ASSOCIATION RULE MINING OF BIOLOGICAL FIELD DATA SETS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Anuj Shrestha

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

September 2017

Fargo, North Dakota

North Dakota State University
Graduate School

Title

ASSOCIATION RULE MINING OF BIOLOGICAL FIELD DATA SETS

By

Anuj Shrestha

The Supervisory Committee certifies that this thesis complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Anne Denton
Chair

Dr. Simone Ludwig

Dr. Peter Bergholz

Approved:

November 16, 2017
Date

Dr. Kendall Nygard
Department Chair

ABSTRACT

Association rule mining is an important data mining technique, yet, its use in association analysis of biological data sets has been limited. This mining technique was applied on two biological data sets, a genome and a damselfly data set. The raw data sets were pre-processed, and then association analysis was performed with various configurations. The pre-processing task involves minimizing the number of association attributes in genome data and creating the association attributes in damselfly data. The configurations include generation of single/maximal rules and handling single/multiple tier attributes. Both data sets have a binary class label and using association analysis, attributes of importance to each of these class labels are found. The results (rules) from association analysis are then visualized using graph networks by incorporating the association attributes like support and confidence, differential color schemes and features from the pre-processed data.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Anne Denton for the continuous support of my master's study and research. I would also like to thank for her encouragement, motivation, enthusiasm and immense knowledge. Her guidance helped me in all the time of my research and writing of this thesis. She has been a tremendous mentor for me.

I would also like to thank the rest of my thesis committee: Dr. Peter Bergholz and Dr. Simone Ludwig, for serving as my committee members and generously offering their precious time, support and good will.

My sincere thanks also go to Dr. Peter Bergholz and Dr. Andre Delorme for their continuous patience, guidance and insightful feedbacks throughout the research. They have been instrumental in preparing this thesis as well.

This material is based upon work supported by the Bioinformatics Seed Grant Program NIH/UND and by the National Science Foundation through grant IIA-1355466.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARM.....association rule mining

lhs .....left-hand-side

rhs.....right-hand-side

Escherichia coli.....*E. coli*

NDAWN.....North Dakota Agricultural Weather Network

NCDC .....National Climate Data Center

RMSE .....Root Mean Square Error

1. INTRODUCTION

The advancement in high-throughput experiments has led to explosive growth in the amount of data in many scientific fields, and biology is no different. The potential for extracting useful information from the biological data sets is immense. The large amount of data in modern biology has changed the field into an information science [1] and made processing and interpretation of such large data sets a challenging area for computer scientists, biologists and alike. In this paper, we discuss two studies of biological data sets, a genomic and a damselfly data set.

Advances in DNA sequencing and mapping techniques have created many opportunities in bioinformatics [2]. The rise of technology like microarrays has led to a rise of subfields like genomics and proteomics to study the mechanisms inside the cell. The amount of data and computational requirement are increasing and anticipated to continue doing so [3]. Similarly, with the increased awareness of climate change, massive climatic information is being processed, stored and analyzed. Future climatic prediction has become a key research area, demanding processing of massive amounts of data [4]. Such large pool of available climatic information allows biologists to model the habitat conditions of various species and see how they are evolving and the effects of changes in climate. Data mining, as a subfield of computer science, plays a significant role in processing such massive quantities of data.

Data mining is the process of discovering useful insights from databases. It includes application of various methodologies and algorithmic approach to preprocess, cluster, classify and associate the information for useful knowledge retrieval [5]. It is influenced by multiple discipline such as machine learning, artificial intelligence, databases, statistics, pattern discovery and visualization [6]. It is used in a growing number of diverse application areas such as finance, biological sciences, web applications, banking, ecological sciences, security, climate modeling to

name a few [7]. In this paper, we use association analysis, an important data mining technique, for discovering useful relationship.

Association Rule Mining (ARM) is one of the important and commonly used data mining techniques. It was first introduced by (Agrawal et al., 1993) [8], and is useful for discovering hidden relationships in large data sets. It is popular for its use in recommender systems, promotional bundling, cross-selling and customer relationship management. It has also been integrated into analysis of Web usage mining, clustering and association-based classification [9]. ARM, unlike clustering and classification, has seen comparatively less use as a data analyst tool in biological sciences and bioinformatics. However, it has been utilized, for example, for extracting useful information from protein-protein interaction data sets and for protein function prediction [10].

1.1. Problem Statement

In this paper, we discuss the application of ARM on two biological data sets – 1. Genome variants association in *Escherichia coli (E. coli)*, and 2. Habitat association in damselflies. The objective is to understand the relationship of either binary or multi-level attributes with a single class label in these data sets. The attributes are either filtered, using graph network and statistical significance, or estimated, using geo-statistics and discretization techniques. We then use association analysis to find out how the attributes, individually and as a group, relates to the class label.

The association analysis focuses on generating "single" and "maximal" rules. The "single" rules helps to identify all individual attribute associated with a class label. Meanwhile, the "maximal" rules helps to identify a group of attributes that together form associations with a class label. Such maximal rules produce a minimal representation of the associated attributes. Likewise, the graphical representation of these rules is critical to make sense of the data. We enhance existing association representation techniques using graph networks by incorporating the association properties like

support and confidence, peculiar data set properties and sizing and coloring the associated attributes based on their relevance to the class label in context.

In data set (1), we have genome variants of *E. coli* originating from surface soils in either "field" or "forest" sources. We intend to find list of genome variants that co-occur for each origin sources using ARM. Such co-occurrences could explain high likelihood of selection for those variants in those soils. Before using ARM, we use filtering for removing binary attributes that could occur randomly without contributing to the class label.

In data set (2), we have geo-locations of two species of damselflies, "River Jewelwing" and "American Rubyspot". Both species are similar, yet rarely found together. We intend to estimate and then compare the climatic habitat of these species using ARM. Such comparison could explain if local climate is a factor in dispersed occurrence of these species. Before using ARM, we use pre-processing for estimating the attribute values and discretizing them into multi-level attribute.

2. CONCEPTS

2.1. General Association Concepts

Association rules helps to uncover hidden relationships from large data sets. The goal of finding association rules is to extract co-occurrence patterns between the items of a data set. Association rules present statements in the form of "if…else" statements, where a set of items co-occurs with a separate set of items. First, the frequent itemsets are generated from candidate itemsets, and then rules are generated from these frequent itemsets.

To extract meaningful itemsets and rules, quality measures are used. The common quality measures are support and confidence. Fixed support threshold (*minsup*) and confidence threshold (*minconf*) are used to remove the uninteresting rules. Only the itemsets that had support greater or equal to *minsup* are used as frequent itemsets. Then, *minconf* is used to filter the important rules, where rules having confidence greater than or equal to *minconf* are considered.

Table 2-1: Example data set with alternative matrix representation

Transaction	Items	Class Label
T1	A, B, C	Q
T2	B, C	Q
T3	A, B	P
T4	B, C	Q
T5	A, B	P

A	B	C	P	Q
1	1	1	0	1
0	1	1	0	1
1	1	0	1	0
0	1	1	0	1
1	1	0	1	0

Note: i. (left) Example data set ii. (right) Alternative representation of example

In Table 2-1 (i), we have an example data set with 5 transactions (or samples). The items are A, B, C and we also have two class labels P or Q for each sample. Table 2-1 (ii) shows an alternative method of representing the example data set, where the class labels are also treated as items, with the

constraint that each transaction must have exactly one class label. Let's consider, *minsup* = 2/5 (0.4) and *minconf* = 80% (0.8). The itemset lattice is shown in Figure 2-1.

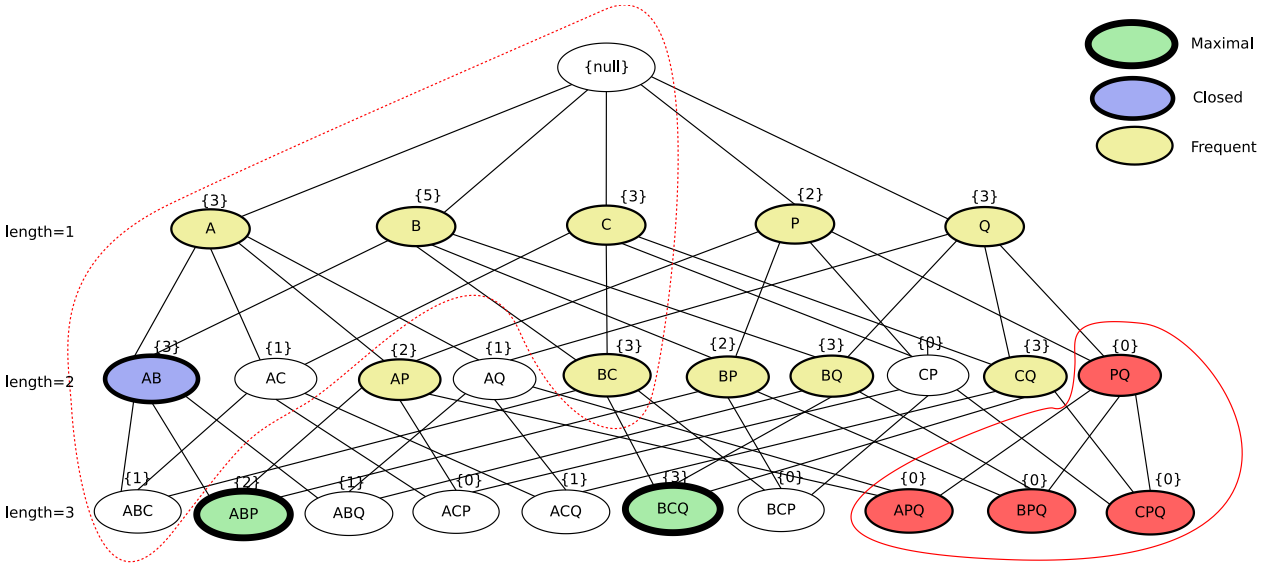

Figure 2-1: Itemset lattice for example data set

In Figure 2-1, the frequent itemsets are denoted by yellow nodes. The nodes inside the dotted red lines represent the itemsets that do not feature any class label and are not useful in our association study. The red nodes represent itemsets that feature two class labels and can never be generated in our association study. The blue and the green nodes represent closed and maximal itemset respectively, which are the most concise frequent itemsets and are discussed later. Itemsets are generated up to the length of 3 and the algorithm stops at that length for the example data set.

We will focus on rules that has at least one or more items in Left-Hand-Side (*lhs*) and only one-item in Right-Hand-Side (*rhs*), i.e., {Antecedent (*lhs*)} → {Consequent (*rhs*)}. In addition, in our study, we set the restriction that possible values of *rhs* can only be a class label (i.e., P or Q in the example data set). This introduces the concept of classification rule mining. Association rule mining finds all the rules existing in the database that satisfy *minsup* and *minconf* constraints. For classification rule mining, the constraint of a fixed *rhs* item is added. Classification rule mining is generally used to build classifier models using such association analysis, but we are not generating any models.

Instead, we want to extract information about the attribute combinations based on the principle of classification rule mining. Such predictors in our study would be:

- Genome Study: Attributes that predict the origin of the soil samples (field or forest).
- Damselfly Study: Attributes that predict the type of the damselfly samples (American Rubyspot or River Jewelwing)

The class labels in the example data set are P and Q. In the case of P, there are two transactions, and the *minsup* is 2/5. In Figure 2-1, the frequent itemsets featuring P are P, AP, BP, ABP. Each of these itemset have a support of 2/5. Similarly, in the case of Q, there are three transactions. The frequent itemsets are Q, BQ, CQ, BCQ with each itemset having support of 3/5. There are some basic problems with such a naïve approach. Such issues are discussed briefly:

- The use of such absolute support threshold for both the class labels is problematic because of the uneven distribution of transaction among the class labels. We utilize the concept of relative support threshold, where the *minsup* for a class label having lower number of transaction is fixed, and then the *minsup* for the other class label is relatively calculated.
- There are some items that do not look significant with a class label, yet they could feature in rules. For instance, BCQ is a frequent item. Rule from this itemset would be BC → Q. The support of the rule is support of BCQ (3/5), and confidence is 100% as whenever B and C are present the class label is always Q. But, if we look closely, B is present in all the transactions. If we consider BQ frequent itemset, then rule would be B → Q. The support is again 3/5, but the confidence is just 60% as BQ has support of 3/5 and B has support of 5/5. Since the *minconf* is 80%, so the rule fails. It could be that it is randomly occurring and does not have significance with the class label or it could have significance when combined with other items (like C). Irrespective of the significance, it is important to identify such items in rules. These items are referred to as *cross-support items*, and the patterns they from are

called as *hyperclique pattern* [11]. In our study, we identify such cross-support items using double pass of association rule mining and then using differential color schemes in the graphical representation of the rules.

- We also see a lot of overlapping frequent itemset, i.e., subsets of a frequent itemset are also frequent. This results in a larger number of rules. To avoid handling such many rules, we need a compact form of these rules. This is where the concepts of closed rules and maximal rules are useful. For closed and maximal rules, we need to formulate closed and maximal itemsets respectively first. Closed itemsets removes the subsets of an itemset if they share the same support. Likewise, maximal itemsets removes all the subset of an itemset. Figure 2-1 shows that there are 13 frequent items, 3 closed items and 2 maximal items in the example data set. It is to be noted that all maximal itemsets are closed itemsets, and all closed itemsets are frequent itemsets [12]. However, in the case of rules where the number of items are larger, closed rules still may produce relatively larger number of rules. This is because there are more subsets and these subsets could have different support. Whereas, in case of maximal rules, the itemset do not have any subset, and thus the rules generated will be minimal in terms of item representation.

2.2. Formal Definitions

Transactional Database: This contains all the data that is used in Association Rule Mining. Each row is a *transaction* (genome / habitat samples), and each column are the *items* (genome variant cluster / climatic variables).

Itemset: A set of one or more items is referred to as *itemset*.

Support: *Support* of an itemset X is the probability that a randomly chosen transaction will contain X. For instance, an item-set X has a support of 0.1 means 10% of the transaction has all the items in X.

Confidence: *Confidence* of a rule X → Y is the conditional probability that a randomly chosen transaction will contain all the items in Y if the transaction contains all the items in X. For instance, a rule X→ Y has a confidence of 0.9 means 90% of transactions that feature X also features Y.

Frequent Itemset: An itemset is *frequent* if it has support value greater than the given threshold support (*minsup*).

Rule: A *rule* is in a form of {Antecedent (*lhs*)} → {Consequent (*rhs*)}. It signifies *lhs* implies *rhs*, where *lhs* and *rhs* are disjoint. A rule provides information in the form of "if...then" statements. The strength of a rule is given by its support and confidence. A rule must pass the given minimum support (*minsup*) and minimum confidence (*minconf*) threshold to be a *frequent rule*. The antecedent and consequent together form the frequent itemset for a frequent rule. The *length* of a rule is the sum of items in antecedent and consequent. Throughout our analysis, we will only have one item in the consequent. For a rule X→ Y,

$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \qquad \& \qquad Confidence, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Closed Frequent Itemset: A frequent itemset X is said to be *closed* if it does not have any immediate superset that has the same support as X [12]. The rules generated from these closed frequent itemset are the *closed rules*. In the test example, AB was closed because it has a different support than ABP even though ABP is its superset and a frequent itemset.

Maximal Frequent Itemset: A frequent itemset X is said to be *maximal* if it does not have any immediate frequent superset [12]. The rules generated from these maximal frequent itemset are the *maximal rules*. In the test example, AB was not maximal because ABP is its superset and a frequent itemset.

2.3. Association Data Formulation

For both problems under consideration, a main portion of the task was to convert the data sets into a format appropriate for mining the association rules.

In the genome study, the actual properties of the attributes were ready for mining association rules, but there were too many attributes to make the process efficient, i.e., 22020 attributes. The problems with such large attributes number are highlighted below:

- many attributes have the same occurrence in each transaction, i.e., redundancy.
- similarly occurring attributes causes mining to longer rule length, i.e., complexity.
- large pool of attributes does not contribute significantly towards the class label, i.e., insignificance.

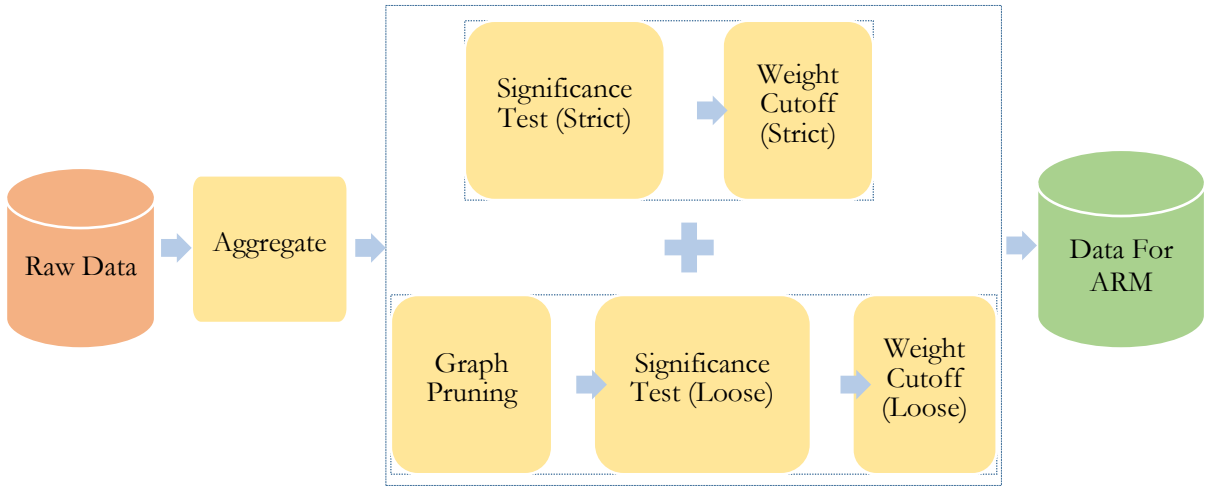

Figure 2-2: Formulation of genome data for ARM

We addressed the problems using a filtering pipeline as shown in Figure 2-2. The figure shows the high-level concepts behind reducing the number of attributes in consideration for the association study. Initially, the attributes that had the same occurrence throughout all the samples were aggregated as a single entity. Then, a graph-based approach was used to create a network of these attributes, where by, similar attributes are clustered together. We introduce a concept of weighted nodes and use it as cutoff and for pruning. We also use test of independence as a

significance testing to remove attributes that are statistically randomly occurring. Finally, we make association specific filtration like *minsup* presence, the finalized attributes are then ready for ARM.

In the damselfly study, we only had the locations and date of the samples. Based on location and date, we had to estimate the climatic conditions for the samples from known climatic sources. We tested with various geo-spatial interpolation methodologies to estimate the values for the climatic conditions at these locations. The problems with such estimations from known climate sources and their interpretation are highlighted below:

- the number and distribution of the source stations are less, i.e., low coverage.
- the climatic variables from the data source are limited, i.e., limited climatic attributes.
- converting continuous values into multi-tier classes i.e., discretization.

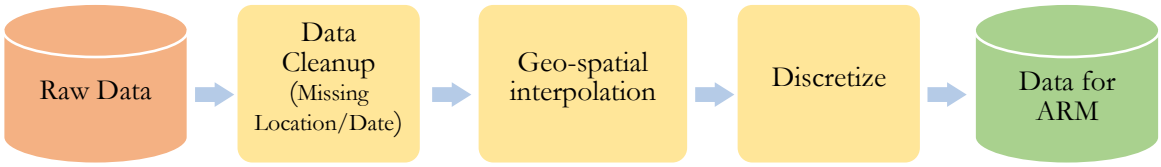

Figure 2-3: Formulation of damselfly data for ARM

We addressed the problems using pre-processing pipeline as shown in Figure 2-3. The figure shows the high-level concepts behind interpolating climatic attributes from known-sources, and then discretizing the values into proper classes. Initially, we had to find a well-known climatic source that contained the climatic attributes of interest and having good coverage. Out of the few alternatives, we found climatic data from North Dakota Agricultural Weather Network (NDAWN) suitable for our data. The NDAWN source contained stations that were spread throughout North Dakota (our region of interest) and covered the date ranges from 1990s to till date. Then, we had to find a suitable method to interpolate data from the source stations to our sample. We chose a method that returned the lowest RMSE. Once we obtained the interpolated continuous values for the sample points, we had to discretize the values for each of the climatic attribute into proper classes.

Discretizing into pre-defined number of classes would be problematic due to variation in the distribution ranges of each attribute. So, we determined the number and distribution of classes for each attribute by minimizing the squares of within cluster distance [13]. The discretized attributes are then ready for ARM.

Table 2-2: Summary of data of both data sets, and preparation for ARM

	Genome	**Damselfly**
Raw Data Set	Significantly more columns than rows. *131 rows VS 20020 columns*	Significantly less columns than rows. *158 rows VS 3 columns*
ARM Data Set Preparation	Depth of mining is an issue. There is a need to filter irrelevant columns (attributes)	No actual attributes in the raw form. There is a need to prepare a list of attributes and estimate its values (bins).
Attribute Breakdown	Each column is an attribute, and 2 class labels	The number of attributes depends on the number of bins in each column, and 2 class labels
Attribute Value Type	Presence / absence	Tier-based (always present)
Origin of Attribute Values	Included in the raw data set	Estimated
Goal of ARM	To find clusters of genome variants that are significant to each class labels, individually and collectively.	To find bins of climatic variables that are significant to each class labels, individually and collectively.

2.4. Association Rule Generation

The association rules were generated with relative support for each of the class labels, as the transactions in both the data sets were not equally distributed among the class labels. Single (1 *lhs* item) rule as well as maximal (2 *lhs* item or more) rule was generated for each data set. The single rules find the single items that are "interesting" for the class label. The maximal rules find the rules

of larger length and help to minimize redundant rules. We used the *apriori* algorithm implemented in the *arules* R package [14].

In the case of single rules for damselflies data set, however, due to the discretization process, the number of classes for different attributes were different. Attributes having less number of classes would result in higher support, and vice-versa. To normalize, we used cutoff support for both the climatic variables and the class labels as well. The first step, we devised the cutoff support for each of the climatic variables by considering the number of classes for that climatic variable. The second step, we used relative support for each class label.

In the case of analysis of the maximal rules, the interesting single items from single rules would get identified through visual representation of the nodes and graph elements, using colors and sizes. For the genome data set, we also introduced new measures of a rule called "interestingness" and "interesting size" that counts the number of interesting clusters and the sum of the items in each of the associated interesting clusters respectively.

2.5. Association Rule Representation

Visualization has a long history of making large data sets better accessible using techniques like selecting and zooming [15]. This is true for association rules as well. Association rules results in enormous number of rules, making it difficult to discover interesting ones. We extended existing techniques to represent such association rules in the form of graphs. Similar graph representation has been used in [15]. We used *visNetwork* R package [16] for creating the graph networks.

The octahedron node denotes the class label (*rhs* of a rule). The triangle node denotes a rule in the association graph. The size of the triangle denotes the support of the rule. The edges from triangle lead to a class label. The weight of the edges denotes the confidence of the rule.

The circle node denotes an item (*lhs* of a rule). The nodes with color like a class label node color denotes the "*interesting*" cluster for that class label. The node with blue color denotes a regular

item that wasn't found interesting for either of the class label. The label of the nodes contains the name of an item or a class label. The edges from the node lead to a rule (triangle).

The size of a circle node (excluding class labels) depends upon the following factors, namely

- Support with the class label (*supp*): From the single rules, the support of the rule featuring the given item with class label.
- Confidence with the class label (*conf*): From the single rules, the confidence of the rule featuring the given item with class label.
- Data set factor (*df*): Data set factor is peculiar in each data set, and we identified elements in both data set that should play a role in determining the size of the node.
    - In the genome study, we identified *multiplicity* (resulting from aggregation) as a crucial factor. Size of node *n* for class label *p* is

$$S(n) = supp(n \rightarrow p) * conf(n \rightarrow p) * log(log(multiplicity(n)))$$

      In case of aggregated graph, support for items in both class label was normalized.
    - In the damselfly study, we identified the *class-size* (resulting from discretization) as a crucial factor. Size of node *n* for class label *p* is:

$$S(n) = supp(n \rightarrow p) * conf(n \rightarrow p) * log(class\_size(n))$$

    - In genome study, log-log scale is used due to high range of values (1-400), where as in damselfly study, log-scale is used due to low range of values (2-9).
- Scaling factor (*sf*): *sf* is chosen experimentally, and fixed for each data set. This is only used to scale the network node elements.

3. GENOME STUDY

From a computational perspective, we treat the problem as consisting of data that only represent presence, and we neglect the absence from Boolean attribute values. This is primarily due to the kind of relationship we are interested to uncover, i.e., items are significant for a class label, but also because the density of the data set is quite low (<0.2). Association rule mining allows using absence data explicitly, but the use of presence data is more common. The data set had 22020 columns (or items), 131 rows (or transactions) and two class labels that identified the source of the transaction. The enormous number of items pose issues for association mining. Such problems have been discussed in Section 2.3, and includes depth, complexity and insignificance. The pipeline in Figure 2-2 shows the proposed solution to deal with the underlying issues. The aggregation process helps to merge the items with same occurrence in all transactions as clusters. Then, the highly significant clusters, from statistical test with the class labels, are identified and chosen for ARM. Graph-based network pruning along with loose criterion of statistical test further reduces the number of clusters used for ARM. At last, association specific pruning, i.e., availability of *minsup* are checked to finalize the list of clusters (items now) for ARM. Once the data for ARM is prepared, the two class labels are also included as items and the class labels items would only feature in the *rhs* of the rules.

The items that would be the most important to a class label would be the one with higher support, higher confidence and higher number of aggregated members associated with it. The high support will filter the high occurrence of the items for the class label. The high confidence will filter the items that occur more frequently with either of the class labels, and cancel out the cross-support items. The high aggregated members size will filter the larger collection of co-occurring columns.

From a micro-biological perspective, we had a genome data set that contained presence or absence (Boolean) data for 22020 genome variants in 131 genome samples of *E. coli* originating from

surface soil of either field or forest land cover. *E. coli* are bacteria found in foods, environment and intestine of animals, including humans. Generally, a harmless bacterium residing in the intestinal microflora in animals including man, it can sometimes cause fatal diseases in humans and other animals [17]. Based on the type of infections and pathogenic strains, *E. coli* are divided into various pathotypes, serotypes and phylotypes. Phylotype is a term often used in microbiology, and is specifically a phylogenetic classification scheme. Currently, there are four well recognized phylotypes for *E. coli* - A, B1, B2, D [18]. The given data belonged to phylotype D of *E. coli*. Phylotype D is the most genetically diverse phylotype, and is often used in experiments.

Out of the 22020 genome variants, 5593 genes were core genome variants, and remaining 16427 were accessory genome variants. Core genome variants result from pool of genes that are common for all the genomes of a species under observation [19]. Accessory genome variants result from genes that move in and out of genomes [20]. The data set had a density of 0.186, i.e., around 18% was presence indicator in the whole data set. We'll refer the origin of the samples as class labels for the data set. Out of 131 samples, 46 samples were from the forest class label and remaining 85 samples were from the field class label.

3.1. Motivation

The idea was to find the co-occurrence of these genome variants and inspect presence with the two class labels, "field" and "forest". This helps to find commonly co-occurring genome variant traits that leads to either or both class labels. We would end up with their association rules, and we filter rules that pass the minimum support and confidence threshold. The "interesting" rules would be further analyzed, and assessed with visualization tools to get meaningful interpretation of their occurrence trait.

We started off with all the genome variants for itemset mining. The regular frequent itemset mining ended up with too many results, around a million frequent itemset for *minsup* of 0.1. This was

not particularly useful. We had to devise a way to decrease the number of genome variant that we use for itemset mining and ultimately use them for finding the association rules. We applied different methodology to reduce the number of genome variants to a sizeable quantity including aggregation, graph pruning, weight consideration and test of independence. Then, we used the filtered set for finding the association rules.

3.2. Pre-Processing

3.2.1. Aggregation

The idea behind aggregation was to find the set of genome variants that had the same occurrence trait across all the samples (genomes). The genome variant having such same occurrence could essentially form a cluster, and hence could be treated as a single entity. After aggregation, the number of genome variants reduced from 22020 to 7580 genome variant clusters. XOR logic was used to detect such same occurrence pattern, by counting number of dissimilarities across all the samples.

These 7580 genome variant clusters had unique occurrence traits. Out of these, 6431 were singly-occurring genome variants cluster and 1149 were multi-occurring genome variant clusters. Singly-occurring genome variant clusters have only one genome variant as a member and that variant has a unique occurrence trait, while the multi-occurring genome variant clusters have two or more genome variants sharing the same occurrence trait.

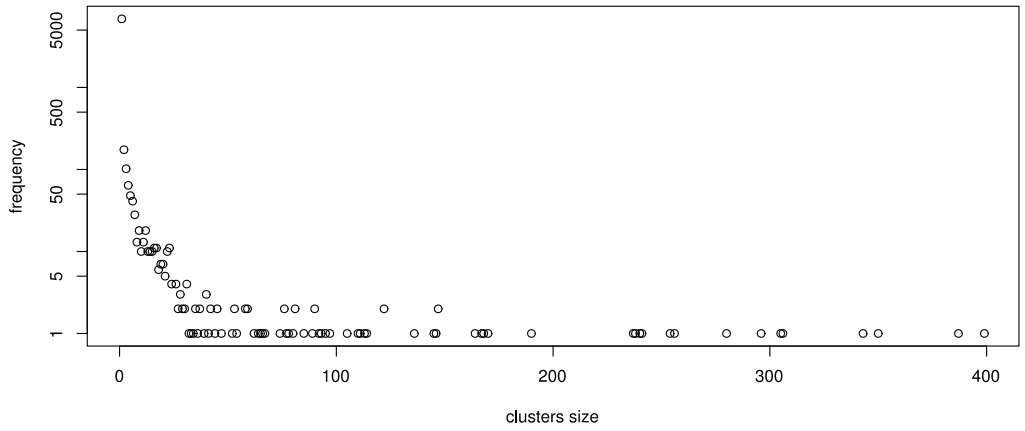

Figure 3-1: Cluster size (multiplicity) distribution

Figure 3-1 shows that the aggregated size of genome variants (cluster size) ranged from 1 to 400. Each of these genome variant clusters represent a totally unique occurrence trait in the data set. Throughout the rest of the paper, we'll represent the cluster size, i.e., the number of genome variants with same occurrence traits, with "*multiplicity*". We'll represent the cluster with the first genome variant that appears in the cluster, and a number within braces will represent its multiplicity.

The next steps include further reduction of the number of genome variant clusters used in the association analysis. A refined set of clusters are chosen from these uniquely occurring genome variant clusters. There are two criterions for such selection, with the necessary terminologies and explanations are presented below:

- Strict Criterion (Highly Significant): These clusters exhibit a very high significance in their occurrence trait with the class labels. This is a stricter criterion for cluster selection. We consider a cluster to be "highly significant" if:
    - it has a graph weight of 13 or more
    - it passes the significance test at alpha level of 0.01
    - it has minimum support used for association
    - it has at least 70% confidence with either of the class labels

- Loose Criterion (Significant): These clusters exhibit significance in their occurrence trait with the class labels. This is a relatively loose criterion for cluster selection. We consider a cluster to be "significant" if:
    - it passes the graph pruning
    - it has a graph weight of 3 or more
    - it passes the significance test at alpha level of 0.05
    - it has minimum support used for association

Once, both the lists are calculated, the union of the set of the genome variant clusters is taken as the final set of clusters for association analysis.

3.2.2. Weighted Graph Network

Once aggregation was done, to further decrease the number of genome variants clusters under analysis, we used a "*weighted graph*" approach. Graph-based approaches have been used extensively for bioinformatics problems as reported in [10] [21] for protein interaction network with several standardized databases offering vast pool of information. However, we used a graph-based approach with local occurrence information from our data set. This required us to find genome variant clusters that were *similar* in their occurrence trait. We defined such similarity property as, "*two genome variant clusters are similar if they have only one difference in their occurrence trait across all samples*". We processed each of the genome variant clusters to find such single difference in their occurrence trait across all the samples. This forms the neighborhood criteria for each of the clusters. Thus, a set of clusters *similar* to a cluster 'C' is the set of neighbors for C. Throughout the rest of the paper, we'll represent the number of neighbors of a cluster with "*neighborhood size*".

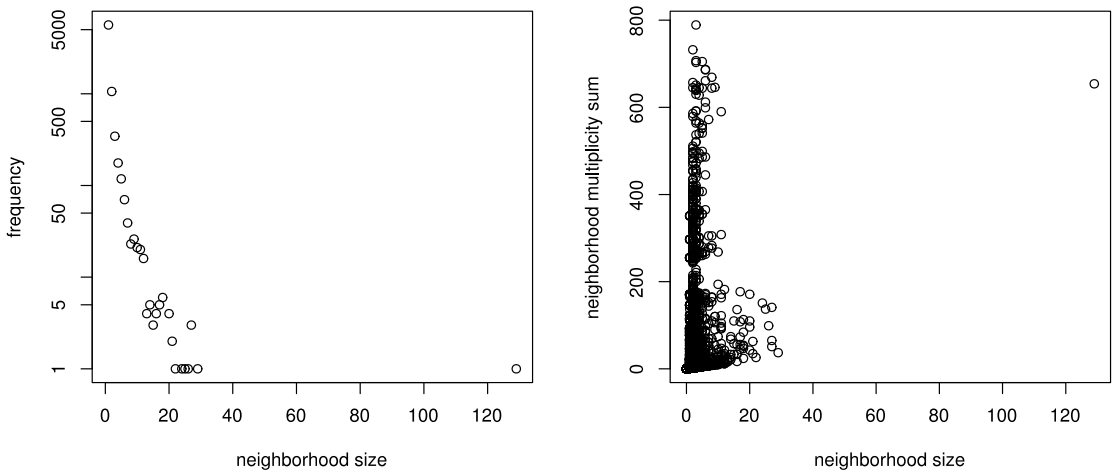

Figure 3-2: i.(left) Neighborhood size distribution. ii.(right) Relation of neighborhood size with neighborhood multiplicity sum

Figure 3-2 (i) shows the neighborhood size distribution. The frequency is shown in logarithmic scale. The size of the neighborhood ranged from 0 to 129, second largest being 29. There were 5768 clusters with no neighbors and 1162 with a single neighbor.

Figure 3-2 (ii) shows the multiplicity summation of neighborhood with respect to the size of the neighborhood. The figure shows that there were a few clusters with very low number of neighbors yet having very high sum of multiplicity of neighbors.

3.2.2.1. Weight Calculation

We calculate the "*weight*" of all the clusters, and each node represents a genome variant cluster in the graph. One basic problem of unweighted graphs is that they cannot indicate the reliability and strength of the nodes [10]. We use *weighted node* to address this issue. We'll use node and genome variant cluster interchangeably. The weight of a node depends on the multiplicity of the cluster and its neighbors. A cluster tends to get more weight if they have a higher multiplicity and a higher neighborhood size, with its neighbors also having higher multiplicity. The weight, Wc, of a node/cluster (c) having multiplicity ($M(c)$), and having neighborhood size (Nc), where N^1_c, N^2_c, …, N^n_c are the neighbors of node, is given as:

$$Wc = \sum_{i=1}^{n} M(c) * M\left(N_c^i\right), if\ Nc\ not\ empty$$

$$Wc = M(c), if\ Nc\ is\ empty$$

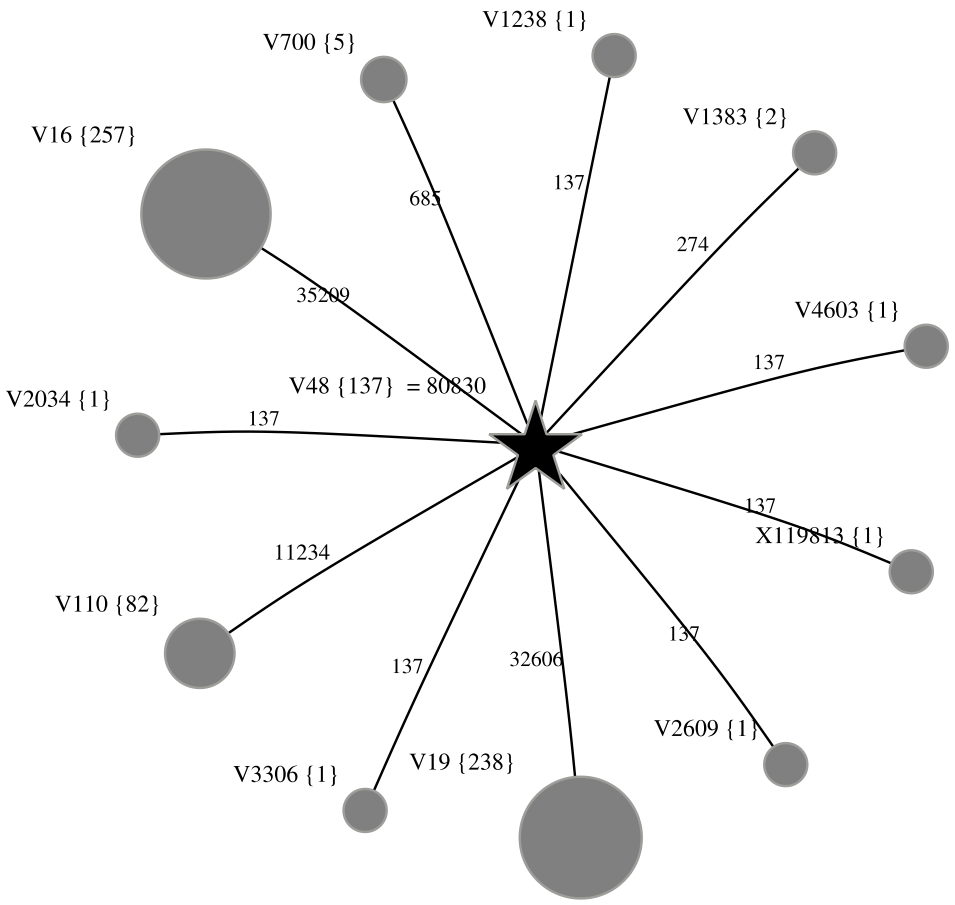

Figure 3-3: Weight calculation for a cluster "V48"

In Figure 3-3, we can see a graph for a cluster "V48". The graph is of order 1, i.e., a cluster and its immediate neighbors. To calculate the weight of "V48", we need multiplicity information of "V48" and all its neighbors. The node labels contain the cluster name with its multiplicity count, 137 in case of "V48". The edge labels denote the contribution of the neighbor to the weight of "V48". The main cluster "V48" has a derived weight of 80830. This is the highest weight, and the traversal of nodes starts from "V48" in our pruning approach. If a node does not have any neighbor, then its multiplicity value is regarded as its multiplicity.

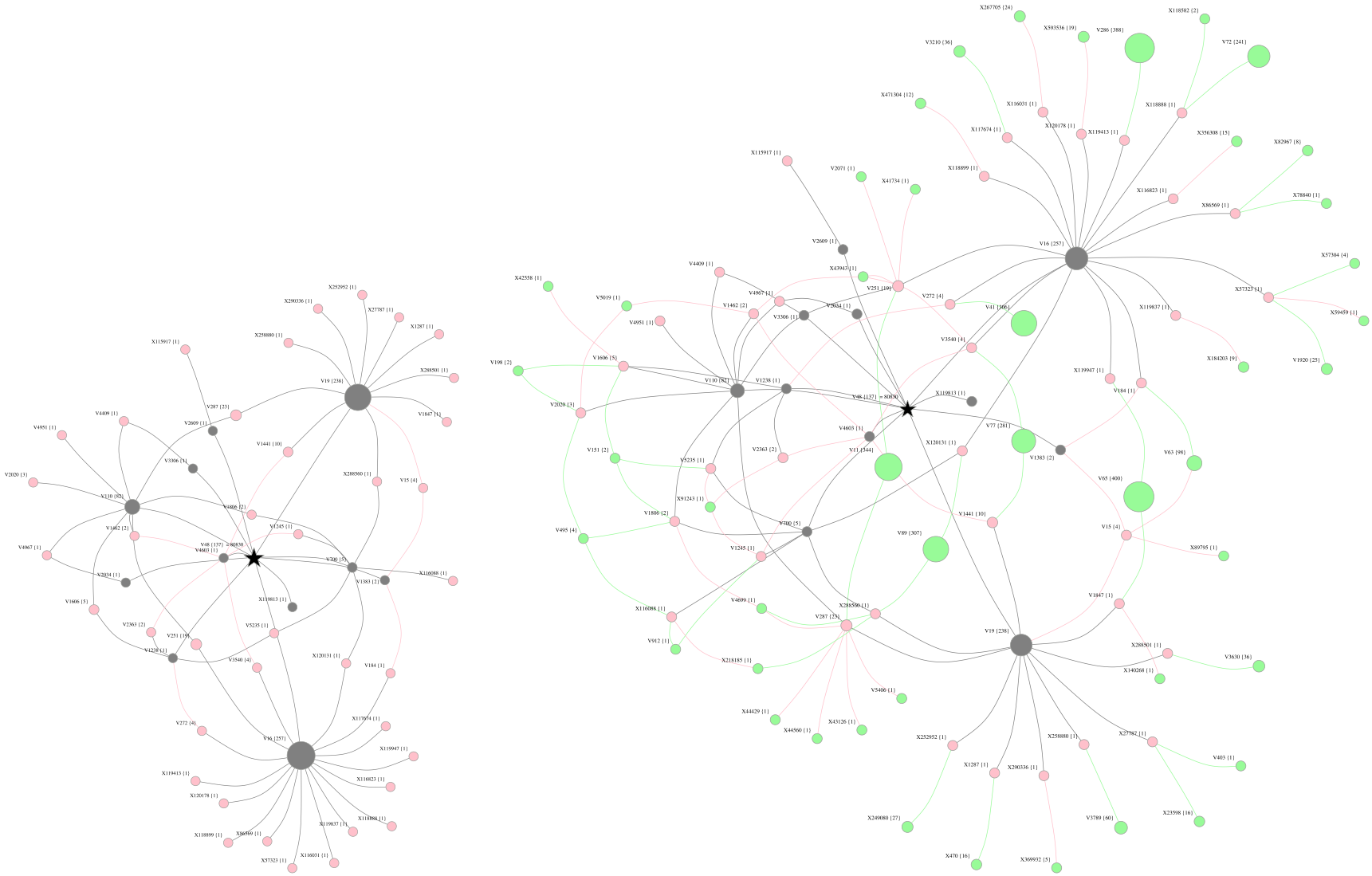

Figure 3-4: i. (left) Network of order 2 for cluster “V48” ii. (right) Network of order 3 for cluster “V48”

Based on the clusters and their neighborhood, we create an undirected network graph. The Figure 3-4 (i) shows a graph of order 2 for the cluster V48, i.e., depth of 2 and Figure 3-4 (ii) shows a graph of order 3, i.e., depth of 3. In Figure 3-4 (ii), the main node (star) is "V48", and nodes with color grey, pink and green represent nodes at depth 1,2 and 3 respectively. We assign the calculated weights to each node.

In the case of stricter criterion for cluster selection, the weights are only used as cutoff to select the nodes (genome variant clusters) with higher weight (12). However, in the case of looser criterion, the weights are used as cutoff (weight of 3) as well as in the pruning process.

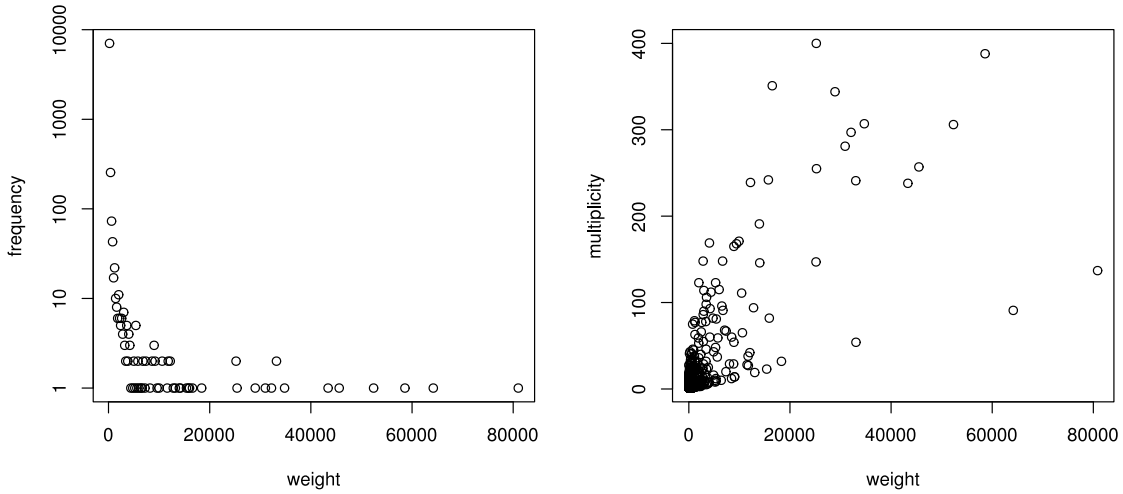

Figure 3-5: i. (left) Cluster weight distribution ii. (right) Relation of cluster weight with multiplicity

Figure 3-5 (i) shows the weight distribution among all the clusters. The frequency is shown in logarithmic scale. A total of 4717 clusters were isolated, clusters with weight of 1. The weight of 1 means that it's multiplicity was 1 and it had no neighbors. Previously, Figure 3-1 shows many singly-occurring clusters, and Figure 3-2 (i) shows many clusters with no neighbors. This explains the reason for such low weight across many clusters.

Figure 3-5 (ii) shows the weight of the cluster with respect to its multiplicity. The graph suggests that as the multiplicity increases the weight of the cluster also increases. The highest weight,

however, is from cluster "V48" which has multiplicity of 137 with a neighborhood of 11. This can be attributed to the fact that its neighbors have relatively larger multiplicity as shown in Figure 3-3.

3.2.2.2. Graph Pruning

In the case of loose criterion for cluster selection, we perform graph pruning approach. We create a tabulated list, say L_{weight}, of all the nodes with weights in descending order. This provides the order of traversal preference on the network graph. We also maintain a pruned list L_{pruned}, to keep track of the pruned nodes. We start from the node at the top of L_{weight}, we prune all its neighbors, and push the neighbors to L_{pruned}. Then, we go to the second node of L_{weight}. If the second node is not in L_{pruned}, we prune all its neighbors, otherwise we move to the next node in L_{weight}. We continue this until we traverse all the nodes.

Algorithm Graph Pruning

```
1: procedure PRUNENODES()
2:     graph ← network with clusters as weighted nodes
3:     Lweight ← clusters with weights sorted in descending order
4:     Lpruned ← list that holds pruned items, empty initially
5:     for each cluster c in Lweight do
6:         if c not in Lpruned then
7:             neighbors ← list of neighbors of c
8:             graph ← update graph by pruning neighbors
9:             Lpruned ← add neighbors to pruned
10:        else
11:            next
12:        end if
13:    end for
14:    Return graph
15: end procedure
```

The idea is to generate diversely occurring genome variant traits to be used in the ARM. If a genome variant cluster feature in the rules generated from ARM, then the clusters in its neighborhood could also be significant, and thus, further analyzed. After graph pruning phase, we had 5756 clusters. These clusters are sent through the loose criterion of cluster selection. Note that some highly significant clusters might also feature in the list.

1825 clusters were pruned. This pruned list might contain some highly significant clusters as well. Hence, the stricter criterion of cluster selection helps to retain these highly significant clusters that might have otherwise been pruned. The whole weighted graph network before the pruning is used for the stricter criterion.

3.2.2.3. Removal of Low Weight Clusters

The weight was used as a cutoff to filter clusters for the association analysis. The stricter criterion had a cutoff weight of 13 or more for all the clusters. Similarly, the looser criterion had a cutoff weight of 3 or more for all the unpruned clusters. The use of such weight clusters ensures that we select clusters that have higher co-occurrence in the form of multiplicity and higher similarity with other clusters in the form of neighborhood.

In case of stricter criterion, we had 1753 number of genome variant clusters having weight of 13 or more. These clusters would be further refined through test of independence, minimum support check and confidence check with the class labels.

In case of looser criterion, we had 5756 number of genome variant clusters after the pruning process. As we can see from the Figure 3-5 (i), there are a lot of genome variant clusters having significantly low weight value. In fact, there are 4737 genome variant clusters having weight less than 3. These clusters have low multiplicity as well as low neighborhood size, and can be regarded as less significant. Once these clusters having weight less than 3 were ignored, 1019 clusters prevailed. These clusters would be further refined through test of independence and minimum support check.

3.2.3. Test of Independence

As discussed before, there were two class labels, "field" and "forest". The idea was to find the genome variant clusters that were statistically significant with the class labels, and then find associations between these significant clusters. This would remove the genome variants that were seen to be statistically randomly occurring from further inspection. We added a column that

represented the class labels to the data set. This was based on whether the sample source was from forest or field. Forest was assigned as '1' and field was assigned as '0'. Even if the assigned values for the class labels are reversed, the result from the independence test would remain the same.

We started off with Chi-square Test of Independence. We tested each cluster for independence with the class label column. The null hypothesis was that the cluster and the class label were independent of one another. The alternative hypothesis was that they are related. In the test of independence, if we find p-value from the chi-squared test to be less than the chosen alpha value, 0.01 for stricter and 0.05 for loose criterion, then we can reject the null hypothesis. The clusters, then, can be regarded to be statistically related to the class label. We will call them "statistically significant" clusters. However, in some clusters, the expected frequency in the contingency table was found to be less than 5. It has been widely reported that the approximation from chi-squared test worsens with such small frequencies. Yates created a correction method for such shortcomings, however we could use some other exact methods. One of the method that overcomes the problem is Fisher's Exact Test of Independence [22].

The statistical independence of all the genome variant clusters was tested against the class label column with Fisher's Exact Test with a significance level of $\alpha = 0.01$ for stricter and $\alpha = 0.05$ for looser criterion. We ended up with 43 "highly significant" and 110 "significant" clusters whose occurrence were "statistically significant" with the class labels.

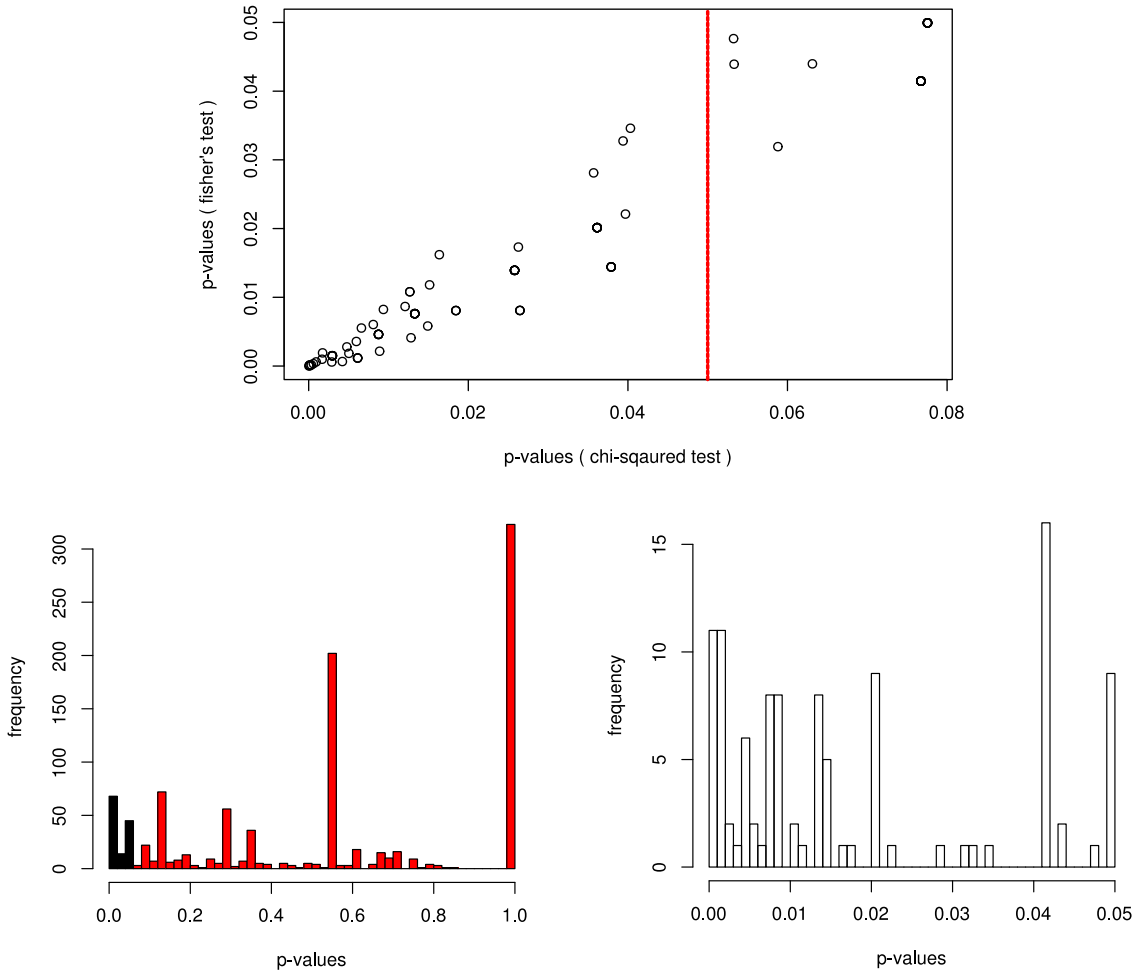

Figure 3-6: i. (top) Comparison between Fisher's test and Chi-squared test ii. (bottom-left) Distribution of p-values of all clusters iii. (bottom-right) Distribution of p-values of significant clusters

Figure 3-6 (i) shows the difference in p-values of the clusters from significance test from chi-squared test with Yates correction and fisher's exact test. There was a total of 110 clusters that were "statistically significant" with significance level of $\alpha = 0.05$. If we had used chi-squared test with Yates Correction instead, 29 of these of clusters would have failed the significance test. The points to the right of the red line represent these clusters.

3.2.4. Association Specific Pruning

In the case of stricter criterion, we used support-based and confidence-based pruning. For support-based pruning, the clusters need to have at least minimum support of 0.05 (we later use this

minsup - 7 out of 131) to feature in the association rules. All the 43 clusters had the support more than 0.05 so no clusters were pruned. For confidence-based pruning, the clusters need to have at least a confidence of 70% with either of the class labels. 12 clusters lacked such confidence with either of the class labels, and were pruned. There was a total of 31 highly significant clusters used in the association analysis.

In the case of looser criterion, we used support-based pruning. Out of the 110 significant clusters, only 67 clusters had the *minsup* of 0.05. This cluster list had 9 highly significant clusters, so there were 58 significant clusters used in the association analysis.

Thus, 31 clusters from stricter criterion and 58 clusters from looser criterion were picked; making a total of 89 clusters in the association study.

3.3. Association Rules

The significant genes, a total of 89 genome variant clusters, along with the class labels are converted to basket-like transactions. The transactions are represented using efficient data structure within "arules" R package [14].

The idea with ARM was to find individual clusters and, a relatively longer list of co-occurring clusters that led to either of the class label. As discussed before, the clusters themselves could have a list of associated genome variants (one or more). The use of such clusters helps to decrease the computational problem size at hand. It helps us to decrease the depth of mining for associated itemsets.

3.3.1. High Support Items

The first step was to find the highly occurring genome variants clusters (1-item frequent itemsets) irrespective of the class labels. These clusters have a relatively higher number of occurrences in the sampled database. Usually, these are genome variant clusters that are seen throughout the available samples. However, there is a possibility that we could see some genome

cluster that could be more associated with a particular class label. Such peculiarity can be further assessed using "single" association rules.

Support of 0.3 was used to find the highly frequent itemsets. Figure 3-7 shows 13 such clusters and includes the two class labels, "field" and "forest". The class label's support denotes the percentage of the origin of the samples. The field has support of 0.6488550, i.e., 85 out of 131 samples belonged to field. The remaining 46 samples belonged to forest. As seen from the figure, some of the genome variants clusters are very common, having support more than 0.65, i.e., it appears in at least 85 samples. These clusters mostly form *cross-support patterns* [10] [17]. These patterns could associate with many small associated clusters due to its high support. Quality measures such as "*h-confidence*" has been proposed to remove such cross-support items, and find hyper clique patterns [11].

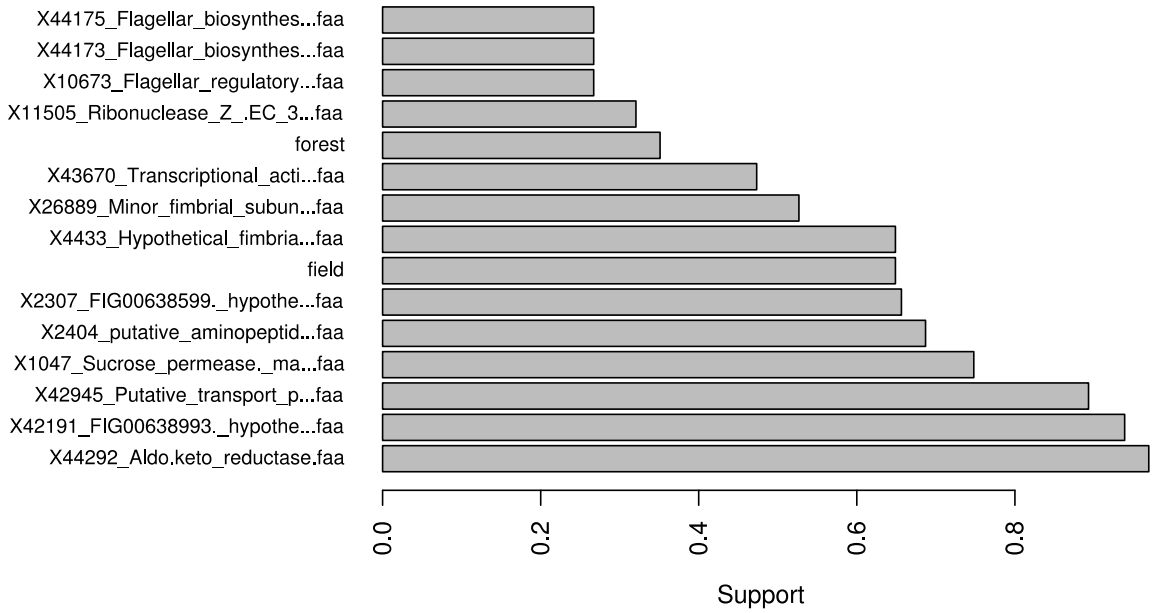

Figure 3-7: Clusters with high support along with class labels

3.3.2. Single Rules

The second step was to find the individual genome variants clusters that were closely related to either of the class label. This means finding association rules that had only 1 item in the *lhs* and

one of the class label in *rhs* of a rule. This allows us to find the high support individual clusters for each of the class label. Once we filter the association rules with higher confidence, it reflects that those genome variant clusters occur more likely in either field or forest. In case of longer rules, if some of these peculiar genome variant clusters occur in the rules then we can say that these clusters are more "*interesting*" for that class label.

We were looking for association rules in pattern of

$$[A\ Genome\ Variant\ Cluster] \rightarrow [Surface\ soil\ source]$$

$$[A\ Cluster] \rightarrow [forest]\ OR\ [field]$$

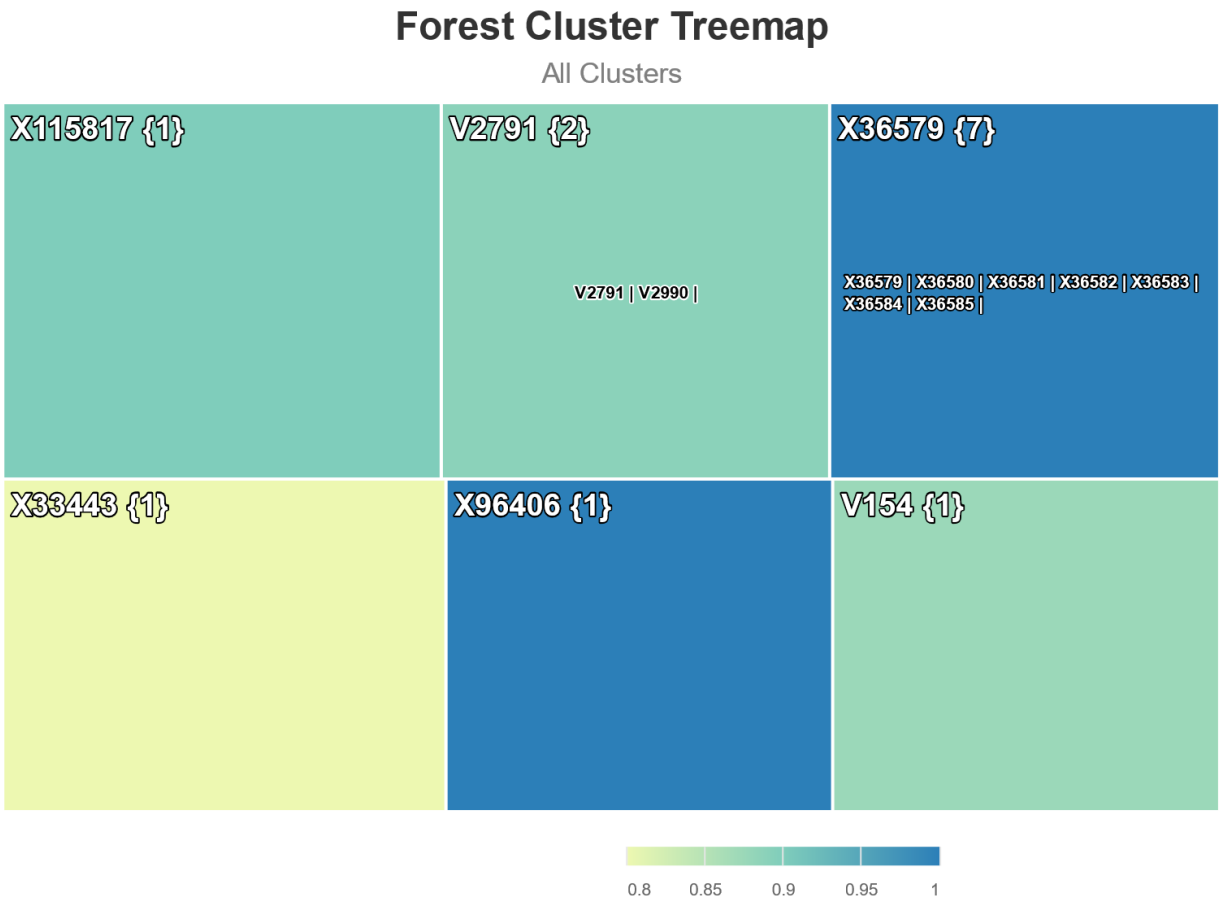

Figure 3-8: Treemap showing "interesting" forest clusters with members

As shown in Figure 3-8, there were six genome variant clusters for forest class label that passed the threshold conditions of 0.05 support and 0.8 confidence. This set of six clusters are the

"interesting" clusters for the forest class label. The size represents the support and the color represent the confidence of these clusters for the forest class label. We will consider longer association rules that will feature these clusters as the interesting rules. Figure 3-8 also shows all the genome variants (members) associated to each cluster. There was a total of 13 individual genome variants in these 6 clusters.

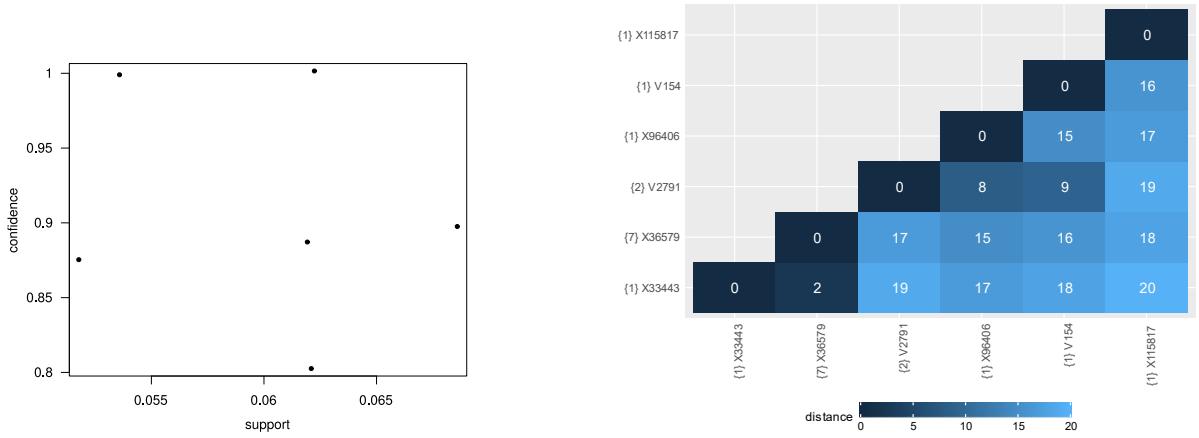

Figure 3-9: i. (left) Support-Confidence plot for forest (single rules) ii. (right) Heatmap showing occurrence trait distances among interesting forest clusters

The support ranged from 0.053 to 0.069. The confidence ranged from 0.8 to 1. The highest support was 0.069 for [X115817] with a confidence of 0.9. The highest confidence was 1 for [X96406] and [X36579, X36580, X36581, X36582, X36583, X36584, X36585] with support of 0.061 and 0.53 respectively.

To understand the occurrence trait of the selected significant clusters for the forest class label, we make use of the heatmap shown in Figure 3-9 that shows occurrence distance between these set of clusters. *"Two clusters are said to have occurrence distance of x if there are x differences between their occurrence across all the available samples"*. The ordering of the clusters is based on the occurrence distance whereby clusters having smaller distances are grouped together. This shows the variety among occurrence trait of clusters. The occurrence distance ranged from 2 to 20. Occurrence distance for a cluster to itself would be 0. Figure 3-9 shows only one such distinctly similar clusters,

[X33443] and [X36579] with an occurrence distance of 2 across all samples. This cluster group could be assessed as being similar in their occurrence in the sampled data. This means that eventually when the larger association rules are generated, they could likely feature together.

Table 3-1: All "interesting" forest clusters

lhs	cluster size	rhs	support	confidence
[X96406_Uncharacterized_prot...faa]	1	[forest]	0.0534351	1.0000000
[V154]	1	[forest]	0.0534351	0.8750000
[V2791]	2	[forest]	0.0610687	0.8888889
[X36579_hypothetical_protein.faa]	7	[forest]	0.0610687	1.0000000
[X33443_CRISPR.associated_he...faa]	1	[forest]	0.0610687	0.8000000
[X115817_hypothetical_protein.faa]	1	[forest]	0.0687022	0.9000000

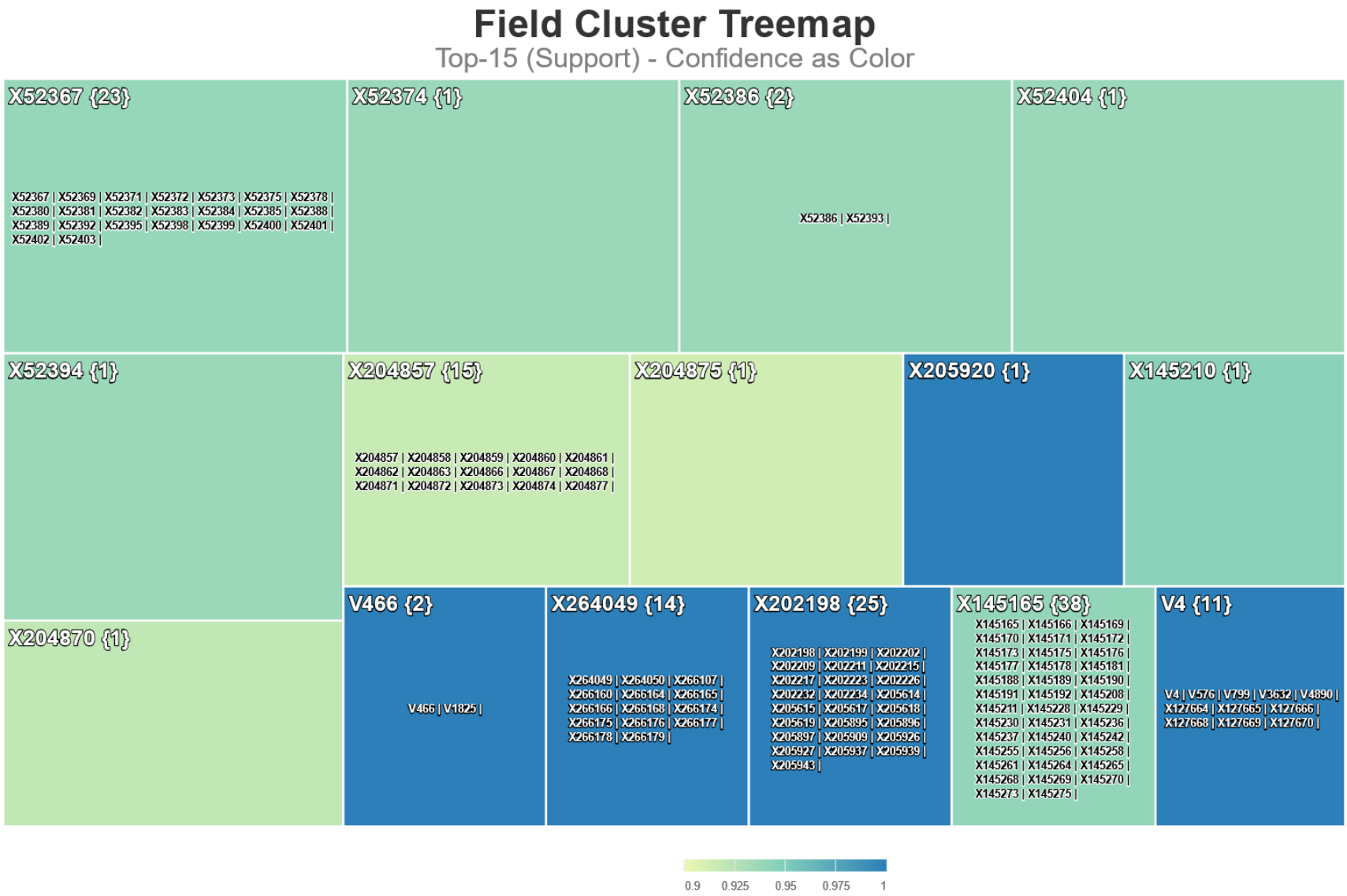

Figure 3-10: Treemap showing “interesting” field clusters with members.

Figure 3-10 shows the top-15 clusters for the field class label that passed the threshold conditions of 0.05 support and 0.8 confidence. The top-15 clusters were chosen as the top 30 clusters based on support. The top-29 cluster out of 51 clusters had support of more than 0.993 (relative support for the field class label).

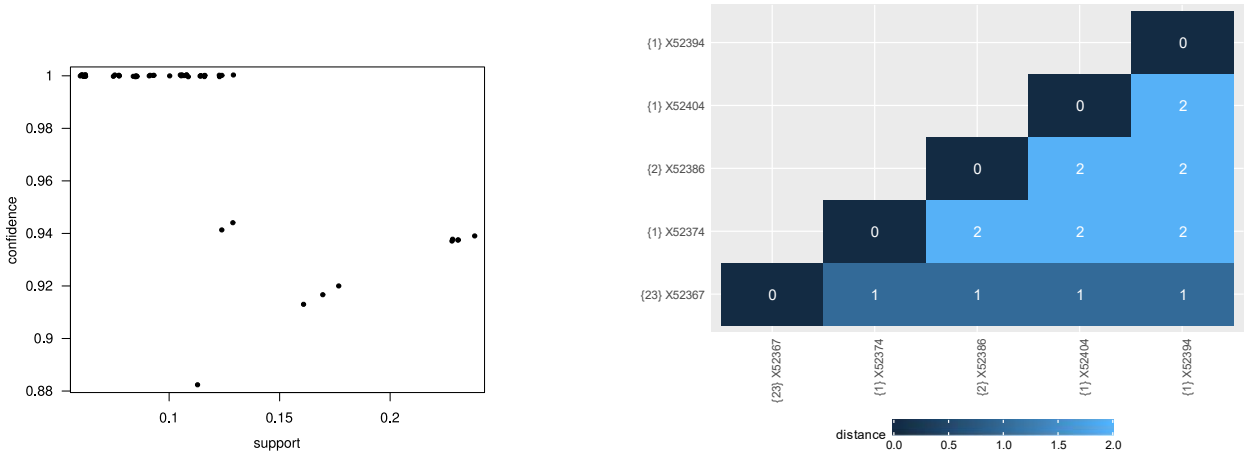

Figure 3-11: i. (left) Support-Confidence plot for field (single rules) ii. (right) Heatmap showing occurrence trait distances among top-5 interesting field clusters

Table 3-2: Top-5 "interesting" field clusters based on support

lhs	cluster size	rhs	support	confidence
[X52367_Regulatory_protein_C...faa]	23	[field]	0.23664122	0.9393939
[X52374_FIG01046174._hypothe...faa]	1	[field]	0.22900763	0.9375000
[X52386_Prophage_lysozyme_.E...faa]	2	[field]	0.22900763	0.9375000
[X52404_Gene_D_protein.faa]	1	[field]	0.22900763	0.9375000
[X52394_Baseplate_assembly_p...faa]	1	[field]	0.22900763	0.9375000

The Figure 3-11(i) shows support-confidence plot for the list of rules for field class label. The points are jittered to show the density of points sharing same support and confidence. There was a total of 51 clusters found for the field class label. The support range was from 0.061 to 0.237. A total of 5 clusters had support of greater than 0.2, highest being 0.237 for [X52367] {23}. The

other clusters crossing 0.2 support were [X52374], [X52386] {2}, [X52404] and [X52394]. Out of the 51 clusters, 40 of the clusters had a confidence of 1. The lowest confidence was 0.88. There was a total of 378 genome variants among these 51 clusters. Figure 3-11 (ii) shows the occurrence distance among these top-5 clusters. We can see that they were in a neighborhood of 1-2. Table 3-2 contains quality measures and cluster size for these top-5 clusters.

Figure 3-12 shows the occurrence trait distances among the 51 "interesting" genome variant clusters for the field class label. The ordering of the clusters has been co-aligned to show similarly occurring clusters together. This shows the variety among occurrence trait of clusters that lead to field class label. The range of occurrence distance was from 1 to 50. There are 5 such distinct similar cluster groups where there are at least 5 clusters. The darker cells near diagonal in the figure shows these closely-related clusters.

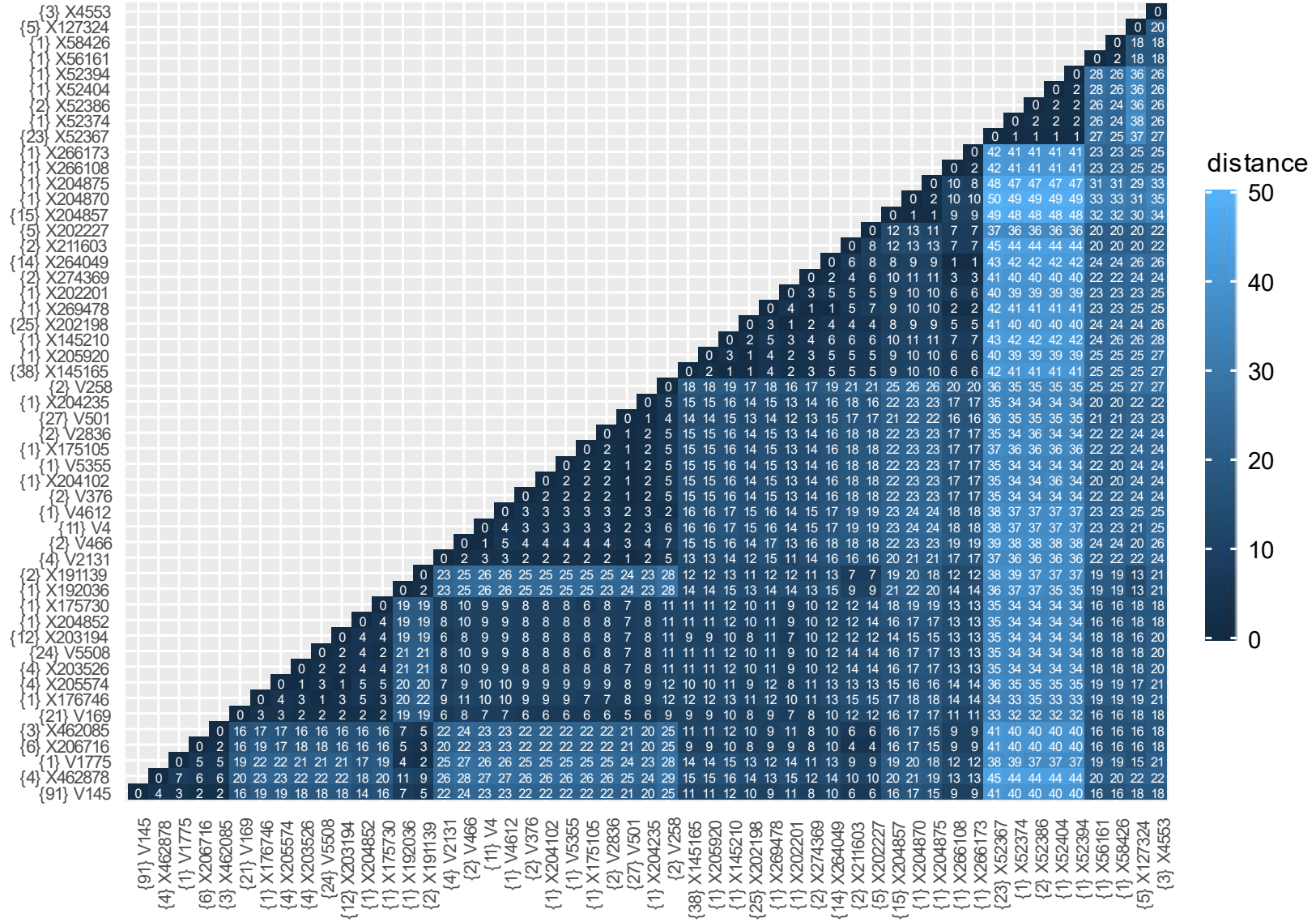

Figure 3-12: Heatmap showing occurrence trait distance among all "interesting" field clusters

3.3.3. Longer Rules

As per our goal of finding longer associations between occurrence traits, we used maximal itemsets to find the association rules. This helped us in avoiding a lot of redundant rules. Since we had a reduced number of clusters, which were lot diverse after aggregation and graph pruning, the depth was not an issue. The result went up to 9 and 11 itemset depth for forest and field respectively. The *minsup* and *minconf* were set at 0.05 and 0.8 respectively. All rules were generated first and then the maximal rules were induced from these rules.

We were looking for association rules in pattern of

$$[LHS\ items] \rightarrow [Surface\ soil\ source]$$

$$[List\ of\ clusters] \rightarrow [forest] OR, [field]$$

We ended up with 122 rules, 18 rules - forest and 104 – field, for a threshold criterion of 0.05 *minsup* and 0.8 *minconf*. The uneven distribution of number of samples among the class labels led to field having large number of rules, as the individual support of field was 0.65 compared to 0.35 for forest. So, to maintain a relative support threshold for each class label, we increased the support threshold for field to 13 out of 131, ~ 0.099. This decreased the number of rules for field to 45 rules.

3.3.3.1. Rule Representation

The light green (octahedron) denotes the class label "forest" and light brown (octahedron) denotes class label "field". The grey (triangle) denotes a rule in the association graph. The size of the triangle denotes the support of the rule. The grey edges from rule edges lead to a class label. The weight of the edges denotes the confidence of the rule.

The blue node (circle) denotes a genome variant cluster. The green node (circle) denotes the "interesting" genome variant cluster for "forest", while the brown (circle) for "field" class label. The

label of the node contains the genome variant at the head of the cluster with the multiplicity count for that cluster. The edges from the node lead to a rule (triangle).

The size of a circle node depends upon 3 factors, namely

- Support with the class label (*supp*)

  From the single rules, the *support* of the rule featuring the given cluster.

- Confidence with the class label (*conf*)

  From the single rules, the *confidence* of the rule featuring the given cluster.

- Multiplicity (*mult*)

  The *multiplicity* of the cluster is used. In calculating the size of the node, logarithmic function ($log\ log$) is used for the multiplicity value to reduce the effect if the multiplicity is on the higher side. For nodes having multiplicity of less than 4, $log\ (log\ (3.4))$ is used to avoid very small nodes.

- In case of a single-class-label network, the size (S) of a node (n) in class label (p) is given by:

$$S(n) = supp(n \rightarrow p) * conf(n \rightarrow p) * log(log(mult(n)))$$

- In case of a multi-class-label network, if the node is a cross-support item then the size (S) of a node (n) in class label (p(1)) and class label (p(2)) is given by:

$$S(n) = \frac{\sum_{i=1}^{2}[supp(n \rightarrow p(i)) * conf(n \rightarrow p(i))]}{1.5} * log(log(mult(n)))$$

3.3.3.2. Forest Rules

The configuration for the rule mining for the forest class label is listed:

$$Support = 0.05\,;Confidence = 0.8\,;Min.Length = 5\,;Max.Length = 9$$

A total of 18 association rules was generated for this class label.

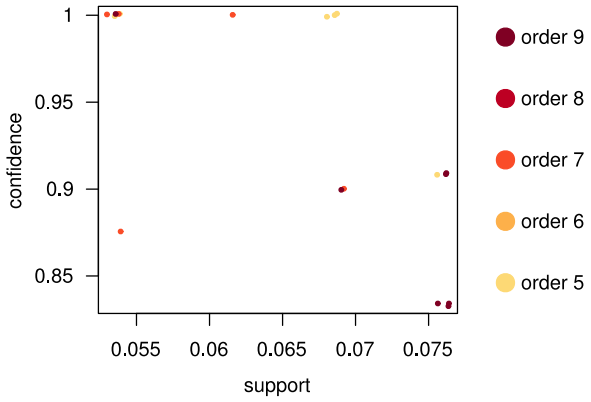

Figure 3-13: Support – Confidence – Order plot for maximal forest rules

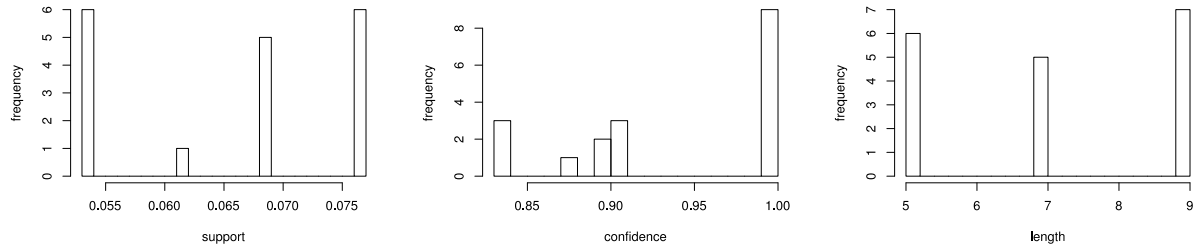

Figure 3-14: i. (left) Support ii. (middle) Confidence iii. (right) Length

The support, confidence and length quality of the rules for the "forest" class label is shown in the Figure 3-13 and Figure 3-14. The support range for the rules was 0.534 - 0.763. The confidence range was 0.87 to 1. The length of the rules ranged from 5 to 9.

Figure 3-15 shows all the rules for the forest class label. Out of the 18 rules, only 5 rules had one or more "interesting" clusters for the forest class label. Previously, we had found 6 clusters "interesting" for the forest class label in Figure 3-8. 4 of these 6 clusters featured in our association rules.

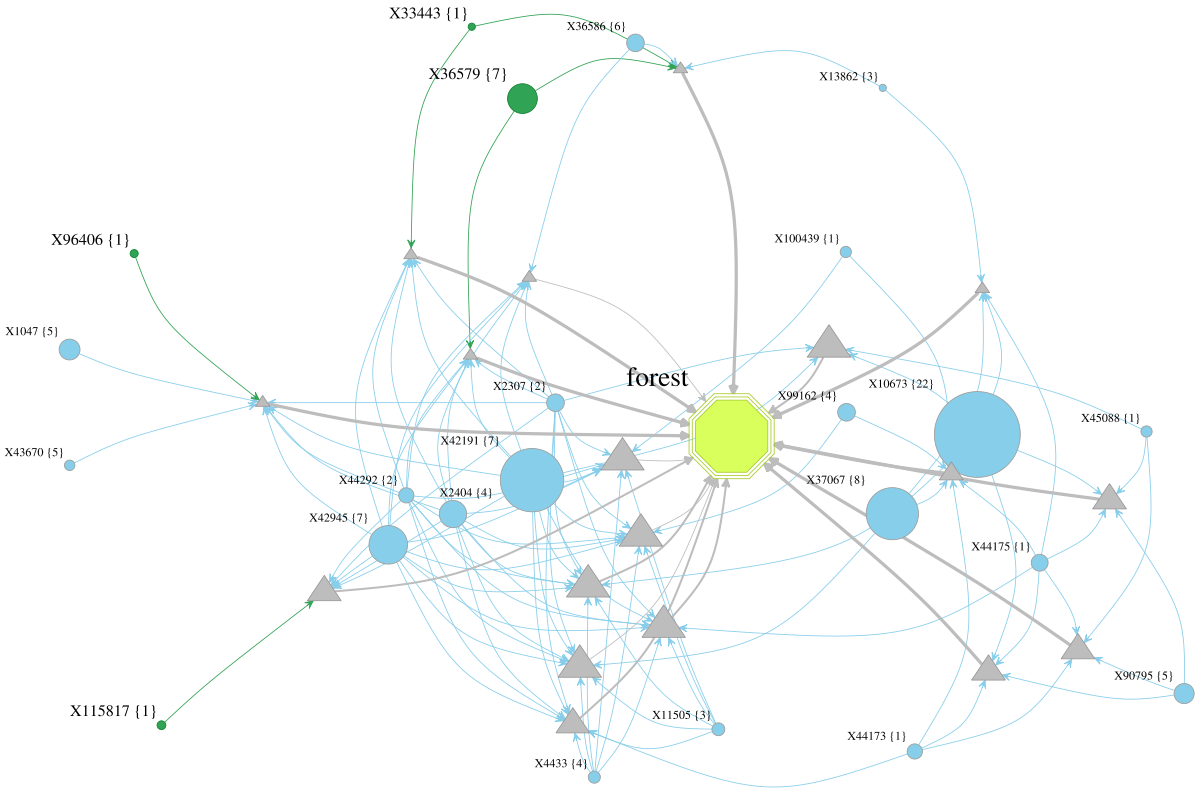

Figure 3-15: Network showing all maximal rules for forest

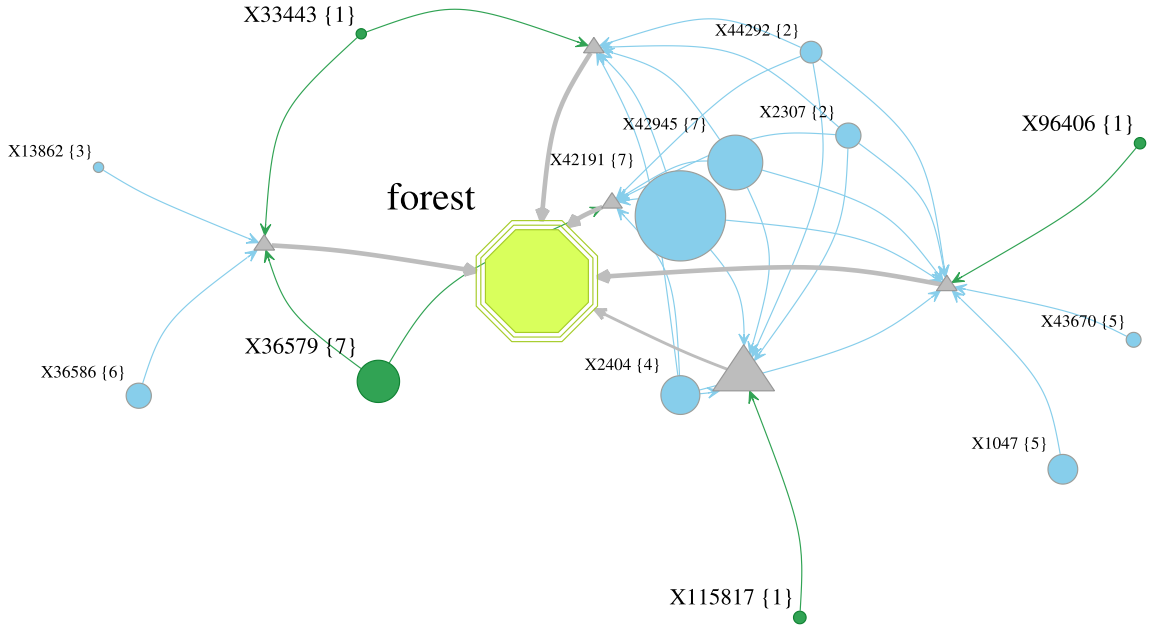

Figure 3-16: Network showing interesting maximal rules for forest

Figure 3-16 shows the rules involving interesting clusters for the forest class label. The most interesting rule is one involving [X33443] and [X36579], both clusters were present as "interesting" for forest. Even though the other clusters, [X36586] and [X13862], were not interesting for forest, they still had confidence of above 0.6 for forest. The other rules are mainly based on cross-support items, clusters that have high support throughout the samples. The lowest overall support among these set of clusters is 0.65, with maximum confidence of 0.43.

3.3.3.3. Field Rules

The configuration for the rule mining for the field class label is listed:

$$Support = 0.099\,;Confidence = 0.8\,;Min.Length = 5\,;Max.Length = 11$$

The total of 45 association rules was generated for this class label.

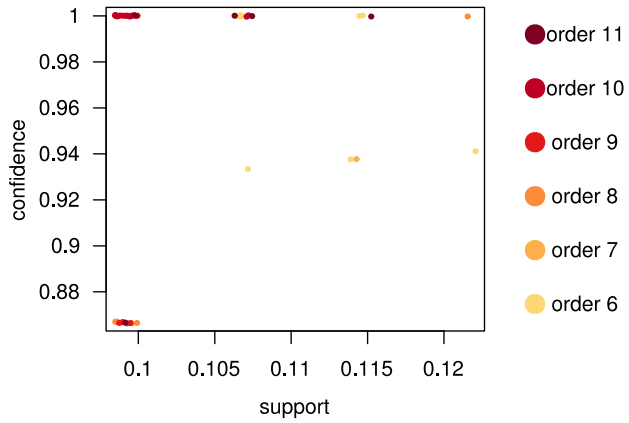

Figure 3-17: Support – Confidence – Order plot for maximal field rules

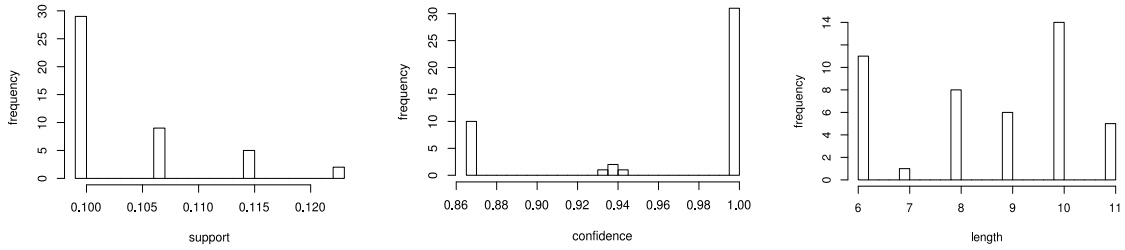

Figure 3-18: i. (left) Support ii. (middle) Confidence iii. (right) Length

The support, confidence and length quality of the rules for the "field" class label is shown in the Figure 3-17 and Figure 3-18. The support range for the rules was 0.0992 - 0.122. The confidence range was 0.867 to 1. The length of the rules ranged from 6 to 11. The points are jittered to show the density of points sharing same support and confidence in Figure 3-17.

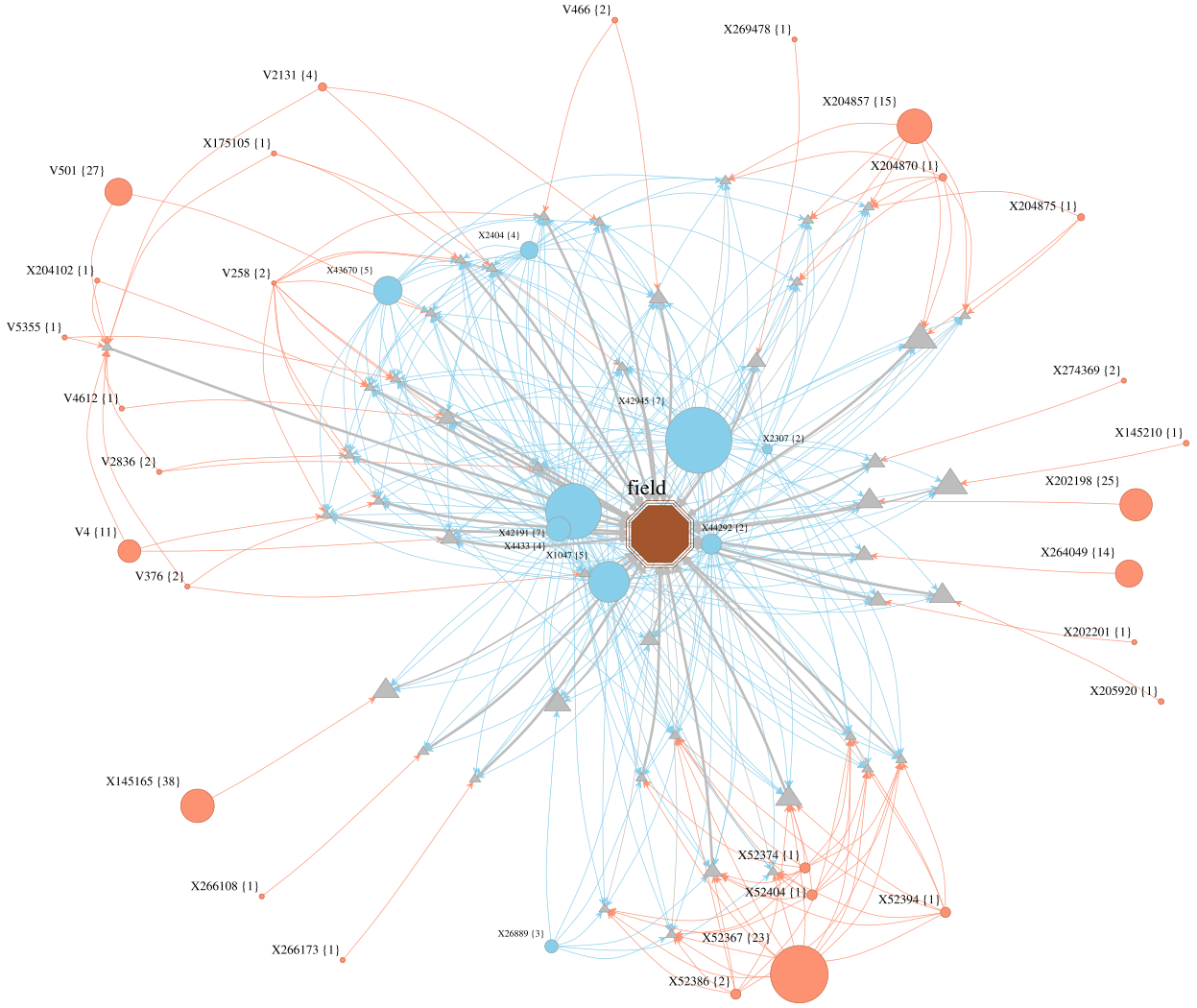

Figure 3-19: Network showing all maximal rules for field

Figure 3-19 shows all the rules for the field class label. Out of the 45 rules, 43 rules had one or more "interesting" clusters for the field class label. There was a total of 38 clusters involved in the rules. Previously, we had found 51 clusters "interesting" for the forest class label. 29 of these 51 clusters featured in our association rules.

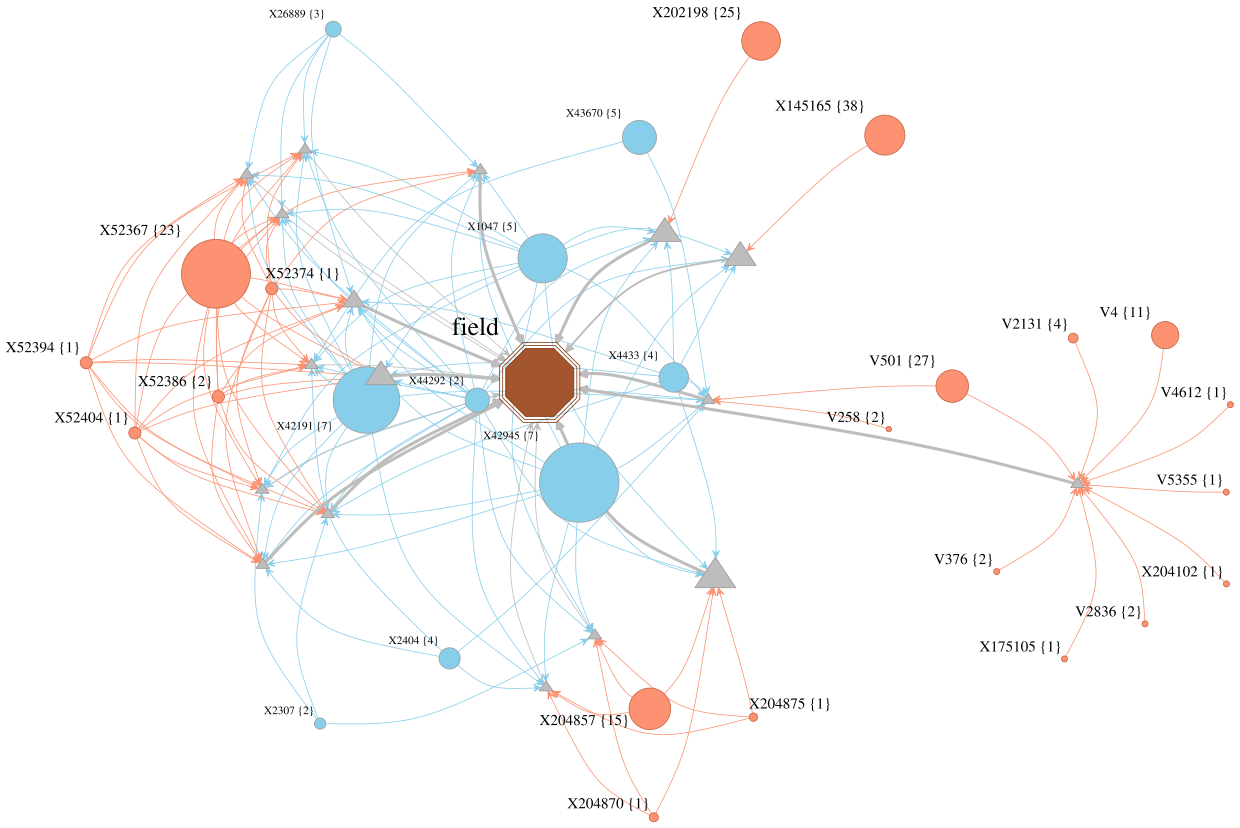

Figure 3-20: Network showing subset of interesting maximal rules for field

To capture the most interesting clusters for the field class label, we introduced new quality measures for the rules as "*interestingness*" and "*interesting size*". The "*interestingness*" quality measured the number of '*interesting'* clusters in a rule. And, the "*interesting size*" quality measured the sum of multiplicity of all the '*interesting'* clusters in a rule. The set of rules, presented in the Figure 3-20, passed the threshold level of 17 for the quality "*interesting size*". The "*interestingness"* of rules is mostly above 3, however, we do see some rules that have single clusters with high multiplicity as well.

There are 5 interesting rules, containing larger pool of clusters or clusters having larger multiplicity. These are clusters involving:

- [X52367] {23}, [X52392] {1}, [X52404] {1}, [X52386] {2}, [X52374] {1}
- [X204857] {15}, [X204870] {1}, [X204875] {1}
- [X145165] {38}

- [X202198] {25}
- [V501] {27}, [V2131] {4}, [V4] {11}, [V612] {1}, [V5355] {1}, [X204102] {1}, [V2836] {2}, [X175105] {1}, V376] {2}

The other rules are mainly based on cross-support clusters that have high support throughout the samples. The lowest support among these set of clusters is 0.64 across all samples, with maximum confidence of 0.717 for field. Among these clusters that feature in more than 60% of the rules for field, X1047 and X4433 are interesting because they have relatively less overall support yet have confidence above 0.7.

3.3.4. Inter Clusters

There were some clusters that were present in maximal rules for both the class labels. These clusters usually have high support, minimum of 0.47 and maximum of 0.94, yet have low confidence to qualify as an "*interesting*" for either of the class label. We'll refer to these clusters as "*inter-clusters*". There was a total of 8 clusters (36 genome variants).

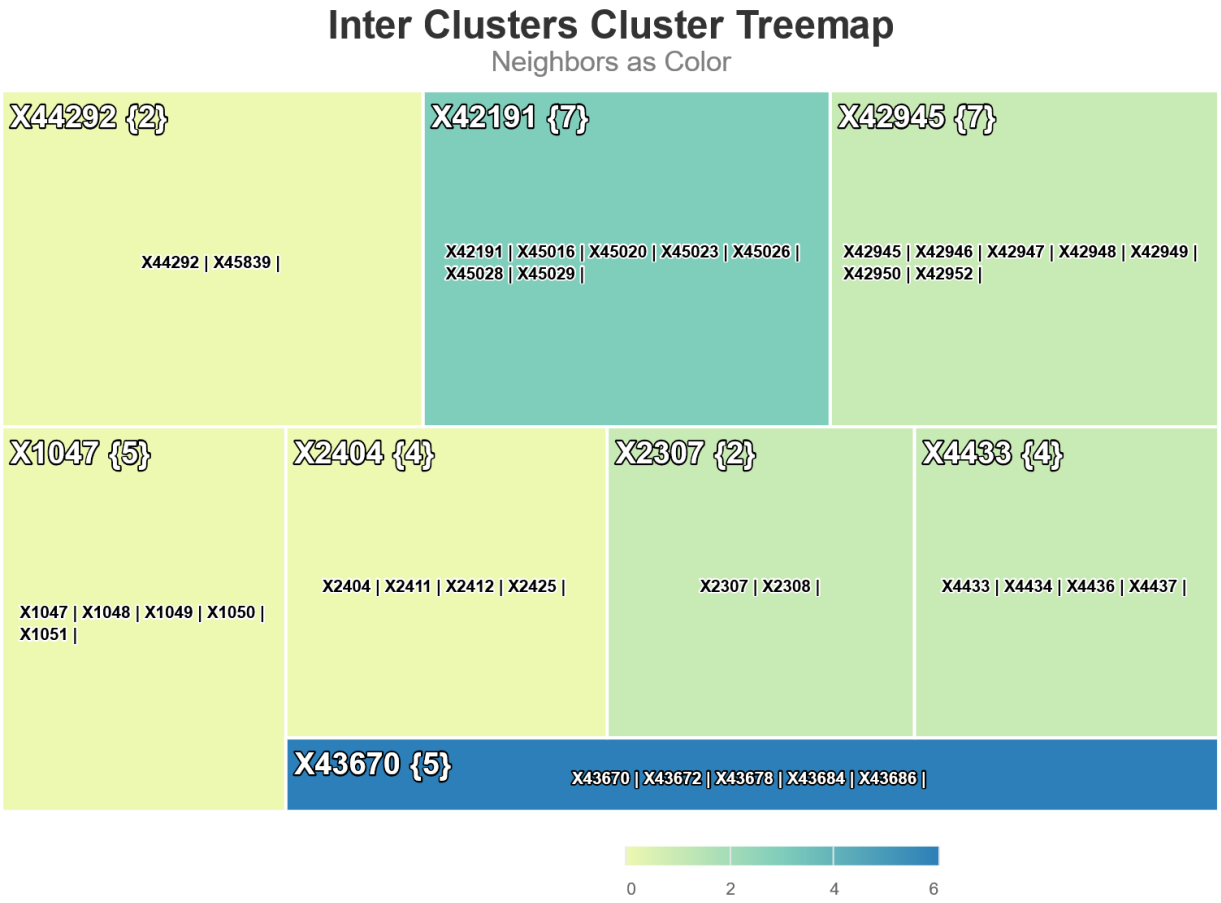

Figure 3-21: Treemap showing "inter-clusters" with members.

The Figure 3-21 shows all the 8 inter-clusters. The size of the cell in the treemap represent the overall support of the cluster across all samples. The color of the cells represents the neighborhood of the cluster. The labels in the cells shows the count of the members and the name of each member in that cluster.

Table 3-3: "Inter-cluster" with their reach across the class label

Cluster	% in Forest rules (18)	% in Field rules (45)	Support	Forest support	Forest confidence	Field support	Field confidence
{5}X1047_Sucrose_permease._ma...faa	5.56	80	0.748	0.214	0.286	0.534	0.714
{2}X2307_FIG00638599._hypothe...fa	66.67	26.67	0.656	0.282	0.430	0.374	0.570
{4}X2404_putative_aminopeptid...faa	61.11	46.67	0.687	0.282	0.411	0.405	0.589
{7}X42191_FIG00638993._hypothe...f	66.67	68.89	0.939	0.351	0.374	0.588	0.626
{7}X42945_Putative_transport_p...faa	61.11	86.67	0.893	0.267	0.299	0.626	0.701
{5}X43670_Transcriptional_acti...faa	5.56	35.56	0.473	0.122	0.258	0.351	0.742
{2}X44292_Aldo.keto_reductase.faa	61.11	97.78	0.969	0.321	0.331	0.649	0.669
{4}X4433_Hypothetical_fimbria...faa	33.33	71.11	0.649	0.183	0.282	0.466	0.718

Table 3-3 shows the properties of the 8 inter-genes. X44292 has the most coverage in terms of overall support, 0.969, and is present in more than 60% of the forest rules and 97% of the field rules. This is the most obvious cross-support item, and forms hyper clique patterns in the rules. The other notable cross-support items are X42945 and X42191. Similarly, X43670 has the least coverage in terms of overall support, 0.473, and is present in less than 6% of the forest rules and 36% of the field rules.

Figure 3-22 shows that these inter clusters are quite different in terms of occurrence. The closest occurrence distance is 10 while the largest is 95.

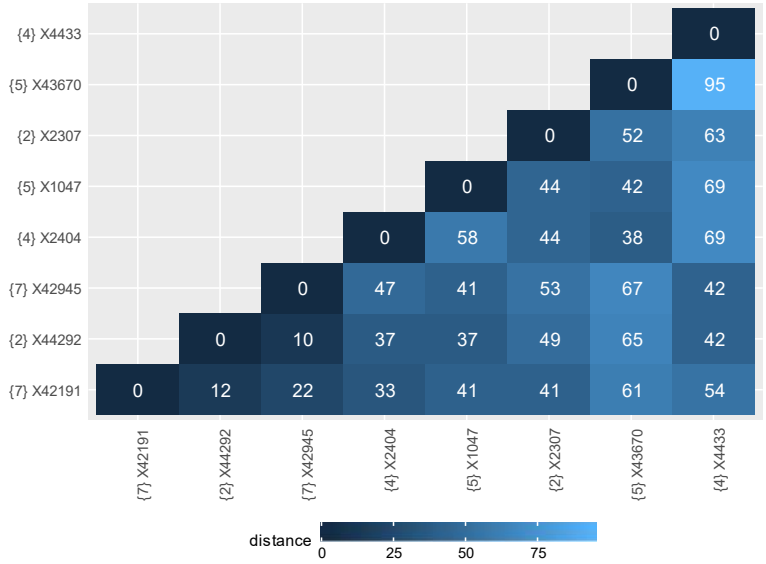

Figure 3-22: Heatmap showing occurrence trait distance among "inter-clusters"

The clusters [X44292], [X42191] and [X42945] are closest compared to other clusters, as seen from Figure 3-22. This was obvious as the minimum overall support among these clusters is 0.89. The multiplicity and member list of these cluster are shown in Table 3-4.

Table 3-4: Common cross-support items with members

Cluster	Size	Member
[X44292]	2	"X44292_Aldo.keto_reductase.faa" "X45839_Uncharacterized_prot...faa".
[X42191]	7	X42191_FIG00638993._hypothe...faa", "X45016_Hypothetical_protein...faa", "X45020_DUF1440_domain.conta...faa", "X45023_CFA.I_fimbrial_chape...faa","X45026_CFA.I_fimbrial_auxil...faa", "X45028_FIGfam014588._Predic...faa" and "X45029_LSU_ribosomal_protei...faa".
[X42945]	7	X42945_Putative_transport_p...faa","X42946_Putative_HTH.type_tr...faa", "X42947_Hypothetical_oxidore...faa","X42948_Uncharacterized_suga...faa"," X42949_Putative_aldolase_Yd...faa","X42950_Hypothetical_zinc.ty...faa" and "X42952_Putative_oxidoreduct...faa".

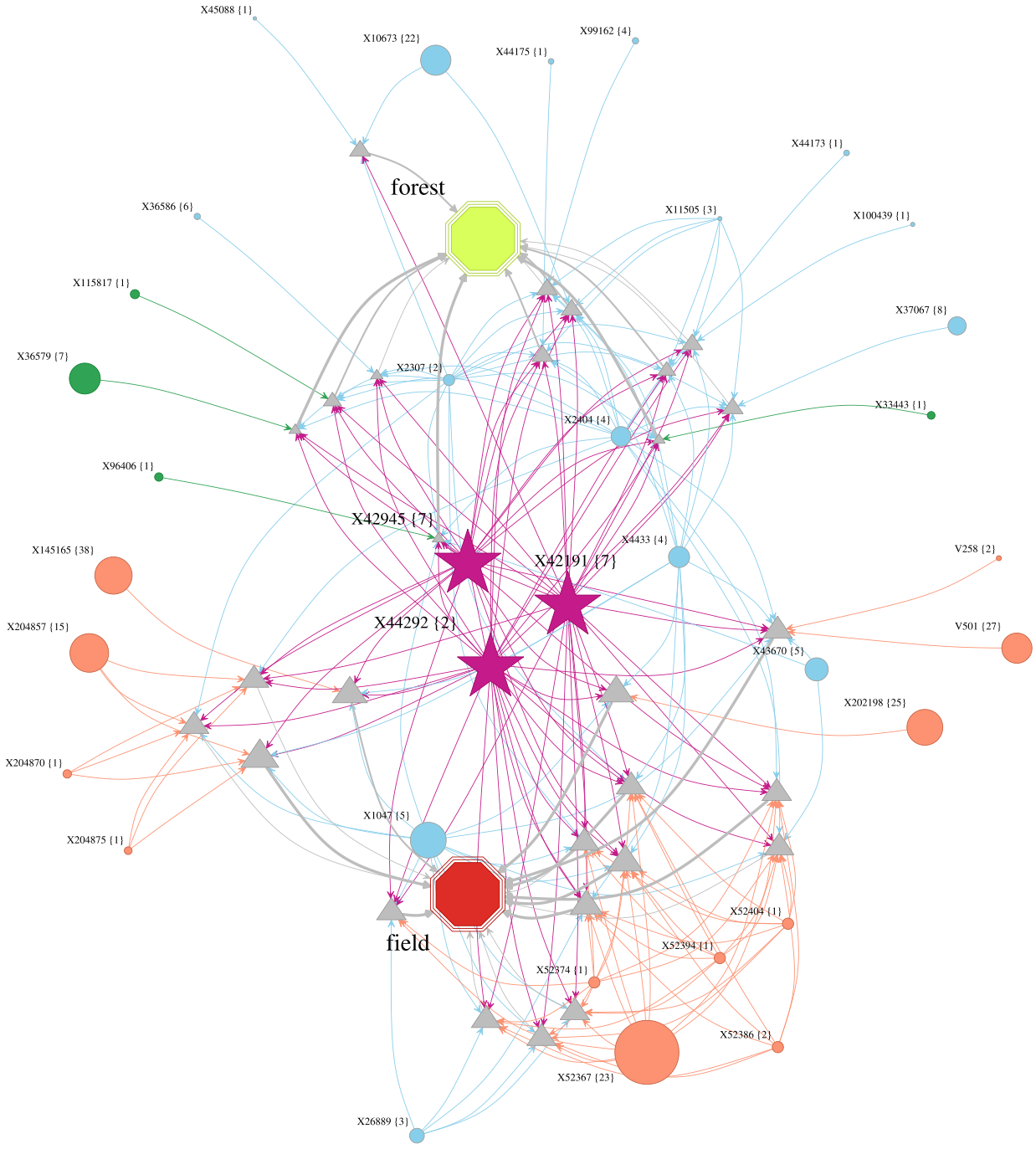

Figure 3-23: Network showing maximal rules involving [X44292], [X42191] and [X42945]

The Figure 3-23 shows the presence of these 3 common clusters, denoted by star (pink) nodes, in combined rules the lead to both forest and field. These common inter-clusters cover 16 out of 17 most interesting rules for field, and 4 out of 5 interesting rules for forest.

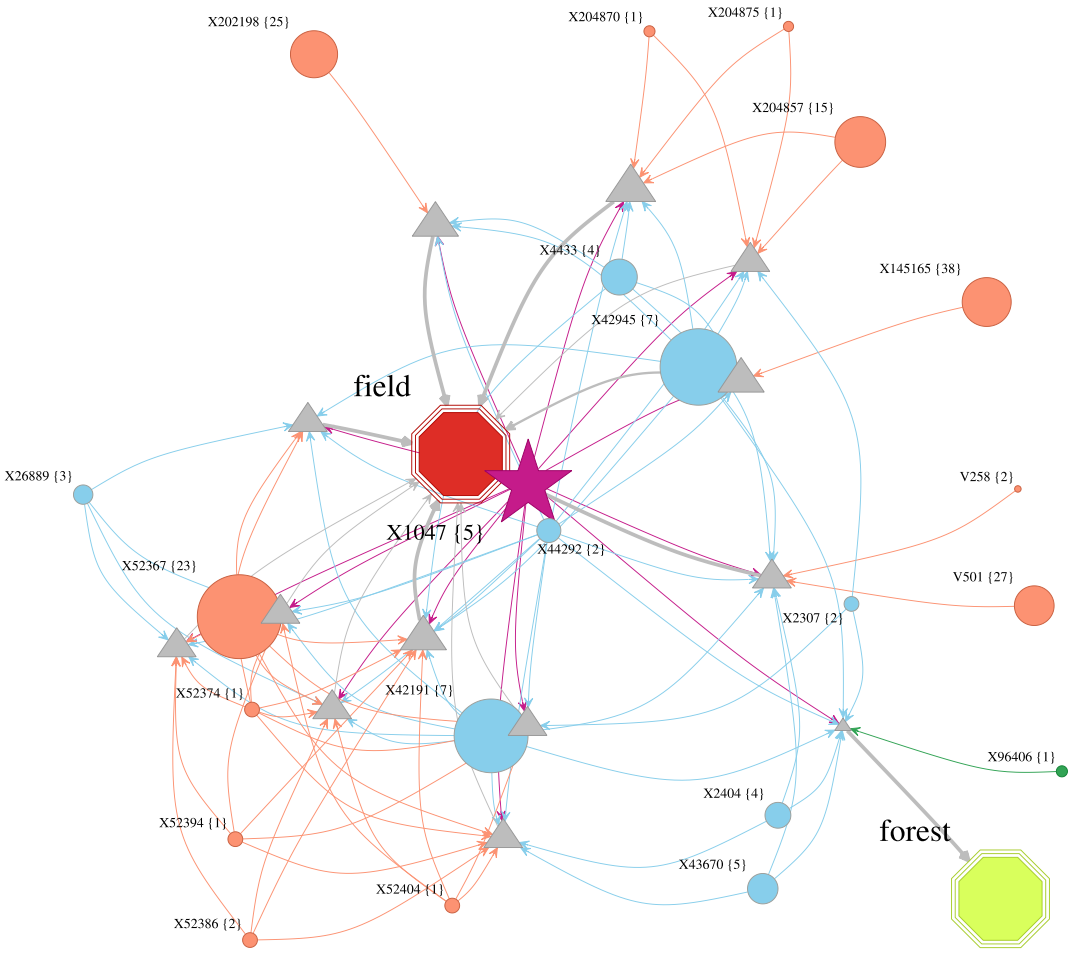

Figure 3-24: Network showing all maximal rules involving [X1047]

The Figure 3-24 shows the interaction of the cluster [X1047], denoted by star (pink) node. It has a multiplicity of 5, and has no other clusters in its neighborhood. The cluster's members are "X1047_Sucrose_permease._ma...faa", "X1048_Fructokinase_.EC_2.7...faa", "X1049_Sucrose.6.phosphate_...faa", "X1050_Sucrose_specific_tra...faa" and "X1051_D.serine_permease_Ds...faa".

As seen from Table 3-3, it covers around 80% of the rules generated for field, whereas just covers around 6%,1 rule, of the rule for the forest class label. It features in 12 out of 17 most interesting rules for the field class label. Similarly, it features in 1 out of 5 interesting rules for forest class label.

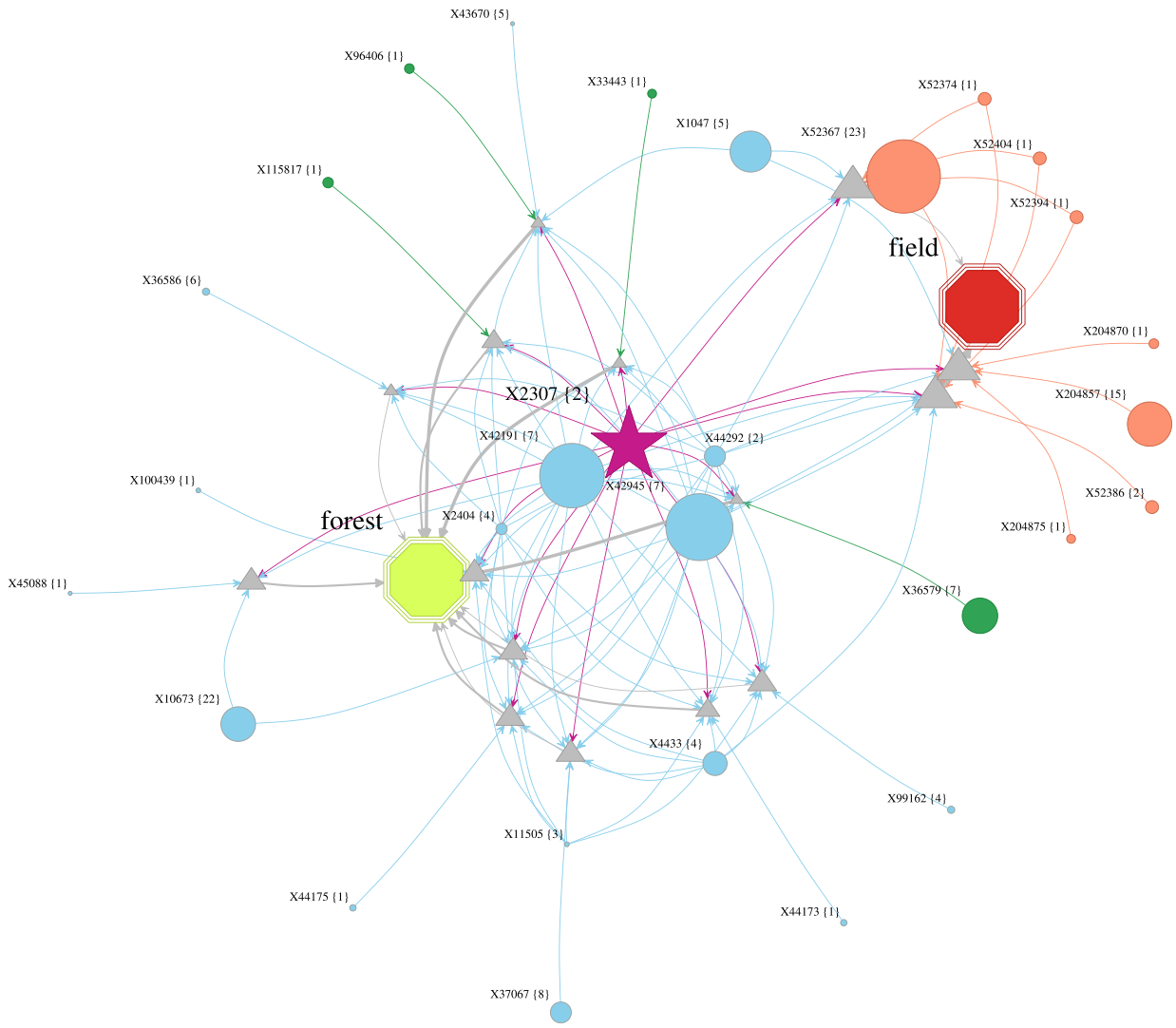

Figure 3-25: Network showing all maximal rules involving [X2307]

The Figure 3-25 shows the interaction of the cluster [X2307], denoted by star (pink) nodes. It has a multiplicity of 2 and a neighborhood of 1. The cluster's members are "X2307_FIG00638599._hypothe...faa" and "X2308_FIG00638505._hypothe...faa". The cluster has a neighbor, containing only "X2310_Phage_portal_protein.faa" as member.

As seen from the Table 3-3, it covers around 67% of the rules for the forest class label, and features in 4 of the 5 interesting rules. Similarly, it covers around 27% of rules for field, and features in 3 out of 17 most interesting rules for field.

4. DAMSELFLIES STUDY

The data set must be transformed to be useful for association analysis, because the climatic attributes of interest would have a continuous data type. We convert these climatic attributes to item format, and find the item values for all location identifiers. Initially, the data set had 158 rows (or transactions) and each row had the class labels with only the location identifier. To estimate the climatic attributes at these location identifiers, we need a climatic data source that contains the appropriate climatic attributes in the interested areas. The problems with selection of such a data source have been discussed in Section 2.3, and includes low coverage, limited attributes and discretization. The pipeline in Figure 2-3 shows the proposed solution to deal with the underlying issues. The proper selection of climatic data source includes examining the source that it contains the climatic attributes that we are interested in and contains enough stations well-dispersed within the area of interest (North Dakota). This is important as the result of interpolation depends on the availability of source near the location identifier. The interpolation methods are cross-validated to find the best performing method. Once the climatic attributes are estimated, they are then discretized to find the bins (clusters) of the climatic attributes. This discretization process is important, as we must find a proper balance between support and confidence. High support could lead to low confidence, and vice-versa. Once discretized, we end up with multi-tier items values. This means that an item could have different instances reflected by the bin number. For the association analysis process, *item1=0* and *item1=1* would be different items even though they represent the same column *item1*. Each transaction can only contain one instance of a climatic attribute. Once the data for ARM is prepared, the two class labels are also included as items and the class labels items would only feature in the *rhs* of the rules.

The items that would be the most important to a class label would be the one with higher support, higher confidence and pass the cutoff support for the bins (instances) of the climatic

attribute. The high support will filter the high occurrence of the items for the class label. The high confidence will filter the items that occur more frequently with either of the class labels, and remove the cross-support items. The discretization process creates different number of bins for each climatic attribute, so they need to be normalized and have different cutoff support for each climatic attribute. The attributes with lower number of bins have higher cutoff support and vice-versa.

From an ecological perspective, we had a species occurrence data set that contained the sampled location (longitude and latitude) for various species, a total of 50798 samples. These set of samples were collected from 1990 to 2015, and mostly during the summer season. We were particularly interested in two types of damselflies, namely *River Jewelwing* and *American Rubyspot*. Both damselflies belong to the order *Odonata* and family *Calopterygidae*. There was a total of 2176 occurrence records of species from *Odonata* order, and 158 samples were from the interested species. Out of 158 occurrence records, 98 belonged to *River Jewelwing* and 60 belonged to *American Rubyspot*. These set of interested samples were collected from 1995 to 2015, mostly during summer. The occurrence location of these samples in and around North Dakota is shown in the Figure 4-1, where green dots represents occurrence of *River Jewelwing*, red dots represents occurrence of *American Rubyspot* and yellow dots represents occurrence of both the species. The dots lying on the same point has been slightly jittered to avoid absolute overlapping of dots.

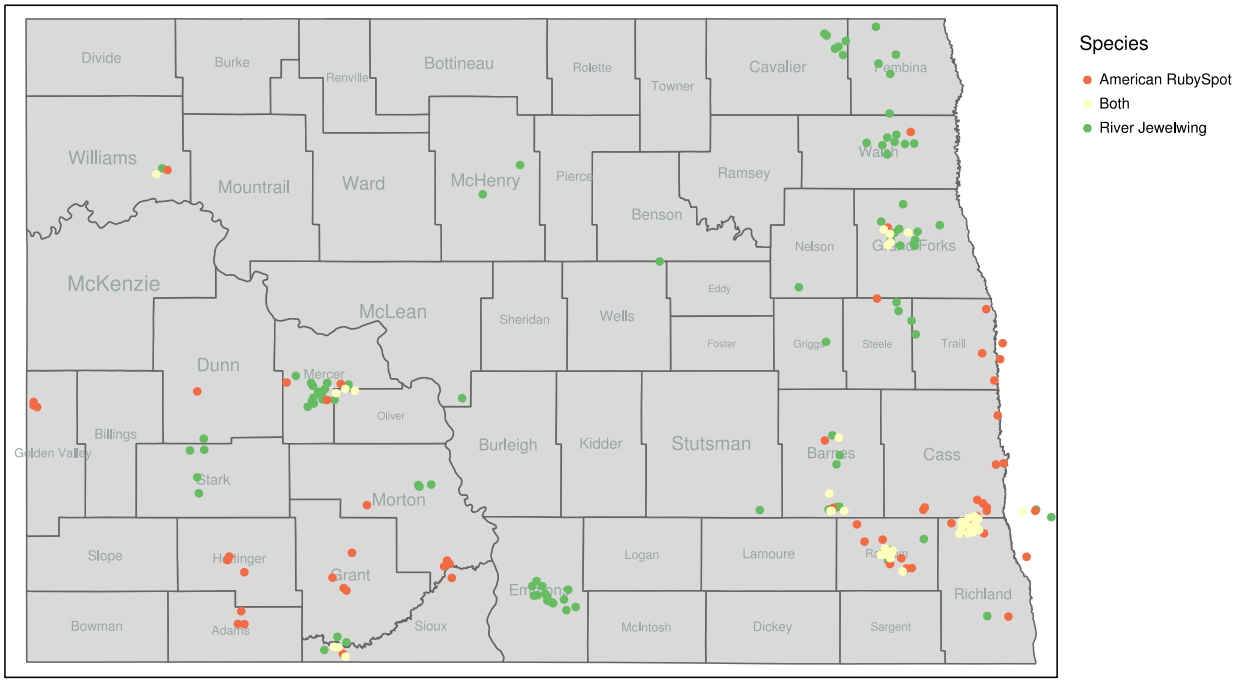

Figure 4-1: Occurrence distribution of American Rubyspot and River Jewelwing

4.1. Motivation

The given two species of damselflies are very similar in nature, yet they are rarely found together. We want to explore whether local climate could be a possible factor behind such distribution. For this, we need climatic conditions for the occurrence points at the dates that they were collected. Then, we categorize each climatic attribute and apply association rule mining to find climatic condition that is mostly prevalent for each class of the given damselflies. The 'interesting' association rules would be analyzed with visualization tools, and examined for significant differences in the climatic conditions of their usual occurrences. We use single and longer association rules to find the relation of a species with individual climatic attributes and group of climatic attributes respectively.

We started off with finding a suitable climatic data set that had good coverage in terms of the time period and location of the samples collected. We, then, validated various geo-spatial interpolation methods by calculating the estimation errors for the climatic data set. We used the best

performing geo-spatial interpolation method to estimate the climatic conditions of the occurrence locations based on the climatic data set. Once estimated, we categorized the climatic attributes by discretizing the estimated values into bins. These bins along with the damselfly class labels are the raw data and basis for association analysis.

4.2. Pre-Processing

4.2.1. Climate Data set

To find the climatic conditions in the region of interest at certain historical points in time, we had to find a climatic data set from a known and established source. This data set should contain the climatic attributes of interest and have good coverage over the region of interest in the given time frame. We identified few sources, out of which climatic data from National Climate Data Center (NCDC) and NDAWN looked promising. We decided to pursue data set from NDAWN, because

- It had most of the climatic attributes that we were interested in
- It had good coverage all over the ND state (region of interest)

Likewise, we have the options of using daily, monthly or yearly data. Note that the monthly and yearly data are averages of the daily data for the given month or year respectively. We settled on using monthly data, which is mean of the daily data for the given month. This avoids any anomaly seen for the daily data, and rather focusses on the overall monthly climatic scenario.

4.2.2. Climatic attributes

The climatic attributes from the NDAWN source are listed below [23]:

- Air Temperature: *Max Temperature*, *Min Temperature* and *Average Temperature*.
- Soil Temperature: *Average Bare Soil Temperature* and *Average Turf Soil Temperature*.
- Wind Speed: *Average Wind Speed* and *Max Wind Speed*.
- Solar Radiation: *Total Solar Radiation*.

- PET: *Average Penman PET* and *Total Penman PET*.
- Rainfall: *Total Rainfall*.
- Dew Point: *Average Dew Point*.
- Wind Chill Temperature: *Average Wind Chill*.

4.2.3. Stations

The distribution of the stations and their availability for all the samples and the interested samples are shown in the Figure 4-2 and Figure 4-3 respectively. As we can see, the climate stations are well distributed throughout all the counties of North Dakota (ND). The stations are denoted by colored dots, and the color of the dots denotes the availability of these stations in months for the given period. The yellow dots represent lower availability whereas the red dots represent higher availability.

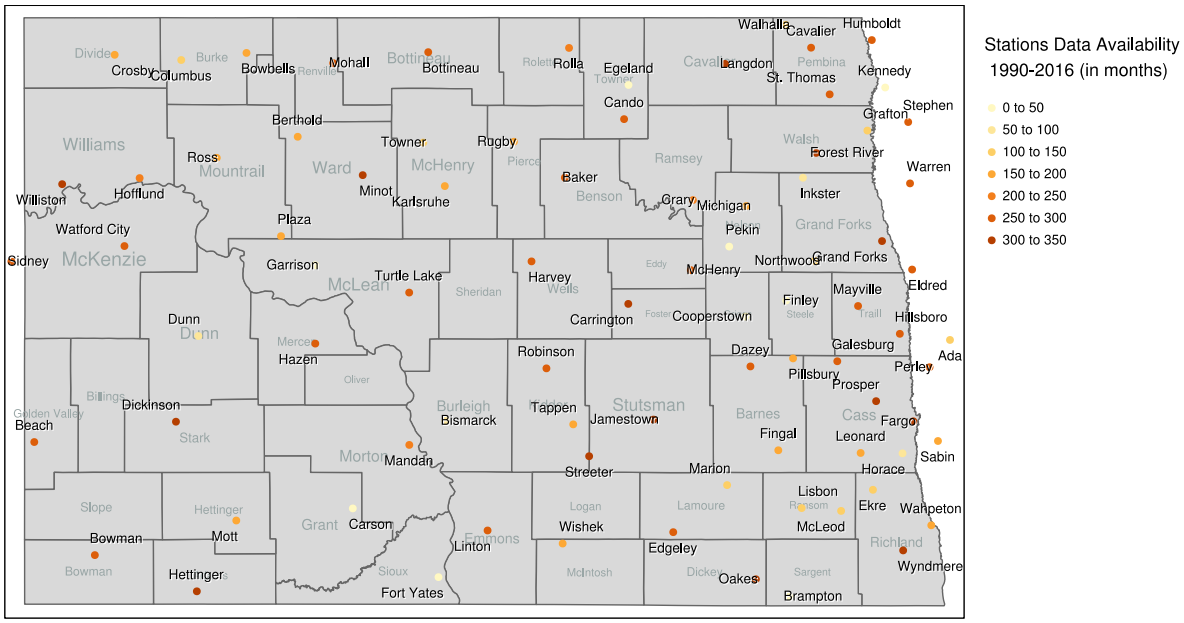

Figure 4-2: Station Availability and Coverage from 1990-2016

The Figure 4-2 shows all the stations with their availability in months for the period 1990-2016. The data were available for a total of 94 stations. The stations having lower coverage are relatively new stations and have been in function only recently.

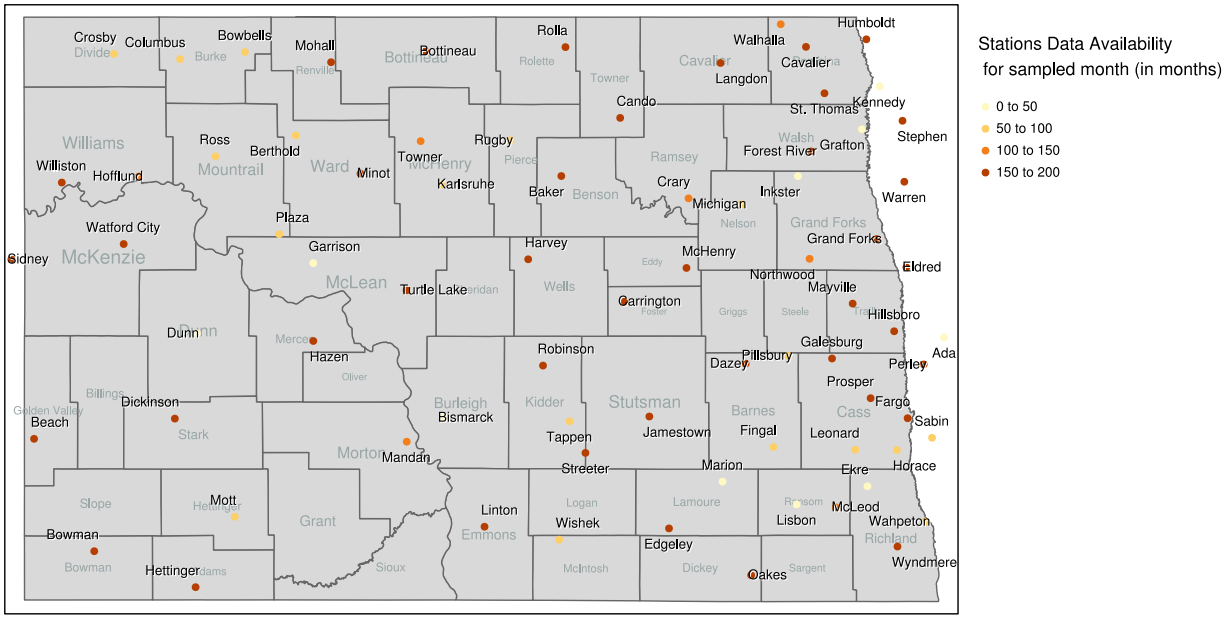

Figure 4-3: Stations Availability and Coverage for the time(month) with occurrence record

The Figure 4-3 shows the stations and their availability in months in which there were one or more occurrence records. The data were available for a total of 81 stations. We can see high coverage stations well distributed throughout ND. The Figure 4-4 shows the coverage of the stations where around half of the stations have data available for more than 150 months. We'll refer the occurrence records of the interested damselflies as 'samples' throughout the remainder of the document.

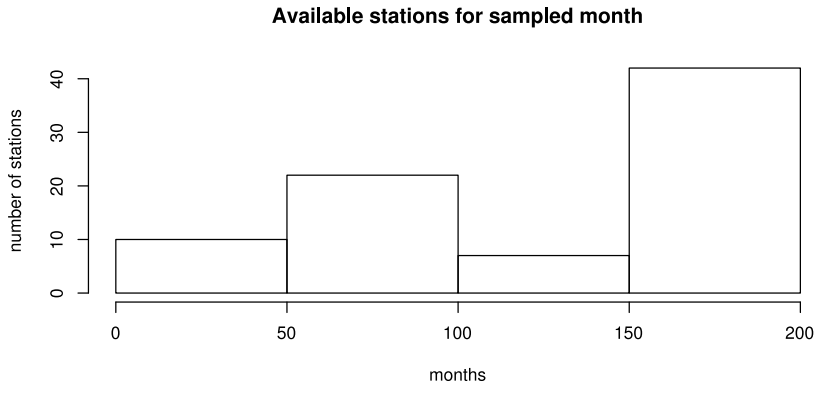

Figure 4-4: Histogram showing availability of stations for the interested records

4.2.4. Interpolation

4.2.4.1. Intro

To estimate climatic conditions for the location of the samples, we need to interpolate data based on the climatic data from the available stations. There exists various spatial interpolation methods, and have been applied to various disciplines. The choice of an interpolation method could be data-specific or even variable-specific. The choice could be affected by numerous factors including the size of the data available, the design of the samples and the properties they represent [24]. In our study, we are interested in predicting environmental attributes for our sample points from reliable and known data set. We look at two methods with their various configurations to come up with an interpolation method that provides the best estimation. We use leave-one-out cross validation to gather the prediction errors, and choose the method that gives us the least RMSE values as the best-performing method.

4.2.4.2. Methods

- IDW (Inverse-Distance Weighted): *Deterministic and non-geostatistical*

IDW is a nearest neighbor interpolation technique in which more than one nearest neighbors are considered. This is a deterministic approach to spatial interpolation [25]. The value at a certain unknown point is obtained by linear combination of the surrounding data points, with each data point given a certain weight [26]. The weight of the data point is determined by the distance from the unknown point by an inverse function of the distance between the two points. The core assumption of this method is that the data point nearer to the unknown point is more similar or significant the data points that are farther away [24].

$$\lambda_i = \frac{1 \,/\, d_i^p}{\sum_{i=1}^{n} 1 \,/\, d_i^p}$$

where, d_i represents the distance between the unknown point and data point *i*, *p* represents the power parameter and *n* represents the number of data points for the model. One of the key factors that affect the accuracy of the model is the value of the power parameter [27]. The weight of the data point diminishes as the distance increases, especially when the power parameter increases. This means that the nearby data points will have higher weights and thus, its effect on the estimation of the unknown point will be higher. We consider all available stations as neighbors in the model. We use *idw()* function from the '*gstat'* [28] R library to find the idw estimation. We use variation of idw in terms of power of 2, 3, 5, 8, 10 and 15.

$$gstat::idw(formula, locations, newdatagrid, idp = power)$$

- Kriging: *Probabilistic and geostatistical*

Kriging is an interpolation technique based on regression against a set of observed z-values of surrounding data points. These surrounding data points are weighted according to spatial covariance values. This is a geostatistical and probabilistic approach to spatial interpolation. Kriging assigns weights based on moderately data-driven weighting function, instead of an arbitrary function like in the case of IDW [29].

Kriging is based on spatial correlation. The basic tool used in geostatistics and kriging is the semi-variogram. It helps to capture the spatial dependence among the samples using semi-variogram against the separation distance [30]. Thus, weights are formulated using the semi variogram model. We use *autofitVariogram()* function from the '*automap'* [31] R library to find the semi-variogram for the model. Then, we use *krige ()* function from the *'gstat'* [28] R library to find the krige estimation.

$$automap::autofitVariogram(formula, locations)$$

$$gstat::krige(formula, locations, grid, variogram\_model)$$

The types of kriging depend primarily on the formula as well as other parameters used in the API calls. The types include:

- Ordinary Kriging: This is the basic form of Kriging. The predicted values from ordinary kriging is a linear combination of the measured values. The mean is unknown [26]. The formula is $attr \sim 1$, where *attr* is the dependable variable (climatic attribute)
- Simple Kriging: Simple Kriging is an advanced form of ordinary kriging, with a known mean. The formula is $attr \sim 1$, where *attr* is the dependable variable (climatic attribute).
  We also use the *beta* argument (additional) that contains the mean coefficients $lm(prop \sim 1, P)\$coefficients$.
- Universal Kriging: In this form of kriging, it uses a regression model to model the mean value expressed as linear or quadratic trend. The formula for *Universal Kriging-1* is $attr \sim x + y$, where *attr* represents the dependable variable, *x* represents the longitude and *y* represents latitude values. The formula means that *attr* is linearly dependent on *x* and *y* [25]. The formula for *Universal Kriging-2* is $attr \sim x + y + I(x^2) + I(y^2) + I(xy)$, where *attr* represents the dependable variable, *x* represents the longitude and *y* represents the latitude value. The formula means that *attr* is quadratically dependent on *x* and *y* [25].

Table 4-1: Summary of interpolation methods

Method	Configuration	Remarks
IDW	IDW - 02	Power Value of 2
	IDW – 03	Power Value of 3
	IDW – 05	Power Value of 5
	IDW – 08	Power Value of 8
	IDW – 10	Power Value of 10
	IDW - 15	Power Value of 15
Krige	Ordinary-Krige	Unknown Mean
	Simple-Krige	Known Mean
	Universal Kriging-1	Linear Trend
	Universal Kriging-2	Quadratic Trend

4.2.4.3. Cross Validation

We used leave-one-out cross validation technique to validate the methods and their configurations. We measured the Root Mean Squared Error (RMSE) of the predicted and the actual values for the stations with each method. These RMSE values represent the variability in the estimation process. Note that only the stations data from NDAWN climate source would be used to validate the methods.

Algorithm Leave-one out Cross Validation and Normalization

procedure X-VALIDATE()
    $ndawn \leftarrow$ climatic data, with attributes and values
    $climaticVar \leftarrow$ list of climatic variables in stations data
    $uniqueDate \leftarrow$ list of unique date $(month, year)$ in ndawn data
    $method \leftarrow$ an interpolation method with a certain configuration
    for each climatic variable cv in $climaticVar$ **do**
        for each date d in $uniqueDate$ **do**
            $station \leftarrow$ extract climatic data of stations for the date d
            $range(cv_d) \leftarrow \max(station_{cv})$ - $\min(station_{cv})$
            $N \leftarrow$ the number of $station$
            for each station data sd in $station$ **do**
                $test \leftarrow$ sd ▷ This is the test data
                $train \leftarrow$ station except sd ▷ This is the train data
                $model \leftarrow$ model based on $method$ with $train$ data
                $actual \leftarrow$ actual cv value for $test$
                $predict \leftarrow$ predicted cv value for $test$ ▷ push to list
            end for
            $RMSE(cv_d) \leftarrow \sqrt{\sum(actual - predict)^2/N}$
        end for
        $RMSE(cv) \leftarrow$ mean value of $RMSE(cv_d)$
        $range(cv) \leftarrow$ mean value of $range(cv_d)$
        $normalize_dRMSE(cv) \leftarrow RMSE(cv)$ / $range(cv)$
    end for
    Return $normalize_dRMSE(cv)$
end procedure

The above algorithm shows the calculation of RMSE and the normalization of raw RMSE values for a method. The NDAWN climatic data is processed based on climatic variables and unique dates. First, all the stations data for an instance of a unique date are gathered. Then, we split the data into testing and training set by using leave-one-out cross-validation technique. This technique is the most extreme form of cross validation where a single data is regarded as test data and all the

remaining data are used for training [32]. The predicted values are generated for all the stations by treating each as testing one by one, and for all the unique dates.

The actual values and the predicted values of each station are temporarily stored in a list and then, the raw RMSE values for each unique date for each climatic variable, $RMSE(cv_d)$, are calculated as shown at line 19. Furthermore, the RMSE for each climatic variable, $RMSE(cv)$, is calculated by getting the mean of $RMSE(cv_d)$ as shown at line 21. The same procedure is repeated for all the available methods by changing the model at line 5 and 15. The RMSE values for each climatic variable for the methods are shown in the Table 4-2.

Table 4-2: Mean RMSE values for each climatic attribute for all methods

Climatic Variable	**IDW -02**	**IDW -03**	**IDW-05**	**IDW -08**	**IDW -10**	**IDW -15**	**Simple Krige**	**Ordinary Krige**	**Universal Kriging-1**	**Universal Kriging-2**
Max.Temp	1.449	1.221	1.168	1.224	1.256	1.310	1.064	1.071	1.008	1.096
Min.Temp	1.422	1.313	1.335	1.409	1.445	1.505	1.236	1.241	1.246	1.229
Avg.Temp	1.217	1.040	1.002	1.046	1.072	1.117	0.904	0.909	0.895	0.917
Avg.Bare.Soil. Temp	2.468	2.541	2.705	2.851	2.910	2.999	2.358	2.395	2.398	2.588
Avg.Turf.Soil. Temp	2.406	2.466	2.616	2.751	2.807	2.891	2.349	2.369	2.368	2.415
Avg.Wind. Speed	1.045	1.073	1.157	1.241	1.274	1.322	1.062	1.070	1.076	1.156
Max.Wind. Speed	1.565	1.588	1.700	1.811	1.855	1.920	1.585	1.594	1.576	1.651
Total.Solar. Rad	16.03	15.73	16.487	17.36	17.72	18.25	15.350	15.450	14.908	15.285
Avg.Penman. PET	0.011	0.010	0.010	0.011	0.011	0.011	0.010	0.010	0.009	0.010
Total.Penman.PET	0.343	0.316	0.317	0.331	0.338	0.350	0.306	0.307	0.290	0.296
Total.Rainfall	0.778	0.751	0.782	0.828	0.848	0.876	0.751	0.755	0.751	0.806
Avg.Dew.Point	1.282	1.127	1.120	1.171	1.197	1.240	1.042	1.031	1.022	1.013
Avg.Wind. Chill	1.556	1.379	1.375	1.451	1.489	1.551	1.292	1.299	1.253	1.314

The predicted values for a station depends on the station values (training) used in the model. If the training values for a climatic variable is widely distributed, then it might lead to larger RMSE

values compared to one which is closely distributed. To assess such disparity, we calculate raw ranges, *range(cv$_d$)*, as the difference of the minimum and maximum values of each climatic variable for each unique date. Then, mean ranges for a climatic variable, *range(cv)*, is calculated by averaging the raw ranges of that climatic variable for all the unique dates. Note that the *range(cv)* would remain the same for all the methods or models used, as it is based on the source data from NDAWN. These ranges are utilized in the normalization process. The mean ranges for the climatic variables are given in the Figure 4-5.

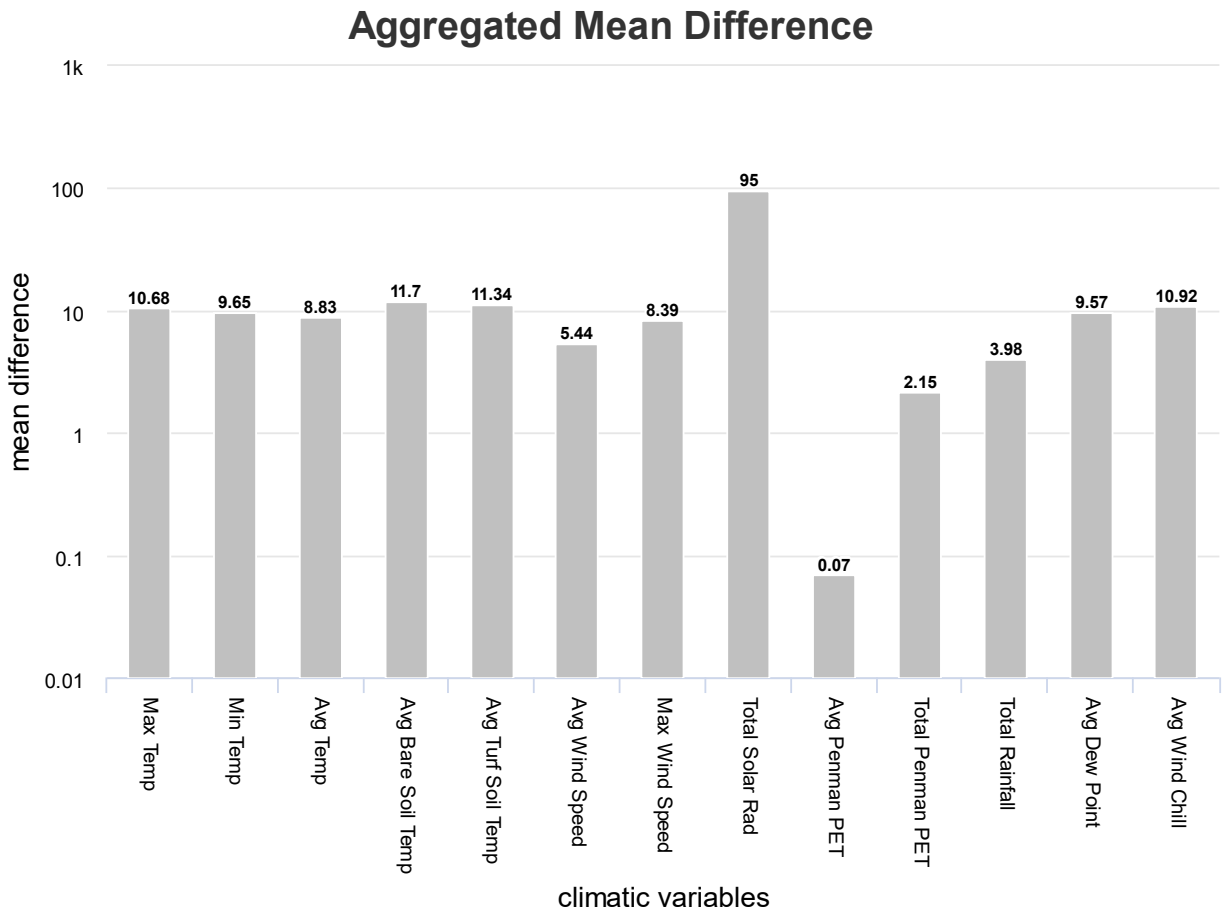

Figure 4-5: Mean Ranges for all the climatic attributes

Figure 4-5 shows that the range for *Total Solar Rad* attribute was the highest, while range for *Avg. Penman PET* attribute was lowest. This explains the anomaly seen in the Table 4-2, where the RMSE for these attributes were in extremes high and low respectively.

There is a need for normalization because the range of values of the various climatic attributes is varying. This means that for some attributes, the RMSE may seems high but in truth, the actual discrepancy in the predicted and actual values might be less. From Table 4-2, we can see that *Total Solar Rad* attribute has the largest RMSE values, yet this may not be a true reflection of the discrepancy. Figure 4-5 shows that the ranges of value for the attribute was around 95, which is very large compared to ranges from all the other attributes.

The normalized RMSE values would then be aggregated to find the overall quality of the interpolation methods. Let's consider a climatic variable (*cv*) for a particular method (*M*) at a given date (*de*) has a raw RMSE ($RMSE<raw>_{M,cv,de}$), and has a range ($Range<raw>_{cv,de}$). Also, let there be (d) number of unique dates present in the source data set. In order to find the normalized RMSE, we first find the mean RMSE ($RMSE<mean>_{M,cv}$) and mean ranges ($Range<mean>_{cv}$) for each climatic variable, by averaging on all the unique dates. Then, each mean RMSE values is divided by the mean range of the respective climatic variable to find the normalized RMSE. The necessary formulas are given below:

$$RMSE < mean >_{M,cv} = \frac{\sum RMSE < raw >_{M,cv,de}}{d}$$

$$Range < mean >_{cv} = \frac{\sum Range < raw >_{cv,de}}{d}$$

$$RMSE < norm. >_{M,cv} = \frac{RMSE < mean >_{M,cv}}{Range < mean >_{cv}}$$

The mean RMSE for the different methods and climatic variables, as shown in the formulae 1, is shown in the Table 4-2. The mean ranges for each climatic variable, as shown in the formulae 2, is shown in the Figure 4-5. The normalized mean RMSE for the different methods and climatic variables, as shown in the formulae 3, is shown the Figure 4-6.

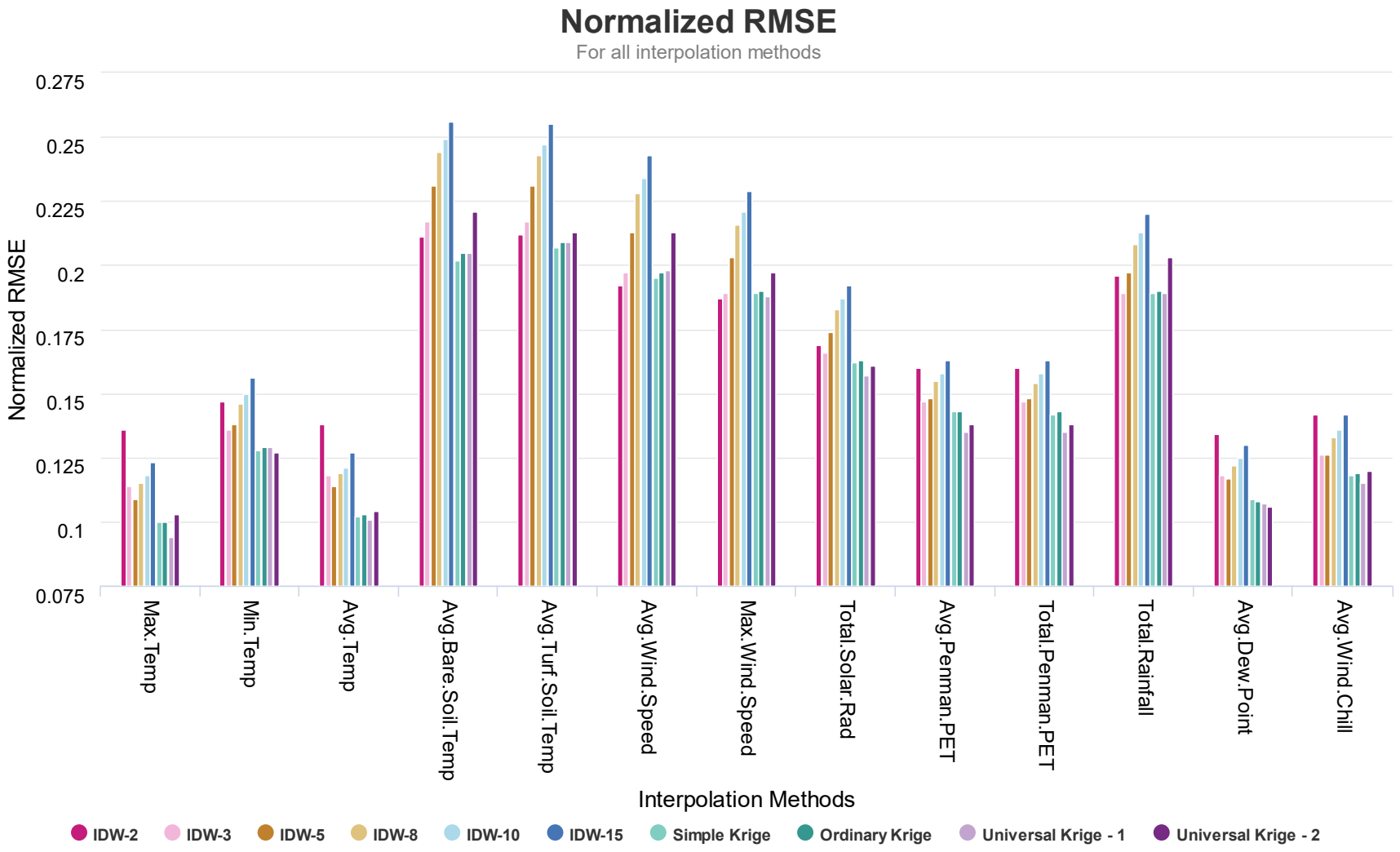

Figure 4-6: Normalized RMSE values

Figure 4-6 shows the measurement of the normalized mean RMSE of different interpolation methods with their configurations for the climatic variables. It shows that *Universal Krige–1* performs best in most cases (in terms of number of climatic variables). The figure also indicates how each climatic variable responded to methods, i.e., the variability in predicted and actual values.

We select the best method of interpolation based on the lowest aggregated RMSE for all climatic variables. Table 4-3 shows these aggregated RMSE values. As we can see, *Universal Kriging-1* performed better with the lowest aggregated RMSE value of 1.962, closely followed *Simple Kriging* with value of 1.986. *IDW-15* and *IDW-10* performed worst with aggregated RMSE values of 2.398 and 2.317 respectively. The best performing IDW method was *IDW-3* with value of 2.082.

Table 4-3: Aggregated normalized RMSE for all methods

Method	Aggregated RMSE
IDW-10	2.317
IDW-15	2.398
IDW-2	2.183
IDW-3	2.082
IDW-5	2.147
IDW-8	2.265
Simple Kriging	1.986
Ordinary Kriging	1.998
Universal Kriging-1	1.962
Universal Kriging-2	2.043

Similarly, we can see the variability in the prediction of the climatic attributes by the aggregating the RMSE values for all the methods. Figure 4-7 shows these aggregated RMSE values for all the climatic attributes. Temperature attributes (*Max. Temp*, *Min. Temp* and *Avg. Temp*) and *Avg. Dew. Point* attribute had the lowest variability, while soil temperature attributes (*Avg. Bare Soil Temp* and *Avg. Turf Soil Temp*) performed worse closely followed by wind attributes (*Avg. Wind Speed* and *Max. Wind Speed*). Figure 4-8 indicates the RMSE values across different methods for each climatic attribute, and thus confirms the claim.

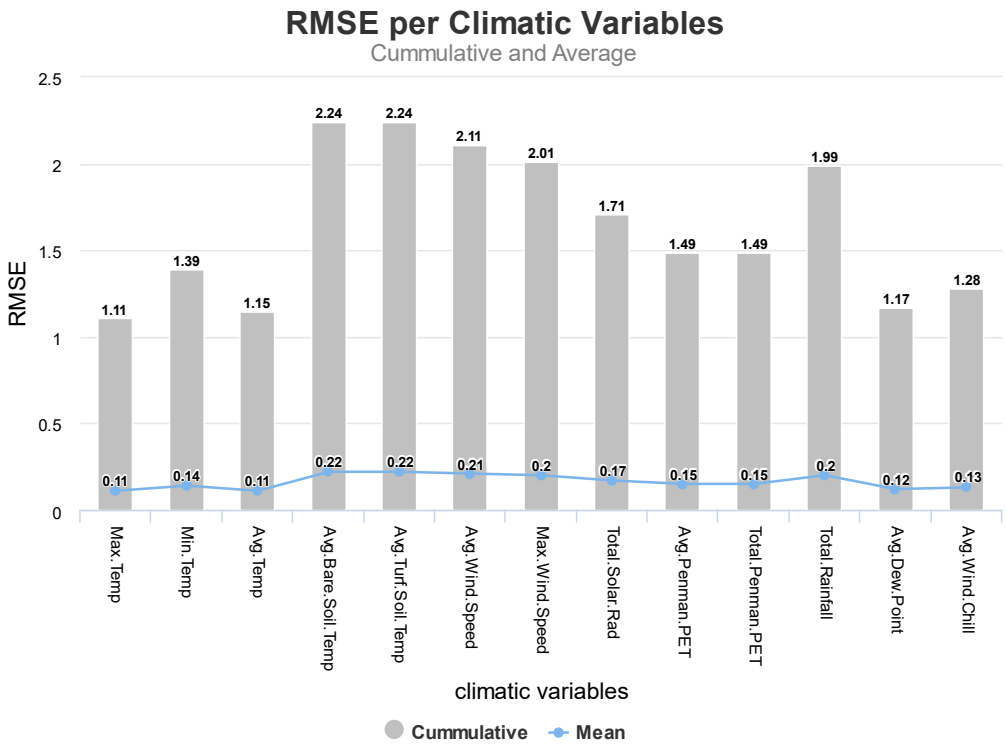

Figure 4-7: Aggregated normalized RMSE for all attributes

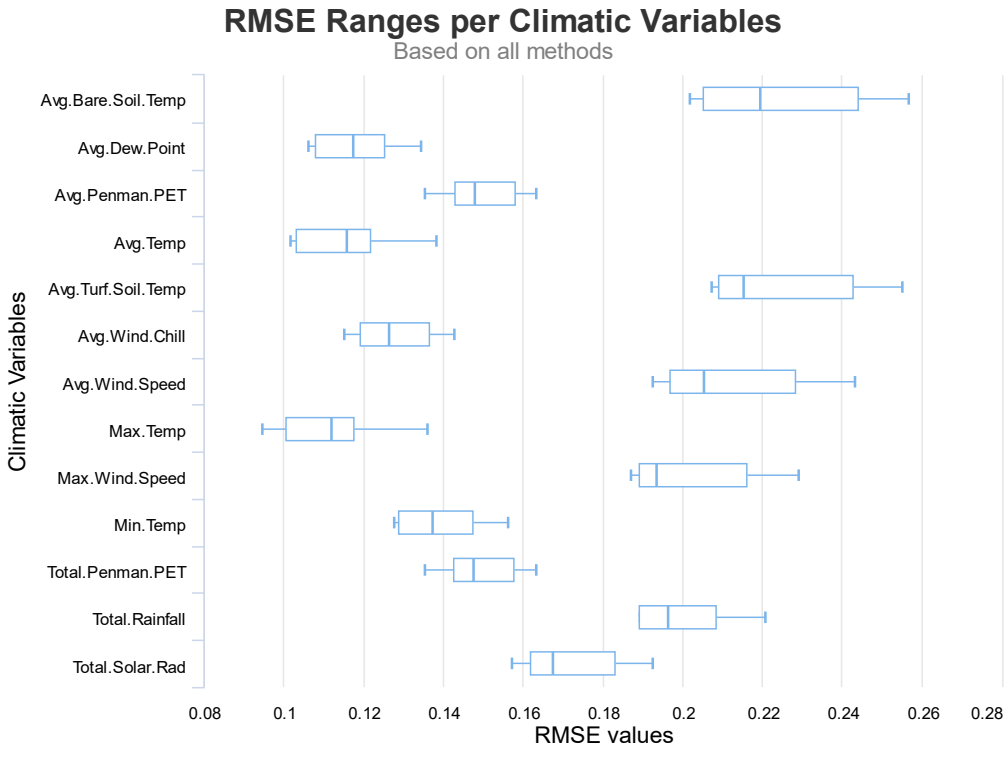

Figure 4-8: Box-plot of the normalized RMSE values for all attributes

4.2.4.4. Prediction

The previously discussed cross-validation technique helps us to find the best method that can be used to interpolate climatic variables for the sample points. As we can see from Table 4-3, *Universal Kriging-1* performed best in terms of cumulative RMSE scores for all the climatic variables. From Figure 4-6, we can see that it does perform best for 7 out of 13 climatic variables and even for the remaining 6 variables, its performance can be considered decent. Moreover, the variability is comparatively low for all the climatic variables where *Universal Kriging-1* performed best. Hence, this method was used as the interpolation method for the estimation of all the climatic attributes in the occurrence locations.

Algorithm Interpolating climatic variables for sample locations

```
1: procedure GetClimateData()
2:     stations ← list of all stations with climatic information
3:     samples ← list of all samples with geolocation
4:     climaticVar ← list of climatic variables in stations data
5:     sampleDate ← list of unique date (month, year) when samples were collected
6:     temp ← empty
7:     for each date d in sampleDate do
8:         station ← extract climatic data of stations for the date d
9:         sample ← extract geolocation of samples for the date d
10:        for each climatic variable cv in climaticVar do
11:            model ← building model from station and cv
12:            extract ← interpolating cv data for sample
13:            Add sample and cv information to temp
14:        end for
15:    end for
16:    Return temp
17: end procedure
```

The above algorithm shows the process of estimation of the occurrence locations. The process is like the cross-validation process. The occurrence locations, for which the climate information is missing, act as the test point in cross-validation. The training points for the model are all the stations data that have information about the climatic attribute for that particular month.

4.3. Significance Test

We used significance test to check if estimated data was statistically significant or not. We used J48 decision tree classifier, a Java-implementation of the popular C4.5 algorithm [33]. It is implemented in RWeka (a R wrapper for Weka) [34].

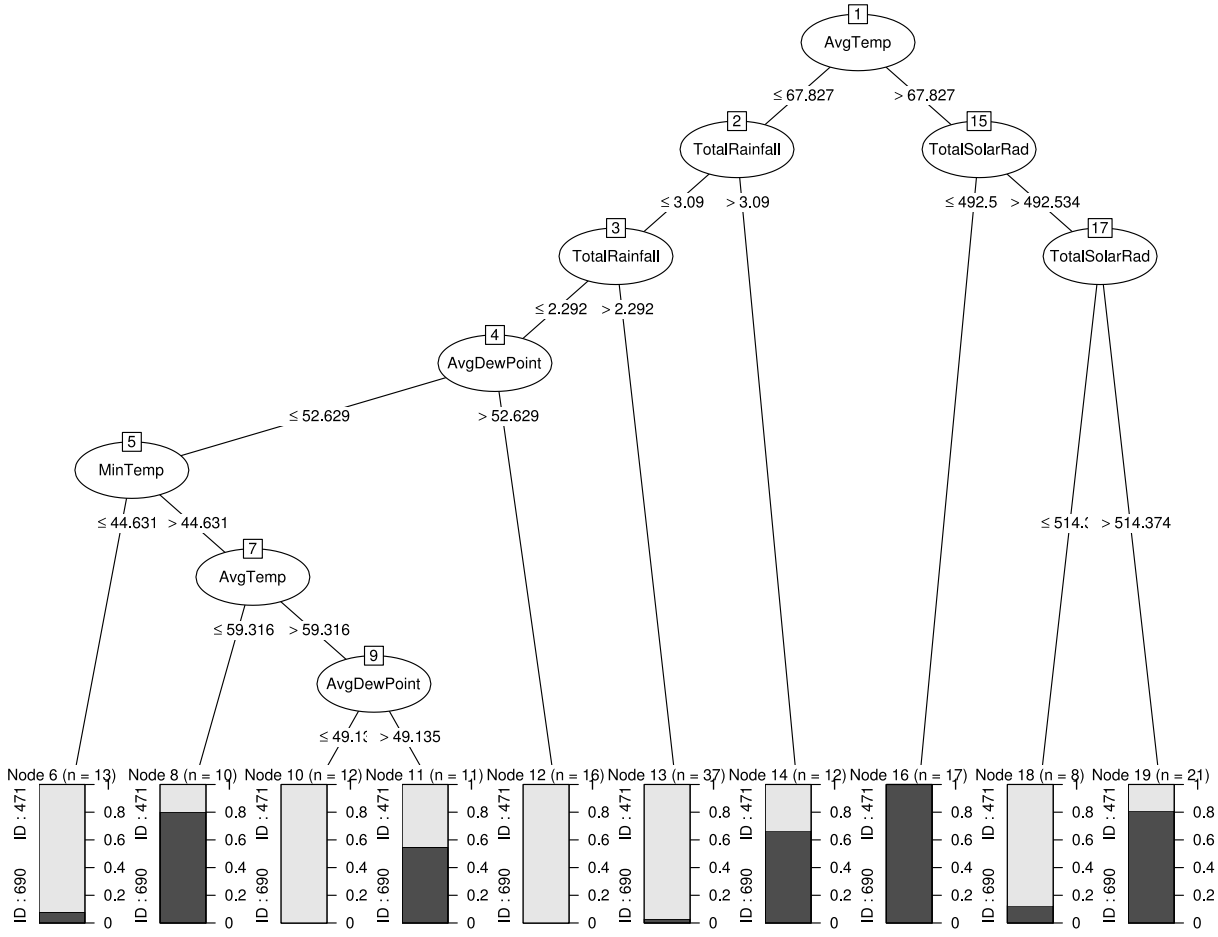

Figure 4-9: J48 Decision Tree on class labels and estimated climatic variables

The Figure 4-9 shows the decision tree generated by J48 for the estimated data. The nodes are the climatic attributes, and each node represents an instance of a climatic attributes that creates a branching for predicting the class labels (*River Jewelwing* or *American Rubyspot*). The higher the placement of the node, higher would be its role in the classification. As shown in the figure, temperature (avg. and max.), solar radiation (total), rainfall (total) and wind (avg.) attributes features as the important nodes.

We used 10-fold cross validation technique using Weka evaluator to evaluate the decision tree model. The results from the evaluation is shown in the following confusion table.

Table 4-4: Confusion matrix from cross-validation of J48 classifier

J48		Actual	
		Jewelwing	*Rubyspot*
Classified	*Jewelwing*	**84**	**23**
	Rubyspot	**14**	**36**

The contingency table, Table 4-4, shows the True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN) for the cross-validation of J48 classifier. The values for TP, FP, FN and TN are 84, 14, 23 and 36 respectively. This means that 84 samples were correctly predicted, and 23 samples were incorrectly predicted as *River Jewelwing*. Meanwhile, 36 samples were correctly predicted, and 14 samples were incorrectly predicted as *American Rubyspot*. We used fishers test to see the significance of the confusion matrix. We got a p-value of 0.000166307 from the Fisher's exact test [22]. This shows that the relationship of the species and climate attributes was quite significant.

We used J48 to find whether the relationship between the species and their climatic habitat is significant or not. We do not intend to use the attributes further. All attributes are treated equally in further steps in discretization as well as association studies.

4.4. Discretization

To apply ARM, we had to convert the estimated values of each climatic attributes from continuous form to discrete form (bins) [35]. We could discretize the values of each climatic attributes into binary or multi-level structure. We chose multi-level structure for the discretization

process because it shows a more realistic class labels for the range of value. For instance, converting to a binary structure would result in a high / low class labels whereas, converting to multi-level structure would result in very-low / low / low-mid / mid / mid-high / high / very-high and so on.

We tested few naive ways to find such clusters using techniques like mean, median and percentiles as split points. However, the results from K-means clustering made more sense as the number of clusters for each climatic attribute would be dependent on the spread of each such attributes. It also helped to segregate the outliers, the rare minimums and the rare maximums.

The K-means clustering method was based on an algorithm implemented in a R package, *Ckmeans.1d.dp* [13]. This algorithm is based on the concept of dynamic programming for finding optimal univariate clusters. It works by minimizing the sum of squares of the within-cluster distances, and guarantees optimality and reproducibility (for the given minimum and maximum number of clusters). Within-cluster distance means the distance of each element from its corresponding cluster mean.

The classical approach to K-means clustering involves fixing the number of clusters first, and then searching for the cluster members. The given library accepts the range of the size of the clusters (minimum and maximum), and finds optimal solution for the given range. We had to find the balance between number of clusters to avoid low support and finding enough confidence in the rules generated from ARM. If the number of clusters is low, then we would have good support, yet the confidence could be relatively lower. Similarly, if the number of clusters is high, we could end up with very less support. We settled for a minimum of 2 clusters and a maximum cluster of 9 based on experimental assumption.

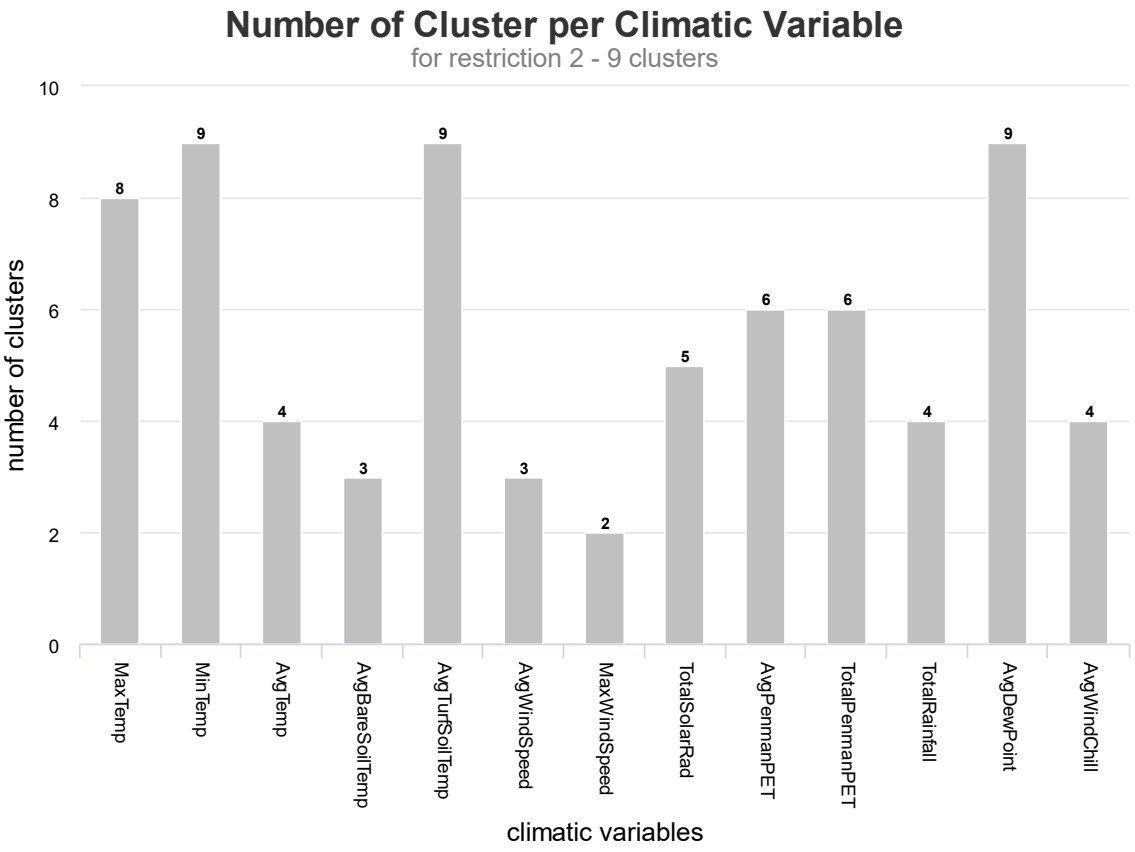

Figure 4-10: Number of clusters for each climatic attribute

The Figure 4-10 shows the number of clusters distribution for each of the climatic attributes. The number of clusters ranged from 2 to 9. The number of clusters for *MaxWindSpeed* was lowest at 2 clusters, and the number of clusters for *MinTemp*, *AvgTurfSoilTemp* and *AvgDewPoint* was highest at 9 clusters each. This is the optimal solution for restriction of cluster size varying from 2 to 9. Higher number of clusters suggest the values for those climatic attributes are more spread, and vice-versa.

Figure 4-11 shows the center for the clusters of the climatic attributes. The cluster center of *TotalSolarRadiation* has been removed in the diagram because of broad range. The Figure 4-12 shows the size of each clusters, i.e., the number of samples that belong to each cluster. As we can see the size of the individual clusters vary quite significantly. Some of the clusters have very small size, and these are mostly outlier values. The data that we have after the discretization process is the raw data for ARM.

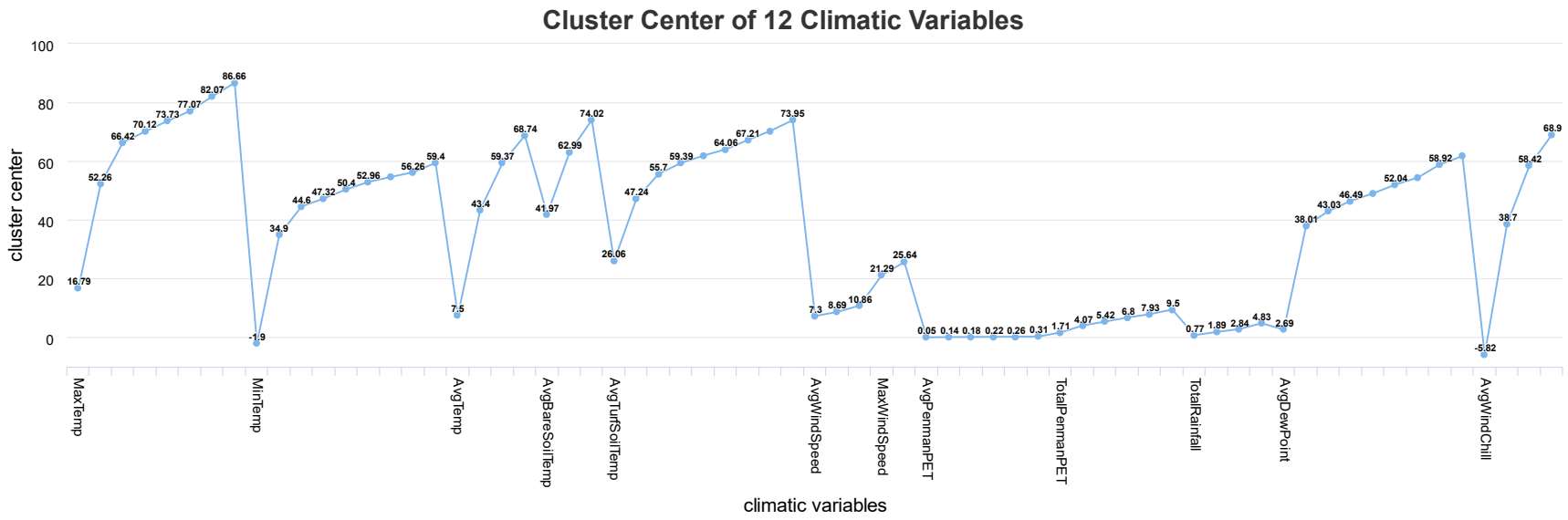

Figure 4-11: The cluster center for each clusters of various climatic attributes

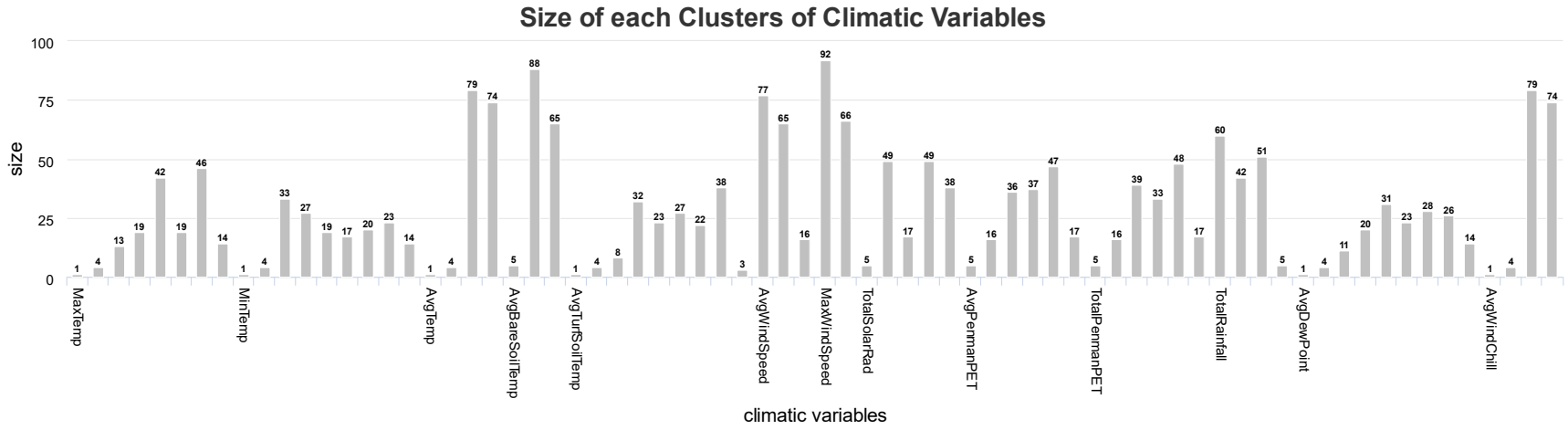

Figure 4-12: The number of samples in each cluster

4.5. Association Rules

The climatic information, a total of 13 attributes, along with the species class label are represented in matrix-like internal structure implemented internally within *arules* R package [14].

The idea with ARM was to find the climatic conditions (represented by numeric labels for each attributes) for both the damselflies and see if there exist notable differences in their estimated habitats. Even though the data set covered 13 climatic attributes, the number of items for the ARM problem increased to 74. This is because each cluster of all climatic attributes are now independent items and includes the two species. For example, *MinTemp=1* and *Mintemp=2* are completely different items for the ARM problem, even though they both represent Minimum Temperature. Similarly, *Species=Jewelwing* and *Species=Rubyspot* are the class labels and are also treated as items. We'll represent each item as an instance of a climatic attribute or a class label. One thing to note is that a sample (transaction) can only contain one instance of a climatic variable and one of the class label.

4.5.1. High Support Items

The first step was to find the high support instances of climatic attributes that were prevalent throughout the samples. The discretization process had uneven distribution of number of clusters. This results in discrepancy between the climatic attributes having less number of clusters and those having higher number of clusters. For instance, for those climatic attributes having only 2 clusters, the total of 158 samples would make it seem that these items are frequent in the general association concepts, as there are only 2 variations for these attributes. Thus, we need to normalize the support of these climatic variable configurations. The factor that affects this is the number of cluster for that climatic attributes. The normalized minimum support ($n_{cv}minsup$) for each climatic variable is defined as:

$$n_{cv}minsup\ (cv_i) = \frac{1}{2 * N(cv_i)}$$

where, cv_i is an instance of some climatic attributes and $N(cv_i)$ is the number of clusters of that climatic variable.

Table 4-5: Normalized Minimum Support for all Climatic Variables

Climatic Variable	Normalized Min. Support ($n_{cv}minsup$)
MaxTemp	0.063
MinTemp	0.056
AvgTemp	0.125
AvgWindChill	0.125
AvgDewPoint	0.056
AvgTurfSoilTemp	0.056
AvgBareSoilTemp	0.167
TotalSolarRad	0.100
AvgPenmanPET	0.083
TotalPenmanPET	0.083
TotalRainfall	0.125
AvgWindSpeed	0.167
MaxWindSpeed	0.250

The $n_{cv}minsup$ values are shown in Table 4-5. The lower $n_{cv}minsup$ indicates that the climatic variable was categorized into larger number of clusters, and vice-versa. *MinTemp*, *AvgDewPoint* and *AvgTurfSoilTemp* had lowest $n_{cv}minsup$ values with a value of 0.056 (9 clusters each). Meanwhile, *MaxWindSpeed* had the highest $n_{cv}minsup$ with a value of 0.250 (2 clusters).

Table 4-6 shows the top-15 high support clusters for the ARM problem. The support for these top-15 clusters range from 0.582 for *MaxWindSpeed=1* to 0.304 for *TotalPenmanPET=5*. Such large support is mostly due to relatively less number of clusters accommodated for the climatic variable during the discretization process. Most of these clusters are likely to feature as cross-support items in the association rules.

Table 4-6: High Support Items with additional information

Variable	Cluster #	Center	Low	High	Size	support
MaxWindSpeed	1	21.29	18.56	23.15	92	0.582
AvgBareSoilTemp	2	62.99	56.63	67.84	88	0.557
AvgWindChill	3	58.42	52.8	63.57	79	0.500
AvgTemp	3	59.37	54.35	63.94	79	0.500
AvgWindSpeed	1	7.3	6.02	7.92	77	0.487
AvgWindChill	4	68.9	64.81	74.54	74	0.468
AvgTemp	4	68.74	64.73	74.21	74	0.468
MaxWindSpeed	2	25.64	23.54	28.57	66	0.418
AvgBareSoilTemp	3	74.02	68.66	81.03	65	0.411
AvgWindSpeed	2	8.69	8.1	9.57	65	0.411
TotalRainfall	1	0.77	-1	1.27	60	0.380
TotalRainfall	3	2.84	2.43	3.48	51	0.323
TotalSolarRad	2	333.58	287.26	367.26	49	0.310
TotalSolarRad	4	496.16	458.46	536.9	49	0.310
TotalPenmanPET	5	7.93	7.39	8.59	48	0.304

Note: Additional information includes cluster identifier and parent cluster, center value, low and high range values, size, and support

4.5.2. Configuration

- n_{cv}*minsup*: cutoff for each climatic variable based on their discretization output
- species relative support: cutoff for each species based on number of samples
- confidence: fixed *confidence* of 0.7
- longer rules: *maximal* representation

4.5.3. Rule Representation

The green (octahedron) denotes *River Jewelwing* damselfly and red (octahedron) denotes *American Rubyspot* damselfly. The grey (triangle) denotes a rule in the association graph. The size of the triangle denotes the support of the rule. The grey edges from rule edges lead to a class label. The weight of the edges denotes the confidence of the rule.

The blue node (circle) denotes an instance of a climatic variable. The green node (circle) denotes an 'interesting' instance of a climatic variable for *River Jewelwing*, while the red node (circle) denotes an 'interesting' instance of a climatic variable for *American Rubyspot*. The edges from these nodes lead to a rule, and implies it is a part of the *lhs* of that rule.

The size of a circle node depends upon 3 factors, namely

- Support with the class label (*supp*),

  From single rules, the *support* of the rule featuring the given instance of climatic variable.

- Confidence with the class label (*conf*)

  From single rules, the *confidence* of the rule featuring the given instance of climatic variable.

- Parent Cluster Size (*pCS*)

  The *Parent Cluster Size* of the instance of climatic variable is used. For example, for *MaxTemp=1*, the *pCS* value would be the number of cluster for *MaxTemp* climatic variable, which is 8. In calculating the size of the node, logarithmic function (log) is used for the *pCS* value to reduce the effect if the value is on the higher side.

  We have single-class-label network, and the size (S) of node (n) for class label (p) is given by:

$$S(n) = supp(n \rightarrow p) * conf(n \rightarrow p) * log(pCS(n))$$

4.5.4. Single Rules

The second step was to find the individual instances of climatic attributes that were closely related to either of the species. This means finding association rules that had only 1 item in the *lhs* and one of the class label in the *rhs* of a rule. This allows us to identify high support climatic attribute instances for each of the species. Once we filter the association rules with higher confidence, it reflects that those climatic attribute instances are more favorable for either *River Jewelwing* or *American Rubyspot*. When the longer rules are generated, if some of these climatic attribute instances features in the rules then we can say that these attributes are more "*interesting*" than the other attributes.

We are looking for association rules in pattern of

$$[\text{A climatic attribute}] \rightarrow [Damselfly\ Species\ Type]$$

$$[An\ instance\ of\ Climatic\ Attribute] \rightarrow [River\ Jewelwing]\ OR\ [American\ Rubyspot]$$

4.5.4.1. River Jewelwing

Figure 4-13 shows that there were 17 climatic attribute instances for River Jewelwing that passed the $n_o minsup$ threshold of each respective climate attribute and 0.7 confidence. These 17 instances are the "interesting" items for River Jewelwing.

Table 4-7: Interesting rules for River Jewelwing that surpass relative support

lhs	rhs	support	confidence
[AvgTurfSoilTemp=4]	[River Jewelwing]	0.171	0.844
[MinTemp=3]	[River Jewelwing]	0.177	0.848
[MaxTemp=5]	[River Jewelwing]	0.190	0.714
[TotalSolarRad=2]	[River Jewelwing]	0.222	0.714
[TotalRainfall=3]	[River Jewelwing]	0.228	0.706
[AvgWindChill=3]	[River Jewelwing]	0.361	0.722
[AvgTemp=3]	[River Jewelwing]	0.361	0.722
[AvgBareSoilTemp=2]	[River Jewelwing]	0.418	0.750

Out of the 17 climatic attribute instances, 8 of the instances would have crossed the relative support (0.163) for River Jewelwing. These 8 instances are shown in Table 4-7. Later, in case of longer rules, these are the instances that may feature as they have enough support. The most notable instances include relatively lower classes of temperature related attributes like *MinTemp*, *MaxTemp*, *AvgTurfSoilTemp*, *AvgTemp*, *AvgBareSoilTemp* and *AvgWindChill*. The list also includes higher classes of *TotalRainFall* as well.

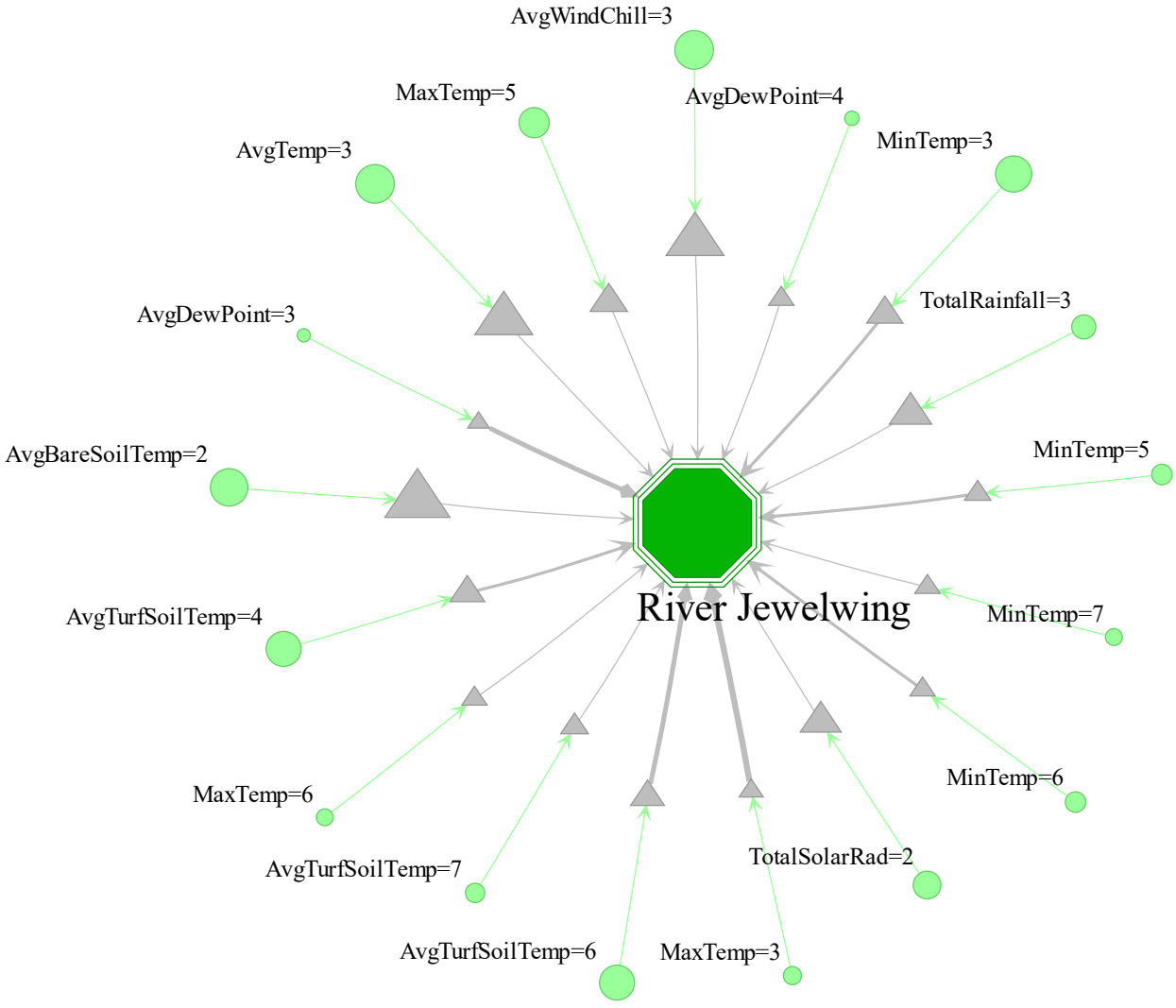

Figure 4-13: Network showing single rules for River Jewelwing

4.5.4.2. American Rubyspot

Figure 4-14 shows that there were 5 climatic attribute instances for American Rubyspot that passed the threshold of $n_{cv}minsup$ for each respective climatic attribute and 0.7 confidence. These 5 instances are the "interesting" items for American Rubyspot.

Table 4-8: Interesting rules for American Rubyspot that surpass relative support

lhs	rhs	support	confidence
[MinTemp=8]	[American Rubyspot]	0.114	0.783
[AvgTurfSoilTemp=8]	[American Rubyspot]	0.184	0.763

Out of the 5 climatic attribute instances, only 2 instances would have crossed the relative support (0.1) for American Rubyspot. These instances are shown in Table 4-8. These are the instances that may feature in the longer rules for American Rubyspot. The notable instances are temperature-related, and the table suggests higher classes of *MinTemp* and *AvgTurfSoilTemp*. We can also see *MaxTemp* and *AvgDewPoint* in Figure 4-14, which reflects habitat in the higher temperature.

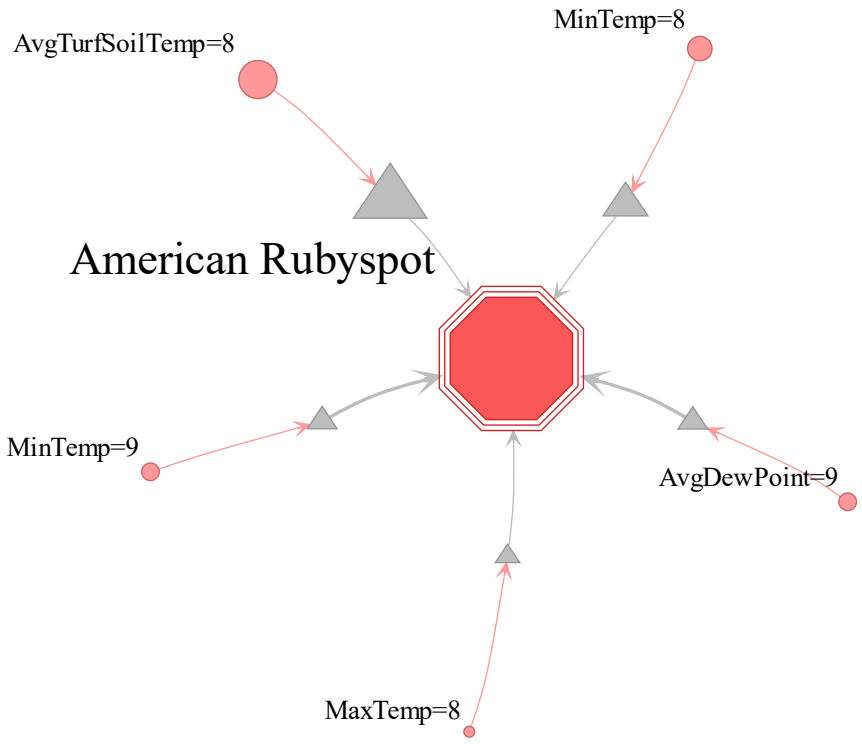

Figure 4-14: Network showing single rules for American Rubyspot

4.5.5. Longer Rules

In case of longer rules, we utilized maximal itemsets to find the association rules to avoid a lot of redundant rules. The relative support cut-off for each species was calculated and used for getting the rules. Since, the support for mining would be different, we either would have to look for maximal rules separately, or find all closed rules, and then use species-specific cutoff support and generate maximal rules from the filtered closed rules. We used the latter version.

We were looking for association rules in pattern of

$$[LHS\ items] \rightarrow [Damselfly\ Species\ Type]$$

$$[List\ of\ climatic\ attribiutes\ instances] \rightarrow [River\ Jewelwing]\ OR\ [American\ Rubyspot]$$

4.5.5.1. River Jewelwing

A total of 8 association rules were generated for River Jewelwing. The configuration for the rule mining for River Jewelwing is listed:

$$Support\ =\ 0.163\ ;Confidence\ =\ 0.7\ ;Min.Length\ =\ 3\ ;\ Max.Length\ =\ 9$$

Figure 4-15 shows the all the maximal rules for River Jewelwing. 8 'interesting' instances identified in the (single-rules) features in the longer rules. The temperature attributes (*AvgTemp=3*, *AvgWindChill=3* and *AvgBareSoilTemp=2*) are the top instances to feature in most rules, and mostly occurs together. Rule with [*AvgBareSoilTemp=2, Total*R*ainfall=3*] has the largest support of 0.196, and noticeably, we don't see any significant cross-support items in the rule.

Overall, we can see that the species is more associated with less degree of temperature, with climatic features such as lower ends of soil temperature and air temperature. Also, we see lower solar radiation (*TotalSolar*R*ad=2*), higher rainfall (*Rainfall=3*) and lower maximum temperature (*Maxtemp=5*).

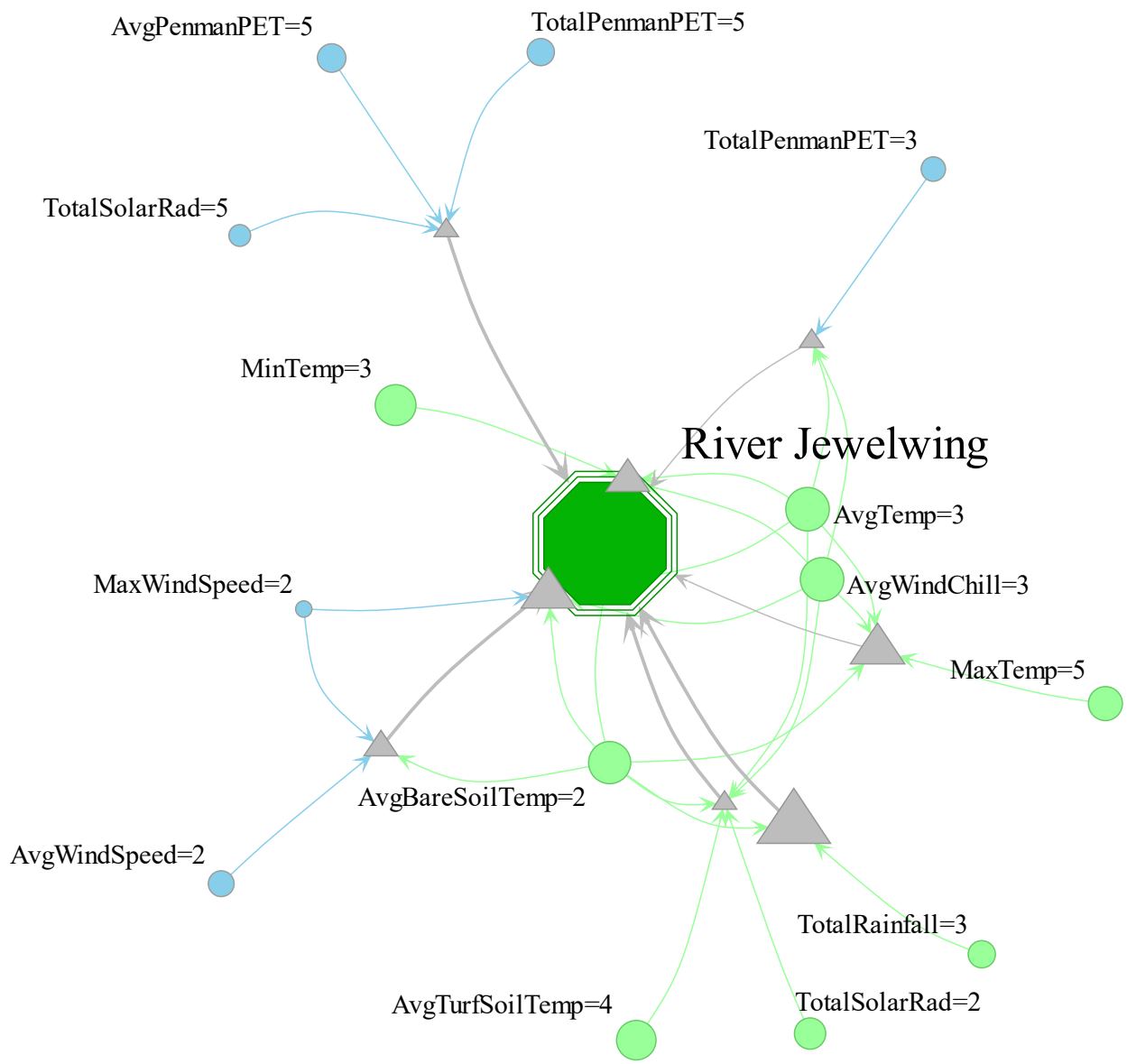

Figure 4-15: Network showing all maximal rules for River Jewelwing

4.5.5.2. American Rubyspot

A total of 4 association rules were generated with the given configuration. The configuration for the rule mining for River Jewelwing is listed:

$$Support = 0.1 ; Confidence = 0.7 ; Min.Length = 3 ; Max.Length = 9$$

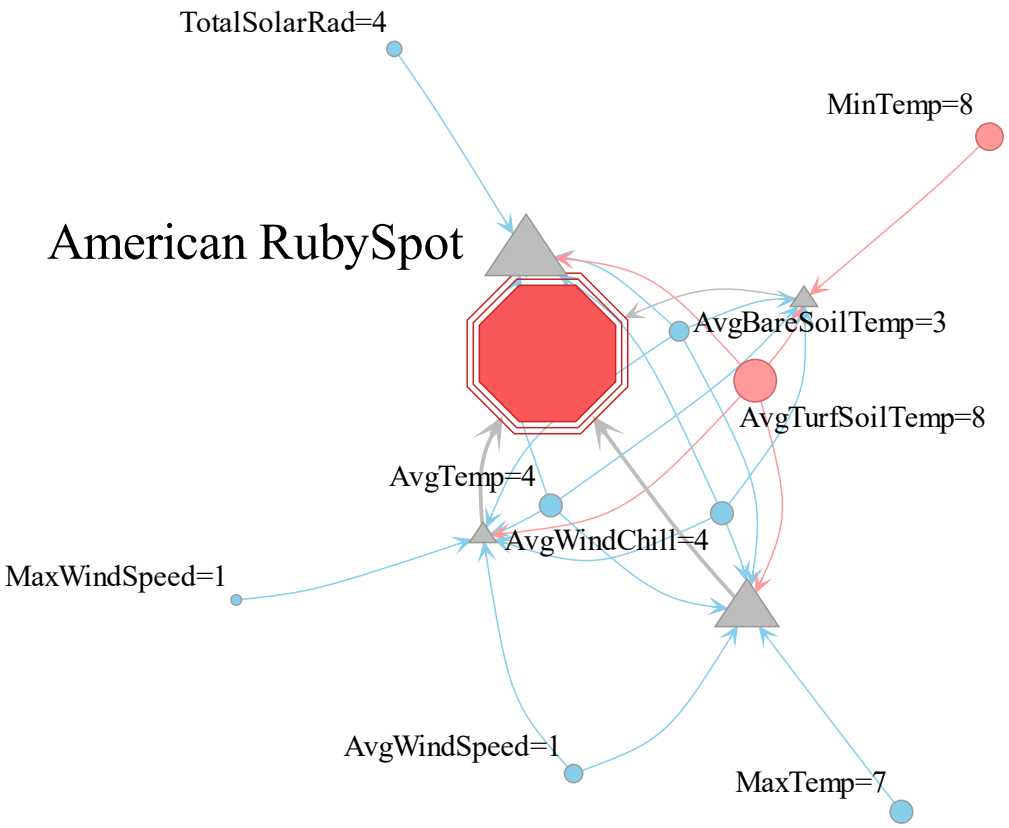

Figure 4-16: Network showing all maximal rules for American Rubyspot

Figure 4-16 shows the all the maximal rules for American Rubyspot. 2 'interesting' instances identified in the (single-rules) features in the longer rules. The temperature attributes (*AvgTurfSoilTemp=8*) is the top interesting instance and features in all the 4 rules.

Other attributes that feature in all the 4 rules include *AvgTemp=4*, *AvgWindChill=4* and *AvgBareSoilTemp=3*. These attributes are cross-support items. One notable distinction from Figure 4-10 is that all these 3 climatic attributes have less number of clusters. If we had more clusters for them, they could have been interesting item for American Rubyspot as well.

Overall, we can see that the species is more associated with higher degree of temperature with climatic features such as higher ends of soil temperature and air temperature and higher wind chill temperature. We also see higher solar radiation and lower wind speed, and missing rainfall.

5. CONCLUSIONS

Occurrence association between the genome variants in the available samples were successfully found with respect to the origin of the samples using ARM. The large data set was minimized using aggregation and filtered using graph pruning techniques and significance analysis. The two-phase ARM, single and maximal rules, helped to uncover the related genome variants along with the information of each individual genome variants for each class label. This information was successfully applied to graph network for visualization. The use of such visualization helped to recognize the interesting genome variants for each class label and separate out the cross-support genome variants. It also incorporated association quality measures like support and confidence for the association rules, and reflected the usefulness of each genome variants for the class label with relative node sizes.

Association of clusters of climatic attributes for the given species were successfully found using ARM. The species and their occurrence information (date and location) was enriched with climatic information by utilizing the available climatic data sets using geo-statistical models. The 2-phase ARM, single and maximal rules, helped to uncover the related climatic attribute clusters associated, individually and collectively respectively, to one of the species. This information was successfully applied to graph network for visualization. Such visualization helped to filter-out the cross-support climatic attribute clusters and recognize the co-occurring important attribute clusters. The importance of the clusters (nodes) with respect to the species was utilized by incorporating association properties like support and confidence, and node sizes based on its importance to the species.

In summary, two biological data sets were successfully formulated into ARM problems and were mined for associations. These associations were presented using graph networks to make easier interpretations of the association rules. Moreover, the use of 2-phase ARM identifies the importance

of the attributes to the class labels individually as well as collectively, and aids in determining size metrics for visualization. Overall, it has been demonstrated that ARM could be a useful to find associations of elements in biological data sets and an effective solution to represent such associations has been successfully presented.

REFERENCES

[1] J. Y. Chen and L. Stefano, "Biological Data Mining," in *Biological Data Mining*, CRC Press, 2009, p. IX.

[2] M. J. Zaki, J. T. L. Wang and H. T. T. Toivonen, "BIOKDD01: workshop on Data Mining in Bioinformatics," *ACM SIGKDD Explorations Newsletter,* pp. 71-73, Volume 3 Issue 2, January 2002.

[3] M. J. Zaki, J. T. Wang and H. T. Toivonen, "Data mining in bioinformatics: report on BIOKDD'03," *ACM SIGKDD Explorations Newsletter,* pp. 198-199, Volume 5 Issue 2, December 2003.

[4] F. Olaiya and A. B. Adeyemo, "Application of Data Mining Techniques in Weather Prediction and Climate Change Studies," *I.J. Information Engineering and Electronic Business,* pp. 51-59, 2012.

[5] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Magazine,* vol. 17, no. 3, pp. 37-54, 1996.

[6] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., San Francisco, CA: Morgan Kaufmann Publishers Inc, 2011.

[7] D. Kumar and D. Bhardwaj, "Rise of Data Mining: Current and Future Application Areas," *IJCSI International Journal of Computer Science Issues,* vol. 8, no. 5, pp. 256-260, 2011.

[8] R. Agrawal, T. Imieliński and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data,* pp. 207-216, 1993.

[9] J. Wang, "Encyclopedia of Data Warehousing and Mining," Information Science Reference, 2009, p. 59.

[10] G. Atluri, R. Gupta, G. Fang, G. Pandey, M. Steinbach and V. Kumar, "Association Analysis Techniques for Bioinformatics Problems," in *BICoB '09 Proceedings of the 1st International Conference on Bioinformatics and Computational Biology*, New Orleans, LA, 2009.

[11] H. Xiong, P.-N. Tan and V. Kumar, "Hyperclique pattern discovery," *Data Mining and Knowledge Discovery,* vol. 13, no. 2, pp. 219 - 242, 2006.

[12] V. Kumar, "Association Analysis: Basic Concepts and Algorithms," in *Introduction to Data Mining*, Addison-Wesley Companion Book Site , 2006, pp. 327-414.

[13] H. Wang and M. Song, "Ckmeans.1d.dp: Optimal k-means Clustering in One Dimension by Dynamic Programming," *The R journal,* vol. 3 2, pp. 29-33, 2011.

[14] M. Hahsler, B. Gruen and K. Hornik, "arules -- {A} Computational Environment for Mining," *Journal of Statistical Software,* vol. 14, pp. 1-25, October 2005.

[15] M. Hahsler and S. Chelluboina, "Visualizing Association Rules: Introduction to the R-extension Package arulesViz," *R project module,* pp. 223-238, 2011.

[16] B. Almende, B. Thieurmel and R. Titouan, "visNetwork," CRAN, 2017.

[17] L. Bélanger, A. Garenaux, J. Harel, M. Boulianne, E. Nadeau and C. Dozois, "Escherichia coli from animal reservoirs as a potential source of human extraintestinal pathogenic E. coli," *FEMS Immunol Med Microbiol,* pp. 1-10, 2011.

[18] T. Hussain, "An introduction to the Serotypes, Pathotypes and Phylotypes of Escherichia coli," *International Journal of Microbiology and Allied Sciences (IJOMAS),* pp. 9-16, August 2015.

[19] L. Rouli, V. Merhej, P.-E. Fournier and D. Raoult, "The bacterial pangenome as a new tool for analysing pathogenic bacteria," *New Microbes New Infect,* pp. 72-85, 2015.

[20] B. Segerman, "The genetic integrity of bacterial species: the core genome and the accessory genome, two different stories," *Front Cell Infect Microbiol,* pp. 2-116, 2012.

[21] C. Besemann, A. Denton and A. Yekkirala, "Differential Association Rule Mining for the Study of Protein-Protein Interaction Networks," in *BIOKDD04: 4th Workshop on Data Mining in Bioinformatics (with SIGKDD Conference)*, Seattle, WA, 2004.

[22] D. C. Howell, "Chi-Square Test - Analysis of Contingency Tables," *International Encyclopedia of Statistical Science,* pp. 250-252, 2001.

[23] NDAWN, "Data Information," 2017. [Online]. Available: https://ndawn.ndsu.nodak.edu/help-data.html#normvars. [Accessed 19 March 2017].

[24] J. Li, A Review of Spatial Interpolation Methods for Environmental Scientists., Geoscience Australia, 2008.

[25] M. Gimond, "Intro to GIS and Spatial Analysis," 2017. [Online]. Available: https://mgimond.github.io/Spatial/spatial-interpolation.html. [Accessed 20 March 2017].

[26] R. Sluiter, Interpolation methods for climate data, De Bilt: KNMI, 2009.

[27] E. Isaaks and R. Srivastava, "An introduction to applied geostatistics," Oxford University Press, USA, 1989, p. 592.

[28] E. Pebesma and B. Graeler, "gstat - Spatial and Spatio-Temporal Geostatistical Modelling, Prediction," CRAN, 2017.

[29] G. Bohling, "Kriging," 19 October 2005. [Online]. Available: http://people.ku.edu/~gbohling/cpe940/Kriging.pdf. [Accessed 22 March 2017].

[30] J. Shaw, Y. You, R. Haberman and D. Maidment, "Geostatistical Analysis," April 2009. [Online]. Available: http://www.ce.utexas.edu/prof/maidment/statwr2009/Ex9/Ex9.doc. [Accessed March 2017].

[31] P. Hiemstra, "automap - Automatic interpolation package," CRAN, 2013.

[32] C. Sammut and G. Webb, "Leave-One-Out Cross-Validation," in *Encyclopedia of Machine Learning*, Boston, MA, Springer US, 2010, pp. 600-601.

[33] K. Hornik, C. Buchta and A. Zeileis, "Open-source machine learning: R Meets Weka," Computational Statistics, 2009.

[34] K. Hornik, C. Buchta, T. Hothorn, A. Karatzoglou, D. Meyer and A. Zeileis, "RWeka : An R interface to Weka (Version 3.9.1)," CRAN, 2017.

[35] M. Vannucci and V. Colla, "Meaningful discretization of continuous features for association rules mining by means of a SOM," in *ESANN'2004 proceedings - European Symposium on Artificial Neural Networks*, Bruges (Belgium), 28-30 April 2004.