# IMPROVING DATA EXTRACTION METHODS FOR LARGE MOLECULAR BIOLOGY DATASETS

by

Robert William Reid

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2010

Approved by:

_____
Dr. Anthony Fodor

_____
Dr. XiuXia Du

_____
Dr. Julie Goodliffe

_____
Dr. ZhengChang Su

_____
Dr. Jennifer Weller

# ABSTRACT

ROBERT WILLIAM REID. Improving data extraction methods for large molecular biology datasets. (Under direction of DR. ANTHONY A. FODOR)

In the past, an experiment involving a pair wise comparison normally involved one or a few dependant variables. Now, 1000s of dependent variables can be measured simultaneously in a single experiment, be it detecting genes via a microarray experiment, sequencing genomes, or detecting microbial species based on DNA fragments using molecular techniques. How we analyze such large collections of data will be a major scientific focus over the next decade. Statistical methods that were once acceptable for comparing a few conditions are being revised to handle 1000's of experiments. Molecular biology techniques that explored 1 gene or species have evolved and are now capable of generating complex datasets requiring new strategies and ways of thinking in order to discover biologically meaningful results. The central theme of this dissertation is to develop strategies that deal with a number of issues that are present in these large scale datasets. In chapter 1, I describe a microarray analytical method that can be applied to low replicate experiments. In chapter's 2-4, the focus is how to best analyze data from ARISA (a PCR based molecular method for rapidly generating a finger print of microbial diversity). Chapter 2 focuses on qualifying ARISA data so that data will best represent its biological source, prior to further analysis. Chapter 3 focuses on how to best compare ARISA profiles to one another. Chapter 4 focuses on developing a software tool that implements the data processing and clustering strategies from chapter's 2 and 3. The findings described herein provide the scientific community with improved analytical strategies in both the microarray and ARISA research areas.

TABLE OF CONTENTS

# TABLE OF FIGURES

# SYNOPSIS

Microarray analysis often involved comparing multiple arrays between some control state and some experimental condition. In instances where the numbers of replicate arrays are low (i.e., less than 3 replicates), the analytical options for analysis are often limited. In chapter 1, we explored the idea that it should in principle be possible to use the high number of probes in each probe set of a microarray experiment to substitute for the lack repeat experiments. That is, instead of using repeated chips to estimate the variance for statistical inference, we exploited the existence of multiple probes per probe set to estimate variance, thus making it possible to analyze low sample size experiments. Previously, Hein and Richardson used a Bayesian hierarchical model (called BGX) that estimates gene expression levels from probe level data. Their model enabled comparisons between single chip to chip comparisons (i.e., N=1 in each condition) [1]. They compared the BGX algorithm to other available methods and demonstrated an increase in performance. However, their algorithm is computationally demanding and it appeared that a better performance could be achieved with less computational demand. As an alternative, we described an algorithm called PINC (PINC is not Cyber-T) based on the Cyber-T algorithm first described by Bali and Long[2] and a method we recently described for generating accurate p-values[3]. We found that PINC has attractive characteristics when compared to BGX, Cyber-T, and other analytical methods when inferring gene expression on Affymetrix microarrays at low sample sizes.

Microbial environments in nature are much more diverse and complex than was thought even a decade ago [4]. Understanding the nature of microbial environments has

often been limited only to species that can be cultured, which may be as low as 1% of the species in any given population [5,6]. To gain further understanding of an entire microbial community, molecular biology techniques were developed that exploit the conserved nature of ribosomal DNA (see [7] for a review). One such technique, ARISA (Automated ribosomal intergenic spacer analysis) attempts to identify microbial species by determining the sizes of the intergenic DNA fragments between adjacent 16S and 23S ribosomal genes [8]. An ARISA experiment yields a dataset consisting of many data peaks, derived from fluorescent signal, which correspond with DNA fragments of varying sizes. Our primary goal here was to develop data processing and quality control methods that assess how well ARISA datasets correspond to known size standards. By comparing peaks to known size standards, distinguishing peaks from baseline signal, and identifying poor experiments, we were able to accurately estimate DNA fragment size and produce cleaner ARISA datasets that were more amenable to cluster analysis.

ARISA can be used as a tool for comparing microbial communities by determining the number and size of DNA base pair lengths in a dataset and comparing these "fingerprints" to other ARISA experiments [8]. A number of methods have been developed to optimize how these comparisons are made, however to date, no rigorous examination of all the current methods has been performed. In chapter 3, a number of methods described in the literature were implemented and compared using various parameters. The clustering methods were applied to a collection of ARISA experiments that examined the composition of microbial communities in the human gut over a 60 day time course. Fifteen subjects were placed on a strictly controlled diet and microbial community composition was determined by both ARISA and by 16S DNA sequencing.

The result of 16S DNA sequencing showed that microbial environments perfectly cluster by subject over the course of the trial. We then used ARISA results to test different clustering strategies to see which parameters would best mirror DNA sequencing. Performance was assessed by comparing cluster trees from ARISA to the DNA sequencing tree cluster. The findings from chapter 3 show that the current methods in the literature fail to perform any better than what would be expected from random chance. The more critical parameter that affects clustering performance is the choice of clustering method. We show that using the nearest neighbor linkage method fails to correctly cluster ARISA compared to Ward's and furthest neighbor linkage methods. Overall, most of the parameters one can choose when ARISA clustering have a negligible effect on clustering performance, with the exception of nearest neighbor linkage, which adversely affects performance.

There is currently a lack of a user friendly software specifically designed for visualizing and analyzing data from molecular fingerprinting techniques such as ARISA. The purpose of chapter 4 was to design and implement an open source software package that will provide biologists and ecologists a tool to simplify the microbial analysis of ribosomal genes. The software tool, Peak Studio, was primarily written in Java by Jon McCafferty and it provides a graphical users interface (GUI) allowing anyone to quickly visualize either ARISA or TRFLP electropherograms. The software is able to perform quality control checks (that were developed in chapter 2) on ARISA datasets so that poor electropherograms can be flagged and removed. Peak Studio also implements all of the cluster comparison methods discussed in chapter 3 resulting in 112 different possible analytical combinations. The Peak Studio project was a team project with members

focusing on different areas of the software package. My role included contributing to the lead design of the software package, implementation of chapter's 2 and 3 into the software tool and the visualization of the tree clusters.

# CHAPTER 1: MICROARRAY ANALYSIS OF SINGLE EXPERIMENTS

## 1.1 Background and significance

In the past decade, there has been an explosion in the technology and understanding of microarray research. Since their introduction[9], microarrays originally promised to be a paradigm shifting research method that allowed a user to simultaneously determine global gene expression in context of a variety of biological scenarios. To an extent microarrays have been able to generate massive quantities of data on gene expression and have been instrumental in guiding research. However, a number of issues in microarray technology do exist including high background noise[10,11], signal inconsistencies[12], secondary structure probe issues[13] and the lack of agreement in the results obtained in different array platforms[14]. One particular issue that arises when interpreting microarray results is that the majority of statistical methods require $N \geq 3$ in each condition to meet the method requirements. However, it is not always possible to obtain this sample size. Reasons for small sample sizes include the expense of microarrays, experimental imperfections such as poor hybridization [15] and limited quantities of available biological sample source. In instances where conditions limit experiments to a single treatment versus control result, there are fewer analytical options for generating robust differential gene expression lists. Even when there 2 replicates of each condition ($N = 2$), the methods remain limited in applicability.

One strategy to overcome the lack of replicates is to exploit the presence of multiple probes for each gene present on some types of chips, treating each probe as an independent measurement. On an Affymetrix expression array, such as the HG-U133A GeneChip, each gene is represented on the array by a number of distinct 25 mer probes that correspond to different parts of the gene sequence. Many popular statistical methods including MAS5[16], RMA[17] and GCRMA [18] aggregate these 25 probes into a single summarized value for the entire probe set before performing statistical inference. Using such a single value still requires that there be results for multiple samples since the method requirement of ($N \geq 3$), remains. There are a number of models that directly utilize the measurements from the individual probes rather than summarizing values at the probe set level. Logit-T [19], Fisher's combined p-value [20], gMOS [21], and multi-mgMOS [22] all perform inferences on probe measurements, rather than with the summarized probe set values; however these methods still require multiple experiments ($N \geq 3$) in each condition.

We explored the idea that it should, in principle, be possible to use the high number of probes in each probe set to substitute for repeat experiments. That is, instead of using repeated chips to estimate the variance for statistical inference, can we exploit the existence of multiple probes per probe set to estimate the variance? Previously, Hein et al. have used a Bayesian hierarchical model to estimate expression levels using this same probe level approach, allowing for analysis with n=1 in each condition [15]. In their algorithm, called BGX, inference is performed at each stage of analysis (background correction, gene expression estimation and differential expression) [15]. Because the BGX algorithm requires a Markov chain Monte Carlo (MCMC) model at each stage of

microarray analysis, it is a very computationally demanding technique. As an alternative, we developed an algorithm called PINC (PINC Is Not Cyber-T), based on the Cyber-T algorithm first described by Baldi and Long [23] and a method we recently described for generating accurate p-values [24]. We show that PINC has attractive characteristics when compared to Cyber-T, BGX and other methods of performing inference on Affymetrix microarrays at low sample sizes.

### 1.1.1 Merits

The merit of this research is that it provides investigators a superior option for analyzing microarray experiments when there are low numbers of replicates. Such sets of experiments are unable to be analyzed via the more popular statistical methods. Often when generating a ranked list of genes, it is difficult to define a cutoff level to determine which genes are truly showing a change of expression. With PINC, low replicate experiments can yield a ranked list of differentiated genes with a predicted probability of being significant. Such lists are valuable for guiding investigators in choosing what genes to pursue in the lab.

A second benefit is that PINC can be used to estimate chip variability when there are multiple experiments. When there are multiple microarray chips used in an experiment, PINC can be applied repeatedly to compare chips to one another, generating ranked gene lists in each comparison. These gene lists can then be used as indicators of variability within the entire experiment as well as identify gene candidates that show consistent levels of expression by appearing on each list.

A third benefit of this work is that PINC provides a measure of variability between technical replicates, enabling one to identify when a set of technical replicates

fails to be consistent. Klebanov and Yakovlev showed that noise derived from technical replicates is generally low [25]. By comparing technical replicates to one another using PINC, inconsistent technical replicates can be identified.

### 1.1.2 Publishing Summary

Chapter 1 was completed in the spring of 2008 and accepted for publication in the fall of 2008 at BMC Bioinformatics. The publication can be found at:


- http://www.biomedcentral.com/1471-2105/9/489
- *BMC Bioinformatics* 2008, **9:**489
- doi:10.1186/1471-2105-9-489


Since first being published online on November 21, 2008, the paper has received a "highly accessed" tag on the BMC Bioinformatics website with over 1373 views.

## 1.2 Methods and materials

### 1.2.1 PINC Details

PINC harnesses Cyber-T, an algorithm that utilizes a Bayesian probabilistic framework to model log-expression values by averaging the canonical variance with a local background variance estimated from genes with similar intensities on the array [23]. The Cyber-T test can be applied to either paired or un-paired samples. The numerator of the Cyber-T test statistic is the same as in a Standard-T test. The denominator, however, has a correction for the local background variance. For example, an unpaired Standard-T test is calculated by:

$$\text{T Test} = \frac{-(m_1 - m_2)}{\sqrt{\frac{(n_1 - 1) * SD_1^{\,2} + (n_2 - 1) * SD_2^{\,2}}{(n_1 + n_2 - 2) * \left(\frac{n_1 + n_2}{n_1 n_2}\right)}}} \qquad (1.1)$$

where $n_1$ is the number of samples in condition 1, $n_2$ is the number of samples in condition 2, $m_1$ and $m_2$ are the means of samples 1 and 2 and $SD_1$ and $SD_2$ are the standard deviation for samples 1 and 2. What distinguishes the Cyber-T test from a Standard-T test for unequal sample size is that the standard deviations for samples 1 and 2 are not given by the canonical formula for standard deviation but rather are given by:

$$SD_{\text{Cyber-T}} = \sqrt{\frac{\text{Conf} * SD_{\text{Window}}^{\,2} + (n - 1) * SD^2}{\text{Conf} + n - 2}} \qquad (1.2)$$

where n is the sample size (the number of arrays in the condition), SD is the standard deviation as it is usually calculated, $SD_{\text{Window}}$ is the average of the standard deviation of the 100 genes with the average intensity closest to the average intensity of the gene under consideration and Conf is an adjustable parameter set to 10 by default in the "v1.0beta" of the Cyber-T distribution for R (http://cybert.microarray.ics.uci.edu). In a single chip treatment versus control experiment, $n_1$ and $n_2$ are equal to the number of probes for a particular gene and $m_1$ and $m_2$ are the averages of each group of probes.

For Affymetrix arrays, Cyber-T is usually used following summation of the probes into a single value for each probeset with an algorithm such as RMA[17] (Examples can be seen in [26], [27]). As an alternative, PINC applies Cyber-T directly to probes within a probeset to determine gene expression scores. For a GeneChip such as the Affymetrix HG-U133A Array, each probe set contains 11 perfect match probes (we ignore mismatch probes). Thus for a single chip experiment (treatment versus control) PINC compares 11 probes in each position using the paired Cyber-T test (with n = 11).

The Cyber-T test generates a p-value for each gene, evaluating the null hypothesis that the gene expression is identical in both conditions. Because the estimate for the variance of each gene's expression measurements is not independent but is instead dependent upon its neighboring gene scores, the authors of the Cyber-T do not expect the Cyber-T test to follow a simple t-distribution with n1+n2-2 degrees of freedom. Instead, the Cyber-T test assumes that Cyber-T scores will follow a t-distribution with 2 * Conf+n1 +n2 –2 degrees of freedom. We have previously shown that the p-values generated in this way are not very accurate [24].

To determine which genes are differentially expressed, PINC determines p-values by way of "Scheme 4" [24]. Scheme 4 assumes that all the test statistics form a single normal distribution and then applies a "Statistical Level Normalization" step which corrects for systematic drift in the t-statistic away from a value of zero [24].

In summary, PINC takes the scores from the paired Cyber-T test at the probe level and uses "Scheme 4" to calculate the p-values rather than using the p-values reported by the Cyber-T software. In this paper, we refer to "Cyber-T" and "Cyber-T paired" as

methods that act on the probe level but do not implement Scheme 4 to generate p-values. In our study, PINC is the only algorithm that has p-values generated by Scheme 4.

1.2.2 FDR and Family-Wise Error Rate algorithms

For the purposes of this analysis, we determined which genes were differentially expressed by either applying a 10% cut off rate via false discovery rates or performed multiple experiment correction via Holm's step down method [28] (p-value cutoff = 0.05).

The Benjamini and Hochberg algorithm (hereafter BH FDR) [29] yields a predicted False Discovery Rate (FDR) for a given gene in a gene list ordered by statistic p-value:

$$N*p(k) / k \tag{1.3}$$

where N is the number of genes in the list and p(k) is the p-value produced by the test statistic under the null hypothesis of no differential expression for gene k in the list. The more conservative Benjamini and Yekutieli FDR algorithm [30] (hereafter BY FDR) relaxes the assumption that the intensities of the genes on the array are independent. The BY FDR for a given gene k in a list of N genes is:

$$\sum_{i=1}^{N} \frac{1}{i} * N*p(k) * \ / k \tag{1.3}$$

### 1.2.3 Other statistical tests

At the probe level, we applied the student's Standard-T test (paired and unpaired), and Wilcoxon Rank Sum test. The BGX algorithm [15] was also applied to the different datasets as a benchmark comparison.

For the Cyber-T and BGX calculations we used an implementation in R from the Bioconductor package. All other statistical tests were implemented in Java (code available at http://www.afodor.net). Results for the Wilcoxon nonparametric test were generated from Java source code made publicly available by D. A. Nix (http://rana.lbl.gov/~nix).

### 1.2.4 Datasets

To assess the effectiveness of PINC, the HG-U133A Latin Square dataset was downloaded from Affymetrix [21]. Two Probe sets with a number of probes other than 11 probes were discarded. For the Latin Square data sets, probesets 209374_s_at, 205397_x_at and 208010_s_at were excluded for all analyses as instructed by the HG-U133A_tag_Latin_Square.xls spreadsheet. We also excluded any probeset not in the spike-in probesets that started with AFFX-. This resulted in 42 true positives and 22,181 true negatives used for assessing effectiveness. The Affymetrix Latin Square dataset was analyzed using N=1 for all 14 2X fold conditions taking the first experiment (i.e., the CEL file ending in R1) for each condition. For the multiple experiment comparisons in Figure 1-5 (when N > 1) probe values were averaged into a single consensus value and then analyzed. CEL files from all datasets were normalized using quantile normalization from dCHIP [31] (except for the BGX algorithm which performs its own normalization).

## 1.3 Results and discussion

### 1.3.1 The Performance of Test Statistics in ranking genes on a control data set at n=1.

On the Affymetrix Latin Square HG-133A dataset, there are 11 probes per probeset. Given 11 independent measures in two samples, there are a variety of statistical tests available to evaluate the null hypothesis for each gene that the expression observed in each sample is identical. These include the Standard-T test, a paired t test (which is equivalent to a two way ANOVA in which the independent variables are probe and sample) and the Wilcoxon test (a non-parametric equivalent to a paired-T test). In addition to these canonical statistical tests, there are variants of the t-test specifically designed for microarrays. These include the paired and unpaired Cyber-T tests [23] in which the variance for each gene is an estimate based on an average of the canonical variance for that gene and a background variance of other genes with similar intensities on each array (see methods).

We applied these different statistical measures to the Affymetrix Latin Square HG-133A dataset, which consists of 14 conditions of 3 replicates each. Each condition has 42 known genes spiked in at different concentrations that are true positives while the remaining 22,181 probe sets on the chip are true negatives. We examined the first replicate from each of the 14 experiments and compared experiments where there is a 2-fold change in spiked in concentration resulting in 13 separate comparisons (Exp 1 vs. Exp 2, Exp 2 vs. Exp 3, etc.). Applying the test statistics to these datasets yields for each statistic a gene list ranked according to the calculated scores. For each of these 13 comparisons, we can generate an ROC curve of the number of true positives [11] versus false positives (FP) at each possible cutoff for these gene lists with n=1 in each condition.

Figure 1.1A shows the average of these 13 ROC curves in which the x-axis displays all 22,181 true negatives. At this scale, it is immediately obvious that the BGX and Wilcoxon tests underperform the other statistics. The differences between the other statistics are more subtle with perhaps a slight advantage going towards the unpaired Cyber-T test.

While the data in Figure 1.1A give a broad overview of how the algorithms perform, the scale of the x-axis does not represent a biologically useful signal. For example, at a false positive rate of 0.05, where the unpaired Cyber-T test has a slight advantage over the other test statistics, a gene list for the HG-133A microarray would have over 1,000 false positives. Clearly such a gene list is not that useful. To better explore a more biologically relevant cutoff, in which a gene list consists of mostly true positives, Figure 1.1B shows the same data as in Figure 1.1A, but with the x-axis scaled to show only gene lists that include a small number of false positives. Figure 1.1C shows the number of true positives captured at a cutoff of n=4 false positives (Figure 1.1B dashed vertical line) for all 13 comparisons. At this more stringent cutoff the paired and unpaired Cyber-T tests clearly outperform the other statistics.

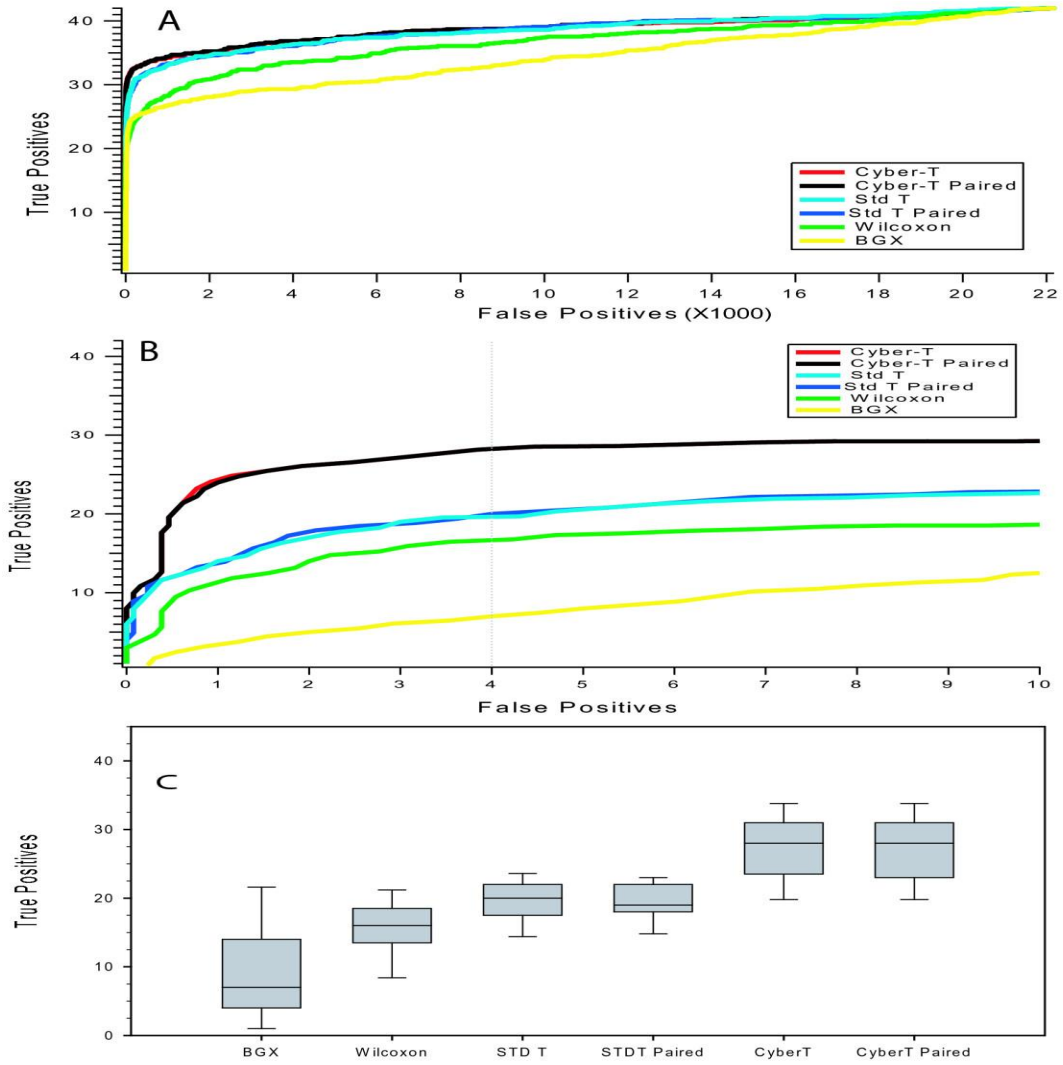



FIGURE 1.1: Average ROC curves for 13 Latin Square experiments. The performance of ranking true and false positives for pairs of N=1 experiments are depicted. The first experiment from 13 2X Latin Square experiments was selected for analysis. For each of the 13 comparisons, an ROC curve was generated. Shown is the average of all 13 ROC curves. Figure 1.1A shows the full-scale performance for all false positives. Figure 1.1B is a zoomed in view of 1.1A with the x and y-axes zoomed to show detail of restrictive cutoffs with few false positives. Figure 1.1C is a box plot of the number of TP detected at an arbitrary cut off level of 4 FP (vertical dashed line in 1B).

### 1.3.2 The Performance of Test Statistics in Providing Accurate p-Values for Inference

ROC curves rank all of the genes in an experiment but generating a gene list in a "real" experiment also requires choosing a cutoff point. That is, it is not enough to rank genes into an ordered list, one must know how many genes to consider significant from the list; each test statistic generates a score for each gene and we wish to determine the threshold score above which genes are considered to be significantly differentially expressed. This has proven to be a challenging problem [32]. In the microarray literature it is generally accepted that family-wise error rates, such as Bonferroni correction, are too conservative in an effort to prevent type-I errors thereby producing an abundance of type-II error [33], [34]. The use of false discovery rates (FDR) has become a popular alternative for controlling error rates (for a review, see [33]) . However, the use of false discovery rates has not been without controversy [35].

In this study, we evaluated the performance of different test statistics using two different FDR cutoff levels described by Benjamini et al. [29] (see methods), as well as the Holm's step down method, a more conservative family wise error rate correction algorithm ([28] and see methods). For the FDR algorithms, we set the cutoff level at 10%, i.e., we are willing to accept that 10% of the genes considered to be significant will be false positives. For the Holm's step down FWER, we set a cutoff level of 0.05 divided by N (22,223) for the highest scoring gene pair. Then for each subsequent gene, the cutoff is recalculated as 0.05 divided by the number of remaining genes. Figure 1.2 shows the sensitivity and specificity for the 13 n=1 comparisons we performed on the Latin Square dataset for p-values produced by various methods under a 10% BH and BY FDR cutoff and a 0.05 Holmes step down cutoff. We define sensitivity as the number of

true positives recovered at each threshold divided by the total number of true positives in the Latin Square data set. We define specificity as the number of true positives recovered at each threshold divided by the total number of genes above the threshold cutoff. An algorithm that generates p-values that are too large would be inappropriately conservative and not consider enough genes significantly differentially expressed. Such an algorithm would yield results with poor sensitivity but high specificity. Under all 3 cutoff schemes, this describes the Wilcoxon non-parametric test, which failed to detect any genes above our cutoff threshold (sensitivity = 0) and is therefore not included in Figure 1.2 or in further analyses.

Because of the poor performance (Figure 1) and high computational cost of the BGX algorithm, it too was not included in this analysis. Of the remaining algorithms, we see that the unpaired Cyber-T and paired and unpaired Standard-T tests also produce p-values that are too large as they yield nearly perfect specificity but poor sensitivity. By contrast, an algorithm that produces p-values that are too small will yield results with high sensitivity but poor specificity. We see that under the BH and BY FDR schemes, this describes the paired Cyber-T test; with a 10% FDR threshold, we would expect a specificity of 0.9 (red lines in Figure 1.2). While the paired Cyber-T test is able to detect a large number of the true positives (highest sensitivity), it also incorrectly detects numerous false positives, resulting in a specificity measure well below the expected level of 0.9. We can say therefore that the paired Cyber-T test has failed to control false discovery rate under BH and BY FDR.

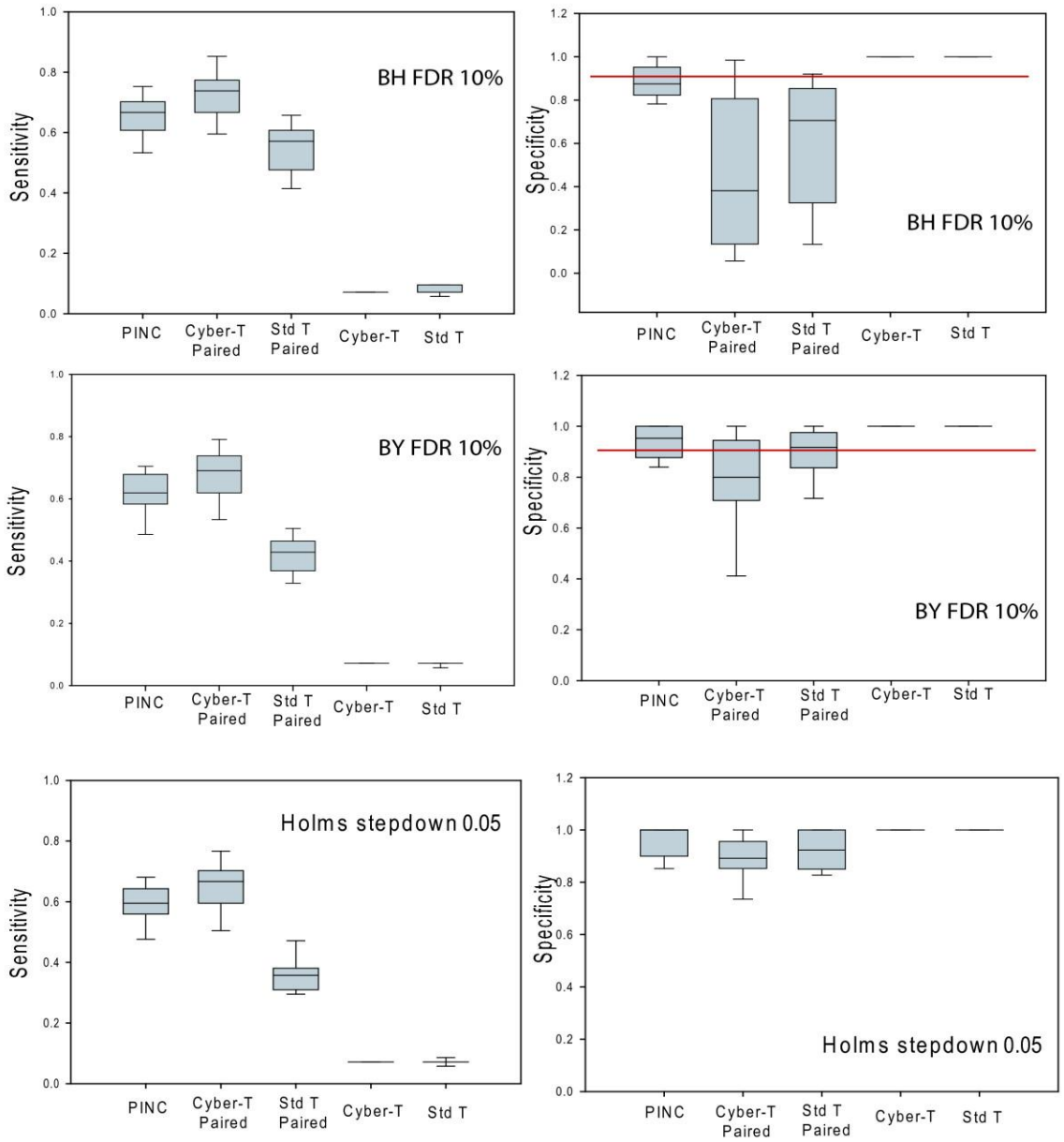



FIGURE 1.2: Sensitivity and specificity for different algorithms applied to the 13 N=1 2X Comparisons from the Latin Square dataset. Left panels are sensitivity scores at different p-value cut off levels and panels on the right are specificity scores. The red lines in the top 2 right panels represent the predicted FDR cutoff value at 10%.

We have previously shown that, when applied at the probeset level, p-values produced by canonical statistics and the unpaired Cyber-T test are not very accurate on control Affymetrix datasets [24]. We proposed as a simple alternative, a method that assumes that all the background values on a microarray form a single distribution ([24] and see methods). We describe a new algorithm PINC (PINC Is Not Cyber-T), which is the paired Cyber-T test performed at the probe level in which the p-values provided by the Cyber-T test are replaced with p-values generated by this assumption of a single background distribution. Applying the PINC algorithm yields a list in which the rank order is identical to the paired Cyber-T test (and therefore would have the same ROC profile in Figure 1) but the p-values differ. In Figure 1.2, we see that p-values generated by PINC do a better job of controlling FDR under both BH and BY FDR; the sensitivity of PINC is nearly as good as the sensitivity shown by Cyber-T paired, but the specificity is much closer to the expected level of 0.9. Indeed, no matter which of the three cutoff schemes we used to determine the threshold p-value of significance, the PINC algorithm nicely balanced sensitivity and specificity, picking up a substantial fraction of true positives with a minimal number of false positives (Figure 1.2). All other algorithms perform poorly on either sensitivity or specificity suggesting that p-values calculated with these algorithms are either inappropriately large or inappropriately small. We conclude that when compared to other algorithms, the p-values produced by the PINC algorithm lead to inference that is less susceptible to bias introduced by the method of determining the threshold cutoff. That is, we argue that the p-values produced by PINC are more robust than p-values produced by the Cyber-T software or by canonical statistical tests.

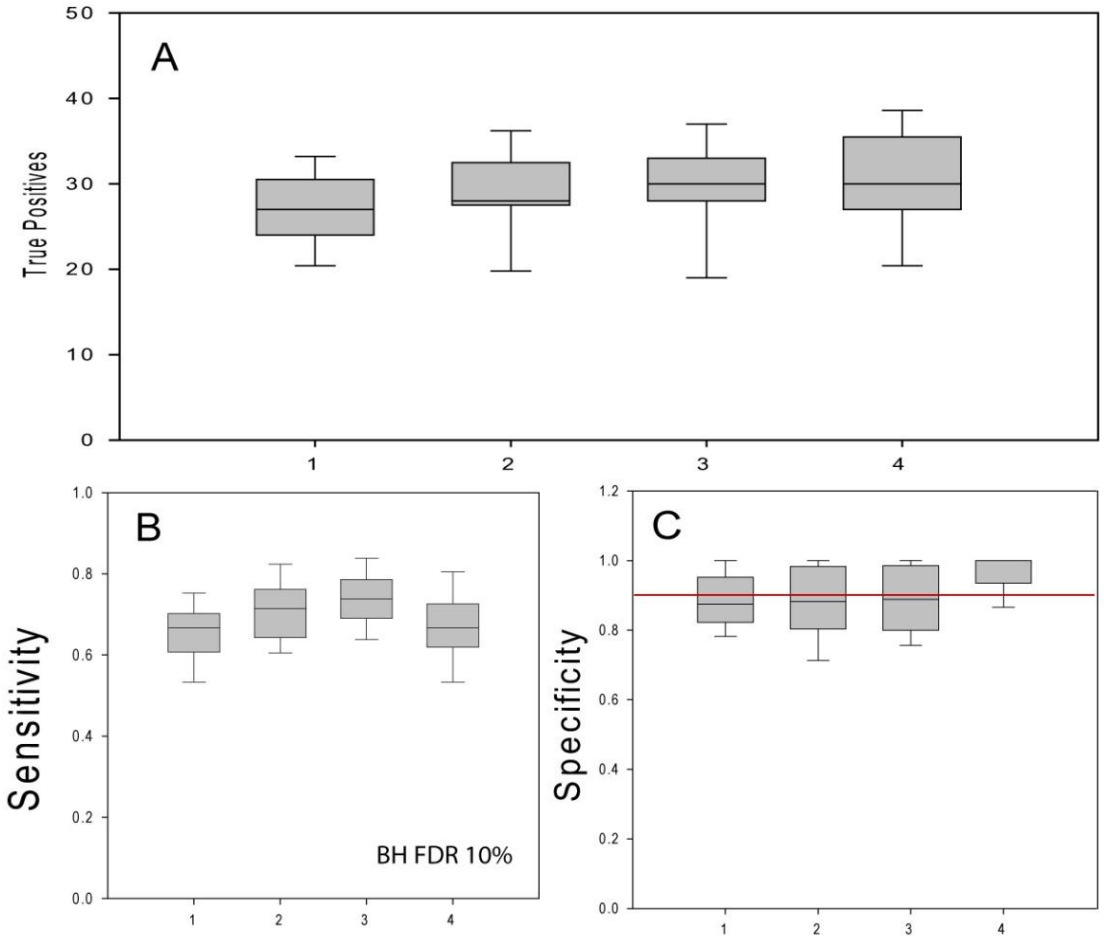
### 1.3.3 Consistency in technical and biological replicates

Our results suggest that, at least on the technical replicates of the Latin Square experiment, the PINC statistic produces p-values that allow for correct inference in discriminating true and false positives. The question remains, however, are n=1 experiments generally a good idea? For tightly controlled datasets such as the Latin Square dataset, the performance of the PINC algorithm at n=1 is clearly acceptable (Figure 1.2). However, what happens when we examine biological datasets in which biological noise, by necessity absent from the technical replicates that make up control datasets, makes up a significant component of the measured signal?

To begin to examine this question we first ask, what are the consequences in the Latin Square experiment of increasing sample size? We applied the PINC algorithm to technical replicates in the Latin Square dataset by analyzing N=1, N =2 and N =3 (conditions 1, 2 and 3 in Figure 1.3). For N=2 and N=3, we determined the average value for each probe and then applied PINC in a pairwise probe to probe comparison similar to when N=1. By contrast, in most microarray experiments an analysis is performed at the probeset level; that is, an algorithm such as RMA is applied to produce for each probeset on each array a single value and a test statistic is then applied to these values[17]. We therefore included a comparison of PINC to a probeset level analysis, in this case using Cyber-T (not paired as the microarrays in the Latin Square experiment do not have a paired relationship). Condition 4 in Figure 1.3 shows the results of using quantile-quantile normalization and RMA summation[17] to power an analysis with Cyber-T an N=3.

Figure 1.3 shows the results of these analyses of different sample sizes on the 13 Latin Square 2X comparisons. Figure 1.3A shows the number of true positives that can be recovered at an arbitrary cutoff of four false positives (similar to Figure 1.2C). Figure 1.3B shows the results of sensitivity and specificity after applying a BH-FDR cutoff of 10% (similar to Figure 1.2A). We see very similar results no matter if we use 1, 2 or 3 microarrays ( conditions 1-3) or use a probeset analysis at N=3 (condition 4). This confirms the observation of Klebanov and Yakovlev that noise derived from technical replicates is generally low [36] and that PINC can yield results similar to a popular probeset algorithm such as Cyber-T despite the use of only one microarray.

We next applied PINC to a series of biological replicates with varying degrees of biological noise. We chose to analyze an Affymetrix dataset from a cell line study (Accession: NCBI Entrez Geo GDS756) that explored changes in gene expression of SW480, a primary colon cancer cell line [37] and an experiment extracted from human tissue with multiple human donors (Accession: GDS2191) that explores the regulation of the ubiquitin cycle in bipolar disorder [16]. We reasoned that the biological noise in the human tissue dataset would be higher than the biological noise from the cell lines, while the cell lines would in turn have more noise than the technical replicates of the Latin Square experiment. These datasets are summarized in supplemental Table 1. The experiments we chose all met the following criteria; the number of paired datasets needed to be at least N =3, the datasets needed to be a control versus treatment type of design, the datasets needed to be Affymetrix HG-U133A datasets and the CEL files available. Within each dataset, samples for analysis were randomly chosen using a random number selection program (http://www.random.org).




1: Quantile-Quantile from DChip → Cyber-T paired comparing 11 probes of the first Latin Square experiment in each 2X condition (e.g. Expt1_R1 vs Expt2_R1, Expt2_R1 vs Expt3_R1, etc.) → Generation of ROC Curves → # of true positives recorded at 4 false positives (panel A) → stat level norm [12] → generation of p-values by PINC → BH FDR applied at 10% → sensitivity (panel B) and specificity (panel C)

2: As A, except before being fed to cyber-T paired, the 11 probes are averaged from the first two Latin square experiments (e.g. Expt1_R1 + Expt1_R2)

3: As A, except before being fed to cyber-T paired, the 11 probes are averaged from all three Latin Square experiments (e.g. Expt1_R1 + Expt1_R2 + Expt1_R3)

4: QQ Normalization and RMA summation from RMA express → Cyber-T (not paired) applied at N=3 → Generation of ROC Curves → # of true positives recorded at 4 false positives (panel A) → statistics level normalization [12] → generation of p-values by scheme 4 [12] → BH-FDR applied at 10% → sensitivity (panel B) and specificity (panel C)

FIGURE 1.3: The effect of sample size on sensitivity and specificity for the 13 Latin Square 2X comparisons. (A) The number of true positives captured at an arbitrary cutoff of four false positives. Sensitivity (B) and Specificity (C) at a cutoff defined by 10% BH-FDR.

Datasets were first analyzed using "Scheme 4" as described previously, which compares datasets at the probeset level using Cyber-T and then calculates p-values by assuming a single background distribution [24]. Scheme 4 and a gene list of significant results were determined using BH-FDR at 10% FDR. We call these gene results the "Scheme 4 N=3 probeset results" (condition 1 in Figure 1.4). Next, using the 6 arrays (3 of condition 1 X 3 of condition 2), we generated 9 different lists of differentiated genes by performing all 9 possible comparisons using PINC under 10% BH-FDR (condition 2 in Figure 1.4).

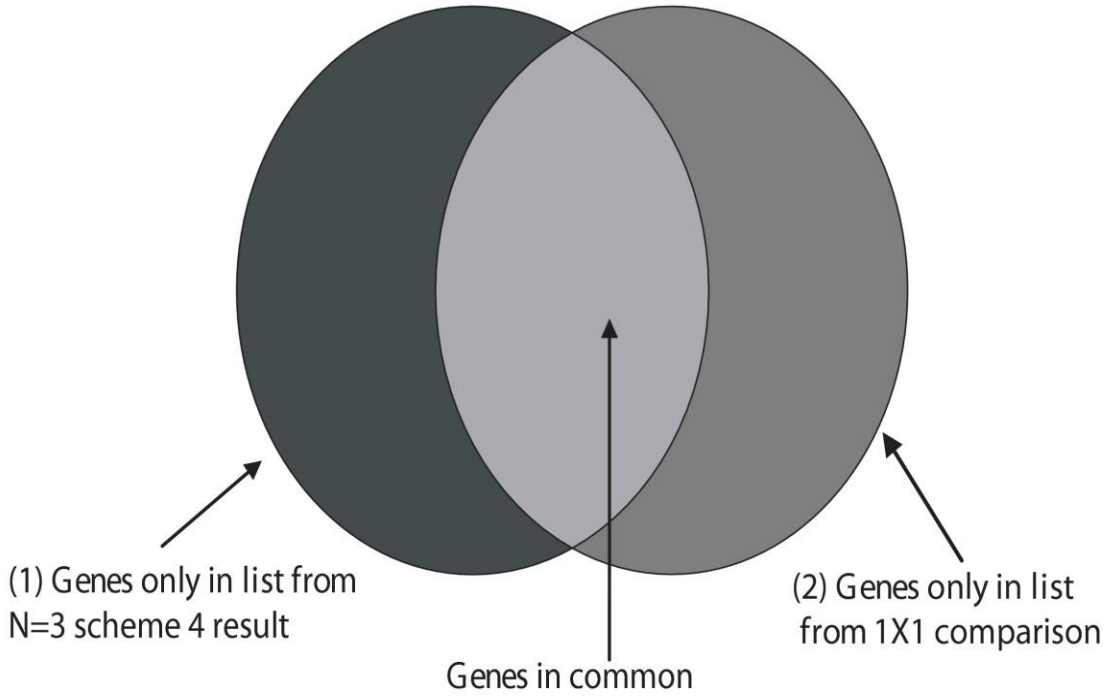



FIGURE 1.4: Venn diagram depicting how genes from each type of analysis are compared in Figure 1.5. If biological variability is low, then the majority of genes detected will be common to both methods of analysis.

We then compared these 9 results to the "Scheme 4 N=3 probeset results" to determine how consistent the gene selection process was. Figure 1.4 depicts a Venn diagram of how these results are interpreted.

Boxplots showing the results of these 9 analyses for each dataset are shown in Figure 1.5. In the Latin Square experiments, genes detected by the 9 different PINC comparisons are in good agreement with the n=3 gene list (average number of consensus genes found via 1X1 comparisons ≈ 88% retained, panel A, Figure 1.5). As we proceed to the more diverse biological datasets, gene list agreement decreases to 68% and 32% for the cell culture experiment and tissue experiment respectively (panels B and C, Figure 1.5). For the human tissue experiment, the gene lists generated from the 9 different 1 to 1 comparisons show the highest level of variability (panel C, Figure 1.5). This is consistent with other tissue microarray experiments we analyzed (data not shown). While this is not a surprise, it does emphasize the danger of analyzing tissue samples via microarray when sample size is low. The extent of variability suggests that when designing a microarray experiment, selection of sample size should reflect the noise of the biological source. These results suggest that a "one-size-fits-all" rule of microarray experimental design (such as always have N = 5) is not always the best use of experimental resources. When biological noise is very low, a single microarray may suffice; when biological noise is high, many microarrays may not capture all of the variability in the system under study.

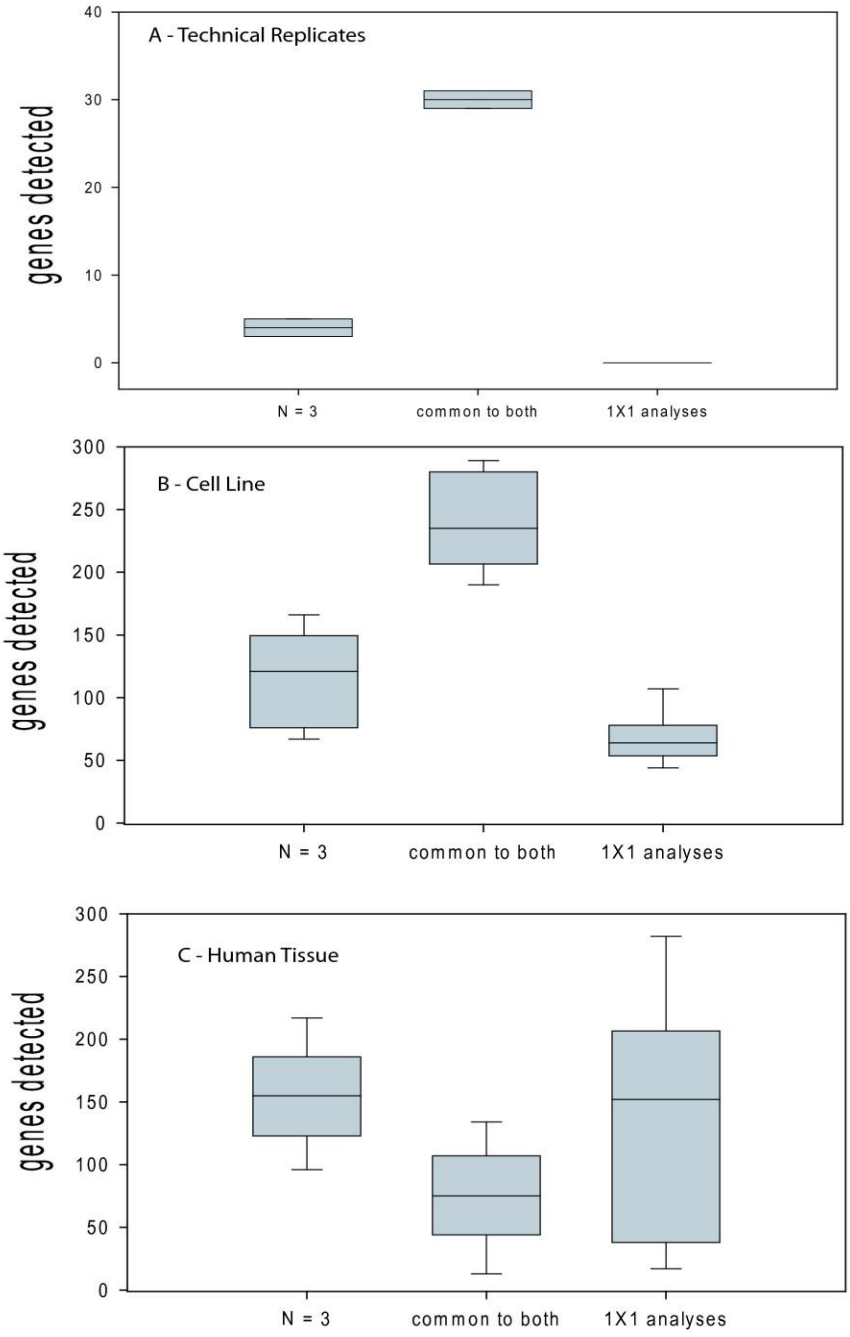



FIGURE 1.5: Comparison of different biological sources using probe-set analytical methods at N=3 and PINC. (A) Latin Square dataset – majority of significant genes are common to both methods. (B) Human cell culture dataset – majority of genes still in agreement, although with an increase in variability. (C) Human tissue dataset – very small selection of genes common to both and a large degree of variability in the 1 to 1 comparison group.

### 1.3.4 Biological confirmations of PINC predictions: Confirmation by qPCR

PINC was applied to a set of *Drosophila* microarray experiments, obtained from Dr.Julie Goodliffe in a study exploring gene expression at different stages of fly development. For every condition in this particular experiment, there were 2 microarray replicates. This means that there were an insufficient number of replicates for probeset comparisons. Using PINC, we were able to generate gene expression predictions for each of the experimental conditions. For each comparison, we generated a set of 4 predictions (2 experiments X 2 experiments, creating 4 possible 1:1 comparisons). We then grouped genes that showed a consistent pattern of expression across all 4 comparisons. A select number of these genes were then chosen and confirmed via qPCR analysis.

## 1.4 Conclusions

Experiments with few numbers of repeats are ineligible for analysis via most published microarray analytical methods. We have shown that when applying analysis at a probe level using PINC, we are able to generate reasonable results on control datasets at N=1 in each condition. For paired single microarrays, PINC outperforms both canonical statistics and a recently published method [15] while offering conceptually simple statistics and fast run-times. Because the p-values are derived from a distribution estimated from all of the genes on the array, PINC also avoids the large p-values usually associated with low sample size microarray experiments. This allows for the possibility of using a more conservative cut off criterion such as family wise error rate, as an alternative to false discovery rate when selecting a p-value cutoff for selecting differentiated genes (Figure 1.2).

The success of the PINC algorithm in performing accurate inference on the Latin Square dataset at N=1 suggests that there is little benefit to performing additional technical replicates with a non-existent or exactly common background. This is consistent with previous literature [36] as is our observation that one gets largely similar results whether one uses n = 1, n = 2 or n = 3 in ranking the 2X Latin Square experiments (Figure 1.3). The ability to analyze single Affymetrix experiments in a statistically rigorous way opens up the possibility of interesting analyses even for experiments in which samples from multiple biological samples are collected. For example, in a cancer study in which cancer tissue is compared against non-cancer tissue from the same patient, we could generate gene lists consisting of genes that are differentially expressed at a given cutoff threshold for every patient in the study. This may yield very different insights than the usual practice of averaging the samples together and performing a single analysis to generate a single gene list. We know that diseases like cancer are very diverse with many different molecular mechanisms presenting similar clinical diagnostics. The ability to evaluate each patient individually in a statistically rigorous way may improve our understanding of the diverse causes of diseases such as cancer and may allow for better use of microarrays in personalized medicine.

# CHAPTER 2: QUALITY CONTROL METHOD DEVELOPMENT FOR ARISA ANALYSIS

## 2.1 Background and significance

Multiple comparison experiments do not only exist in the microarray world of gene expression. Many molecular based techniques have been developed to help identify and characterize living matter in different environments around us. As in the microarray world, surveys of complex microbial communities involve low sample replicate sizes and simultaneous measurements that have many dependent variables. Solving the complexity of these types of experiments will better enable us to identify how these communities function.

One of the goals in biology is to identify the microbial taxa that exist within a given habitat. Knowing what taxa are present is a crucial step in ecology for controlling pollution [38], [39], [40], determining soil composition [7], in biogeochemical cycles[7], assessing disease (e.g. [41]), regulating the composition of the atmosphere and recycling nutrients, and global nitrogen utilization[42]. The field of metagenomics, first defined by Handelsman et al. is the study of genetic material extracted directly from a natural environment[43]. Since the vast majority of microbial species within a given environment are not amenable to cell culture[44,45,46], DNA sequencing in addition to molecular techniques based on DNA/RNA properties [47] have been developed to identify taxa based on genetic makeup of shared elements, alleviating the need to cultivate microbes [7,48]. A recent focus has been to utilize deep DNA sequencing to identify the microbial

diversity of the available genetic material. Kunin et al. provide an excellent review of role of DNA sequencing in the field of metagenomics [49]. DNA sequencing alone is not the final answer, however, as the need for better pipeline analyses and bioinformatics is required to properly cleanse the data and to avoid misidentifying taxa within microbial communities. For example, Sogin et al. used sequencing to show that microbial diversity is much more complex than previously thought, underestimating the numbers of microbial species by several orders of magnitude [50], indicating the presence of a biosphere of rare, unknown species. However, more recent publications have questioned such conclusions, showing that these early findings are perhaps nothing more than sequencing error [51,52].

Though DNA sequencing costs continue to decrease, the costs of performing such DNA sequencing studies remain prohibitively high for the majority of scientists. Figure 2.1 summarizes the currently available sequencing strategies for characterizing microbial communities. Untargeted sequencing generates information about all of the genomes of all species in a DNA extraction product. A large number of sequences in such experiments fail to match DNA in public databases. Assembly of these DNA sequences (< 500 nt reads) also remains a difficult problem. As an alternative, techniques that exploit conserved regions of RNA/DNA genes can be used. That is, rather than sampling randomly from an entire DNA extraction product, we can focus on characterizing just the 16S rDNA (small subunit ribosomal RNA genes) regions, which is cheaper, and yields data for which there is a very large comparative set, allowing us to make the best available characterization for microbial taxa (middle paragraph of figure 2.1). Even

cheaper and quicker than sequencing are the many molecular fingerprinting techniques that allow us to make a general snapshot of the community.

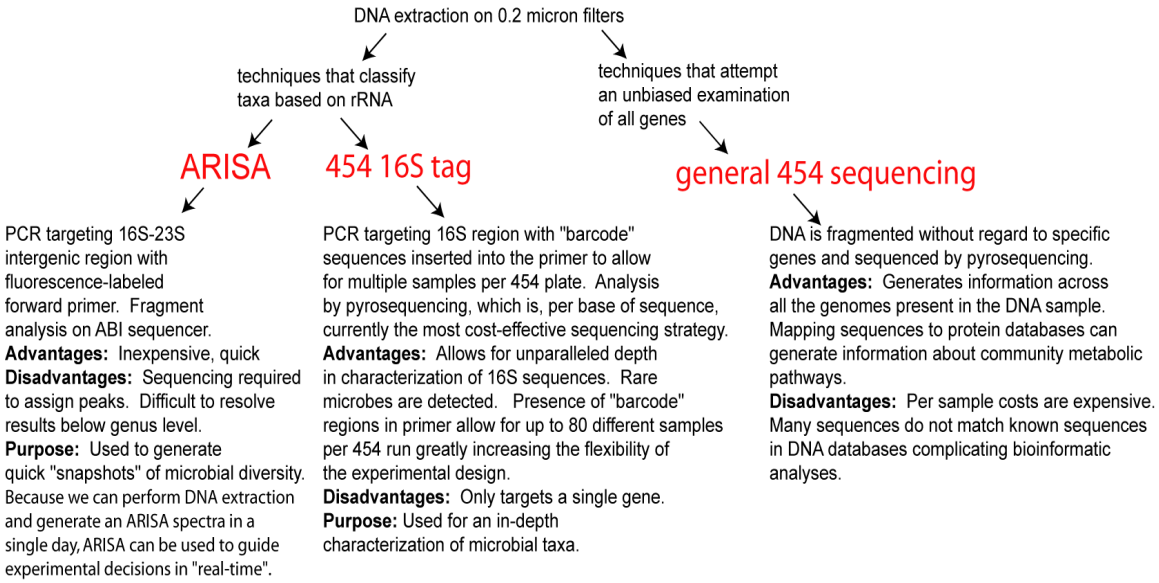



FIGURE 2.1: Techniques used to characterize microbial communities.

There are many published examples of assays using molecular techniques that target the highly conserved 16S rRNA gene region in bacteria. For example, in a technique called terminal restriction fragment length polymorphisms (T-RFLP), fluorescently labeled primers bind to a conserved 16S rDNA region which is amplified prior to restriction endonuclease digestion of the PCR product [53,54]. These size differentiated DNA products are the basis for identifying what species are present within the microbial community. Other similar techniques include ARDRA[55], DGGE[56], 2D-PAGE[57] and ARISA[8].

The technique that is the focus of Chapter's 2, 3 and 4 is the automated method of ribosomal intergenic spacer analysis (ARISA)[8] which is an automated modification of

the molecular biology technique RISA, first described by Borneman and Triplett [58]. García-Martínez et al. provide an excellent overview of the RISA process [59]. In prokaryotes, genes encoding for 16S and 23S RNA subunits have been largely conserved throughout the course of evolution and most often these genes are located in close proximity to one another. However, the intergenic region between the 2 ribosomal genes does not display the same level of conservation (upper panel, Figure 2.2). As a result the intergenic region varies widely across species in terms of composition and size [59]. These size differences in the intergenic region are what ARISA identifies in order to identify taxa and establish a profile within a given ecosystem.

Isolation of the intergenic region begins with the selection of fluorescent DNA primers that target the end regions of the anchoring conserved genes: 16S (3 primed end) and 23S genes (5 primed end). Like T-RFLP, a PCR amplification step then increases the quantity of the targeted section of the DNA. In T-RFLP, different sized DNA fragments are created by way of a restriction enzyme. For ARISA the fragments consist of the complete intergenic sequence plus the 2 ends of 16S and 23S genes. The amplified DNA product(s) is then run on a separation matrix, so that if multiple species are present they will be discriminated by length. The end result is an electropherogram in which each DNA species is characterized by a length and fluorescent intensity.

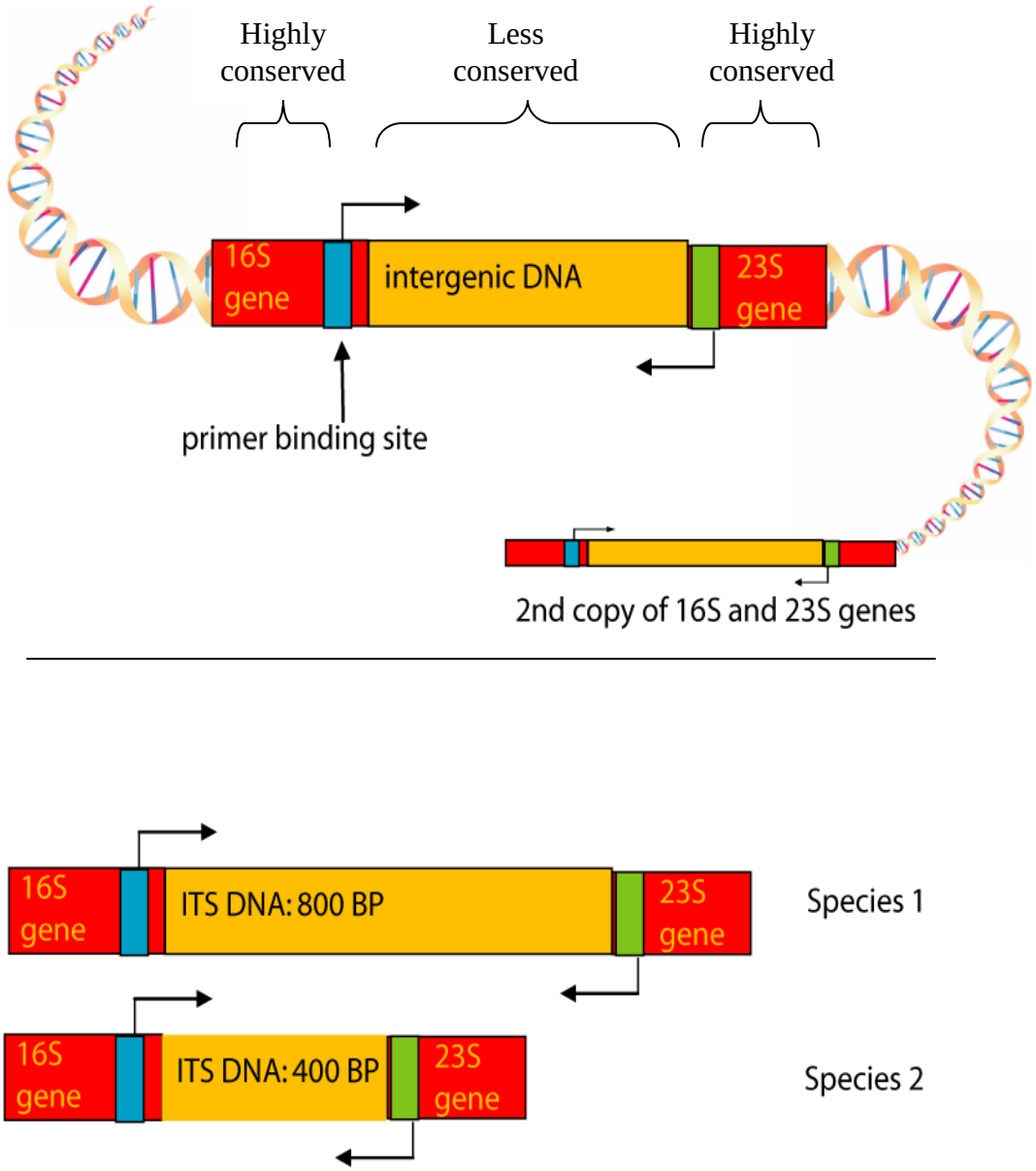



FIGURE 2.2: Intergenic region between the 16S and 23S genes of DNA sequences. A single species may have multiple copies of 16S and 23S genes with intergenic lengths of varying size (upper panel) and different species often have different intergenic lengths (lower panel). 16S and 23S genes are highly conserved (red boxes) while the intergenic region (yellow box) is more variable in length and in composition. The blue and green boxes represent the primer binding sites on the 16S and 23S genes used for PCR amplification.

Figure 2.3 shows an example of an electropherograms, in which the axes are fluorescent intensity versus time to the detector (a proxy for the DNA length). Adding known DNA size standards allows for the estimation of size for each DNA fragment (in DNA nucleotide space), using one of several interpolation algorithms. Estimates of the number of different species are then made based on the number of intergenic lengths (peaks) observed. Taxa calling can also be attempted based on the presence or absence of these same peaks [8], [60].

The ARISA method is subject to several limitations, the first of which is the assumption that the regions of the 16S and 23S genes against which primer have been designed have been conserved and the second is that they are in close enough proximity that the PCR conditions allow product amplification. There are known instances in which the 16S and 23S genes are thousands of base pairs apart due to insertion or DNA rearrangement events. For example, the species *Thermoplasma Volcanium* has an intergenic distance of 155,293 base pairs between the 16S and 23S genes. Species such as this are undetectable via ARISA.

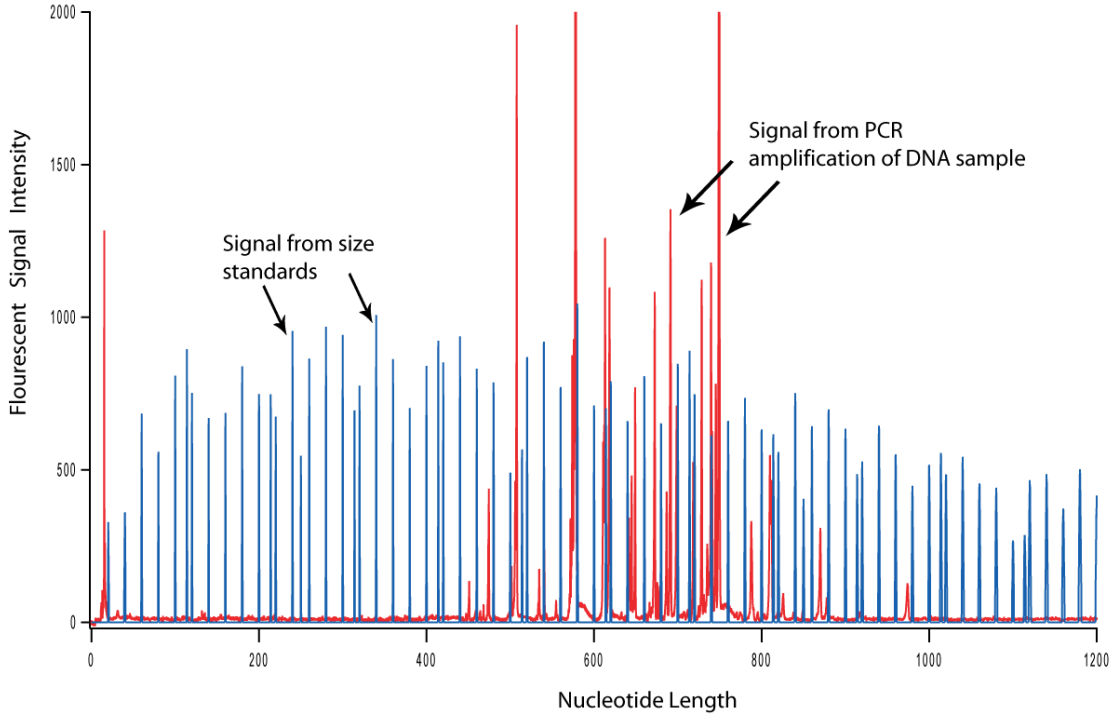



FIGURE 2.3: Example of an ARISA electropherogram. Red lines represent fluorescent signal generated from labeled DNA fragments. Light blue lines represent signal from known DNA size standards which are used to estimate the size of DNA fragment peaks.

In addition, instances are known where duplication events have resulted in multiple copies of 16S and 23S in one genome, which leads to multiple intergenic lengths for a single species. For example, *Vibrio Vulnificus* has 4 separate copies of 16S and 23S genes, all of which have different intergenic lengths (421, 508, 665, and 742 nucleotides (NT)). The presence of this one species alone should yield 4 distinct peaks in an electropherogram.

Any intergenic region that is larger than 1200 base pairs will be difficult to detect using ARISA. The largest size standards included in sequencing are around 1200 nucleotides, so lengths beyond 1200 nt cannot be accurately determined. Also, there are PCR related difficulties when amplifying larger DNA fragments; together these factors

render data in these regions of an electropherogram inconsistent. To estimate how much of a problem this might be, Fisher and Triplett determined that 85-90% of the intergenic distances in bacterial genomes available in GenBank fell within the range of 150 to 600 base pairs[8]. ARISA fragments include both the intergenic region and parts of the 16S and 23S genes (primer to primer distance), so the smallest lengths are around 400 base pairs. Given these constraints, this leaves approximately an 800 base pair window in which to resolve a microbial footprint. Because all of the DNA amplicons in a sample are labeled with the same primers, when multiple species share the same ARISA length there is no way to distinguish between them.

The selection of primers for PCR can significantly affect the outcome. For example, Maggi and Breitschwerdt describe how primer selection changes the accuracy in detecting *Bartonella sp* using ARISA [61]. Also, Jones et al. showed that using 2 different primer sets results in differences in bacterial profiles [62].

Despite these limitations, ARISA has found functionality as a fingerprinting technique allowing comparisons between microbial communities. Within a given community the electropherogram produced is unique and can be considered a "molecular fingerprint". These fingerprints are worthwhile in a number of scenarios including: tracking changes to environmental microbial samples over time, monitoring gut micro biota amongst healthy and diseased specimens, or comparing geographically distinct soil samples. The original ARISA paper by Fisher and Triplett [8] has been cited 260 times.

The question remains, how does one best use these "ARISA fingerprints" to compare and distinguish microbial environments from one another? Similar microbial communities would be expected to cluster together based on similar ARISA fingerprint

profiles. In order to compare these profiles, all peaks within an electropherogram must be accurately identified and the corresponding nucleotide length that each peak represents must be precisely determined. The purpose of the remainder of chapter 2 is to describe data processing and the QC filtering techniques that we have implemented to achieve these goals. Here, we describe how ARISA peaks are identified and how size standards are used in assigning NT length; in addition we describe filtering techniques that identify poor experiments. By developing these data processing techniques, we establish a framework for comparing microbial communities in chapter 3.

## 2.2 Experimental approach

### 2.2.1 Merits

The development of quality control methods is a critical step prior to performing comparative analyses. When comparing ARISA profiles, clustering errors can arise if there is a lack in technical consistency, i.e., 2 highly similar ARISA profiles (e.g. technical replicates) could fail to group together if there is an error in any of the steps. This is obviously undesirable. The methods described here will aid in minimizing some types of errors though they do not eliminate all errors. The biological noise inherent in these types of experiments contributes to the complexity of ARISA results.

Poor experiments can result from many sources including: poor sequencing separation runs, poor reagents, PCR failure and operator error. In order to compare microbial environments (chapter 3) accurately, we must include only results that best represent their biological source. We devised the following strategy to identify good quality ARISA experiments. First we applied a linear interpolation scheme to identify

peaks within the spectra. Peaks were identified by determining patterns in which signal increases with a positive slope, has an inflection point and then has a negative slope. Upon identifying peaks, size estimates are made by assigning known length values to peaks in the size standard and then using the function to assign nucleotide sizes to each peak in the ARISA elements of the electropherogram. Once peaks have been assigned a nucleotide length, ARISA experiments are tested for technical replicate consistency, followed by a novel QC step that assesses how size standards are allocated. At each step in the process, poor experiments are flagged, leaving a subset of ARISA experiments that are better suited for comparative analyses.

### 2.2.2 Linear interpolation

One of our first data processing steps is to apply a linear interpolation scheme to distinguish peak signal from baseline data and to accurately identify each of the size standards. We simply cannot proceed without first correctly identifying each of the size standard peaks in the electropherogram. Dr. Anthony Fodor wrote a peak calling and linear interpolation algorithm, in java, for processing ARISA and T-RFLP fragment data to replace our previous ARISA processing method, in which we identified peaks by identifying upward slopes between consecutive data points. This linear interpolation algorithm identifies peaks based on a number of configurable parameters including: slope distances, inter peak distances, intra peak distances, peak lengths, peak heights relative to background, and so forth. These parameters can be adjusted such that, for the majority of electropherograms in a dataset, the size standard peaks are correctly identified.

For a given spectrum, the linear interpolation algorithm begins by traversing across the spectra, identifying the slope at each data point by calling a linear regression function that uses neighboring data points (the number of which can be configured in the Peak Studio in chapter 4). Each data point is assigned to one of four ‘phases’: a nonpeak phase, an upslope phase, an inter peak phase or a down slope phase (Figure 2.4). Starting in a nonpeak phase, each data point is checked to see if the slope at that location is greater than a pre defined threshold. If so, then the data point is labeled as being part of an upslope phase. The slope threshold is a configurable parameter that can be adjusted. Subsequent data points are identified as part of the upslope phase until a slope change equals 0 or less, at which point the phase changes to the inter-peak phase (i.e., the inflection point at which the peak no longer rises but begins to level off) . The slopes for each peak in the inter- peak phase are determined until a negative slope is identified that surpasses another pre defined threshold, at which point data points are defined as being in the down slope phase. The algorithm continues to traverse down the slope until a data point produces a slope = 0, at which point the phase reverts back to being a nonpeak phase (right side of Figure 2.4).

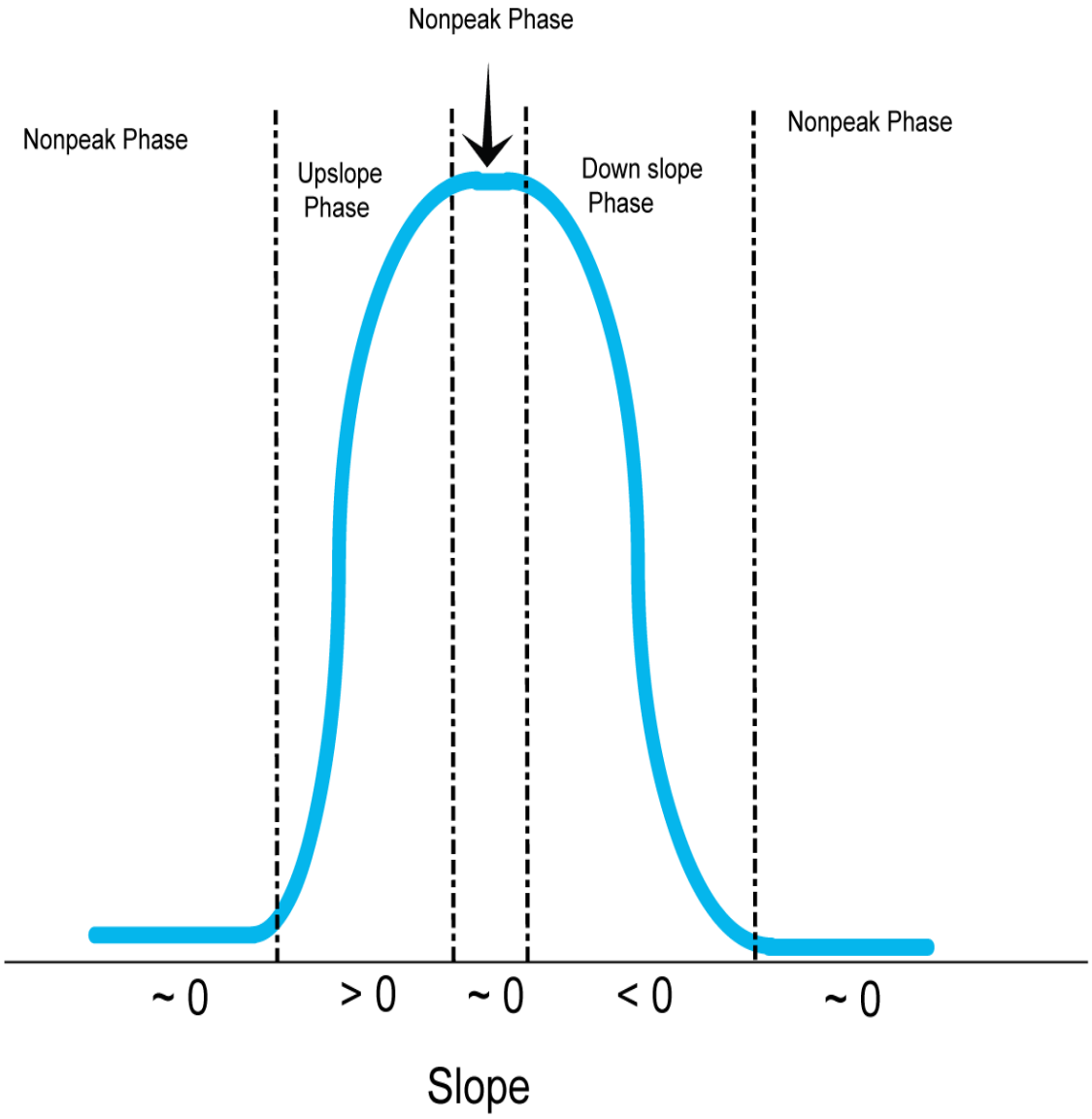



FIGURE 2.4: Example of the 4 phases of peak identification in the linear interpolation algorithm. Linear interpolation and peak calling algorithm designed and written by Dr. Anthony Fodor.

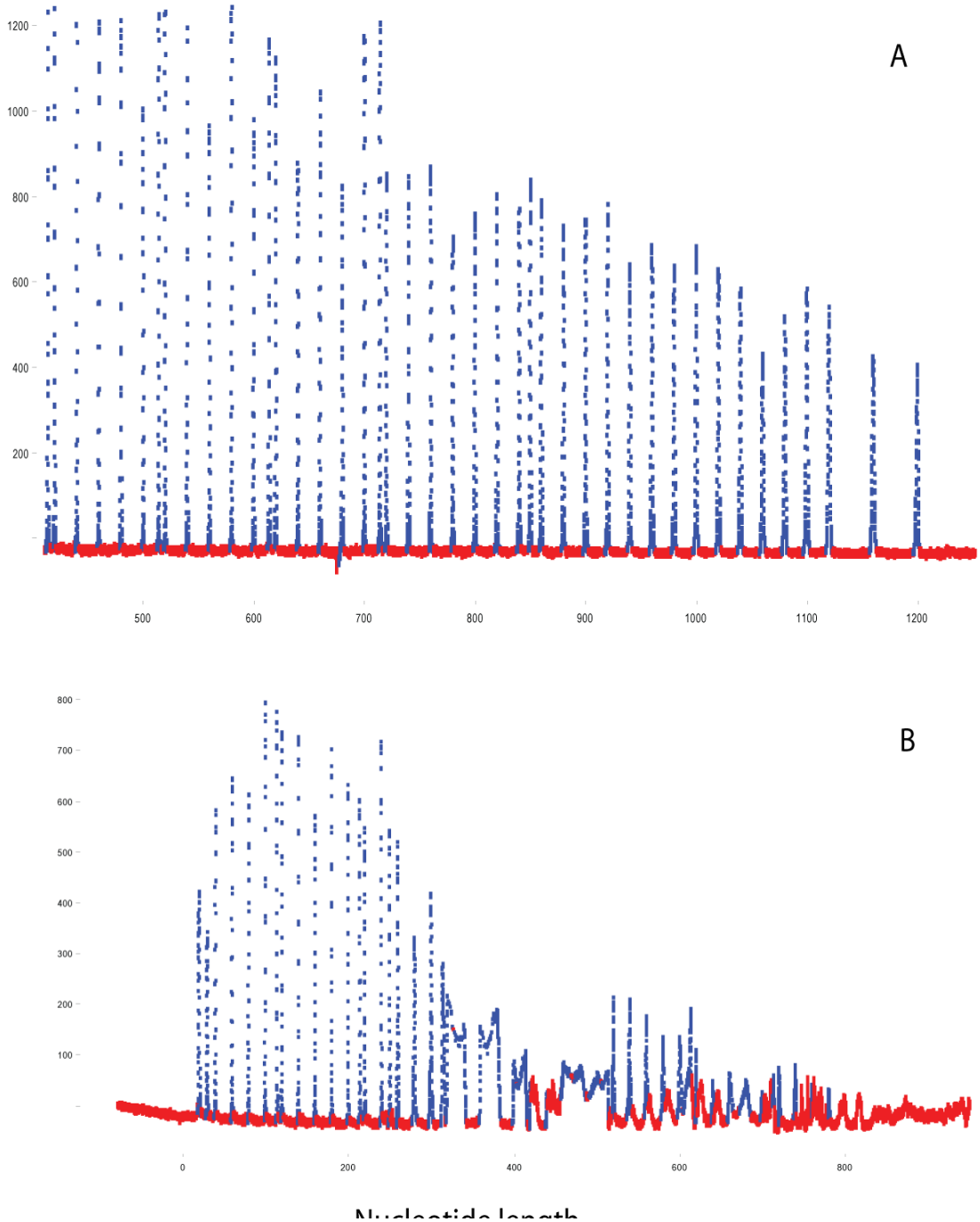



FIGURE 2.5: Two examples of peak identification of size standards from linear interpolation. Blue data points have been identified as part of a peak, while red data points represent baseline signal. Panel A shows a close up of an electropherogram where the expected size standard peaks were correctly identified. Panel B shows a case where the chosen parameters fail to identify the appropriate number of peaks. In this case, it is also difficult to manually identify what constitutes a size standard peak.

Once all of the data points in a scan have been identified as one of the 4 phases in Figure 2.4, peaks and non peak regions can be identified. A peak is identified as starting at beginning of an upslope phase and ending at the end of the down slope phase. For each peak, the height is determined by taking the difference between the highest and lowest data point within the peak region. If the peak height fails to surpass an adjustable height threshold, the peak is relabeled as a non peak region. An additional parameter was implemented to improve peak identification including a parameter that tests the proximity from one peak to the next.

After linear interpolation, all data is identified as either peak or baseline signal (i.e., nonpeak phase). After the linear interpolation step we can assign a value (corresponding to the known length) to each peak in the size standard set. Since the number of peaks in the size standard spectra is known, all of the parameters in the linear interpolation algorithm can be adjusted to optimize the identification of the size standard peaks. Figure 2.5 shows an example of good size standard signal (panel A) and poor size standard signal (panel B). In panel B, it is difficult to reliably identify size standards due to inherent noise.

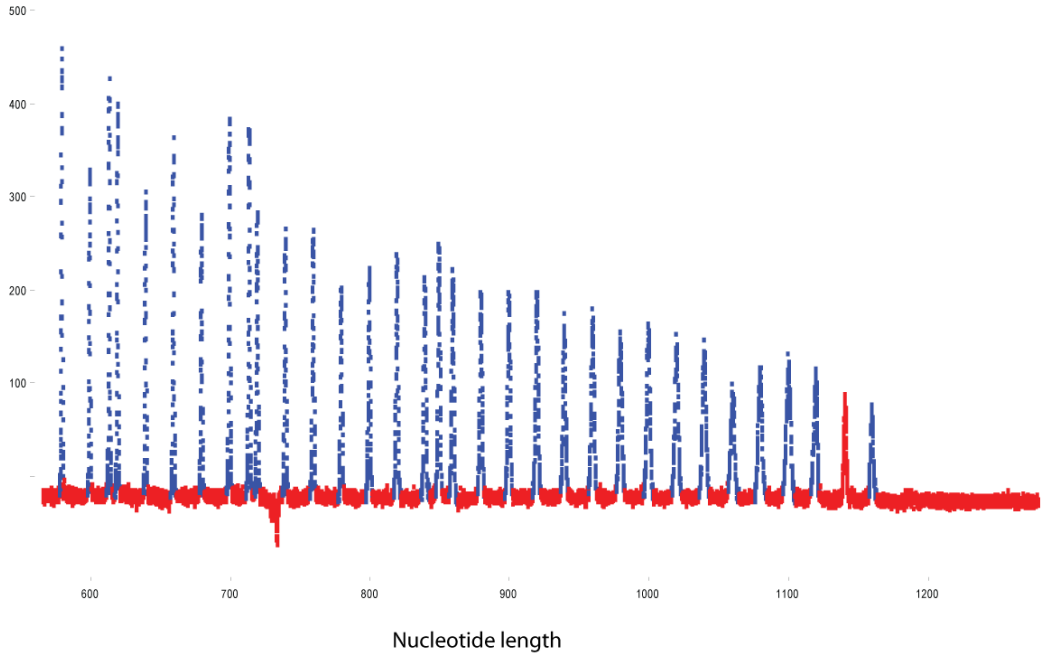



FIGURE 2.6: Example of peak identification from linear interpolation where 1 peak fails to correctly be identified as a size standard peak (shown as the red peak, second to last from the right). Blue data points have been identified as part of a peak, while red data points represent non peak signal. The second to last size standard peak fails to be accurately identified as a peak.

When many size standards fail to be identified, these experiments are examined visually and possibly flagged for removal. In cases where an experiment misidentifies only a handful of the size standards, we can manually correct for this in the code by adding or removing peaks. Figure 2.6 shows an experiment in which one of the size standard peaks failed to be identified using a given set of parameters in the linear interpolation code. In this instance, the peak was manually added into the code and the experiment did not need to be discarded. From these steps, a subset of ARISA experiments can be identified for further QC analysis.

### 2.2.3 Technical replicate consistency

Upon correctly identifying each of the size standards in an ARISA experiment, there still remains the possibility that while the size standard spectrum is good, the ARISA signal is poor. We apply the peak calling parameters used in linear interpolation to identify ARISA signals. If some error arises in the ARISA signal (such as sample loading error, lack of fluorescent tagging, PCR error), a poor result could occur that is not reflective of the microbial environment. ARISA reactions are often run twice per DNA sample, creating a technical replicate to confirm PCR and fragment separation consistency. Using such replicates, we determined the Pearson's correlation coefficient (after assigning the nucleotide length) for the data from each corresponding pair. That is, we bin neighboring data points into bins of 1 NT in length and then run the correlations. Generally, technical replicates should show a high degree of reproducibility. Experiments with a correlation below a specified threshold (0.85 is the threshold shown), were excluded from further analysis. Figure 2.7 shows 2 pairs of replicates, one with a poor correlation between the replicates and one with a good correlation.

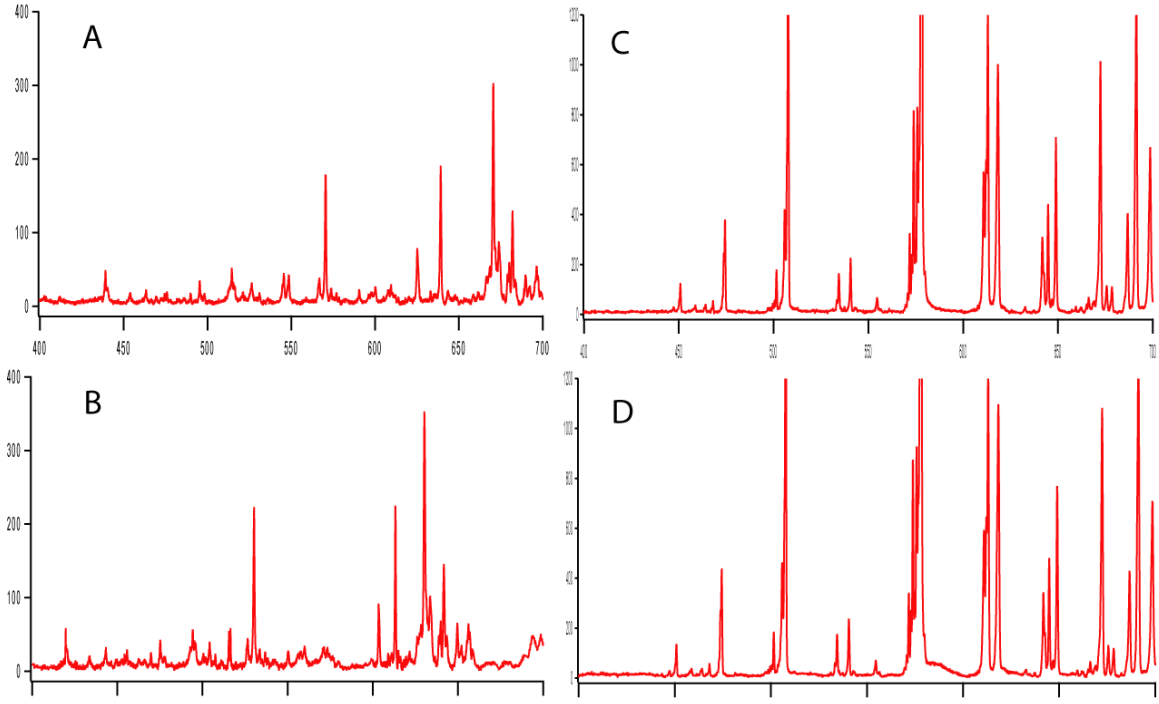



FIGURE 2.7: Zoomed in regions of electropherograms that demonstrate QC correlation Performance: A&B: Low correlating pair of technical replicates ($r^2 = 0.04$) where the top spectra has shifted and been stretched relative to the lower panel producing a poor match. C&D: High correlating pair of technical replicates $r^2 = 0.99$).

Technical replicate consistency is fairly effective at identifying where technical replicates fail to correlate; a likely cause in this instance is degradation of the capillary quality of the genetic analyzer. This is helpful for troubleshooting the source of error (platform behavior versus sample preparation steps). If the ARISA experiments are run only as pairs and the two paired experiments fail to correlate with each other, we are forced to throw both experiments away since there is no way to determine which experiment failed to run correctly.

In order to save a good experiment when the technical replicate correlation is low, we correlated each ARISA experiment to all the other experiments within a given dataset, i.e., not just to its replicate partner. If there is a reasonable expectation that profiles will be largely similar then this is a defensible approach. In chapter 3 we

demonstrate with Sanger and 454 sequencing that in a human subject experiment the similar ARISA profiles also show similar sequence profiles. That is, for a well-designed and conducted ARISA experiment, some portion of the results should correlate highly with others in that dataset. The likelihood is that a poor technical replicate will fail to correlate with any of the other results, barring a consistent error. By generating correlations in an "All versus All" result matrix, we were able to retain 21 ARISA datasets that we otherwise would have thrown away had we used only technical replicate correlations.

### 2.2.4 Assessment of size standard assignments

In addition to establishing precision through a technical replicate QC step, we need to ensure that, when we assign a length to an ARISA peak, it is as accurate as possible. Inconsistencies in assigning size standard lengths skew the sizing of ARISA data peaks. To ensure consistency, we developed a QC method where we assign nucleotide lengths to the ARISA spectra but rather than using the entire size standard list (e.g. 68 size standards), we only use every second size standard (i.e., only using half the size standards, e.g. 34 size standards). At each spectra location where the size standard is skipped, we get a predicted value at that location that is determined by the neighboring size standards. Our predicted value can then be compared to the size we would have assigned had we used the size standard at that location. The differences between the predicted size (predictedSize) and the actual observed size (observedSize) are determined for each skipped size standard and the absolute sum of these differences is used to define a QC score (with a lower score being better).

$$\text{QC Score} = \sum_{i=1}^{N} | \, predictedSize_i - observedSize_i \, | \qquad (2.1)$$

Experiments with high QC scores are then discarded from further analysis. Figure 2.8 shows a poorly performing experiment (high QC score) compared to a better performing experiment. In the top panel of figure 2.8, the noise present in the standard signal electropherogram is evident in the latter half as the size standard intensities dramatically decrease and the ability to qualitatively identify the peaks becomes difficult. In this instance, using the "every second size standard" QC method results in a higher QC score (~2.6). This QC method is redundant for obviously poor results such as in the top panel of Figure 2.8; however it does have the advantage of rapidly assessing an entire dataset without having to manually visualize each dataset and then decide whether or not the size standards are poor.

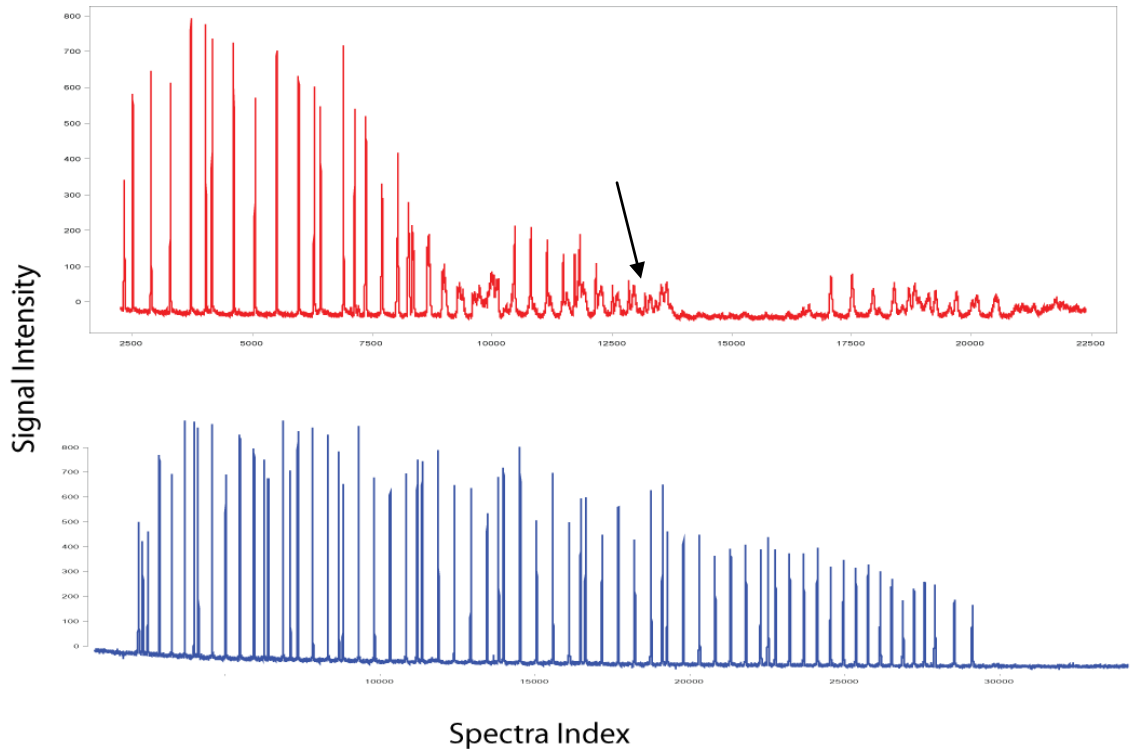



FIGURE 2.8: QC results showing differences in good and poor size standards. Top Panel (red) depicts the ARISA spectra (size standards only, not raw ARISA data) for a poor QC result (QC score = 2.6) from a human subject sample. The expected 68 standards peaks are difficult to resolve (black arrow, top panel). Bottom Panel (blue) depicts the spectra of size standards of a typical experiment with a better performing QC result from the same subject (QC score = 0.2). The size standards are easily defined and spaced apart as expected.

### 2.2.5 QC comparison to ABI's GeneMapper software

The ARISA experiments used in our analysis were produced from an Applied Biosystems 3130 genetic analyzer to produce data files in .fsa format. We compared our QC methods to the default settings in ABI's GeneMapper® Software v.4.0 to determine how well the QC methods agree with one another. GeneMapper provides researchers with the ability to size DNA fragments based on size standards and offers QC tests that estimate the integrity of the DNA sizing. Samples that fail to meet the QC criteria are flagged and are not available for size calling or for further downstream analysis.

We tested 214 ARISA experiments using both our QC methods and GeneMapper. Figure 2.9 depicts a Venn diagram showing the number of ARISA experiments that pass the various QC checks. Of the 214 ARISA experiments, 110 experiments passed QC checks using both our QC checks and GeneMapper. For GeneMapper, over half of experiments were in agreement with our QC methods while GeneMapper found an additional 50 that we excluded using our QC methods.

We tested 2 additional parameters found in GeneMapper so see what effect there was on identifying QC experiments. The first parameter was GeneMapper's size calling methods of which there are 5: 2nd order least squares, 3rd order least squares, cubic spline interpolation, local Southern method and Global Southern method. Details about these sizing methods are available in the GeneMapper Software User Guide. Regardless of which of these size calling methods is selected, the same experiments were identified as being poor. The second parameter was GeneMapper's data smoothing option. Users can choose between light, heavy or no smoothing. Figures 2.9 (no data smoothing) and 2.10 (heavy smoothing) show that using the smoothing option greatly effects the number of experiments that will pass the QC test. By applying the heavy smoothing option we get 128 experiments that agree with our QC results. From these comparisons alone, we cannot say whether one QC method is more valid than the other, only that there are different results depending on which parameters are selected.

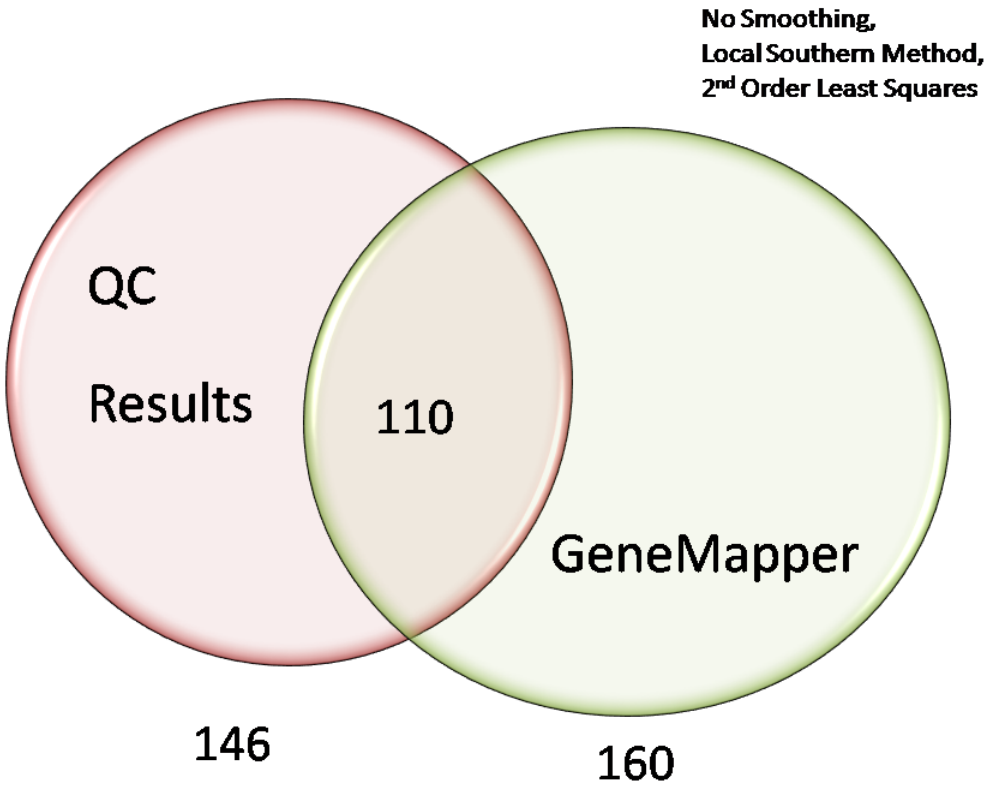



FIGURE 2.9: Venn diagram comparing QC results to GeneMapper without using GeneMapper's data smoothing option. Of 214 experiments, our QC results identified 146 suitable for further analysis, while GeneMapper identified 160 experiments that meet their criteria. For GeneMapper, 2 different size calling methods ($2^{nd}$ order least squares and local Southern method) identified the same sets of experiments as being poor.

In Figure 3.15 (chapter 3) we explored how well data generated by GeneMapper could cluster a set of ARISA results and found that GeneMapper derived clusters failed to match what was expected. Data generated by our QC methods and peak calling matched almost perfectly. In addition, GeneMapper does not allow a user to perform an entire pipeline of analysis but instead needs to export the data for further analysis, while our methods allow for 1 continuous pipeline with little user intervention. For experiments that GeneMapper determines to be poor, there is no option allowing for export, making it

difficult to recover data that might have only minor errors identifying size standards. Our code allows us to manually add or remove peaks in the pipeline that the peak caller misses.

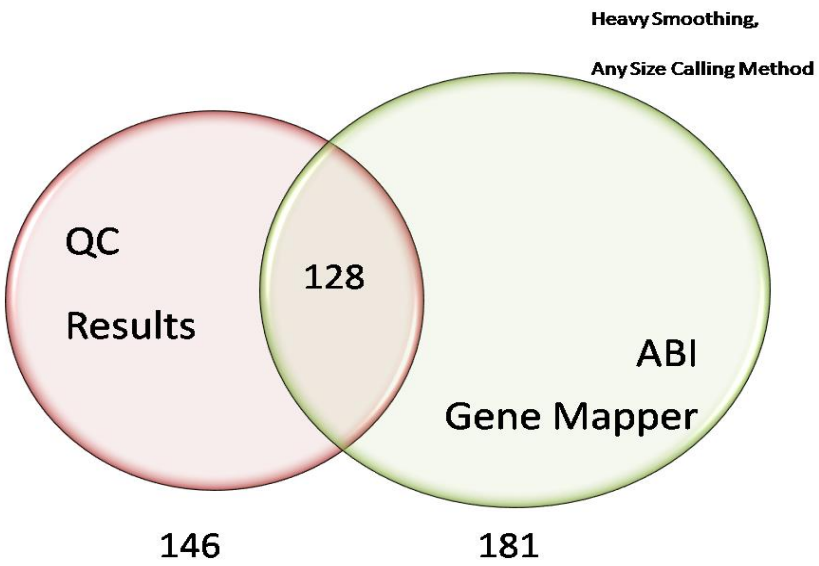



FIGURE 2.10: Venn diagram depicting the number of successful ARISA experiments using our QC methods and GeneMapper using "Heavy" smoothing option. Using heavy smoothing, GeneMapper identified 181 experiments that meet their criteria. All 5 of the GeneMapper size calling methods (2nd order least squares, 3rd order least squares, cubic spline interpolation, local Southern method and Global Southern method) resulted in the same number experiments being identified as poor.

There are some favorable attributes to GeneMapper. GeneMapper allows for a rapid visualizations and comparisons between ARISA experiments. The QC threshold levels are configurable and users can rapidly assess the quality of their ARISA experiments qualitatively. However GeneMapper is not freely available. Therefore, we developed a java based ARISA viewer that also allows for the rapid viewing and comparison of ARISA spectra but is free to the research community (chapter 4).

## 2.3 Quality Control Summary

QC steps were applied to an ARISA dataset that explored the microbial community of the human gut in an effort to filter poor experiments from subsequent analysis. First we applied the peak calling and linear interpolation scheme to distinguish peak signal from baseline data. We adjusted a number of parameters so that the majority of standard peaks could be identified. We discarded experiments that failed to identify the majority of size standards and retained experiments where all size standards could be easily identified by visual inspection. We then applied the technical replicate correlation QC filter, followed by the size standard assessment allocation QC filter. Upon applying these QC filtering methods, we removed 68 of 214 ARISA experiments to produce 146 experiments suitable for clustering and for the comparison study in Chapter 3.

# CHAPTER 3: A COMPARISON OF ARISA CLUSTERING METHODS

## 3.1 Background, merits and significance

Chapter 3 continues the focus on the automated method of ribosomal intergenic spacer analysis (ARISA) [8,39], a molecular biology technique derived from RISA, first described by Borneman and Triplett [58]. ARISA determines the structure of the microbial community by PCR amplifying the intergenic region between the 16S and 23S genes. ARISA can provide a rapid profile of an entire microbial community at a very low cost compared to DNA sequencing. In the generation of an ARISA profile, DNA from a community is isolated and the intergenic regions are PCR amplified. The resulting DNA fragments are separated via a genetic analyzer according to size, and each fragment length can be estimated from known size standards that are concurrently run along with the DNA fragments.

When distinguishing different microbial communities via ARISA, there are many choices during data processing and clustering that can potentially influence the results. We compared different parameters involved in ARISA data processing, in an effort to understand which had the most influence on differentiating one microbial environment from another. A common analytical strategy is to group neighboring data signals into bins and assign an appropriate nucleotide length to the bin based on a function fit using size standards. The sizes of these bins can vary, and there have been numerous binning

strategies reported in the literature (Table 3.1). To date there has been no systematic exploration comparing these different binning strategies.

In addition to bin size and strategy, we explored how technical replicates affected clustering performance. ARISA experiments often use technical replicates to identify poor experiments and ensure fragment pattern consistency. For each intergenic fragment, technical replicates can be used to estimate the average size and, if a sufficient number of replicates are run, the variance [63,64].

To test the various parameters, we used ARISA data from a human subject time course study, for which the microbial community composition has been confirmed independently, using DNA sequencing. It was shown that the microbial community present in the human gut is clearly unique to each subject, over a time course of 60 days. Given this baseline we were able to test for those parameters that yielded the best congruence between the ARISA and DNA sequencing results. Of the parameters influencing the clustering of ARISA data, it was the clustering method itself that most affected the outcome.

Table 3.1: Summary of recent articles and the variety of bin sizes used in analysis.

| Article | Bin Size (NT = nucleotide) |
|---|---|
| Soo et al., 2009[65] | Simple bin of 2 NT |
| Popa et al., 2009[63] | Calculated fragment length based on average and variability of technical replicates |
| Li et al., 2008[64] | Calculated fragment length based on average of 3 technical replicates |
| Ramette, 2009[66] | Shifting bin method [67] |
| Denman et al., 2008[68] | Simple bin of 2 NT |
| Wood et al., 2008[69] | Simple bin of 2 NT |
| Wood et al., 2008[70] | Simple bin of 3 NT |
| Lear et al., 2008[71] | Simple bin of 1 NT |

## 3.2 Materials & Methods

### 3.2.1 Sample preparation

Microbial community analyses were performed as part of an ongoing NIH research (DK55965) study exploring the effects of common genetic polymorphisms that confer susceptibility to choline depletion. Stool samples were collected from fifteen human female subjects, who were hospitalized at the General Clinical Research Center (GCRC) of the UNC at Chapel Hill over a 60 day time course. The experimental design included placing subjects on diets that were strictly controlled and monitored for fat,

carbohydrate and protein calories and for nutrients. Five to six fecal samples per subject were obtained at specific intervals during the study .

After human fecal samples were collected and then shipped, on dry ice, to UNC Charlotte. DNA extraction from human fecal samples was performed using the Qiagen Stool Mini Prep kits. Approximately 180 to 220mg of human stool was measured for each patient per time point and bacterial DNA was extracted according to the Qiagen protocol. Approximately 180 to 220mg of fecal matter was measured for each patient per time point and bacterial DNA was extracted according to the manufacturer supplied protocol and then stored at -20 ºC until use.

### 3.2.2 ARISA Preparation

ARISA PCR was performed using universal bacterial primers 1406F-FAM (FAM+TGY ACA CAC CGC CCG T) and 125R (GGG TTB CCC CAT TCR G). Reactions were set up using 50ng of template DNA, estimated using to a NanoDrop ND-1000 spectrophotometer (Thermo Fisher). Thermal cycling as follows: An initial denaturation step at 94°C for 2 minutes was followed by 35 cycles of 94°C for 25 seconds; 56.5°C for 30 seconds; 72°C for 60 seconds. Finally, an extension was carried out at 72°C for 5 minutes. Samples were loaded on an Applied Biosystems 3130 or 3130XL genetic analyzer. Applied Biosystems GeneScan™ 1200 LIZ® size standard was used to determine sizing up to 1200 nucleotides in length.

### 3.2.3 454 DNA Sequencing

The PCR products for 454 tagged sequencing were prepared with primers, reaction conditions, and thermal cycling parameters as described in Fierer et al. [72].

The 454 Life Sciences primer B with a "TC" linker and bacterial 27F primer (5'-GCCTTGCCAGCCCGCTCAGTCAGAGTTTGATCCTGGCTCAG-3') and 454 Life Sciences primer A with a "CA" linker, 12 mer barcode and bacterial primer 338R (5'-GCCTCCCTCGCGCCATCAGNNNNNNNNNNNNCATGCTGCCTCCCGTAGGAGT-3') were used to target the V1-V2 variable regions of the 16S rRNA gene. PCR reactions used Platinum Taq DNA polymerase (Invitrogen) according to the supplier's protocol, with 100ng of bacterial genomic DNA as a template. Each reaction template was quantified using a PicoGreen assay (Invitrogen/Molecular Probes) on a NanoDrop ND-3300 fluorospectrometer (Thermo Fisher). Samples were pooled in equimolar amounts and concentrated in a vacuum centrifuge before being submitted for 454 sequencing.

### 3.2.4 Quality Control (QC) to identify poor ARISA experiments

ARISA experiments were performed on aliquots of the same DNA used to generate samples submitted for 454 DNA sequencing. A total of 214 ARISA results were generated including technical replicates. In analyzing these data, we used the simple linear interpolation method described in chapter 2 to identify peak size in the spectra. We applied the QC filtering methods from chapter 2 and removed 61 of the 214 ARISA experiments, leaving 153 sample results available for clustering. Of these 153 samples, 71 were chosen because they matched the 71 conditions used in 454 DNA sequencing experiment. When choosing between replicates, we chose the experiment with the better QC score.

3.2.5 Clustering methods

Four different clustering methods were applied to assess binning performance (average distance (UPGMA), nearest neighbors, furthest neighbors and the Wards clustering method [73]. The average distance method is the simplest way to generate distance measures between 2 clusters. The distance *(d),* is determined by taking the absolute difference between each data point bin $x_i$ and bin $y_i$ (for each bin across the spectra).

$$d(x,y) = \sqrt{\sum_i (x-y)^2} \tag{3.1}$$

The average of all the distances is then determined (average distance = $d(x,y)/N$ ). In nearest neighbors clustering, the differences between cluster's *x* and *y* are again calculated, but the smallest distance between $x_i$ and $y_i$ is determined and used as the distance. Furthest neighbor clustering (also referred to as complete linkage clustering) [74] is identical to nearest neighbor, except that the largest distance between $x_i$ and $y_i$ is used for a distance.

Wards clustering method uses an analysis of variance approach to minimize the squared differences where *(d)* is calculated with an additional squaring step.

$$(d)_{\text{wards}} = \frac{n_x * n_y}{n_x + n_y} * \Delta centroid^2 \tag{3.2}$$

$$\Delta centroid = \sqrt{\sum_i (x-y)^2} \quad (3.3)$$

The values $n_x$ and $n_y$ represent the number of branches at the cluster levels $x$ and $y$. The purpose of Ward's is akin to ANOVA where the smallest distance is determined by minimizing the sum of the squared distances (delta centroid). The clustering methods were implemented in java using a heavily modified version of ClusterLib, an open source implementation by Schulte et al. [75].

### 3.2.6 Cluster Scoring Strategy

How each bin is quantified is a choice that requires consideration. When defining the signal for every nucleotide, the data signal can be defined as the sum of all data points within that particular range. Or the data signal could be defined as the signal of the largest peak, i.e., the largest signal observed within that bin range. Other options include using the mean or median signal within a given region. We briefly explored 4 choices in bin scoring (taking the sum of all signals, maximum peak calling, median and mean) to see whether any have a pronounced effect on clustering performance. We found that, regardless of which one is chosen, the clustering outcomes were practically identical.

Due to bias in the PCR step of ARISA, the magnitude of fluorescent signal for a given intergenic fragment does not always correspond with concentration of species living in a given biological sample, i.e., the signal does not always match the relative abundance of species. To deal with this, it is has been suggested that bin scoring not be

subject to the size of the signal but rather simply to whether or not a signal is present. The Jaccard index is a binary scoring method (present versus absent) that can be used in lieu of the other scoring methods such as peak calling, sum, or median determination [67,76].

$$\text{Jaccard Index} = \frac{W}{(a1 + a2 - W)} \tag{3.4}$$

The Jaccard index is equal to *W*, the number of shared bins between 2 populations, divided by the number of bins in each population (*a1* and *a2*) that differ. The advantage to this is there is no less of a concern about bin scoring strategy. One still needs to determine the presence or absence of a peak by setting a threshold for detection. If a threshold is set too high, false negatives will occur (intergenic regions that are actually present will fail to be detected) and if the threshold is set too low, false positives will result.

DNA deep-sequencing results provide an independent, frequency-based measure of organism presence. If the sequencing is of 16S rDNA there is a natural connection to the ISSR fragments used in ARISA. The frequencies can be used as scoring metrics, to assess the influence of various ARISA parameters on the accuracy of the clustering results. UniFrac is a software tool that compares microbial communities based on phylogenetic differences, and determines if the communities are significantly different [77] [78]. Given a phylogenetic tree and an environmental condition for each leaf of the tree, UniFrac tests the null hypothesis that the pair wise comparisons between all

environments represented in the input phylogenetic tree are not significantly different. While UniFrac is usually performed on trees derived from 16S rRNA sequences, the statistic can be applied to a phylogenetic tree derived from any phenotype, including binned ARISA results.

However, we also attempted to implement two additional tree comparison metrics in order to not rely on just one way of scoring. The second metric tested was TreeDist, which computes distances between trees by calculating a "Branch Score Distance" and incorporates branch length into the calculations [79]. The third metric tested was GeoMeTree, a tree comparison algorithm similar to TreeDist, that attempts to calculate a geodesic distance between weighted trees[80]. The differences in the scoring metrics are attributable to how each algorithm handles tree branch lengths in the dendrograms.

For large clusters, the GeoMeTree implementation could not fully implement its own algorithm and therefore required us to use an approximated scoring scheme that produced results virtually identical to TreeDist (Figure 3.1). Therefore, GeoMeTree offered no additional information to TreeDist, and was removed from further investigation. TreeDist had to be abandoned for the purposes of our testing because of how TreeDist handles its branch lengths. In order to use TreeDist, the datasets in question need to be the same size in order to be a valid comparison. Figure 3.2 shows how TreeDist scores are affected by different bin sizes and the number of total bins present in each experiment. In the case of Simple Bin 1, there are 800 data points (i.e., 800 bins), while Simple Bin 10 has only 80 data points due to the larger bin size. We observed that the TreeDist scores were a reflection of the total number of data points in each bin as represented by the red line (log of number of bins for each binning method) in Figure 3.2.

Decisions made during the generation of the dendrograms, such as how one defines branch lengths, over shadows the subtle differences seen between the various binning methods. Because of this, we excluded TreeDist comparison from the final analysis.

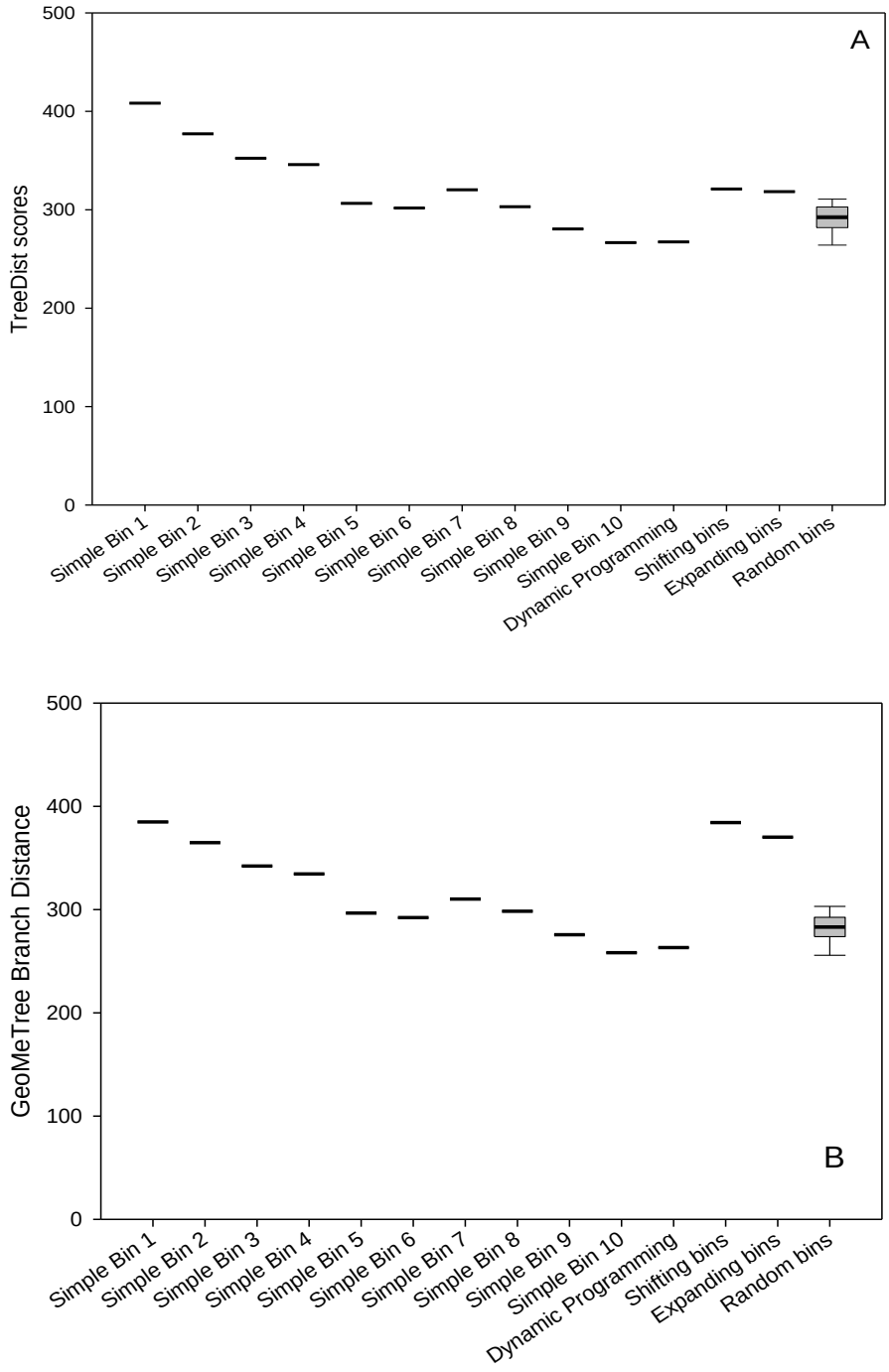



FIGURE 3.1: A comparison of scores generated by TreeDist (panel A) and GeoMeTree (panel B). For 14 binning methods and 20 iterations of random binning, scores from each metric were nearly the same.

Upon excluding TreeDist, UniFrac was the only remaining metric available as a scoring metric. For testing communities of unknown composition, using UniFrac is ideal since one can predict an outcome and then can compare ARISA clusters to the prediction without having to generate weights in their prediction. The other two scoring metrics required a weighted tree for comparison, which wasn't suited for our application. In addition, UniFrac's simple web interface made it an easier implementation and attractive option for analysis without fear of bias due to branch weighting.

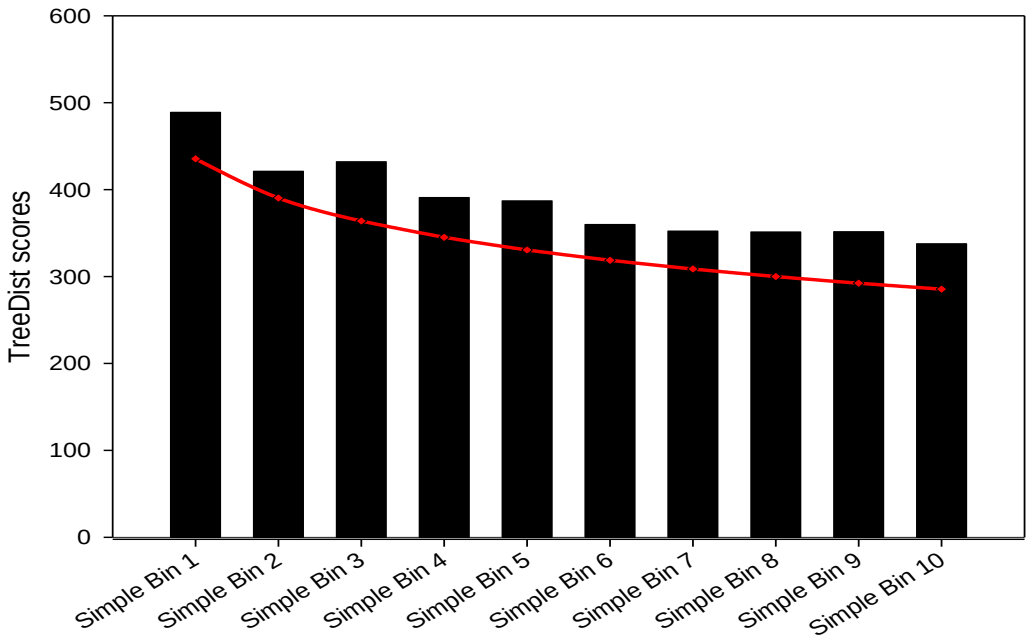



FIGURE 3.2: A comparison of Bin size and the effect of total number of bins on TreeDist scores. Black bars represent TreeDist scores for each of the binning methods. Red line represents the log of the number of bins for each binning method.

### 3.2.7 Software development

The code developed for analysis was written in Java 6.0. Each of the binning methods used was implemented in Java 6.0. Clusterlib was modified to analyze ARISA datasets (open source software available upon request). Tree viewing of clusters was performed using Archaeopteryx (http://phylosoft.org/archaeopteryx) [81], an open source phylogenetic tree viewer written in Java. UniFrac analyses were performed using a modified version of the UniFrac software [77,78] written in Python. All code used is available upon request and is available at afodor.net (http://afodor.net).

To compare ARISA clustering methods, all existing methods from the literature have been rewritten in java. This is a step that allows for quicker comparisons and ensures that the methods are correctly implemented and robust. The dynamic programming binning method by Ruan et al. [82] involves a greater degree of complexity and it's source code is freely available in R. However, the code as given is not user friendly and as written is specific to the original author's experiments. A fair amount of recoding would be required to adapt it to our experimental design. For example, the code works on small clusters ($N < 12$) but breaks on larger clustering datasets. For this reason, the dynamic programming binning method was implemented in java, based on the algorithm described in the original manuscript. For other methods such as the shifting bin method described by Hewson and Fuhrman [67], the algorithm was implemented as an Excel macro (AAAray), comparisons are made using a commercial product XLStat (by Addinsoft SARL) and neither are freely available. My revision of the above methods into java makes them now freely available to the research community via Peak Studio (chapter 4).

## 3.3 Results

A common use of ARISA is to cluster the ARISA fingerprints to determine similarities between different microbial communities. Figure 3.3 summarizes choices that can be made during the workflow for a set of ARISA experiments highlighting options (ovals) that can be made during analysis. We evaluated each of the options within an oval to determine how these choices affect the performance of clustering algorithms.

### 3.3.1 DNA sequencing

An ideal evaluation of algorithms that cluster ARISA data would utilize a dataset in which the expected outcome is known. In this paper, we take advantage of a large dataset of human gut microbiome samples for which we have both the ARISA results and the 16s rRNA sequences generated from 454 sequencing. This dataset was generated as part of a choline depletion study where patients were placed on a tightly controlled diet over a 60 day time course to study the effects of choline depletion on the body (paper in submission). All subjects within the study were placed on identical diets, stool samples were periodically collected and DNA was extracted, and 16S rRNA DNA sequencing was performed to determine how gut microbial communities are influenced by diet. Multiple time points were taken over the course of the study, before choline depletion, during and after repletion. Both ARISA and DNA sequencing results were obtained for each time point for each patient in the study.

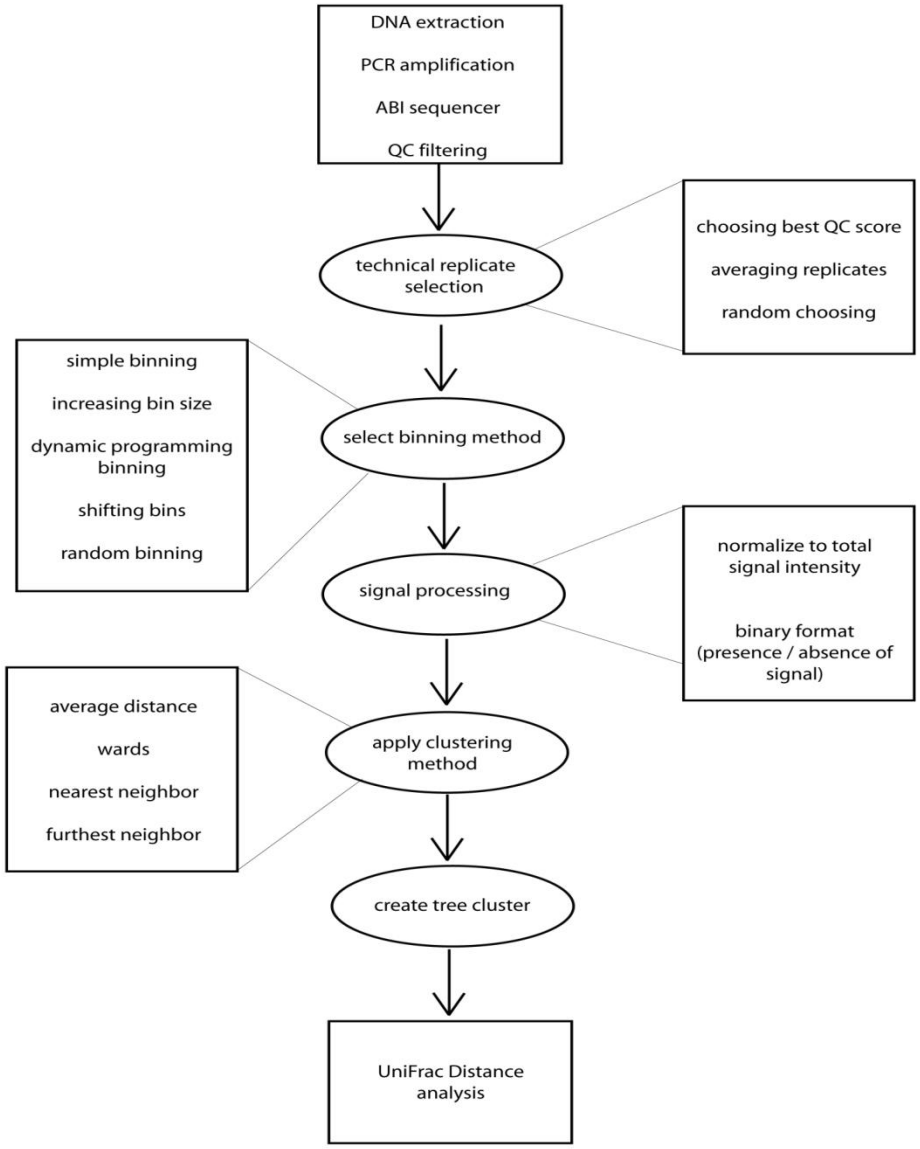



FIGURE 3.3: Workflow for ARISA clustering. DNA is first extracted from the sample in question, PCR amplified, and then fragments are separated on a genetic analyzer. QC filtering techniques can be applied to identify poorly run experiments. Data signals are converted into nucleotide length, and then converted into fractions of total intensity or binary format. Technical replicates are handled prior to binning peaks via three different strategies. Binned datasets are compared via a clustering method and dendrograms are created. Each cluster is compared to the model cluster based on 16S ribosomal gene region DNA sequencing using UniFrac. Each of the steps (ovals) has multiple options, which in this paper were tested for clustering performance.

Sequencing for all time points for each patient was undertaken using 454 sequencing technology. Primers were selected to target the V1 region of the 16S ribosomal gene, ~200,000 DNA sequences were collected and assigned to an OTU (operational taxonomic unit) with 97% similarity. The top 200 most commonly occurring OTUs were selected across the entire sequencing dataset for comparing time points and patients. For each individual time point, the number of sequence reads for each of the 200 OTUs was tabulated. All time points across all patients are then correlated with one another and clustered via Wards clustering method, in order to classify profiles and determine which time points have similar OTU profiles. Figure 3.4 depicts the results of the hierarchical clustering procedure, for all of the time points within the choline depletion study. Each time point clusters by subject and not by experimental condition. It was expected that a well run set of ARISA experiments on the same samples should match the cluster in Figure 3.4 where the time points cluster by patient.

In addition to the 454 pyrosequencing, a small subset of samples was analyzed via Sanger sequencing targeting the 16S ribosomal gene and the resulting DNA sequences were again clustered based on OTUs. The Sanger sequencing OTUs confirm the 454 sequencing results, in that the microbial communities so identified cluster by patient and not by experimental condition over the 60 day time course (Figure 3.5).

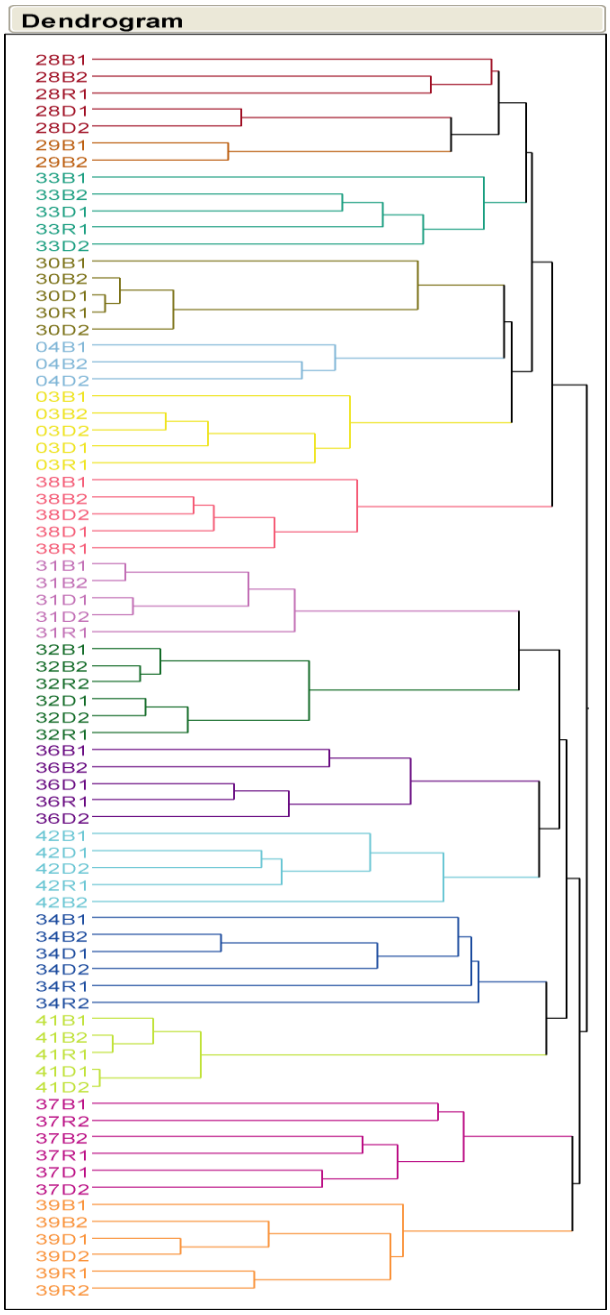



FIGURE 3.4: Hierarchical Cluster of V1 region from 16S ribosomal genes in microbial gut of human subjects via 454 sequencing. Hierarchical clustering of the top 200 Operational taxonomic units (OTUs) of DNA sequences. Clustering method = Wards.

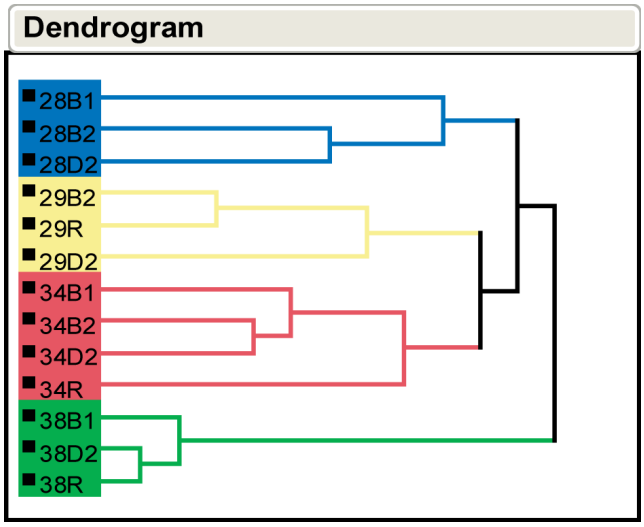



FIGURE 3.5: Hierarchical Cluster of a subset of human subject samples using Sanger sequencing: A perfect separation occurs by subject.

### 3.3.2 Technical Replicate Selection

For purposes of quality control, ARISA experiments are often run as technical replicates in which the same DNA is input into separate PCR reactions. By running replicates one can ensure technical consistency and if there are enough replicates, one can estimate the amount of variability involved in defining the intergenic fragment sizes. But it is not immediately clear how to use technical replicates in clustering analysis. Including all technical replicates can skew downstream analyses by violating the assumption of independence. For example, if a statistic is evaluating a null hypothesis that two environments have different ARISA profiles, that null hypothesis would likely be erroneously rejected if all technical replicates were included as independent samples. Treating technical replicates as an explicit factor in linear models would of course solve this problem, but in most studies, only two technical replicates are run per sample and this is an insufficient sample size to accurately estimate the within-group variance of

technical replicates. For these reasons, therefore, it is often desirable to choose just one of the technical replicates to include in further analyses. We explored three different strategies for producing a single profile from multiple technical replicates. The first strategy involves selecting the best replicate based on QC score. The second strategy averages two or more replicates together into one measurement prior to clustering, while the third strategy randomly selects one of the two technical replicates. We compared each of the three strategies by clustering the choline depletion study dataset using a bin size = 1, Ward's clustering method and each signal normalized as fraction of total signal intensity. Figure 3.6 shows the UniFrac distant scores for the three different strategies. Choosing a technical replicate based on the best QC score or by averaging together two technical replicates offers no performance improvement over randomly picking a technical replicate for this dataset.

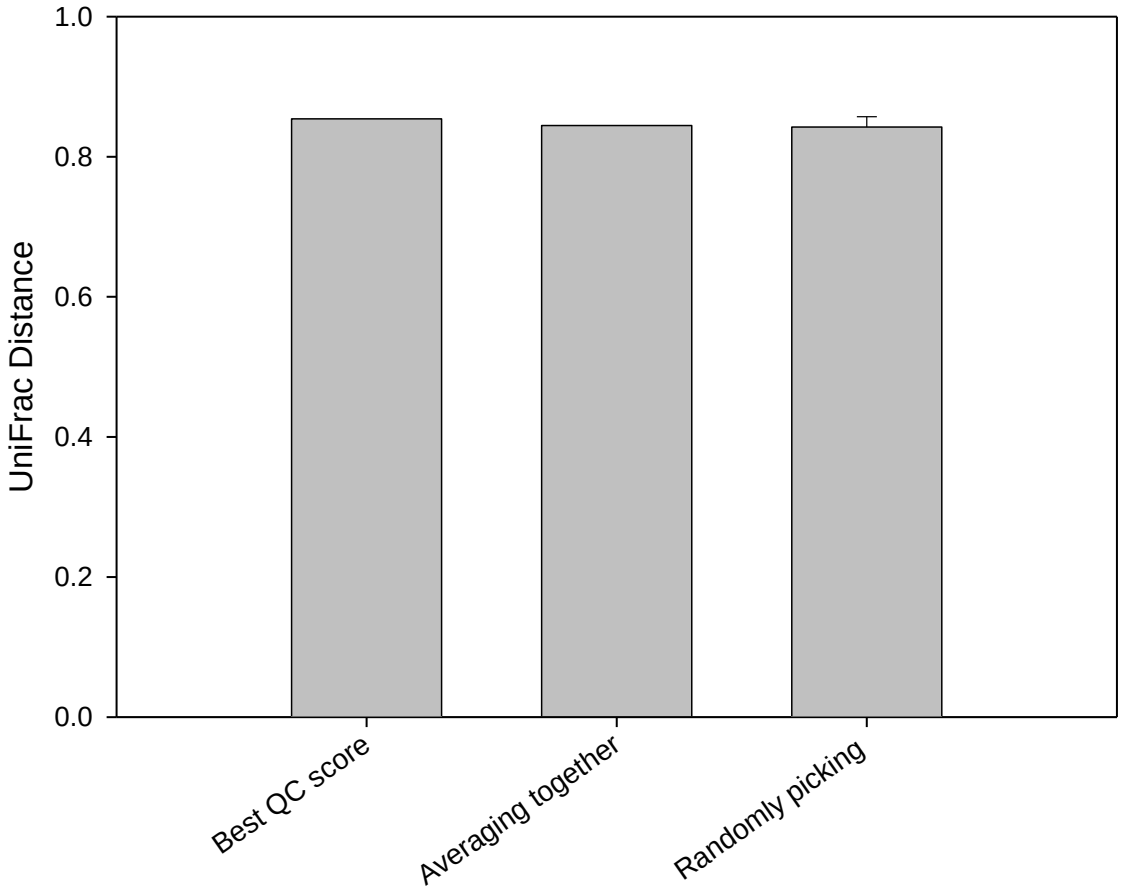



FIGURE 3.6: Comparing technical replicate strategies, using fraction of total signal intensity and Wards method and bin filling using bin sizes = 1. Error bars on the random picking strategy represent standard deviation of ten iterations of randomly picking a technical replicate. There is no significant difference between the first two strategies and randomly picking ($P > 0.4$). UniFrac score is based off the ideal clustering environment where each time clusters by patient.

### 3.3.3 Bin Size Strategies and Clustering Performance

We clustered the 71 ARISA results using a variety of bin sizing strategies to determine which binning method produces clusters that most closely approximate our observed DNA sequencing cluster in Figure 3.4. The simplest of these binning methods is to group neighboring data signal into bins and assign an appropriate nucleotide length

based on size standards (method 1 in Figure 3.7). Each bin represents different sized nucleotide fragments. Figure 3.8 depicts a tree generated using a bin size of 3 nucleotides using Wards clustering and normalizing the bins as fractions of total signal intensity.

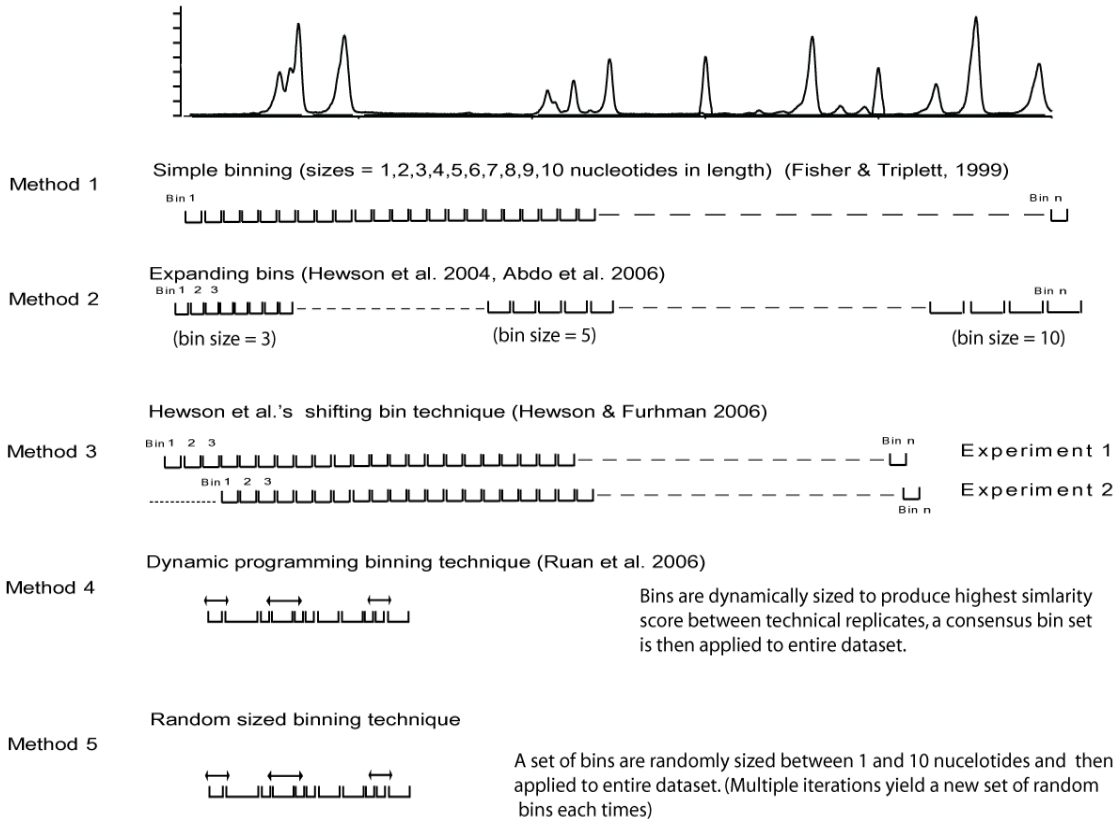



FIGURE 3.7: Depiction of various binning methods used in ARISA cluster analysis.

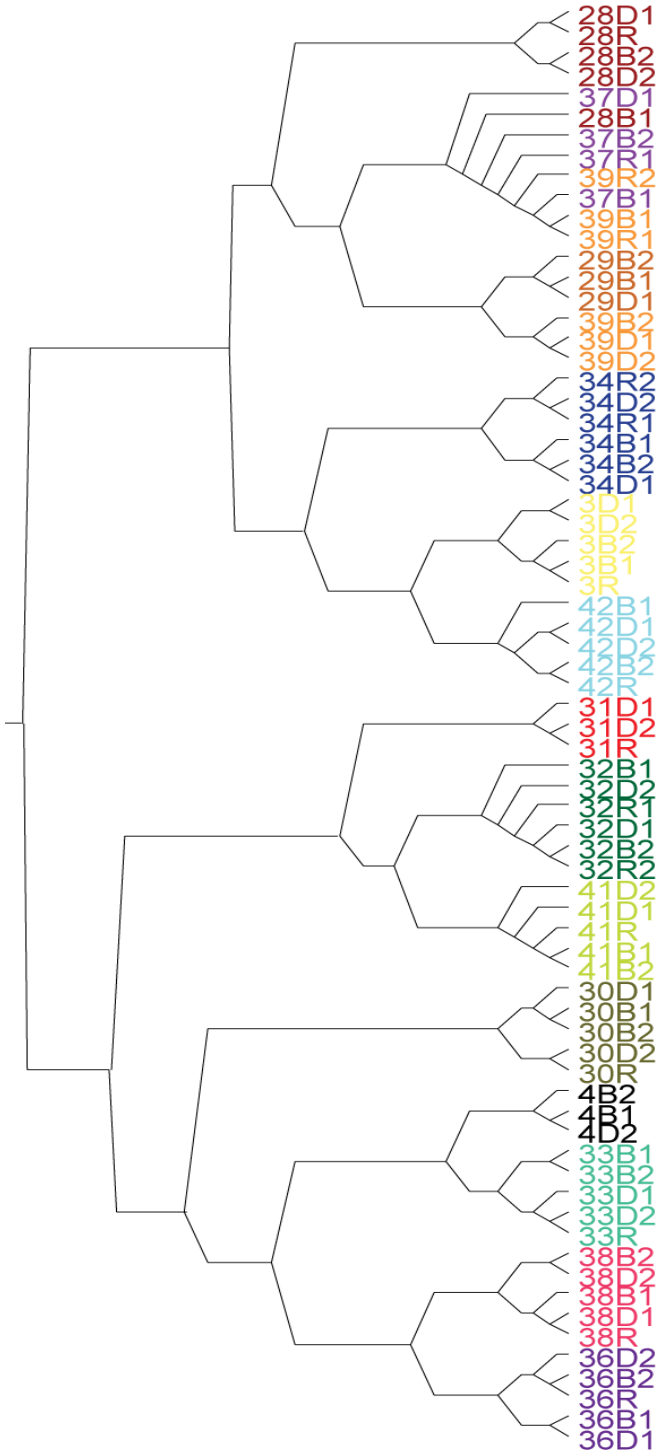



FIGURE 3.8: Hierarchical cluster using Ward's clustering method on 71 ARISA experiments from human gut micro biome (value is fraction of total intensity, bin size = 3). The ARISA cluster profile resembles the DNA sequencing OTU cluster in Figure 3.4 with a few exceptions.

A possible issue with this “simple bin” binning strategy is that electropherograms are often observed to have minor shifts in the relative position of peaks when compared to one another. This can result in bin mismatches that should otherwise be the same, especially when bin sizes are smaller. Fisher and Triplett observed size variations of 1-2 NT for fragments less than 1000 base pairs long and variations up to 13 NT for larger DNA fragments [8]. To address these inconsistencies, larger bin sizes have been used to accommodate separation medium variability and loss of precision with larger fragments[8]. A bin size of 3 base pairs or larger can accommodate small shifts across the range of the electropherograms. We will refer to all methods which use a constant bin size across the electropherogram as “simple bins”. A potential downside to these strategies is that as the bin size increases, there is a danger of grouping multiple peaks into a single bin (thereby losing resolution) and therefore we evaluated simple bin sizes ranging from 1 to 10 NT in length.

A variation on simple binning is to expand bin sizes for the larger DNA fragments to accommodate the loss of reproducibility in separation (method 2, Figure 3.7). Since there is greater accuracy for smaller fragment lengths it has been suggested that bin size = 3 NT for DNA fragments less than 500 NT, and bin size = 7 NT for DNA lengths greater than 500 [83] is a good compromise. Abdo et al. further suggest bin sizes of 3 NT from 400-700, 5 NT from 700-1000 and a bin size equal to 10 NT from 1000-1200 base pairs [84]. In both methods larger bin sizes are used for longer DNA base pair lengths. These larger bins accommodate the more pronounced drift observed with longer DNA fragments, while still allowing high resolution for the smaller base pair lengths.

Since technical replicates are commonly run as a quality control test, a further attempt to improve upon previous binning strategies was suggested by Hewson and Fuhrman [67] utilizing the technical replicates. They used a shifting bin strategy to minimize the differences observed in replicate experiments (method 3 in Figure 3.7) where an entire set of bins are shifted one nucleotide at a time and tested for similarity between replicates. Each replicate pair is compared by determining a distance metric where the differences within each bin are scored. Similar scoring bins will have smaller differences and therefore smaller overall distance scores. The bin shifting technique then shifts the data of one of the two replicates by a single nucleotide and then recalculates distance score for the replicate pair. This method repeats this shifting step for as many times as there are nucleotides in the largest bin, each time calculating scores until the best shift is found that that minimize the distance score between the replicates. Once the best shift for each technical replicate pair is determined, the *most commonly occurring* best shift among all pairs is then applied to the entire dataset prior to clustering. A potential weakness of this method stems from this last step where the most common best performing shift is applied to all the datasets. The shift could adversely affect a small subset of the experiments that would have benefited from a different shift or no shift at all.

A more recently published ARISA clustering method implements a dynamic programming strategy for binning [82]. Instead of bins of a set size, Ruan et al., attempt to dynamically allocate the bin sizes across a set of experiments. This is done again by comparing replicate experiments to one another and selecting criteria that will yield the most similar results between the 2 replicates. For dynamic programming, bin sizes are

varied on a per bin basis (ranging in bin sizes from 3 to 10 nt) for each replicate pair. The best bin size is determined for every base pair position along the electropherogram (again determined by minimized scoring distance between replicate pairs). An ideal set of bin sizes is then selected by tracing back through the best bins. The dynamic programming portion of the algorithm involves determining bin scores that minimizes the Euclidian distance between 2 replicates and the subsequent trace back [82]. Method 4 in Figure 3.7 summarizes the dynamic programming binning method. Once the best bin sizes are determined for each replicate pair, a single composite profile of the most commonly occurring bin sizes in base pair space is then applied to all the experiments in the dataset.

To assess how well the different binning strategies perform, we developed a random binning strategy that creates a series of random bin sizes between 1 and 10 nucleotides in length (method 5 in Figure 3.7). This single set of randomly generated bins is then applied to the entire set of experiments and clustering performance is assessed. Unlike other binning methods discussed here, this method can be run multiple times, generating a new set of bins each time that is then applied across all datasets. We ran each random binning method 20 times per condition and compared the results to the other binning methods.

Each of the different bin assignment methods described above was used to obtain a vector of values used as input for clustering the 71 ARISA experiments. Figure 3.9 summarizes the effect of bin size on clustering performance using Ward's clustering method. Scoring was determined by UniFrac, assigning a distance score based on how well the ARISA results match DNA sequencing clustering (score of 1 = perfect match to DNA sequencing results while a score approaching 0 represents what one would expect

from random clustering). The random bin sizing method was performed 50 times and the average score and standard deviation was calculated (far right box plot on panel's A and B, Figure 3.9). All binning methods were then compared to the random binning scores. In panel A, when data is normalized as fractions of total fluorescent signal, no binning method scored significantly better or worse than random binning ($P > 0.0019$, Bonferroni corrected). When converting data to binary scoring (panel B, Figure 3.9), a slight increase in variability is seen amongst the various binning methods but again no method was significantly better or worse than random binning. Of the 71 ARISA results, we counted the number of cluster experiments (branches) that failed to group with at least one other member in their expected environment. The smallest number of mistakes ranged from four (simple bin sizes 1 and 3) to at worst six (simple bin 5, 6 and10), meaning that at least 84% of the experimental time points clustered as expected and that the differences between the various binning methods were minor.

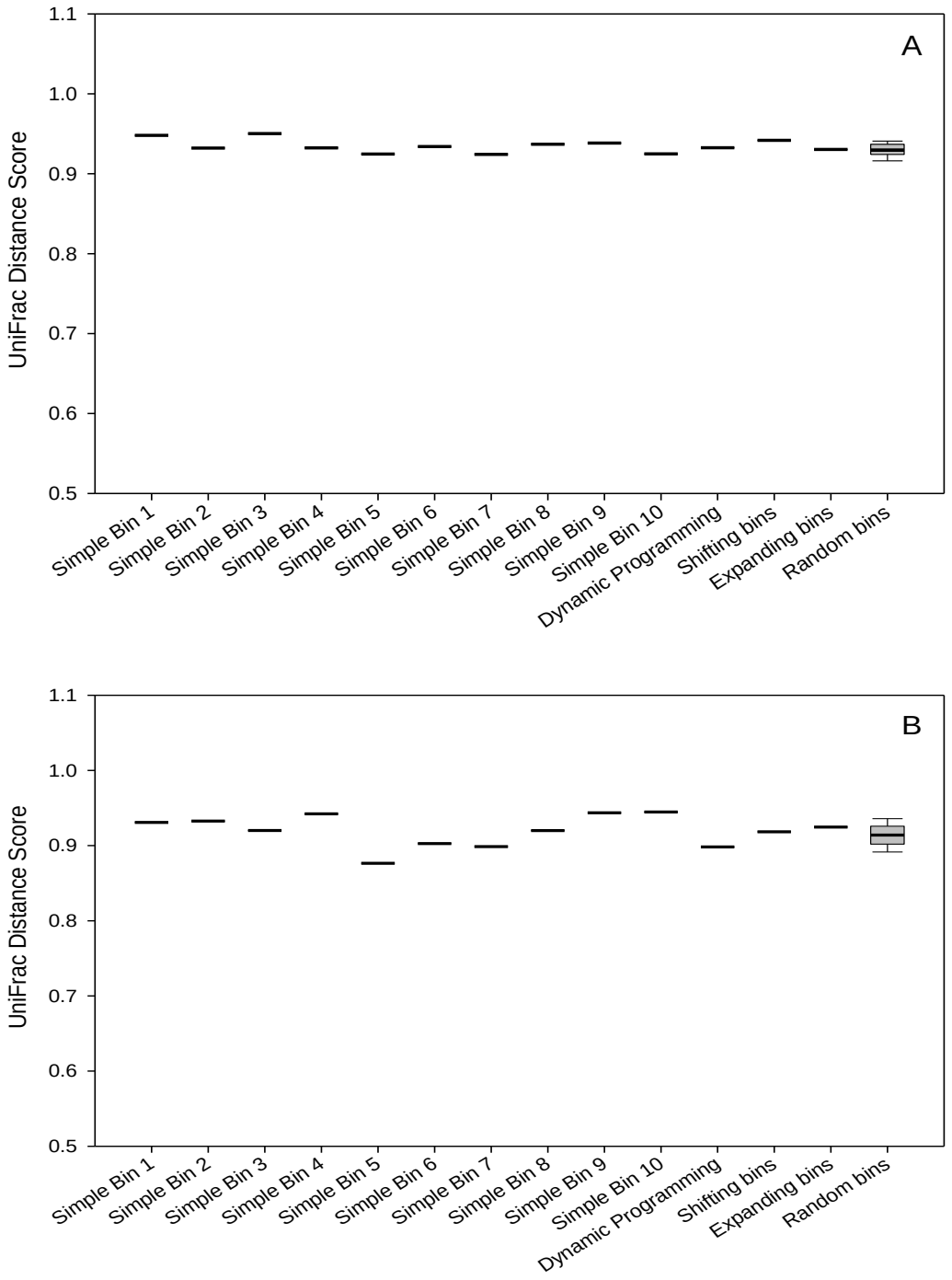



FIGURE 3.9: Ward's hierarchical cluster on 71 samples (Panel A = fraction of total intensity, Panel B = binary format). None of the 13 binning methods was significantly better than random bin sizing (Bonferroni corrected, $P > 0.0019$). Random binning was repeated 50 times to generate the box plot on the far right in panel's A and B.

### 3.3.4 Clustering methods

We tested 3 additional clustering methods on the 71 ARISA experiments (Average distance, nearest neighbor, furthest neighbor in addition to Wards). Regardless of bin size chosen, the nearest neighbor algorithm performed poorly when compared to the other 3 clustering methods (Figures 3.10, 3.11 and 3.12). The furthest neighbor and Wards methods produced consistently higher scores regardless of binning method. The average distance method was worse across all binning methods when using a fraction of total signal intensity but did show better scores when using binary format and larger bin sizes. However for random binning, the average distance method produced lower scores when using binary format (Figure 3.12).

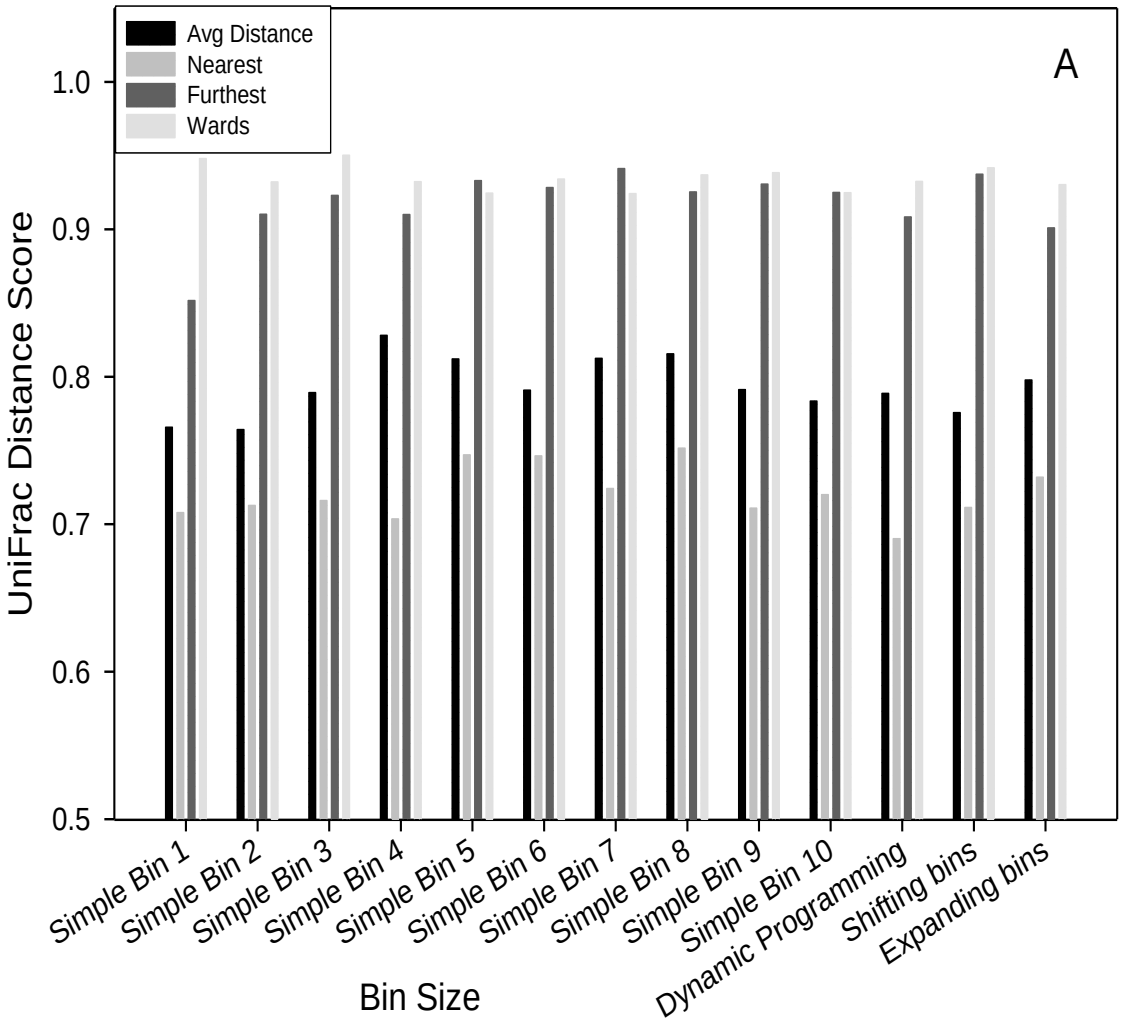



FIGURE 3.10: Comparison of different clustering methods using UniFrac. Four clustering methods were compared using non Binary format across different binning methods. Using the UniFrac metric, Wards cluster method performs best for the majority of binning methods, with furthest neighbor also performing well in most instances. Nearest Neighbor clustering method performs poorly regardless of bin size.

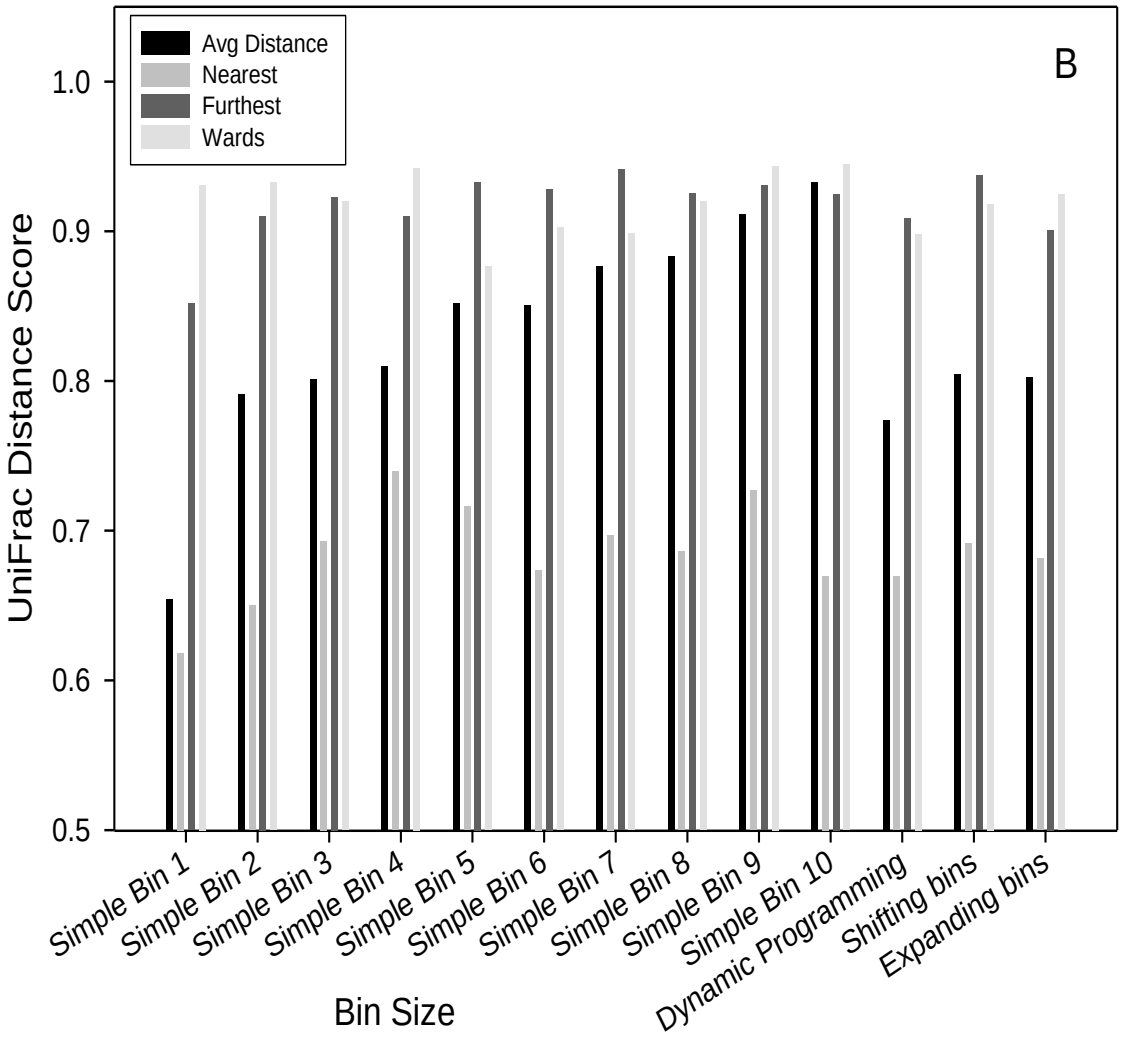



FIGURE 3.11: Comparison of different clustering methods using UniFrac. Four clustering methods were compared using Binary format across different binning methods.

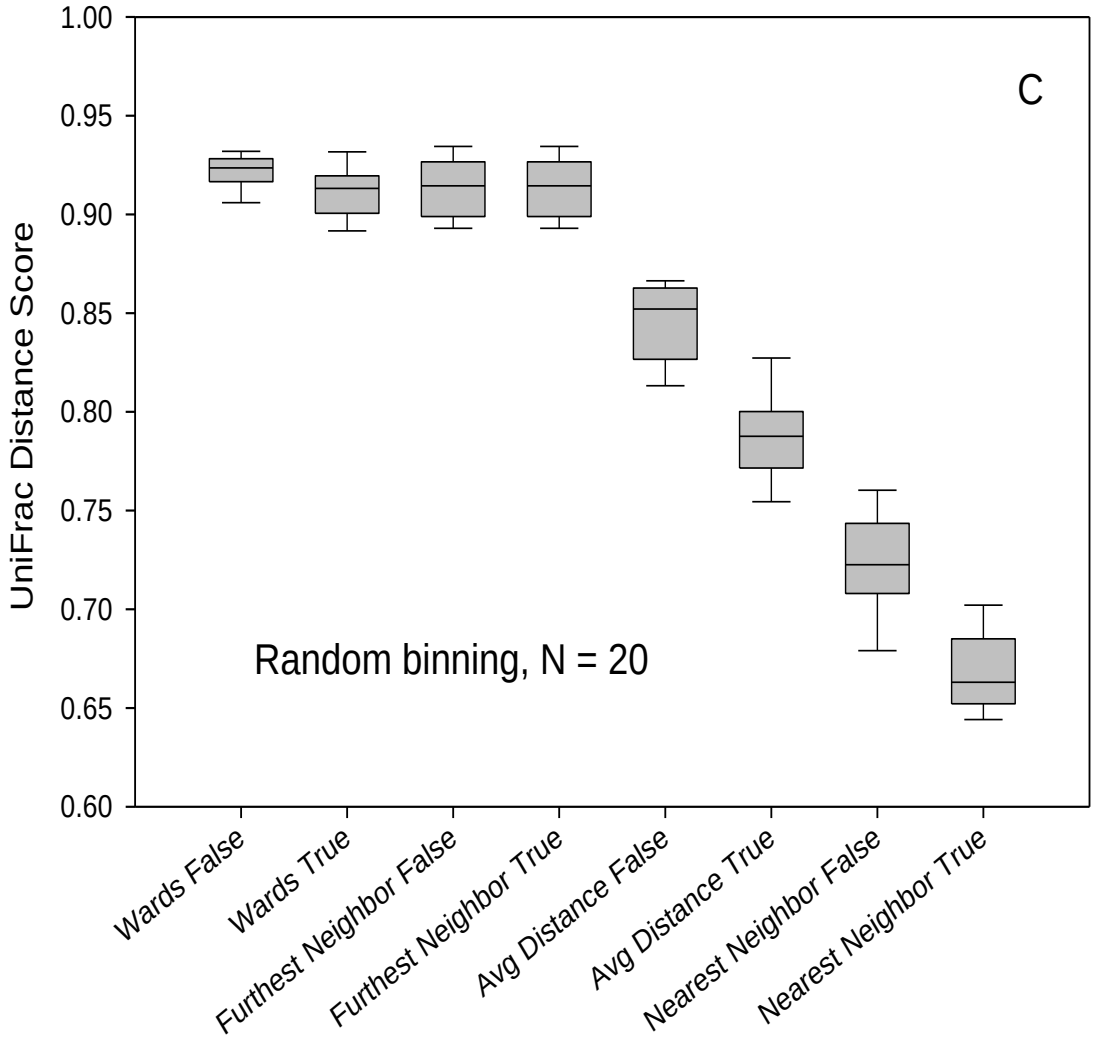



FIGURE 3.12: Comparison of different clustering methods using UniFrac. For 20 iterations of random binning, average distance and nearest neighbor methods clearly yield poorer UniFrac distance scores with binary formatting contributing to a further decrease compared to the non binary format.

## 3.4 Discussion

### 3.4.1 Parameters influence on ARISA performance

In this study we generated over 360 tree dendrograms (4 clustering methods * 2 formatting methods * 13 bin strategies + 260 random binning trials) using 71 ARISA experiments and a variety of different parameters, in an effort to create a tree that matches the result obtained by DNA sequencing. Using 454 DNA sequencing, the microbial communities are distinguishable between subjects with perfect separation (Figure 3.4). In the ARISA experiments, none of the clustering results completely recapitulated this perfect separation of the subjects. For some of the analysis paths the ARISA results did come reasonably close. Considering that ARISA is currently much less expensive than sequencing, it remains a viable option for analysis; however ARISA does not provide the same amount of resolution. We have demonstrated that the choices in parameters for an ARISA analysis can matter.

Using the UniFrac distance metric, we found that the random bin sizing method consistently approximated the DNA sequencing cluster while no other binning method performed significantly better. For a few of the binning methods, the increased computational cost and implementation time do not appear to be worth the effort. For the shifting bin method, it scored similar to the expanding bin method but requires an additional computation step to define a consensus shift. The expanding bin strategy tended to score the same as simple bin sizes that are 5 or 6 NT big. We only explored one type of expanding strategy (bin sizes of 3, 5 and 10 for specific sizes of ARISA fragments), but we expect other expanding strategies would fare about the same. For the dynamic programming algorithm, that is most computationally demanding and difficult

to implement, we observed that larger bins tended to get selected during the dynamic phase of the algorithm and as a result the dynamic programming algorithm yielded performance scores similar to a bin size of 10.

From these results it appears that the differences that might arise due to one binning method versus another are negligible. If all binning methods are generally in agreement then one can easily select the smallest binning method that will accommodate the variability seen amongst the technical replicates. With the current sequencing technology, resolution of a single base pair is highly feasible depending on the method to separate fragments (i.e., ABI genetic analyzer, or similar instrument). And for technical replicates, as long as a rigorous QC process has been used to identify poor experiments, it should make no difference how one proceeds in handling replicates.

Similar to the bin sizes, the use of binary scoring makes only minor differences to the outcome, whether one is normalizing to signal intensity or using the binary method to define the presence / absence of peaks. The binary method contains the same information as the fractional intensity method but does not take into account peak size, and that may have contributed to its poor performance when using the average distance and nearest neighbor clustering methods. If one has an interest in particular microbial species within a community that is known to have good amplification efficiency during the PCR process, then using signal intensity might be more appropriate.

Of all the decision parameters for this dataset, choice in clustering method has the most drastic impact. The Wards clustering method was the overall top performer here. Our Ward's implementation has performance similar to the findings of Mangiameli et al. where Ward's cluster outperformed the majority of other hierarchical clustering methods

tested [85]. The nearest neighbor clustering method showed degraded clustering performance, producing up to 13 mis-categorized branches within the tree, depending on the binning method (data not shown). Of all the parameters tested, the nearest neighbor clustering method was the poorest overall choice for ARISA clustering.

### 3.4.2 Clustering performance with increased noise

It has been suggested that observed performance of Ward's clustering method may not hold true for datasets that have greater noise. Milligan showed that different sources of noise and error can greatly affect the clustering performance and that Ward's clustering can be "strongly affected" by outliers in the data while a method like single linkage suffers no such influence [86]. To test the robustness of the Ward's clustering algorithm, we tested the four different clustering algorithms on the same 71 ARISA experiments while adding various levels of background noise.

The 71 ARISA experiments from the choline depletion study were used as a template to create simulated datasets with greater varying levels of background noise. For each experiment, background noise was increased using a pseudo randomly generated number between -0.5 and 0.5 (provided by java's Math.random class, uniform distribution). This pseudo random number was amplified by some constant noise multiplier (ranging from 10 to 5000) to generate a new dataset. For each new noise multiplier, a new set of pseudo randomly generated numbers were generated and a new dataset was generated. As the noise multiplier increases so does the background noise level. Cluster performance was assessed by calculating p-values derived from the UniFrac distance metric.

Figures 3.13 and 3.14 summarize how the 4 different clustering strategies score as noise increases. We used p-values generated from UniFrac and analyzed data as fractions of total fluorescent intensity. We used 6 binning methods to cover both large bins (simple bins 9, 10 and dynamic programming) and small bins (simple bins 1, 2 and 3). The influence of bin size did not appear to have any pronounced effect on UniFrac performance.

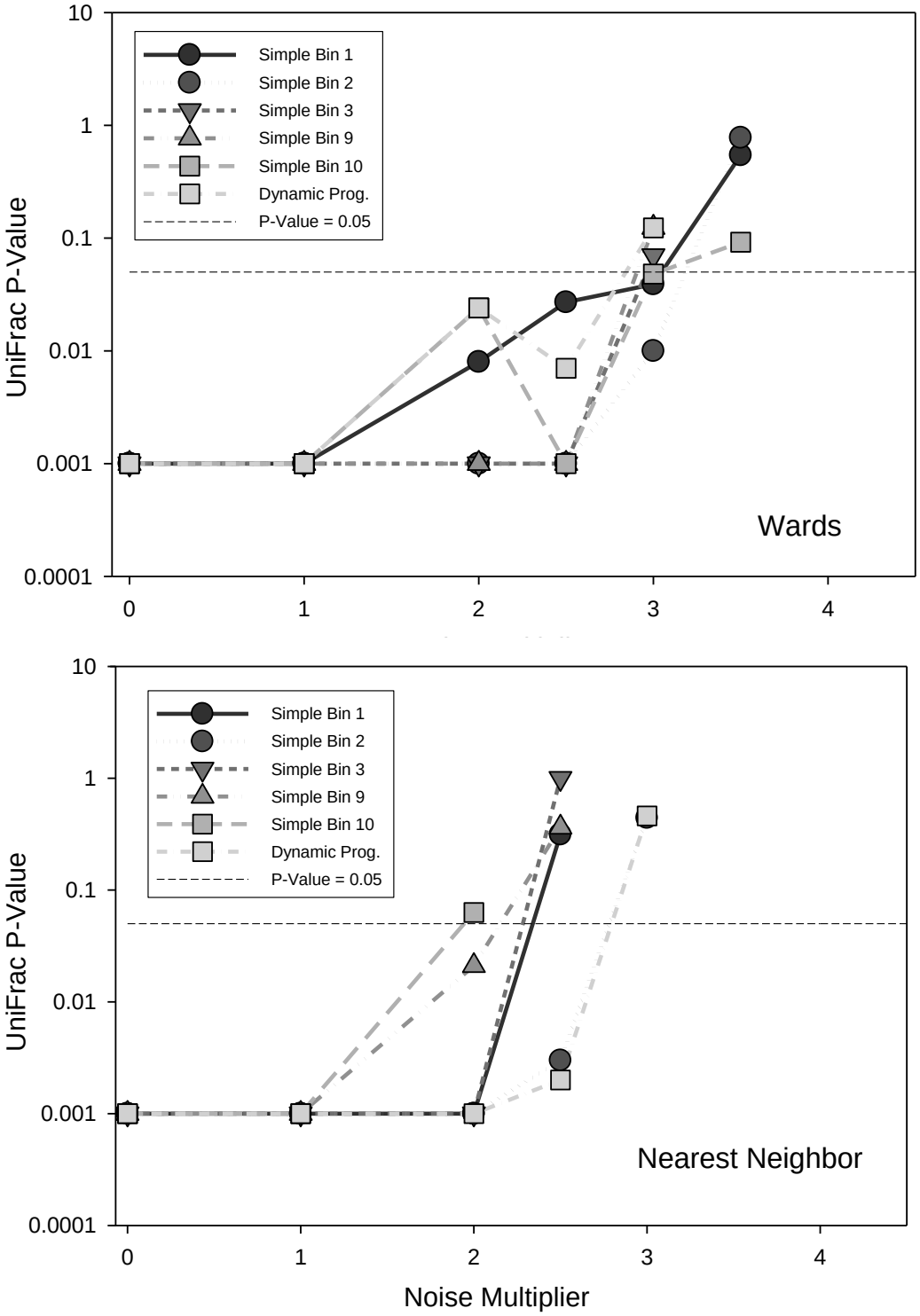



FIGURE 3.13: UniFrac P-values when adding Gaussian noise to the choline depletion ARISA dataset (Wards and Nearest Neighbor). Clustering performance was assessed using 4 separate clustering methods based on UniFrac P-value with the addition of Gaussian noise (binary format). No single binning method shows any sort of consistent performance change with the addition of noise.

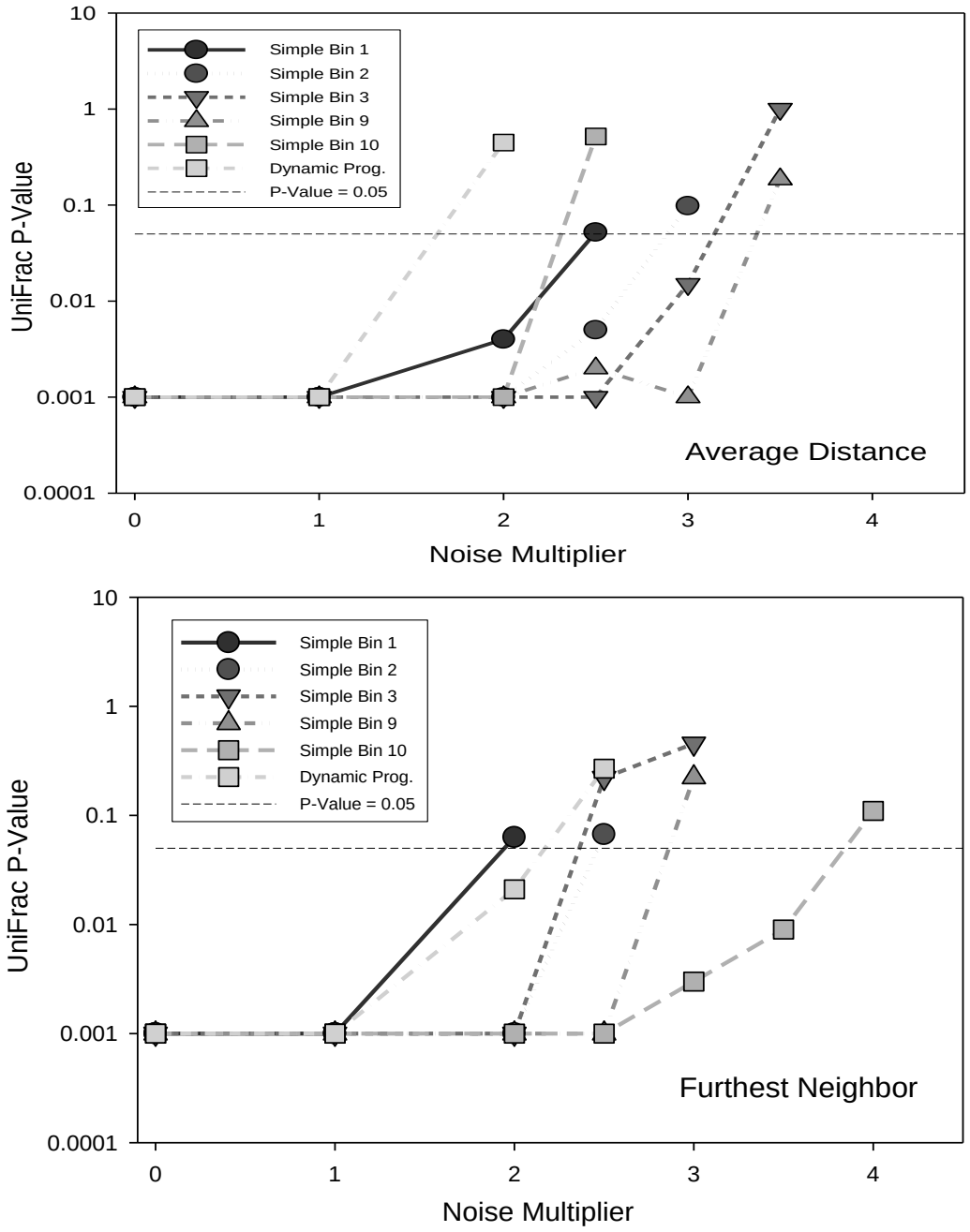



FIGURE 3.14: UniFrac P-values when adding Gaussian noise to the choline depletion ARISA dataset (Average Distance and Nearest Neighbor). Clustering performance was assessed using 4 separate clustering methods based on UniFrac P-value with the addition of Gaussian noise (binary format). No single binning method shows any sort of consistent performance change with the addition of noise.

### 3.4.3 CABS, the post binning correction method

One of the goals of this dissertation was to attempt to improve on existing methods. We tried to improve hierarchical clustering performance by including an additional "post bin" shifting step. In hierarchical clustering, each experiment is compared to all other experiments and their similarities are determined via Pearson correlation. After binning, an additional step was added that attempted to maximize the correlation within each experiment versus experiment comparison. We called this method the "Correlation Adjusting Bin Shifting" method or CABS method for short. The CABS method is similar to Hewson's binning technique (method 4 in Figure 3.7) except that we no longer applied the most common shift across all experiments but rather apply the best shift for every 2 experiments that are to be compared in a clustering process (not just technical replicates). Using any of the existing binning methods, CABS takes 2 binned ARISA datasets and shifts the entire experiment in relation to the other by 1 unit (or bin). Correlation between the 2 experiments was then recalculated and the best correlation is kept. This process was repeated, each time shifting one additional unit (or bin). Figure 3.14 summarizes how CABS was implemented. This process was repeated for each experiment to experiment comparison and the results are fed directly into the hierarchical clustering algorithm.

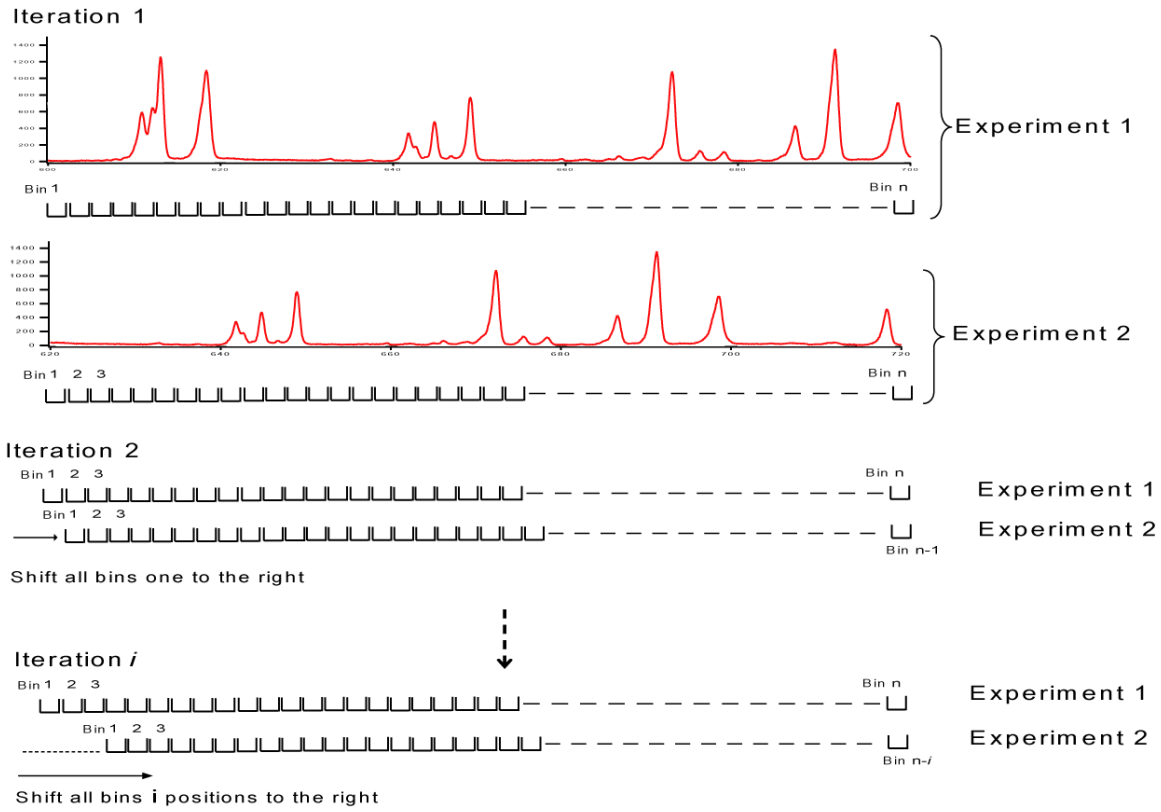



FIGURE 3.14: Summary of CABS. For the 2 ARISA results to be compared, CABS shifts the data for experiment 2 by 1 data point (or bin) and recalculates the correlation between the 2 experiments. The process is reiterated until all plausible shifts are tested and the best correlation is determined.

We expected that the CABS method should improve upon the current methods used. However, in the tests that we ran using CABS, there was a decrease in clustering performance in every instance that CABS was implemented. In addition there is a computational cost as CABS requires many correlation calculations for each experiment, therefore more processing time. We therefore abandoned the CABS method from further investigation.

### 3.4.5 Clustering using ABI's GeneMapper output

In chapter section 2.5.5, we showed that using GeneMapper identified good and bad experiments differently than our QC methods and that the parameter choices greatly influenced the results. We attempted to test how well the GeneMapper peak calling method works by exporting data from GeneMapper after size calling and binning the GeneMapper output as single bins for each of the 71 experiments used in the above analysis.

Of the 71 experiments used in the analysis, GeneMapper identified 15 experiments as poor and was unable to call peak sizes for these experiments. We clustered the output from the remaining 56 experiments to determine how well the experiments cluster by subject. Figure 3.15 shows a hierarchical cluster of the GeneMapper output using Ward's clustering method. Using the peaks generated and exported from GeneMapper, no clustering by subject is observed regardless of the clustering method used. This shows that our QC methods and peak calling are better for generating datasets as our results can closely approximate the DNA sequencing results. In contrast, using the default settings in GeneMapper yields a dataset that shows no similarity to the DNA sequencing results. In addition, GeneMapper failed to accept 15 experiments that in our hands clustered very well.

One of the possible reasons for the differences in clustering is that GeneMapper identifies a greater number of peaks in the ARISA spectra when using the default settings. The average number of peaks per spectra is 40 ± 10 for GeneMapper while our peak calling algorithm identified 32 ±10 peaks on average. The additional peaks

identified by GeneMapper might be contributing to the poor clustering outcome. Exactly how GeneMapper identifies peaks ultimately remains a mystery, as the code is not available for examination. There are clear descriptions and references in the user manual outlining how the various options work but there is no way of knowing exactly what happens to the data without some amount of reverse engineering. More thorough testing of our analytical methods versus GeneMapper would be a worthwhile future step as we expect that our methods will more accurately capture the biology behind the ARISA experiments, as we have seen in the choline depletion study.

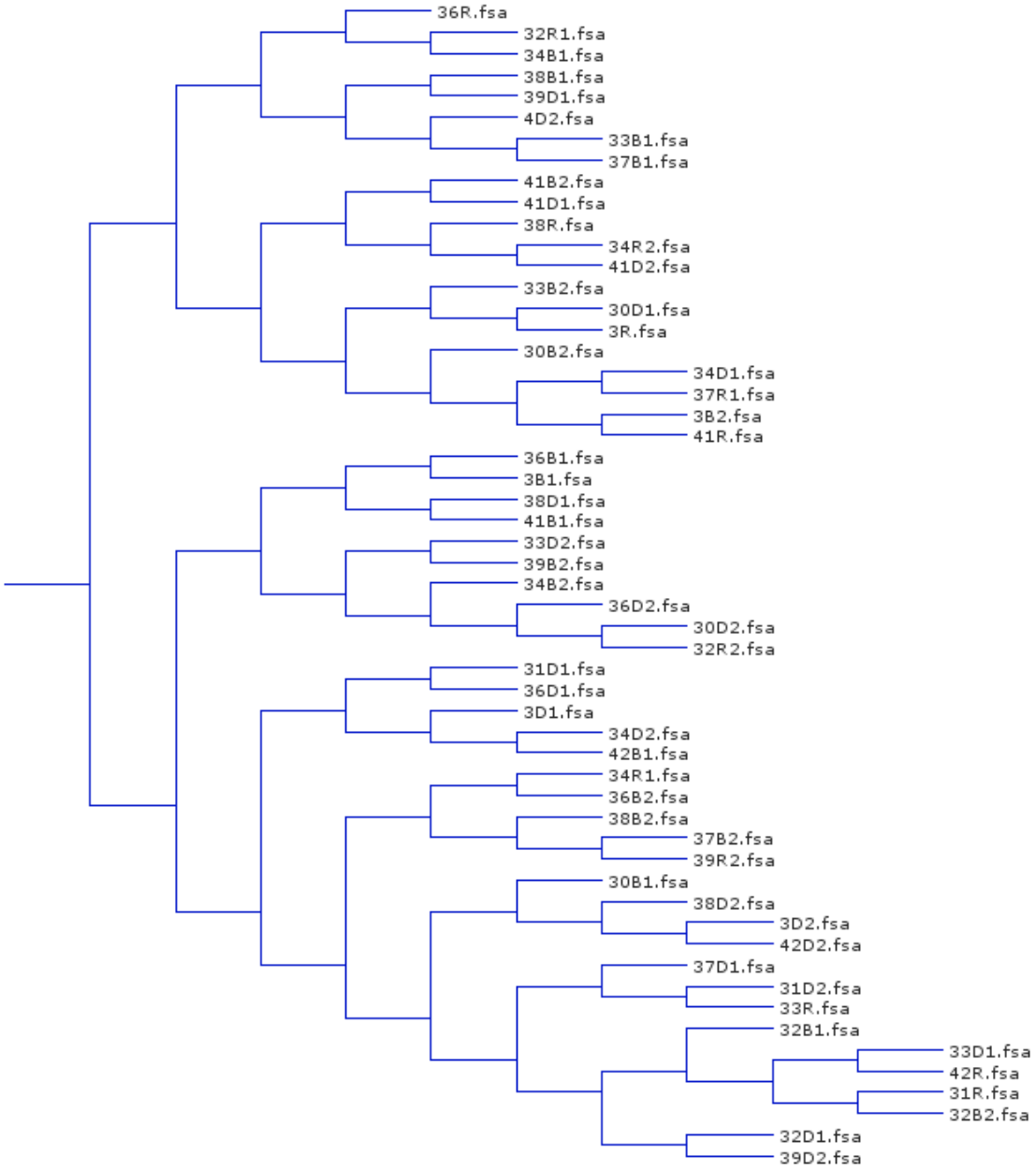



FIGURE 3.15: Hierarchical cluster of 56 ARISA experiments from human gut micro biome using the exported data from GeneMapper. No obvious separation by subject (i.e., by number) is observed regardless of clustering method or use of binary method (data not shown).

## 3.5 Summary

We explored a number of different choices one can make when clustering ARISA datasets and demonstrated that ARISA can distinguish human gut microbial communities nearly as well DNA sequencing. No set of ARISA parameters selected, however, led to the perfectly separated environment by subject clusters achieved using sequence data. We showed that bin size, for our dataset, is not an important factor and that randomly choosing different sized bins often does as well or better than previously described methods. Choices in dealing with technical replicates, and adding a post bin optimization step to the data processing pipeline, also appear to have little influence, while choices of clustering method have the most pronounced effects on clustering outcome.

# CHAPTER 4: A SOFTWARE TOOL TO VISUALIZE AND SIMPLIFY ARISA ANALYSIS

Chapter 3 explored different strategies used for clustering ARISA experiments. In order to compare the various binning methods reported in the literature, it was necessary to implement each of the methods described in Figure 3.7. The process of so doing was time consuming and would be beyond the technical scope of many labs engaged in metagenomics research. We identified a need in the research community for a simple way to analyze ARISA experiments so that they can be visualized quickly and analyzed easily. Chapter 4 describes the team development and implementation of an open source software package, called PEAK Studio, which provides biologists and ecologists with a software tool to simplify ARISA analysis.

## 4.1 Merits

Because Peak Studio is designed for use by Biologists and not Computer Scientists, it is imperative that the tool be functional while remaining simple to install and use. The goal was to develop software that an end user can run by simply downloading the software, launching the program and selecting the data they wish to analyze. Our primary goal was to combine simplicity and functionality when viewing and analyzing ARISA data. This software package is an open source project written in java and made freely available online. The development of the code was a team effort within the Fodor lab, with Jon McCafferty and this author being the primary developers. My primary role

in the project was the overall project design and the integration of the analytical tools for clustering into the software package.

## 4.2 Outline and preliminary data

### 4.2.1 Design Document

One of the first steps in the Peak Studio development process was to create a design document that governed the layout and defined the desired features of the software tool. This document defined the scope of the project and identified key components that were to be added in the first phase of the project. Features and attributes were identified and classified into 1 of 2 categories. The most critical aspects of Peak Studio were included in phase 1 and future desirable features were to be added later. The components described in this dissertation were all part of phase 1. At the end of phase 1, Peak Studio would be a fully operational ARISA viewer with the added ability to cluster many spectra using the options defined in Chapter 3. Prior to the start of phase 2, an application note will be submitted for publication that describes Peak Studio, with Jon McCafferty as primary author. Since this project was a collaborative effort, the design document aided in maintaining team member focus and allowed each individual contributor to know what they were responsible for so that no overlap in coding would occur.

### 4.2.2 Use Case Diagram

Figure 4.1 shows a use case diagram with many of the features currently implemented in Peak Studio. My contribution to the project involved each of the red ovals in Figure 4.1. The majority of my work focused on implementing the cluster analysis for ARISA experiments. A user can load the ARISA files in the .fsa binary file that is generated by the ABI genetic analyzer. They can then view the electropherograms,

check their QC status, based on size standards, and decide which experiments to use for clustering. The user can then select one of the 112 different ARISA cluster parameter combinations from a user friendly dialog box and rapidly perform a hierarchical cluster using ClusterLib[75] with a simple click of a button. From each clustering result, the user gets a tree output file in Newick format and can view a visual representation via the Archaeopteryx software package (which is incorporated into Peak Studio) [81,87]. The user can then export either the ARISA spectra or tree cluster results as images. Peak Studio also provides the user with the option to launch Archaeopteryx directly in order to view previously generated trees.

### 4.2.3 Data input

The primary data input are .fsa binary files generated from the Applied Biosystems sequencing software suite (AB DNA Sequencing Analysis Software V 5.2). The end user can opt to choose a single file, or multiple files. Sample binary files will be provided as part of the software download package. Other formats are not currently incorporated within the scope of the project but may be added later if need or demand warrants it.

Upon selecting the files, the user needs to select an appropriate size standard text file that corresponds to the size standards used in the ARISA experiment (large red arrow, Figure 4.2). The size standard text file lists each of the size standards in ascending order and is used to assign a DNA length to each spectral peak in the size standard electropherograms (such as the peaks in panel B of Figure 4.3).

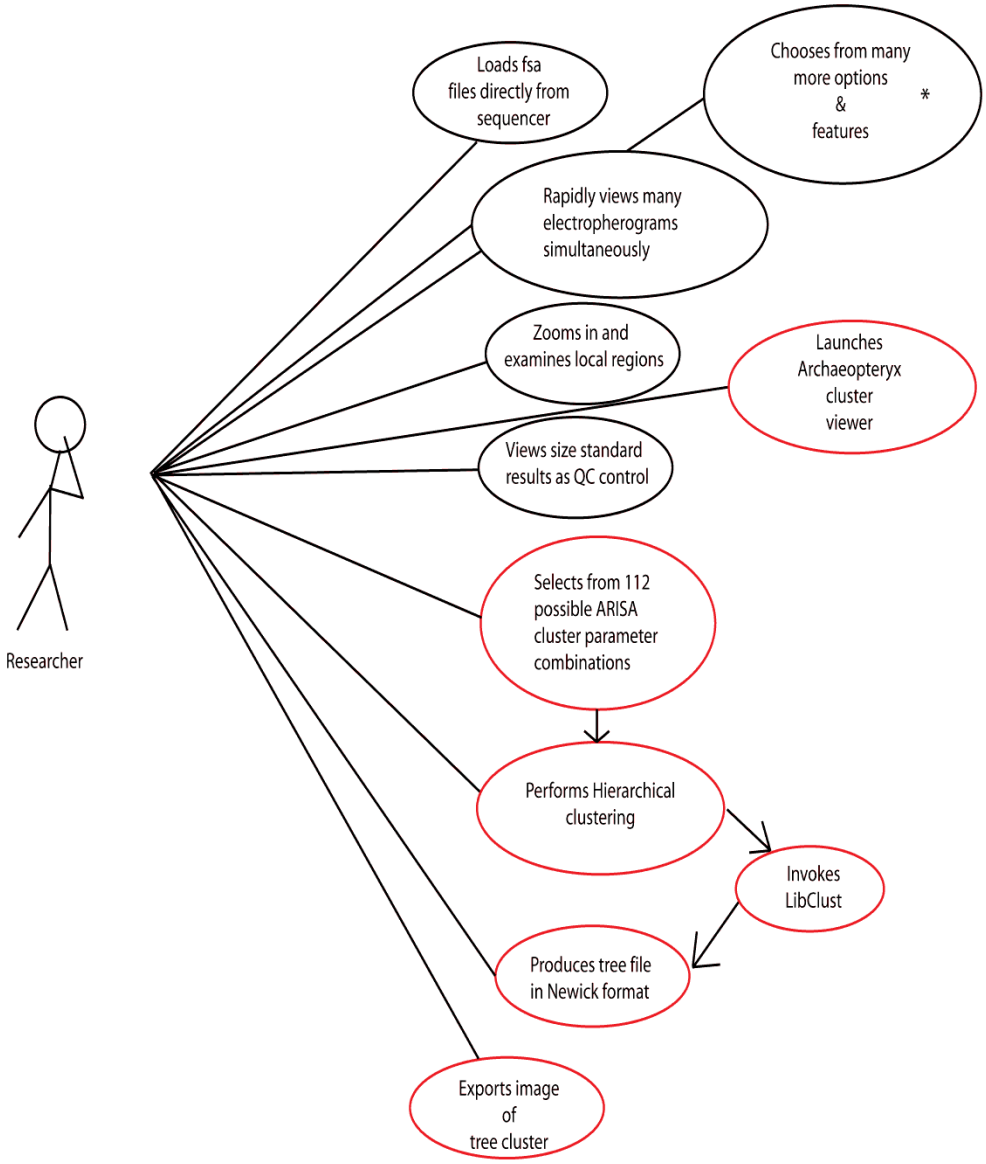



FIGURE 4.1: Use Case Diagram for Peak Studio. Black ovals represent Peak Studio components that were implemented by Jon McCafferty. Red ovals depict components of Peak Studio implemented by Rob Reid. *Numerous other details are featured in Peak studio but not depicted here.

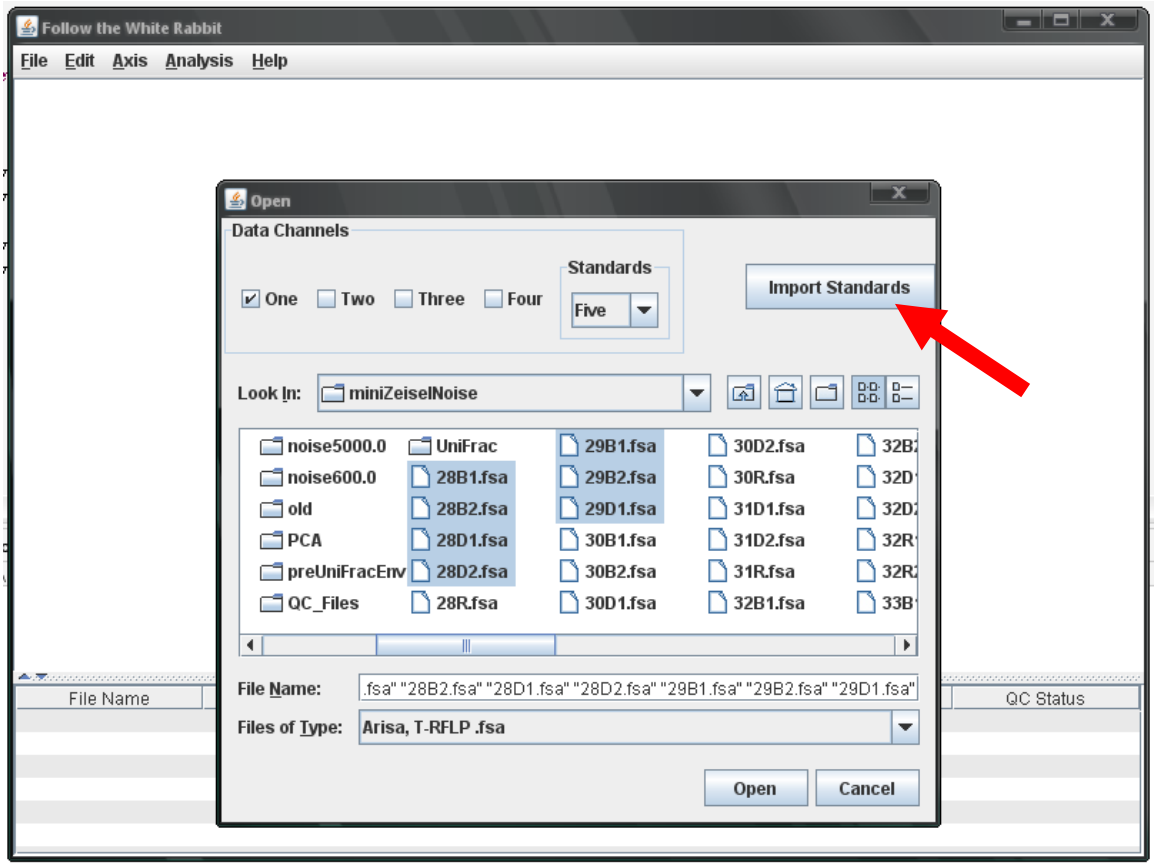



FIGURE 4.2: A depiction of the file selection menu of PeakStudio. Designed and written in java by Jon McCafferty. Each fsa input file has an .fsa extension and is the binary output file from the ABI sequencer. An "import size standard" option is available for the end user to choose an appropriate standards file that corresponds to the files selected (red arrow).

### 4.2.4 Visualization

Figure 4.3 shows an example of the GUI interface in Peak Studio. Spectra of three ARISA experiments show how one can rapidly compare and contrast experiments. The viewer has a number of features including zooming, changing color, resizing and displaying size standard spectra. The user has the option to toggle the visibility any of the experiments and can view both the ARISA data spectra (panel A, Figure 4.3) and the size

standard spectra (panel B, Figure 4.3). The user can also rapidly identify the lengths of prominent speaks by mousing over the region of interest.

As part of visualization, the ARISA viewer is able to zoom into and out of regions of interest. The user will have the option of saving a screen capture of the visualization in a number of popular image formats. From the table in the bottom panel of Peak Studio, one selects the desired experiments that are to be used for ARISA cluster analysis.

### 4.2.5 Clustering

Once the experiments are chosen, a cluster analysis can be executed. The user chooses “Analysis” from the top menu and selects the “ARISA_Cluster” option. This opens the ARISA Cluster Options dialog box where the user chooses which options they want for clustering (Figures 4.4 and 4.5).

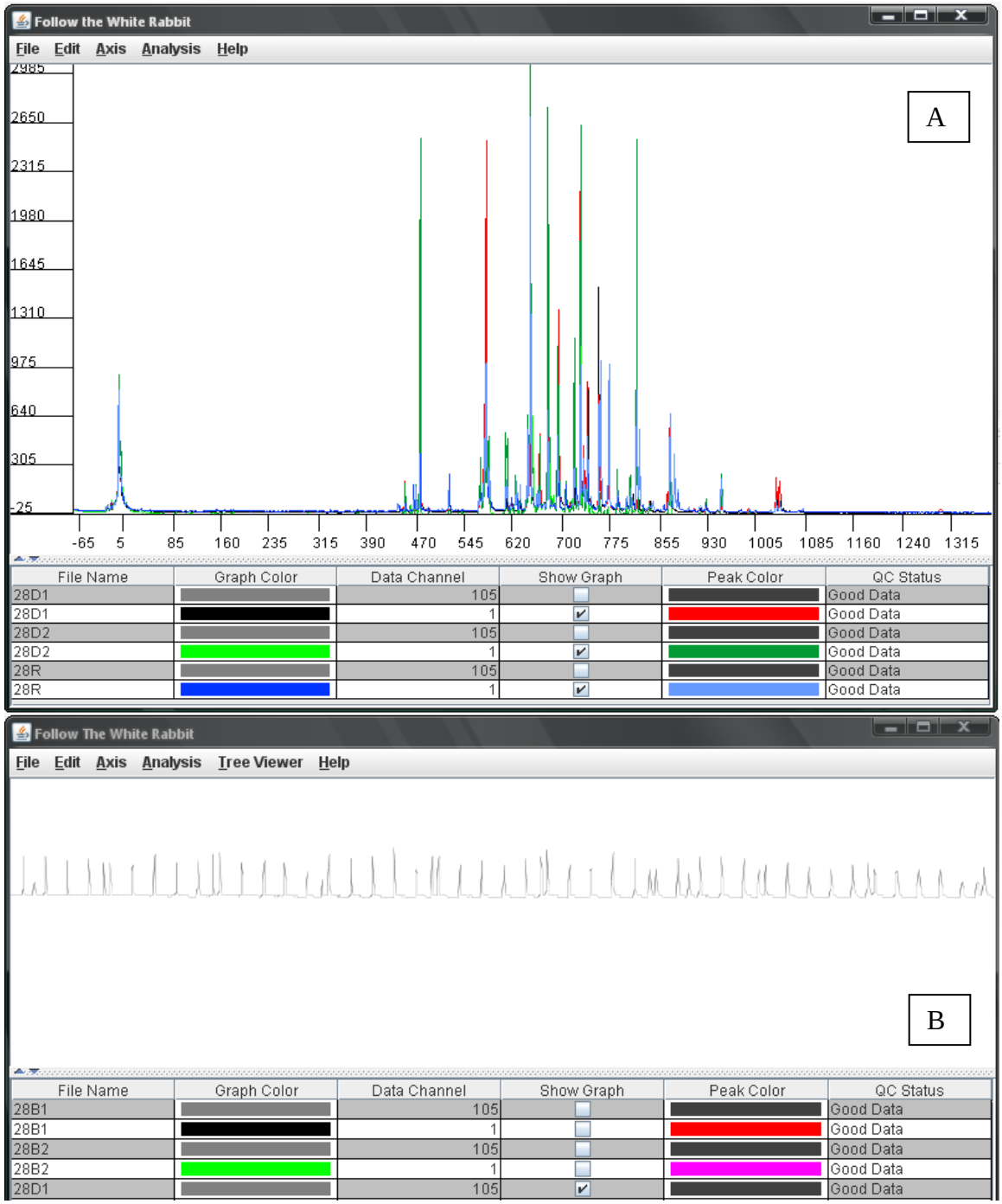



FIGURE 4.3: Depiction of Peak Studio. Designed and written in java by Jon McCafferty. (A) ARISA spectra from a human microbial community are superimposed and color coded. Options include custom colorization, toggling ability to display individual graphs, and the option to toggle size standard spectra. For a given spectra, peak and background can be distinguished by different colors. (B) Display of size standard spectra for 1 of the ARISA experiments.

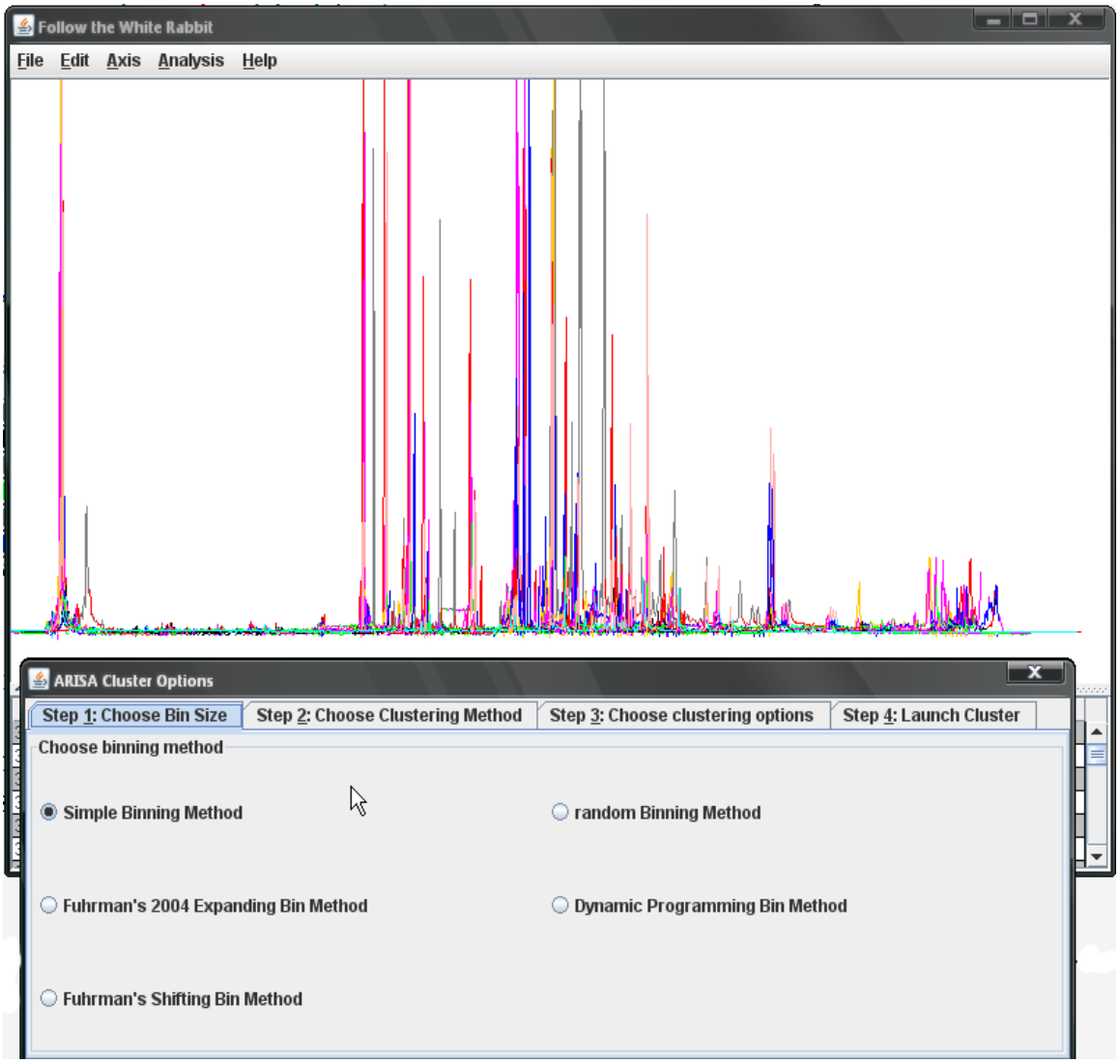



FIGURE 4.4: Dialog box for ARISA cluster analysis. Each of the options is broken down into tabs for easier selection. Step 1 involves choosing 1 of the 5 different binning methods.

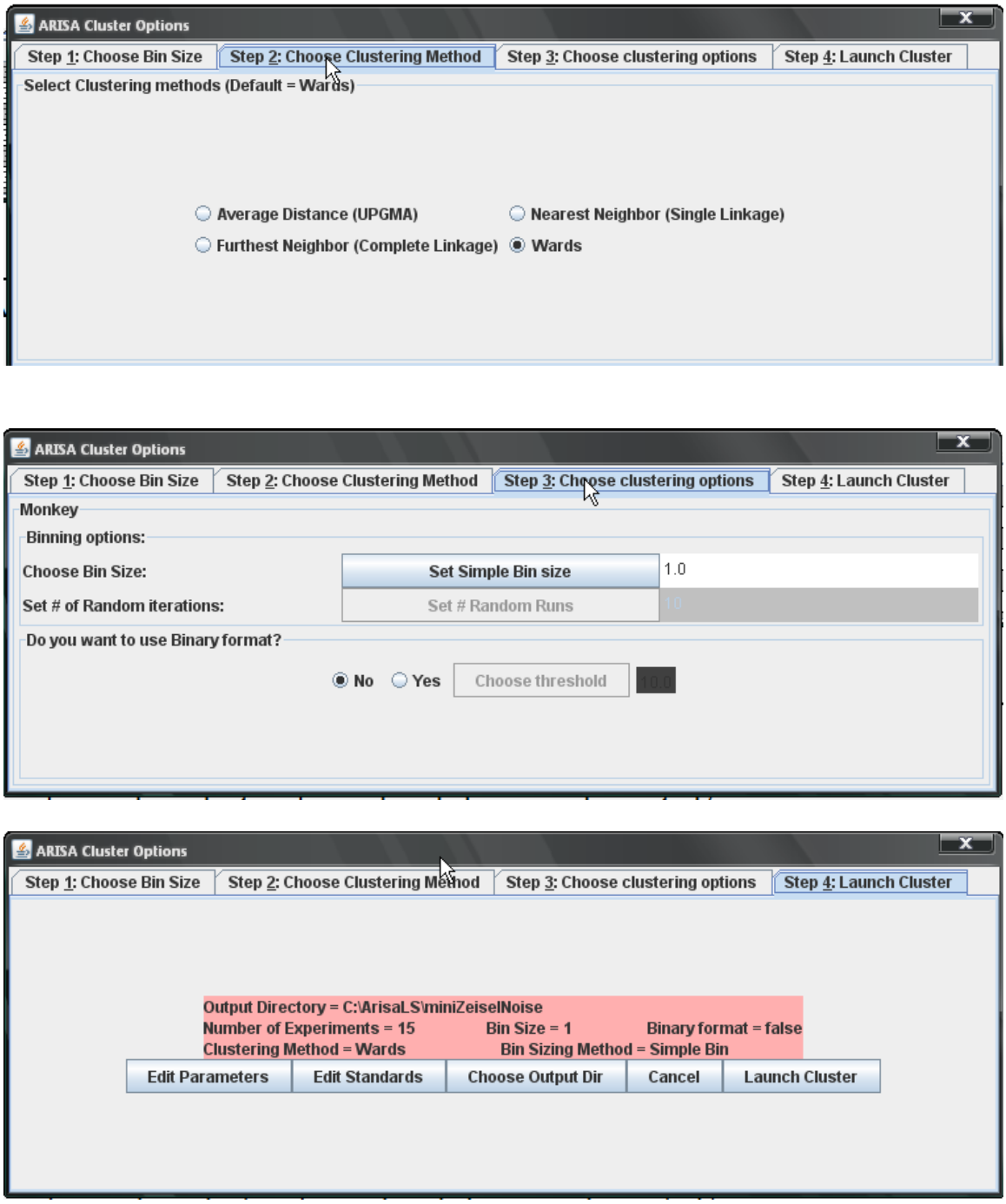



FIGURE 4.5: Different views of each tab for the ARISA cluster analysis dialog box. The upper, middle and lower panels show the various options involved at each step of the process.

In Figure 4.4, the first tab shows where the user chooses a binning method to use for analysis. The upper panel of Figure 4.5 shows the second tab where the user selects the type of clustering method. By default, the choice is Ward's clustering, based on the results from Chapter 3. The middle panel of Figure 4.5 allows the user to choose whether or not to use binary format, in addition to some bin size options. If the user has chosen simple bins, here they choose what size bin to use. If using random binning, the user can choose how many random bin runs they want to run. Boxes that are not relevant to the binning method chosen are grayed out and not selectable. As can be seen in the middle panel of Figure 4.5, the random binning selection is not available because random binning was not selected as the binning method. The lower panel of Figure 4.5 shows the last tab of the ARISA cluster dialog. Here the user can start the analysis, make changes to some of the parameter settings, edit the size standard settings, and select the output directory for the Newick tree output. If an insufficient number of experiments are chosen, the "Launch Cluster" button is grayed out and unavailable until the proper number of experiments is chosen. The pink text box summarizes the binning choices made and shows the number of experiments to be clustered. Once the desired choices are made, and a sufficient number of experiments are chosen, cluster analysis can be launched and a tree file in Newick format is made.

### 4.2.6 Tree cluster visualization via Archaeopteryx

Upon clustering ARISA spectra, it is preferable to visualize the dendrogram as well as provide the data. For visualization, I implemented Archaeopteryx, an open source freely distributable java package that is designed for visualizing and annotating phylogenetic trees [81,87]. Archaeopteryx is an adaptation of a class of libraries known

as Forester, which was a java based tool first developed for visualizing complex phylogenetic trees [81]. The authors of Forester have been granted permission to use and modify the code so long as we adhere to the licensing agreement, not use the code for commercial gain and make the source code accessible.

Archaeopteryx reads in tree cluster files in the Newick format and then generates a customizable visualization of the tree with numerous options. In Figure 4.6, a dendrogram is depicted using a subset of the human subject ARISAs described in chapter 3. We chose a  customized color setting of a blue tree on white background with the option of branch lengths being drawn according to length. For the 3 subjects in Figure 4.6, we can clearly see three distinct clusters without any further need to tweak any of the available options. Archaeopteryx also provides a number of export options that we get for free, including the export of PDFs, jpegs, PNGs, GIFs and BMPs.

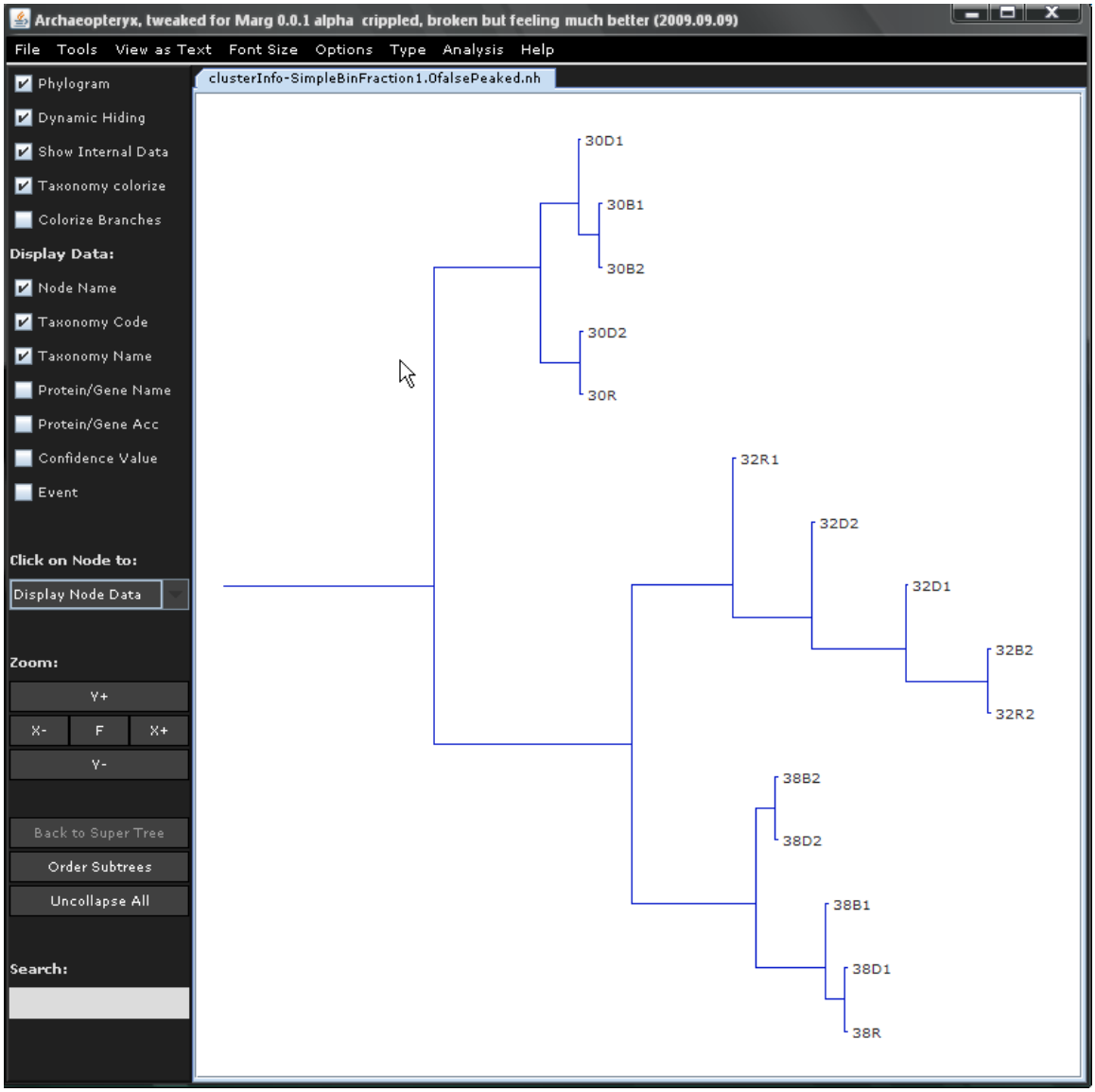



FIGURE 4.6: Depiction of Peak Studio's implementation of Archaeopteryx. We fail to utilize much of Archaeopteryx's phylogenetic functionality but do make it available for users if they so desire.

### 4.2.7 Quality Control

A quality control check was implemented in Peak Studio that attempts to identify whether the appropriate conditions are met for further analysis. Assuming a user has selected a size standard file to associate with the uploaded file or files, Peak Studio checks that the number of expected size standard peaks matches the actual number. In order to do so, all of the peaks need to be distinguished from background noise. For each potential size standard peak in the spectra, a number of parameters need to be satisfied in order for it to be labeled as a size standard peak. These parameters include: meeting a minimum height threshold, having appropriate rising and descending slopes and ensuring that the rising slope and falling slope of the peak are within a specified distance of each other. If the number of peaks identified fails to match the expected number of size standards, a warning window is displayed as in the left panel of Figure 4.7. The QC status in the data table will then reflect the error (right panel, Figure 4.7).

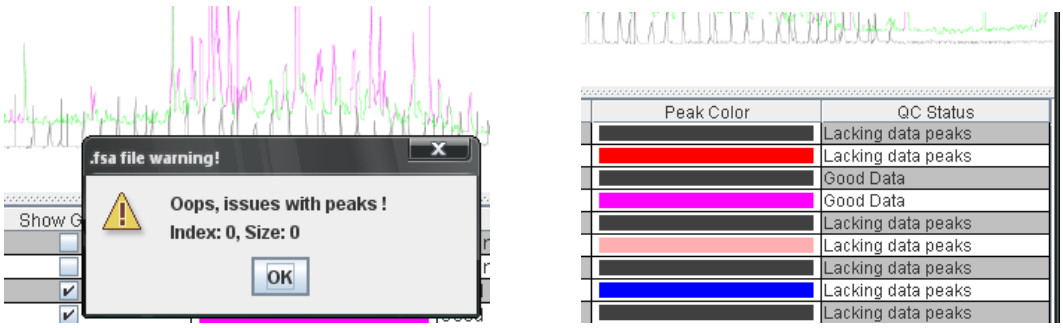



FIGURE 4.7: Example of Peak Studio output when failing QC check. Left Panel: A warning message is displayed when a loaded file fails to match the parameter settings. Right Panel: The QC status in the table displays a short message pertaining to the QC status of the loaded file. The file can still be viewed in the viewer but no further analysis is possible.

The Quality Control section was created by Anthony Fodor, Jon McCafferty and this author. The quality control check of each peak was written by Anthony Fodor, while Jon McCafferty and this author integrated the QC code into Peak Studio and tested.

### 4.2.8 User Access

The development of Peak Studio has approached the end of phase 1 in the development cycle and is fully functional. The package has been released to the public in its current form under the GNU license. The code is available for use and for further development at the SVN repository located at:

https://peakstudio.svn.sourceforge.net/svnroot/peakstudio

The repository is part of the Peak Studio project hosted by sourceforge.net at:

http://peakstudio.sourceforge.net/

Users are currently welcome to download the Peak Studio jar file, test it out or run their own set of data. Developers are also welcome to join in the Peak Studio development.

## 4.3 Summary

Our goal with Peak Studio was to provide an easy to use software package that allows users to fully visualize ARISA data and generate hierarchical clusters. Peak Studio provides users with the ability to simultaneously view many ARISA spectra and easily produce hierarchical clusters with many cluster options. Users can cluster data using a choice of binning strategies and different clustering methods, as well as rapidly view the

results of these clusters. Now that Peak Studio is accessible to the public, the benefits of the ARISA implementations from chapter's 2 and 3 are immediately accessible for all to use. These implementations have taken the better part of 2 years and specialized expertise to develop, therefore they are beyond the capabilities of many research labs. This contribution to the research community hopefully will aid many scientists in the ARISA analytical process.

## 4.4 Conclusions and suggestions for further work

The central theme of this dissertation was to solve problems that arose from complex and large datasets. In chapter 1, we described PINC, a microarray analytical method that can be applied to low replicate experiments. Microarray experiments can produce many 1000s of results, but are not necessarily performed with many replicates due to significant costs. PINC is particularly useful when microarray experiments have less than 3 replicates. The majority of other analytical methods are based on assumptions that require larger sample sizes, while, for the methods that do allow smaller numbers of replicates, they failed to perform as well as PINC. An additional benefit of PINC is that we were able to develop a way to estimate biological noise in microarrays, because PINC allows us to do single chip to chip comparisons. By performing many of these single chip comparisons and using the knowledge that technical replicates tend to be consistent, we can identify where variability is most likely caused by biological sources.

Future work in microarrays would involve expanding the PINC's functionality to multiple microarray platforms. Currently, the PINC software package is only suited for a small range of Affymetrix gene expression chips, limiting its scope of application. In addition, PINC is not as user friendly as it could be. PINC requires a user to manipulate properties files, install the R statistical software package and have some knowledge of how to install packages in R.

Shifting away from microarrays, chapter's 2-4 focused on how to best analyze data from ARISA, the molecular marker technique that produces highly reproducible, DNA fragments. ARISA is used to identify the length of DNA fragments extending between the 16S and 23S gene regions for all bacterial members of a microbial

community meeting primer sequence compatibility and PCR product efficiency conditions. One of ARISA's primary uses has been to compare communities to one another, often by way of hierarchical clustering. While less complex than microarray output, ARISA experiments produce a large volume of multidimensional data, which often raises questions about how to best go about analyzing and comparing ARISA experiments. In the literature, multiple methods have been described to process and explain ARISA results, but there has not been a systematic, standardized comparison between these methods to determine which perform best. We therefore, opted to explore a number of these different ARISA analytical methods in the context of clustering performance.

Chapter 2 focused on the processing of ARISA data so that spectral peaks in an experiment can be accurately identified and sized. By doing so, the sample peaks in a spectrum, (which each correspond to one or more DNA intergenic fragments, from which we infer the presence of some microbial species) are reliably identified and therefore more likely to represent a real characteristic of their biological source. This was primarily done via a linear interpolation method, to determine peak size, and a custom peak calling method to distinguish peaks from background noise.

Chapter 3 built on the processing methods developed in chapter 2 and focused on clustering ARISA experiments using a variety of methods. We chose to assess clustering performance by comparing the results to those obtained when clustering sequences from 454 sequencing of 16S rDNA sequencing. Using a cluster structure derived from a DNA sequencing experiment, we tested various ARISA clustering methods to see how well they matched. We discovered that many of the binning strategies discussed in the

literature yielded no appreciable benefit to clustering performance, while choices in clustering methods (such as Wards or Furthest neighbor clustering) did produce a benefit. We also observed that the data processing pipeline that we developed performed considerably better than when processing data via ABI's GeneMapper, albeit with default parameters. A future topic of focus would be to test out many of the GeneMapper options to see for which parameters each method prevails. For many researchers, exporting data from GeneMapper to a second program is their only option. If our peak calling and QC methods are superior, this will provide a tremendous benefit to the research committee and the software to carry out this action is already developed and available. Other commercial software solutions do exist (such as GelCompar II from the AppliedMaths) but they are expensive and not widely employed, so they do not represent benchmarks we must meet.

Chapter 4 focused on the development of Peak Studio, a software tool that incorporates the data processing and clustering strategies described in chapter's 2 and 3. Peak Studio includes a graphical user interface intended to make it simple for a user to select data, methods and parameters to accomplish ARISA analysis, in particular, the ability to view ARISA spectra and produce ARISA clusters. Peak Studio is publicly available and downloadable online for use or further development. A number of useful extensions have been suggested for improvements to Peak Studio. User feedback at this point will be the largest determining factor for what features Peak Studio should incorporate next. One suggestion put already put forth is to allow a user a way to tag individual peaks in the electropherogram with information such as taxa, fragment length or a species ID. A second suggestion is to associate the peaks in the ARISA with an

intergenic DNA fragment database so that each peak would potentially identify a list of possible species that match a peak of that particular size.

Peak Studio will be expanded to provide support for T-RFLP data. The code has a number of implementations already in place so that .fsa files from T-RFLP data can loaded and viewed. T-RFLP clustering has also been implemented, and follows the same general principles as ARISA analyses. In fact, any type of spectrum-based data (e.g., HPLC, LC, MS, GC, and capillary electrophoresis) could be viewed using the visualization component of Peak Studio once the appropriate parsers are written. Peak Studio is based on a flexible model and thus can be readily adapted to different data sets; it is hoped that it will provide great benefit to the research community.

To conclude, the findings of this dissertation provide the scientific community with improved analytical strategies in microarray and ARISA research, as well as provide open source software packages to aid in these types of analysis.

# REFERENCES

1. Hein AM, Richardson S (2006) A powerful method for detecting differentially expressed genes from GeneChip arrays that does not require replicates. BMC Bioinformatics 7: 353.

2. Baldi P, Long AD (2001) A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. Bioinformatics 17: 509-519.

3. Fodor AA, Tickle TL, Richardson C (2007) Towards the uniform distribution of null P values on Affymetrix microarrays. Genome Biol 8: R69.

4. von Wintzingerode F, Gobel UB, Stackebrandt E (1997) Determination of microbial diversity in environmental samples: pitfalls of PCR-based rRNA analysis. FEMS Microbiol Rev 21: 213-229.

5. Wagner M, Amann R, Lemmer H, Schleifer KH (1993) Probing activated sludge with oligonucleotides specific for proteobacteria: inadequacy of culture-dependent methods for describing microbial community structure. Appl Environ Microbiol 59: 1520-1525.

6. Staley JT, Konopka A (1985) Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats. Annu Rev Microbiol 39: 321-346.

7. Kirk JL, Beaudette LA, Hart M, Moutoglis P, Klironomos JN, et al. (2004) Methods of studying soil microbial diversity. J Microbiol Methods 58: 169-188.

8. Fisher MM, Triplett EW (1999) Automated approach for ribosomal intergenic spacer analysis of microbial diversity and its application to freshwater bacterial communities. Appl Environ Microbiol 65: 4630-4636.

9. Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270: 467-470.

10. Naef F, Magnasco MO (2003) Solving the riddle of the bright mismatches: labeling and effective binding in oligonucleotide arrays. Phys Rev E Stat Nonlin Soft Matter Phys 68: 011906.

11. Irizarry RA HB, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4: 249-264.

12. Li C HW (2001) Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. Genome Biol 2.

13. Ratushna VG, Weller JW, Gibas CJ (2005) Secondary structure in the target as a confounding factor in synthetic oligomer microarray design. BMC Genomics 6: 31.

14. Park PJ, Cao YA, Lee SY, Kim JW, Chang MS, et al. (2004) Current issues for DNA microarrays: platform comparison, double linear amplification, and universal RNA reference. J Biotechnol 112: 225-245.

15. Hein AMK SR (2006) A powerful method for detecting differentially expressed genes from GeneChip arrays that does not require replicates. BMC Bioinformatics 7: 353.

16. Ryan MM, Lockstone HE, Huffaker SJ, Wayland MT, Webster MJ, et al. (2006) Gene expression analysis of bipolar disorder reveals downregulation of the ubiquitin cycle and alterations in synaptic genes. Mol Psychiatry 11: 965-978.

17. Bolstad BM, Irizarry R. A., Astrand, M., and Speed (2003) A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance. Bioinformatics 19: 185-193.

18. Wu Z, Irizarry RA (2004) Preprocessing of oligonucleotide array data. Nat Biotech 22: 656-658.

19. Lemon WJ, Liyanarachchi S, You M (2003) A high performance test of differential gene expression for oligonucleotide arrays. Genome Biol 4: R67.

20. Hess A, Iyer H (2007) Fisher's combined p-value for detecting differentially expressed genes using Affymetrix expression arrays. BMC Genomics 8: 96.

21. Milo M, Fazeli A, Niranjan M, Lawrence ND (2003) A probabilistic model for the extraction of expression levels from oligonucleotide arrays. Biochem Soc Trans 31: 1510-1512.

22. Liu X, Milo M, Lawrence ND, Rattray M (2005) A tractable probabilistic model for Affymetrix probe-level analysis across multiple chips. Bioinformatics 21: 3637-3644.

23. Baldi P AL (2001) A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. Bioinformatics 17: 509-519.

24. Fodor AA TT, C Richardson (2007) Towards the uniform distribution of null P values on Affymetrix microarrays. Genome Biology 8: R69.

25. Klebanov L, Yakovlev A (2007) How high is the level of technical noise in microarray data? Biol Direct 2: 9.

26. De Mees C JL, J Bakker, J Smitz, B Hennuy, P Van Vooren, P Gabant, J Szpirer, C Szpirer (2006) Alpha-Fetoprotein Controls Female Fertility and Prenatal Development of the Gonadotropin-Releasing Hormone Pathway through an Antiestrogenic Action. Mol Cell Biol 26: 2012–2018.

27. Sommer P PLR, H Gillingham, A Berry, M Kayahara, T Huynh, A White, D W Ray (2007) Glucocorticoid receptor overexpression exerts an antisurvival effect on human small cell lung cancer cells. Oncogene 26: 7111–7121.

28. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Statist 6: 70.

29. Benjamini Y HY (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. J R Stat Soc B 57: 289–300.

30. Benjamini Y YD (2001) The control of the false discovery rate in multiple testing under dependency. Ann Stat 29: 1165-1168.

31. Li C WWI, edited by . , 2003, p. 1–455., editor (2003) DNA-chip analyzer (dChip). New York: Springer. 1–455 p.

32. Cui X GC (2003) Statistical tests for differential expression in cDNA microarray experiments. Genome Biol 4: 210.

33. Cheng C SP (2007) False discovery rate paradigms for statistical analyses of microarray gene expression data. Bioinformation 1: 436-446.

34. Pounds S (2006) Estimation and control of multiple testing error rates for microarray studies. Briefings in Bioinformatics 7: 25-36.

35. Pawitan Y KM, S Michiels, A Ploner (2005) Bias in the estimation of false discovery rate in microarray studies. Bioinformatics 21: 3865-3872.

36. Klebanov L, Qiu X, Welle S, Yakovlev A (2007) Statistical methods and microarray data. Nat Biotech 25: 25-26.

37. Provenzani A, Fronza R, Loreni F, Pascale A, Amadio M, et al. (2006) Global alterations in mRNA polysomal recruitment in a cell model of colorectal cancer progression to metastasis. Carcinogenesis 27: 1323-1333.

38. Anne Kirstine Müller KWSCSJS (2001) The effect of long-term mercury pollution on the soil microbial community. FEMS Microbiology Ecology 36: 11-19.

39. Clement BG, Kehl LE, DeBord KL, Kitts CL (1998) Terminal restriction fragment patterns (TRFPs), a rapid, PCR-based method for the comparison of complex bacterial communities. Journal of Microbiological Methods 31: 135-142.

40. Andreas Tom-Petersen TDLTLMON (2003) Effects of copper amendment on the bacterial community in agricultural soil analyzed by the T-RFLP technique. FEMS Microbiology Ecology 46: 53-62.

41. Shadi Sepehri RKCNBDOK (2007) Microbial diversity of inflamed and noninflamed gut biopsy tissues in inflammatory bowel disease. Inflammatory Bowel Diseases 13: 675-683.

42. Huse SM, Dethlefsen L, Huber JA, Welch DM, Relman DA, et al. (2008) Exploring Microbial Diversity and Taxonomy Using SSU rRNA Hypervariable Tag Sequencing. PLoS Genetics 4: e1000255.

43. Handelsman J, Rondon MR, Brady SF, Clardy J, Goodman RM (1998) Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. Chem Biol 5: R245-249.

44. Borneman J, Skroch PW, O'Sullivan KM, Palus JA, Rumjanek NG, et al. (1996) Molecular microbial diversity of an agricultural soil in Wisconsin. Appl Environ Microbiol 62: 1935-1943.

45. Pace NR (1997) A molecular view of microbial diversity and the biosphere. Science 276: 734-740.

46. Ovreas L, Torsvik VV (1998) Microbial Diversity and Community Structure in Two Different Agricultural Soil Communities. Microb Ecol 36: 303-315.

47. Eisen JA (2007) Environmental Shotgun Sequencing: Its Potential and Challenges for Studying the Hidden World of Microbes. PLoS Biology 5: e82.

48. Nocker A, Burr M, Camper AK (2007) Genotypic microbial community profiling: a critical technical review. Microb Ecol 54: 276-289.

49. Kunin V, Copeland A, Lapidus A, Mavromatis K, Hugenholtz P (2008) A bioinformatician's guide to metagenomics. Microbiol Mol Biol Rev 72: 557-578, Table of Contents.

50. Sogin ML, Morrison HG, Huber JA, Welch DM, Huse SM, et al. (2006) Microbial diversity in the deep sea and the underexplored rare biosphere. Proceedings of the National Academy of Sciences 103: 12115-12120.

51. Engelbrektson A, Kunin V, Wrighton KC, Zvenigorodsky N, Chen F, et al. (2010) Experimental factors affecting PCR-based estimates of microbial species richness and evenness. ISME J.

52. Kunin V, Engelbrektson A, Ochman H, Hugenholtz P (2010) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. Environ Microbiol 12: 118-123.

53. Tiedje JM, Asuming-Brempong S, Nüsslein K, Marsh TL, Flynn SJ (1999) Opening the black box of soil microbial diversity. Applied Soil Ecology 13: 109-122.

54. Liu WT, Marsh TL, Cheng H, Forney LJ (1997) Characterization of microbial diversity by determining terminal restriction fragment length polymorphisms of genes encoding 16S rRNA. Appl Environ Microbiol 63: 4516-4522.

55. Moyer CL, Dobbs FC, Karl DM (1994) Estimation of diversity and community structure through restriction fragment length polymorphism distribution analysis of bacterial 16S rRNA genes from a microbial mat at an active, hydrothermal vent system, Loihi Seamount, Hawaii. Appl Environ Microbiol 60: 871-879.

56. Ferris MJ, Muyzer G, Ward DM (1996) Denaturing gradient gel electrophoresis profiles of 16S rRNA-defined populations inhabiting a hot spring microbial mat community. Appl Environ Microbiol 62: 340-346.

57. Jones CM, Thies JE (2007) Soil microbial community analysis using two-dimensional polyacrylamide gel electrophoresis of the bacterial ribosomal internal transcribed spacer regions. J Microbiol Methods 69: 256-267.

58. Borneman J, Triplett EW (1997) Molecular microbial diversity in soils from eastern Amazonia: evidence for unusual microorganisms and microbial population shifts associated with deforestation. Appl Environ Microbiol 63: 2647-2653.

59. Garcia-Martinez J, Acinas SG, Anton AI, Rodriguez-Valera F (1999) Use of the 16S--23S ribosomal genes spacer region in studies of prokaryotic diversity. J Microbiol Methods 36: 55-64.

60. Brown MV, Schwalbach MS, Hewson I, Fuhrman JA (2005) Coupling 16S-ITS rDNA clone libraries and automated ribosomal intergenic spacer analysis to show marine microbial diversity: development and application to a time series. Environ Microbiol 7: 1466-1479.

61. Maggi RG, Breitschwerdt EB (2005) Potential limitations of the 16S-23S rRNA intergenic region for molecular detection of Bartonella species. J Clin Microbiol 43: 1171-1176.

62. Jones SE, Shade AL, McMahon KD, Kent AD (2007) Comparison of primer sets for use in automated ribosomal intergenic spacer analysis of aquatic bacterial communities: an ecological perspective. Appl Environ Microbiol 73: 659-662.

63. Popa R, Mashall MJ, Nguyen H, Tebo BM, Brauer S (2009) Limitations and benefits of ARISA intra-genomic diversity fingerprinting. J Microbiol Methods 78: 111-118.

64. Li W, Dowd SE, Scurlock B, Acosta-Martinez V, Lyte M (2009) Memory and learning behavior in mice is temporally associated with diet-induced alterations in gut bacteria. Physiol Behav 96: 557-567.

65. Soo RM, Wood SA, Grzymski JJ, McDonald IR, Cary SC (2009) Microbial biodiversity of thermophilic communities in hot mineral soils of Tramway Ridge, Mount Erebus, Antarctica. Environ Microbiol 11: 715-728.

66. Ramette A (2009) Quantitative community fingerprinting methods for estimating the abundance of operational taxonomic units in natural microbial communities. Appl Environ Microbiol 75: 2495-2505.

67. Hewson I, Fuhrman JA (2006) Improved strategy for comparing microbial assemblage fingerprints. Microb Ecol 51: 147-153.

68. Denman SE, Nicholson MJ, Brookman JL, Theodorou MK, McSweeney CS (2008) Detection and monitoring of anaerobic rumen fungi using an ARISA method. Lett Appl Microbiol 47: 492-499.

69. Wood SA, Jentzsch K, Rueckert A, Hamilton DP, Cary SC (2009) Hindcasting cyanobacterial communities in Lake Okaro with germination experiments and genetic analyses. FEMS Microbiol Ecol 67: 252-260.

70. Wood SA, Mountfort D, Selwood AI, Holland PT, Puddick J, et al. (2008) Widespread distribution and identification of eight novel microcystins in antarctic cyanobacterial mats. Appl Environ Microbiol 74: 7243-7251.

71. Lear G, Anderson MJ, Smith JP, Boxen K, Lewis GD (2008) Spatial and temporal heterogeneity of the bacterial communities in stream epilithic biofilms. FEMS Microbiol Ecol 65: 463-473.

72. Fierer N, Hamady M, Lauber CL, Knight R (2008) The influence of sex, handedness, and washing on the diversity of hand surface bacteria. Proc Natl Acad Sci U S A 105: 17994-17999.

73. Ward JH, Jr. (1963) Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58: 236-244.

74. Ishii S, Kadota K, Senoo K (2009) Application of a clustering-based peak alignment algorithm to analyze various DNA fingerprinting data. Journal of Microbiological Methods 78: 344-350.

75. Schulte im Walde S (2003) Experiments on the Automatic Induction of German Semantic Verb Classes. . AIMS Report , Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

76. Jackard P (1912) The distribution of flora in the alpine zone. New Phytol 11: 37-50.

77. Lozupone C, Hamady M, Knight R (2006) UniFrac--an online tool for comparing microbial community diversity in a phylogenetic context. BMC Bioinformatics 7: 371.

78. Lozupone C, Knight R (2005) UniFrac: a new phylogenetic method for comparing microbial communities. Appl Environ Microbiol 71: 8228-8235.

79. Felsenstein (1989) PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics 5: 164-166.

80. Kupczok A, Haeseler AV, Klaere S (2008) An exact algorithm for the geodesic distance between phylogenetic trees. J Comput Biol 15: 577-591.

81. Zmasek CM, Eddy SR (2001) ATV: display and manipulation of annotated phylogenetic trees. Bioinformatics 17: 383-384.

82. Ruan Q, Steele JA, Schwalbach MS, Fuhrman JA, Sun F (2006) A dynamic programming algorithm for binning microbial community profiles. Bioinformatics 22: 1508-1514.

83. Hewson I, Fuhrman JA (2004) Richness and diversity of bacterioplankton species along an estuarine gradient in Moreton Bay, Australia. Appl Environ Microbiol 70: 3425-3433.

84. Abdo Z, Schuette UM, Bent SJ, Williams CJ, Forney LJ, et al. (2006) Statistical methods for characterizing diversity of microbial communities by analysis of terminal restriction fragment length polymorphisms of 16S rRNA genes. Environ Microbiol 8: 929-938.

85. Mangiameli P, Chen SK, West D (1996) A comparison of SOM neural network and hierarchical clustering methods. European Journal of Operational Research 93: 402-417.

86. Milligan G (1980) An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrika 45: 325-342.

87. Han MV, Zmasek CM (2009) phyloXML: XML for evolutionary biology and comparative genomics. BMC Bioinformatics 10: 356.

# APPENDIX

## Aims Summary

Sub aims of Aim #1
Implement existing methods capable of analyzing single microarray methods in Java – (COMPLETED)
Student's T-test (paired and unpaired)
Wilcoxon non parametric test (paired and unpaired)
BGX algorithm (implemented in R)
CyberT algorithm (paired and unpaired
Implement PINC and compare to previous methods – (COMPLETED)
Test methods on Affymetrix Latin Square microarrays to determine sensitivity and specificity – (COMPLETED)
Test PINC on technical replicates – (COMPLETED)
Apply PINC to larger experiments of different biological sources to assess PINC's ability to determine variability – (COMPLETED)
Submit paper for publication – (COMPLETED)

Sub aims of Aim #2
Write parser to convert binary data from sequencer into base pair space based on size standards (majority of this work completed by Melanie Spencer) –(COMPLETED)
Implement correlation method to compare pairs of technical replicates –(COMPLETED)
Implement ½ Size standard Assessment method to assess base pair assignment – (COMPLETED)
Implement a size standard peak height detection method–(COMPLETED)

Sub aims of Aim #3
Implement existing binning methods in the literature (methods depicted in Figure 2-3) -(COMPLETED)
Develop random binning technique -(COMPLETED)
Develop CABS, the post binning correction step -(COMPLETED and abandoned)
Decide on a scoring metric to determine how well each binning method performs -(COMPLETED)
Compare binning methods using small and large datasets to determine Binning performance -(COMPLETED)
Submit paper for publication -(COMPLETED)

Sub aims of Aim #4
Write a Design document to govern the scope of the project -(COMPLETED)

Develop a visual component of the software tool allowing the display of single or multiple ARISA experiments -(COMPLETED)
Implement QC steps from Aim #2 into software tool -(COMPLETED)
Implement Binning methods from Aim #3 -(COMPLETED)
Design and generate code to export results in user friendly formats -(COMPLETED)

# VITA

Robert Reid
robr@msu.edu

800 Honeysuckle Lane, Midland, NC, 28107
704-888-8266 (H), 704-668-6964(C)

## Education

University of North Carolina Charlotte, Charlotte, NC, USA
Candidate for PhD in Bioinformatics, Summer 2010 (GPA = 3.93)

- Graduate level courses completed in Computational Comparative Genomics, Analysis of Microarray Data, Computational Structural Biology, Molecular Sequence Analysis.

Michigan State University, East Lansing, Michigan, USA
Master of Science in Physiology, August 2001

- Graduate level courses completed in genome physiology, cell signaling biochemistry, molecular biology, protein structure, pharmacology of excitable cells.

University of Waterloo, Waterloo, Ontario, Canada
Honors Bachelor of Science, May 1997

- Labs and coursework taken in anatomy, statistics, biomechanics, computer science, public speaking, organic chemistry, physics, calculus and physiology.

## Publications

Reid RW, Fodor AA. Determining gene expression on a single pair of microarrays. BMC Bioinformatics. 2008, 9:489. (* Highly accessed)

Fujita K, Forsyth M, MacFarlane D, Reid RW, Elliott GD. Unexpected improvement in stability and utility of cytochrome c by solution in biocompatible ionic liquids. Biotechnol Bioeng, 2006.

Luisi DL, RW Reid. Excipient Analysis of an IgG 1 by Differential Scanning Calorimetry. Current Trends in Microcalorimetry Conference Poster Abstracts. 2003.

Jayaraman RC, Reid RW, Foley JM, Prior BM, Dudley GA, Weingand KW, Meyer RA. MRI evaluation of topical heat and static stretching as therapeutic modalities for the

treatment of eccentric exercise-induced muscle damage. Eur J Appl Physiol. 2004 Oct;93(1-2):30-8.

Meyer RA, Towse TF, Reid RW, Jayaraman RC, Wiseman RW, McCully KK. BOLD MRI mapping of transient hyperemia in skeletal muscle after single contractions. NMR Biomed. 2004 Oct;17(6):392-8.

RW Reid, JM Foley, RC Jayaraman, BM Prior, RA Meyer. Effect of aerobic capacity on the T2 increase in exercised skeletal muscle. J Appl Physiol, 90:897-902, 2001.

Prior BM, Jayaraman RC, Reid RW, Cooper TG, Foley JM, Dudley GA, Meyer RA. Biarticular and monoarticular muscle activation and injury in human quadriceps muscle. Eur J Appl Physiol, 85:185-190, 2001.

RW Reid, JM Foley, BM Prior, KW Weingand, RA Meyer. Mild topical heat increases popliteal blood flow as measured by MRI. Med Sci Spo Exec, 31:5, abstract suppl, 1999.

JM Foley, RW Reid, DW Vaughn, MT Andary, RA Meyer. Assessment of motor unit pathology by functional MRI. Med Sci Spo Exec, 31:5, abstract suppl, 1999.