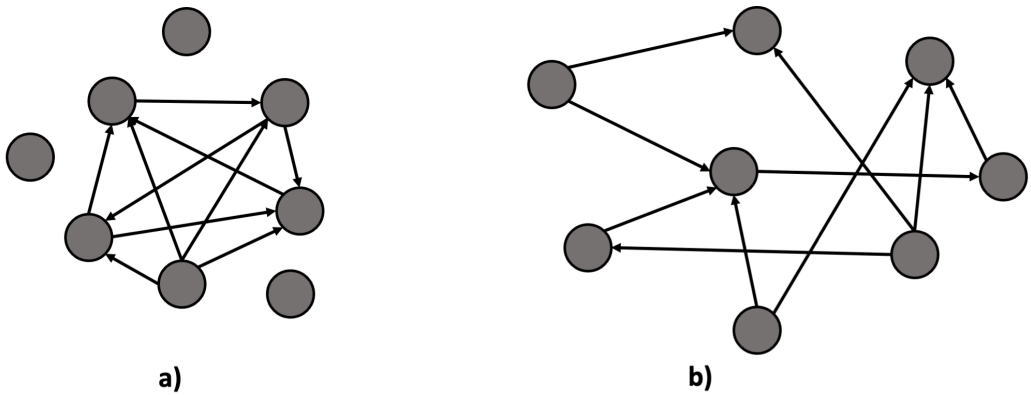



**Figure 3.6:** Two networks with eight nodes, each with the 'top' ten edges provided from an association metric. Network a) contains three nodes with no associations at this threshold while network b) contains fewer edges for each node but is able to retain relationships for all nodes in the network.

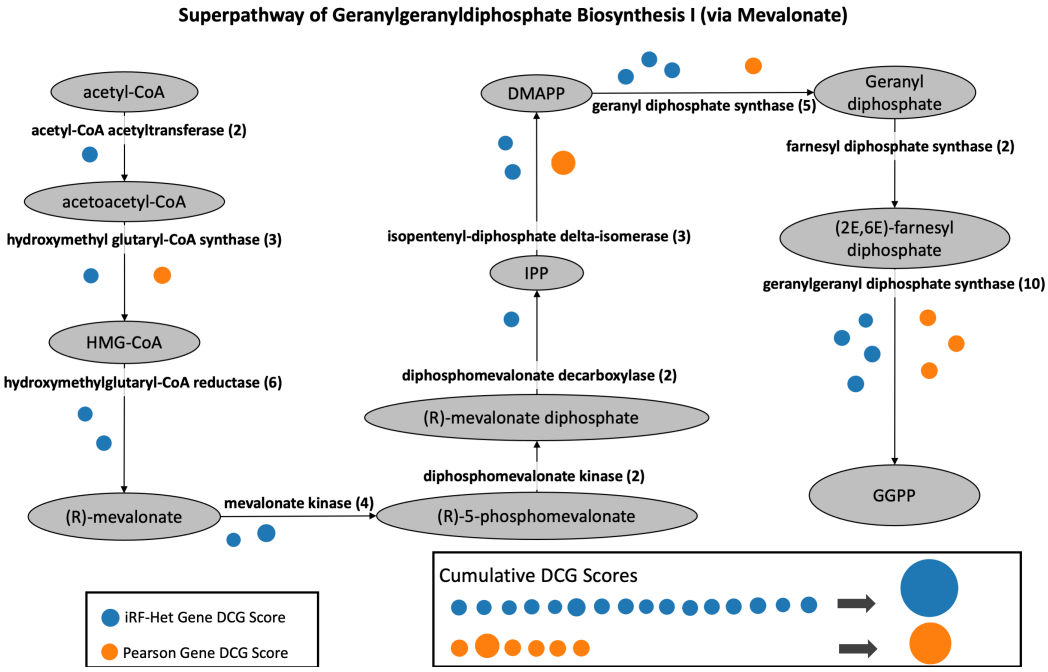



**Figure 3.7:** An illistration of the DCG scores for genes in the Geranylgernayldiphosphate Biosynthesis I Superpathway, with metabolites within the gray ovals and the associated gene function along the arrow. As provided by PlantCyc, multiple paralogs were listed for each gene function, indicated here by the numbers in parenthesis. DCG scores from both iRF and Pearson results are shown as colored circles sized to their respective scores, with the cumulative score for each in the bottom right corner. Results are for the 25,000 Edge-based threshold.

## 3.5 Conclusion

We have presented here iRF Cross-Layer, a new method for multi-omic associations using iterative Random Forest, an X-AI method. We have shown that heterogeneous networks produced from iRF Cross-Layer and iRF-LOOP are both better than iRF-LOOP alone and also better than Pearson-based heterogeneous networks. From this, we can conclude that the additional of additional layers of information and new associations from multiple omics layers can provide relevant results even when only one of the omics layers is of direct interest. We have also shown that iRF's ability to provide strong associations across the whole set of nodes at very stringent thresholds, rather than Pearson's tight set of associations for a small number of nodes, is more broadly useful for finding relevant associations with Random Walk with Restart. This ability should help provide a variety of high-value associations for hypothesis generation and downstream validation processes.

Of interest for future work would be the addition of the metabolites themselves in the DCG ranking. However, to provide these would require accurate lists of metabolites associated with the various pathways and a confident mapping of those to the metabolite profiles used in the analyses - which are often provided as mass/charge ratios and retention times rather than with names. Alternatively, these heterogeneous networks could help identify various metabolites based on the pathways within which they provide high rankings and strong associations.

Larger heterogeneous networks could also be produced with more omics layers. These would require iRF-based models for each layer pair, but would still be a significant reduction in calculations to brute-force all-to-all analyses. These larger heterogeneous networks could provide avenues of association through a variety of omics layers. Downstream of the creation of these iRF-based heterogeneous networks, the various associations provided by iRF could be used for the identification of potential epistatic interactions between genes, as well as higher order interactions between genes and metabolites, using Random Intersection Trees[151]. Random Intersection Trees are agnostic of the feature data and can provide interacting sets of features for each of the various sub-networks, which could then be included as set associations within the heterogeneous networks themselves.

# Chapter 4

# X-AI Network Resource with Multi-layer Interactions in *Arabidopsis thaliana*

## 4.1 Abstract

*Arabidopsis thaliana* is used as model organism for a variety of biological studies, resulting in a large amount of public data from a variety of omics layers and conditions. By constructing biological networks that can relate these various omics layers to each other, it is possible to create a resource that enables a systems biology view of these data types. Here, we provide a network in *Arabidopsis thaliana* using iterative Random Forest, an explainable-AI association method, for the omics layers of gene methylation, gene expression, metabolomics, and microbial presence. This network can help with a variety of researcher needs by providing a resource for hypothesis generation, Lines of Evidence validation, and other research questions without requiring reanalysis of these layers and associations.

## 4.2 Introduction

Systems Biology approaches focus on broad-scope levels of analysis, where nothing lives in isolation but where everything is interconnected and interacting. Complex relationships within and between biological omics layers can become cumbersome to portray with simple graphs and data tables. For these types of analyses and results, biological networks are often a useful tool for providing diverse types of relationships and details in a cohesive manner.

Networks have been used often in biological studies from the basics of simple pathway overlays to the Weighted Gene Correlation Network Analysis (WGCNA)[194] process and Gene Regulatory Networks (GRNs). Concurrently, new types of association networks are also being developed and expanded, such as GENIE3[78], which provides a new GRN with machine learning-based feature associations and GNE, a deep learning method to learn topological and gene expression properties from gene interaction networks[91].

Iterative Random Forest[13] provides explainable-AI (X-AI) based associations between biological units of a single omics layer with iRF-Leave One Out Prediction (LOOP)[32], as well as across omics layers with iRF Cross-Layer. Together, these two methods are better able to capture useful biological associations with fewer network edges than Pearson's Correlation - a commonly used metric for biological associations[32]. By providing these

associations with fewer edges, the resulting networks are able to limit the potentially false positive associations, providing the researcher with a more streamlined experience and a more useful tool.

*Arabidopsis thaliana*, as a model organism, has a wide variety of useful omics data sets available to the public. However, these data sets are often dispersed in ways that make it hard for future researchers to combine the results for new analyses. AtMAD (*Arabidopsis thaliana* multi-omics association database)[98], released in late 2020, provides a set of omics associations for public use. However, the association metrics used for AtMAD, while useful, are obfuscated to simple p-values and FDR metrics, hiding the true associative strength of the metrics.

Here, we provide an iRF-based multi-omic association network in *Arabidopsis thaliana* as a resource to facilitate hypothesis generation and other systems biology analyses. Current data sources include the omics layers of gene methylation, gene expression, metabolome, and microbiome. The iRF-based associations provide a comparable importance score metric across the diverse omics relationships, providing a understandable and universal association strength and the method allows for smaller, more refined networks with comparable, or better, associations to traditional processes. We also provide some association type descriptions, GO enrichment analyses for the omics layers, and suggestions for potential uses.

## 4.3 Materials and methods

*Gene expression and microbiome data processing done by Joao Gabriel Felipe Machado Gazolla, Manesh Shah, Doug Hyatt, David Kainer, and Piet Jones. Methylation data guidance provided by Jaclyn Noshay.*

### 4.3.1 Gene Expression Data

The original *Arabidopsis thaliana* RNA-seq data was generated as per Kawakatsu T., et al., *Epigenomic Diversity in a Global Collection of Arabidopsis thaliana Accessions* (2016)[89] and was obtained from NCBI Accession PRJNA319904. The accession provided 6584 fastq

files across 728 SRAs. STAR[47] mapping was completed on the fastq files, with *Arabidopsis thaliana* reference genome and annotation TAIR10.1, which is based on the Col-0 genotype, resulting in a 38,186 gene matrix with raw expression counts for each of the fastq files. Due to missing GSM IDs, 15 SRAs and a corresponding 186 fastq files were removed. Counts were then summed by SRA. Genes with less than 50 reads for 10% or more of the SRAs were then removed, eliminating 18,199 sparse genes. Two more SRAs, GSM2135743 and GSM2136308, were removed due to low overall counts. The final raw count matrix of 711 SRAs and 19,987 gene expression profiles was then processed using the geTMM normalization steps as described in Marcel, S., et. al. *Gene length corrected trimmed mean of M-values (GeTMM) processing of RNA-seq data performs similarly in intersample analyses while improving intrasample comparisons*[152].

### 4.3.2 Gene Methylation Data

Methylation data was obtained from Kawakatsu, et al.[89], from the 1001 Epigenome Project. Gene-body methylation data for CG, CHG, and CHH methylation types were provided. When trimmed to just leaf tissue and ambient temperature (a.t.) condition, there remained 846 accessions, and 27,061 methylated genes, with a methylation vector for each type at each gene, for a total of 81,183 total methylation vectors. Removal of methylation vectors with missing data entries resulted in the removal of 41,232 methylation vectors. The final data set included 39,951 methylation vectors across the three types for, 846 accessions.

### 4.3.3 Metabolite Data

Metabolite data was obtained from Wu S., et al[189]. The data contains 94 primary metabolites measured in 314 unique accessions. As per Wu S., "Metabolite extraction and derivatization from *Arabidopsis thaliana* leaves using GC–MS were performed as described by Lisec et al[Lisec et al.]. The GC–MS data were obtained using an Agilent 7683 series auto-sample (Agilent Technologies, http://www.home.agilent.com), coupled to an Agilent 6890 gas-chromatograph–Leco Pegasus two time-of-flight mass spectrometer (Leco; http://www.leco.com/). Identical chromatogram acquisition parameters were applied to

those previously used [36]. Chromatograms were exported from LECO CHROMATOF software (version 3.34) to R software. Ion extraction, peak detection, retention time alignment and library searching were obtained using the TargetSearch package from Bioconductor[39]. Day-normalization and sample median-normalization were conducted; the resulting data matrix was used for further analysis." After analysis for variance and distribution of values, the provided metabolite data was determined to be used as-is from Wu S., et al, providing a total of 94 metabolite profiles for 314 accessions.

### 4.3.4 Microbiome Data

The microbial data was gathered from the same data as the gene expression data from section 4.3.1. Following the STAR alignment process, as described in section 4.3.1, unmapped reads for 711 SRAs were then fed through the Parakraken[62] pipeline, which consists of partitioning reads into sliding kmer windows, matching kmer profiles to a database, and resolving ambiguous alignment. Parakraken is a map-reduce version of kraken2[187], which uses the NCBI genome database and was update to NCBI's current database on April 30, 2021. This resulted in a matrix of organism read counts for each *Arabidopsis thaliana* genotype.

All resulting read counts for taxonomy levels with depth no deeper than Family were removed. All *Veridplantae* Kingdom vectors were also removed, as the intent here is to identify microbes rather than other plants. Read counts were then summed at the Genus level. For each Genus, the average genome length was calculated by summing the associated species genome lengths and providing a weighted average by count of each species. Genome lengths and counts provided by the NCBI database on September 23, 2021. Each genus-level vector of read counts was then divided by the log of its corresponding average genome length, providing read count per total gene count and allowing microbes with vastly different genome sizes to be appropriately comparable. The final matrix provided 453 unique Genus-level microbe counts across 711 *Arabidopsis thaliana* accessions.

### 4.3.5 Iterative Random Forest

Iterative Random Forest (iRF)[13] is a machine learning method based off of Leo Breiman's Random Forest (RF)[19]. For both iRF and RF, the base learner is the decision tree. When provided with a set of features and a target for a set of samples, a decision tree finds the best break point within the features that best divide the samples into groups with similar target values. This process is repeated until the a predetermined stopping parameter is met, generally either the number of samples in each group or the total number of decisions made. A Random Forest uses a group of decision trees together to create a forest of learners. However, for these trees, rather than checking every feature at each split, a random subset of features are evaluated. This process allows for the analysis of highly interacting feature sets without requiring brute force analysis of each possible interaction. Once all of the trees are made, an importance score is calculated for each feature, based on how useful each was in the splitting process.

Iterative Random Forest expands on Random Forest by creating iterative forests, where the weighting scheme used for the feature sampling is based on the feature importance scores from the prior forest. This allows the forest to slowly consolidate feature importance scores for features that may be providing very similar information and this process orders the path through the decision trees themselves, which shows conditionality between features. This type of model, by including all features at once, also provides a background context for each feature-target relationship.

### 4.3.6 iRF-LOOP

Iterative Random Forest Leave One Out Prediction[32] (iRF-LOOP), uses iRF as the base learner to make predictive association networks for a provided omics layer. Once for each feature within the set of features, an iRF model is created where that feature is the target and the rest are the features. Importance scores are calculated for each feature. After all models are created, the importance scores between all sets of features can be treated as edge weights for a network, and used to show directional relationships between all features in the omics layer. Compared to correlation-based association networks, such as Pearson's

Correlation, in Cliff, et al., we were able to show that the iRF-LOOP networks were better able to capture more of a gold standard network. This showed that iRF is able to provide high fidelity associations with fewer edges based on its feature distillation process.

### 4.3.7 iRF Cross-Layer

iRF Cross-Layer uses a similar process to iRF-LOOP, with an additional omics layer. For a given set of features in one omics layer, and a set of dependent variables in a second omics layer, iRF Cross-Layer creates an iRF model using all of the features for one of the dependent variables. By repeating this process for each dependent variable in the second layer, a complete set of associations between every feature in the first omics layer and every dependent variable in the second is create. Graphically, this relationship can be viewed as a bipartite network, with importance score edge weights crossing the two layers.

Together with iRF-LOOP, these two methods provide the ability to build predictive associations both within and across omics layers using a single algorithmic model.

### 4.3.8 Data Mapping

To retain the most possible information and provide more samples for any given model, each model's sample overlap was determined independently. For the iRF-LOOP models for the four omics layers, the full set of samples for each layer was kept. For each of the iRF Cross-Layer models analyzed here, the layers and sample overlap was determined separately. The full set of samples, features, and targets (dependent variables, y-vecs, etc.) are provided in table 4.1.

### 4.3.9 Network Construction

For each of the iRF-LOOP and iRF Cross-Layer models, a set of relationships from the features to targets are created. For the iRF-LOOP models, the features and targets are one and the same, providing one set of nodes (features/targets) with edges between them representing the corresponding feature importance scores. For each node, edges point *to* the target and *from* the feature. Figure 4.1a) shows an example of an iRF-LOOP network with

**Table 4.1:** The table provides the total number of samples, features, and targets (dependent variable, y-vec, etc.) for each of the iRF-LOOP models, providing also the total number of samples and features for each of the listed data sets. The following rows list the iRF Cross-Layer models for this analysis and the corresponding sample overlap, features, and targets used for each model.

| **Layer 1** | **Layer 2** | **Samples** | **Features** | **Targets** |
|---|---|---|---|---|
| Methylation | — | 846 | 39,951 | 39,951 |
| Expression | — | 711 | 19,987 | 19,987 |
| Metabolites | — | 314 | 94 | 94 |
| Microbes | — | 711 | 453 | 453 |
| Methylation | Expression | 684 | 39,951 | 19,987 |
| Methylation | Metabolites | 101 | 39,951 | 94 |
| Methylation | Microbiome | 684 | 39,951 | 453 |
| Expression | Metabolites | 57 | 19,987 | 94 |
| Expression | Microbiome | 711 | 19,987 | 453 |
| Metabolites | Microbiome | 57 | 94 | 453 |

features 1, 2, 3, and 4 - such as the genes in the gene expression layer. Due to iRF's iterative process of paring down the features, there may not exist edges between all pairs of nodes because those importances, and thus edge weights, have been set to zero.

Figure 4.1b) provides a visual example for the iRF Cross-Layer associations, where the grey nodes 1, 2, 3, and 4 were features and the oranges nodes W, X, Y, and Z were targets - such as the genes in the gene expression layer and the metabolites in the metabolite layer. Figure 4.1c) then shows the merged networks for those from parts a) and b) - such as the gene expression iRF-LOOP model merged with the gene expression to metabolite iRF Cross-Layer model, where the features from the gene expression appeared in both networks and are represented by a single node each in the merged network.

The overall *Arabidopsis thaliana* network provided here merges all ten different iRF-based models into a single, large network with one node each for the various features in each layer and corresponding edges between nodes across all ten models, each with an edge weight provided by the appropriate iRF normalized importance score.

### 4.3.10 Network Thresholding

To provide a useful network tool requires thresholding the models to retain a fraction of the total edges, leaving those with the strongest associations and importances. For each of the ten models, and within those, for each of the iRF runs performed for a given target, a vector of feature importance scores was generated, and normalized (see iRF-LOOP and iRF Cross-Layer methods for details). Each value in one of these normalized importance vectors can be viewed as percent of that target's *variance-explained*, with the entirety of the importance vector representing the whole quantity of the target's variance explained by the iRF model. This is valid due to the ranger-based iRF implementation[32] uses variance reduction in the decision making process, which can be attributed to the features used for those decisions. From each of these variance-explained vectors, the *majority* of the variance can be attributed to the top X features with the highest variance, cumulatively summed until 55% of the variance is accounted for. These top X features, and their corresponding normalized importance scores/variance-explained quantities are retained. By repeating this process for each of the iRF runs within each of the iRF-LOOP and iRF Cross-Layer models

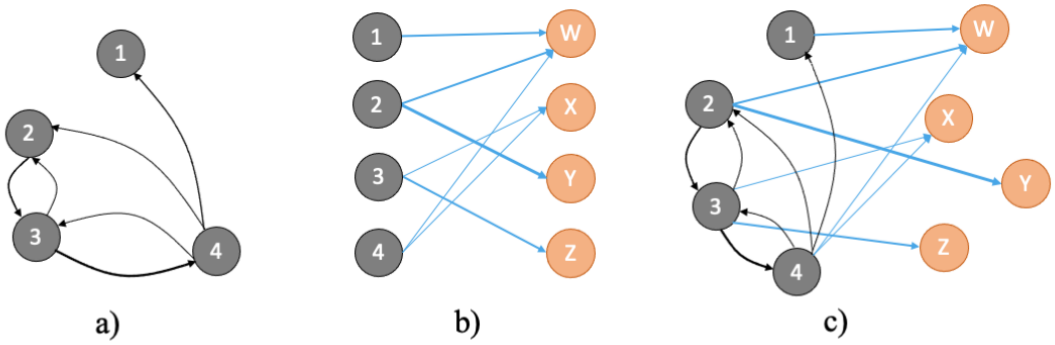



**Figure 4.1:** a) Depiction of an iRF-LOOP network, where the nodes are connected by edges point from them as a feature, and to them as a target. b) Depicts an iRF Cross-Layer network, where edges point from features in the first layer to targets in the second layer. c) Depiction of an iRF-LOOP network and iRF Cross-Layer network merged. The new network contains all edges from both prior networks, on overlayed nodes.

guarantees a majority of the total variance for every individual target is retained, but the many low value and low importance relationships are removed.

### 4.3.11 GO Enrichment

To provide a broad first look into the types of associations and groupings the large network may provide, a set of Gene Ontology enrichment analyses was completed. Gene Ontology (GO) enrichment refers to the process of statistically determining whether or not a subset of genes is 'enriched', or highly associated, with any specific GO term, such as a biological process, molecular function, or cellular component. The GOATOOLS Python package[94] provides a GO enrichment, using Fisher's Exact test, for sets of genes when provided with the gene subset, the total gene list for the population, the GO hierarchy[7, 34], and associated GO annotations for the appropriate species - provided for *Arabidopsis thaliana* by The Arabidopsis Information Resource (TAIR), https://www.arabidopsis.org/download/index-auto.jsp?dir=%2Fdownload_files%2FGO_and_PO_Annotations%2FGene_Ontology_Annotations, on www.arabidopsis.org.

Using the thresholded network, as described in section 4.3.10, a breadth-first search is completed for each unique node, referred to here as a '1-hop', or single step, network. For each 1-hop network, the gene expression nodes included can then be used as the input genes for a GO enrichment analysis. To constrain the analyses to those most likely to provide meaningful results, gene sets in the 50-200 count range were used for 1-hops originating from methylation or gene expression nodes, as these nodes often had larger quantities of edges overall. For the microbe originated 1-hops, gene sets of size larger than ten were included. Due to the small edge counts for the metabolite nodes and small number of nodes overall, all gene sets for the metabolite originated 1-hops were included.

The resulting GO enrichment lists, one per 1-hop gene set, provided a GO ID, count of relevant genes, and p-values for the FDR and other statistical methods for each GO enrichment hit. For each list, the enrichments were pruned to only those with an FDR p-value of 0.05 or less, which is standard practice for FDR p-value analyses. The resulting enrichments were then separated into the three GO term categories, biological process, molecular function, or cellular component, and the total count for each GO term was summed

across all the enrichment lists for each omic layer. This provides a cumulative view of which GO terms were highly enriched across the network, to varying degrees.

### 4.3.12 Association Types

While the specific biological functions tied to given network edges and associations will depend heavily on the exact genes, metabolites, and microbes within the interactions, the types of interactions across omics layers will follow certain patterns and may provide some guidance into the specifics of the associations. Given the variety of associations possible in the network, here we have outlined some anticipated types.

Figure 4.2 a-d) provides some examples of one-to-one associations for the various omics layers. Figure 4.2 a) indicates a methylation node (i.e. a methylated gene) interacting, or associating, with a gene's expression. The two genes could be the same or could be different, indicating different biological processes at play. Likely, the methylation regulates the expression of a given gene. Figure 4.2 b) shows a gene expression node interacting with another gene expression node. There are a variety of ways genes may influence each others interactions, such as transcription factor regulation or co-expression. Figure 4.2 c) indicates a gene's expression associating with the presence of a metabolite. This could be as straightforward as the gene being directly involved in the synthesis of the metabolite or as a broad as two entities in a much larger metabolic pathway. Figure 4.2 d) shows a metabolite feature associating with a microbe. This may be a relationship where the presence of a metabolite protects from a parasite or a more generic metabolic response to external stress.

Figure 4.2 e-g) provide a few examples of more complex associations. Figure 4.2 e) shows a methylated gene and a gene's expression both interacting with another gene's expression. This may be a combination of methylated gene regulation and transcription factor regulation, or may indicate a more complex relationship worth further analysis. Figure 4.2 f) indicates a methylated gene associating with a gene's expression, which is in turn associated with a metabolite. Together, these may indicate a methylation regulated metabolic process. Figure 4.2 g) shows two metabolites which have associations to and from each other, and an additional association to a microbe. The two-way association between metabolites may

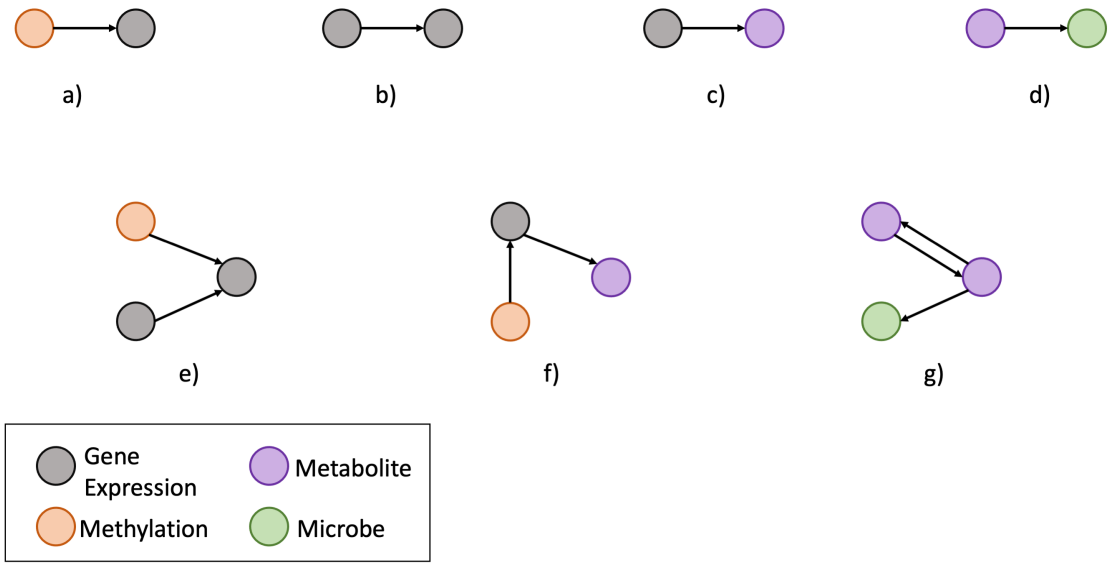



**Figure 4.2:** Shown here a variety of associations types that may appear in the multi-layer network. a) A methylated gene associates with a gene's expression. b) A gene's expression associates with another gene's expression. c) A gene's expression associates with a metabolite. d) A metabolite associates with a microbe. e) A methylated gene and a gene's expression associate with another gene's expression. f) A methylated gene associates with a gene's expression, which then associates with a metabolite. g) A two-association between two metabolites, and an association to a microbe.

indicate that the two share a common metabolite pathway, and that one is involved in microbial defense or some other microbial process.

While not exhaustive, these interaction types provide a first look into the various possible associations within the network, and provide some avenues of first thoughts when identifying association types.

### 4.3.13 Computing Resources

The analyses described here used the supercomputers Summit and Andes, both of which are Oak Ridge Leadership Computing Facility (OLCF) supercomputers located at Oak Ridge National Laboratory (ORNL). Summit is an IBM system with approximately 4,600 nodes, each node with two IBM POWER9 processors and each processor with 22 cores (176 hardware threads) and three NVIDIA VOLTAV100 GPUs. Each node also contains 512 GB of DDR4 memory. Andes is a smaller cluster, with 704 nodes, each node with 2 AMD 16-core processors and 256GB of RAM, used for post-processing the larger runs completed on Summit, as well as smaller analytical processes.

## 4.4 Results and Discussion

The full network was created for the four omics layers of methylation, gene expression, metabolites, and microbes, using iRF-LOOP for the within layer associations and iRF Cross-Layer for the cross-omic associations. Using the supercomputer Summit, a total of 82,019 individual iRF runs were completed, for each of the various targets across the iRF-based models. Together, these models and the final network account for 2.8 trillion possible associations, covering not just the direct relationship between a particular feature and target, but also the context and conditionality of the background provided by the rest of each feature set.

### 4.4.1 Network Thresholding

Using the process described in section 4.3.10, the resulting network was thresholded to a more manageable set of edges, while retaining all included nodes. Figure 4.3 shows the total number of nodes for each omics layer as a feature and as a target, and each cell in the matrix provides the total number of edges retained after thresholding and the percent of edges retained compared to the total all-to-all associations possible for each omics pair. While the percentages are quite small, these edges still retain a majority of the explanatory power provided by the network, allowing for a significantly more streamlined network and associations without significant loss in information.

### 4.4.2 GO Enrichment

Using the process described in 4.3.11, GO enrichment analyses were done for a selection of 1-hop sub-networks. Rates of enrichment were calculated as the total number of 1-hops of that omics layer with that specific GO enrichment divided by the total number of 1-hops from that omics layer with any GO enrichment, providing a look at how likely a given area of the network is to be enriched for a specific aspect. Figures 4.4 and 4.5 provide the top results for the gene methylation and gene expression Molecular Function enrichment rates. Of potential interest, for the gene methylation 1-hops that contained some level of enrichment for Molecular Function, almost a third were enriched for copper ion binding, with various enrichment rates for other metal ion binding also included in the top hits. The top Molecular Process hits for the gene expression 1-hops, however, seemed to focus more on methyltransferase activity, as well as DNA binding and activity. These indicate that there are definitive differences in the relationships within and across the omics layers, which provides a variety of useful relationships for various research aims and hypothesis analyses. Figures showing the top results for metabolites and microbes for Molecular Function, and for each of the four omics layers for Biological Process and Cellular Function are provided in Appendix figures 9 through 14

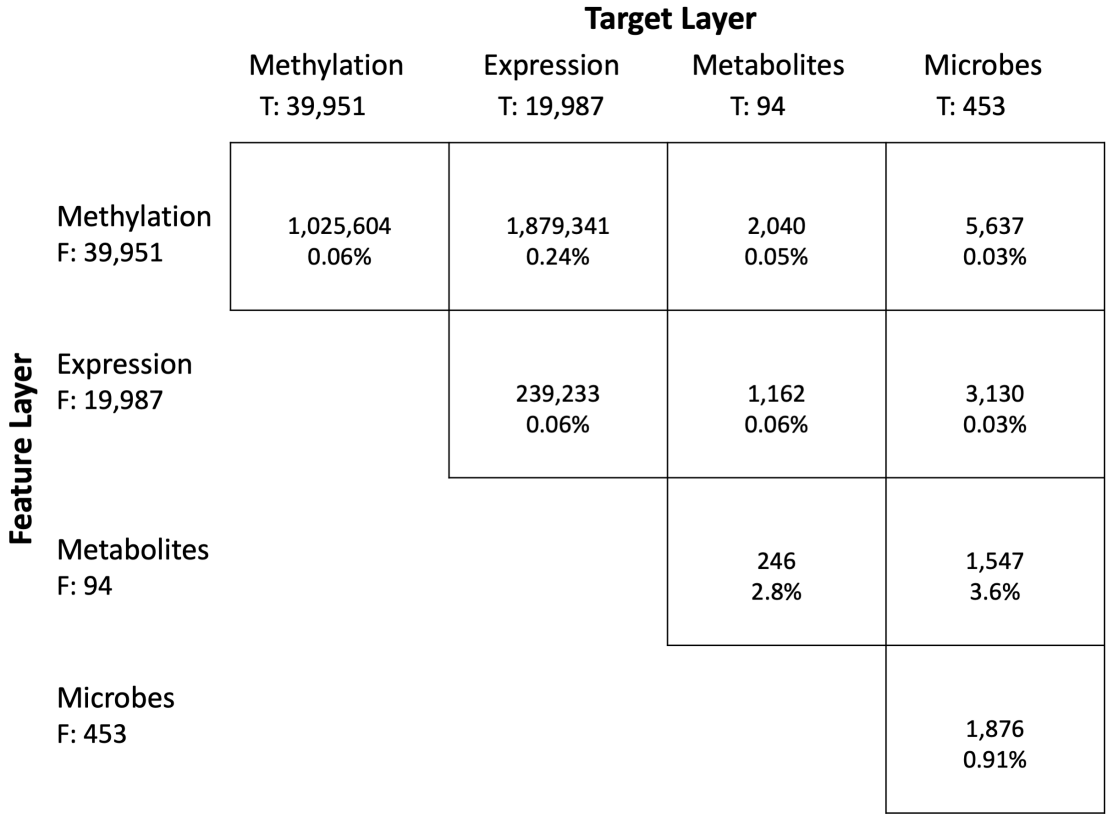



**Figure 4.3:** Each row corresponds to the data layer used for the iRF *features* (F) and each column corresponds to the data layer used for the *targets* (T), with total counts for both listed. The top value in each cell indicates the total number of associations for the corresponding feature and target data layers, after the top 55% variance thresholding. The value below is the percentage of edges retained given the total possible all-to-all associations for each set of features and targets. Note that this is still with retention of 55% or more of variance explained for each model, indicating that these small fractions of edges retain a majority of the explanatory power provided by the network.

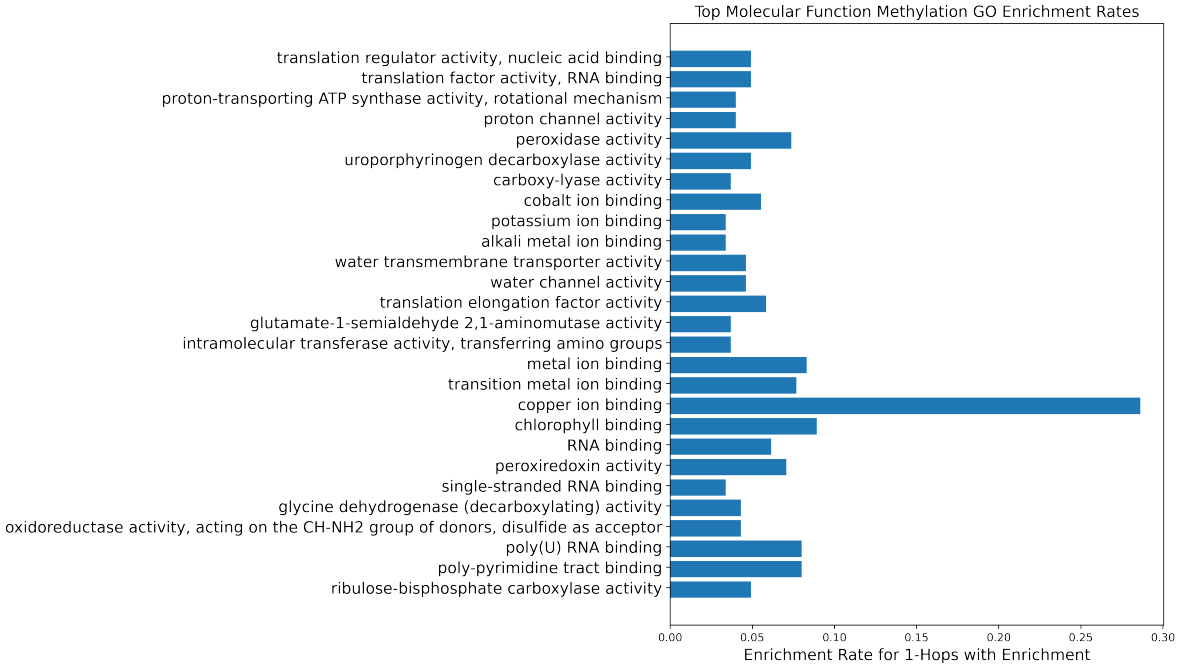



**Figure 4.4:** The rate of enrichment of the most highly enriched GO Molecular Functions terms for the genes from the 1-hop networks that originated from the gene methylation nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

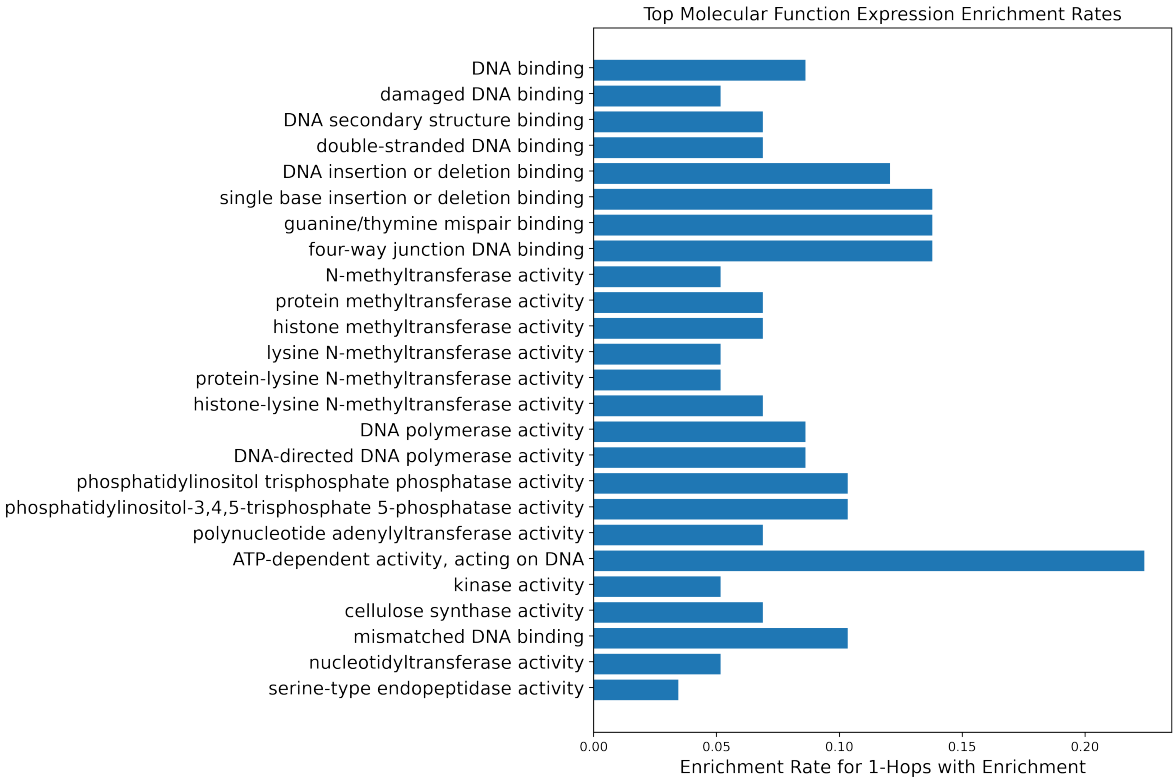



**Figure 4.5:** The rate of enrichment of the most highly enriched GO Molecular Functions terms for the genes from the 1-hop networks that originated from the gene expression nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

### 4.4.3 Potential Uses

Uses for this network resource are broad in scope, with different aspects of particular use given the researcher's aims. A specific aim could be furthering understanding of a metabolic pathway. Figure 4.6 provides a diagram indicating just one of the many ways these omics layers could interact biologically, namely methyl groups regulating gene expression, which in turn produces enzymes for metabolite catalysis resulting in antimicrobial metabolites. A central aim for many analyses is hypothesis generation, where a starting point, such as a specific gene or metabolite, may be known and its niche within the system may be of interest. The depth into the network, indicated by the number of steps taken away from the starting node or nodes of interest, depends on the type of relationships and patterns the researcher may be searching for.

Some analyses may need solely direct associations, where edges are directly connected to a specific node of interest, as was done with the 1-hop networks used for the GO enrichment process. These types of uses may focus on what metabolites are directly associated with a specific gene of interest, or which genes' expression are influenced by a specific gene's methylation. These types of direct connections can help elucidate gene functions and find pleiotropic gene activity. A gene whose expression (i.e. a gene expression node) has a high number of associations with the expression of other genes (i.e. other gene expression nodes), as seen in figure 4.2b), a candidate to be a transcription factor because the relationships of the edges indicates that the potential transcription factor gene is involved, or associated with, the expression of other genes. Pleiotropic evidence may come in the form of a gene's expression associating heavily with multiple metabolites or microbes, or both. For CRISPR[11] projects, which intend to change the expression of a certain gene or set of genes, these associations can be used to determine potential off-target effects identifiable via pleiotropy.

While direct associations focus on a single node and connected edges, indirect relationships may branch out farther into the network, finding multiple hops of associations and interactions, and may focus on a single path through the network rather than analyses of all edges of a single node. Because the edges in the network are weighted by importance, nodes that are multiple steps apart, but connected by high weight edges, are likely of importance

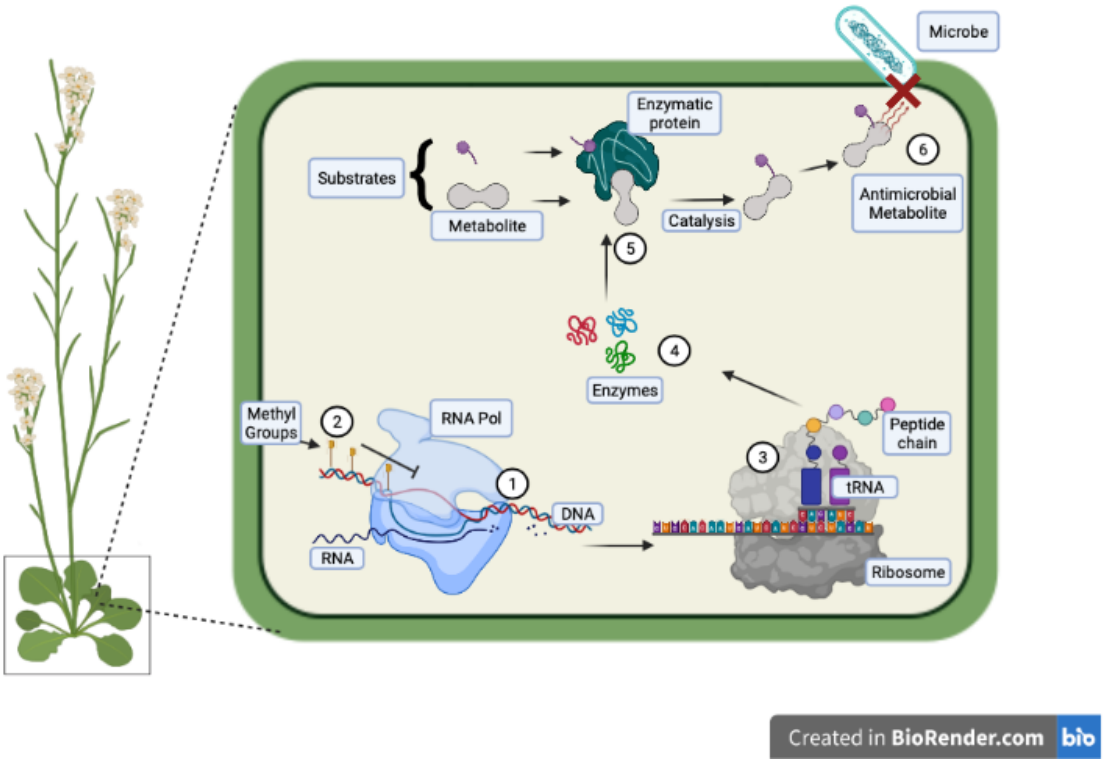



**Figure 4.6:** A diagram of potential biological interactions possible with the omics layers of methylation, gene expression, metabolites, and microbes included in the provided network. Step 1) DNA transcription in to RNA, where the quantity of RNA created is how heavily a gene is *expressed*. Step 2) Methyl groups prior to or within a gene may regulate the transcription process. Step 3) RNA strands are transcribed into tRNA and peptide chains within a ribosome. Step 4) The peptide chains fold into enzymatic proteins which may be involved in Step 5) metabolite catalysis. Step 6) Where the catalyzed metabolite may aid in antimicrobial behavior. This diagram indicates one of many possible ways for these omics layers to interact biologically.

to each other without a direct connection between them. A multi-step relationship between a methylation node, a gene expression node, and multiple metabolites may help elucidate alternate metabolic pathways dependent on methylation.

Some types of analyses may need both direct and indirect contributions, such as when using Lines of Evidence (LOE)[60] approaches where both the immediate relationships and surroundings are informative. One area of research that may benefit from both aspects are plant microbe interactions. By providing microbial interactions as an omics layer, we acknowledge that these relationships are directly influenced by and themselves influence internal plant systems. By focusing on the direct microbial interactions as well as the general communities and their surroundings, hypotheses may be furthered in regards to plant defense mechanisms, plant stress response, and potential symbiotic relationships. These may be through direct associations from gene to microbe or metabolite to microbe, or indirect relationships where microbes appear around certain metabolic pathways or regulatory processes.

## 4.5 Conclusion

Provided here is a new association network for *Arabidopsis thaliana* using X-AI associations, for use as a resource for hypothesis generation or other systems biology analyses. This network provides associations for gene expression, gene methylation, metabolites, and microbiome, a useful and unique set of omics layers. Together, the network provides the most informative and relevant associations from the 2.8 trillion total possible associations and was able to appropriately account for the background provided by the other features in each omics layer and the conditionality it may entail. With the thresholded network, a majority of the variance explained by these associations was retained, while limiting the large quantity of informative edges to those of the highest values. For a subset of 1-hop networks, GO enrichment analyses were done and results were provided for the various omics layers, showing likely functional interactions across the whole network. Also shown are some templates for potential association types and how to start interpreting them biologically, as well as some potential biological analyses that could take advantage of this resource.

Because of the modular nature of the network layers, future omics layers can be included without requiring complete reanalysis of the current results. Of particular interest are the inverse cross-layer associations to those included in this product, potentially providing further elucidation for associations that are heavily dependent on the directionality of the relationship, as well as the strength of the association in the context of the rest of the omics layer's features. Other types of associations, such as Pearson's Correlation, direct regulatory associations, or other biological associations could also be integrated into the network, provided a cohesive edge-type metric was maintained. This resource should maintain long-term usefulness because of its ability to integrate new information.

# Chapter 5

# Conclusion

Iterative Random Forest (iRF), as an expansion of Random Forest, is shown here as a useful method for feature selection and association, particularly in biological omics data. The method provides a consolidation of feature importance scores, providing better converging results for large feature sets, and uses the feature space as background conditionality for the associations. Together, these abilities allow iRF-based analyses to handle large, interacting feature sets and provide concise, meaningful results.

Computationally, iRF is limited by the system it is run on and the implementation of the algorithm itself. To handle truly large, multi-omic biological data sets, iRF needed an implementation that could scale to these data sets and complete large numbers of comparisons in a timely manner. By designing an implementation suited for High Performance Computing systems, such as Titan and Summit, using MPI the Ranger-based iRF implementation is capable of feature selection and association analyses for thousands of features in a fraction of the time possible on personal computers. This increase in efficient computation has opened the door to use iRF as the base model for processes such as iRF-LOOP and iRF Cross-Layer for large biological data sets, both of which require the completion of many iRF models.

iRF-LOOP and iRF Cross-Layer provide the ability to create X-AI association networks with biological omics data, such as gene expression, metabolomics, and methylation. iRF-LOOP provides associations within an omics layer while iRF Cross-Layer provides associations between two layers. Used together, the two methods can be used to create

heterogeneous networks, a full set of associations within and across multiple omics layers. These iRF-based heterogeneous networks are able to provide biologically relevant results better than iRF-LOOP alone or correlation methods like Pearson's Correlation.

With the methods themselves validated and shown to provide informative associations, they can be used to produce network resources for biological systems. *Arabidopsis thaliana* is an important model organism for plant science and has a wide variety of publicly available data sets. Here, a large-scale X-AI based association resource for *Arabidopsis thaliana* omics layers of gene methylation, gene expression, metabolomics, and microbiomics has been provided. This will support a wide variety of future research as a hypothesis generation and lines of evidence network resource.

Together, this dissertation provides a background on each of the main aspects of work, three new methods - iRF-LOOP, iRF Cross-Layer, and iRF-based heterogeneous networks, and a computational systems biology resource with multi-omics associations.

# Bibliography

[1] Abel, H. J. and Duncavage, E. J. (2013). Detection of structural DNA variation from next generation sequencing data: a review of informatic approaches. *Cancer Genetics*, 206(12):432–440. 17

[2] Adadi, A. and Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160. 2, 3

[3] Ahmed, H., Howton, T., Sun, Y., Weinberger, N., Belkhadir, Y., and Mukhtar, M. S. (2018). Network biology discovers pathogen contact points in host protein-protein interactomes. *Nature communications*, 9(1):1–13. 34

[4] Aittokallio, T. and Schwikowski, B. (2006). Graph-based methods for analysing networks in cell biology. *Briefings in Bioinformatics*, 7(3):243–255. 3

[5] Aldous, D. and Fill, J. A. (2002). Reversible markov chains and random walks on graphs. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/$\sim$aldous/RWG/book.html. 31

[6] Arcidiacono, G. (2020). Statistics calculator: Variance. 24

[7] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–29. 92

[8] Ayres, D. L., Cummings, M. P., Baele, G., Darling, A. E., Lewis, P. O., Swofford, D. L., Huelsenbeck, J. P., Lemey, P., Rambaut, A., and Suchard, M. A. (2019). BEAGLE 3: Improved Performance, Scaling, and Usability for a High-Performance Computing Library for Statistical Phylogenetics. *Systematic Biology*, 68(6):1052–1061. 2

[9] Ayres, D. L., Darling, A., Zwickl, D. J., Beerli, P., Holder, M. T., Lewis, P. O., Huelsenbeck, J. P., Ronquist, F., Swofford, D. L., Cummings, M. P., Rambaut, A., and Suchard, M. A. (2012). BEAGLE: An Application Programming Interface and

High-Performance Computing Library for Statistical Phylogenetics. *Systematic Biology*, 61(1):170–173. 14

[10] Backiyarani, S., Sasikala, R., Sharmiladevi, S., and Uma, S. (2021). Decoding the molecular mechanism of parthenocarpy in Musa spp. through protein–protein interaction network. *Scientific reports*, 11(1):1–14. 34

[11] Barrangou, R. and Doudna, J. A. (2016). Applications of CRISPR technologies in research and beyond. *Nature Biotechnology*, 34(9):933–941. 100

[12] Basu, S. and Kumbier, K. (2017). *iRF: iterative Random Forests*. R package version 2.0.0. 39

[13] Basu, S., Kumbier, K., Brown, J. B., and Yu, B. (2018). Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences*, 115(8):1943–1948. 5, 38, 58, 60, 83, 87

[14] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550. 9

[15] Biau, G. and Scornet, E. (2016). A random forest guided tour. *TEST*, 25(2):197–227. 4

[16] Borrill, P. (2020). Blurring the boundaries between cereal crops and model plants. *New Phytologist*, 228(6):1721–1727. 14

[17] Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *COLT '92*. 9

[18] Boulesteix, A., Janitza, S., Kruppa, J., and König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507. 3

[19] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. 3, 4, 38, 60, 87

[20] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. 3, 4, 39

[21] Brundrett, M. C. and Tedersoo, L. (2018). Evolutionary history of mycorrhizal symbioses and global host plant diversity. *New Phytologist*, 220(4):1108–1115. 20

[22] Caspi, R., Altman, T., Dale, J. M., Dreher, K., Fulcher, C. A., Gilham, F., Kaipa, P., Karthikeyan, A. S., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Paley, S., Popescu, L., Pujar, A., Shearer, A. G., Zhang, P., and Karp, P. D. (2010). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 38(suppl_1):D473–D479. 3

[23] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28. 8, 9, 38

[24] Chartrand, G. (1977). *Introductory graph theory*. Courier Corporation. 26

[25] Chen, S. and Senar, M. A. (2019). Exploring efficient data parallelism for genome read mapping on multicore and manycore architectures. *Parallel Computing*, 87:11–24. 2

[26] Chen, X. and Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, 99(6):323–329. 38

[27] Chen, X., Zhang, B., Wang, T., Bonni, A., and Zhao, G. (2020). Robust principal component analysis for accurate outlier sample detection in RNA-Seq data. *BMC Bioinformatics*, 21(1):269. 23

[28] Chhetri, H. B., Furches, A., Macaya-Sanz, D., Walker, A. R., Kainer, D., Jones, P., Harman-Ware, A. E., Tschaplinski, T. J., Jacobson, D., Tuskan, G. A., et al. (2020). Genome-wide association study of wood anatomical and morphological traits in populus trichocarpa. *Frontiers in plant science*, 11:1391. 22

[29] Chhetri, H. B., Macaya-Sanz, D., Kainer, D., Biswal, A. K., Evans, L. M., Chen, J.-G., Collins, C., Hunt, K., Mohanty, S. S., Rosenstiel, T., et al. (2019). Multitrait genome-wide association analysis of populus trichocarpa identifies key polymorphisms controlling morphological and physiological traits. *New Phytologist*, 223(1):293–309. 22

[30] Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, 10(1):35. 2, 3, 22, 26

[31] Clarke, L., Glendinning, I., and Hempel, R. (1994). The MPI Message Passing Interface Standard BT - Programming Environments for Massively Parallel Distributed Systems. pages 213–218, Basel. Birkhäuser Basel. 13

[32] Cliff, A., Romero, J., Kainer, D., Walker, A., Furches, A., and Jacobson, D. (2019). A high-performance computing implementation of iterative random forest for the creation of predictive expression networks. *Genes*, 10(12). 34, 58, 61, 72, 83, 87, 90

[33] Collin, F., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., Marin, J., and Estoup, A. (2021). Extending Approximate Bayesian Computation with Supervised Machine Learning to infer demographic history from genetic polymorphisms using DIYABC Random Forest. *Molecular Ecology Resources*. 3

[34] Consortium, T. G. O. (2021). The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Research*, 49(D1):D325–D334. 92

[35] Corporation, M. and Weston, S. (2018). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.14. 46

[36] Cregger, M. A., Carper, D. L., Christel, S., Doktycz, M. J., Labbe, J., Michener, J. K., Dove, N. C., Johnston, E. R., Moore, J. A., Velez, J. M., et al. (2021a). Plant–microbe interactions: From genes to ecosystems using populus as a model system. *Phytobiomes Journal*, 5(1):29–38. 20

[37] Cregger, M. A., Carper, D. L., Christel, S., Doktycz, M. J., Labbé, J., Michener, J. K., Dove, N. C., Johnston, E. R., Moore, J. A. M., Vélez, J. M., Morrell-Falvey, J., Muchero, W., Pelletier, D. A., Retterer, S., Tschaplinski, T. J., Tuskan, G. A., Weston, D. J., and Schadt, C. W. (2021b). Plant–Microbe Interactions: From Genes to Ecosystems Using Populus as a Model System. *Phytobiomes Journal*, 5(1):29–38. 16

[38] Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227(5258):561–563. 20

[39] Cuadros-Inostroza, Á., Caldana, C., Redestig, H., Kusano, M., Lisec, J., Peña-Cortés, H., Willmitzer, L., and Hannah, M. A. (2009). TargetSearch - a Bioconductor package for the efficient preprocessing of GC-MS metabolite profiling data. *BMC Bioinformatics*, 10(1):428. 86

[40] Darriba, D., Taboada, G. L., Doallo, R., and Posada, D. (2010). ProtTest-HPC: Fast selection of best-fit models of protein evolution. In *European Conference on Parallel Processing*, pages 177–184. Springer. 14

[41] Darst, B. F., Malecki, K. C., and Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genetics*, 19(1):65. 25

[42] Davidson, E. and Levin, M. (2005). Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(14):4935 LP – 4935. 34

[43] Davidson, E. H. (2010). Emerging properties of animal gene regulatory networks. *Nature*, 468(7326):911–920. 34

[44] Dessi, N., Pascariello, E., and Pes, B. (2013). A Comparative Analysis of Biomarker Selection Techniques. *BioMed research international*, 2013:387673. 7

[45] Desvergne, B., Michalik, L., and Wahli, W. (2006). Transcriptional Regulation of Metabolism. *Physiological Reviews*, 86(2):465–514. 22

[46] Dettmer, K., Aronov, P. A., and Hammock, B. D. (2007). Mass spectrometry-based metabolomics. *Mass Spectrometry Reviews*, 26(1):51–78. 19

[47] Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics (Oxford, England)*, 29(1):15–21. 85

[48] Douglas, C. J. (2017). Populus as a model tree. In *Comparative and Evolutionary Genomics of Angiosperm Trees*, pages 61–84. Springer. 15

[49] Dupret, G. (2011). Discounted Cumulative Gain and User Decision Models BT - String Processing and Information Retrieval. pages 2–13, Berlin, Heidelberg. Springer Berlin Heidelberg. 71

[50] Ellis, B., Jansson, S., Strauss, S. H., and Tuskan, G. A. (2010). Why and how populus became a "model tree". In *Genetics and genomics of Populus*, pages 3–14. Springer. 15

[51] Enright, A. J., Van Dongen, S., and Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584. 30

[52] Erwin, D. H. and Davidson, E. H. (2009). The evolution of hierarchical gene regulatory networks. *Nature Reviews Genetics*, 10(2):141–148. 34

[53] Evans, L. M., Slavov, G. T., Rodgers-Melnick, E., Martin, J., Ranjan, P., Muchero, W., Brunner, A. M., Schackwitz, W., Gunter, L., Chen, J.-G., et al. (2014). Population genomics of populus trichocarpa identifies signatures of selection and adaptive trait associations. *Nature genetics*, 46(10):1089–1096. 16

[54] Fabris, F., Doherty, A., Palmer, D., de Magalhães, J. P., and Freitas, A. A. (2018). A new approach for interpreting Random Forest models and its application to the biology of ageing. *Bioinformatics*, 34(14):2449–2456. 11

[55] Fields, S. and Johnston, M. (2005). Whither model organism research? *Science*, 307(5717):1885–1886. 14

[56] Finnegan, E. J., Peacock, W. J., and Dennis, E. S. (1996). Reduced DNA methylation in Arabidopsis thaliana results in abnormal plant development. *Proceedings of the National Academy of Sciences*, 93(16):8449 LP – 8454. 18

[57] Fleischer, T., Tekpli, X., Mathelier, A., Wang, S., Nebdal, D., Dhakal, H. P., Sahlberg, K. K., Schlichting, E., Børresen-Dale, A.-L., and Borgen, E. (2017). DNA methylation at enhancers identifies distinct breast cancer lineages. *Nature communications*, 8(1):1–14. 21

[58] Flint-Garcia, S. A., Thornsberry, J. M., and Buckler, E. S. (2003). Structure of Linkage Disequilibrium in Plants. *Annual Review of Plant Biology*, 54(1):357–374. 17

[59] Fraimout, A., Debat, V., Fellous, S., Hufbauer, R. A., Foucaud, J., Pudlo, P., Marin, J.-M., Price, D. K., Cattel, J., and Chen, X. (2017). Deciphering the routes of invasion of Drosophila suzukii by means of ABC random forest. *Molecular biology and evolution*, 34(4):980–996. 3

[60] Furches, A., Kainer, D., Weighill, D., Large, A., Jones, P., Walker, A. M., Romero, J., Gazolla, J. G. F. M., Joubert, W., Shah, M., Streich, J., Ranjan, P., Schmutz, J., Sreedasyam, A., Macaya-Sanz, D., Zhao, N., Martin, M. Z., Rao, X., Dixon, R. A., DiFazio, S., Tschaplinski, T. J., Chen, J.-G., Tuskan, G. A., and Jacobson, D. (2019). Finding New Cell Wall Regulatory Genes in Populus trichocarpa Using Multiple Lines of Evidence. *Frontiers in Plant Science*, 10. 44, 102

[61] Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R. H., Daniel, D. J., Graham, R. L., and Woodall, T. S. (2004). Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary. 13, 41

[62] Garcia, B. J., Labbé, J. L., Jones, P., Abraham, P. E., Hodge, I., Climer, S., Jawdy, S., Gunter, L., Tuskan, G. A., Yang, X., Tschaplinski, T. J., and Jacobson, D. A. (2018). Phytobiome and Transcriptional Adaptation of Populus deltoides to Acute Progressive Drought and Cyclic Drought. *Phytobiomes Journal*, 2(4):249–260. 86

[63] Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2015). VSURF: An R Package for Variable Selection Using Random Forests. *The R Journal*, 7(2):19–33. 3, 11

[64] Goh, W. W. B., Wang, W., and Wong, L. (2017). Why Batch Effects Matter in Omics Data, and How to Avoid Them. *Trends in Biotechnology*, 35(6):498–507. 25

[65] Gradin, R., Lindstedt, M., and Johansson, H. (2019). Batch adjustment by reference alignment (BARA): Improved prediction performance in biological test sets with batch effects. *PLOS ONE*, 14(2):e0212669. 25

[66] Gregorutti, B., Michel, B., and Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678. 25

[67] Gropp, W., Gropp, W. D., Lusk, E., Skjellum, A., and Lusk, A. D. F. E. E. (1999). *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press. 13

[68] Gropp, W., Lusk, E., Doss, N., and Skjellum, A. (1996). A high-performance, portable implementation of the mpi message passing interface standard. *Parallel Computing*, 22(6):789 – 828. 13

[69] Guerra, F. P., Suren, H., Holliday, J., Richards, J. H., Fiehn, O., Famula, R., Stanton, B. J., Shuren, R., Sykes, R., and Davis, M. F. (2019). Exome resequencing and GWAS for growth, ecophysiology, and chemical and metabolomic composition of wood of Populus trichocarpa. *BMC genomics*, 20(1):1–14. 16

[70] Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16. 9

[71] Hacquard, S. and Schadt, C. W. (2015). Towards a holistic understanding of the beneficial interactions across the populus microbiome. *New Phytologist*, 205(4):1424–1430. 20

[72] Harfouche, A., Jacobson, D., Kainer, D., Romero, J., Harfouche, A. H., Scarascia Mugnozza, G., Moshelion, M., Tuskan, G., Keurentjes, J., and Altman, A. (2019). Accelerating climate resilient plant breeding by applying next-generation artificial intelligence. *Trends in Biotechnology*. Accepted and Selected for Cover. 38, 40

[73] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. 10

[74] Horvath, S. and Dong, J. (2008a). Geometric Interpretation of Gene Coexpression Network Analysis. *PLOS Computational Biology*, 4(8):e1000117. 3

[75] Horvath, S. and Dong, J. (2008b). Geometric interpretation of gene coexpression network analysis. *PLOS Computational Biology*, 4(8):1–27. 32

[76] Horvath, S. and Dong, J. (2008c). Geometric Interpretation of Gene Coexpression Network Analysis. *PLOS Computational Biology*, 4(8):e1000117. 34

[77] Huynh-Thu, V. A. and Geurts, P. (2018). dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data. *Scientific Reports*, 8(1):3384. 10

[78] Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLOS ONE*, 5(9):1–10. 3, 10, 34, 44, 83

[79] Iacobucci, C., Götze, M., and Sinz, A. (2020). Cross-linking/mass spectrometry to get a closer view on protein interaction networks. *Current opinion in biotechnology*, 63:48–53. 34

[80] Jansson, S. and Douglas, C. J. (2007). Populus: a model system for plant biology. *Annu. Rev. Plant Biol.*, 58:435–458. 15

[81] Jiang, H., Deng, Y., Chen, H. S., Tao, L., Sha, Q., Chen, J., Tsai, C. J., and Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5:1–12. 3

[82] Jiang, P., Wu, H., Wang, W., Ma, W., Sun, X., and Lu, Z. (2007). MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic acids research*, 35(suppl$_2$) : $W339 - -W344$.3

[83] Jin, J., Tian, F., Yang, D.-C., Meng, Y.-Q., Kong, L., Luo, J., and Gao, G. (2016). PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. *Nucleic Acids Research*, 45(D1):D1040–D1045. 45

[84] Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127. 25

[85] Jones, D. T. (2019). Setting the standards for machine learning in biology. *Nature Reviews Molecular Cell Biology*, 20(11):659–660. 25

[86] Jones, O. A. and Cheung, V. L. (2007). An introduction to metabolomics and its potential application in veterinary science. *Comparative Medicine*, 57(5):436–442. 19

[87] Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. 7

[88] Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780. 34

[89] Kawakatsu, T., Huang, S.-s. C., Jupe, F., Sasaki, E., Schmitz, R. J., Urich, M. A., Castanon, R., Nery, J. R., Barragan, C., He, Y., Chen, H., Dubin, M., Lee, C.-R., Wang, C., Bemm, F., Becker, C., O'Neil, R., O'Malley, R. C., Quarless, D. X., Alonso-Blanco, C., Andrade, J., Becker, C., Bemm, F., Bergelson, J., Borgwardt, K., Chae, E., Dezwaan, T., Ding, W., Ecker, J. R., Expósito-Alonso, M., Farlow, A., Fitz, J., Gan, X., Grimm, D. G., Hancock, A., Henz, S. R., Holm, S., Horton, M., Jarsulic, M., Kerstetter, R. A., Korte, A., Korte, P., Lanz, C., Lee, C.-R., Meng, D., Michael, T. P., Mott, R., Muliyati, N. W., Nägele, T., Nagler, M., Nizhynska, V., Nordborg, M., Novikova, P., Picó, F. X., Platzer, A., Rabanal, F. A., Rodriguez, A., Rowan, B. A., Salomé, P. A., Schmid, K., Schmitz, R. J., Seren, Ü., Sperone, F. G., Sudkamp, M., Svardal, H., Tanzer, M. M., Todd, D., Volchenboum, S. L., Wang, C., Wang, G., Wang, X., Weckwerth, W., Weigel, D., Zhou, X., Schork, N. J., Weigel, D., Nordborg, M., and Ecker, J. R. (2016b). Epigenomic Diversity in a Global Collection of Arabidopsis thaliana Accessions. *Cell*, 166(2):492–505. 84, 85

[90] Kawakatsu, T., Huang, S.-S. C., Jupe, F., Sasaki, E., Schmitz, R. J., Urich, M. A., Castanon, R., Nery, J. R., Barragan, C., He, Y., Chen, H., Dubin, M., Lee, C.-R., Wang, C., Bemm, F., Becker, C., O'Neil, R., O'Malley, R. C., Quarless, D. X., Consortium, . G., Schork, N. J., Weigel, D., Nordborg, M., and Ecker, J. R. (2016a). Epigenomic diversity in a global collection of arabidopsis thaliana accessions. *Cell*, 166(2):492–505. 42

[91] KC, K., Li, R., Cui, F., Yu, Q., and Haake, A. R. (2019). GNE: a deep learning framework for gene network inference by aggregating biological information. *BMC Systems Biology*, 13(2):38. 34, 83

[92] Kearsey, M. J. (1998). The principles of QTL analysis (a minimal mathematics approach). *Journal of Experimental Botany*, 49(327):1619–1623. 21

[93] Kearsey, M. J. and Farquhar, A. G. L. (1998). QTL analysis in plants; where are we now? *Heredity*, 80(2):137–142. 21

[94] Klopfenstein, D. V., Zhang, L., Pedersen, B. S., Ramírez, F., Warwick Vesztrocy, A., Naldi, A., Mungall, C. J., Yunes, J. M., Botvinnik, O., Weigel, M., Dampier, W., Dessimoz, C., Flick, P., and Tang, H. (2018). GOATOOLS: A Python library for Gene Ontology analyses. *Scientific Reports*, 8(1):10872. 92

[95] Laibach, F. (1943). Arabidopsis thaliana (L.) Heynh. als Objekt für genetische und entwicklungsphysiologische Untersuchungen. *Bot. Archiv*, 44:439–455. 15

[96] Laird, P. W. (2010). Principles and challenges of genome-wide dna methylation analysis. *Nature Reviews Genetics*, 11(3):191–203. 18

[97] Lan, P., Shi, Q., Zhang, P., Chen, Y., Yan, R., Hua, X., Jiang, Y., Zhou, J., and Yu, Y. (2020). Core genome allelic profiles of clinical Klebsiella pneumoniae strains using a random forest algorithm based on multilocus sequence typing scheme for hypervirulence analysis. *The Journal of infectious diseases*, 221(Supplement$_2$) : $S263 - -S271$. 3

[98] Lan, Y., Sun, R., Ouyang, J., Ding, W., Kim, M.-J., Wu, J., Li, Y., and Shi, T. (2021). Atmad: Arabidopsis thaliana multi-omics association database. *Nucleic Acids Research*, 49(D1):D1445–D1451. 15, 21, 22, 84

[99] Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., and Nowe, A. (2012). A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(4):1106–1119. 7

[100] Lee, H., Merzky, A., Tan, L., Titov, M., Turilli, M., Alfe, D., Bhati, A., Brace, A., Clyde, A., and Coveney, P. (2021). Scalable HPC & AI infrastructure for COVID-19 therapeutics. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–13. 14

[101] Leek, J. T. (2014). svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic acids research*, 42(21):e161–e161. 25

[102] Leys, C., Ley, C., Klein, O., Bernard, P., and Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766. 23

[103] Li, H. (2011). A statistical framework for snp calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993. 42

[104] Li, Y. and Patra, J. C. (2010). Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, 26(9):1219–1224. 70

[105] Li, Y., Tschaplinski, T. J., Engle, N. L., Hamilton, C. Y., Rodriguez, M., Liao, J. C., Schadt, C. W., Guss, A. M., Yang, Y., and Graham, D. E. (2012). Combined inactivation of the Clostridium cellulolyticum lactate and malate dehydrogenase genes substantially increases ethanol yield from cellulose and switchgrass fermentations. *Biotechnology for Biofuels*, 5(1):2. 60

[106] Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461. 28

[107] Lin, W.-Z., Fang, J.-A., Xiao, X., and Chou, K.-C. (2011). iDNA-Prot: identification of DNA binding proteins using random forest with grey model. *PloS one*, 6(9):e24756. 3

[108] Lis, J. T. (2019). A 50 year history of technologies that drove discovery in eukaryotic transcription regulation. *Nature Structural & Molecular Biology*, 26(9):777–782. 18

[Lisec et al.] Lisec, J., Schauer, N., Kopka, J., Willmitzer, L., and Fernie, A. R. Gas chromatography mass spectrometry–based metabolite profiling in plants. *Nature Protocols*, 1(1):387–396. 85

[110] Liu, G., Wong, L., and Chua, H. N. (2009). Complex discovery from weighted PPI networks. *Bioinformatics*, 25(15):1891–1897. 3

[111] Liu, T., Lu, D., Zhang, H., Zheng, M., Yang, H., Xu, Y., Luo, C., Zhu, W., Yu, K., and Jiang, H. (2016). Applying high-performance computing in drug discovery and molecular simulation. *National science review*, 3(1):49–63. 14

[112] Liu, Z.-P., Wu, L.-Y., Wang, Y., Zhang, X.-S., and Chen, L. (2010). Prediction of protein–RNA binding sites by a random forest method with combined features. *Bioinformatics*, 26(13):1616–1622. 3

[113] Loucoubar, C., Paul, R., Bar-Hen, A., Huret, A., Tall, A., Sokhna, C., Trape, J.-F., Ly, A. B., Faye, J., Badiane, A., Diakhaby, G., Sarr, F. D., Diop, A., Sakuntabhai, A., and Bureau, J.-F. (2011). An Exhaustive, Non-Euclidean, Non-Parametric Data Mining Tool for Unraveling the Complexity of Biological Systems – Novel Insights into Malaria. *PLOS ONE*, 6(9):1–16. 2

[114] Lovász, L. and Prömel, H. J. (2004). Combinatorics. *Oberwolfach Reports*, 1(1):5–110. 31, 67, 70

[115] Luo, J., Schumacher, M., Scherer, A., Sanoudou, D., Megherbi, D., Davison, T., Shi, T., Tong, W., Shi, L., Hong, H., Zhao, C., Elloumi, F., Shi, W., Thomas, R., Lin, S., Tillinghast, G., Liu, G., Zhou, Y., Herman, D., Li, Y., Deng, Y., Fang, H., Bushel, P., Woods, M., and Zhang, J. (2010). A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *The Pharmacogenomics Journal*, 10(4):278–291. 25

[116] Maizel, J. V. (1988). Supercomputing in molecular biology: applications to sequence analysis. *IEEE Engineering in Medicine and Biology Magazine*, 7(4):27–30. 14

[117] Mangeot-Peter, L., Tschaplinski, T., Engle, N., Veneault-Fourrey, C., Martin, F., and Deveau, A. (2020). Impacts of soil microbiome variations on root colonization by fungi and bacteria and on the metabolome of populus tremula× alba. *Phytobiomes Journal*, 4(2):142–155. 20

[118] Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., and Califano, A. (2006). Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(1):S7. 34, 44

[119] McKown, A. D., Klápště, J., Guy, R. D., El-Kassaby, Y. A., and Mansfield, S. D. (2018). Ecological genomics of variation in bud-break phenology and mechanisms of response to climate warming in Populus trichocarpa. *New Phytologist*, 220(1):300–316. 16

[120] McKown, A. D., Klápště, J., Guy, R. D., Geraldes, A., Porth, I., Hannemann, J., Friedmann, M., Muchero, W., Tuskan, G. A., Ehlting, J., et al. (2014). Genome-wide association implicates numerous genes underlying ecological trait variation in natural populations of populus trichocarpa. *New Phytologist*, 203(2):535–553. 16

[121] Miles, C. and Wayne, M. (2008). Quantitative trait locus (QTL) analysis. *Nature Education 1 (1)*, 208. 21

[122] Muthukrishnan, R. and Rohini, R. (2016). LASSO: A feature selection technique in predictive modeling for machine learning. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pages 18–20. 9

[123] Navarro, F. C. P., Mohsen, H., Yan, C., Li, S., Gu, M., Meyerson, W., and Gerstein, M. (2019). Genomics and data science: an application within an umbrella. *Genome Biology*, 20(1):109. 2

[124] Nicodemus, K. K. and Malley, J. D. (2009). Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics*, 25(15):1884–1890. 24

[125] Niederhuth, C. E., Bewick, A. J., Ji, L., Alabady, M. S., Kim, K. D., Li, Q., Rohr, N. A., Rambani, A., Burke, J. M., Udall, J. A., Egesi, C., Schmutz, J., Grimwood, J., Jackson,

S. A., Springer, N. M., and Schmitz, R. J. (2016). Widespread natural variation of DNA methylation within angiosperms. *Genome Biology*, 17(1):194. 18

[126] Niu, B., Liang, C., Lu, Y., Zhao, M., Chen, Q., Zhang, Y., Zheng, L., and Chou, K.-C. (2020). Glioma stages prediction based on machine learning algorithm combined with protein-protein interaction networks. *Genomics*, 112(1):837–847. 34

[127] Nygaard, V., Rødland, E. A., and Hovig, E. (2016). Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, 17(1):29–39. 25

[128] Osama, K., Mishra, B. N., and Somvanshi, P. (2015). Machine learning techniques in plant biology. In *PlantOmics: The Omics of plant science*, pages 731–754. Springer. 2

[129] Osbourn, A. (2010). Gene Clusters for Secondary Metabolic Pathways: An Emerging Theme in Plant Biology. *Plant Physiology*, 154(2):531 LP – 535. 3

[130] Pan, J.-Y., Yang, H.-J., Faloutsos, C., and Duygulu, P. (2004). Automatic Multimedia Cross-Modal Correlation Discovery. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 653–658, New York, NY, USA. Association for Computing Machinery. 31

[131] Paznikov, A. and Shichkina, Y. (2018). Algorithms for Optimization of Processor and Memory Affinity for Remote Core Locking Synchronization in Multithreaded Applications. 13

[132] Pelley, J. W. (2012). 18 - Recombinant DNA and Biotechnology. pages 161–169. W.B. Saunders, Philadelphia. 19

[133] Perkins, A. D. and Langston, M. A. (2009). Threshold selection in gene co-expression networks using spectral graph theory techniques. *BMC Bioinformatics*, 10(Suppl 11):S4. 32, 33

[134] Perrin, B.-E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d'Alché–Buc, F. (2003). Gene networks inference using dynamic Bayesian networks . *Bioinformatics*, 19(suppl$_2$) : $ii138 - -ii148$. 44

[135] Qi, Y. (2012). Random forest for bioinformatics. *Ensemble Machine Learning: Methods and ApplicatiOns*, pages 307–323. 3

[136] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106. 5

[137] Radivojević, T., Costello, Z., Workman, K., and Martin, H. G. (2020). A machine learning Automated Recommendation Tool for synthetic biology. *Nature communications*, 11(1):1–14. 2

[138] Rapaport, F., Khanin, R., Liang, Y., Pirun, M., Krek, A., Zumbo, P., Mason, C. E., Socci, N. D., and Betel, D. (2013). Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome biology*, 14(9):1–13. 18

[139] Redei, G. P. (1975). ARABIDOPSIS AS A GENETIC TOOL. *Annual Review of Genetics*, 9(1):111–127. 15

[140] Reich, D. E., Cargill, M., Bolk, S., Ireland, J., Sabeti, P. C., Richter, D. J., Lavery, T., Kouyoumjian, R., Farhadian, S. F., Ward, R., and Lander, E. S. (2001). Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204. 17

[141] Remeseiro, B. and Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in Biology and Medicine*, 112:103375. 7

[142] Roberts, L. D., Souza, A. L., Gerszten, R. E., and Clish, C. B. (2012). Targeted Metabolomics. *Current Protocols in Molecular Biology*, 98(1):30.2.1–30.2.24. 19

[143] Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25. 59

[144] Rohart, F., Gautier, B., Singh, A., and Lê Cao, K.-A. (2017). mixOmics: An R package for 'omics feature selection and multiple data integration. *PLOS Computational Biology*, 13(11):e1005752. 7

[145] Rousseeuw, P. J. and Hubert, M. (2018). Anomaly detection by robust statistics. *WIREs Data Mining and Knowledge Discovery*, 8(2):e1236. 23

[146] Ruan, J., Dean, A. K., and Zhang, W. (2010). A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC systems biology*, 4(1):1–21. 34

[147] Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517. 7

[148] Sanbonmatsu, K. Y. and Tung, C.-S. (2007). High performance computing in biology: Multimillion atom simulations of nanoscale systems. *Journal of Structural Biology*, 157(3):470–480. 14

[149] Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64. 27, 28

[150] Shah, R. D. and Meinshausen, N. (2014a). Random intersection trees. *The Journal of Machine Learning Research*, 15(1):629–654. 55

[151] Shah, R. D. and Meinshausen, N. (2014b). Random intersection trees. *Journal of Machine Learning Research*, 15:629–654. 81

[152] Smid, M., Coebergh van den Braak, R. R. J., van de Werken, H. J. G., van Riet, J., van Galen, A., de Weerd, V., van der Vlugt-Daane, M., Bril, S. I., Lalmahomed, Z. S., Kloosterman, W. P., Wilting, S. M., Foekens, J. A., IJzermans, J. N. M., Coene, P. P. L. O., Dekker, J. W. T., Zimmerman, D. D. E., Tetteroo, G. W. M., Vles, W. J., Vrijland, W. W., Torenbeek, R., Kliffen, M., Carel Meijer, J. H., vd Wurff, A. A., Martens, J. W. M., Sieuwerts, A. M., and study Group, o. b. o. t. M. (2018). Gene length corrected trimmed mean of M-values (GeTMM) processing of RNA-seq data performs similarly in intersample analyses while improving intrasample comparisons. *BMC Bioinformatics*, 19(1):236. 59, 85

[153] Somerville, C. and Koornneef, M. (2002). A fortunate choice: the history of arabidopsis as a model plant. *Nature Reviews Genetics*, 3(11):883–889. 14, 15

[154] Sommer, C. and Gerlich, D. W. (2013). Machine learning in cell biology–teaching computers to recognize phenotypes. *Journal of cell science*, 126(24):5529–5539. 2

[155] Stracke, S., Kistner, C., Yoshida, S., Mulder, L., Sato, S., Kaneko, T., Tabata, S., Sandal, N., Stougaard, J., and Szczyglowski, K. (2002). A plant receptor-like kinase required for both bacterial and fungal symbiosis. *Nature*, 417(6892):959–962. 20

[156] Sun, J., Wu, Q., Shen, D., Wen, Y., Liu, F., Gao, Y., Ding, J., and Zhang, J. (2019). TSLRF: Two-Stage Algorithm Based on Least Angle Regression and Random Forest in genome-wide association studies. *Scientific Reports*, 9(1):18034. 11

[157] Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N. T., Legeay, M., Fang, T., and Bork, P. (2021). The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, 49(D1):D605–D612. 34

[158] Tarca, A. L., Carey, V. J., Chen, X.-w., Romero, R., and Drăghici, S. (2007). Machine learning and its applications to biology. *PLoS computational biology*, 3(6):e116. 2

[159] Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288. 9

[160] Timm, C. M., Pelletier, D. A., Jawdy, S. S., Gunter, L. E., Henning, J. A., Engle, N., Aufrecht, J., Gee, E., Nookaew, I., Yang, Z., Lu, T.-Y., Tschaplinski, T. J., Doktycz, M. J., Tuskan, G. A., and Weston, D. J. (2016). Two Poplar-Associated Bacterial Isolates Induce Additive Favorable Responses in a Constructed Plant-Microbiome System. *Frontiers in Plant Science*, 7:497. 60

[161] Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization. *J. Mach. Learn. Res.*, 3:1415–1438. 9

[162] Tringe, S. G. and Hugenholtz, P. (2008). A renaissance for the pioneering 16S rRNA gene. *Current Opinion in Microbiology*, 11(5):442–446. 20

[163] Tschaplinski, T. J., Abraham, P. E., Jawdy, S. S., Gunter, L. E., Martin, M. Z., Engle, N. L., Yang, X., and Tuskan, G. A. (2019). The nature of the progression of drought stress drives differential metabolomic responses in populus deltoides. *Annals of Botany*, 124(4):617–626. 19

[164] Tschaplinski, T. J., Plett, J. M., Engle, N. L., Deveau, A., Cushman, K. C., Martin, M. Z., Doktycz, M. J., Tuskan, G. A., Brun, A., Kohler, A., et al. (2014). Populus trichocarpa and populus deltoides exhibit different metabolomic responses to colonization by the symbiotic fungus laccaria bicolor. *Molecular Plant-Microbe Interactions*, 27(6):546–556. 19, 22

[165] Tschaplinski, T. J., Standaert, R. F., Engle, N. L., Martin, M. Z., Sangha, A. K., Parks, J. M., Smith, J. C., Samuel, R., Jiang, N., Pu, Y., Ragauskas, A. J., Hamilton, C. Y., Fu, C., Wang, Z.-Y., Davison, B. H., Dixon, R. A., and Mielenz, J. R. (2012). Down-regulation of the caffeic acid O-methyltransferase gene in switchgrass reveals a novel monolignol analog. *Biotechnology for Biofuels*, 5(1):71. 60

[166] Tullsen, D. M., Eggers, S. J., Emer, J. S., Levy, H. M., Lo, J. L., and Stamm, R. L. (1996). Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor. *SIGARCH Comput. Archit. News*, 24(2):191–202. 12

[167] Tullsen, D. M., Eggers, S. J., and Levy, H. M. (1995). Simultaneous Multithreading: Maximizing on-Chip Parallelism. In *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, ISCA '95, pages 392–403, New York, NY, USA. Association for Computing Machinery. 12

[168] Tullsen, D. M., Lo, J. L., Eggers, S. J., and Levy, H. M. (1999). Supporting fine-grained synchronization on a simultaneous multithreading processor. In *Proceedings Fifth International Symposium on High-Performance Computer Architecture*, pages 54–58. 13

[169] Tuskan, G. (2007). Bioenergy, genomics, and accelerated domestication: a us example. *FAO, Papers and Presentations from The Role of Agricultural Biotechnologies for Production of Bioenergy in Developing Countries, http://www.fao. org/biotech/seminaroct2007.htm*. 15

[170] Tuskan, G., Slavov, G., DiFazio, S., Muchero, W., Pryia, R., Schackwitz, W., Martin, J., Rokhsar, D., Sykes, R., Davis, M., et al. (2011a). Populus resequencing: towards genome-wide association studies. In *BMC Proceedings*, volume 5, pages 1–1. BioMed Central. 16

[171] Tuskan, G., Slavov, G., DiFazio, S., Muchero, W., Pryia, R., Schackwitz, W., Martin, J., Rokhsar, D., Sykes, R., Davis, M., Studer, M., and Wyman, C. (2011b). Populus resequencing: towards genome-wide association studies. *BMC Proceedings*, 5(7):I21. 16

[172] Tuskan, G. A., Difazio, S., Jansson, S., Bohlmann, J., Grigoriev, I., Hellsten, U., Putnam, N., Ralph, S., Rombauts, S., Salamov, A., et al. (2006a). The genome of black cottonwood, populus trichocarpa (torr. & gray). *science*, 313(5793):1596–1604. 16

[173] Tuskan, G. A., DiFazio, S., Jansson, S., Bohlmann, J., Grigoriev, I., Hellsten, U., Putnam, N., Ralph, S., Rombauts, S., Salamov, A., Schein, J., Sterck, L., Aerts, A., Bhalerao, R. R., Bhalerao, R. P., Blaudez, D., Boerjan, W., Brun, A., Brunner, A., Busov, V., Campbell, M., Carlson, J., Chalot, M., Chapman, J., Chen, G.-L., Cooper, D., Coutinho, P. M., Couturier, J., Covert, S., Cronk, Q., Cunningham, R., Davis, J., Degroeve, S., Déjardin, A., dePamphilis, C., Detter, J., Dirks, B., Dubchak, I., Duplessis, S., Ehlting, J., Ellis, B., Gendler, K., Goodstein, D., Gribskov, M., Grimwood, J., Groover, A., Gunter, L., Hamberger, B., Heinze, B., Helariutta, Y., Henrissat, B., Holligan, D., Holt, R., Huang, W., Islam-Faridi, N., Jones, S., Jones-Rhoades, M., Jorgensen, R., Joshi, C., Kangasjärvi, J., Karlsson, J., Kelleher, C., Kirkpatrick, R., Kirst, M., Kohler, A., Kalluri, U., Larimer, F., Leebens-Mack, J., Leplé, J.-C., Locascio, P., Lou, Y., Lucas, S., Martin, F., Montanini, B., Napoli, C., Nelson, D. R., Nelson, C., Nieminen, K., Nilsson, O., Pereda, V., Peter, G., Philippe, R., Pilate, G., Poliakov, A., Razumovskaya, J., Richardson, P., Rinaldi, C., Ritland, K., Rouzé, P., Ryaboy, D., Schmutz, J., Schrader, J., Segerman, B., Shin, H., Siddiqui, A., Sterky, F., Terry, A., Tsai, C.-J., Uberbacher, E., Unneberg, P., Vahala, J., Wall, K., Wessler, S., Yang, G., Yin, T., Douglas, C., Marra, M., Sandberg, G., Van de Peer, Y., and Rokhsar, D. (2006b). The genome of black cottonwood, populus trichocarpa (torr. & gray). *Science*, 313(5793):1596–1604. 44, 59

[174] Tuskan, G. A., Mewalal, R., Gunter, L. E., Palla, K. J., Carter, K., Jacobson, D. A., Jones, P. C., Garcia, B. J., Weighill, D. A., Hyatt, P. D., et al. (2018). Defining the genetic components of callus formation: A gwas approach. *PloS one*, 13(8):e0202519. 22

[175] Tyler, A. L., Asselbergs, F. W., Williams, S. M., and Moore, J. H. (2009). Shadows of complexity: what biological networks reveal about epistasis and pleiotropy. *BioEssays*, 31(2):220–227. 2

[176] Van Dongen, S. M. (2000). Graph clustering by flow simulation. 30

[177] Van Norman, J. M. and Benfey, P. N. (2009). Arabidopsis thaliana as a model organism in systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(3):372–379. 15

[178] Vijesh, N., Chakrabarti, S. K., and Sreekumar, J. (2013). Modeling of gene regulatory networks: A review. *Journal of Biomedical Science and Engineering*, 6(02):223. 34

[Walker and Dongarra] Walker, D. W. and Dongarra, J. J. Mpi: a standard message passing interface. *Supercomputer*. 13, 39

[180] Wang, K., Liu, S., Svoboda, L. K., Rygiel, C. A., Neier, K., Jones, T. R., Colacino, J. A., Dolinoy, D. C., and Sartor, M. A. (2020). Tissue- and Sex-Specific DNA Methylation Changes in Mice Perinatally Exposed to Lead (Pb) . 18

[181] Wang, W. Y. S., Barratt, B. J., Clayton, D. G., and Todd, J. A. (2005). Genome-wide association studies: theoretical and practical concerns. *Nature Reviews Genetics*, 6(2):109–118. 21

[182] Weighill, D., Jones, P., Bleker, C., Ranjan, P., Shah, M., Zhao, N., Martin, M., DiFazio, S., Macaya-Sanz, D., Schmutz, J., et al. (2019). Multi-phenotype association decomposition: unraveling complex gene-phenotype relationships. *Frontiers in genetics*, 10:417. 21

[183] Weighill, D., Jones, P., Shah, M., Ranjan, P., Muchero, W., Schmutz, J., Sreedasyam, A., Macaya-Sanz, D., Sykes, R., Zhao, N., Martin, M. Z., DiFazio, S., Tschaplinski, T. J., Tuskan, G., and Jacobson, D. (2018a). Pleiotropic and Epistatic Network-Based Discovery: Integrated Networks for Target Gene Discovery. *Frontiers in Energy Research*, 6:30. 3

[184] Weighill, D., Jones, P., Shah, M., Ranjan, P., Muchero, W., Schmutz, J., Sreedasyam, A., Macaya-Sanz, D., Sykes, R., Zhao, N., Martin, M. Z., DiFazio, S., Tschaplinski, T. J.,

Tuskan, G., and Jacobson, D. (2018b). Pleiotropic and Epistatic Network-Based Discovery: Integrated Networks for Target Gene Discovery. *Frontiers in Energy Research*, 6:30. 15, 59

[185] Whitacre, J. M. (2010). Degeneracy: a link between evolvability, robustness and complexity in biological systems. *Theoretical Biology and Medical Modelling*, 7(1):6. 2

[186] Wisecaver, J. H., Borowsky, A. T., Tzin, V., Jander, G., Kliebenstein, D. J., and Rokas, A. (2017). A global coexpression network approach for connecting genes to specialized metabolic pathways in plants. *The Plant Cell*, 29(5):944–959. 34

[187] Wood, D. E., Lu, J., and Langmead, B. (2019). Improved metagenomic analysis with Kraken 2. *Genome Biology*, 20(1):257. 20, 86

[188] Wright, M. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77(1):1–17. 5, 40

[189] Wu, S., Alseekh, S., Cuadros-Inostroza, Á., Fusari, C. M., Mutwil, M., Kooke, R., Keurentjes, J. B., Fernie, A. R., Willmitzer, L., and Brotman, Y. (2016). Combined Use of Genome-Wide Association Data and Correlation Networks Unravels Key Regulators of Primary Metabolism in Arabidopsis thaliana. *PLOS Genetics*, 12(10):1–31. 85

[190] Wullschleger, S. D., Jansson, S., and Taylor, G. (2002). Genomics and forest biology: Populus emerges as the perennial favorite. 15

[191] Xu, C. and Jackson, S. A. (2019). Machine learning and complex biological data. *Genome Biology*, 20(1):76. 2

[192] Yang, B., Gao, W., and Li, M. (2019). On the Robust Splitting Criterion of Random Forest. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1420–1425. 5

[193] Yao, D., Zhan, X., Zhan, X., Kwoh, C. K., Li, P., and Wang, J. (2020). A random forest based computational model for predicting novel lncRNA-disease associations. *BMC bioinformatics*, 21(1):1–18. 3

[194] Zhang, B. and Horvath, S. (2005a). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1). 34, 58, 83

[195] Zhang, B. and Horvath, S. (2005b). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1). 34

[196] Zhang, B. and Horvath, S. (2005c). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1). 34

[197] Zhang, H., Lang, Z., and Zhu, J.-K. (2018a). Dynamics and function of dna methylation in plants. *Nature reviews Molecular cell biology*, 19(8):489–506. 17

[198] Zhang, J., Yang, Y., Zheng, K., Xie, M., Feng, K., Jawdy, S. S., Gunter, L. E., Ranjan, P., Singan, V. R., Engle, N., Lindquist, E., Barry, K., Schmutz, J., Zhao, N., Tschaplinski, T. J., LeBoldus, J., Tuskan, G. A., Chen, J.-G., and Muchero, W. (2018b). Genome-wide association studies and expression-based quantitative trait loci analyses reveal roles of hct2 in caffeoylquinic acid biosynthesis and its regulation by defense-responsive transcription factors in populus. *New Phytologist*, 220(2):502–516. 44, 59

[199] Zhang, Y., Parmigiani, G., and Johnson, W. E. (2020). ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genomics and Bioinformatics*, 2(3). 25

[200] Zhao, Q., Zeng, Y., Yin, Y., Pu, Y., Jackson, L. A., Engle, N. L., Martin, M. Z., Tschaplinski, T. J., Ding, S.-Y., Ragauskas, A. J., and Dixon, R. A. (2015). Pinoresinol reductase 1 impacts lignin distribution during secondary cell wall biosynthesis in Arabidopsis. *Phytochemistry*, 112:170–178. 60

[201] Zhao, W., Langfelder, P., Fuller, T., Dong, J., Li, A., and Hovarth, S. (2010). Weighted gene coexpression network analysis: state of the art. *Journal of biopharmaceutical statistics*, 20(2):281–300. 34

[202] Zilberman, D., Gehring, M., Tran, R. K., Ballinger, T., and Henikoff, S. (2007). Genome-wide analysis of arabidopsis thaliana dna methylation uncovers an interdependence between methylation and transcription. *Nature genetics*, 39(1):61–69. 18

[203] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320. 10

# Appendix

# A Chapter 2 Supplementary Images

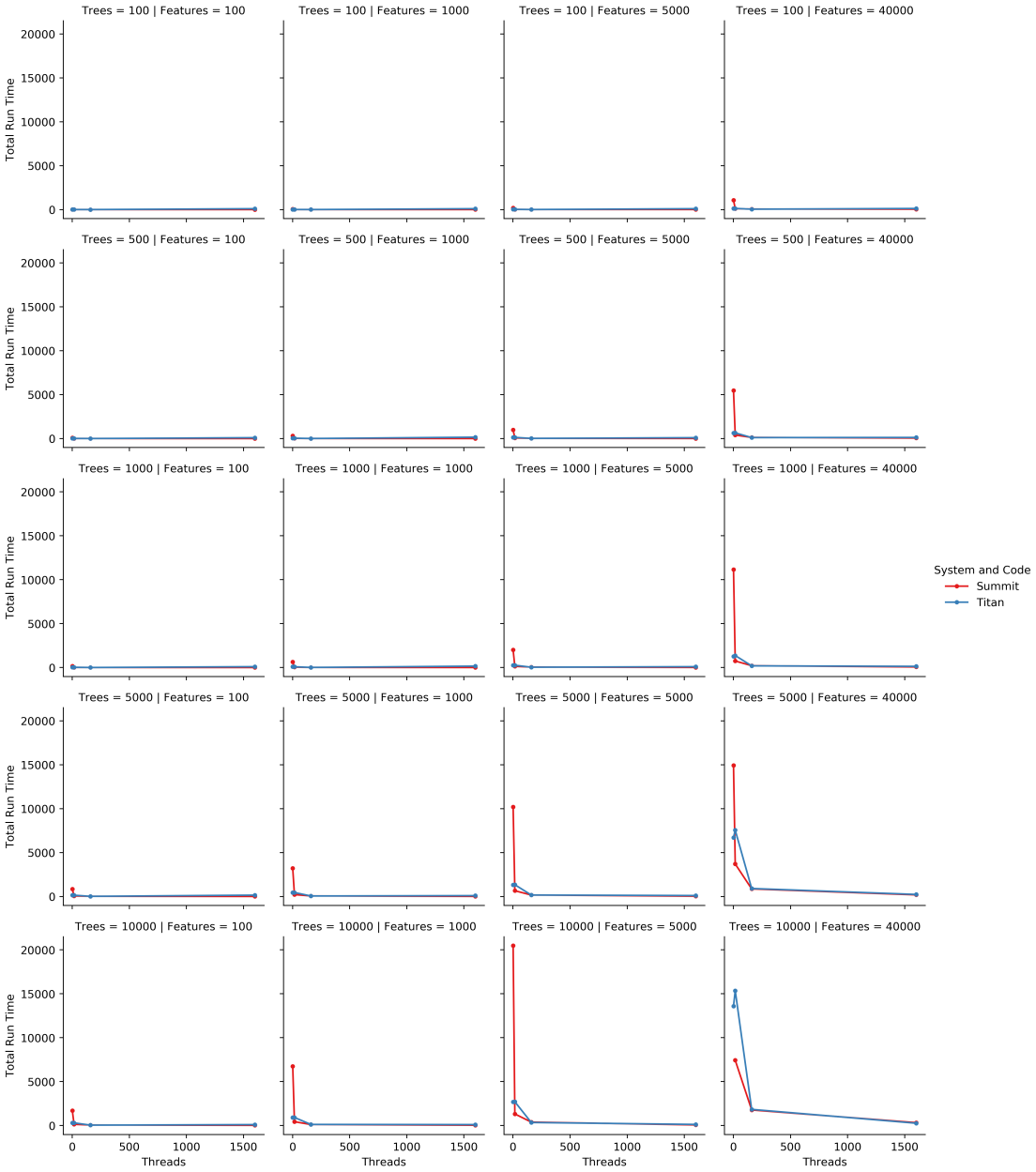



**Figure 1:** Runtime on HPC Systems. The full grid of graphs comparing Summit and Titan times to completion for the C++ code. The red line is for Summit and the blue line is for Titan.

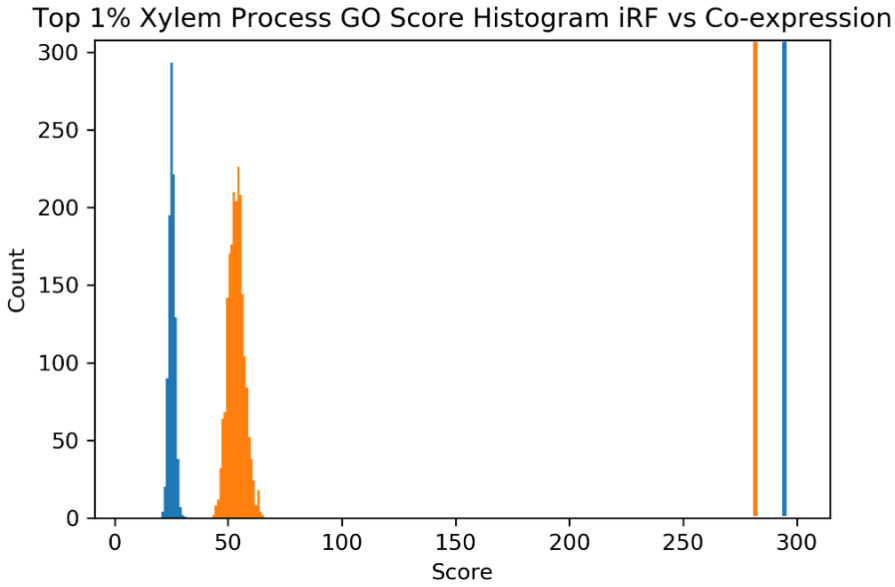



**Figure 2:** The null distribution histogram and true intersect score of the iRF network is shown in blue. The co-expression null distribution and corresponding network score are in orange.

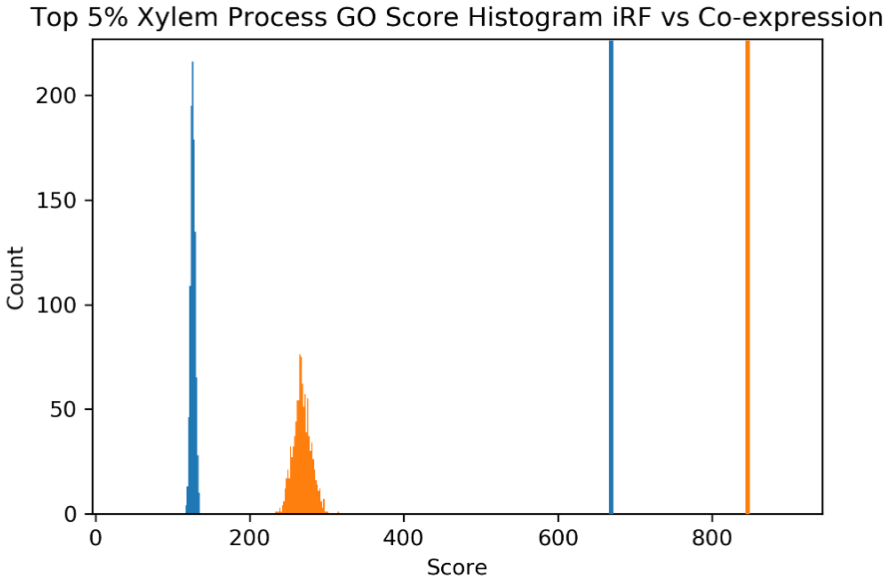



**Figure 3:** The null distribution histogram and true intersect score of the iRF network is shown in blue. The co-expression null distribution and corresponding network score are in orange.

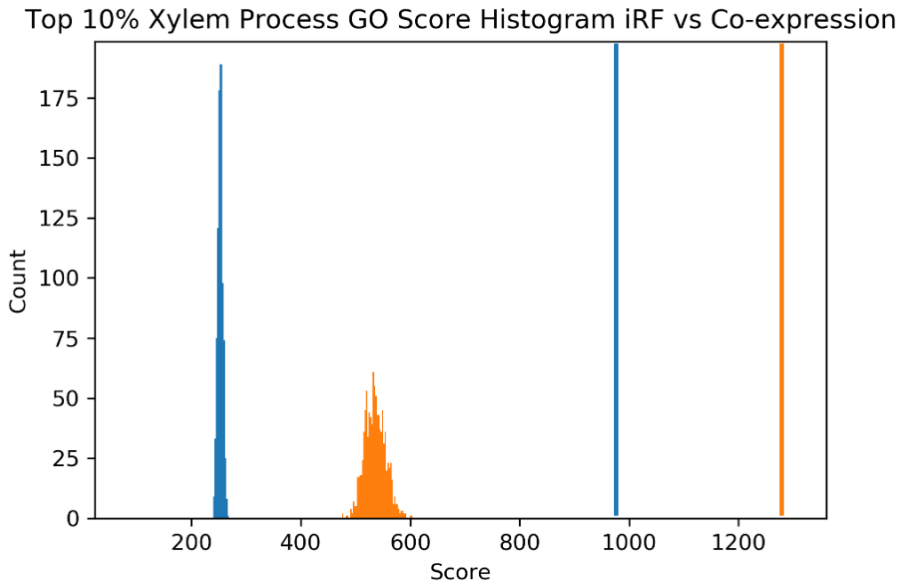



**Figure 4:** The null distribution histogram and true intersect score of the iRF network is shown in blue. The co-expression null distribution and corresponding network score are in orange.

# B Chapter 4 Supplementary Images

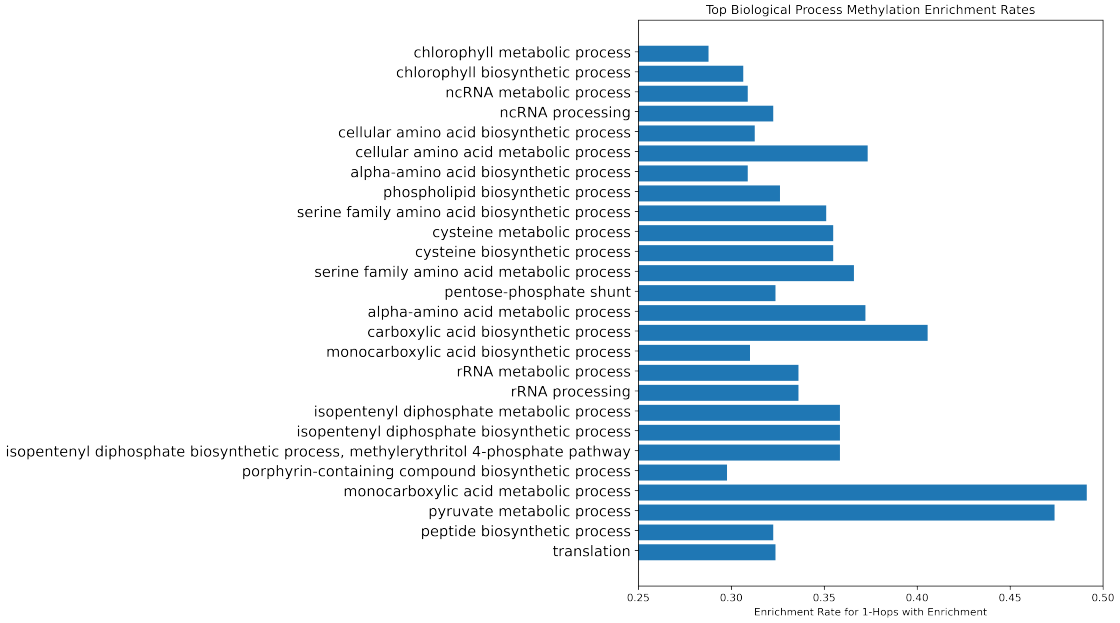



**Figure 5:** The rate of enrichment of the most highly enriched GO Biological Process terms for the genes from the 1-hop networks that originated from the methylation nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 6, limiting GO terms to those 6 steps deep or lower, removing the broad generic terms.

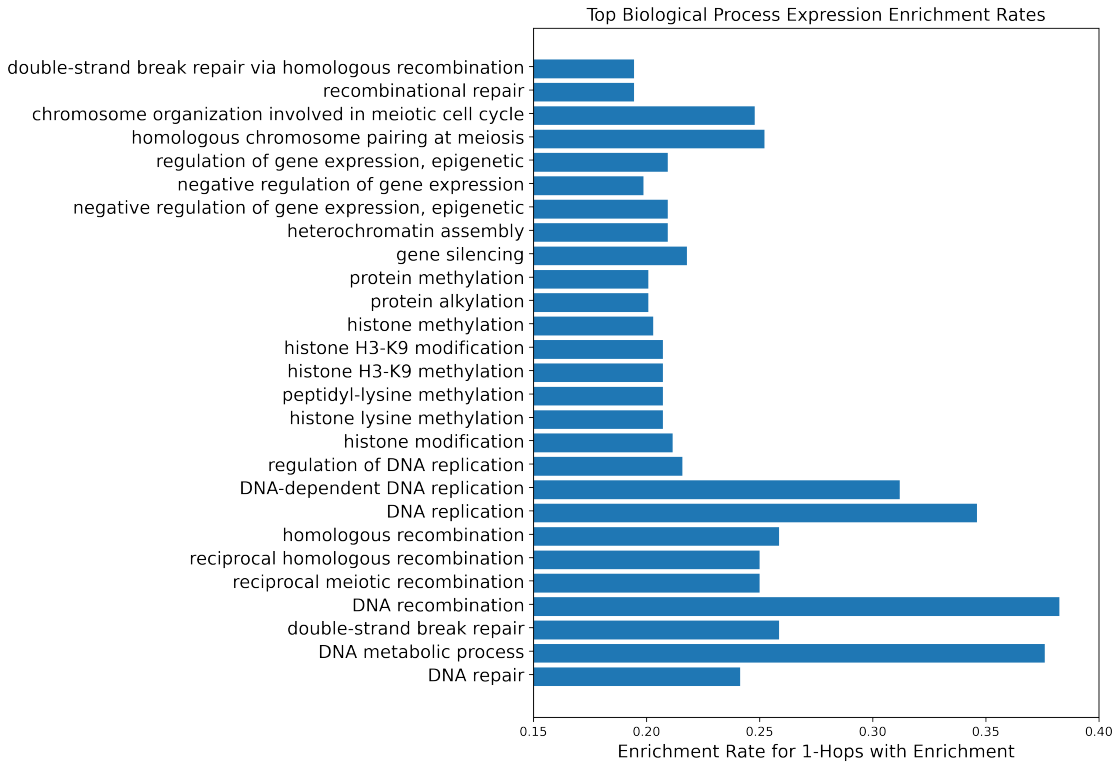



**Figure 6:** The rate of enrichment of the most highly enriched GO terms for the genes from the 1-hop networks that originated from the gene expression nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 6, limiting GO terms to those 6 steps deep or lower, removing the broad generic terms.

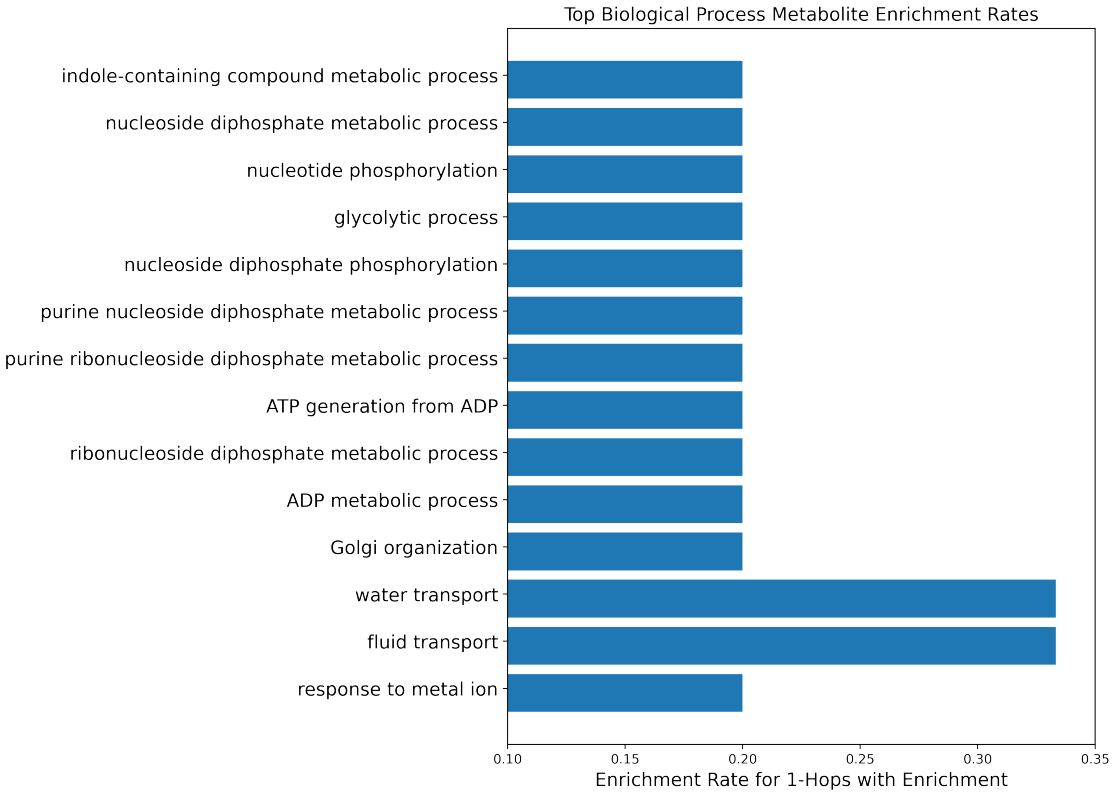



**Figure 7:** The rate of enrichment of the most highly enriched GO Biological Process terms for the genes from the 1-hop networks that originated from the metabolite nodes. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

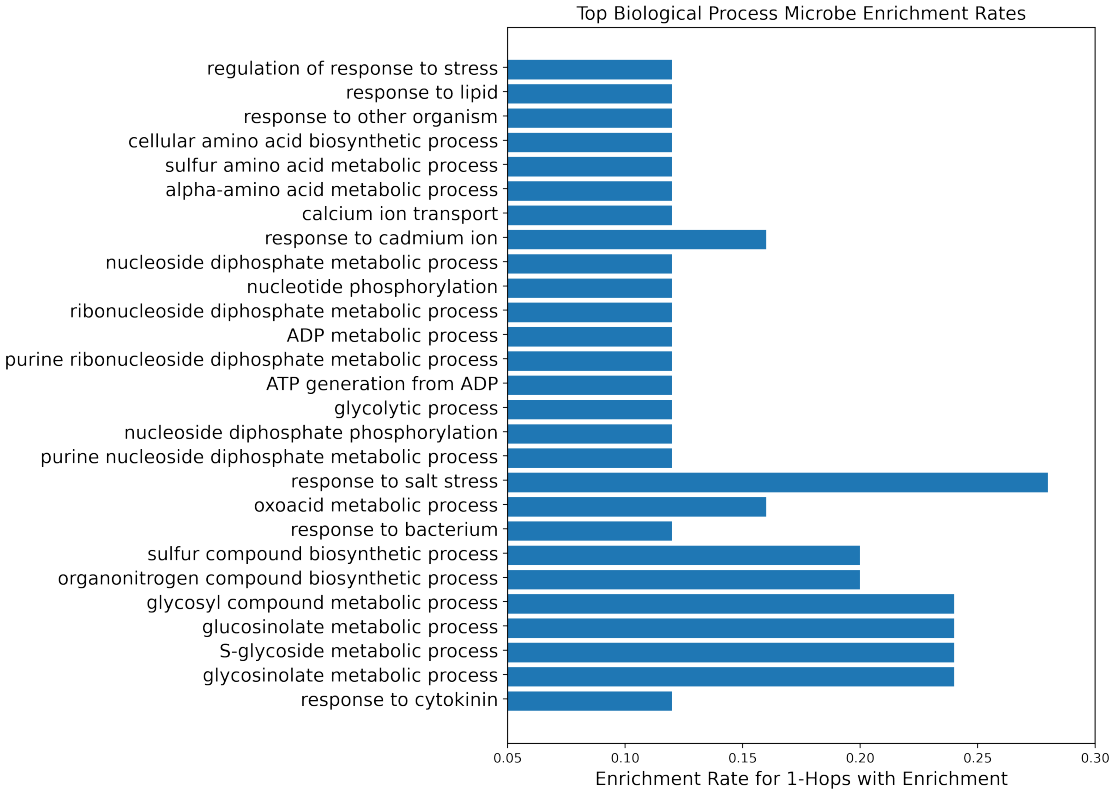



**Figure 8:** The rate of enrichment of the most highly enriched GO Biological Process terms for the genes from the 1-hop networks that originated from the microbe nodes that met the 10-gene connection cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

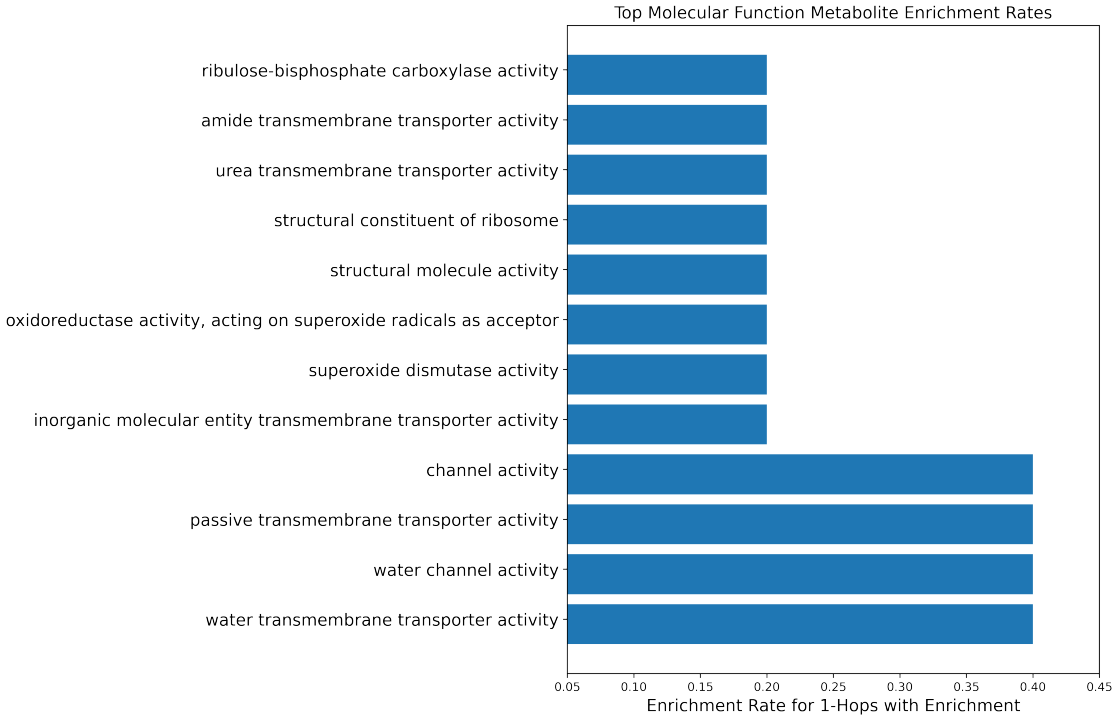



**Figure 9:** The rate of enrichment of the most highly enriched GO Molecular Functions terms for the genes from the 1-hop networks that originated from the metabolite nodes. Enrichments with p-value lower 0.05 were maintained. Depth was not limited due to scarcity of overall terms, due to gene set sizes.

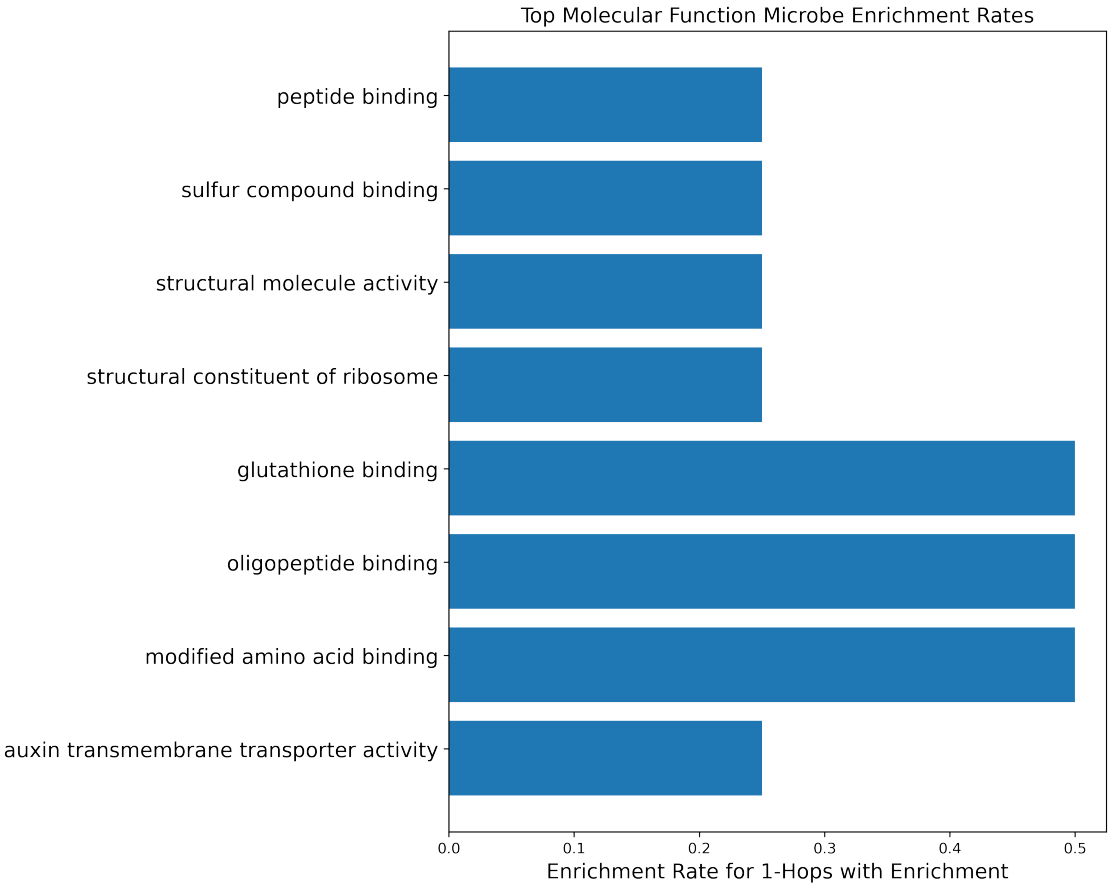



**Figure 10:** The rate of enrichment of the most highly enriched GO Molecular Functions terms for the genes from the 1-hop networks that originated from the microbe nodes that met the 10-gene cut off. Enrichments with p-value lower 0.05 were maintained. Depth was not limited due to scarcity of overall terms, due to gene set sizes.

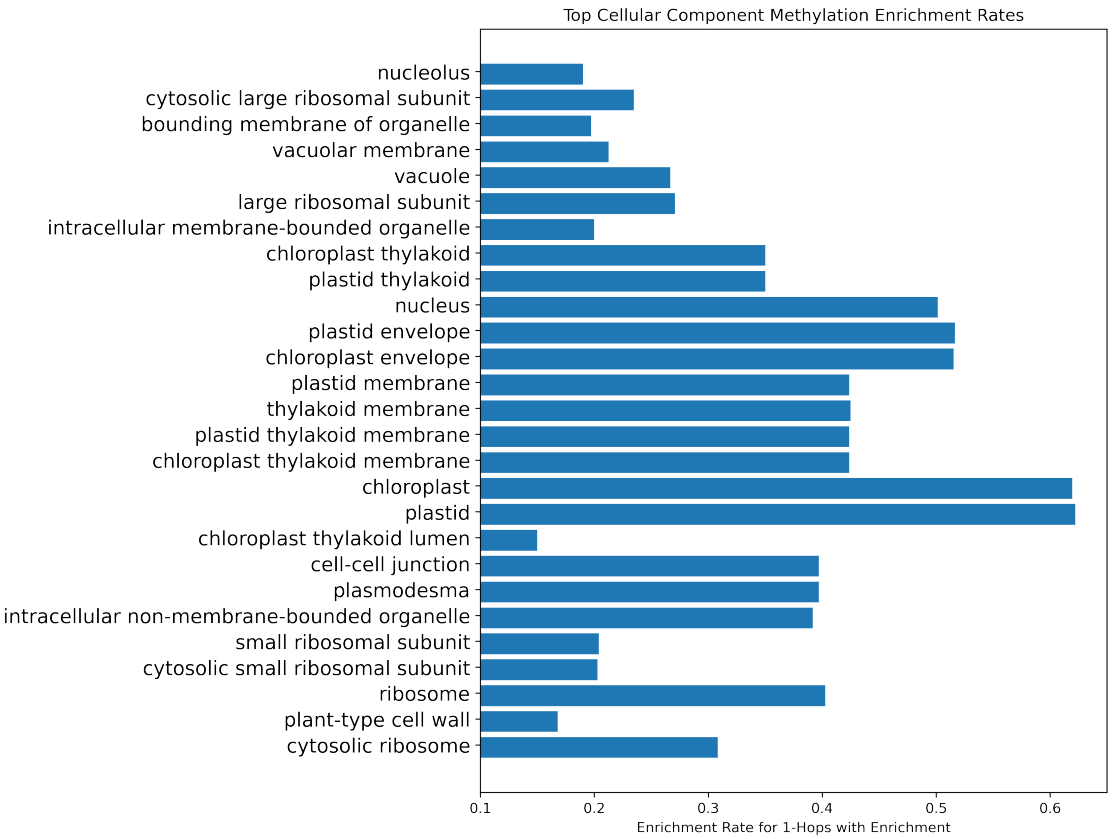



**Figure 11:** The rate of enrichment of the most highly enriched GO Cellular Component terms for the genes from the 1-hop networks that originated from the methylation nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

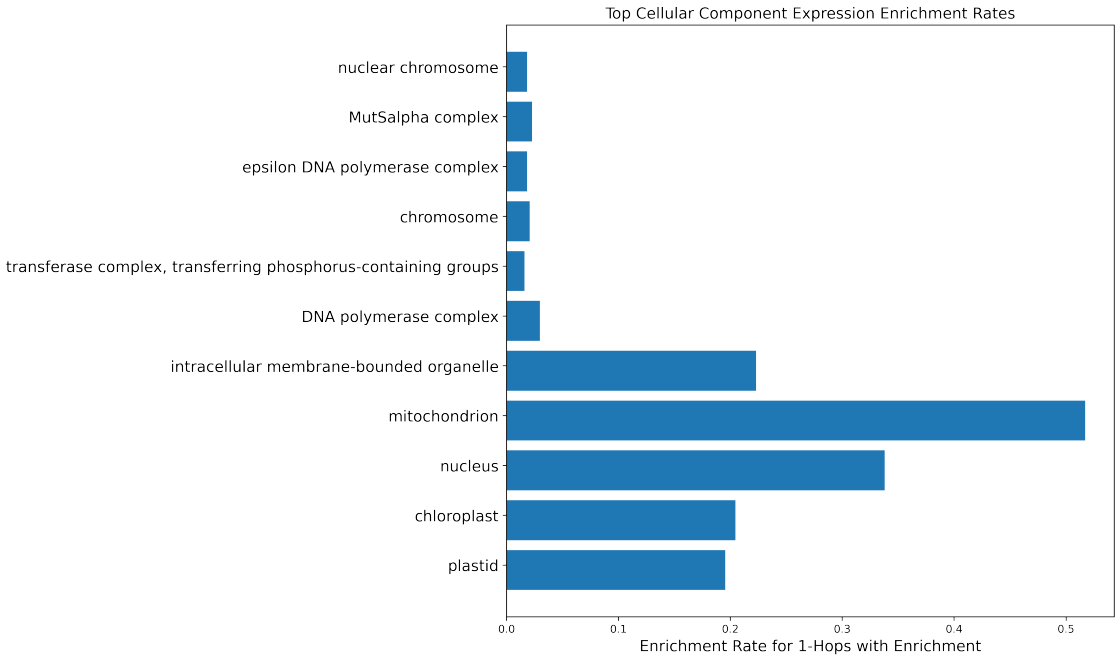



**Figure 12:** The rate of enrichment of the most highly enriched GO Cellular Component terms for the genes from the 1-hop networks that originated from the gene expression nodes that met the 200-500 node cut off. Enrichments with p-value lower 0.05 were maintained. Depth was set to 4, limiting GO terms to those 4 steps deep or lower, removing the broad generic terms.

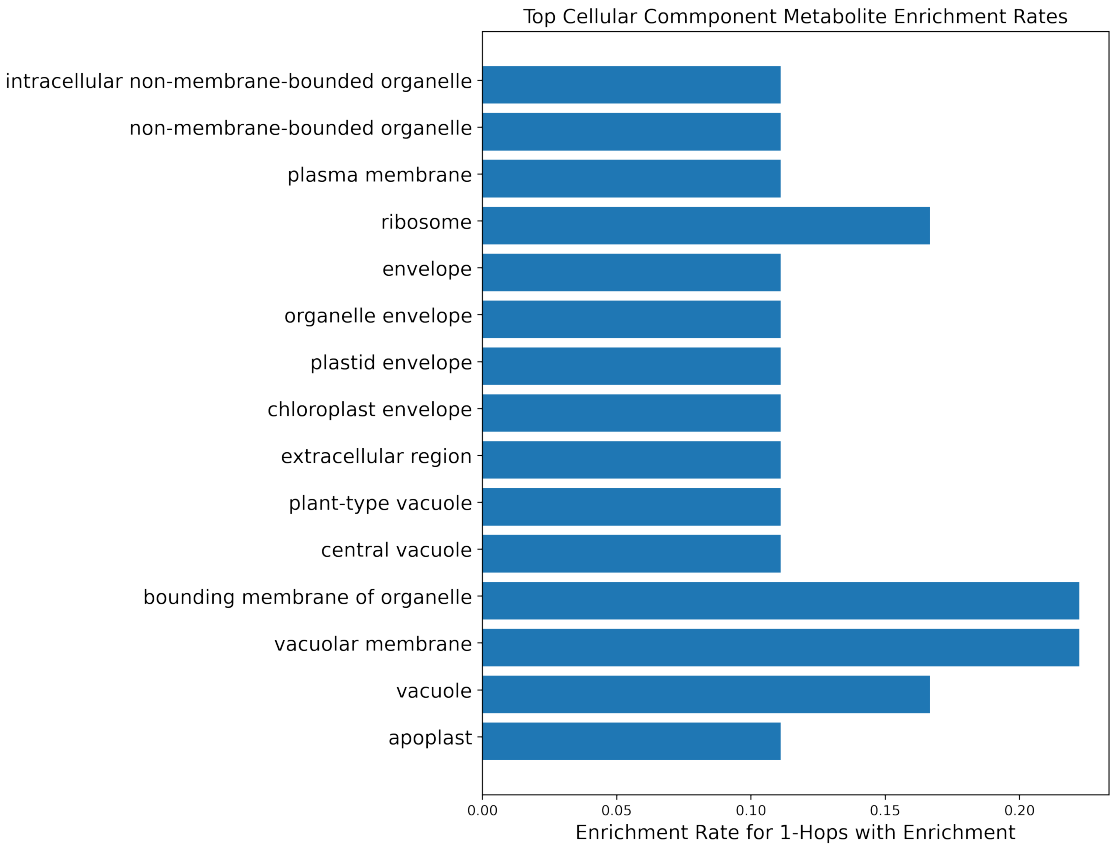



**Figure 13:** The rate of enrichment of the most highly enriched GO Cellular Component terms for the genes from the 1-hop networks that originated from the metabolite nodes. Enrichments with p-value lower 0.05 were maintained. Depth was not limited due to scarcity of overall terms, due to gene set sizes.

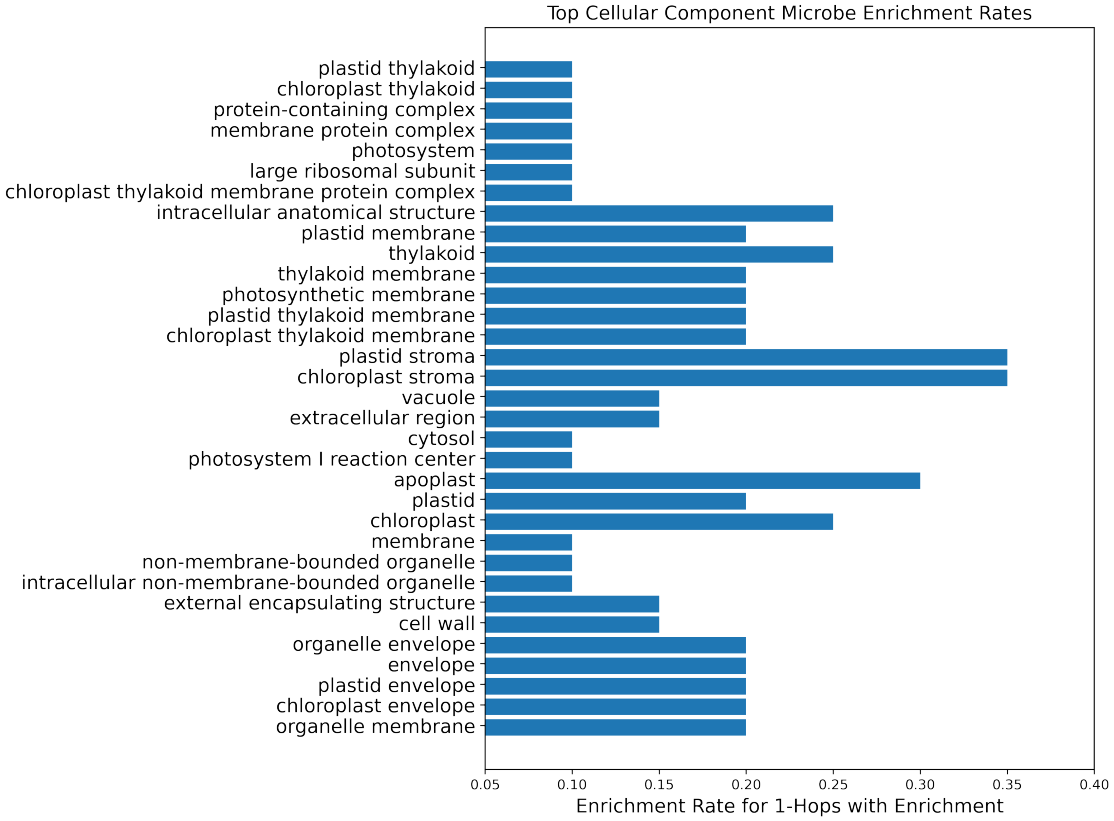



**Figure 14:** The rate of enrichment of the most highly enriched GO Cellular Component terms for the genes from the 1-hop networks that originated from the microbe nodes that met the 10-gene cut off. Enrichments with p-value lower 0.05 were maintained. Depth was not limited due to scarcity of overall terms, due to gene set sizes.

# Vita

Ashley was born in northern California, and moved to Decorah, Iowa with her family at the age of ten. She attended Decorah High School from 2008 to 2012, and went on to obtain a Bachelors of Arts Degree in Physics and Computer Science from Central College, in Pella, Iowa in 2016. Ashley pursued her doctorate in Data Science and Engineering through the Bredesen Center for Interdisciplinary Research and Graduate Education at the University of Tennessee Knoxville, and her work on explainable AI methods and uses was overseen by Dr. Daniel Jacobson, a PI in Computational Systems Biology at Oak Ridge National Laboratory.