2017

# Statistical methods for analyzing sequencing data with applications in modern biomedical analysis and personalized medicine

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

# STATISTICAL METHODS FOR ANALYZING SEQUENCING DATA WITH APPLICATIONS IN MODERN BIOMEDICAL ANALYSIS AND PERSONALIZED MEDICINE

by

**SOLAIAPPAN MANIMARAN**

B.Sc., Indian Institute of Technology, Kharagpur, 1995
M.E., Indian Institute of Science, Bangalore, 1999
M.B.A., University of Connecticut, 2012
M.A., Boston University, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2017

Approved by

First Reader

W. Evan Johnson, Ph.D.
Associate Professor of Medicine and Biostatistics

Second Reader

Paola Sebastiani, Ph.D.
Professor of Biostatistics

Third Reader

Josée Dupuis, Ph.D.
Professor of Biostatistics

## DEDICATION

I would like to dedicate this work to my mother Kanthimathi and father Solaiappan, and my wife Nivetha and my wonderful daughter Shwetha.

## ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my advisor Dr. Evan Johnson for his guidance throughout my PhD program. He has advised me on all aspects of my career at Boston University and beyond. I cannot thank him enough for all that I could achieve in my career. I would like to thank my thesis committee members Dr. Paola Sebastiani, Dr. Josée Dupuis, Dr. Ching-Ti Liu and Dr. Stefano Monti for critical review of my thesis related documents and presentations and for their constant support and encouragement to complete my PhD program successfully. I would like to thank Johnson's lab current members Supriya, Tyler, David and Yuqing, and former members Changjin, Ying and Bing for their support and encouragement. I would like to thank all of our collaborators Keith Crandall, Eduardo Castro Nallar, Allyson Byrd, Joe Perez-Rogers, Jeffrey Leek, Claire Ruberman, Hector Corrada Bravo, Kwame Okrah, Alice H. Lichtenstein, Nirupa R. Matthan, Anne Kane, Owen Francis and Matthew Bendall. I would like to thank all Computational Biomedicine friends for all the CBM meeting presentations and feedback. I would like to thank all my friends from the Biostatistics department for group studies during the coursework and would like to especially thank Avery, Revathi, Pranab and Dan for group studies during qualifying exams. I would like to thank all of the administrative staff and especially Marisa for support. I would like to thank my brothers Thirumani, Manikandan and Manikumar, my mother-in-law Jeevarekha and father-in-law Murugesh for their love and support. Finally, last but not the least, I would like to thank and dedicate this dissertation to my mother Kanthimathi, father Solaiappan, wife Nivetha and daughter Shwetha for their kind love and support for ever.

**STATISTICAL METHODS FOR ANALYZING SEQUENCING DATA WITH APPLICATIONS IN MODERN BIOMEDICAL ANALYSIS AND PERSONALIZED MEDICINE**

**SOLAIAPPAN MANIMARAN**

Boston University Graduate School of Arts and Sciences, 2017

Major Professor: W. Evan Johnson, Associate Professor of Medicine and Biostatistics

ABSTRACT

There has been tremendous advancement in sequencing technologies; the rate at which sequencing data can be generated has increased multifold while the cost of sequencing continues on a downward descent. Sequencing data provide novel insights into the ecological environment of microbes as well as human health and disease status but challenge investigators with a variety of computational issues. This thesis focuses on three common problems in the analysis of high-throughput data. The goals of the first project are to (1) develop a statistical framework and a complete software pipeline for metagenomics that identifies microbes to the strain level and thus facilitating a personalized drug treatment targeting the strain; and (2) estimate the relative content of microbes in a sample as accurately and as quickly as possible.

The second project focuses on the analysis of the microbiome variation across multiple samples. Studying the variation of microbiomes under different conditions within an organism or environment is the key to diagnosing diseases and providing

personalized treatments. The goals are to (1) identify various statistical diversity measures; (2) develop confidence regions for the relative abundance estimates; (3) perform multi-dimensional and differential expression analysis; and (4) develop a complete pipeline for multi-sample microbiome analysis.

The third project is focused on batch effect analysis. When analyzing high dimensional data, non-biological experimental variation or "batch effects" confound the true associations between the conditions of interest and the outcome variable. Batch effects exist even after normalization. Hence, unless the batch effects are identified and corrected, any attempts for downstream analyses, will likely be error prone and may lead to false positive results. The goals are to (1) analyze the effect of correlation of the batch adjusted data and develop new techniques to account for correlation in two step hypothesis testing approach; (2) develop a software pipeline to identify whether batch effects are present in the data and adjust for batch effects in a suitable way.

In summary, we developed software pipelines called PathoScope, PathoStat and BatchQC as part of these projects and validated our techniques using simulation and real data sets.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

BU ..... Boston University

PathoLib ..... PathoScope Library preparation module

PathoMap ..... PathoScope alignment Mapping module

PathoID ..... PathoScope Identification module

PathoRep .....PathoScope Report module

PathoStat ..... PathoScope Statistics module

BatchQC..... Batch effects Quality Control module

# CHAPTER ONE

## Introduction

There has been tremendous recent advancement in sequencing technologies; the rate at which sequencing data can be generated has increased multifold while the cost of this sequencing continues on a downward descent. Sequencing data provide novel insights into the ecological environment of microbes as well as human health and disease status.

This thesis focuses on three common problems in the analysis of high-throughput data, which are addressed by the three projects here. The first project is on methods and software for complete single-sample metagenomic analysis and we developed a software pipeline called PathoScope as part of this project. The second project develops a toolkit for microbiome variation analysis and we developed a R shiny app package called PathoStat as part of this project. The third project focuses on batch effects analysis and we developed a R shiny app package called BatchQC as part of this project.

Any sample that is collected from an environment of microbes will contain a multitude of microbes at different proportions. Metagenomics is the study of genomic data, usually sequencing data, from these samples, typically with the goal of characterizing the microbial communities present in the sample. Although many tools have emerged for analyzing metagenomic-sequencing data, these existing methods are often not efficient and or may have low specificity. These tools usually do not include a complete metagenomic data analysis framework. The goals of the first project are (1) to develop a statistical framework and a complete software pipeline for metagenomic

analysis of next-generation sequencing data that identifies microbes and pathogens to the strain level; (2) to estimate the relative content of pathogens and benign microbial flora in a sample as accurately as possible, while developing tools that can analyze massive amounts of data in a short timeframe. Broadly, the project is structured into the following aims: A) Determine how accurately microbial proportions can be estimated when there is a mixture of multiple microbes with varying proportions in the sample. A Bayesian mixture modeling approach generally works best in this context because it can integrate information across all genomes and converges to a solution quickly using an expectation maximization algorithm. The aim is to evaluate the accuracy of the Bayesian mixture modeling approach in comparison to other methods and also to evaluate how prior information in the Bayesian mixture modeling affects (or improves) these estimates. B) Evaluate how many sequencing reads are needed (read coverage) to estimate pathogen proportions to a given confidence limit, and determine how long it takes to estimate microbial proportions within a given confidence limits using our mixture model and in comparison to other methods. C) Develop a complete software pipeline, PathoScope, for metagenomics analysis of next-generation sequencing data.

The microbiome can vary significantly between different biological conditions such as disease status or treatment differences or other covariates of interest within an organism or environment. Studying the variation of microbiomes under different conditions within an organism or environment is the key to diagnosing diseases and providing personalized treatments. For the second project, the goals are the following: A) Identify various statistical measures such as alpha and beta diversity for characterizing

the microbiome variation under different conditions and develop a module for visualization of the statistical measures; B) Develop a module to calculate and display the confidence regions for the relative abundance estimates; C) Perform Multi-dimensional and differential expression analysis of microbiomes under various conditions of interest; D) Develop a software pipeline called PathoStat for Microbiome variation analysis.

When analyzing high dimensional data, non-biological experimental variation or "batch effects" confound the true associations between the conditions of interest and the outcome variable. As shown in the Figure 1 of the nature review article (Leek, Scharpf, Bravo et al., 2010), batch effects exist even after normalization. Hence, unless the batch effects are identified and removed, any attempts for downstream analyses, such as network inference and estimation, will likely be error prone and may lead to false positive results. Furthermore, many batch adjustment approaches artificially induce a correlation structure in the batch adjusted data that can often exaggerate the significance of results (e.g. p-values) or even introduce spurious relationships in the data. This motivates the need for a computational framework to systematically identifying batch effects. Here, the aim is to develop a tool called BatchQC that visually depicts aspects of high dimensional data and evaluates the extent to which batch effects impact the association between the conditions of interest. Broadly, the aims of the third project are to do the following: A) Analyze the effect of correlation of the batch adjusted data and develop new techniques to account for correlation in two step hypothesis testing approach. B) Develop a software pipeline called BatchQC to identify whether batch effects are present in the data and adjust for batch effects in a suitable way.

The three projects together provide a complete set of toolkits and methodology necessary for analysis of genomic data for these applications. In particular, the PathoScope toolkit is useful during a new outbreak situation and helps in identifying the particular strain of the pathogen causing the outbreak and thereby facilitating a personalized treatment plan targeting the particular strain of the pathogen. The PathoStat toolkit for microbiome variation analysis is useful in studying the microbiome variations associated with various diseases such as skin disease (Speeckaert, Lambert, Grine et al., 2016), celiac disease (Harnett, Myers & Rolfe, 2016; Leonard & Fasano, 2016; Scher, 2016) and gastrointestinal cancer (Wroblewski, Peek & Coburn, 2016). When performing multi-dimensional and differential expression analysis using genomic data, often times they are affected by technical variation attributable to both observed and unobserved factors (Leek *et al.*, 2010) and BatchQC can help to identify whether batch effects are present in the data and adjust for batch effects in a suitable way. We hope that these toolkits and methodology developed as part of these projects is of tremendous help to the scientific community in performing genomic data analysis.

# CHAPTER TWO

## Project 1: Methods and Software for Complete Metagenomic Analysis

### Introduction

Metagenomics is the study of genomic data, usually sequencing data, typically with the goal of characterizing the microbial communities present in the sample. Although many tools have emerged for analyzing metagenomic-sequencing data, these existing methods are often not efficient and or may have low specificity. These tools usually do not include a complete metagenomic data analysis framework. The goals of this project are (1) to develop a statistical framework and a complete software pipeline for metagenomic analysis of next-generation sequencing data that identifies microbes and pathogens to the strain level; (2) is to estimate the relative content of pathogens and benign microbial flora in a sample as accurately as possible, while developing tools that can analyze massive amounts of data in a short timeframe. Broadly, the project is structured into the following aims: A) Determine how accurately microbial proportions can be estimated when there is a mixture of multiple microbes with varying proportions in the sample. A Bayesian mixture modeling approach generally works best in this context because it can integrate information across all genomes and converges to a solution quickly using an expectation maximization algorithm. The aim is to evaluate the accuracy of the Bayesian mixture modeling approach in comparison to other methods and also to evaluate how prior information in the Bayesian mixture modeling affects (or improves) these estimates. B) Evaluate how many sequencing reads are needed (read

coverage) to estimate pathogen proportions to a given confidence limit, and determine how long it takes to estimate microbial proportions within a given confidence limits using our mixture model and in comparison to other methods. C) Develop a complete software pipeline, PathoScope, for metagenomics analysis of next-generation sequencing data.

## Metagenomic Analysis

The primary initial analysis goal for each sample from a metagenomics study is to identify the microbes present and also to quantify the exact proportion of each of the microbes in a sample.

Before the introduction of our PathoScope approach, one or more of the following three general approaches were most common for metagenomic analysis: 1) composition or pattern matching (Wood & Salzberg, 2014; Segata, Waldron, Ballarini et al., 2012; Brady & Salzberg, 2009; McHardy, Martin, Tsirigos et al., 2007), 2) taxonomic mapping (Patil, Roune & McHardy, 2012; Segata *et al.*, 2012; Gerlach & Stoye, 2011; Monzoorul Haque, Ghosh, Komanduri et al., 2009; Meyer, Paarmann, D'Souza et al., 2008; Huson, Auch, Qi et al., 2007), and 3) whole genome assembly (Bhaduri, Qu, Lee et al., 2012; Kostic, Ojesina, Pedamallu et al., 2011).

In composition and pattern matching algorithms, pre-determined patterns in the data, such as taxonomic clade markers (Segata *et al.*, 2012), k-mer frequency and GC content along with classification algorithms such as support vector machines (Patil *et al.*, 2012; McHardy *et al.*, 2007) or interpolated Markov Models (Brady & Salzberg, 2009)

are used to classify reads to the species of interest. Often times, these approaches require a huge amount of preprocessing to be done on the genomic database before they can be applied and the results of these methods can also vary based on the size and composition of the genome database.

In taxonomy based approaches, the most specific taxonomic group for each read is identified and they are also called the "lowest common ancestor" approach (Huson *et al.*, 2007). In this approach, reads are assigned to the lowest taxonomic group that contains all the genomes with which the reads share homologous regions. While these methods are very accurate for higher-level taxonomic levels (e.g. phylum and family), they are not very accurate for lower levels (e.g. species and strain) (Gerlach & Stoye, 2011). These approaches fail to identify the specific species or strains in the sample, particularly when the reads originate from a strain that is closely related to another one in the database, but rather maps the reads to higher-level taxonomies, which is not very informative.

Assembly-based algorithms can accurately identify to the strain level, but these methods are time consuming requiring the construction of a whole genome. They are also computationally difficult and require large numbers of reads in the order of 50-100X coverage of the target genome to achieve an adequate accuracy (Schatz, Delcher & Salzberg, 2010). For purified samples, the current sequencing depths level of coverage is usually sufficient, but for mixed samples or in multiplex sequencing runs, the level of

coverage is usually not sufficient. These assembly approaches also require multiple filtering and alignment steps to obtain reads specific to the pathogen of interest, when data collection at a crime scene or hospital include additional environmental components such as host genome or naturally occurring bacterial and viral species.

Here, we describe our PathoScope computational framework for strain identification in environmental or clinical sequencing samples to analyze next-generation sequence data. PathoScope consists of four core modules namely PathoLib, PathoMap, PathoID and PathoReport, and two optional modules namely PathoDB (database to store detailed information about the reference sequences) and PathoQC (quality control of input read sequences). PathoLib is a module for generating custom reference genome libraries. PathoMap is a module for aligning reads to the target reference library and filter out any reads that aligns better to filter reference library. PathoID is a module that utilizes a modified pseudo likelihood model based on a Bayesian modeling framework (Francis, Bendall, Manimaran et al., 2013) for reassigning all ambiguous reads to the most likely source genome in the reference library. PathoReport is a module for generating a report in TSV (Tab Separated Value) format from the alignment file generated by PathoMap after reassignment processing by PathoID. Our PathoID reassignment approach accommodates information on mapping quality, read length and provides posterior probabilities of matches to a known database of reference genomes. Importantly, our PathoID approach incorporates the possibility that multiple species can be present in the sample or that the target strain is not even contained within the reference database. It also

accurately discriminates between very closely related strains of the same species with much less than 1X coverage of the genome and without the need for sequence assembly or complex preprocessing of the database or taxonomy. No other method in the literature can identify species or substrains in such a direct and automatic manner and without the need for large numbers of reads. We later describe in detail two versions of PathoID (versions 1.0 and 2.0) which have the same base statistical model, but differ in the way the data they input and handle aligned sequencing data, and PathoID 2.0 has more flexibility in prior parameter selection. Specifically, in PathoID version 1.0, we used SAM -MAPQ (https://samtools.github.io/hts-specs/SAMv1.pdf) score to provide relative alignment probabilities. MAPQ score is used to capture the goodness of mapping, which equals −10 log10 Pr(mapping position is wrong), rounded to the nearest integer. It is essentially a metric that measures the likelihood that a read comes from the reported position. In PathoID 2.0, we use the read alignment bit score (AS) from the Bowtie2 (Langmead & Salzberg, 2012) output. An alignment score quantifies how similar the read sequence is to the reference sequence aligned to. The higher the score, the more similar they are. A score is calculated by subtracting penalties for each difference (mismatch, gap, etc) and, in local alignment mode, adding bonuses for each match. The scores can be configured with the --ma (match bonus), --mp (mismatch penalty), --np (penalty for having an N in either the read or the reference), --rdg (affine read gap penalty) and --rfg (affine reference gap penalty) options.(http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#scores-higher-more-similar). In PathoID 2.0 the AS is also standardized by adding the read length and normalizing the score (Hong,

Manimaran, Shen et al., 2014b). PathoID 2.0 also has an option to accept a more varied selection of prior parameters for the Bayesian model. More details about the PathoScope modules are provided later in this chapter.

## Aim 1A

Accurately estimate microbial proportions, using a Bayesian mixture modeling approach, when there is a mixture of multiple microbes with varying proportions in the sample. We will also evaluate how prior information in the Bayesian mixture modeling affects (or improves) these estimates.

### *Objective*

Use in silico mixtures of microbial reads from biological isolates using varying proportions of microbes present in each of the samples. Study the accuracy of the metagenomic estimates for different cases using a Bayesian mixture modeling approach with different priors and parameterizations.

### *Rationale*

When pathogens are present in equal proportions in a sample or if the aim is merely to detect the presence of pathogens, there are several sequence aligners available, such as Bowtie2 (Langmead & Salzberg, 2012) or BLAST (Altschul, Madden, Schaffer et al., 1997), to accomplish this task. In a sample with many sequencing reads, there will be some reads (often a small proportion) that will uniquely align to the reference genomes of some of the pathogens present in the sample, and the aligners can simply

detect those pathogens. However, when the pathogens are present in varying proportions, it is very difficult to estimate the proportions of each of the pathogens accurately. Reassignment of reads aligned by these aligners to the correct pathogens using Bayesian mixture models have the potential to outperform other methods both in accuracy and speed, but no one has conducted a formal evaluation of these tools. Hence, a study to estimate the accuracy of the Bayesian mixture model approach when there is a mixture of multiple pathogens with varying proportions is very important to establish the efficacy of the method.

### *Experimental Setup*

Samples are simulated by mixing reads in silico from pathogens in different proportions. As an example, we selected reads from three different bacterial strains, with sequencing data from experimental or clinical isolates, that are available from the NCBI Sequence Read Archive (SRA; http://www.ncbi.nlm.nih.gov/sra). We collected reads from the *Y. pestis* KIM D27 (SRR032501), *E. coli* K-12 MG1655 (SRR031601), and *F. tularensis subsp. holarctica* OSU18 (SRR032505) data sets. All of these reads were sequenced using the 454 platform (Roche). These data sets consisted of 318332, 143836 and 28221 reads respectively. Read lengths typically range from 77 to 277 base pairs. We picked a maximum of 5770 reads from each of the data sets based on the lowest number of mapped reads available in the *F. tularensis* dataset, and mixed them in different proportions to form 1000 samples.

For another thorough validation of the method, we simulated reads from 25 strains of bacteria, at different proportions. These 25 strains including five closely related

*Escherichia coli* substrains, five closely related *Staphylococcus aureus* substrains, five closely related *Streptococcus pneumoniae* substrains and 10 other commonly occurring human bacterial strains that are more distantly to each other and the other strains in the sample. We estimated pathogen proportions using our PathoID Bayesian mixture model with different parameterizations and priors and used the true proportions to establish the efficacy of the method.

### *Analysis Plan*

The effect of different prior parameters, read lengths and quality of the samples on the pathogens proportions estimate of the Bayes mixture model for different samples with varying proportions of closely related and unrelated mixture of pathogens will be analyzed. Read lengths and a weighted alignment score will be incorporated in the likelihood calculation of the Bayes mixture model and choose the best alignment parameters to use for samples generated from different sequencing platforms.

### *Two Component Mixture Model*

In the context of a simple mixture model, let Y be a mixture of two distributions $Y1 \sim \emptyset_{\theta_1}(y)$ and $Y1 \sim \emptyset_{\theta_2}(y)$ with $X\ unknown\ and\ X \sim Bernoulli(\pi)$ as follows:

$$Y = (1 - X)Y_1 + XY_2$$

$$X \in [0,1]$$

Then the marginal distribution of Y based on the observed data, integrating out the missing data is the following:

$$g_Y(y) = (1-\pi)\emptyset_{\theta_1}(y) + \pi\emptyset_{\theta_2}(y)$$

$$\Pr(X = 1) = \pi$$

We will let $\theta = (\pi, \theta_1, \theta_2)$ represent the parameters of this model.

The log-likelihood based on N training cases of data Z is:

$$l(\theta; Z) = \sum_{i=1}^{N} log\left[(1-\pi)\emptyset_{\theta_1}(y_i) + \pi\emptyset_{\theta_2}(y_i)\right]$$

Direct maximization of $l(\theta; Z)$ is quite difficult. Instead, we use the unobserved latent variable $\Delta_i$ and represent the complete data (observed data plus missing data) likelihood equivalently but more conveniently as follows.

$$L(\theta; Z, X) = \prod_{i=1}^{N}\left[\left(\emptyset_{\theta_1}(y_i)\,\pi\right)^{(1-X_i)}\left(\emptyset_{\theta_2}(y_i)\,(1-\pi)\right)^{X_i}\right]$$

Then the complete data log-likelihood function would be given as follows.

$$l(\theta; Z, X) = \sum_{i=1}^{N}\left[(1-X_i)log\,\emptyset_{\theta_1}(y_i) + X_i log\emptyset_{\theta_2}(y_i)\right]$$

$$+ \sum_{i=1}^{N}[(1-X_i)log\,\pi + X_i\log(1-\pi)]$$

We use the Expectation Maximization (EM) algorithm (Dempster, 1977) to effectively impute the missing data and estimate the model parameters. Because the log-likelihood is linear in the missing data, the E-step reduces to computing the expected values of $X_i$ given the current values of the model parameters $\theta$ (using reasonable starting values for initiation), $\delta_i(\theta) = E(X_i|\theta, Z) = \Pr(X_i = 1|\theta, Z)$, and inserting these expectations into the log-likelihood. The M-step then consists of computing $\theta$ that

maximizes the complete data log likelihood using the imputed missing data. The E- and M- steps are then iterated to convergence. This EM approach has been shown to be equivalent to maximizing the parameters over the marginal likelihood of the observed data given above (Hastie, Tibshirani & Friedman, 2009).

### *Bayesian Mixture Model*

Now we extend the simple two component mixture model to Bayesian Mixture model. To formally describe the Bayesian Mixture model, let $i=1,...,R$ be the index of the reads and let $j=1,...,G$ be the index of the genomes in the database. Let $\mathbf{x_i}=(x_{i1},x_{i2},...,x_{iG})=[x_{ij}]$ be a set of genome indicators for read $i$ where $x_{ij}=1$ if the read originated from the $j$th genome and $x_{ij}=0$ if the read did not come from genome $j$. Note that by assumption one and only one element in the vector $\mathbf{x_i}$ can be equal to 1 (i.e. each read has only one template genome). It is assumed that $\mathbf{x_i}$ follows a multinomial distribution, with probability of success $\boldsymbol{\pi}=(\pi_1,\pi_2,...,\pi_G)=[\pi_j]$ where $\pi_j$ is the proportion of the reads that originated from the $j$th genome.

### *Likelihood model*

For the Likelihood model, as we have seen in the simple two component mixture model, it is difficult to perform the direct maximization of the likelihood based on the observed data. For those reads that are uniquely mapped, the likelihood is directly known, but for those reads that are not mapped uniquely, we use a penalized mixture model that penalizes the likelihood contributions from reads that map to multiple genomes, and thus relies more heavily on reads that align to only one genome. To facilitate the penalty on

the multi-map reads, let us define $y_i$ as the *uniqueness indicator* for read *i*, namely letting $y_i$=1 if read *i* uniquely maps to one genome and $y_i$=0 otherwise. We also define a second set of parameters, $\boldsymbol{\theta}=(\theta_1,\theta_2,...,\theta_G)=[\theta_j]$ to represent the multinomial distribution that $y_i$ is assumed to follow. For the *unique reads*, we directly observe the genome indicator $\mathbf{x_i}$. For the *non-unique* reads, the genome indicator $\mathbf{x_i}$ is unobserved or *missing data* and the observations are partial mapping qualities for each of the genomes. Alignment programs Bowtie2 (Langmead & Salzberg, 2012) and BLAST (Altschul *et al.*, 1997) reports a score for the goodness of the alignment of the read with each of the genomes. In general, many read alignment approaches use some variant of a Smith-Waterman algorithm (Durbin, Eddy, Krogh et al., 1998). These algorithms employ dynamic programing approaches to identify the optimal sequence alignment between two nucleic acid or protein sequences. To summarize succinctly, assume the following simple scoring function for an alignment of two sequences: +2 for each base/peptide matched in the alignment, +1 for each base/peptide mismatched, and 0 for each gap inserted into the alignment, and then are summed to generate an individual score, *r*, for each proposed alignment. Then for example, aligning the nucleic acid sequence 'AGTAGAC' to 'ATACGA' has an optimal alignment of:

```
AGTA-GAC
|  ||  |  |
A-TACG-A
```

Which has an alignment score of *r*=9 (four matches, one mismatch, three gaps). The Smith-Waterman dynamic programming algorithm can be used to identify the best possible local alignment (optimal *r*) in an efficient way (Durbin *et al.*, 1998). Note that in

these algorithms, the exponentiated alignment score, $q=e^r$, can have a likelihood interpretation. Namely, we assume that each sequence position in the alignment is *i.i.d.* and follows a multinomial distribution with three choices, match, mismatch, and gap, with the selected scoring function determining the multinomial probabilities. Now, extending this likelihood interpretation to the case of aligning reads to multiple genomes, the exponentiated alignment score of read $i$ to genome $j$, denoted $q_{ij}$, be interpreted as the relative likelihood that read $i$ originated from genome $j$. BLAST and Bowtie2 use heuristic simplifications of the Smith-Waterman to calculate read scores and attempt to find the best alignments. Although these algorithms are not guaranteed to find the optimal sequence alignment, but usually do find the best-scoring alignment and do so in a manner that is computationally efficient, often orders of magnitude faster than the much slower Smith-Waterman. For this reason, we will use these algorithms and assume that the alignments produced represent all optimal and suboptimal alignment possibilities, and the exponetiated BLAST, Bowtie2 or (unexponentiated) MAPQ scores represent relative likelihood alignment scores, $q_{ij}$, to be used in the mixture model below.

In PathoID 2.0, we use the relative likelihood mapping scores described later in the Modified Pseudo Likelihood model section and denoted by $\mathbf{q_i}=(q_{i1},q_{i2},...,q_{iG})=[q_{ij}]$. For unique reads, the $q_{ij}$ values are equal to the $x_{ij}$ values. For non-unique reads, the mapping scores represent the uncertainty in mapping and need to be rescaled—or equivalently these reads need to be reassigned to the correct template genome of origin. In order to do this, we use the parameters $\boldsymbol{\theta}=(\theta_1,\theta_2,...,\theta_G)=[\theta_j]$ for performing the reassignment and here $\theta_j$ parameter represents the proportion of the non-unique reads that

need to be reassigned to the *j*th genome. The complete data likelihood of the parameters (**π, θ**) given the observed data (reads mapping quality $q_{ij}$, $y_i$, unique $\mathbf{x_i}$) and the missing data (non-unique $\mathbf{x_i}$) is calculated as follows:

The unconditional distribution of $\mathbf{x_i}$ is given by the following multinomial distribution:

$$f(x_i|\pi) = \prod_{j=1}^{G} \pi_j^{x_{ij}}$$

Since $\mathbf{x_i}$ is not observed, we cannot directly compute the likelihood $L(\boldsymbol{\pi}|\mathbf{x_i})$ and we can instead compute the conditional likelihood given the observed data $y_i$ and $\mathbf{q_i}$, where $\mathbf{q_i}$ is the quality score.

$$f(x_i, y_i|\pi, \theta, q_i) = f(x_i|y_i, \pi, \theta, q_i) f(y_i|\pi, \theta, q_i)$$

$$f(x_i, y_i|\pi, \theta, q_i) = \left(\prod_{j=1}^{G} \pi_j^{x_{ij}}\right)\left(\prod_{j=1}^{G} \left(\theta_j^{(1-y_i)} \frac{q_{ij}}{\sum_{k=1}^{G} q_{ik}}\right)^{x_{ij}}\right)$$

We can rewrite the above equation as follows:

$$f(x_i, y_i|\pi, \theta, q_i) \propto \left(\prod_{j=1}^{G} \left(\pi_j \theta_j^{(1-y_i)} q_{ij}\right)^{x_{ij}}\right)$$

Since each reads are assumed to be independent reads, we get the following .

$$f(x, y|\pi, \theta, q) \propto \left(\prod_{i=1}^{R} \prod_{j=1}^{G} \left(\pi_j \theta_j^{(1-y_i)} q_{ij}\right)^{x_{ij}}\right)$$

And the likelihood is the following.

$$L(\boldsymbol{\pi}, \boldsymbol{\theta}|\mathbf{x_i}, \mathbf{q_i}, \mathbf{y}) \propto \prod_{i=1}^{R} \prod_{j=1}^{G} [\pi_j \theta_j^{(1-y_i)} q_{ij}]^{x_{ij}}$$

Although the reassigned reads (estimated $\mathbf{x_i}$) and reassignment parameters (estimated **θ**) are very informative, the quantities of interest from the modeling steps are the estimates for the genome read proportions (estimated **π**). These probabilities will identify the single

or multiple organisms from the database that are present in the samples, based on the proportion of the reads that are assigned to the genome after the reads are reassigned.

*Bayesian Prior Distribution*

We will assume that *a priori* the variables $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ follow Dirichlet distributions, the densities of which can be seen in the following equations:

$$f(\boldsymbol{\pi}|\mathbf{a}) \propto \prod_{j=1}^{G} \pi_j^{a_j-1} \text{ and } f(\boldsymbol{\theta}|\mathbf{b}) \propto \prod_{j=1}^{G} \theta_j^{b_j-1}.$$

If $a_j$=1 for all $j=1,\ldots,G$, this is equivalent to adding one unique read for each of the $G$ genomes, and $a_j=n$ would be the equivalent of adding $n$ unique reads to the $j$th genome. Similarly, $b_j=n$ is the equivalent of adding $n$ reads of non-unique read probabilities to the $j$th genome. In our PathoID 1.0 model, we use a non-informative prior distribution and the likelihood as described above.

*Modified Pseudo Likelihood model*

For PathoID 2.0, we develop a modified Pseudo Likelihood model that penalizes the likelihood based on read length and read alignment score and utilizes prior information about the genome proportions of different strains. In this model, the likelihood that read $i$ is from genome $j$ ($q_{ij}$) is constructed as a normalized score of the sum of read alignment score and read length as follows.

$$q_{ij} = \frac{s_{ij} + l_i}{\sum_{k=1}^{G}(s_{ik} + l_i)}$$

$q_{ij} := Normalized\ score\ that\ read\ i\ is\ from\ genome\ j$
$s_{ij} := Alignment\ score\ that\ read\ i\ is\ from\ genome\ j$
$l_i := Read\ length\ of\ read\ i$

This is a pseudo likelihood model because it's a likelihood model that is intuitively changed so it performs better, but no longer a likelihood. PathoID 2.0 uses this pseudo-likelihood model, whereas PathoID 1.0 uses the actual, proper likelihood for reassignment. In the software implementation of PathoID 2.0, the user can easily modify the prior values (most importantly for θ), whereas the software for PathoID 1.0 does not accommodate prior values besides the built-in non-informative values.

*EM algorithm*

Estimation of the model parameters and reassignment of the reads is accomplished using an Expectation-Maximization (EM) algorithm (Dempster, 1977). In the E-step, the expected value of $\mathbf{x_i}$ is computed for each combination of *i=1,...,R* and *j=1,...,G* based estimates of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, as well as the observed data $\mathbf{q_i}$ and $\mathbf{y}$. In the E-step, the expected values of the elements of $\mathbf{x_i}$ are estimated as follows:

Based on the likelihood that we computed above, we get the following marginal distribution of $x_{ij}$ and its expectation as follows.

$$f(x_{ij}=1)=\frac{\pi_j\theta_j^{(1-y_i)}q_{ij}}{\sum_{k=1}^{G}\pi_k\theta_k^{(1-y_i)}q_{ik}}$$

$$f(x_{ij}=0)=1-f(x_{ij}=1)$$

$$\hat{\delta}_{ij}=\mathrm{E}(x_{ij})=1.f(x_{ij}=1)+0.f(x_{ij}=0)$$

$$\hat{\delta}_{ij}=\mathrm{E}(x_{ij})=\frac{\pi_j\theta_j^{(1-y_i)}q_{ij}}{\sum_{k=1}^{G}\pi_k\theta_k^{(1-y_i)}q_{ik}}$$

Next, the M-step calculates the new estimates of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ given $\mathbf{q_i}$, $\mathbf{y}$ and the current expected values $\hat{\delta}_{ij}$. The formulas for estimating $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, provide the Bayesian maximum

a posteriori (MAP) estimates; however, if the prior information $a_j$ and $b_j$ are set to 0 for all $j$ *g*enomes, these equations provide the maximum likelihood estimates. Letting $\mathrm{N} = \sum_{k=1}^{G}\sum_{i=1}^{R}\hat{\delta}_{ik}$, these estimates are as follows:

$$f(x, y, \pi, \theta) = f(x, y|\pi, \theta)f(\pi, \theta)$$

From the calculation above, we have the following:

$$f(x, y|\pi, \theta, q) \propto \left(\prod_{i=1}^{R}\prod_{j=1}^{G}\left(\pi_j\theta_j^{(1-y_i)}q_{ij}\right)^{x_{ij}}\right)$$

From the prior distribution, we have the following:

$$f(\pi|\mathrm{a}) \propto \prod_{\mathrm{j}=1}^{\mathrm{G}}\pi_{\mathrm{j}}{}^{\mathrm{a_j}-1} \text{ and } f(\theta|\mathrm{b}) \propto \prod_{\mathrm{j}=1}^{\mathrm{G}}\theta_{\mathrm{j}}{}^{\mathrm{b_j}-1}$$

$$f(x, y, \pi, \theta|q) \propto \left(\prod_{i=1}^{R}\prod_{j=1}^{G}\left(\pi_j\theta_j^{(1-y_i)}q_{ij}\right)^{x_{ij}}\right)\prod_{\mathrm{j}=1}^{\mathrm{G}}\pi_{\mathrm{j}}{}^{\mathrm{a_j}-1}\prod_{\mathrm{j}=1}^{\mathrm{G}}\theta_{\mathrm{j}}{}^{\mathrm{b_j}-1}$$

The value of $\pi\ and\ \theta$ that maximizes the likelihood also maximizes the following:

$$L_1 = \left(\prod_{j=1}^{G}\pi_{\mathrm{j}}{}^{\mathrm{a_j}}\prod_{i=1}^{R}(\pi_j)^{x_{ij}}\right)$$

and the following:

$$L_2 = \left(\prod_{j=1}^{G}\theta_{\mathrm{j}}{}^{\mathrm{b_j}}\prod_{i=1}^{R}\left(\theta_j^{(1-y_i)}\right)^{x_{ij}}\right)$$

We need to maximize $L_1$ with the condition $\sum_{j=1}^{G}\pi_j = 1$

$$L_1 = \left(\prod_{j=1}^{G}\pi_{\mathrm{j}}{}^{\sum_{i=1}^{R}x_{ij}+\,\mathrm{a_j}}\right)$$

This reduces to solving for $\pi_{\mathrm{j}}$ similar to the regular multinomial distribution case but $\hat{\delta}_{ij}$ substituted for $x_{ij}$ because $x_{ij}$ is estimated using the EM algorithm.

For the regular multinomial distribution, the solution is obtained as follows:

$$L = \prod_{i=1}^{N} {p_i}^{x_i}$$

Taking the log likelihood:

$$l = \sum_{i=1}^{N} x_i \log(p_i)$$

Put $p_N = 1 - (p_1 + \cdots + p_{N-1})$ in the above equation and partial differentiate w.r.t. $p_i$ gives the following equations:

$$\frac{x_i}{p_i} - \frac{x_N}{1 - (p_1 + \cdots + p_{N-1})} = 0; i = 1 \; to \; N - 1$$

Solving these equations gives: $\widehat{p_i} = \frac{x_i}{\sum_{i=1}^{N} x_i}$

By applying the above result, the value of $\hat{\pi}_j$ that maximizes $L_1$ is given by the following:

$$\hat{\pi}_j = \frac{\sum_{i=1}^{R} \hat{\delta}_{ij} + a_j}{N + \sum_{k=1}^{G} a_k}$$

Similarly, we need to maximize $L_2$ with the condition $\sum_{j=1}^{G} \theta_j = 1$.

$$L_2 = \left( \prod_{j=1}^{G} \theta_{\mathrm{j}}^{\sum_{i=1}^{R} x_{ij} + \mathrm{a_j}} \right)$$

The value of $\hat{\theta}_j$ that maximizes $L_2$ is given by the following:

$$\hat{\theta}_j = \frac{\sum_{i=1}^{R} (1 - y_i) \hat{\delta}_{ij} + b_j}{\sum_{i=1}^{R} (1 - y_i) + \sum_{k=1}^{G} b_k}$$

The E-step is then repeated using the updated estimates of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ followed again by the M-step. These steps are repeated until the expected value of $\mathbf{x_i}$ and the estimates of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ converge to stable values across iterations.

The performance of the modified Pseudo Likelihood model with the Bayesian distribution assumption is evaluated as mentioned in the experimental plan above.

*Mixed Simulation: Evaluation of PathoID 1.0*

To determine the accuracy of the pathogen proportion estimate—a mixture of samples is simulated by mixing reads of multiple pathogens in varying proportions. Specifically, mixture of samples are created from *Y. pestis* KIM D27 (SRR033501), *E. coli* K-12 MG1655(SRR031601), and *F. tularensis subsp. Holarctica* OSU18 (SRR032505) by random sampling of reads and mixed in random proportions to determine the accuracy of the pathogen estimation method. The Table 1 below shows the results from 1,000 random mixtures of approximately 5,770 reads from the *Y. pestis* KIM D27 (SRR033501), *E. coli* K-12 MG1655 (SRR031601) and *F. tularensis subsp. holarctica* OSU18 (SRR032505) datasets.

First we define a naïve approach for comparison with our PathoID approach. In the naïve approach the estimated proportion of genomes is computed by the sum of alignment proportions as follows:

$$\widehat{\pi_j} = \frac{1}{R}\sum_{i=1}^{R}\frac{\exp(a_{ij})}{\sum_{j=1}^{G}\exp(a_{ij})}$$

$\widehat{\pi_j} := Estimated\ proportion\ of\ genome\ j$
$a_{ij} := Alignment\ bit\ score\ (AS)\ that\ read\ i\ is\ from\ genome\ j$
$R := Total\ number\ of\ reads$
$G := Total\ number\ of\ genomes$

The proportion estimates from the naïve approach (Bowtie 2 with default parameters) were extremely biased, typically underestimating the true read proportion, while PathoID 1.0 estimated the true proportions with high precision. In addition, the naïve approach (Bowtie 2) consistently ranked genomes in the sample lower than many genomes that were not in the sample. PathoID 1.0 performed very well in these comparisons with high precision on species proportions and consistently correct genome ranking. However, PathoID 1.0 did fail to identify the *E. coli* substrain in some of the samples—in these cases, PathoID 1.0 identified a nearly identical K12 substrain (~25 base differences in 5 million base genomes) or split the reads between the three K12 substrains in the database.

| Organism | Naïve Mapping (Bowtie 2) | *PathoID 1.0* |
|---|---|---|
| **Average Absolute Difference** | | |
| *Y. pestis* KIM D27 | 0.3160 | 0.0008 |
| *E. coli* K-12 MG1655 | 0.3073 | 0.0092 |
| *F. tularensis* subsp. *holarctica* OSU18 | 0.2708 | 0.0038 |
| **Average Ranking (among 131 full genomes)** | | |
| *Y. pestis* KIM D27 | 13.1 | 2.0 |
| *E. coli* K-12 MG1655 | 7.4 | 2.2 |
| *F. tularensis* subsp. *holarctica* OSU18 | 4.4 | 2.0 |
| **Number of Times Not Ranked in Top 3 (not in Top 10)** | | |
| *Y. pestis* KIM D27 | 964 (627) | 4 (0) |
| *E. coli* K-12 MG1655 | 613 (140) | 67 (1) |
| *F. tularensis* subsp. *holarctica* OSU18 | 311 (79) | 1 (0) |

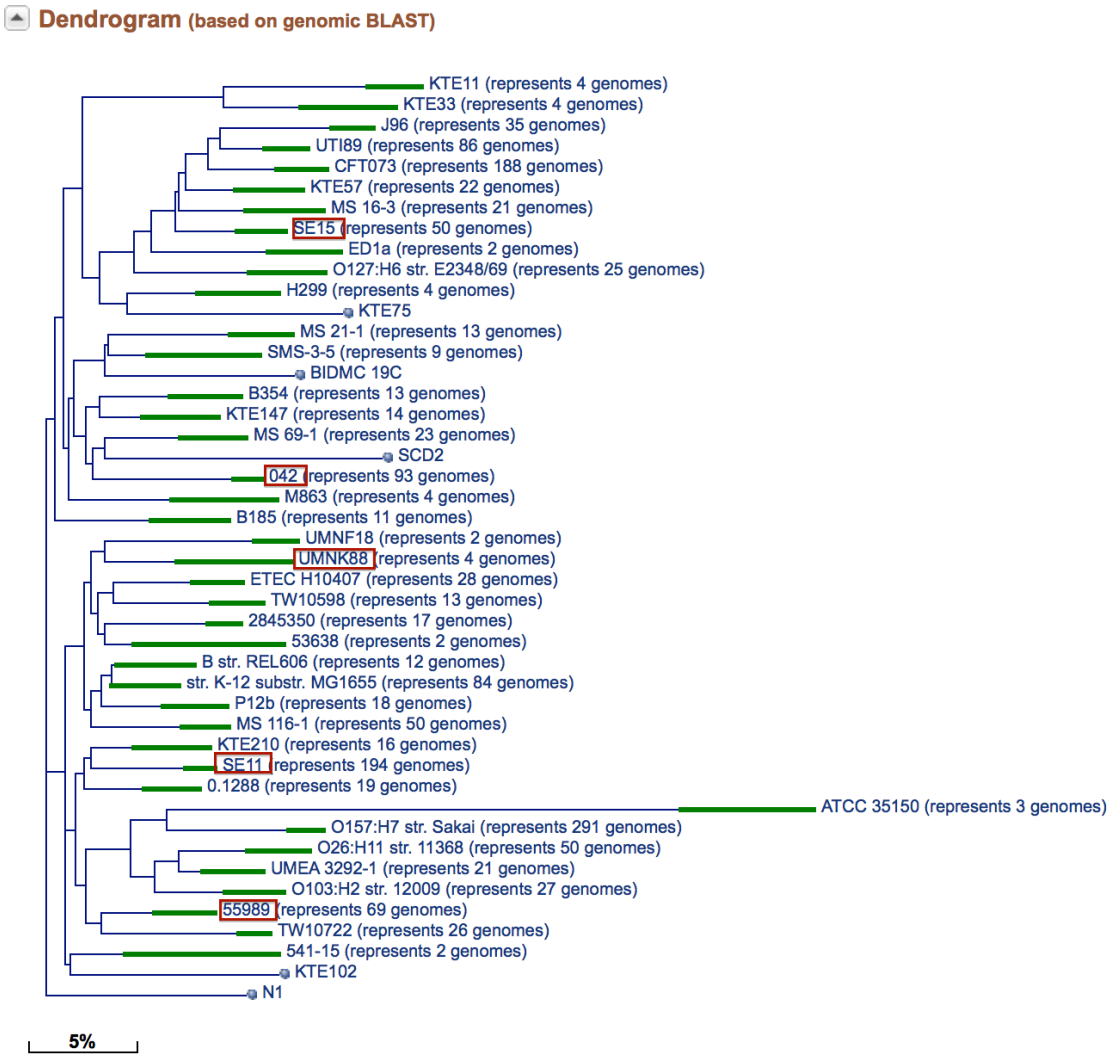
Results from 1000 random mixtures of ~5770 reads from the *Y. pestis* KIM D27 (SRR033501), *E. coli* K-12 MG1655 (SRR031601), and *F. tularensis* subsp. holarctica OSU18 (SRR032505) data sets. The naïve mapping here is done by Bowtie 2 with default parameters. Average absolute difference represents the average of the absolute difference between the true proportion and the estimated proportion. PathoID 1.0 has average absolute difference close to zero and the average rank close to 2 as expected when there is a mixture of three genomes, indicating very good performance. And the number of times PathoID 1.0 has not ranked the three genomes in the top three is very low indicating very good performance in comparison to naïve mapping.

**Table 1: Mixed Proportion Simulation results**

*Equal Proportion Simulation Study: Comparing PathoID 1.0 vs 2.0 and sensitivity of PathoID 2.0 to different prior θ values*

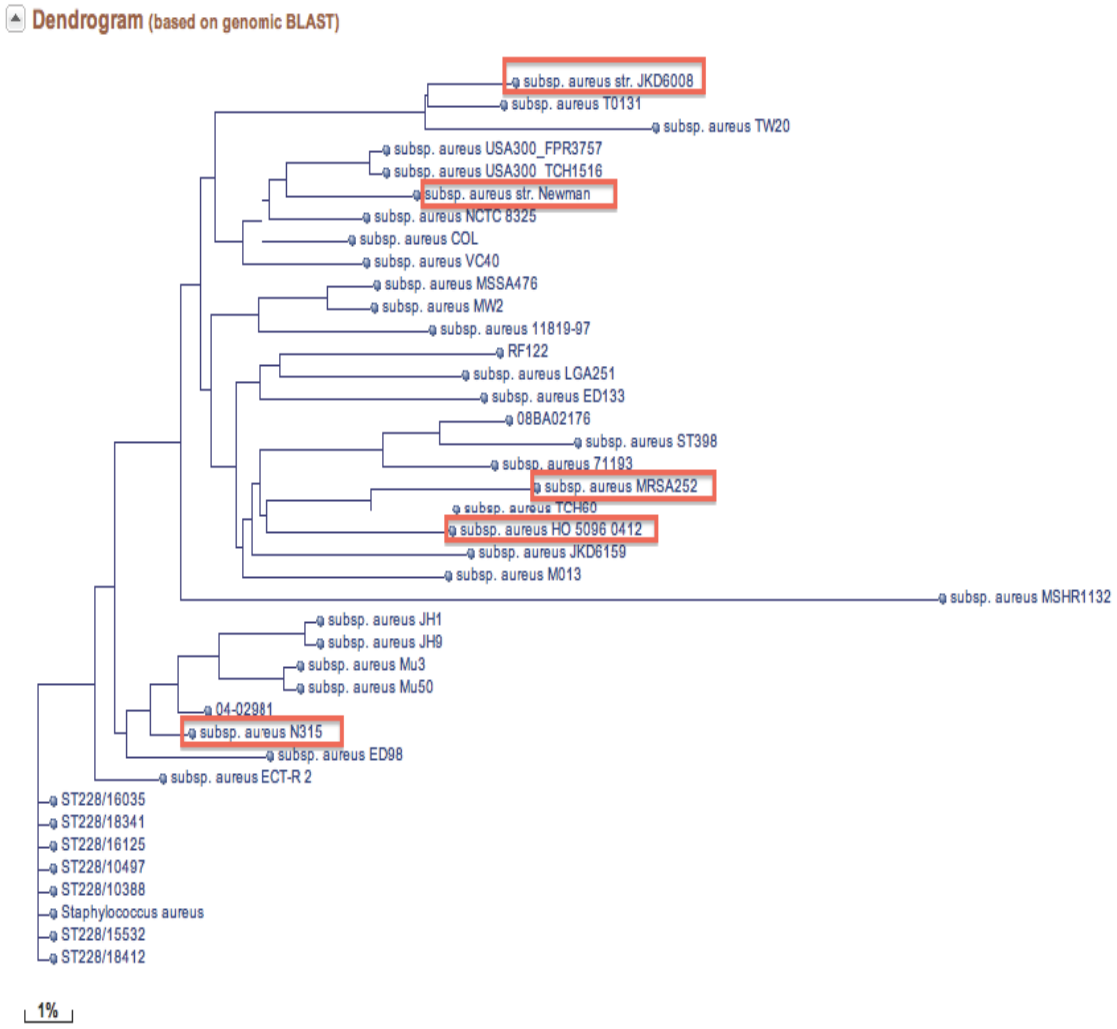
We simulated sequencing reads from 25 strains of bacteria which includes five *Escherichia coli* strains (O42, 55989, SE11, SE15 UMNK88), five *Staphylococcus aureus* substrains (JKD6008, Newman, MRSA252, HO 5096 0412, N315), five *Streptococcus pneumoniae* substrains (670, ATCC_700669, G54, Hungary19A, Taiwan19F) and ten other common bacterial strains (*Bacteroides fragilis* 638R, *Bifidobacterium bifidum* BGN4, *Clostridium perfringens* ATCC 13124, *Enterococcus faecalis* V583, *Haemophilus influenzae* 10810, *Neisseria meningitidis* MC58, *Pseudomonas aeruginosa* DK2, *Staphylococcus epidermidis* ATCC 12228, *Streptococcus mitis* B6, *Streptococcus mutans* UA159). The phylogenetic relationships between these strains and other strains available in the NCBI database are given in Figure 1, Figure 2 and Figure 3 below. We used the Mason read simulator (Holtgrewe, 2010) to generate five sets of 100,000 reads for each strain simulating 100 bp single-end sequencing reads using an 'Illumina-like' sequencing error model; Mason parameters: 'illumina -s ## -N 100000 -sq -n 100 -i -hs 0.0 -hi 0 -hnN -nN' (-s (Seeds) = 1101, 1102, 1103, 1104, 1105). We used PathoLib (-t 2 --subTax) to generate a reference library containing all bacteria. We used PathoMap (described below; default parameters) to index and align the reads to the bacterial library. PathoMap automatically splits the bacterial library into smaller parts (< 4.3GB in size) that the Bowtie2 aligner can handle process and combines the alignment filesthem together for the final alignment results. We then applied PathoID (versions 1.0 and 2.0) to the simulated datasets. PathoID version 1.0 uses the mapping

quality score in the alignment file and has no option to accept theta prior information, while PathoID version 2.0 uses the sequence alignment score and has options to accept theta prior information. PathoID version 2.0 was applied with default parameters and with two informative priors (low, high). The low informative prior corresponds to '-thetaPrior 1000' (equivalent to adding 1,000 non-unique reads to each genome) and high informative prior corresponds to '-thetaPrior 10**88'. The thetaPrior value here represents the number of non-unique reads that are not subject to reassignment. When we use a high thetaPrior value, it essentially removes the theta parameter and the non-unique indicator $y_i$ from the model. This is useful, when we know that there are multiple closely related strains possible in a sample, as in a disease caused by multiple strains of pathogens.



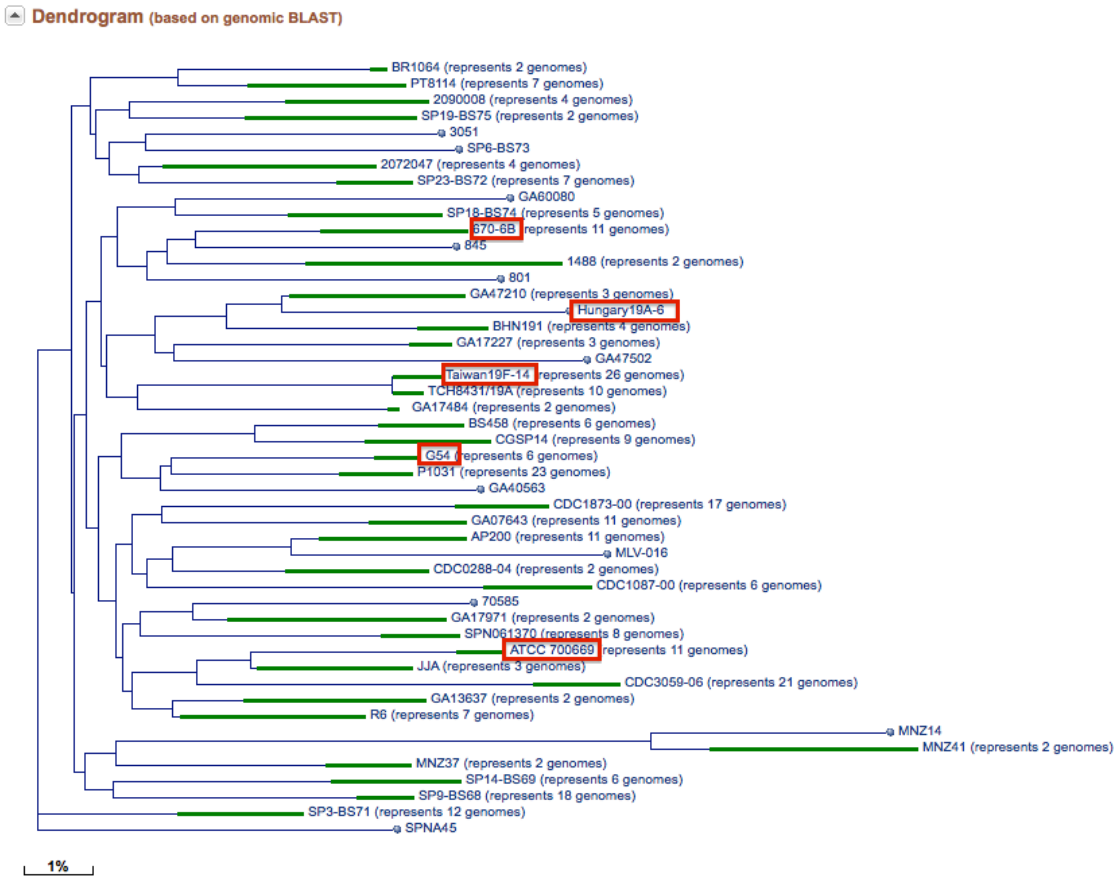

Five substrains of *Escherichia coli* were selected at random to cover all parts of the taxonomy tree.
Previously downloaded from:
http://www.ncbi.nlm.nih.gov/genome/?term=escherichia%20coli

**Figure 1: Selected 5 *Escherichia coli* substrains in the Taxonomy tree**




Five substrains of *Staphylococcus aureus* were selected at random to cover all parts of the taxonomy tree.
Previously downloaded from:
http://www.ncbi.nlm.nih.gov/genome/?term=staphylococcus%20aureus

**Figure 2: Selected 5 *Staphylococcus aureus* substrains in the Taxonomy tree**




Five substrains of *Streptococcus pneumoniae* were selected at random to cover all parts of the taxonomy tree.
Previously downloaded from:
http://www.ncbi.nlm.nih.gov/genome/?term=streptococcus%20pneumoniae

**Figure 3: Selected 5 *Streptococcus pneumoniae* substrains in the Taxonomy tree**

Our simulated data consisted of five sets of 100,000 simulated Illumina reads derived from each of the 25 strains of bacteria. First, we processed the reads for each of the 25 bacterial strains individually using both PathoID version 1.0 and 2.0 with default parameter values and also PathoID 2.0 using a highly informative prior. Although PathoID 1.0 was able to estimate the correct proportions of reads at the *species level* (100% to the particular species) for each of these 25 samples, it was not able to estimate the correct proportions of reads at the *strain level* (100% to the particular strain) for six samples (Table 2). In contrast, PathoID 2.0 using default parameters estimated the correct proportions of reads at the *strain level* (100% to the particular strain) for all the 25 samples. PathoID 2.0 with an informative prior estimated the correct proportions of reads at the *strain level* (100% to the particular strain) for 24 of the 25 samples, but was unable to estimate the correct proportion for one sample with *S. aureus* N315. The results are consistent for all five sets of simulated samples. The result for the first set of simulated samples is shown in Table 2.

We also combined all the reads from the 25 strains, to create a dataset with all bacteria in equal proportions (expected proportion for each strain: 4%). Again, we applied both PathoID version 1.0 and 2.0 with default parameter values and added two informative theta priors (low and high as mentioned above). We saw marked improvement in PathoID version 2.0 over version 1.0, including increased accuracy with

informative priors (Table 2). For the 10 bacterial species that only included one strain per species, all methods performed well and estimated the correct read proportions. For the three species that contained multiple strains (*E. coli*, *S. aureus*, and *S. pneumoniae*), PathoID 1.0 and 2.0 at default parameters were able to identify all of the strains present, but struggled to estimate the correct read proportions. This failure demonstrates the tendency of the PathoID algorithm (at default parameters) to identify a single strain for each species, and if multiple strains or substrains are present it may reassign too many of the reads to a single strain. This tendency is an advantage in cases where there is only one strain of each species in the sample, but it leads to inaccuracies in the proportion estimates when multiple strains or substrains of the same species are present in the sample. The result of PathoID 2.0 with a highly informative prior matched closely with the expected proportions for 24 of the 25 strains, including 14 of the 15 cases with multiple strains of the same species. This demonstrates the value of using a highly informative prior when there are multiple strains of the same species in the sample, but we note that this comes at reduced effectiveness when there is only a single strain of each species in the sample.

The one strain that was not estimated well using a highly informative prior was S. aureus N315, which had a final average read assignment percentage of 1.01%. After further evaluation, we observed that PathoID failed with this strain due to the 'sequencing errors' in our simulated reads that caused some of the N315 reads to align more closely to the related strains. This phenomenon limited the ability of PathoID to correctly estimate the correct read proportions for this strain.

| | PathoID 2.0 | | | PathoID 1.0 | | | Bowtie2 |
|---|---|---|---|---|---|---|---|
| Sample | Strain Level Rank | Strain Level Proportions | Thetaprior (High) Strain Level Proportions | Species Level Proportions | Strain Level Rank | Strain Level Proportions | Strain Level Proportions |
| *Bacteroides fragilis* 638R | 1 | 99.94% | 99.87% | 99.97% | 1 | 99.97% | 55.62% |
| *Bifidobacterium bifidum* BGN4 | 1 | 99.99% | 99.99% | 99.99% | 1 | 99.99% | 57.31% |
| *Clostridium perfringens* ATCC_13124 | 1 | 99.90% | 99.89% | 99.91% | 1 | 99.91% | 72.58% |
| *Enterococcus faecalis* V583 | 1 | 99.89% | 99.89% | 99.92% | 1 | 99.92% | 46.15% |
| *Escherichia coli* 042 | 1 | 99.82% | 99.71% | 99.87% | 1 | 99.87% | 31.49% |
| *Escherichia coli* 55989 | 1 | 99.71% | 94.74% | 99.75% | 1 | 99.75% | 8.84% |
| *Escherichia coli* SE11 | 1 | 99.67% | 99.40% | 99.73% | 1 | 99.73% | 12.41% |
| *Escherichia coli* SE15 | 1 | 99.81% | 99.63% | 99.87% | 1 | 99.87% | 18.24% |
| *Escherichia coli* UMNK88 | 1 | 99.89% | 99.78% | 99.93% | 1 | 99.93% | 16.16% |
| *Haemophilus influenzae* 10810 | 1 | 99.79% | 99.77% | 99.85% | 1 | 99.85% | 49.68% |
| *Neisseria meningitidis* MC58 | 1 | 99.83% | 97.78% | 81.24% | 2 | 18.72% | 20.90% |
| *Pseudomonas aeruginosa* DK2 | 1 | 99.92% | 99.91% | 99.95% | 1 | 99.95% | 30.31% |
| *Staphylococcus epidermidis* ATCC_12228 | 1 | 99.71% | 99.28% | 99.81% | 1 | 99.81% | 48.70% |
| *Streptococcus mitis* B6 | 1 | 99.73% | 99.73% | 99.75% | 1 | 99.75% | 90.32% |
| *Streptococcus mutans* UA159 | 1 | 99.85% | 99.73% | 90.39% | 2 | 9.59% | 43.61% |
| *Stapylococcus aureus* HO_5096_0412 | 1 | 99.82% | 99.75% | 99.88% | 1 | 99.88% | 25.30% |
| *Stapylococcus aureus* JKD6008 | 1 | 99.78% | 95.53% | 97.52% | 2 | 2.35% | 8.51% |
| *Stapylococcus aureus* MRSA252 | 1 | 99.80% | 99.10% | 99.86% | 1 | 99.86% | 19.00% |
| *Stapylococcus aureus* N315 | 1 | 99.83% | 51.70% | 95.23% | 3 | 0.45% | 6.02% |
| *Stapylococcus aureus* Newman | 1 | 99.82% | 96.96% | 93.29% | 2 | 6.61% | 8.15% |
| Streptococcus pneumoniae 670-6B | 1 | 99.77% | 99.76% | 99.85% | 1 | 99.85% | 17.95% |
| Streptococcus pneumoniae ATCC_700669 | 1 | 99.84% | 99.81% | 99.87% | 1 | 99.87% | 15.80% |
| *Streptococcus pneumoniae* G54 | 1 | 99.69% | 99.64% | 99.76% | 1 | 99.76% | 15.01% |
| *Streptococcus pneumoniae* Hungary19A-6 | 1 | 99.77% | 99.75% | 99.84% | 1 | 99.84% | 21.54% |
| *Streptococcus pneumoniae* Taiwan19F-14 | 1 | 99.79% | 92.40% | 99.30% | 2 | 0.58% | 11.10% |

**25 Samples with single strain of Bacteria (Expected proportion: 100%)**
The simulated data consists of five sets of 100,000 simulated Illumina reads derived from each of the 25 strains of bacteria, which include five Escherichia coli substrains, five Staphylococcus aureus substrains, five Streptococcus pneumoniae substrains and ten other commonly occurring human bacterial strains. PathoID is our module for reassignment of reads based on the Bayesian Pseudo Likelihood mixture model. PathoID version 1.0 uses the mapping quality score in the alignment file and has no option to accept theta prior information, while PathoID version 2.0 uses the sequence alignment score and has options to accept theta prior information. We applied PathoID version 1.0 and 2.0 to the individual alignment file generated by PathoMap for each of the 25 bacterial strains separately. PathoID 1.0 identified correctly with the right proportions to the species level for all the strains of bacteria, but was not able to identify correctly to the strain level for few strains of bacteria. However, PathoID 2.0 correctly identified with the right proportions to the strain level for all of the 25 bacterial strains. The results are consistent for all the five sets of samples. Here, we present the first set of the samples.

**Table 2: Single Strain Samples**

| Organism | PathoID 1.0 | PathoID 2.0 | | |
|---|---|---|---|---|
| | Default | Default | ThetaPrior (Low) | ThetaPrior (High) |
| *Bacteroides fragilis* 638R | 4.00% | 3.99% | 3.99% | 3.99% |
| *Bifidobacterium bifidum* BGN4 | 4.00% | 3.99% | 3.99% | 3.99% |
| *Clostridium perfringens* ATCC_13124 | 3.99% | 3.99% | 3.99% | 3.99% |
| *Enterococcus faecalis* V583 | 3.98% | 3.99% | 3.99% | 4.00% |
| *Escherichia coli* 042 | 17.15% | 4.01% | 4.01% | 4.02% |
| *Escherichia coli* 55989 | 0.57% | 0.50% | 0.52% | 3.83% |
| *Escherichia coli* SE11 | 0.28% | 10.10% | 10.03% | 3.74% |
| *Escherichia coli* SE15 | 0.71% | 3.43% | 3.44% | 3.82% |
| *Escherichia coli* UMNK88 | 1.29% | 1.95% | 1.99% | 4.16% |
| *Haemophilus influenzae* 10810 | 3.98% | 3.99% | 3.99% | 3.99% |
| *Neisseria meningitidis* MC58 | 3.25% | 3.99% | 3.99% | 3.92% |
| *Pseudomonas aeruginosa* DK2 | 4.00% | 3.99% | 3.99% | 3.99% |
| *Staphylococcus epidermidis* ATCC_12228 | 3.97% | 3.98% | 3.98% | 3.98% |
| *Streptococcus mitis* B6 | 2.94% | 3.82% | 3.82% | 3.94% |
| *Streptococcus mutans* UA159 | 3.58% | 3.99% | 3.99% | 3.99% |
| *Stapylococcus aureus* HO_5096_0412 | 0.31% | 1.76% | 1.77% | 3.80% |
| *Stapylococcus aureus* JKD6008 | 0.05% | 15.82% | 15.75% | 3.79% |
| *Stapylococcus aureus* MRSA252 | 0.69% | 1.46% | 1.48% | 3.85% |
| *Stapylococcus aureus* N315 [1] | 0.00% | 0.68% | 0.70% | 1.01% |
| *Stapylococcus aureus* Newman | 0.15% | 0.33% | 0.34% | 3.48% |
| Streptococcus pneumoniae 670-6B | 0.92% | 3.28% | 3.23% | 4.43% |
| Streptococcus pneumoniae ATCC_700669 | 0.29% | 0.99% | 1.02% | 4.23% |
| *Streptococcus pneumoniae* G54 | 0.16% | 0.44% | 0.47% | 3.35% |
| *Streptococcus pneumoniae* Hungary19A-6 | 19.42% | 14.75% | 14.70% | 4.33% |
| *Streptococcus pneumoniae* Taiwan19F-14 | 0.00% | 0.66% | 0.69% | 2.83% |

1: Strain with maximum difference

**One Sample with 25 strains of Bacteria in equal proportions**
**Shown here is the proportion of reads (Expected strain proportion: 4%)**

The result of PathoID 2.0 with high theta priors matched closely with the expected proportion of 4%, except for one bacterial strain in particular - S. aureus N315, where the proportion was less by about 3%. The low informative prior corresponds to '-thetaPrior 1000' and high informative prior corresponds to '-thetaPrior 10**88'.

**Table 3: Equal Proportion Simulation Study Results**

*Sensitivity Analysis with unequal proportions*

We performed a simulation with unequal proportions of reads with the 25 strains of bacteria from the study mentioned above to perform sensitivity analysis of PathoID when there are closely related strains in different proportions. We noticed as shown in Table 4 below that PathoID 2.0 performed very well. The results were consistent with the mixtures at equal proportions, that is, the strains for which the proportions were not accurate with the mixtures at equal proportions matched with that of the mixtures at random proportions.

The maximum average difference between the true proportion and the proportion predicted by PathoID is 2.5% and the total difference is only 6.48%. Even in the extreme case where multiple closely related strains are present and in combination with read sequencing errors, the total difference is less than 8%. This shows the capability of PathoID to distinguish even closely related strains in the samples and also the ability to estimate accurately the proportions of pathogens found in the samples. Hence, we can conclude that if the reads are of good quality, the proportion estimates of pathogens by PathoID is very reliable.

| Sample Id | Sample 1 | | | Sample 2 | | |
|---|---|---|---|---|---|---|
| Organism | Actual | PathoID2 | \|Diff\| | Actual | PathoID2 | \|Diff\| |
| *Bacteroides fragilis* 638R | 7.41% | 7.40% | 0.01% | 4.00% | 3.99% | 0.01% |
| *Bifidobacterium bifidum* BGN4 | 3.70% | 3.70% | 0.01% | 8.00% | 8.00% | 0.00% |
| *Clostridium perfringens* ATCC_13124 | 11.11% | 11.09% | 0.02% | 0.00% | 0.00% | 0.00% |
| *Enterococcus faecalis* V583 | 7.41% | 7.42% | 0.01% | 0.00% | 0.00% | 0.00% |
| *Escherichia coli* 042 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| *Escherichia coli* 55989 | 3.70% | 3.58% | 0.12% | 4.00% | 3.92% | 0.08% |
| *Escherichia coli* SE11 | 7.41% | 7.08% | 0.32% | 4.00% | 3.77% | 0.23% |
| *Escherichia coli* SE15 | 3.70% | 3.48% | 0.23% | 8.00% | 7.97% | 0.03% |
| *Escherichia coli* UMNK88 | 7.41% | 7.76% | 0.36% | 0.00% | 0.00% | 0.00% |
| *Haemophilus influenzae* 10810 | 3.70% | 3.68% | 0.02% | 12.00% | 11.97% | 0.03% |
| *Neisseria meningitidis* MC58 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| *Pseudomonas aeruginosa* DK2 | 3.70% | 3.68% | 0.02% | 4.00% | 3.99% | 0.01% |
| *Staphylococcus epidermidis* ATCC_12228 | 3.70% | 3.70% | 0.01% | 4.00% | 3.97% | 0.03% |
| *Streptococcus mitis* B6 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| *Streptococcus mutans* UA159 | 3.70% | 3.70% | 0.00% | 8.00% | 7.97% | 0.03% |
| *Stapylococcus aureus* HO_5096_0412 | 0.00% | 0.00% | 0.00% | 8.00% | 7.97% | 0.03% |
| *Stapylococcus aureus* JKD6008 | 3.70% | 3.34% | 0.37% | 4.00% | 3.65% | 0.35% |
| *Stapylococcus aureus* MRSA252 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| *Stapylococcus aureus* N315 [1] | 3.70% | 1.03% | 2.67% | 4.00% | 0.99% | 3.01% |
| *Stapylococcus aureus* Newman | 3.70% | 3.37% | 0.33% | 4.00% | 3.47% | 0.53% |
| Streptococcus pneumoniae 670-6B | 7.41% | 8.21% | 0.80% | 4.00% | 4.07% | 0.07% |
| Streptococcus pneumoniae ATCC_700669 | 3.70% | 3.84% | 0.13% | 12.00% | 12.70% | 0.70% |
| *Streptococcus pneumoniae* G54 | 3.70% | 3.02% | 0.69% | 4.00% | 3.12% | 0.88% |
| *Streptococcus pneumoniae* Hungary19A-6 | 3.70% | 3.89% | 0.18% | 4.00% | 4.08% | 0.08% |
| *Streptococcus pneumoniae* Taiwan19F-14 | 3.70% | 2.49% | 1.22% | 0.00% | 0.00% | 0.00% |
| Max Difference | | 2.67% | 2.67% | | 3.01% | 3.01% |
| Total Difference | | 7.52% | 7.52% | | 6.09% | 6.09% |

1: Strain with maximum difference

**Result of PathoID 2.0 with high prior value on unequal proportions mix of 25 strains of bacteria**

We ran a simulation with the 25 strains of bacteria mixed in unequal proportions and created ten different samples, which we analyzed using PathoID 2.0 with very high theta priors. Shown here are two samples and the results are consistent across the ten samples. The maximum average difference between the true proportion and the proportion predicted by PathoID 2.0 is 2.5% and the total difference is only 6.48%. Even in the extreme case where multiple closely related strains are present, the total difference is less than 8%.

**Table 4: Unequal proportion Simulation Study Results**

## Aim 1B

Evaluate how many sequencing reads are needed (read coverage) to estimate pathogens proportions within a given confidence limit, and determine how long it takes to estimate microbial proportions within a given confidence limits using our mixture model and in comparison to other methods.

### *Objective*

Develop statistical methods for deriving confidence intervals for metagenomic sequencing data. Establish the minimum number of reads required to estimate microbial proportions to a given confidence level. Establish the time and read coverage required for estimating the pathogen proportions to these limits.

### *Rationale*

It costs time and money to perform sequencing and analysis. Hence, it is important to estimate the pathogen proportions accurately within the given confidence limit using minimal number of reads to save on cost and time. Our methods and this simulation study will give insights into the number of reads and time required under different scenarios.

### *Experimental Plan*

To find the effect of the number of reads on accuracy, a simulation of samples with varying number of reads will be performed. As an example, we will use 92,370 sequencing reads from *E.coli* O104:H4 pathogen (SRR227300). A simulation of 1000 samples with randomly picked sets of 9,237 reads (0.13X read coverage), 924 reads (0.01X read coverage), and 92 reads (0.001X read coverage) will be performed to compare the true proportions with the estimated proportions using our Bayesian mixture model approach.

### *Comparison of Metagenomics Analysis methods*

The key to identifying and estimating the correct proportions of pathogens present in a sample is to perform the task of reassignment of reads to the correct genome after alignment of reads by aligners such as Bowtie and BLAST. The Bayesian Pseudo Likelihood mixture model with EM (Expectation Maximization) parameter estimation mentioned above (PathoID 2.0) for reassignment is a powerful approach that is expected to be both accurate and fast. The performance of EM approach is compared with other methods such as naïve alignment, ReadScan (Naeem, Rashid & Pain, 2013), RINS (Bhaduri *et al.*, 2012), PhymmBL (Brady & Salzberg, 2009), MetaPhlAn (Segata *et al.*, 2012) and MEGAN (Huson *et al.*, 2007). We also applied an alignment approach using the Trinity assembler (Grabherr, Haas, Yassour et al., 2011) to assemble high-quality contiguous sequences (contigs) from the reads, followed by the probabilistic alignment of the contigs to the database.

### *Real data samples*

For the comparison of our Bayesian mixture model with other methods, real data samples are selected from fecal specimens obtained from patients with diarrhea during the 2011 European outbreak of Shiga-toxigenic *Escherichia coli* (STEC) O104:H4 (NCBI accession: ERP001956) (Rohde, Qin, Cui et al., 2011; Turner, 2011). Misidentification was among the issues that resulted in a 3-wk delay in accurate diagnosis of this outbreak, resulting in over 3800 infections across 13 countries in Europe with 54 deaths (Frank, Werber, Cramer et al., 2011).

### *Analysis*

For this example, we used the full dataset of 92,370 reads, representing 1.3X coverage of the reference O104:H4 genome, as well as reduced datasets using 1,000 random subsamples reads for each of the following sample sizes: 9,237 (0.13X), 924 (0.01X), and 92 (0.001X). For the smaller subsets (92, 924, 9,237), we compared the average accuracy and range across samples for each method. These smaller sets were designed to evaluate algorithmic performance when the reads are generated using multiplexed sequencing runs or when they originate from contaminated samples that may be dominated by other genomic sources. However, we note that for MEGAN (graphical user interface), and PhyloPhythiaS (manual webserver), and the assembly approach, we did not use 1,000 random datasets; rather we used a single random sample of each data set size, as they would either require thousands of manual submissions or an excessive amount of

computation time. Table 5 contains the average accuracy and range across samples for each algorithm.

**Naïve alignment, PhymmBL, and MetaPhlAn**:

The naïve algorithm consistently assigned around 16% of the read probability to the O104:H4 strain independent of the number of reads used. However, on average between 7.4% and 9.4% of the read probability was assigned to the 55989 strain of *E. coli*, which is the closest fully sequenced genome to the O104:H4 strain (Rohde *et al.*, 2011; Turner, 2011). Several other *E. coli* strains received 1-3% of the reads, and several species in the *Shigella* genus also received 1-2% of the reads. In all, roughly 93% of the read probabilities were assigned to an *E. coli* strain. The PhymmBL algorithm assigned 14.7% on average to O104:H4 strain and exhibited similar profiles of false mapping to other strains and species. Overall the performance of PhymmBL was only slightly better than the naïve approach. The MetaPhlAn algorithm aligns reads to taxonomic clade-specific markers, which in its current implementation can only identify DNA templates at the species level—and therefore cannot distinguish between strains or substrains of the same species. In addition, because it only uses short clade markers are used, merely 815 (0.9%) of the reads were assigned by MetaPhlAn. Of these reads only 90.0% were aligned to *E. coli*, whereas 9.6% were incorrectly assigned to *S. dysenteriae*. The method gave inconsistent results for the subsamples of 9,237 and most of the time failed to assign any reads to *E. coli* for the subsamples of 92 and 924. From these approaches, it is clear that an *E. coli* strain is present in the sample and the naïve and PhymmBL approaches

point to O104:H4 as the most likely source, but all results are ambiguous as to whether there are multiple *E. coli* strains or other species present in the sample.

**Genome assembly approach**:

For the assembly approach, no contigs were generated from the 92 and 924 read data sets. For the data set with 9237 reads, only five contigs were generated, ranging in length from 221 bases to 442 bases in length. Although these five contigs best matched to the O104:H4 strain, they also aligned to several other (incorrect) genomes in the database. Finally, on the complete sequencing run representing 1.3X coverage of the genome, the assembler constructed 3637 short contigs with only 21.5% of the contig mapping probability being assigned to the correct strain. Therefore, although this approach is a slight improvement over the naïve approach, it is clear that a single sequencing run for a purified (single source) sample is not sufficient for strain attribution using an assembly-based approach.

**PathoID reassignment**:

In contrast, PathoID reassigned, on average, 99.4% of the read probability directly to the O104:H4 strain for the data sets with 92 reads and averaged 99.6% of the reads correctly for the larger data sets. These results imply that PathoID is a substantial improvement over naïve mapping, context mapping, and assembly-based methods for species identification and strain attribution.

**Identification of the nearest genome**:

The results from the MEGAN and PhyloPythiaS analyses were not included in the previous section because the annotation tables used by these approaches do not contain the O104:H4 strain (and cannot be manually added by the user). For this reason, we removed the O104:H4 strain from our reference database and reanalyzed the query reads using the naïve mapping and PathoID reassignment. In addition, we note that the PhyloPythiaS web server only allowed for a maximum of 10,000 reads for each submission, so the results presented here were based on random sets of 92, 924, and 9237 only (and not the full data set).

For the naïve mapping with O104:H4 removed, most of the aligned reads (99.8%) mapped to at least one strain of *E. coli*, thus rapidly and clearly identifying the species of origin. However, 96.1% of these reads aligned ambiguously to multiple *E. coli* strains. The 55989 strain received the largest proportion of the aligned reads (9.5%), followed by the O103:H2 strain (3.2%), the B7A, O26:H11, E24377A, and the E22 strains (3.1%), then the SE11 and IAI1 strains (3.0%). Therefore, although the correct species was easily identified using a naïve mapping strategy, the identification of the correct strain within the species proves to be more difficult, and a simple mapping strategy leaves much uncertainty in the process of identifying the strain most similar to the origin strain. This uncertainty can prove to be important for *E. coli*—which contains both benign and harmful strains—as the misclassification of the origin or nearest strain might lead to negative economic and human health consequences.

In contrast, the lowest common ancestor approach utilized by MEGAN assigned 80.2% of the reads to the family taxonomic level or higher. The remaining reads were assigned at the genus level; 19.7% of the total reads were assigned to the *Escherichia* genus and 0.2% of the reads were incorrectly assigned to the *Shigella* genus. MEGAN did not assign any reads at the species or strain level for any of the data sets. PhyloPhythiaS also performed poorly on this example: Overall, >84% of the reads were assigned to the family level or above, and <50% of all the reads were correctly assigned *E. coli* taxonomy levels. Furthermore, 32 incorrect genera received more reads than *Escherichia*, and five incorrect species received more reads than *E. coli.*

After application of the PathoID reassignment, 89.5% of the reads were reassigned to the 55989 strain. The genomes with the next highest read proportions were the O157:H7 strain (3.2%) and the O103:H2 strain (1.1%). Therefore, even though our approach did not completely converge on one genome (as it should not, because in this analysis the origin strain was not present in the database), it is clear that PathoID can clearly and definitively identify the closest fully sequenced neighboring strain with high confidence.

To evaluate whether the lack of sensitivity for MEGAN and PhyloPythiaS is due to the missing O104:H4 annotation, we applied MEGAN and PhyloPhythiaS to our analysis of reads from the *E. coli* K-12 MG1655 substrain, which is contained in the annotation. For MEGAN, the result was similar, in that all of the reads were assigned at the genus level or higher. For PhyloPythiaS, 98.5% of the reads were assigned to the genus level or above, and 34.7% of the reads were assigned to incorrect taxonomies.

The *E. coli* species only received 1.4% of the reads, and no reads were assigned at the strain or substrain level. Therefore, these methods can fail to identify substrains, even when they are present in the annotation.

**Computational Time**:

MetaPhlAn was by far the fastest algorithm (Table 5), requiring only 1 minute to complete because it aligns the reads to a set of small clade markers, however, the approach assigned less than 1% of the reads in this example. The naïve approach required 38 minutes for a BLAST alignment and 13 minutes for Bowtie2. PathoID and MEGAN used the naïve alignments and required an additional 7 minutes and 3 minutes, respectively. PylopythiaS required a total of 7 minutes to assign 9,237 reads. PhymmBL required ~36hrs of database preprocessing, and then approximately 2 hours to assign the reads. Finally, the assembly approach required 30 min to complete.

## *Conclusion*

It is clear that PathoID is the most effective algorithm for strain identification. Based on this experiment, we recommend that for single strain identification, about 0.1X coverage of reads is sufficient to get more than 99% accuracy using PathoID.

| | % of Reads to Correct Genome (Second Highest) [Range for 1,000 Random Samples] | | | | Time Required (Full Dataset) |
|---|---|---|---|---|---|
| **Number of Reads (Coverage)** | **92 (0.001X)** | **924 (0.01X)** | **9,237 (0.13X)** | **92,370 (1.3X)** | |
| Naïve mapping | 12.9 (6.5) [7.5-20.9] | 12.9 (6.1) [10.5-15.5] | 12.9 (7.4) [12.2-13.5] | 12.9 (7.4) | BLAST: 38min Bowtie2: 13min |
| PathoID | 99.4 (0.5) [95.1-100.0] | 99.6 (0.3) [98.0-100.0] | 99.6 (0.3) [99.3-99.8] | 99.6 (0.3) | Naïve + 7min |
| PhymmBL | 14.7 (7.0) [4.3-26.1] | 14.7 (7.0) [11.3-18.5] | 14.7 (7.1) [13.6-15.7] | 14.7 (7.1) | 13hrs** |
| MetaPhlAn (species only) | -- | 36.1 (0.0) [0.0-100.0] | 96.9 (2.4) [54.1-100.0] | 90.0 (9.6) | 1 min |
| Trinity Contigs | -- | -- | 70.8 (22.6) | 21.5 (13.4) | 30 min |
| PhylopythiaS* | | | | | 7 min** |
| Family (or above) | 47.8 (7.6) | 48.4 (2.2) | 45.6 (2.8) | -- | |
| Genus | 0.0 (2.2) | 0.1 (1.6) | 0.1 (1.2) | -- | |
| Species | 0.0 (0.1) | 0.0 (0.2) | 0.1 (0.3) | -- | |
| MEGAN* | | | | | Naïve + 3min |
| Family (or above) | 84.7 (0.0) | 79.5 (0.0) | 80.2 (0.0) | 80.2 (0.0) | |
| Genus | 16.3 (0.0) | 20.5 (0.0) | 19.6 (0.2) | 19.7 (0.2) | |
| Species/Strain* | -- | -- | -- | -- | |

Shown here are the results from the application of several species identification approaches to subsets of the 92,370 sequencing reads from the first O104:H4 Ion Torrent sequencing run (Guilford, CT) (SRR227300) (Li, Xi, Zhao et al., 2011). It shows the percentages of all reads assigned to the correct genome along with the second highest scoring genome in parenthesis. It is clear that PathoID is the most effective algorithm for strain identification. Based on this experiment, we recommend that for single strain identification, about 0.1X coverage of reads is sufficient to get more than 99% accuracy using PathoID. For MEGAN and PhyloPythiaS, the O104:H4 annotation is not available, so the nearest strain E. coli 55989 was considered the 'correct' strain.

*Source strain was not contained in annotation
** PhymmBL also required 36 hours of database preprocessing, and PhyloPhythiaS was only applied to 9,237 reads and not the whole dataset.

**Table 5: PathoScope comparison on O104:H4 dataset**

## Aim 1C

Develop a complete software pipeline for metagenomics analysis of next-generation sequencing data in collaboration with others with leading and major contribution to the design and development of the pipeline

### *Objective*

Develop a complete and flexible 'plug-n-play' software pipeline, called PathoScope, for the metagenomic analysis of samples, including library preparation, alignment of reads, and reassignment of reads to the correct source genome, and preparing a complete report with the proportion of each the genomes that are present in the sample.

### *Rationale*

There are several software packages that are currently available for performing specific parts of these types of metagenomic analysis. However, none of them provides a complete solution to perform the complete analysis that includes library preparation, target alignment and filtration, and reassignment to the correct source genome. The software pipeline developed in this project is efficient, utilizing a powerful the Bayesian framework, and provides a complete metagenomics analysis toolkit.

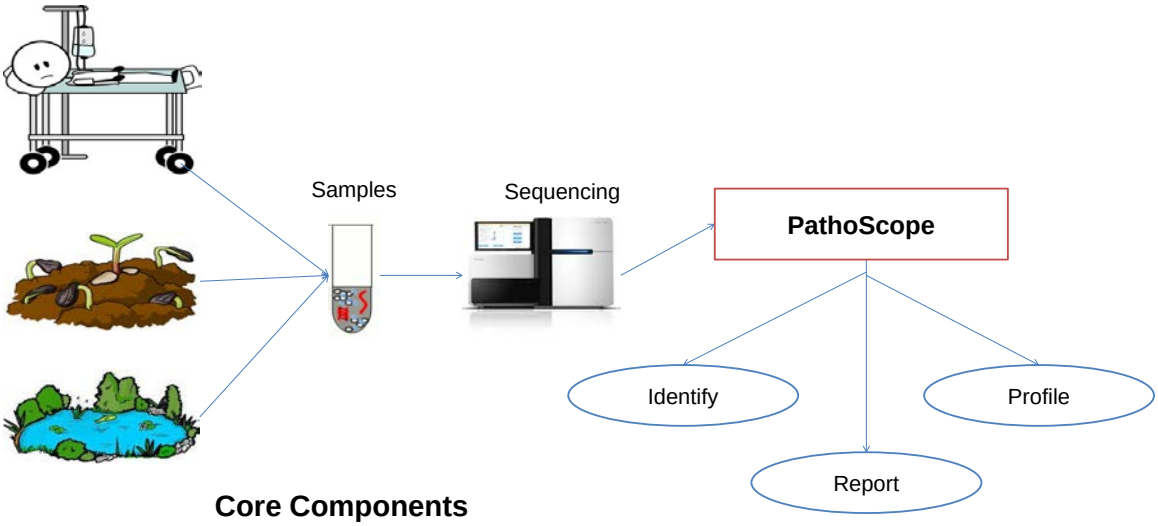
## *Experimental Plan*

Develop PathoScope, a complete software pipeline comprising of four core modules and with the option to add addition optional modules into a software toolkit for metagenomics analysis of sequencing samples. The four core PathoScope modules are the following 1) PathoLib: automatic reference library curation and preparation; 2) PathoMap: alignment of reads to target reference libraries and filtration of host or commensal species; 3) PathoID: accurate identification of species present in the sample and quantification of the relative abundances of each species; and 4) PathoReport: detailed, annotated results in a user-friendly report and output format. These approaches are described in detail below.

## *Methods*

We have developed a complete software pipeline with different plugin modules for easy extensibility and maintainability. Along with the four core modules namely PathoLib, PathoMap, PathoID and PathoReport, there are two optional modules namely PathoDB (Database to store detailed information about the reference sequences) and PathoQC (Quality control of input read sequences) that can be plugged in to add more functionality to this pipeline.

PathoScope 2.0 provides a complete modular bioinformatics workflow to analyze metagenomic sequence data from clinical or environmental samples as shown in Figure 4 below.



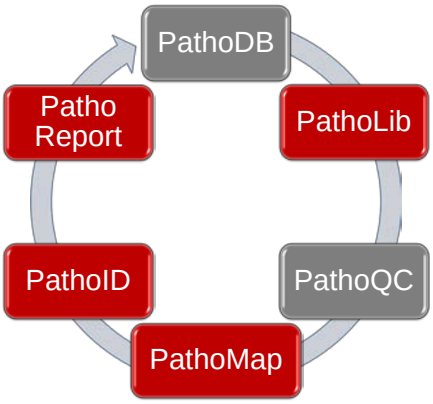

**Core Components**

**PathoLib:**
Automatic genome library extraction

**PathoMap:**
Alignment and Filtering of reads

**PathoID:**
Identify strains or substrains and estimate proportions

**PathoReport:**
Detailed XML report

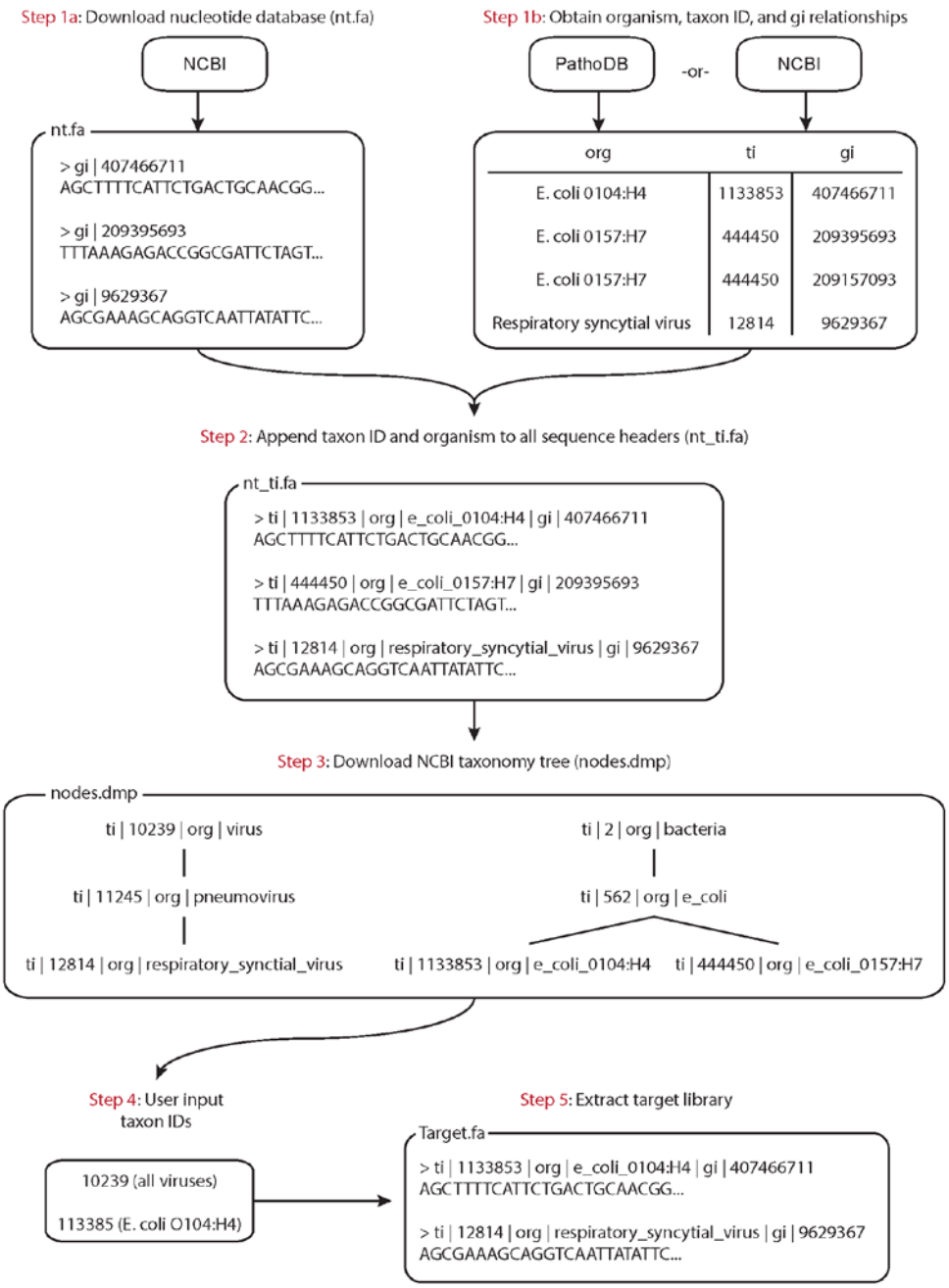



The PathoScope workflow start with collection of samples from patients or soil or some environmental conditions for metagenomics analysis. The sample is processed through a sequencing platform for DNA/RNA sequencing of data and typically a fastq/fasta file is obtained as an output from the sequencing machine. The data read file (fasta/fastq) is analyzed for quality of reads using PathoQC (optional module). PathoLib is used for building reference library such as virus/bacteria. The reads are mapped to the target reference genomes and filtering of read mappings to filter reference genomes (e.g. host genomes) using PathoMap. The alignment file from PathoMap is reassigned to the correct genomes of origin using PathoID and reports are generated using PathoReport with information from PathoDB.

**Figure 4: PathoScope workflow**

*PathoLib: Automatic reference library extraction*

PathoLib is a module for generating custom reference genome libraries. The careful selection of a refined reference sequence library is crucial for all downstream analyses. The PathoLib module allows the user to automatically generate custom reference genome libraries for specific scenarios or datasets. The user supplies a set of NCBI taxonomy identification (taxID) numbers for organisms to be included in the library (Figure 5). The user can construct both a 'target library' (that is, pathogen genomes of interest) and a 'filter library' (for example, host genome or benign flora) for later use in the PathoMap module. The PathoLib module will extract all sequences in the NCBI nucleotide database associated with the taxIDs (for example, complete genomes, transcripts, plasmids, partially assembled fragments, and so on). In addition, if a high-level taxID is given (for example, kingdom, family, genus), PathoLib can also optionally extract all lower level sequences in the NCBI taxonomy tree. As PathoLib extracts the reference library, the NCBI GeneInfo number is linked to the taxID, and the taxID and organism name are appended to the sequence headers to further link sequences in downstream analyses.

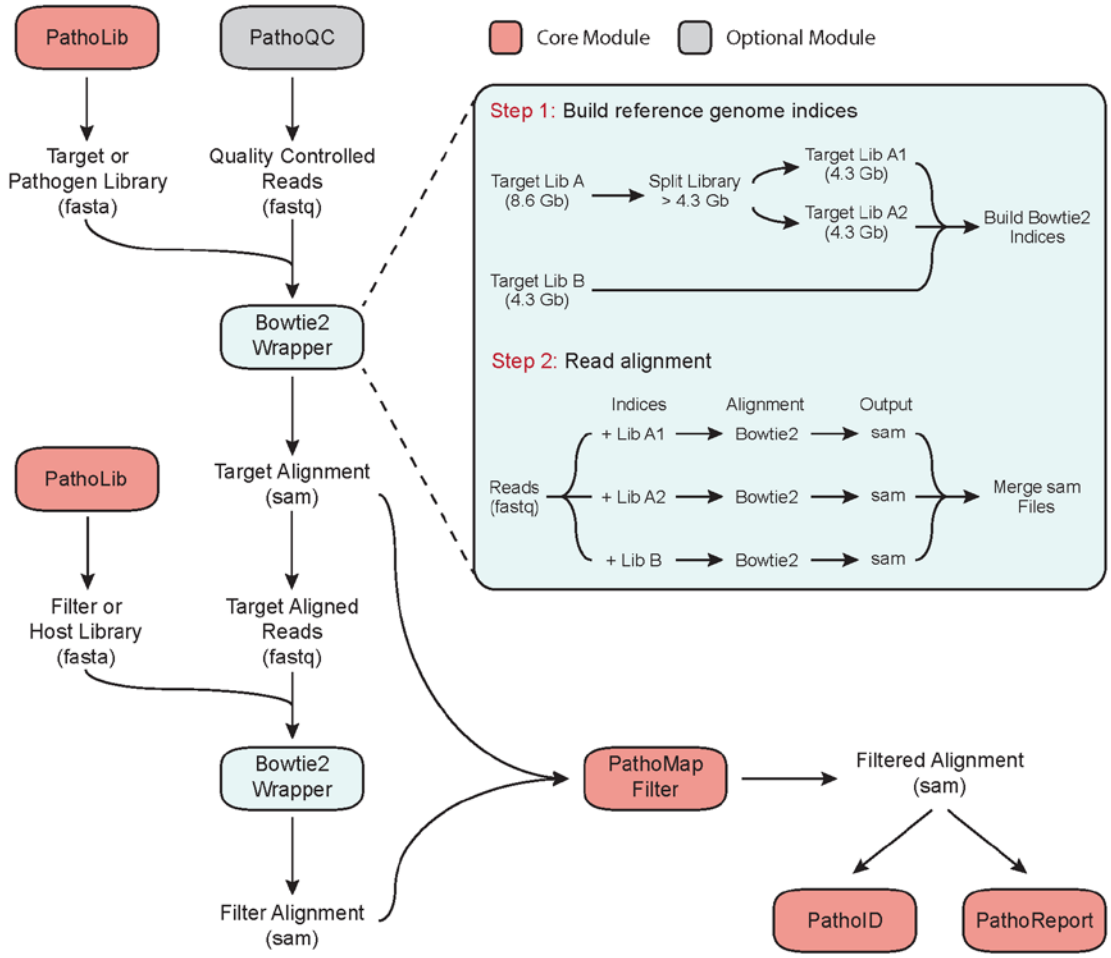



The PathoLib module will extract a reference library containing all genomes, chromosomes, transcripts, and other sequence fragments in the NCBI redundant nucleotide database associated user-defined taxonomic clade (NCBI taxID). If a higher-level taxID is given, PathoLib will optionally extract all sequences from lower-lever taxonomic designations based on the NCBI taxonomy tree.

**Figure 5 PathoLib module workflow**

*PathoMap: Efficient read alignment and filtering*

PathoMap is a module for aligning reads to the target reference library and filter out any reads that aligns better to filter reference library (Figure 6). Inputs for this module are the raw read file (FASTQ) and both the target and filter reference libraries (FASTA format). PathoMap will: (1) index the reference library (splitting the library into multiple indices if necessary); (2) align the reads to the target library; and (3) filter any of the target-matching reads that also match the filter library. The current version of PathoMap includes a Bowtie 2 (Langmead & Salzberg, 2012) wrapper (see Figure 6) with predetermined optimal alignment parameters for different read generation technologies (for example, Illumina: '–very-sensitive -k 100 –score-min L,-0.6,-0.6'; PacBio: '–very-sensitive -k 100 –score-min L, -0.6, -1.5'). The module also allows flexibility for the user to manually input Bowtie 2 parameters, or to conduct any part of the alignments outside the PathoMap framework by supplying an alignment file in SAM format (Li, Handsaker, Wysoker et al., 2009). Finally, the module is constructed in a way that wrappers for additional alignment algorithms can easily be substituted for the Bowtie 2 wrapper to accommodate diverse user preferences.

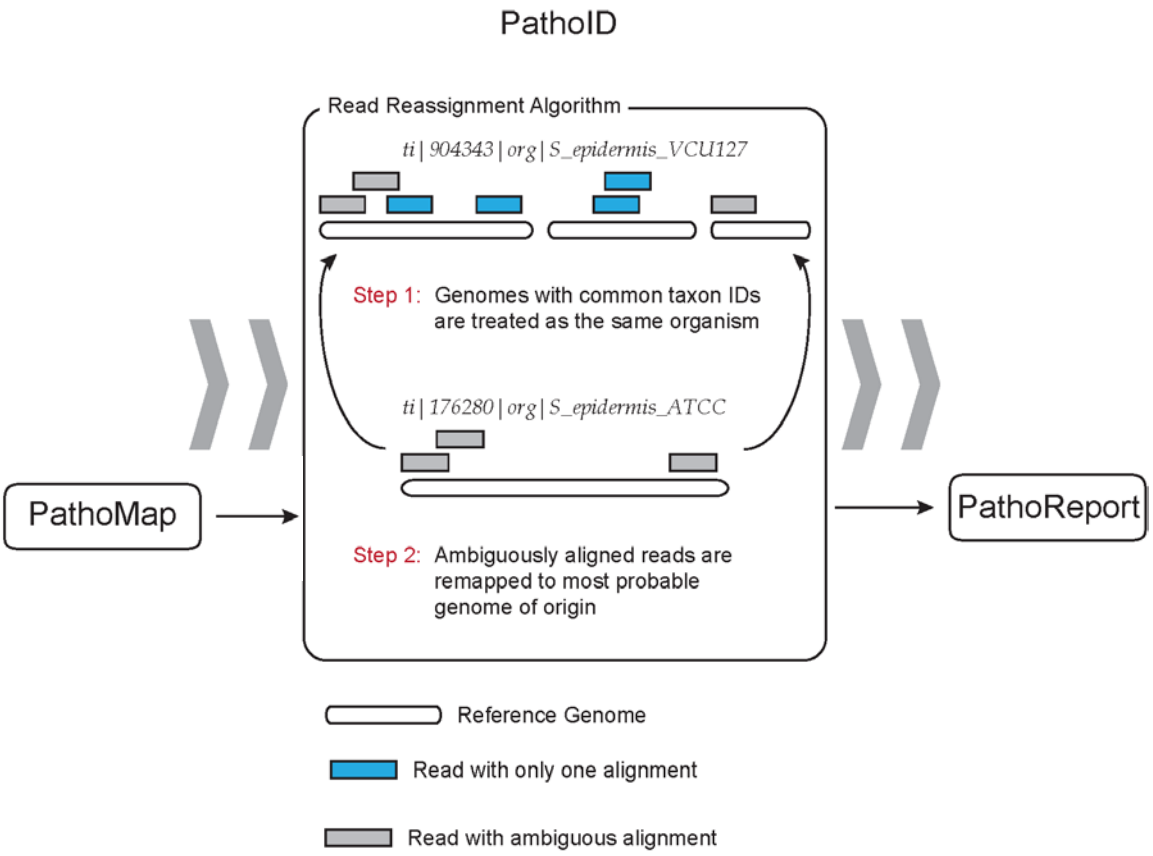



The PathoMap module aligns reads to the target library and removes any sequences that align to the filter library. PathoMap will: (1) index the reference library; (2) align the reads to the target library; and (3) filter any of the target-matching reads that also match the filter library. The current version of PathoMap includes a Bowtie 2 wrapper and allows users to conduct any part of the alignments outside the PathoMap framework.

**Figure 6 PathoMap module workflow**

*PathoID*

PathoID is a module for reassigning all ambiguous reads to the most likely source genome in the reference library based on the Bayesian Pseudo Likelihood mixture model. This model is described in detail in Aim 1A. This module will take either a SAM or BLAST alignment file as input and outputs a TSV (Tab Separated Value) with the genomes ranked according to the final guess of the proportions estimated after running an EM (Expectation Maximization) to maximize the likelihood calculated based on the Bayesian Pseudo Likelihood mixture model. The mixture model will include the alignment scores, reference genome length and user-defined priors for read proportions and ambiguity penalties and weights the likelihood accordingly to increase the number of reads that are correctly assigned to the source genome. This module will also optionally generate an updated SAM file with the updated alignment MapQ scores based on the final reassignment score this module computes for each read alignment.

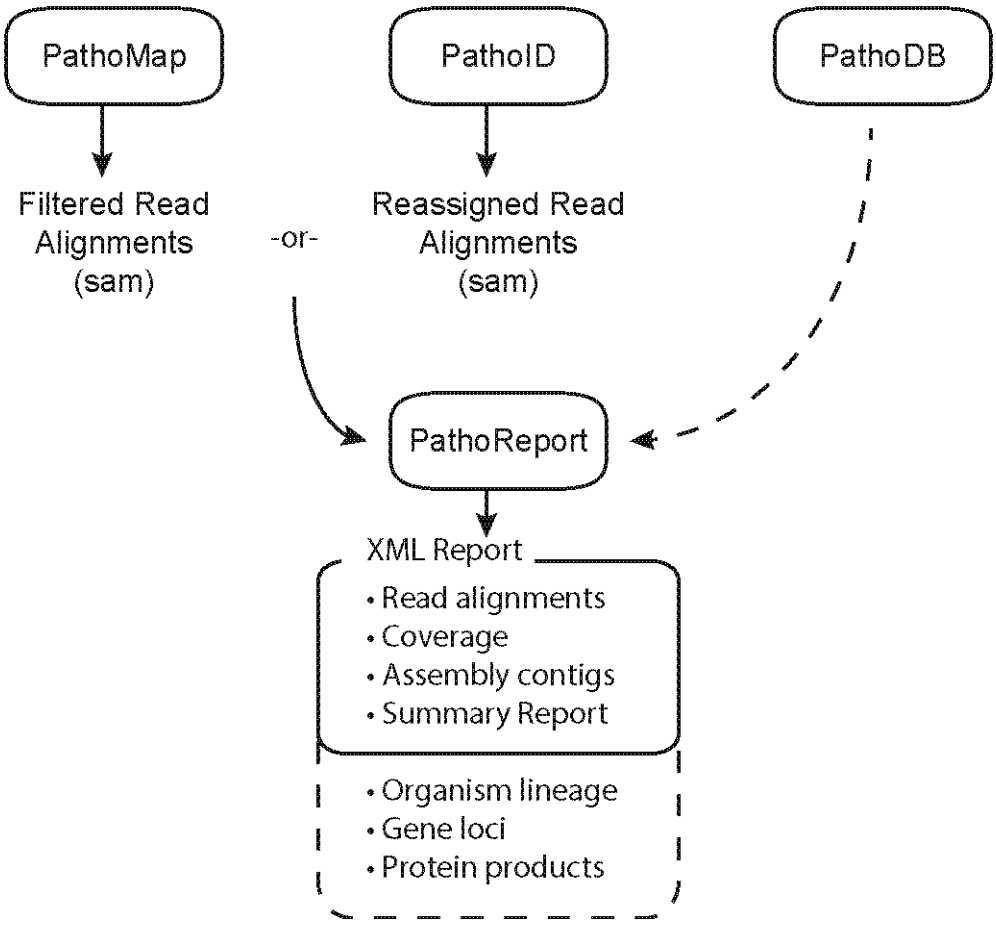



This module will take either a SAM or BLAST alignment file as input and outputs a TSV (Tab Separated Value) with the genomes ranked according to the final guess of the proportions estimated after running an EM (Expectation Maximization) to maximize the likelihood calculated based on the Bayesian Pseudo Likelihood mixture model. The mixture model will include the alignment scores, reference genome length and user-defined priors for read proportions and ambiguity penalties and weights the likelihood accordingly to increase the number of reads that are correctly assigned to the source genome. This module will also optionally generate an updated SAM file with the updated alignment MapQ scores based on the final reassignment score this module computes for each read alignment.

**Figure 7 PathoID module workflow**

### *PathoReport*

PathoReport is a module for detailed result reporting and annotation. The PathoReport module (Figure 8) outputs two files from the pipeline. The first output file is a tab-delimited (.tsv) report that contains the genomes that were identified by the previous steps sorted by rank, along with high/low confidence read numbers and proportions assigned to each genome. The second file, in XML format, contains more detailed results, including the reads assigned to each genome and contiguous sequences (contigs) constructed from overlapping reads. In addition, in concert with the plug-in module PathoDB (described below), PathoReport will add additional annotation into the report such as organism lineage, gene loci, and protein products for genes covered by the reads. This XML output provides useful information for evaluating the quality of the results and facilitating downstream interpretation and analysis. For example, the specific reads assigned to a genome can be an important quality check for a metagenomic analysis to check if the reads are low complexity or contain multiple PCR duplicates. The contigs show the breadth of genomic coverage, can identify sequence variation from the reference, and facilitate scaffold-based genome assembly. The gene annotations identify the specific genes covered by the reads, can be used to annotate SNPs in specific genes, and (in RNA-seq studies) can identify which pathogenic genes are actively expressed.

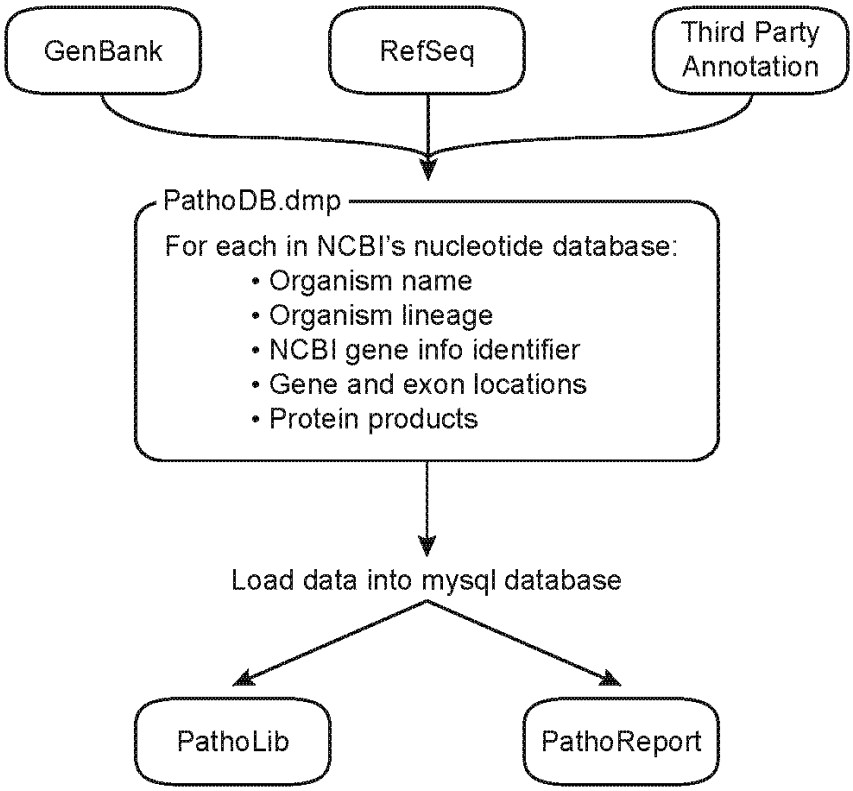



The PathoReport module outputs two report files including: (1) a tab-delimited (.tsv) report that contains a ranked list of genomes (with proportions) identified by the pipeline; and (2) an XML file containing detailed results including the reads assigned to each genome, contigs constructed from overlapping reads, and so on.

**Figure 8 PathoReport module workflow**

*PathoDB (optional module)*

PathoDB is an optional database module that contains taxonomy, gene, and protein product annotation for all sequences in the NCBI nucleotide database (Figure 9). The information in the PathoDB is used to generate detailed XML reports for further analysis.

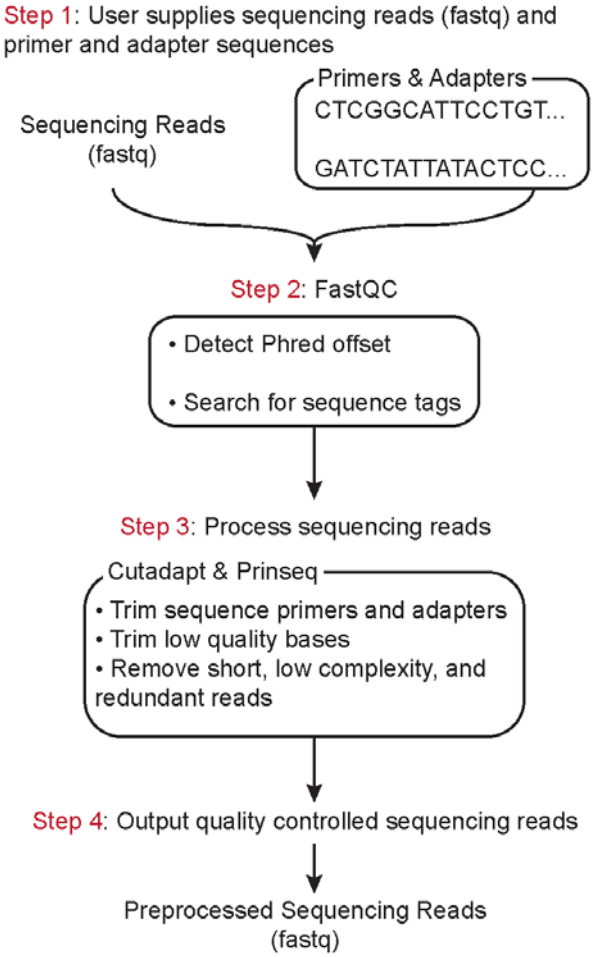



PathoDB is an optional module of pre-compiled annotation for all sequences in the NCBI nucleotide database. The PathoDB module automatically interacts with PathoReport to provide additional annotation in the detailed (XML) report such as organism lineage, gene loci, and protein products for any genes covered by the reads.

**Figure 9 PathoDB module workflow**

*PathoQC (optional module)*

PathoQC is an optional module for quality control of input read sequences. It performs several read quality control steps including trimming adapters, trimming low quality bases, and filtering low complexity reads (Figure 10). I helped with integrating PathoQC into the PathoScope framework and is described in detail in the Cancer Informatics publication (Hong, Manimaran & Johnson, 2014a).




PathoQC is an optional module for quality control of input read sequences. Shown are the steps performed by PathoQC.

**Figure 10 PathoQC module workflow**

The complete design of the PathoScope 2.0 framework is presented in the Appendix. The pipeline helps researchers to efficiently generate custom reference libraries, align reads to a target library, filter host reads, overcome read alignment ambiguity, characterize target diversity, and annotate results. Our simulated and real-data examples show that PathoScope 2.0 is a highly sensitive and efficient approach for metagenomic analysis, without the need for computationally intensive database preprocessing and time-consuming de novo assembly. PathoScope 2.0 is a fast and modularized pipeline for which we provide a comprehensive command line interaction so that more advanced users can selectively run parts of the modules, but is user-friendly enough to be used by researchers without strong computational backgrounds. I have led the design and development of the PathoScope 2.0 pipeline, which is described in detail in the Microbiome journal publication (Hong *et al.*, 2014b). PathoScope 2.0 is available as a python module for download from the sourceforge at https://sourceforge.net/projects/pathoscope/. I have contributed to the development of a clinical version of PathoScope called clinical PathoScope, which is described in detail in the BMC Bioinformatics publication (Byrd, Perez-Rogers, Manimaran et al., 2014) and is available for download from the same sourceforge URL given above.

### *PipelineBuild*

I have developed a python program called PipelineBuild to build custom pipeline for analysis using PathoScope modules. Using PipelineBuild, the user specifies a template of commands to run for each sample. An example template file is given below,

where the user wants to run PathoMap and PathoID and clean intermediate files. With PipelineBuild, it will automatically generate all the commands for all the samples found in a directory based on the template for one sample. This is very useful when a user wants to analyze many samples using the PathoScope pipeline.

Example Template:

```
pathoscope/pathoscope.py MAP -1 @@FILE1@@ -2 @@FILE2@@ -
targetIndexPrefixes bacteria_ti_0,bacteria_ti_1 -filterIndexPrefixes
human1_bowtie2,human2_bowtie2 -indexDir indexdir -outDir outdir -outAlign
@@FILE1MATCH@@.sam -expTag @@FILE1MATCH@@ -targetAlignParams "--
very-sensitive-local -k 10 --score-min L,20,1.0"
rm -f outdir/@@FILE1MATCH@@-*.*
pathoscope.py --verbose ID --noUpdatedAlignFile --noDisplayCutoff -alignFile
outdir/@@FILE1MATCH@@.sam -outDir outdir -expTag @@FILE1MATCH@@
```

The PipelineBuild program is available at the following URL: https://github.com/mani2012/PipelineBuild.

### *SplitQsub*

I have also developed a python program called SplitQsub for creating scripts for parallel execution in a server cluster environment. Using SplitQsub the user specifies a template for the generation of a qsub file (http://www.bu.edu/tech/support/research/system-usage/running-jobs/submitting-jobs/)

to be submitted to a server cluster for execution. An example qsub header file is given below. Using SplitQsub, users can generate qsub files for a set of commands, usually generated in combination with the PipelineBuild program mentioned above. Once these qsub files are generated, they can be submitted to the server cluster for execution. Usually multiple qsub files are submitted together for execution of multiple commands in parallel in a server cluster environment.

Example Template:

```
#!/bin/bash -l
#
#$ -cwd
#$ -N @@QNAME@@
#$ -o @@QNAME@@Log
#$ -j y
#$ -m be
#$ -M <put your email address here>
#$ -P pathoscope
#$ -pe single_node 8-8
### -l h=scc-cb4
### -l h_rt=24:00:00
echo "============================================================"
echo "Starting on       : $(date)"
echo "Running on node   : $(hostname)"
echo "Current job ID    : $JOB_ID"
echo "Current job name  : $JOB_NAME"
echo "Task index number : $TASK_ID"
echo "============================================================"
module load python/2.7.5
```

The SplitQsub program is available at the following URL: https://github.com/mani2012/SplitQsub.

## Conclusion

We developed a statistical framework for metagenomic analysis of next-generation sequencing data using a Bayesian mixture model with a modified pseudo likelihood model. Based on this model, we developed a complete software pipeline called PathoScope to identify microbes and pathogens to the strain level. We performed simulation studies to determine how accurately microbial proportions can be estimated when there is a mixture of multiple microbes with varying proportions in the sample. We evaluated the accuracy of the Bayesian mixture modeling approach in comparison to other methods and also evaluated how prior information in the Bayesian mixture modeling improves these estimates. We performed a simulation study to evaluate the read coverage needed to estimate pathogen proportions to a given confidence limit. Based on this study, we recommend that for single strain identification, about 0.1X coverage of reads is sufficient to get more than 99% accuracy using PathoScope. PathoScope 2.0 is available as a python module for download from the sourceforge at https://sourceforge.net/projects/pathoscope/.

# CHAPTER THREE

## Project 2: A Toolkit for Microbiome Variation Analysis

### Introduction

The microbiome can vary significantly between different biological conditions such as disease status or treatment differences or other covariates of interest within an organism or environment. Studying the variation of microbiomes under different conditions within an organism or environment is the key to diagnosing diseases and providing personalized treatments. For this project, the goals are the following: A) Identify various statistical measures such as alpha and beta diversity for characterizing the microbiome variation under different conditions and develop a module for visualization of the statistical measures; B) Develop a module to calculate and display the confidence regions for the relative abundance estimates; C) Perform Multi-dimensional and differential expression analysis of microbiomes under various conditions of interest; D) Develop a software pipeline called PathoStat for Microbiome variation analysis.

### Aim 2A

Summarize various statistical measures such as alpha and beta diversity for characterizing the microbiome variation under different conditions and develop a pipeline module, PathoStat, for visualization of the statistical measures.

### *Objective*

Develop a software pipeline to visualize the difference between samples in terms of various statistical diversity measures, particularly when samples contain microbes of varying proportions.

### *Rationale*

When samples contain microbes of varying proportions, it is difficult to characterize the microbial variations between samples. The first step in this analysis is to visualize the variations and compare the statistical diversity measures to numerically characterize the variations.

### *Experimental Plan*

Identify multiple statistical measures such as alpha and beta diversity and develop a software pipeline module, PathoStat, to visually display the microbiome variations in the multidimensional space by projecting along user selected dimensions and including the statistical diversity measures for comparison between samples.

### *Taxonomy Levels*

When analyzing microbiome variations, it is useful to group together similar organisms and study its relative abundance and the variations of it as a group with different biological conditions. Biologically, organisms are grouped based on shared characteristics in a taxonomic hierarchy (Nomenclature., Ride, Nomenclature. et al., 1999). The taxonomic ranks that we have used are species, genus, family, order, class, phylum and kingdom.

For example, let us consider a particular species, the red fox Vulpes vulpes. Its genus is *Vulpes*, which comprises of all the 'true foxes'. The next higher rank is the family *Canidae,* which includes their closest relatives, dogs, wolves, jackals, all foxes, and other caniforms such as bears, badgers and seals; the next level, the order *Carnivora*, includes feliforms and caniforms (lions, tigers, hyenas, wolverines, etc.), plus other carnivorous mammals. This order is one group of the class *Mammalia*, all animals with backbones are classified in the *Chordata* phylum rank, which can be found among all other animals in the *Animalia* kingdom rank (Nomenclature. *et al.*, 1999).

### *Diversity Measures*

The terms ‘alpha’, ‘beta’ and ‘gamma’ diversity were introduced originally by R. H. Whittaker (Whittaker, 1972). Alpha diversity refers to diversity within a local site or as a diversity measure of microbiome within a sample or samples from the same covariate conditions of interest. Beta diversity refers to diversity across multiple sites or the diversity measure across multiple samples from different covariate conditions. Gamma diversity corresponds to the total diversity within and across multiple sites and in our case refers to the total diversity both within and across samples of different covariate conditions. Furthermore, researchers have commonly used principal components analysis (PCA) (Yeung & Ruzzo, 2001) or principal coordinates analysis (PCoA) to analyze and visualize microbiome data. Our software pipeline called PathoStat integrates all these types of measures and exploratory analyses for microbiome variation analysis in one place. There is another software called QIIME (Navas-Molina, Peralta-Sanchez,

Gonzalez et al., 2013; Kuczynski, Stombaugh, Walters et al., 2012) that does some of these analysis for 16SrDNA (Woo, Lau, Teng et al.) data, but our software PathoStat can perform the analysis on 16SrDNA data as well as on metagenomic (Thomas, Gilbert & Meyer, 2012) data and it has more functionalities such as different types of multi-dimensional analysis.

### *Alpha diversity*

PathoStat has a module to display the alpha diversity based on Shannon's diversity index, Simpson index and Inverse Simpson index. These measures are discussed in the following paragraphs below.

#### *Shannon's diversity Index:*

Shannon's diversity index is popular in the ecological literature (Morris, Caruso, Buscot et al., 2014). This measure was originally introduced by Claude Shannon to quantify the entropy in strings of text (Shannon, 1948). It quantifies the uncertainty in predicting a species when randomly choosing a species from a dataset with multiple species. If $p_i$ represents the proportion of species $i$ in the dataset of interest, then the Shannon index is computed as follows.

$$H' = -\sum_{i=1}^{R} p_i \ln p_i = -\sum_{i=1}^{R} \ln p_i^{p_i} = \ln\left(\frac{1}{\prod_{i=1}^{R} p_i^{p_i}}\right)$$

Since $p_i$ represents the proportion of species in a dataset, the sum of the $p_i$ values equals unity by definition and hence the denominator in the above equation equals the weighted geometric mean of the $p_i$ values, with the $p_i$ values themselves being used as the weights (exponents in the equation). Shannon entropy is a standard measure of entropy / heterogeneity that works well with discrete data. This index ranges from 0 (when there is only one species) to infinity when there are a lot of species with high alpha diversity.

### *Simpson Index:*

Edward H. Simpson introduced Simpson index first in 1949 when measuring the degree of concentration when individuals are classified into types (Simpson, 1949). The measure is equal to the probability that two entities taken at random from a dataset represents the same type. It is given by the following equation:

$$\lambda = \sum_{i=1}^{R} p_i^2$$

This is also equal to the weighted arithmetic mean of the proportions $p_i$, with the proportional values $p_i$ themselves being used as the weights. This index ranges from 0 when there are a lot of species with high alpha diversity to 1 (when there is only one species). There is also a variation of this index called Gini–Simpson index (Jost, 2006), which is equal to $1 - \lambda$. The Gini–Simpson index equals the probability that the two entities taken at random from a dataset represent different types.

### *Inverse Simpson Index:*

This index is simply the inverse of the Simpson index (Zhou, Xia, Treves et al., 2002) given by the following:

$$1/\lambda = \frac{1}{\sum_{i=1}^{R} p_i^2}$$

This is equal to the average number of entities when the weighted arithmetic mean is used to quantify the average proportional abundance of entities in a dataset. This index ranges from 0 (when there is only one species) to infinity when there are a lot of species with high alpha diversity.

### *Beta Diversity*

PathoStat has a module to display the beta diversity across samples based on Bray–Curtis dissimilarity and Weighted Unifrac measures.

### *Bray–Curtis dissimilarity:*

J. Roger Bray and John T. Curtis introduced the Bray–Curtis dissimilarity statistic (Bray & Curtis, 1957), which is used to quantify the compositional dissimilarity between two different sites, based on counts at each site. Suppose if there are N number of sites and in our example datasets below, N number of samples, the statistic measure between two different sites/samples is given by the following equation:

$$BC_{ij} = 1 - \frac{2C_{ij}}{S_i + S_j}; i, j = 1 to\ N$$

where $S_i$ and $S_j$ are the total number of specimens counted at both sites. $C_{ij}$ is the sum of the lesser values of the counts between the two sites for only those species in common between both sites. Suppose there are K species that are common between sites/samples i

and j, and the number of each of those species in site/sample i is $A_{ik}$ and number of those species in site/sample j is $B_{jk}$.

$$C_{ij} = \sum_{k=1}^{K} \min(A_{ik}, B_{jk})$$

The measure ranges from 0 (when the sites/samples are same with respect to the number of species and its counts) to 1 (when they are very diverse with nothing shared).

### *Weighted Unifrac:*

Unifrac measure differs from Bray-Curtis dissimilarity in that it incorporates information on the relative relatedness (shared phylogenetic tree) of community members by incorporating phylogenetic distances between observed organisms in the computation (Lozupone, Hamady, Kelley et al., 2007). Initially, a phylogenetic tree is constructed by placing all taxa found in one or both samples. A branch leading to taxa from both samples is marked as "shared" and branches leading to taxa which appears only in one sample are marked as "unshared". The Unifrac distance between the two samples is equal to the following:

$$\frac{the\ sum\ of\ "unshared"\ branch\ lengths}{the\ sum\ of\ all\ tree\ branch\ lengths\ (=\ shared + unshared)}$$

In weighted Unifrac measure, the branch lengths are weighted by the relative abundance of the lineages in the samples. This measure also ranges from 0 (nothing shared) to 1 (everything shared).

### *Example datasets*

For our analysis, we have used two datasets: 1) Diet Study dataset and 2) Asthma study dataset.

### *Diet Study dataset*

This is a dataset from a study performed in collaboration with Dr. Lichtenstein's lab in Tufts University. In this diet study, there were 11 subjects with each person taking three different types of diets in random order. The three different types of diets were simple sugars, refined carbohydrates and unrefined carbohydrates. Each person took each of the diets in random order with one diet in the time period 0-5 weeks, and another diet in 8-13 weeks and another diet in 16-21 weeks. Samples from fecal matter were collected from each of the subjects for analysis at the end of each type of diets, which are at the end of 5 weeks, 13 weeks and 21 weeks. The sample characteristics are shown in Table 6 below. We performed metagenomics analysis on 16SrDNA (Woo *et al.*) and RNA-Seq data (Conesa, Madrigal, Tarazona et al., 2016) on these samples to characterize the microbes in each of the samples and the variations of the abundance of the microbes among different types of samples.

| | | Sex | Age | TC | LDL-C | HDL-C | VLDL-C | TG |
|---|---|---|---|---|---|---|---|---|
| **Subject ID** | **Diet** | | **years** | **mg/dL** | **mg/dL** | **mg/dL** | **mg/dL** | **mg/dL** |
| 1 | Simple | F | 71 | 233.0 | 146.0 | 48.3 | 38.7 | 193.3 |
| 2 | Simple | F | 78 | 206.3 | 124.7 | 48.7 | 33.0 | 167.7 |
| 3 | Simple | F | 64 | 220.3 | 133.7 | 76.0 | 10.7 | 55.0 |
| 7 | Simple | M | 60 | 158.3 | 89.3 | 36.0 | 33.0 | 166.0 |
| 6 | Simple | F | 67 | 258.0 | 170.3 | 56.3 | 31.3 | 159.3 |
| 8 | Simple | M | 57 | 160.0 | 93.0 | 32.3 | 34.7 | 174.0 |
| 9 | Simple | F | 77 | 191.0 | 114.0 | 48.3 | 28.7 | 146.0 |
| 10 | Simple | M | 57 | 184.7 | 113.0 | 45.7 | 26.0 | 132.0 |
| 12 | Simple | M | 58 | 286.7 | 174.3 | 58.3 | 54.0 | 272.0 |
| 13 | Simple | F | 62 | 183.3 | 110.3 | 45.7 | 27.3 | 139.0 |
| 15 | Simple | F | 64 | 235.0 | 149.0 | 64.0 | 22.0 | 113.5 |
| 1 | Refined | F | 71 | 237.0 | 151.3 | 51.3 | 34.3 | 174.7 |
| 2 | Refined | F | 78 | 203.3 | 126.0 | 48.3 | 29.0 | 147.7 |
| 3 | Refined | F | 64 | 204.7 | 125.7 | 65.7 | 13.3 | 67.7 |
| 7 | Refined | M | 60 | 192.3 | 107.7 | 35.0 | 49.7 | 250.3 |
| 6 | Refined | F | 67 | 272.3 | 176.3 | 60.7 | 35.3 | 179.3 |
| 8 | Refined | M | 57 | 198.0 | 127.0 | 38.0 | 33.0 | 165.0 |
| 9 | Refined | F | 77 | 185.3 | 105.0 | 45.3 | 35.0 | 176.7 |
| 10 | Refined | M | 57 | 189.3 | 119.3 | 45.0 | 30.3 | 154.0 |
| 12 | Refined | M | 58 | 275.3 | 172.7 | 48.3 | 54.3 | 273.3 |
| 13 | Refined | F | 62 | 221.3 | 139.0 | 58.7 | 23.7 | 121.3 |
| 15 | Refined | F | 64 | 227.7 | 148.3 | 59.3 | 20.0 | 101.3 |
| 1 | Unrefined | F | 71 | 248.3 | 159.7 | 49.7 | 39.0 | 196.3 |
| 2 | Unrefined | F | 78 | 184.3 | 107.0 | 48.0 | 29.3 | 149.3 |
| 3 | Unrefined | F | 64 | 204.7 | 126.0 | 67.0 | 11.7 | 60.3 |
| 7 | Unrefined | M | 60 | 167.3 | 86.7 | 32.7 | 48.0 | 241.0 |
| 6 | Unrefined | F | 67 | 266.7 | 170.3 | 63.7 | 32.7 | 166.0 |
| 8 | Unrefined | M | 57 | 174.3 | 117.3 | 31.0 | 26.0 | 133.0 |
| 9 | Unrefined | F | 77 | 178.0 | 105.3 | 43.7 | 29.0 | 147.7 |
| 10 | Unrefined | M | 57 | 200.7 | 119.0 | 48.0 | 33.7 | 169.3 |
| 12 | Unrefined | M | 58 | 251.7 | 166.0 | 54.7 | 31.0 | 156.3 |
| 13 | Unrefined | F | 62 | 156.0 | 88.3 | 42.3 | 25.3 | 127.7 |
| 15 | Unrefined | F | 64 | 213.7 | 134.3 | 63.0 | 16.3 | 83.3 |

Shown here are the sample characteristics for the diet study dataset. TC: Total Cholesterol; LDL-C: LDL Cholesterol; HDL-C: HDL Cholesterol; VLDL-C: VLDL Cholesterol; TG: Triglycerides.

**Table 6: Diet Study Sample Characteristics**

### *Asthma Study dataset*

This is a dataset from a study titled "Integrating microbial and host transcriptomics to characterize asthma-associated microbial communities" (Castro-Nallar, Shen, Freishtat et al., 2015). Samples from brushed nasal epithelial cells of 14 children: 8 with asthma and 6 controls were obtained and RNA-Sequencing were performed on those samples. The RNA-Seq (Conesa *et al.*, 2016) data collected from them was used for performing microbiome analysis to characterize the microbial variations between normal children and children with asthma.

### *Visualization*

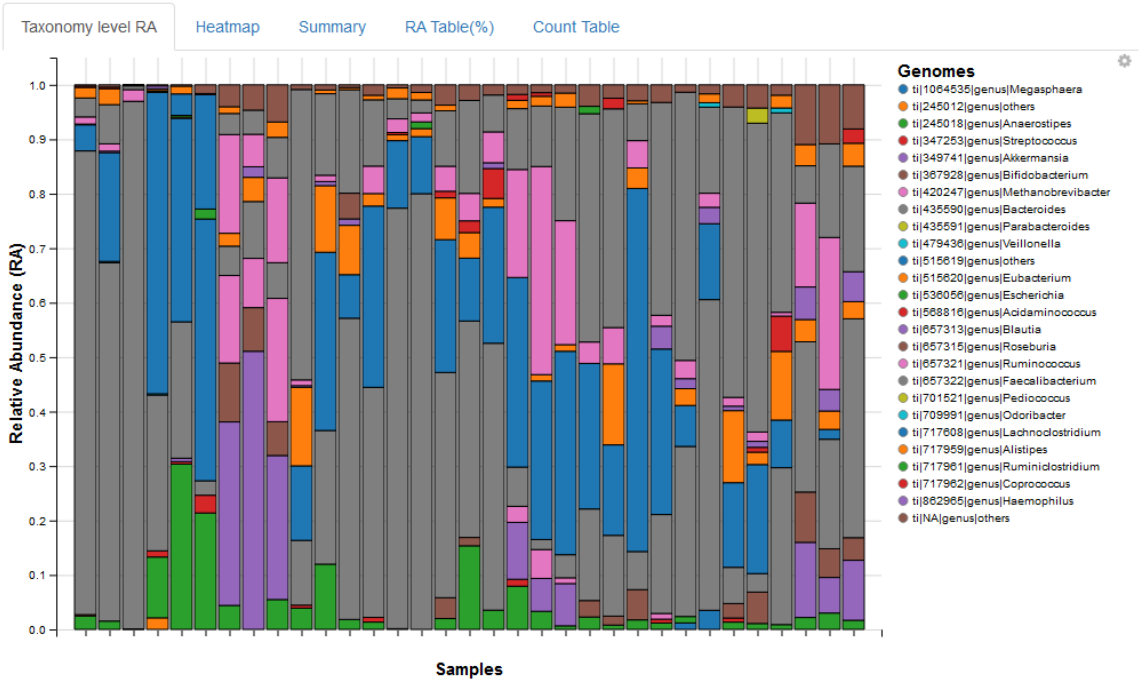
The first step in any statistical analysis starts with the summary measures and proper visualization of the measures. We have used various statistical measures such as alpha and beta diversity measures in the context of analyzing sequencing data to develop a pipeline for appropriate visualization to characterize the microbiome variation under different sources of variations such as disease status or treatment differences or other covariates of interest within an organism or environment.

### *PathoStat Shiny App R-Package*

We have developed an interactive Shiny app visualization module as part of the PathoStat package in the R statistical programming language. The purpose of this package is to perform statistical microbiome analysis on metagenomics results from sequencing data samples. In particular, it supports analyses on the PathoScope generated report files. PathoStat provides various functionalities including relative abundance

charts, diversity estimates and plots, tests of differential abundance, multi-dimensional analysis including principal component and principal coordinate analysis, time Series visualization, and core OTU (Operational Taxonomic Unit) (Blaxter, Mann, Chapman et al., 2005) analysis. The important feature of the package is the interactive feature of all the plots, allowing the user to choose various parameters and variables of interest and visualize all the dynamically generated plots customized according to the user selected criteria.

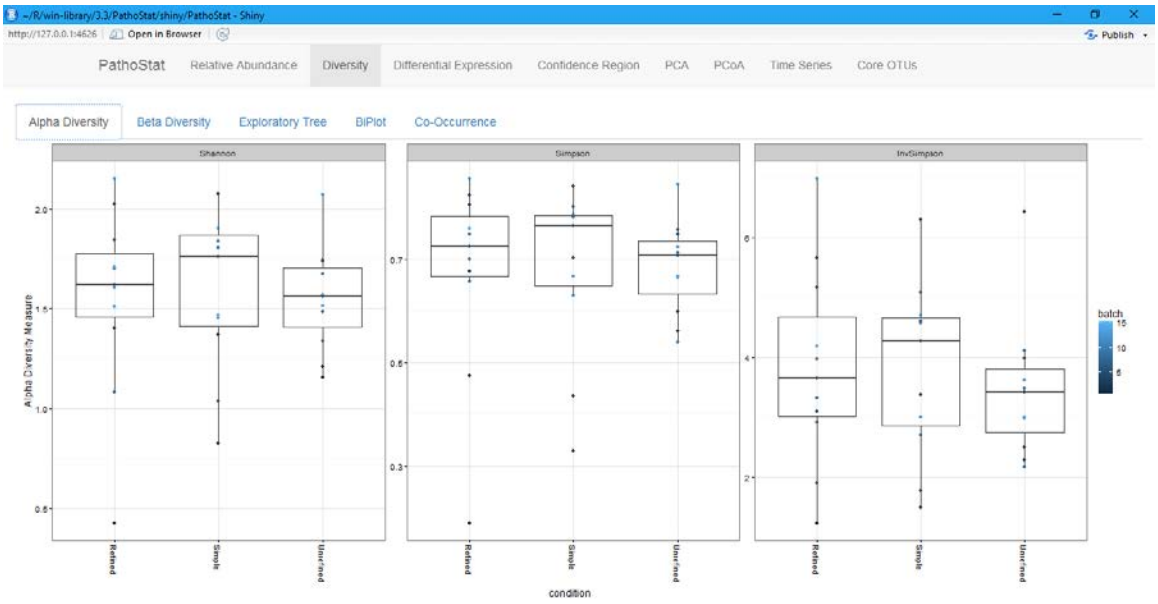
The Figure 11 below, shows the PathoStat relative abundance plot at the genus level for the samples from the diet study example 16SrDNA (Conesa *et al.*, 2016) dataset that is included as part of the PathoStat R package.




Shown here is the relative abundance plot of the microbes at the genus level for the 33 samples from the diet study example 16SrDNA dataset that is included as part of the PathoSat R package. The color and size of the bar in this plot is used to identify the genus and its relative abundance.

**Figure 11: PathoStat Relative Abundance plot at the genus level from the diet study example 16SrDNA dataset in PathoStat R package**

The Figure 12 below shows the PathoStat alpha diversity plots based on the three alpha diversity measures namely Shannon's diversity index, Simpson index and Inverse Simpson index for the example dataset that is included as part of the PathoStat package. The Figure 13 below shows the PathoStat beta diversity plots using the default Bray–Curtis dissimilarity measure.




Shown here is the alpha diversity plot based on the Shannon, Simpson and Inverse Simpson measures for the microbial relative abundance at the genus level for the 33 samples grouped by the types of diets (Refined, Simple and Unrefined) from the diet study example 16SrDNA dataset that is included as part of the PathoSat R package. For this example, we do not see any significant difference in alpha diversity between the three types of diets on all the three alpha diversity measures - Shannon, Simpson and Inverse Simpson measures.

**Figure 12: PathoStat Alpha Diversity plot for the diet study example 16SrDNA dataset that is included as part of the PathoSat R package**


Shown here is the beta diversity plot based on the default Bray–Curtis dissimilarity measure between each pair of the 33 samples from the diet study example 16SrDNA dataset that is included as part of the PathoSat R package. For this example, we that the samples are clustered by subjects indicating that the difference in microbes between subjects is much more than the difference between diets.

**Figure 13: PathoStat Beta Diversity plot for the diet study example 16SrDNA dataset that is included as part of the PathoSat R package**

## Aim 2B

Develop a module to calculate and display the confidence regions for the microbial relative abundance estimates and incorporate this module as part of the PathoStat toolkit for microbiome variation analysis.

### *Objective*

Develop a software pipeline to visualize the confidence regions of the microbiome variations along the dimensions selected by the user. By default it will show the 95% confidence region for the top two microbiomes that are present in the selected sample.

### *Rationale*

When analyzing samples for microbial content using a metagenomics approach, it is often desirable to accurately characterize the relative abundance of pathogens and to determine the confidence level for these estimates. This will aid in both research and clinical contexts to develop more appropriate targeted therapies. Hence, a pipeline to display the confidence regions for the microbiome variation estimates will be helpful for physicians and researchers in developing personalized and targeted treatment plans based on this analysis.

### *Experimental Plan*

Add to PathoStat pipeline a module for the display of confidence regions for the microbial relative abundance estimates for the microbes selected by the user. There will be an option to choose the microbes for each of the X and Y axis, for the display of confidence region along those components. Evaluate this module using real and simulated data examples.

*Confidence Region Calculation*

We use the large sample multivariate distribution of the Maximum Likelihood Estimator (MLE) for the relative abundance estimates in the calculation of the confidence regions for the microbial proportions of the microbes chosen by the user.

Let $X = (X_1, X_2, \dots, X_g)$ represent the counts for each of the g microbes in a sample. The relative abundance MLE estimates of the sample is given by $\hat{p} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_g) = \frac{1}{\sum(X_1 + X_2 + \dots + X_g)}(X_1, X_2, \dots, X_g) = \left(\frac{X_1}{n}, \frac{X_2}{n}, \dots, \frac{X_g}{n}\right); n = \sum(X_1 + X_2 + \dots + X_n)$.

**Fisher Information:**

In order to find the asymptotic distribution of an estimator, we use the Fisher Information, which is defined as follows:

$$I(\theta) = -E[\frac{\partial^2}{\partial^2\theta}\log f(x,\theta)] = E[(\frac{\partial}{\partial\theta}\log f(x,\theta))^2] = V[\frac{\partial}{\partial\theta}\log f(x,\theta)]$$

For the MLE $\hat{p}$, we have the following asymptotic approximation:

$$\hat{p} \sim N\big(p, I_n(\hat{p})\big)\,; I_n(\hat{p}) = Fisher\ Information$$

For the multinomial distribution with 'g' number of events, Fisher Information is computed as follows:

$$f(x,p) \propto \prod_{i=1}^{g} p_i^{x_i}\,; \sum_{i=1}^{g} p_i = 1$$

$$\log\big(f(x,p)\big) = c + \sum_{i=1}^{g-1} x_i log(p_i) + x_g log\left(1 - \left(p_1 + p_2 + \cdots + p_{g-1}\right)\right)$$

$$\frac{\partial}{\partial p_i}\log f(x,p) = \frac{x_i}{p_i} - \frac{x_g}{p_g}$$

$$\frac{\partial^2}{\partial^2 p}\log f(x,p) = (-1)\begin{pmatrix} \frac{x_1}{p_1^2} + \frac{x_g}{p_g^2} & \frac{x_g}{p_g^2} & \cdots & \frac{x_g}{p_g^2} \\ \frac{x_g}{p_g^2} & \frac{x_2}{p_2^2} + \frac{x_g}{p_g^2} & \cdots & \frac{x_g}{p_g^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_g}{p_g^2} & \cdots & \cdots & \frac{x_{g-1}}{p_{g-1}^2} + \frac{x_g}{p_g^2} \end{pmatrix}$$

$$I\left(\underline{p}\right) = -E\left[\frac{\partial^2}{\partial^2 p}\log f(x,p)\right] = n\begin{pmatrix} \frac{1}{p_1} + \frac{1}{p_g} & \frac{1}{p_g} & \cdots & \frac{1}{p_g} \\ \frac{1}{p_g} & \frac{1}{p_2} + \frac{1}{p_g} & \cdots & \frac{1}{p_g} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{p_g} & \cdots & \cdots & \frac{1}{p_{g-1}} + \frac{1}{p_g} \end{pmatrix}$$

**Confidence Regions:**

For the MLE $\hat{p}$, we have the following:

$$\hat{p} \sim N\left(p, I_n(\hat{p})\right)$$

$$\left(I_n(\hat{p})\right)^{-1/2}(p - \hat{p}) \rightarrow \left(Z_1, Z_2, .. Z_{g-1}\right); Z_i iid \sim N(0,1); i = 1, \ldots, g-1$$

From this we get the following:

$$(\underline{p} - \underline{\hat{p}_n})' I_n(\underline{\hat{p}_n})(\underline{p} - \underline{\hat{p}_n}) \xrightarrow{d} \chi^2_{g-1}$$

We use this asymptotic approximation in our computation of the confidence region in terms of standard chi-square distribution with g-1 degrees of freedom. In the context of our problem, we use this to compute the confidence region for the relative abundance of the microbes in each of the samples.

Suppose we are interested in the confidence interval of some linear scalar functions of the parameters of the form $t'p$, and estimated by $t'\underline{\hat{p}}$. The confidence interval for linear scalar function of the estimator can be obtained in terms of the Fisher Information as follows:

$$t'\underline{\hat{p}} \pm Z\sqrt{t'\left(I_n\left(\underline{\hat{p}}_n\right)^{-1}\right)t}$$

**Confidence Regions of the logit transformation:**

When the number of counts of microbes in a sample is small and also when the microbial relative abundance is small, the logit transformation of the proportion gives a better

estimate of the confidence region. We can compute the confidence region for the logit transformation of the proportion as follows using a method called delta-method.

For an estimator $X_n$ of $\theta$ with $\sqrt{n}(X_n - \theta) \to N(0, \sigma^2)$, if we want to find an estimator for $g(\theta)$, we get the following using delta-method:

$$\sqrt{n}(g(X_n) - g(\theta)) \to N\left(0, \sigma^2\left(g'(\theta)\right)^2\right).$$

For a logit transformation g(p) = logit(p), we can compute $g'(p) = \frac{1}{p(1-p)}$

For multivariate case, we get the following:

$$\left(I_n(\hat{p})\right)^{1/2}(p - \hat{p}) \to \left(Z_1, Z_2, .. Z_{g-1}\right); Z_i iid \sim N(0,1); i = 1, \ldots, g-1$$

$$\left(I_n(\hat{p})\right)^{\frac{1}{2}}\left(logit(p) - logit(\hat{p})\right) \to g'\left(\underline{\hat{p}}\right)\left(Z_1, Z_2, .. Z_{g-1}\right);$$

$$Z_i iid \sim N(0,1); i = 1, \ldots, g-1$$

where $g'\left(\underline{\hat{p}}\right) = \begin{pmatrix} \frac{1}{p_1(1-p_1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{p_2(1-p_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{p_{g-1}(1-p_{g-1})} \end{pmatrix}$; $\sum_{i=1}^{g-1} p_i < 1$

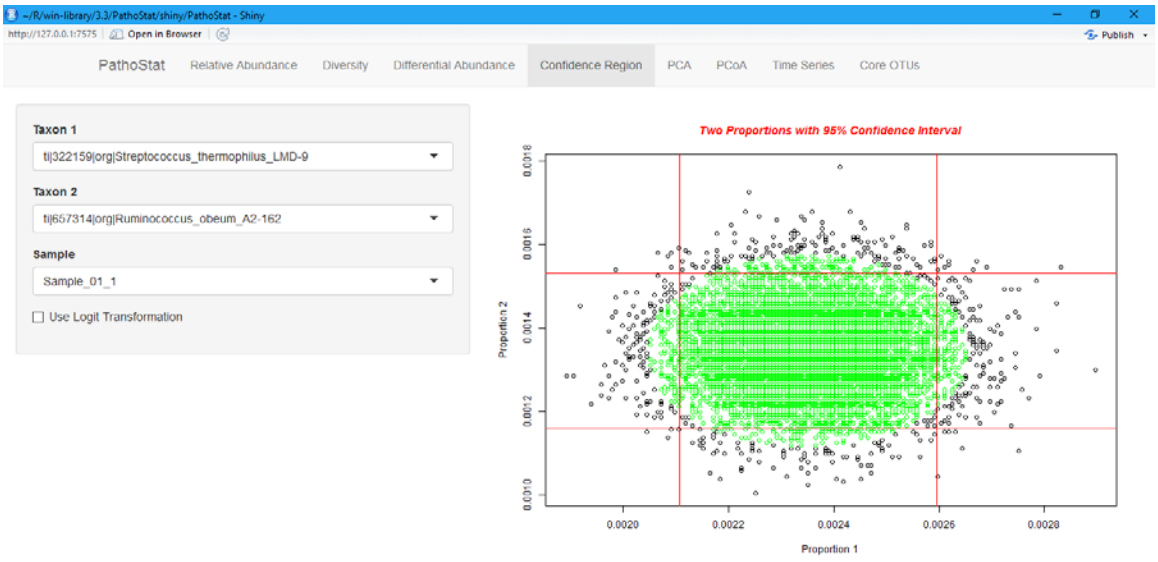
From this, we get the following asymptotic approximation for the confidence region of the logit transformation in terms of standard chi-square distribution with g-1 degrees of freedom.

$$(G'(\underline{\hat{p}}) * (logit(\underline{p}) - logit(\underline{\hat{p}})))' I_n(\underline{\hat{p}_n})(G'(\underline{\hat{p}}) * (logit(\underline{p}) - logit(\underline{\hat{p}}))) \xrightarrow{d} \chi^2_{g-1}$$

where $G'(\underline{\hat{p}}) = \begin{pmatrix} p_1(1-p_1) & 0 & \cdots & 0 \\ 0 & p_2(1-p_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{g-1}(1-p_{g-1}) \end{pmatrix}$; $\sum_{i=1}^{g-1} p_i < 1$

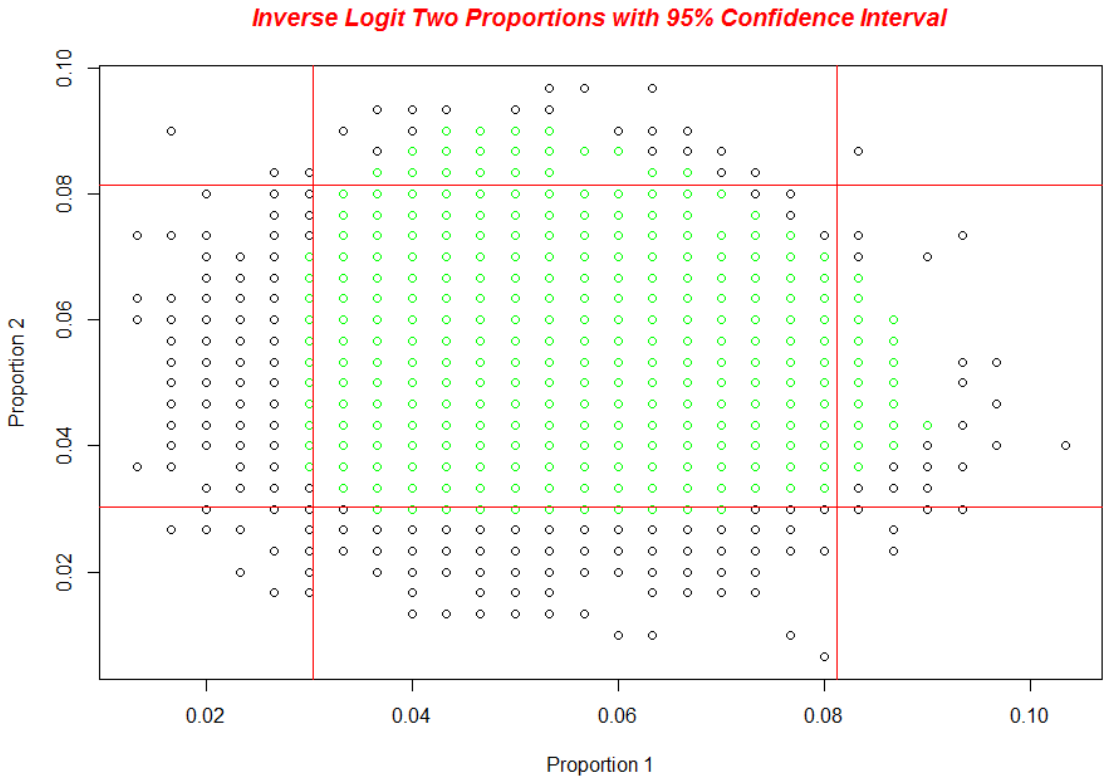
**Confidence Region Module:**

The confidence region module of the PathoStat software package will compute the confidence region of the microbial proportions using the methods described above. The multiple dimension confidence region is displayed by extracting the marginal distribution of two components at a time for displaying as a two dimensional graph with the user having the option to interactively choose the dimensions of interest. The boundary will be clearly marked for the user to infer the variability of the microbial composition based on the confidence regions that is displayed. This module is developed as a component of the interactive Shiny app for the PathoStat module. A screenshot of this PathoStat confidence region module is shown in Figure 14 and a confidence region plot with inverse logit transformation is shown in Figure 15 below.




Shown here is the confidence region for relative abundance estimates of two microbes of one sample from the diet study example 16SrDNA dataset that is included as part of the PathoSat R package.

**Figure 14: PathoStat Confidence Region module**




Shown here is the confidence region for relative abundance estimates of two microbes of one sample with a low count of 300 total reads and microbial proportion estimate of 5% for the two microbes, computed using logit transformation and inverting it back to get it on the original proportion scale.

**Figure 15: Low proportion 95% Confidence Region with Inverse Logit Transformation on a simulated data**

## Aim 2C

Perform differential abundance analysis of microbiomes under various conditions of interest and identify several multi-dimensional analysis that can be performed under the context of analyzing microbiome variations.

### *Objective*

Evaluate differential abundance analysis techniques using example datasets and identify different multi-dimensional analysis, which can be used to design modules that will be part of the PathoStat microbiome analysis toolkit and help with performing these type of analysis.

### *Rationale*

In order to study the microbiome variations along the conditions of interest, it is important to perform multi-dimensional analysis and identify the variables of interest that are associated with the microbiome variations. We can also perform differential abundance analysis to statistically characterize the variations in the relative abundance of microbes with respect to the conditions of interest after accounting for other covariates.

### *Experimental Plan*

Perform differential abundant analysis on the diet study example dataset with suitable model to capture the subject effect and the correlation structure induced by repeated measures on the subjects for multiple diets. Identify different types of multi-dimensional analysis including principal component and principal coordinate analysis and incorporate those techniques into our PathoStat microbiome analysis toolkit.

### *Differential Abundance Analysis*

We performed differential abundance analysis on the results of the metagenomics analysis on 16SrDNA and RNA-Seq data set from the diet study example mentioned

above. The results were consistent and there was not any statistically significant microbe that had a relative abundance which is differentially expressed across the three diets after accounting for multiple testing at the 0.05 significance level. The results from the 16SrDNA data analysis are summarized in Table 7 below.

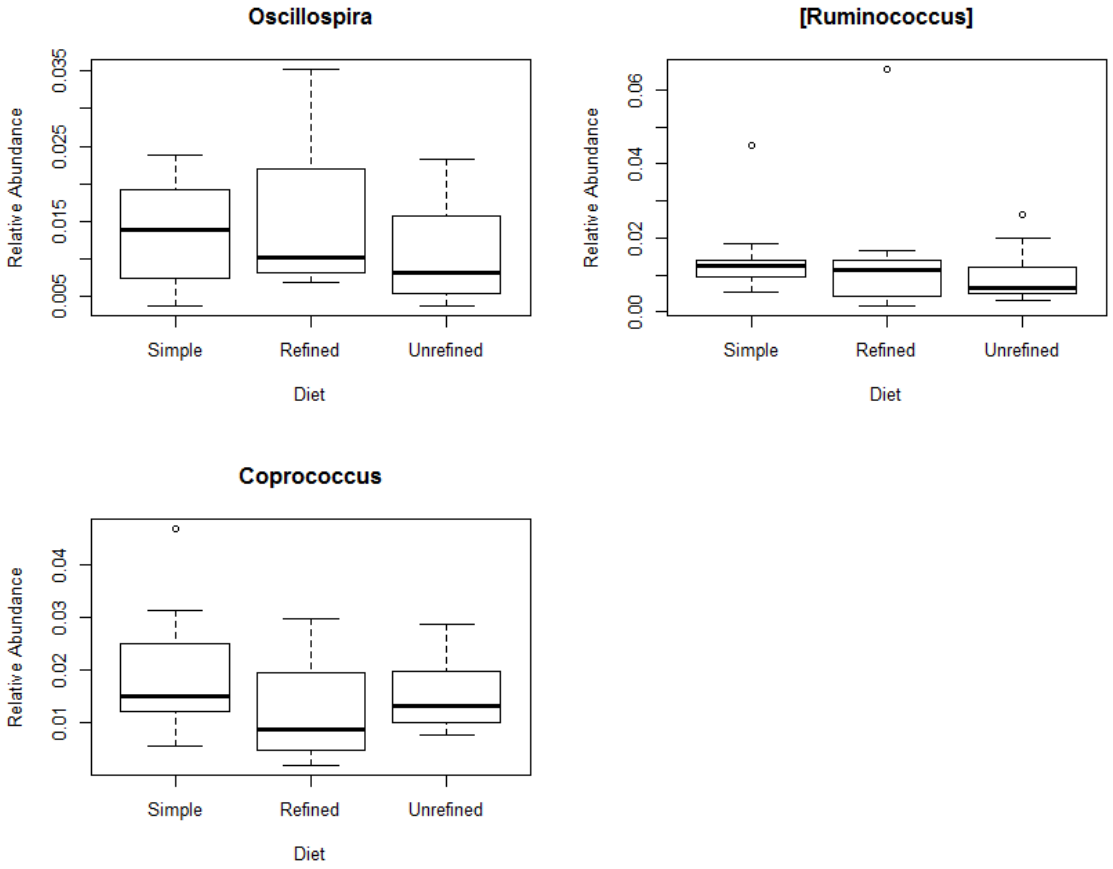
We used linear mixed effects model for analysis with normal distribution assumption on the logit transformed relative abundance data with a random intercept for subject as random effect and including a correlation structure of compound symmetry among repeated measures by the same subject for different diets. After accounting for subjects using random effect as mentioned above, the following three genera (*Coprococcus*, *Ruminococcus* and *Oscillospira*) out of the total 21 genera that we tested, have a relative abundance that significantly varies across diets with p-values of 0.0207, 0.0400 and 0.0457, and have at least a mean relative abundance of 1%. However, we do not find any genera that significantly varies across diets after adjusting for multiple testing using either Bonferroni method (Aickin & Gensler, 1996; Dunn, 1961) to maintain the familywise error rate (FWER) at a level of 0.05 or using Benjamini Hochberg method (Benjamini & Hochberg, 1995) to control False Discovery Rate (FDR) at a level of 0.05. Here, FWER is the probability of making at least one type I error in the family of all microbes that are compared for differential abundance and FDR is the proportion of false discoveries among the discoveries of differentially abundant microbes. We also performed analysis for differential abundance of microbes with respect to diets and included some covariates such as age and sex and we noted that there is no confounding of these covariates with the diet variable and hence we omitted those

covariates from our analysis. We also performed an analysis for variation of the relative abundance of the top differentially abundant microbes from the diet analysis with respect to LDL, HDL cholesterol level and Triglycerides (TG) level. We did not find any of those microbes significantly differentially abundant with respect to LDL and HDL cholesterol levels. We found that the relative abundance of *Oscillospira* significantly varies with respect to TG level after accounting for individual subjects through the random effects model described above, with a p-value of 0.0368 and the relative abundance is correlated with TG with a correlation (r) of (-0.5293). The relative abundance box plot for the top three genera is shown in Figure 16.

| Coprococcus | | | | | |
|---|---|---|---|---|---|
| P-Value: | 0.0207 | Bonferroni Adjusted P-Value: | | | 0.4342 |
| Relative Abundance | | | | | |
| Simple | | Refined | | Unrefined | |
| Mean | 1.92% | Mean | 1.20% | Mean | 1.54% |
| StdDev | 1.20% | StdDev | 0.98% | StdDev | 0.74% |
| Min | 0.56% | Min | 0.18% | Min | 0.76% |
| Max | 4.69% | Max | 2.97% | Max | 2.87% |
| | | | | | |
| **Ruminococcus** | | | | | |
| P-Value: | 0.0400 | Bonferroni Adjusted P-Value: | | | 0.8402 |
| Relative Abundance | | | | | |
| Simple | | Refined | | Unrefined | |
| Mean | 3.08% | Mean | 4.65% | Mean | 4.90% |
| StdDev | 2.85% | StdDev | 4.02% | StdDev | 4.17% |
| Min | 0.29% | Min | 0.18% | Min | 0.04% |
| Max | 10.99% | Max | 13.57% | Max | 12.20% |
| | | | | | |
| **Oscillospira** | | | | | |
| P-Value: | 0.0457 | Bonferroni Adjusted P-Value: | | | 0.9597 |
| Relative Abundance | | | | | |
| Simple | | Refined | | Unrefined | |
| Mean | 1.38% | Mean | 1.58% | Mean | 1.10% |
| StdDev | 0.72% | StdDev | 0.95% | StdDev | 0.74% |
| Min | 0.37% | Min | 0.70% | Min | 0.37% |
| Max | 2.39% | Max | 3.53% | Max | 2.32% |

Shown here is the summary results from the example diet study 16SrDNA data. Three genera (*Coprococcus*, *Ruminococcus* and *Oscillospira*) out of the total 21 genera that we tested, have a relative abundance that significantly varies across diets with p-values of 0.0207, 0.0400 and 0.0457, and have at least a mean relative abundance of 1%. However, we do not find any these genera significantly vary across diets after adjusting for multiple testing using either Bonferroni method or Benjamini Hochberg method.

**Table 7: Summary results from diet study 16SrDNA data**




Shown here is the box plot of the relative abundance of microbes grouped by diet types (simple, refined carbohydrates and unrefined carbohydrates) of top three genera (*Coprococcus*, *Ruminococcus* and *Oscillospira*) out of the total 21 genera that we tested, from the example diet study 16SrDNA data.

**Figure 16: Relative Abundance of top 3 genera of the Diet Study example**

*Multi-dimensional Analysis*

We have identified the following types of multi-dimensional analysis that can be performed in the context of analyzing microbiome data.

**Principal Component Analysis**:

Principal Component Analysis (PCA) (Jolliffe & Cadima, 2016; Pearson, 1901) is a statistical procedure that converts a set of observations which could be correlated into a set of linearly uncorrelated variables called principal components using an orthogonal transformation. The number of principal components is less than or equal to the number of original variables. This transformation is performed in such a way that the first principal component always has the largest possible variance accounting for as much of the variability in the data as possible, and each succeeding components account for the largest remaining variance and are also orthogonal to the preceding components.

For a data X with p components, the transformation W using p vector of weights $(w_1, w_2, \dots, w_p)$ is defined mathematically as follows.

$$T_{n\times p} = X_{n\times p} W_{p\times p}$$

The transformation W is constructed in such a way that the individual variables of T considered over the data set successively inherit the maximum possible variance from X, with each weighting vector w constrained to be a unit vector.

By keeping only the first L principal components, produced by using only the first L weighting vectors, we get the following truncated transformation.

$$T_{n\times L} = X_{n\times p} W_{p\times L}$$

We have also identified other multi-dimensional analysis namely Principal coordinates analysis (PCoA) (Pavoine, Dufour & Chessel, 2004) also known as metric multidimensional scaling and Non-metric multidimensional scaling (NMDS) (Levine, 1977; McGee, 1968) that are useful in the context of microbiome variation analysis.

## Aim 2D

Develop a software pipeline toolkit called PathoStat for microbiome variation analysis.

### *Objective*

Incorporate the modules and techniques developed above into a complete toolkit for microbiome variation analysis.

### *Rationale*

In order to perform microbiome variations analysis, a good toolkit is needed with rich set of visualization modules and statistically sound analysis modules. By incorporating all the modules mentioned in the aims above and designing a toolkit with easy extensibility, will be of great help to the scientific community in performing these types of analysis.

### *Experimental Plan*

Develop a module for differential abundance analysis and present the results in a suitable format for easy visualization. Develop exploratory tree module where the relative abundance of microbes is presented in a taxonomic tree format with abundance level

differentiated based on conditions of interest. Develop interactive principal component analysis and principal coordinate analysis plots where the user can choose the principal components of interest. Develop a module to perform all the multi-dimensional analysis identified in the previous aim. Implement these modules into our PathoStat software tool, which will be an interactive software pipeline developed as a Shiny app R package.

### *Exploratory Tree*

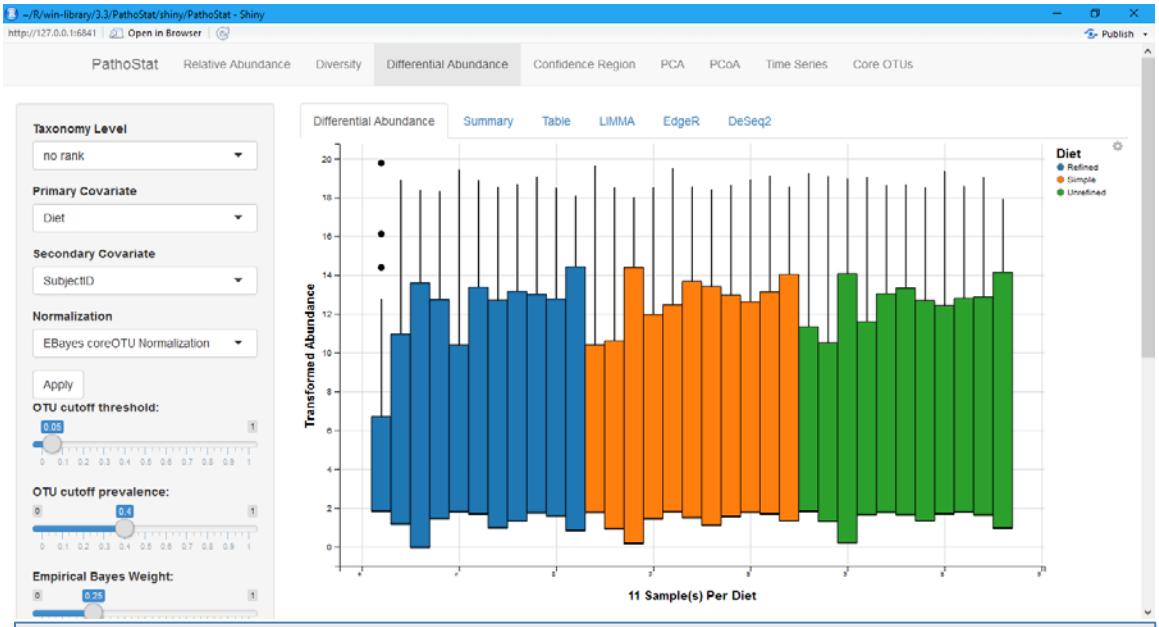
We have developed a module to show all the microbes in the samples in the form of a phylogenetic tree format. In Figure 17 below, the PathoStat exploratory tree plot with the relative abundance of the microbes at the genus level is shown for each of the samples. A circle drawn next to a microbe indicates a sample that has that microbe. The size of the circle represents the relative abundance of the microbe in that sample. The color of the circle represents the type of the sample and in this example represents one of the diet types: red for refined carbohydrates diet, green for simple diet and blue for unrefined carbohydrates diet. We can infer from this plot whether a microbe is present in a sample with a particular type of diet and whether the relative abundance of the microbe is higher or lower in a particular type of diet.


Shown here is the exploratory tree plot with the relative abundance of the microbes at the genus level for each of the samples. A circle drawn next to a microbe indicates a sample that has that microbe. The size of the circle represents the relative abundance of the microbe in that sample. The color of the circle represents the type of the sample and in this example represents one of the diet types: red for refined carbohydrates diet, green for simple diet and blue for unrefined carbohydrates diet. We can infer from this plot whether a microbe is present in a sample with a particular type of diet and whether the relative abundance of the microbe is higher or lower in a particular type of diet.

**Figure 17: PathoStat Exploratory Tree for the diet study 16SrDNA dataset**

*Differential Abundance*

We have developed a module to show the differential abundance of the microbes across the samples. Figure 18 below shows the differential abundance plot for the diet study example . Here, each bar corresponds to a sample and represents the abundance range of all the microbes in that sample. The color of the bar represents the type of the samples and in this example: blue represents refined carbohydrates, orange represents simple and green represents unrefined carbohydrates. It can be inferred that there is no significant visible differential abundance across the samples from the three types of diets.




Shown here is the differential abundance plot for the diet study example. Here, each bar corresponds to a sample and represents the abundance range of all the microbes in that sample. The color of the bar represents the type of the samples and in this example: blue represents refined carbohydrates, orange represents simple and green represents unrefined carbohydrates. It can be inferred that there is no significant visible differential abundance across the samples from the three types of diets.

**Figure 18: PathoStat differential abundance box plot for diet study example**

*Multi-dimensional analysis using BiPlot*

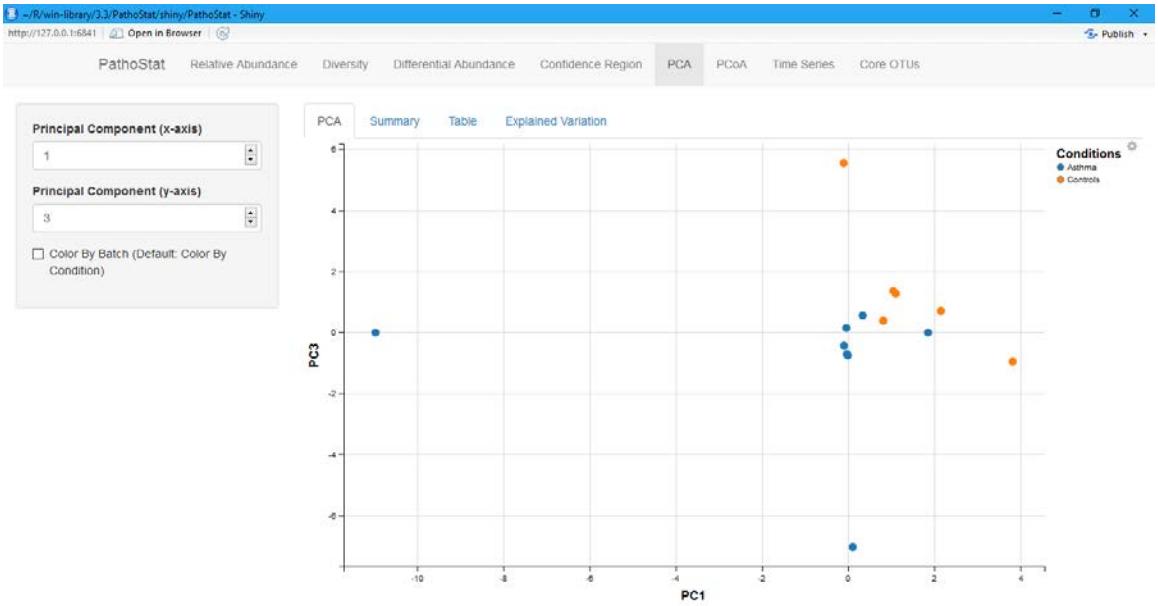
We have developed a module to perform multi-dimensional analysis using BiPlot. In Figure 19 below, multi-dimensional analysis using non-metric multidimensional scaling (NMDS) method is shown for diet study example. BiPlot features both the microbes and samples in the same plot with options to choose different methods for calculating distances between them. The samples are colored and shaped here based on diet types. The label is shown for the microbes at the genus level. We can see that there is no clustering of samples based on diet types and the microbes are spread apart indicating that they are diverse.


Shown here is the BiPlot which features both the microbes and samples in the same plot with options to choose different methods for calculating distances between them and non-metric multidimensional scaling NMDS option selected here. The samples are colored and shaped here based on diet types. The label is shown for the microbes at the genus level. We can see that there is no clustering of samples based on diet types and the microbes are spread apart indicating that they are diverse.

**Figure 19: Multi-dimensional Analysis using BiPlot**

*Principal Component Analysis*

We have developed a module to perform Principal Component Analysis (PCA). In Figure 20 below, the Principal Component Analysis (PCA) plot for asthma study dataset with the first and third principal components selected is shown. The samples are colored here based on the two types of sample conditions. We can see that there is clustering of samples based on whether the samples are from children with asthma or controls, indicating clear microbial diversity based on the sample conditions. This module also allows the user to interactively select any of the two principal components to analyze if there is any clustering of samples in any of the dimensions.

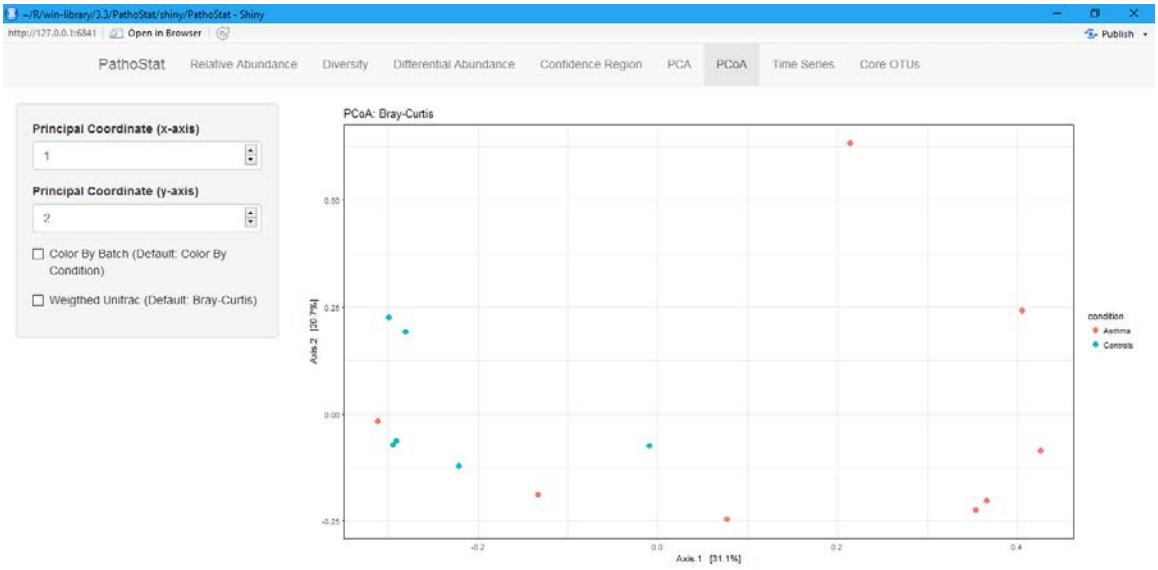



Shown here is the Principal Component Analysis (PCA) plot for asthma study dataset with the first and third principal components selected. The samples are colored here based on the two types of sample conditions: blue represents asthma and orange represents controls. We can see that there is clustering of samples based on whether the samples are from children with asthma or controls, indicating clear microbial diversity based on the sample conditions.

**Figure 20: PathoStat Principal Component Analysis plot for asthma study dataset**

*Principal Coordinate Analysis*

We have developed a module to perform the Principal Coordinate Analysis (PCoA). In Figure 21 below, the PathoStat Principal Coordinate Analysis plot for asthma study dataset with the first two principal components selected is shown. The samples are colored here based on the two types of sample conditions. We can see that there is clustering of samples based on whether the samples are from children with asthma or controls, indicating clear microbial diversity based on the sample conditions. This module also allows the user to interactively select any of the two principal coordinates to analyze if there is any clustering of samples in any of the dimensions.




Shown here is the Principal Coordinate Analysis (PCoA) plot for asthma study dataset with the first two principal coordinates selected. The samples are colored here based on the two types of sample conditions: orange represents asthma and blue represents controls. We can see that there is clustering of samples based on whether the samples are from children with asthma or controls, indicating clear microbial diversity based on the sample conditions.

**Figure 21: PathoStat Principal Coordinate Analysis plot for asthma study dataset**

## *Conclusion*

We have developed a toolkit for microbiome variation analysis called PathoStat as a shiny app R-package. It is available for download from Bioconductor at http://bioconductor.org/packages/PathoStat. PathoStat provides a rich set of visualization modules. Some of the salient features are relative abundance charts, diversity estimates and plots, tests of differential abundance and multi-dimensional analysis including principal component and principal coordinate analysis. The important feature of the package is the interactive feature of all the plots, allowing the user to choose various parameters and variables of interest and visualize all the dynamically generated plots customized according to the user selected criteria. We have developed a methodology for computing confidence region for the relative abundance estimates of the microbes in a sample and a module for displaying it. We have performed differential abundance analysis on a diet study dataset and used that as an example for design and development of the PathoStat toolkit. The toolkit is structured so that new modules can be easily added in the future. We hope that you will find this toolkit very useful when you want to analyze microbiome data.

# CHAPTER FOUR

## Project 3: Batch Effects Analysis

### Introduction

When analyzing high dimensional data, non-biological experimental variation or "batch effects" confound the true associations between the conditions of interest and the outcome variable. As shown in the Figure 1 of the Nature Reviews Genetics article (Leek *et al.*, 2010), batch effects exist even after normalization. Hence, unless the batch effects are identified and removed, any attempts for downstream analyses, such as network inference and estimation, will likely be error prone and may lead to false positive results. Furthermore, many batch adjustment approaches artificially induce a correlation structure in the batch adjusted data that can often exaggerate the significance of results (e.g. p-values) or even introduce spurious relationships in the data. This motivates the need for a computational framework to systematically identifying batch effects. Here, the aim is to develop a tool called BatchQC that visually depicts aspects of high dimensional data and evaluates the extent to which batch effects impact the association between the conditions of interest. Broadly, the aims of this project are to do the following: A) Analyze the effect of correlation of the batch adjusted data and develop new techniques to account for correlation in two step hypothesis testing approach. B) Develop a software pipeline to identify whether batch effects are present in the data and adjust for batch effects in a suitable way.

## Aim 3A

Analyze the effect of correlation of the batch adjusted data and develop an appropriate workflow to account for correlation in two-step Hypothesis Testing approach.

### *Objective*

Batch effects introduce bias and confounding into experiments when experimental designs are unbalanced. Batch adjustment using a linear model introduces correlation into the data values within each batch whether or not experimental covariates are included. For unbalanced designs this correlation structure can lead to exaggerated significance in downstream analyses. This problem can be avoided if batch effect is directly modeled into the downstream analyses, but this procedure can be a daunting task for the normal user when doing complex tasks. Also, the mean-only adjustment may not be sufficient because the batch effect may affect the mean, variance and higher order moments of the distribution of the data. The objective is to account for this correlation and develop better techniques for analysis of data in the presence of batch effects.

### *Rationale*

It is common to have both known and unknown batch effects in sequencing and microarray expression data and it is automatically adjusted in many cases without the availability of original unadjusted data. However, when the expression data are adjusted for batch effects, unwanted correlation is introduced and can lead to exaggerated

significance in unbalanced designs. Even when the data are unadjusted, unless both the known and unknown batch effects are accounted for, analysis for differential expression with the conditions of interest will not be accurate. Batch effects can be modeled directly as a mixed effect model, but this procedure can be a daunting task for the normal user, who would prefer to first adjust for batch using tools designed for adjusting batch effects, especially when the batch effect affects the mean, variance and higher order moments of the distribution of the data. Hence, the new approaches and workflows to account for correlation when performing two-step hypothesis testing after adjusting for batch effects using tools designed for batch effects, will have profound applicability in the context of analyzing expression data.

### *Experimental Plan*

Determine a way to represent the variance structure for multiple genes and correlated samples in the analysis. Estimate the covariance matrix and develop appropriate significance tests or procedures accounting for the correlation with respect to the differential expression analysis with conditions of interest.

### *Example datasets*

For our analysis, we have used three datasets: 1) Bladder cancer, 2) Nitric oxide and 3) Oncogenic signature.

*Bladder cancer dataset*

This is a dataset from the R package called bladderbatch (Leek, 2016) (http://bioconductor.org/packages/bladderbatch/). This package contains microarray gene expression data on 57 bladder samples from 5 batches and 3 conditions. The batch and condition are highly confounded in this data set making it a suitable data set for experimenting with batch effect analysis. The number of samples in each batch and condition of the data set is presented in the table below. The batch and condition are confounded with a standardized Pearson correlation coefficient of 0.89.

| | Batch 1 | Batch 2 | Batch 3 | Batch 4 | Batch 5 |
|---|---|---|---|---|---|
| Condition Biopsy | 0 | 0 | 0 | 5 | 4 |
| Condition Cancer | 11 | 14 | 0 | 0 | 15 |
| Condition Normal | 0 | 4 | 4 | 0 | 0 |

**Table 8: Number of samples in each batch and condition of bladder cancer dataset**

*Nitric oxide dataset*

This dataset is described in this publication (Johnson, Li & Rabinovic, 2007). This dataset resulted from an oligonucleotide microarray (Affymetrix HG-U133A) experiment. This experiment was repeated at three different times or in three batches (totaling 12 samples). In this dataset the batch and condition are perfectly balanced and thus providing another example case of batch effect analysis. The number of samples in each batch and condition of the data set is presented in the table below. The batch and condition are perfectly balanced with a standardized Pearson correlation coefficient of 0.

| | Batch 1 | Batch 2 | Batch 3 |
|---|---|---|---|
| Condition Control 0 | 1 | 1 | 1 |
| Condition Control 7 | 1 | 1 | 1 |
| Condition NO 0 | 1 | 1 | 1 |
| Condition NO 7 | 1 | 1 | 1 |

**Table 9: Number of samples in each batch and condition of nitric oxide dataset**

*Oncogenic signature dataset*

This dataset consists of sequencing data captured from human mammary epithelial cells after activating key growth pathway genes (GEO accession GSE73628) (Manimaran, Selby, Okrah et al., 2016). The data consists of three batches and ten different conditions corresponding to control and activation of nine different pathways. The batch and condition are confounded in this data set and thus providing another example case of batch effect analysis. The number of samples in each batch and condition of the data set is presented in the table below. The batch and condition are confounded with a standardized Pearson correlation coefficient of 0.92.

| | Batch 1 | Batch 2 | Batch 3 |
|---|---|---|---|
| Condition 1 | 6 | 12 | 9 |
| Condition 2 | 6 | 0 | 0 |
| Condition 3 | 0 | 6 | 0 |
| Condition 4 | 0 | 6 | 0 |
| Condition 5 | 0 | 5 | 0 |
| Condition 6 | 0 | 6 | 0 |
| Condition 7 | 0 | 6 | 0 |
| Condition 8 | 0 | 0 | 9 |
| Condition 9 | 0 | 0 | 9 |
| Condition 10 | 0 | 0 | 9 |

**Table 10: Number of samples in each batch and condition of oncogenic signature dataset**

### *Hypothesis Testing using simulated data*

We will use the bladder cancer dataset described above as a reference to simulate different samples without condition effect (null data) but with batch variations and analyze the effect of batch variation on the hypothesis testing for variation on the conditions of interest. This simulated data set will have the same number (n=57) and experimental design as the original bladder cancer data set, thus the null data will have the same batch and condition information. Only the expression data of those samples are modified using this simulated data with no condition effects (null data) but with batch effects. Based on this analysis the best approach for analyzing data in the presence of batch is selected.

### *Analysis methods in the presence of batch*

There are broadly two types of analysis that can be performed in the presence of batch. They are *one step* analysis and *two step* analysis methods. In one step analysis, the batch analysis and the downstream analysis are performed in one step — batch is directly accounted for in the downstream model. In two step analysis, batch analysis and adjustment is performed in the first step to batch adjust the data and the downstream analysis is performed in the batch adjusted data in the second step.

*One step analysis simulation*

**Goal**: Establish that when we perform differential expression analysis in one step by including batch as a covariate, the type I error rate is controlled at the significance level of the test, as expected.

**Method**: Let y represent the measured expression data. In the one-step analysis, both the condition of interest and the batch effect covariates are adjusted for at the same time as follows:

$$y = X_1\beta_1 + X_2\beta_2 + \alpha_0 + e$$

$$X_1 : Biological\ condition\ design;$$

$$X_2 : Batch\ Design$$

Null Data simulated from standard normal distribution with variance 1 and means with no effect for the Biological condition of interest ($\beta_1 = 0$) and five batches with means equal to 0, 1, 2 , 3 and 4 respectively.

**Models Compared**:

$$y = X_1\beta_1 + X_2\beta_2 + \alpha_0 + e$$

$$y = X_2\beta_2 + \alpha_0 + e$$

**Result**:

The p-values distribution of one-step analysis with null data is shown in Figure 22 below, which looks similar to a uniform distribution as expected for p-values to follow (Murdoch, Tsai & Adcock, 2008). We also ran 100 runs of the simulation and the mean

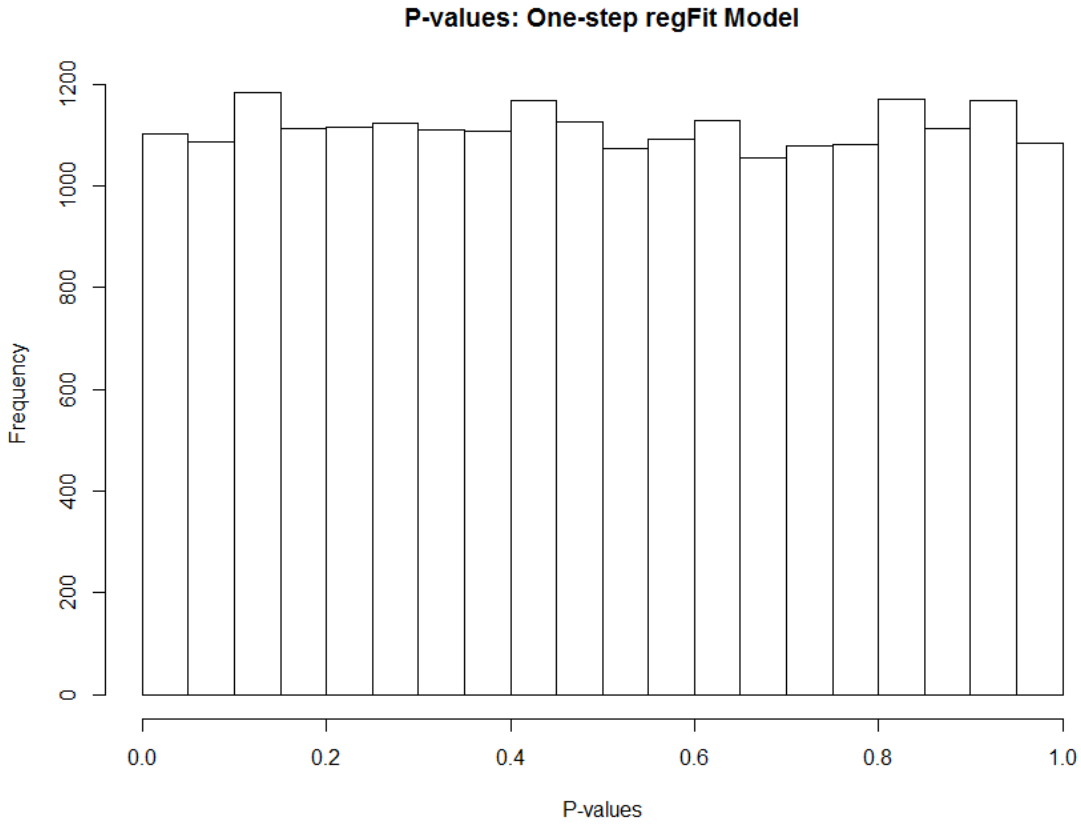
proportion of genes that were significant at 0.05 level were 0.0501 with a range of 0.0462 to 0.0535, which is very close to the 0.05 level that is expected.

**Partial Correlation Test**:

We also performed partial correlation test for the differential expression of the expression data of genes with the conditions of interest after accounting for the batch covariate. The p-values distribution of one-step analysis with null data using partial correlation test is shown in Figure 23 below, which looks similar to a uniform distribution as expected as expected for p-values to follow. We also ran 100 runs of the simulation and the mean proportion of genes that were significant at 0.05 level were 0.0498 with a range of 0.0449 to 0.0543, which is very close to the 0.05 level that is expected.

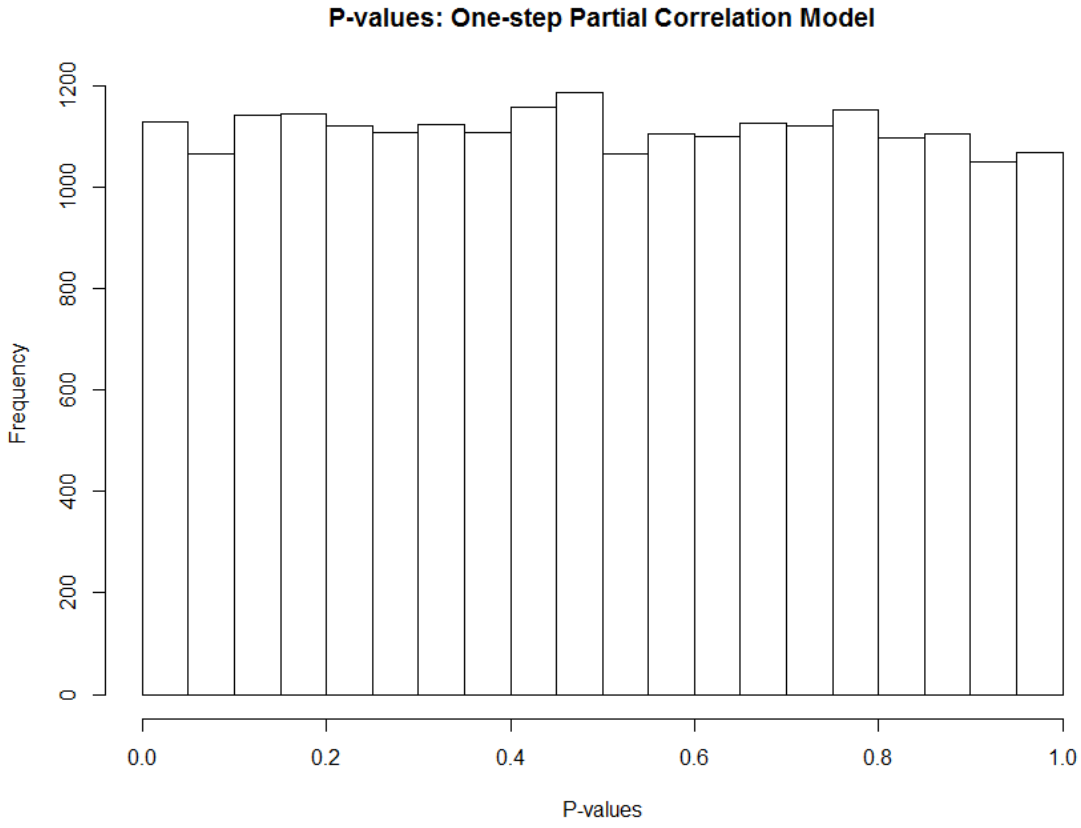
**Conclusion**:

One step differential expression analysis by including batch as a covariate or with a partial correlation test between expression data and conditions of interest while accounting for batch, maintains the type I error rate at the significance level of the test.




Shown here is the p-values distribution of one-step analysis with no condition effect (null) data. This looks similar to a uniform distribution as expected.

**Figure 22: P-values distribution of one-step analysis with null data**




Shown here is the p-values distribution of one-step analysis with no condition effect (null) data using partial correlation test between expression data and condition while controlling for batch. This looks similar to an uniform distribution as expected.

**Figure 23: P-values distribution of one-step analysis with null data using partial correlation test**

*Two-step analysis simulation*

**Goal:** Establish that there is a problem of "exaggerated significance" when we perform differential expression analysis on batch adjusted data in two-step.

**Method:** In the two-step analysis, the data is first adjusted for batch effect covariates (y*: batch adjusted data) and then a study is performed for association with respect to conditions of interest as follows:

$$\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{X_2}\widehat{\boldsymbol{\beta}}_{\boldsymbol{2}}$$

**Models Compared**:

$$y^* = X_1\beta_1 + \alpha_0 + e$$

$$y^* = \alpha_0 + e$$

**Result**:

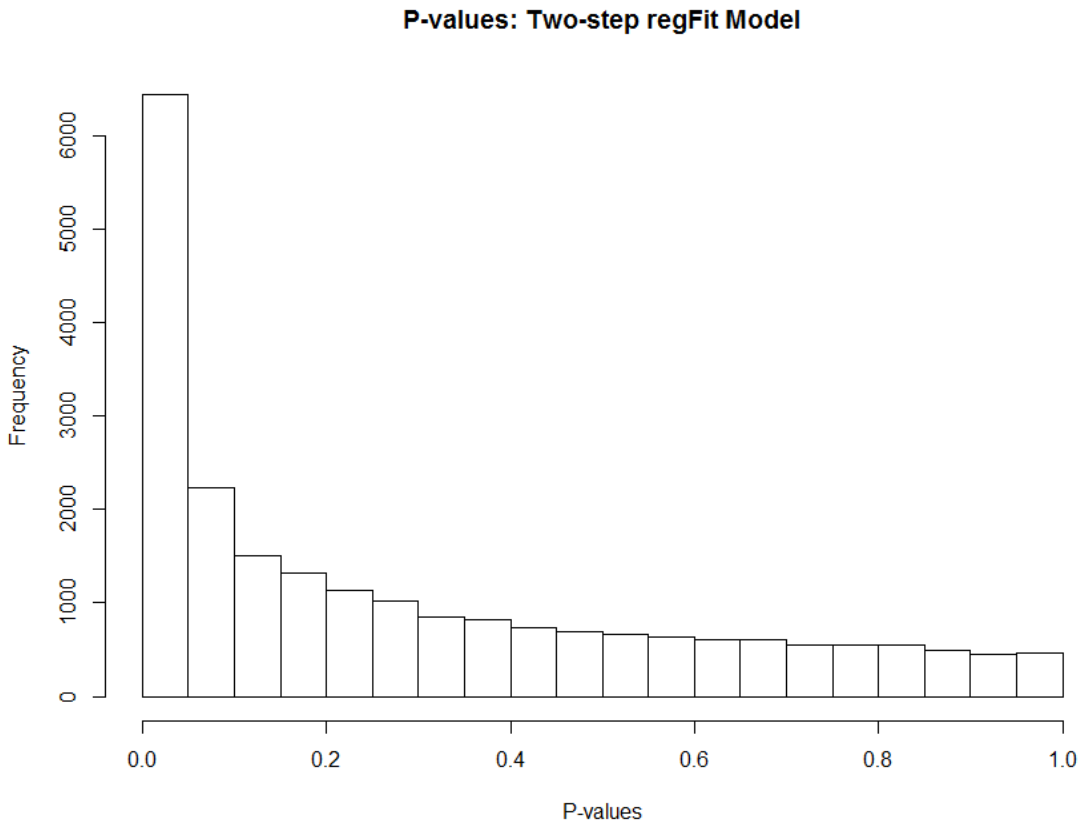
The p-values distribution of two-step analysis with null data without including batch as a covariate is shown in Figure 24 below. We also ran 100 runs of the simulation and the mean proportion of genes that were significant at 0.05 level were 0.2866 with a range of 0.2771 to 0.2936. Under the null hypothesis, the proportion of false positive tests is expected to be equal to the alpha significance level. With more p-values shifted to the lower end, we see this problem of "exaggerated significance".

**Two-step Partial Correlation Test**:

We also performed partial correlation test for the differential expression of the batch adjusted expression data of genes with the conditions of interest after accounting for the batch covariate. The p-values distribution of two-step analysis with null data using partial correlation test is shown in Figure 25 below. We also ran 100 runs of the simulation and the mean proportion of genes that were significant at 0.05 level were 0.2094 with a range of 0.2028 to 0.2178. With more p-values shifted to the lower end, we again see this problem of "exaggerated significance".

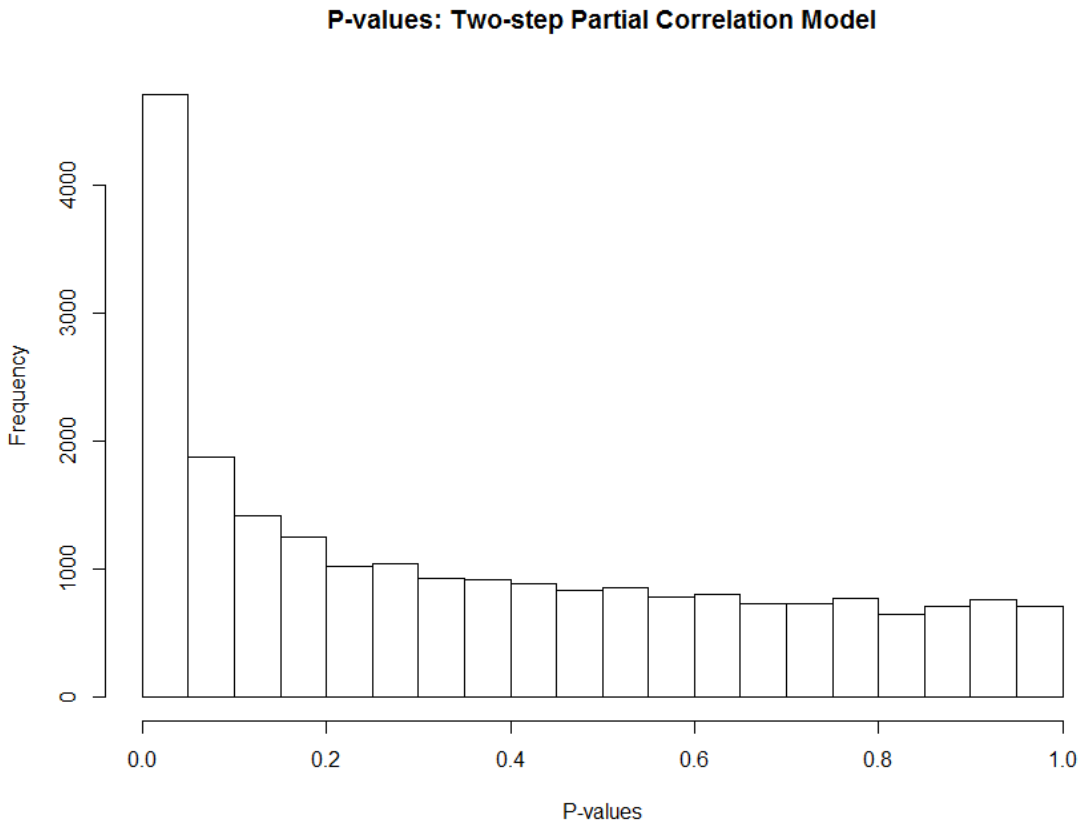
**Conclusion**:

Two step differential expression analysis by first adjusting for batch and then testing without batch as covariate or with a partial correlation test between the batch adjusted expression data and conditions of interest while accounting for batch, does not maintain the type I error rate at the significance level of the test when the number of cases and controls is unbalanced in each batch and we see exaggerated significance due to the unbalanced design and because the batch adjusted data is correlated with reduced degrees of freedom.




Shown here is the p-values distribution of two-step analysis on batch adjusted data with no condition effect (null) data. We expect to see uniform distribution for null data, but we see exaggerated significance here with more p-values at the lower end due to the unbalanced design and because the batch adjusted data is correlated with reduced degrees of freedom.

**Figure 24: P-values distribution of two-step analysis with null data**




Shown here is the p-values distribution of two-step analysis using partial correlation test between expression data and condition while controlling for batch on batch adjusted data with no condition effect (null) data. We expect to see uniform distribution for null data, but we see exaggerated significance here with more p-values at the lower end due to the unbalanced design and because the batch adjusted data is correlated with reduced degrees of freedom.

**Figure 25: P-values distribution of two-step analysis with null data using partial correlation test**

*Batch Adjusted Data: Correlated*

We will show that the batch adjusted data is correlated with the derivation for the distribution of the batch adjusted data.

Let us define the following notations.

y: the original data

y*: Batch adjusted data

$X_2$: Batch design matrix

$X_1$: Design matrix for conditions of interest

I: Identity matrix

Let $y = X\beta + e$ and the samples in y are independent and identically distributed (i.i.d.).

Let us partition the model into two components as follows:

$$X = [X_1 \quad X_2]$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$$

$$y = [X_1 \quad X_2] \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + e$$

In order to find the least square estimate of $\beta$ given by $\hat{\beta}$, we need to find $\beta$ that minimizes the distance $(y - X\beta)'(y - X\beta)$. In order to minimize the distance, we need to partial differentiate w.r.t. $\beta$ and equate to 0.

By doing that, we get the following:

$$X'(y - X\hat{\beta}) = 0$$

By partitioning $X$ in to $X_1$ $and$ $X_2$, we get the following equations:

$$X_1'X_1\hat{\beta}_1 + X_1'X_2\hat{\beta}_2 = X_1'y \quad (1)$$

$$X_2'X_1\hat{\beta}_1 + X_2'X_2\hat{\beta}_2 = X_2'y \quad (2)$$

From the first equation we get the following:

$$\hat{\beta}_1 = (X_1'X_1)^{-1}X_1'(y - X_2\hat{\beta}_2)$$

We need to eliminate $\hat{\beta}_1$ from equation (2) in order to obtain $\hat{\beta}_2$. For this purpose, we first multiply equation (1) by $X_2'X_1(X_1'X_1)^{-1}$ to get the following.

$$X_2'X_1\hat{\beta}_1 + X_2'X_1(X_1'X_1)^{-1}X_1'X_2\hat{\beta}_2 = X_2'X_1(X_1'X_1)^{-1}X_1'y \quad (3)$$

When the third equation is taken from the second equation, we get the following.

$$(X_2'X_2 - X_2'X_1(X_1'X_1)^{-1}X_1'X_2)\hat{\beta}_2 = X_2'y - X_2'X_1(X_1'X_1)^{-1}X_1'y \quad (4)$$

Let us define the following:

$$H_1 = X_1(X_1'X_1)^{-1}X_1' \quad (5)$$

The fourth equation can be rewritten as follows.

$$(X_2'(I - H_1)X_2)\hat{\beta}_2 = X_2'(I - H_1)y \quad (6)$$

From the above, we get the following.

$$\hat{\beta}_2 = (X_2'(I - H_1)X_2)^{-1}X_2'(I - H_1)y \quad (7)$$

Let us define the following:

$$H_{12} = X_2(X_2'(I - H_1)X_2)^{-1}X_2'(I - H_1)$$

The batch adjusted data is the following:

$$y^* = y - X_2\hat{\beta}_2$$

$$y^* = y - X_2(X_2'(I - H_1)X_2)^{-1}X_2'(I - H_1)y$$

$$y^* = (I - H_{12})y$$

If we assume $y \sim N(X\beta, \sigma^2 I)$ with variance $\sigma^2$, then

$$y^* \sim N(X_1\beta_1, \sigma^2(I - H_{12})(I - H_{12})').$$

Thus we see that the batch adjusted data is correlated (as long as the batch design is unbalanced).

Because the batch adjusted data is correlated, we can use the correlation information present in the batch adjusted data to use it in all downstream analysis for accurate analysis of differential expression along the conditions of interest ($X_1$). We will develop a two-step analysis approach that takes in to account the correlation of the data.

*Two-step analysis with batch as covariate*

**Goal:** Develop a two-step analysis method without the exaggerated significance problem we mentioned above, when we perform differential expression analysis on batch adjusted data in two steps.

**Method:** Here, the data is first adjusted for batch effect covariates (y*: batch adjusted data) and then a study is performed for association with respect to conditions of interest and including batch as covariate as follows:

$$\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{X_2}\widehat{\boldsymbol{\beta}}_{\boldsymbol{2}}$$

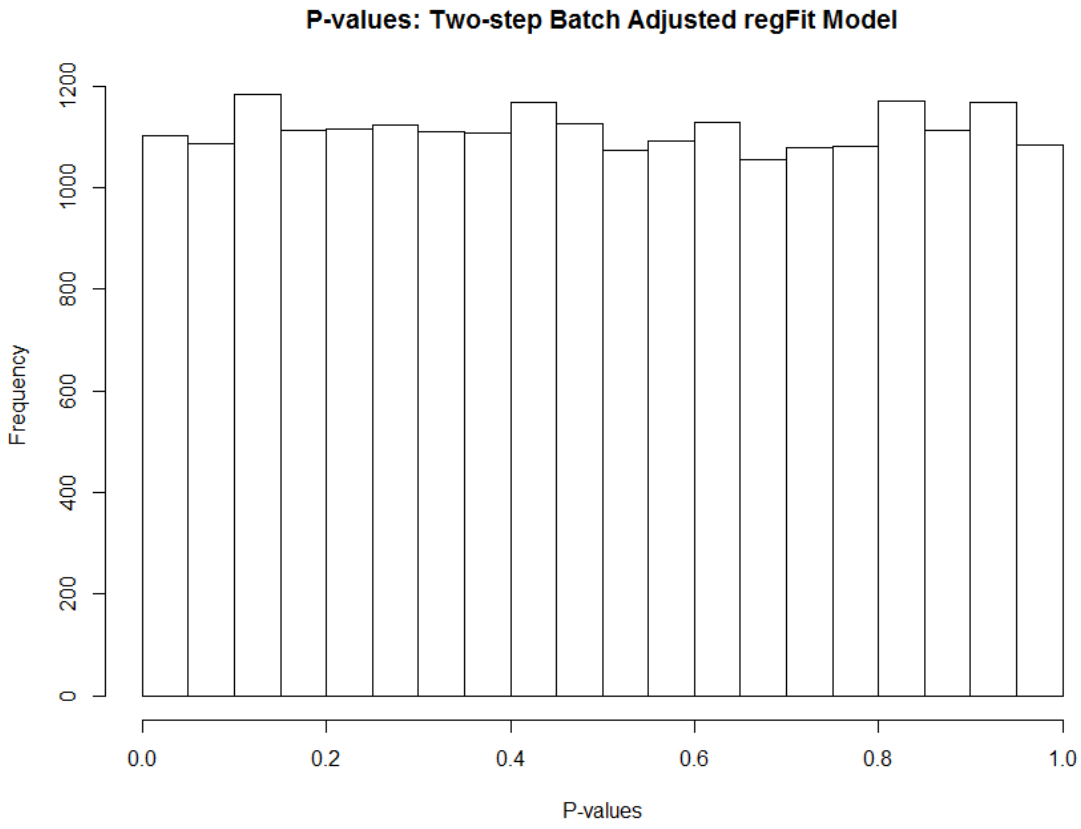
**Models Compared**:

$$y^* = X_1\beta_1 + X_2\beta_2 + \alpha_0 + e$$

$$y^* = X_2\beta_2 + \alpha_0 + e$$

**Result**: Once the batch is included as a covariate in the two-step analysis with batch adjusted data, the distribution of the p-values as shown in Figure 26, remains similar to a uniform distribution as expected with null data. We also ran 100 runs of the simulation and the mean proportion of genes that were significant at 0.05 level were 0.0501 with a range of 0.0462 to 0.0535, which is very close to the 0.05 level that is expected.

**Conclusion**: If we know the batch design of the batch adjusted data, then two-step analysis with batch as covariate maintains the type-I error rate at the given significance level.




Shown here is the p-values distribution of two-step analysis with batch as covariate on batch adjusted data using no condition effect (null) data. This looks similar to an uniform distribution as expected.

**Figure 26: P-values distribution of two-step analysis with batch as covariate using no condition effect (null) data**

*Two-step analysis with correlation*

**Goal:** Develop a two-step analysis method without the exaggerated significance problem as mentioned above, when we perform differential expression analysis on batch adjusted data in two steps, for the case when batch design is not known.

**Method:** In the two-step analysis, the data is first adjusted for batch effect covariates (y*: batch adjusted data) and then a study is performed for association with respect to conditions of interest with estimated correlation. Here the covariance matrix ($\Sigma$) is estimated as the sample covariance of the batch adjusted expression data using the information from multiple genes. We then use the Generalized Least Squares method as follows:

$$\boldsymbol{y^*} = \boldsymbol{y} - \boldsymbol{X_2\widehat{\beta}_2}$$

**Models Compared**:

$$y^* = X_1\beta_1 + \alpha_0 + e$$

$$y^* = \alpha_0 + e$$

In the generalized least squares approach, first the covariance matrix ($\Sigma$) is decomposed in to triangular matrices using Cholesky Decomposition (Dereniowski & Kubale, 2004) method and the transformed data and design matrices are used in the regular regression method as follows:

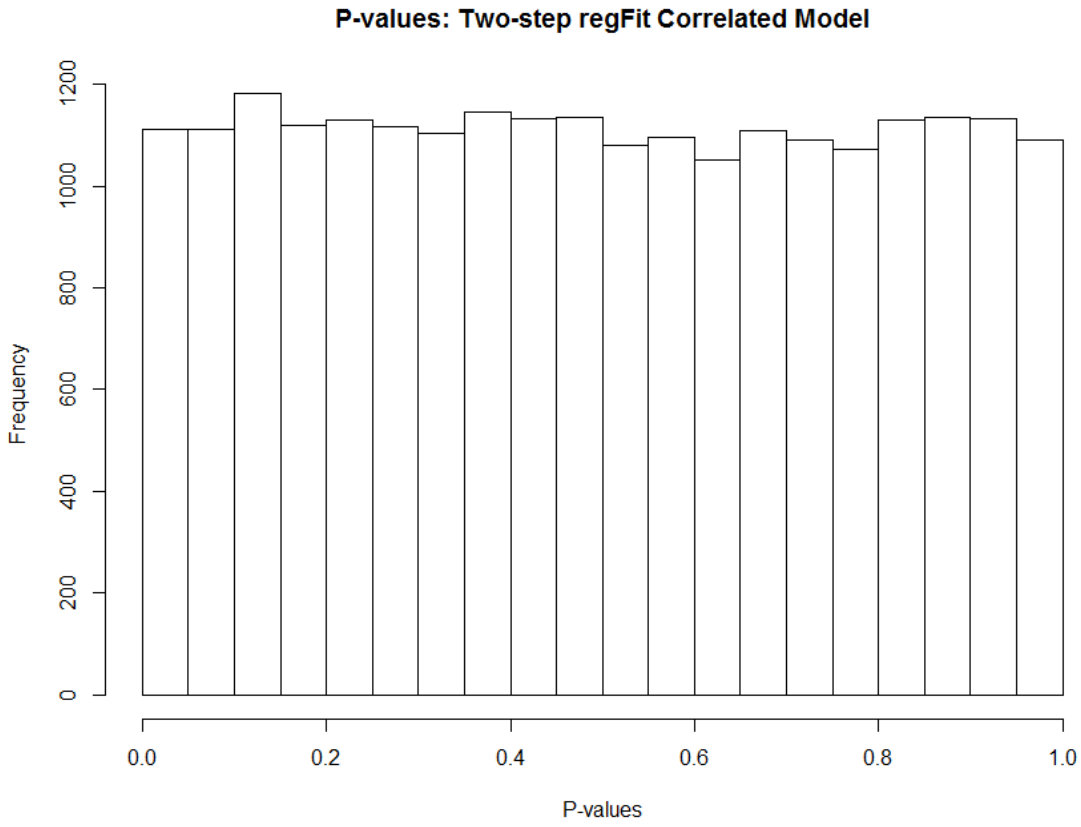
$$\Sigma = SS^T$$

S is a triangular matrix here using the Choleski Decomposition. In the generalized least squares approach, solving $y = X\beta + \varepsilon$, reduces to the following:

$$S^{-1}y = S^{-1}X\beta + S^{-1}\varepsilon$$

$$y' = X'\beta + \varepsilon'$$

One issue that has to be noted here is that the batch adjusted data is not full rank and hence the covariance matrix will not be positive definite for performing the above generalized least squares method. Hence, a reduced data matrix to make it full rank is needed to perform the above operation.




Shown here is the p-values distribution of two-step analysis with correlation analysis on batch adjusted data using no condition effect (null) data. This looks similar to an uniform distribution as expected.

**Figure 27: P-values distribution of two-step with correlation analysis on null data**

**Result**:

We see that the distribution of p-values as shown in Figure 27 looks similar to a uniform distribution as expected for null data, when performing the two-step with correlation analysis as described above. We also ran 100 runs of the simulation and the mean proportion of genes that were significant at 0.05 level were 0.0509 with a range of 0.0478 to 0.0545, which is very close to the 0.05 level that is expected.

**Conclusion**:

If we use correlation of the batch adjusted data in the two-step analysis, the type-I error rate is maintained at the given significance level. This gives an effective way for solving the exaggerated significance problem that is introduced in the two-step approach even when the exact batch design is unknown, as long as we can estimate the correlation in the samples through multiple gene measurements.

*Multiple methods comparison*

**Goal:** Perform a thorough analysis of the batch effect exaggerated significance problem using three different datasets. Also analyze the performance by including batch as a covariate when adjusting for batch using ComBat (Johnson *et al.*, 2007) and using LIMMA (Ritchie, Phipson, Wu et al., 2015) for differential expression analysis.

**Method:** We repeated the same analysis mentioned above on three different datasets: Nitric Oxide data set, Bladder batch cancer data set and Oncogenic signature data set. We used 50 runs of random simulated data with no condition effect (null data) for

differential expression analysis. We also tested the performance with combinations of ComBat for batch adjustment and using LIMMA for differential expression analysis.

**Results**: We noticed that including batch as a covariate addresses the exaggerated significance problem directly as summarized in Table 11 below. We note here that when performing differential expression analysis on a random null data using LIMMA with batch as a covariate, approximately 5% of the genes are significant at an alpha level of 0.05 as expected and thus avoiding the exaggerated significance problem. When adjusting for batch using ComBat, we adjusted for both mean and variance differences in batch. In the simulated random data, there was no variance differences by batch included in the simulation. When performing differential expression analysis on a random null data using LIMMA with Batch as a covariate on batch adjustment using ComBat, approximately 8.6%, 6.2% and 5.5% of the genes are significant at an alpha level of 0.05 for the three data sets Nitric Oxide, Bladder cancer and Oncogenic signature data sets respectively, with a slight exaggerated significance because of the additional adjustment on batch variances which was not there in the simulated data set.

**Conclusion**: We conclude that for the users who prefer to first adjust for batch using tools designed for adjusting batch effects and conduct the differential expression analysis in two-step, we have presented a way to include batch as a covariate when the batch design is available, or estimate the covariance matrix and conducting the two-step analysis with correlation structure that effectively avoids the exaggerated significance problem.

| | Proportion of Significant Genes (alpha = 0.05) | | | |
|---|---|---|---|---|
| Dataset | LIMMA without Batch | ComBat + LIMMA without Batch | LIMMA+ Batch | ComBat + LIMMA + Batch |
| Nitric oxide | 109/22,283<br>0.00487<br>(0.0039 - 0.00565) | 2,917/22,283<br>0.131<br>(0.126 - 0.138) | 1,116/22,283<br>0.05<br>(0.045 - 0.054) | 1,927/22,283<br>0.086<br>(0.0802 - 0.0919) |
| Bladder cancer | 17,374/22,283<br>0.78<br>(0.774 - 0.787) | 3,574/22,283<br>0.16<br>(0.155 - 0.166) | 1,116/22,283<br>0.05<br>(0.047 - 0.053) | 1,384/22,283<br>0.062<br>(0.058 - 0.066) |
| Oncogenic signature | 10,286/18,052<br>0.57<br>(0.561 - 0.578) | 1,628/18,052<br>0.09<br>(0.085 - 0.095) | 911/18,052<br>0.05<br>(0.047 - 0.055) | 985/18,052<br>0.0546<br>(0.051 - 0.059) |

Shown here is the proportion and range of significant genes that are differentially expressed when performing analysis through combination of different methods including LIMMA for differential expression analysis and ComBat for batch adjustment with 50 runs of random simulated dataset. The proportion of genes expected to be differentially expressed at an alpha significance level of 0.05 is 5%. We note that in LIMMA with Batch as a covariate analysis, approximately 5% of the genes are significant at an alpha level of 0.05 as expected and thus avoiding the exaggerated significance problem.

**Table 11: Proportion of significant genes with different combination of batch adjustment using ComBat and differential expression analysis using LIMMA**

## Aim 3B

Develop a software pipeline to identify whether batch effects are present in the data and adjust for batch effects in a suitable way in collaboration with others by leading and providing major contribution to the design and development of the pipeline.

### *Objective*

Develop a software pipeline to visualize the expression data and its variation along known and unknown batches. Determine whether batch effects exist and automatically adjust for batch effects in the appropriate way. Batch effects can be adjusted using either mean only batch adjustment method or both mean and variance batch adjustment method and optionally include higher moments. When adjusting for batch effects, whether parametric assumption for the underlying distribution of the expression can be made needs to be determined, and if not, a more general empirical distribution needs to be used for the given dataset. The best possible batch adjustment method will be chosen for the given data after determining whether batch effects need to be addressed in the dataset.

### *Rationale*

It is often the case that sequencing/microarray samples need to be collected in multiple batches over time. There exist both known and unknown batches in these samples that unless properly adjusted for, can lead to bias in the downstream analysis.

There are several batch adjustment tools that are currently available, but none of them can a priori indicate whether batch adjustment need to be done, and also what is the best way to adjust for batch before proceeding with the analysis. The new software pipeline called BatchQC will address these issues with respect to batch effects.

### *Experimental Plan*

Develop a software pipeline called BatchQC that will first summarize the data and display a boxplot of the expression data with samples colored by batch for easy visualization to look for batch effects. The report will have a heatmap of the expression for selected top differentially expressed genes and the samples colored by batch at the top, which will help in identifying patterns of differential gene expression with respect to batch. It will have a sample correlation heatmap to display the pairwise correlations between samples so that we can see whether samples from the same batch are correlated. The report will also have principal components plots with an option for the user to select the principal components and by default will display the top two principal components. The principal component plots will also be colored by batches so that the user can see whether there is any clustering based on batches. In addition, there will be some statistical test performed through linear regression of principal components with the batch variables to test whether the batch variables are significantly associated with the each of the principal components. Once whether batch adjustment needs to be performed is established through the help of the above tests and plots, the method to adjust for batch is determined by comparing the assumed distribution plots of the data from one batch to

another to see whether there are differences in mean, variance or higher order moments across batches.

*BatchQC shiny app R-package*

BatchQC is a user friendly interactive Shiny app R-package that is easy to install and use. It is available for R versions 3.3.0+ (and also requires pandoc 1.12.0+). The entry point of this interface is the function batchQC, which inputs the data matrix, batch variables, and conditions of interest and produces three output formats: R Shiny App interface in 'interactive' mode, an optional static html report, and an optional BatchQCout data object. The Shiny App interface then allows the user to interactively evaluate sample and batch effects, and run batch adjustment using the ComBat and SVA methods. The interface makes batch effect evaluation interactive and 'hands-on' and makes BatchQC accessible for users with diverse backgrounds. The html report and data object will allow BatchQC to be integrated into automated pipeline projects. BatchQC can be applied to multiple types of '-omics' data that can be generated from multiple profiling platforms, including sequencing and microarrays. The tool can be applied to high-throughput data from multiple applications, including genomics, epigenomics, proteomics, and metabolomics, etc.

***Need for multiple batch effect metrics:*** The major advantage of BatchQC are the multiple evaluations available to identify and explore batch effects. These evaluations are important because datasets often successfully pass some diagnostic metrics, but then fail others. Using the first simulated dataset included in the package, batch differences were

clearly seen in the boxplots and the Principal Components Analysis (PCA) analysis. However, using the second simulated dataset, PCA did not reveal these batch differences, while the boxplot and heatmaps did show batch differences. Also, using a real data example included in the package, batch differences were not seen in the boxplots and heatmaps, but the PCA analysis identified strong batch effects. The batch effects in these datasets indicate that different metrics are frequently needed for batch diagnostics in different datasets, and that multiple diagnostics can help the researcher efficiently develop a batch adjustment strategy in each individual case.

## Analysis

BatchQC begins with a Summary tab that provides Confounding, Variation Analysis and P-value Analysis subtabs. The Confounding subtab provides a summary table of the experimental design and displays the number of samples in each batch and condition. In addition, it provides the standardized Pearson Correlation Coefficient and Cramer's V Confounding Coefficient that measure the level of confounding in the experimental design between batch and condition of interest. Because the simulated data comes from a balanced experimental design (e.g. no batch/condition confounding), the Standardized Pearson Correlation Coefficient and Cramer's V Confounding Coefficient are both 0—correctly illustrating that there is no confounding between batch and condition. In contrast, in the signature dataset these coefficients are 0.92 and 0.80, due to the fact that the batches do not share any experimental conditions in common except for the control samples. This indicates that failure to remove batch effects in the signature data could potentially lead to significant confounding in downstream analyses (e.g. differential

expression) due to batch variation. In contrast, removing the batch effects in the simulated data is likely to be less important since the balanced design leads to independence between the batch and condition variables.

The Variation Analysis subtab presents the percentage variation explained by full (condition + batch), condition and batch across the genes as a box plot and table. In Figure 28 and Figure 29, the percentage variation explained by batch in both datasets is greater than the average percent of variation explained by condition. Because the batch effect variation is larger than the experimental/condition variation, there is a clear need for batch adjustment in these data. The P-value Analysis subtab shows the distributions of batch and condition effect p-values across genes. The distribution of Batch Effect p-values in Figure 30, illustrate many genes with low p-values (more than expected by chance), indicating that there is a batch effect in the real data. The simulated dataset (not shown) shows an even more extreme need for batch adjustment based on this metric.

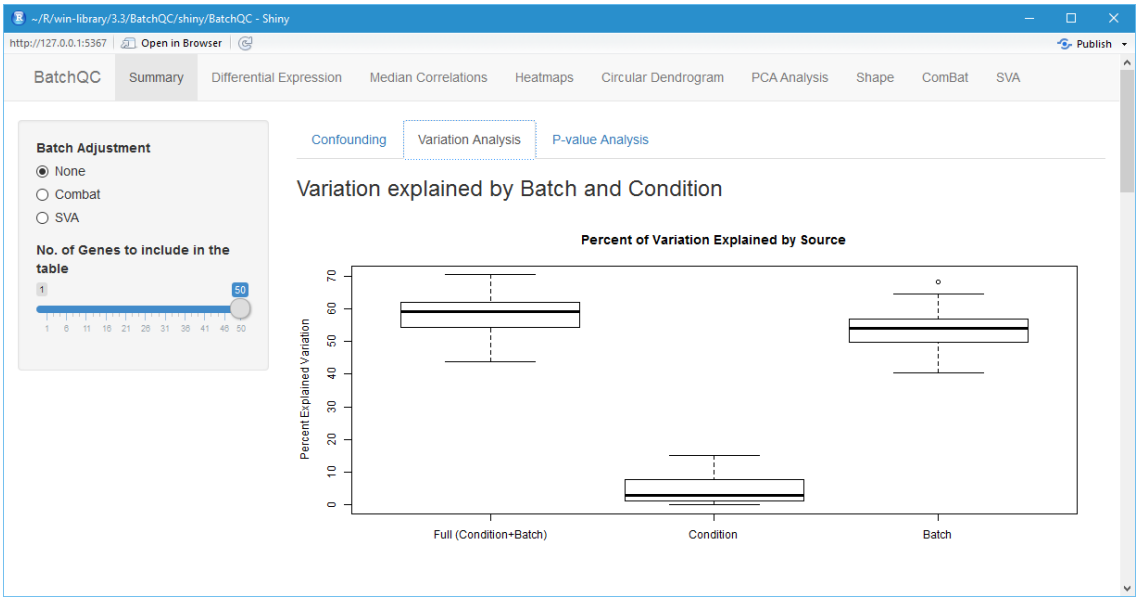



**Figure 28: Variation explained by Batch and Condition for the simulated data:**

**Huge variation explained by Batch than Condition**

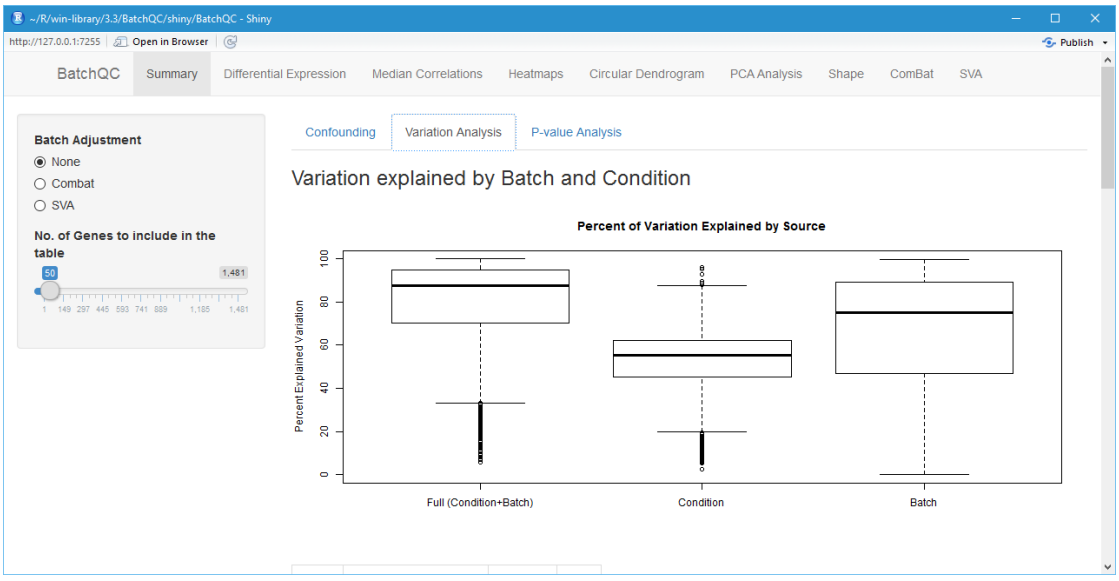



**Figure 29: Variation explained by Batch and Condition for the signature data: High overlap of variation explained by Condition and Batch**


**Figure 30: Distribution of batch effect p-value for the signature data: many more low p-values than expected by chance**

**Differential Expression:** In BatchQC, the interactive boxplots of the expression data enable the user to select whether to sort or color the samples by condition or batch, and the expression differences are easily identified. In the simulated data, after sorting by condition and coloring by batch, the box plot in Figure 31 clearly reveals the difference in mean and variance of the expression data across batches for each condition. BatchQC makes it easy to identify the effect of batch adjustment on the expression data. For the signature data, Figure 32 shows the expression data before and after batch adjustment using ComBat (which can be applied interactively using the "ComBat" tab). Interestingly, although the signature dataset shows only mild differences between batches removed by ComBat, other metrics (such as PCA) shown below provide a very different perspective (see Figure 39 below).

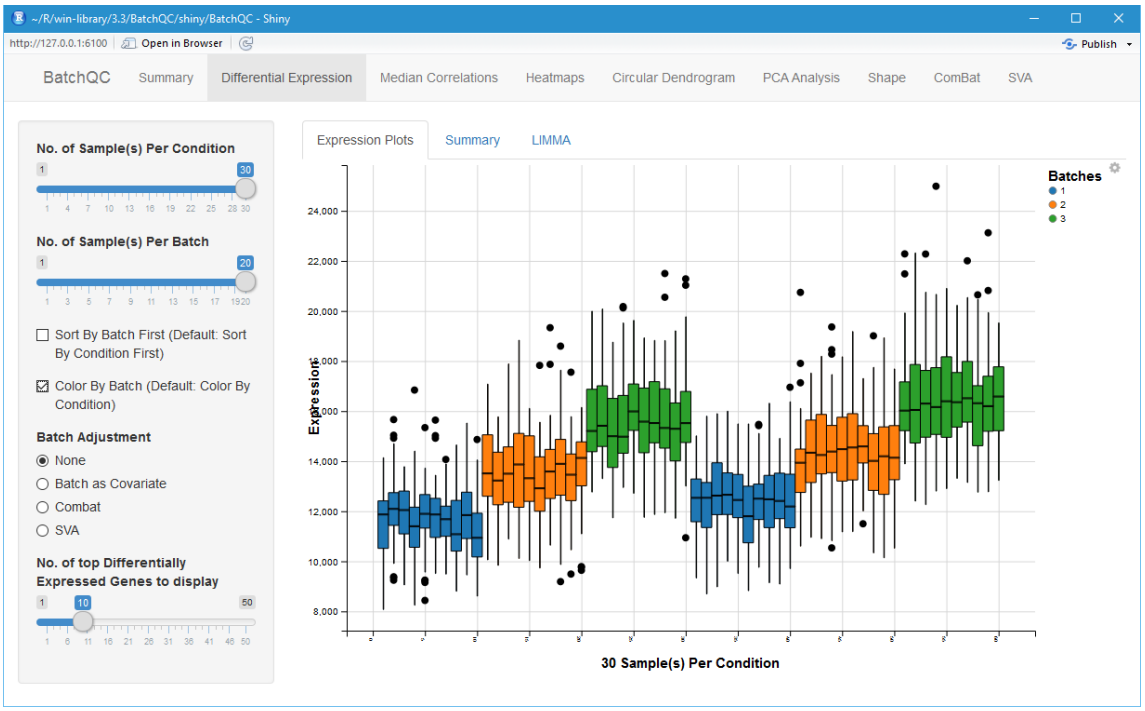



**Figure 31: Differential Expression of simulated dataset: Tooltip with Sample and Batch information as the user rolls the cursor over the plot**

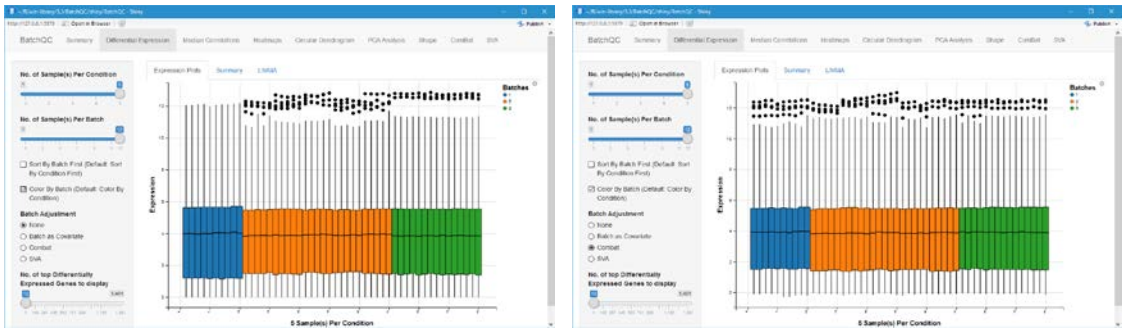
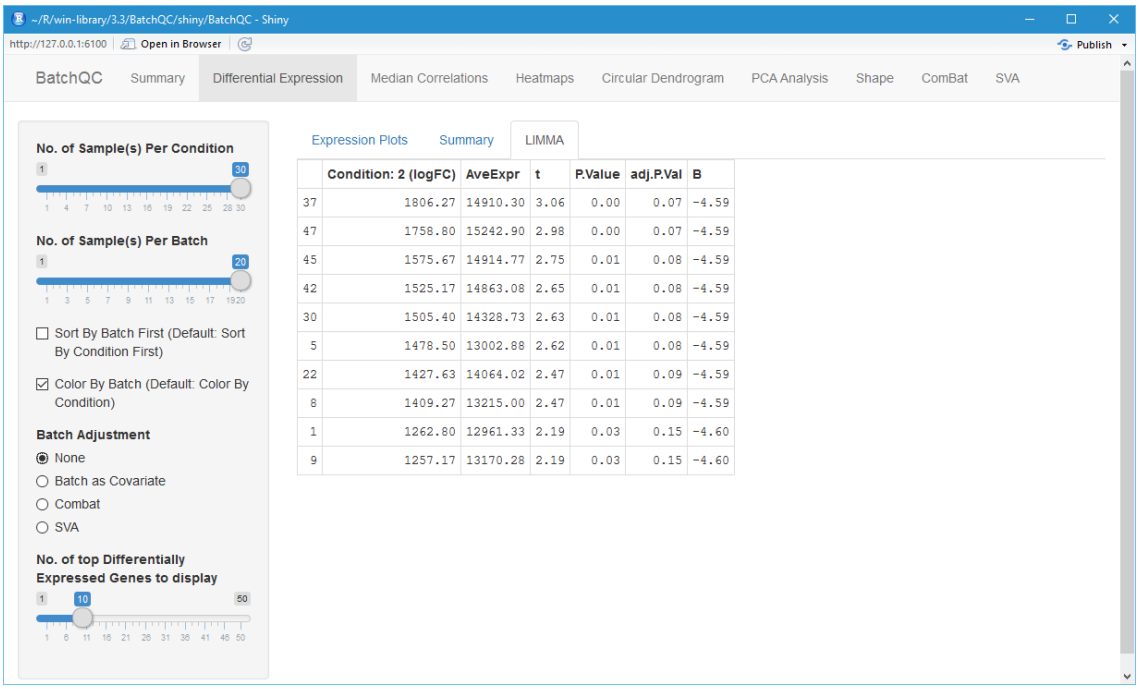
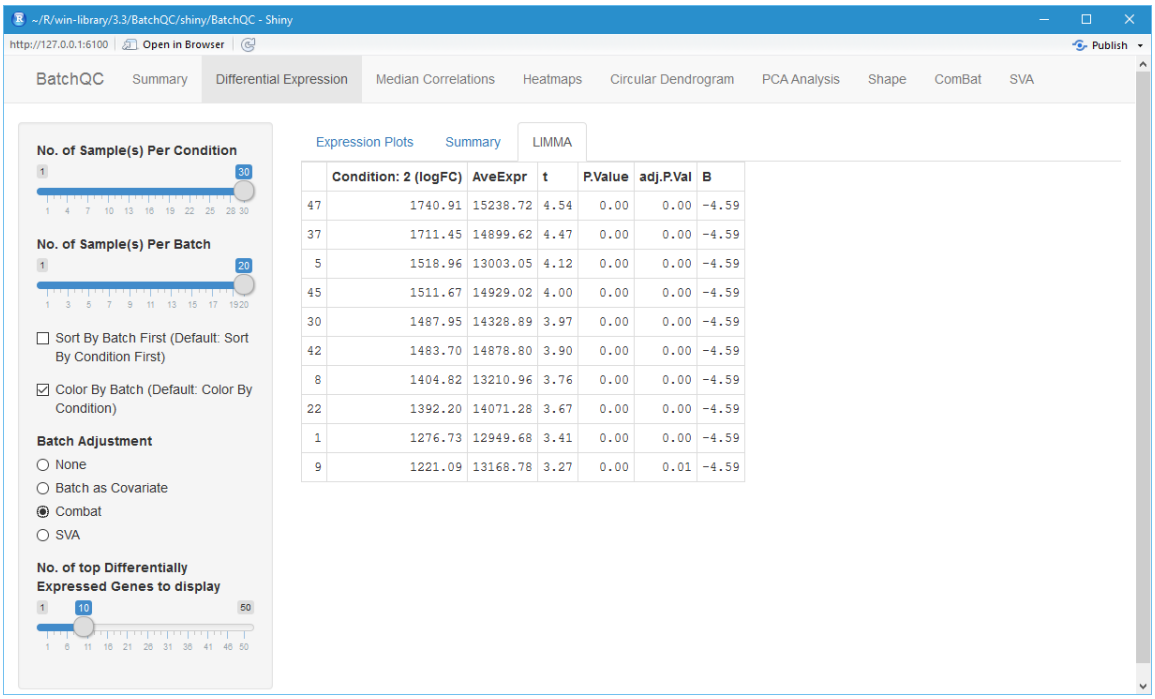


**Figure 32: Differential Expression of the signature data colored by batch before and after ComBat**




| | Condition: 2 (logFC) | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| 37 | 1806.27 | 14910.30 | 3.06 | 0.00 | 0.07 | -4.59 |
| 47 | 1758.80 | 15242.90 | 2.98 | 0.00 | 0.07 | -4.59 |
| 45 | 1575.67 | 14914.77 | 2.75 | 0.01 | 0.08 | -4.59 |
| 42 | 1525.17 | 14863.08 | 2.65 | 0.01 | 0.08 | -4.59 |
| 30 | 1505.40 | 14328.73 | 2.63 | 0.01 | 0.08 | -4.59 |
| 5 | 1478.50 | 13002.88 | 2.62 | 0.01 | 0.08 | -4.59 |
| 22 | 1427.63 | 14064.02 | 2.47 | 0.01 | 0.09 | -4.59 |
| 8 | 1409.27 | 13215.00 | 2.47 | 0.01 | 0.09 | -4.59 |
| 1 | 1262.80 | 12961.33 | 2.19 | 0.03 | 0.15 | -4.60 |
| 9 | 1257.17 | 13170.28 | 2.19 | 0.03 | 0.15 | -4.60 |

**Figure 33: Top Differentially Expressed Genes found by LIMMA for the simulated dataset**




| | Condition: 2 (logFC) | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| 47 | 1740.91 | 15238.72 | 4.54 | 0.00 | 0.00 | -4.59 |
| 37 | 1711.45 | 14899.62 | 4.47 | 0.00 | 0.00 | -4.59 |
| 5 | 1518.96 | 13003.05 | 4.12 | 0.00 | 0.00 | -4.59 |
| 45 | 1511.67 | 14929.02 | 4.00 | 0.00 | 0.00 | -4.59 |
| 30 | 1487.95 | 14328.89 | 3.97 | 0.00 | 0.00 | -4.59 |
| 42 | 1483.70 | 14878.80 | 3.90 | 0.00 | 0.00 | -4.59 |
| 8 | 1404.82 | 13210.96 | 3.76 | 0.00 | 0.00 | -4.59 |
| 22 | 1392.20 | 14071.28 | 3.67 | 0.00 | 0.00 | -4.59 |
| 1 | 1276.73 | 12949.68 | 3.41 | 0.00 | 0.00 | -4.59 |
| 9 | 1221.09 | 13168.78 | 3.27 | 0.00 | 0.01 | -4.59 |

**Figure 34: Top Differentially Expressed Genes found by LIMMA after batch adjustment by Combat**

BatchQC also performs differential expression analysis using LIMMA (5) and other methods to identify the top differentially expressed genes. BatchQC shows how the top differentially expressed genes are affected between (A) the unadjusted data, (B) after using batch as a covariate in a LIMMA model, or (C) after batch adjusting using ComBat or SVA. For the simulated data (Figure 33), gene 37 is the most differentially expressed gene before adjusting for batch with a log fold change of 1806.27 and an adjusted p-value of 0.07. However, after adjusting for batch using ComBat as shown in Figure 34, the most differentially expressed gene is gene 47 with a log fold change of 1740.91 and an adjusted p-value $< 0.01$ which is highly significant at 0.05 level. Similarly, there are also differences seen in the lower order of genes after adjusting using ComBat. Large changes in differentially expressed gene lists (increased/decreased significance) are strong indicators that a batch effect exists in the data and that some level of batch correction is needed.

**Median Correlations**

Finding the similarity or dissimilarity between samples is often essential to determine whether to exclude some outlier samples or whether batch effects impact the differential expression analysis of the data. We use the median correlation, or the median of the pairwise correlation of each samples with the other samples. In BatchQC, the median correlation plot, colored by batch, can be used to deduce whether there are outlier samples and whether batch effects affect the median correlation of samples. The median correlation plot for the simulated data does not show any obvious outlier samples, but the median correlation plot for the signature data (Figure 35) shows that batch 1 has a median

correlation that is very different from the other batches, indicating a potentially strong batch effect.

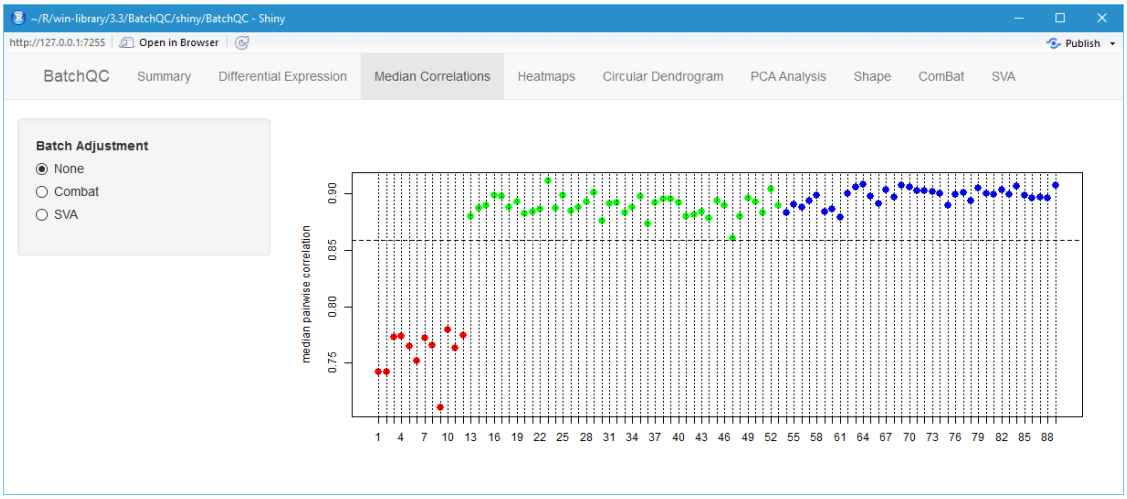



**Figure 35: Median correlation plot of the signature dataset**

### Heatmaps: Differential Expression Analysis

Heatmaps are a useful way to visually determine if there is differential expression of data along the condition of interest. Batch effects can either mask the differential expression of the condition of interest or spuriously exhibit differential expression when it is not there. For the simulated data (Figure 36), we see clustering by batch in the original data but after adjusting for batch by ComBat, we see the clustering along the condition of interest. BatchQC also displays a sample correlation heatmap to identify whether samples are correlated with one another. There is no sample correlation that is of a concern for the simulated dataset.

**Figure 36: Expression Heatmap of simulated dataset before and after ComBat**

**Circular Dendrogram**

A Circular Dendrogram provides another way to cluster samples in a circular fashion for easy identification of clustering based on batch and condition effects. BatchQC provides an option to display a Circular Dendrogram after batch adjustment using either ComBat or SVA methods. For the signature dataset (Figure 37), the samples are clustered perfectly by batch, but after batch adjustment using ComBat, the samples are clustered perfectly by condition (Figure 38). A similar result is also seen after batch adjustment using SVA. In the signature dataset, the circular dendrogram clearly illustrates the batch effect before batch adjustment (Figure 37), and the condition effect after batch adjustment (Figure 38).


**Figure 37: Circular Dendrogram for Signature dataset colored by batch**

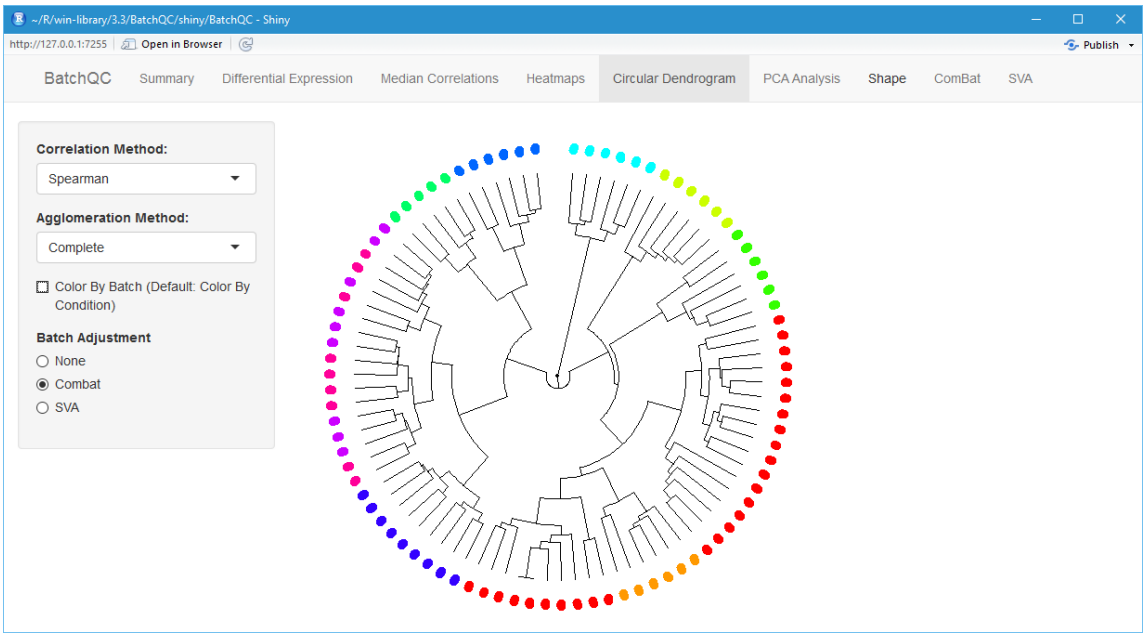



**Figure 38: Circular Dendrogram for Signature dataset colored by condition after ComBat**

## PCA Analysis

Principal Components Analysis (PCA) is frequently used to cluster samples and identify batch effects. BatchQC provides an option to select a principal component for the x-axis and another for the y-axis, and displays the selected components for each samples. The scatter plot is colored by either batch or condition for the easy identification of clusters. BatchQC also provides an option to display the principal component scatter plot after batch adjustment using ComBat or SVA adjusted data. The before and after PCA visualization enables the user to see whether batch adjustment helps in removing clustering that may exist based on batch. For example in the signature dataset (Figure 39), the clustering is based on the batches. After batch adjustment using ComBat (Figure 40), the clustering is not based on the batches. The Explained Variation subtab has a table (Figure 41) that lists the percentage variation explained by batch and condition combinations for each of the top principal components.

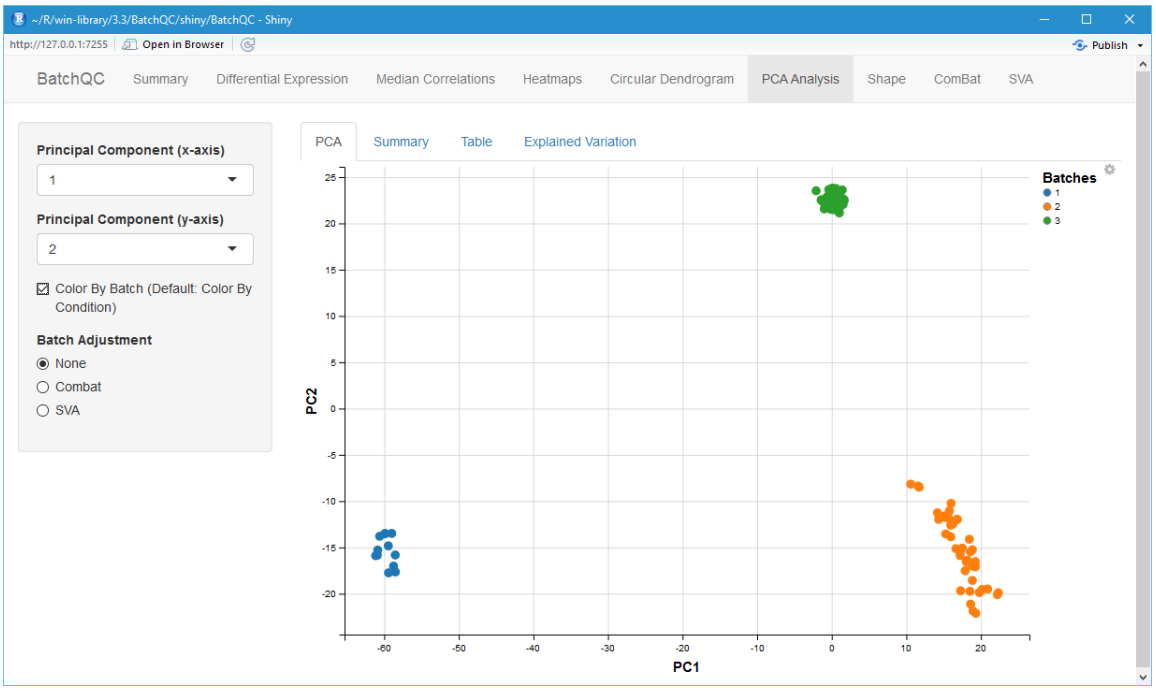
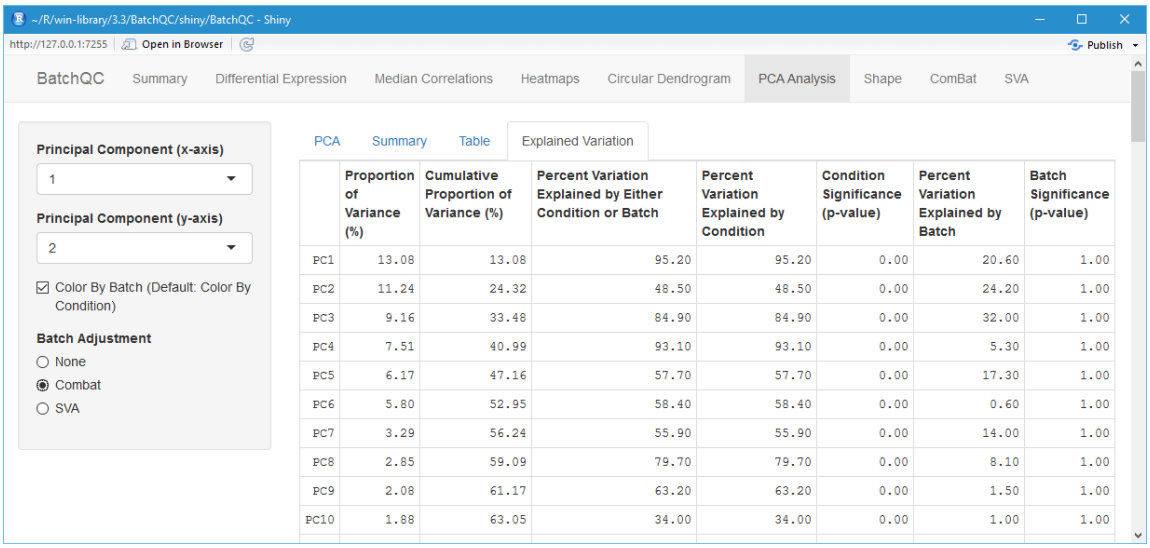



**Figure 39: PCA plot for signature dataset before batch adjustment**


**Figure 40: PCA plot for signature dataset after batch adjustment using Combat**




| | Proportion of Variance (%) | Cumulative Proportion of Variance (%) | Percent Variation Explained by Either Condition or Batch | Percent Variation Explained by Condition | Condition Significance (p-value) | Percent Variation Explained by Batch | Batch Significance (p-value) |
|---|---|---|---|---|---|---|---|
| PC1 | 13.08 | 13.08 | 95.20 | 95.20 | 0.00 | 20.60 | 1.00 |
| PC2 | 11.24 | 24.32 | 48.50 | 48.50 | 0.00 | 24.20 | 1.00 |
| PC3 | 9.16 | 33.48 | 84.90 | 84.90 | 0.00 | 32.00 | 1.00 |
| PC4 | 7.51 | 40.99 | 93.10 | 93.10 | 0.00 | 5.30 | 1.00 |
| PC5 | 6.17 | 47.16 | 57.70 | 57.70 | 0.00 | 17.30 | 1.00 |
| PC6 | 5.80 | 52.95 | 58.40 | 58.40 | 0.00 | 0.60 | 1.00 |
| PC7 | 3.29 | 56.24 | 55.90 | 55.90 | 0.00 | 14.00 | 1.00 |
| PC8 | 2.85 | 59.09 | 79.70 | 79.70 | 0.00 | 8.10 | 1.00 |
| PC9 | 2.08 | 61.17 | 63.20 | 63.20 | 0.00 | 1.50 | 1.00 |
| PC10 | 1.88 | 63.05 | 34.00 | 34.00 | 0.00 | 1.00 | 1.00 |

**Figure 41: Principal Components Explained Variation for signature dataset**

**Shape: Expression distribution shape analysis**

The shape analysis is used to identify the distribution shape of the batch variations of the expression data. Any distribution can be characterized by the mean, variance, and higher order moments. If batch effects exist, it is very useful to identify whether mean, variance or higher order moments vary between the batches. The subtab Batch Variation (Figure 42) displays the mean, variance, skewness, and kurtosis of the expression distribution across genes for all of the samples. The batch effect variation for mean and variance is highly significant ($p<0.0001$), but Figure 43 illustrates that after batch adjustment using ComBat, the batch effect variations for the mean and variance are no longer significant, with gene-wise comparison p-values of 0.9999 and 0.7298, respectively.

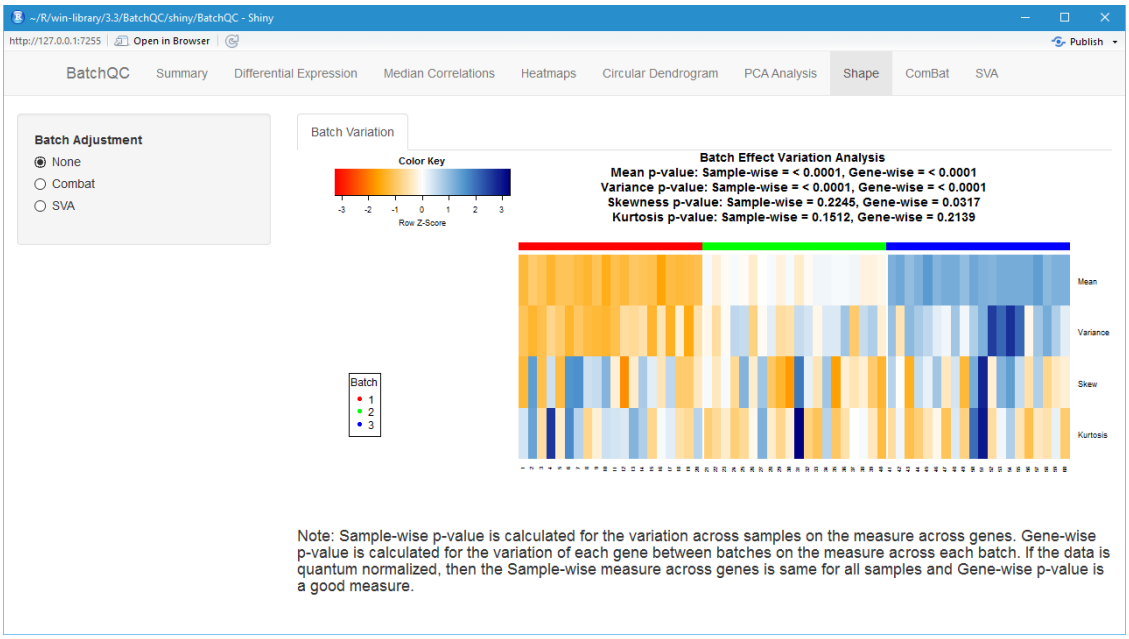



**Figure 42: Shape Batch Variation for simulated dataset**

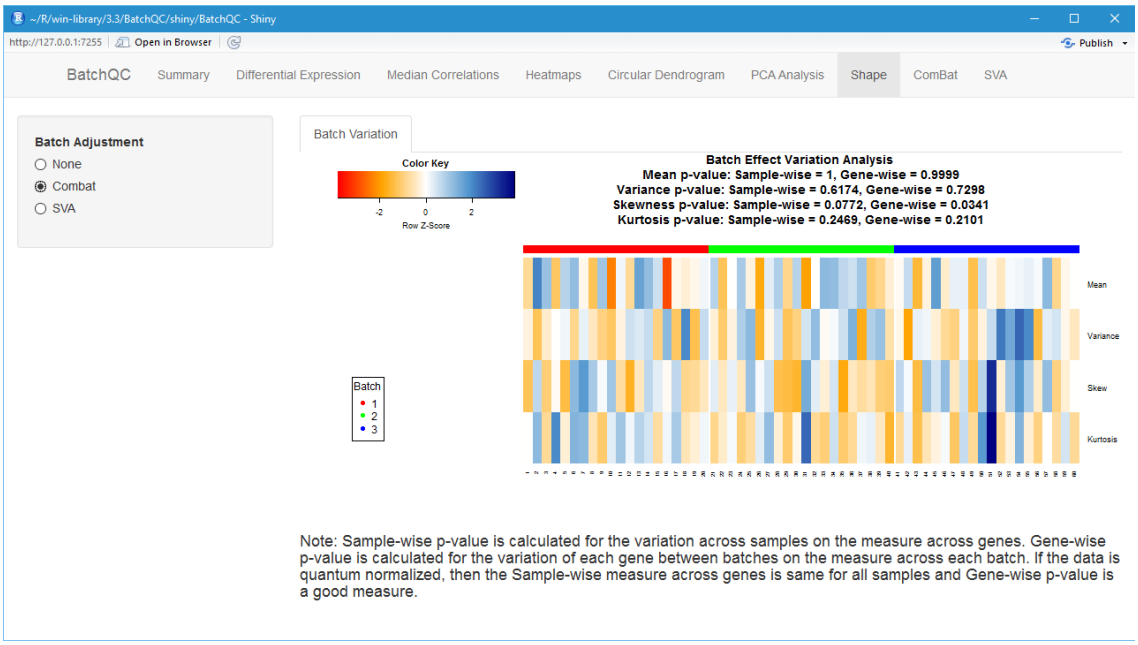



**Figure 43: Shape Batch Variation for simulated dataset after batch adjustment by Combat**

**ComBat: Batch Adjustment**

An advantageous feature of BatchQC is the ability to interactively apply the ComBat (1) batch adjustment approach to the data. The user can compare the ComBat adjusted data with the raw data in all of the diagnostics mentioned above after performing the batch adjustment by clicking the 'Run ComBat' button. In addition, this tab contains subtabs to evaluate the optimal ComBat parameter values for adjusting the data. Under the ComBat Plots subtab, the distributions of the mean and variance of the batch effects for the selected batch are plotted under the ComBat parametric curve assumption as well as an empirical distribution. Under the parametric curve assumption, ComBat assumes that the mean follows a normal distribution and variance follows an inverse gamma distribution. The density curves and Q-Q plots comparing the normal parametric assumption with the empirical distribution curve helps to decide which batch adjustment is most suitable for the given data. In Figure 44, the density and QQ plots on the top correspond to the batch mean and the bottom plots correspond to the batch variance, and since the parametric and non-parametric curves are not very different, the parametric assumption is sufficient for the simulated dataset. There is also a summary measure of this comparison of curves using a formal Kolmogorov-Smirnov test listed under the Summary subtab, and we see that for the simulated dataset the p-value is not significant, implying parametric assumption is sufficient.

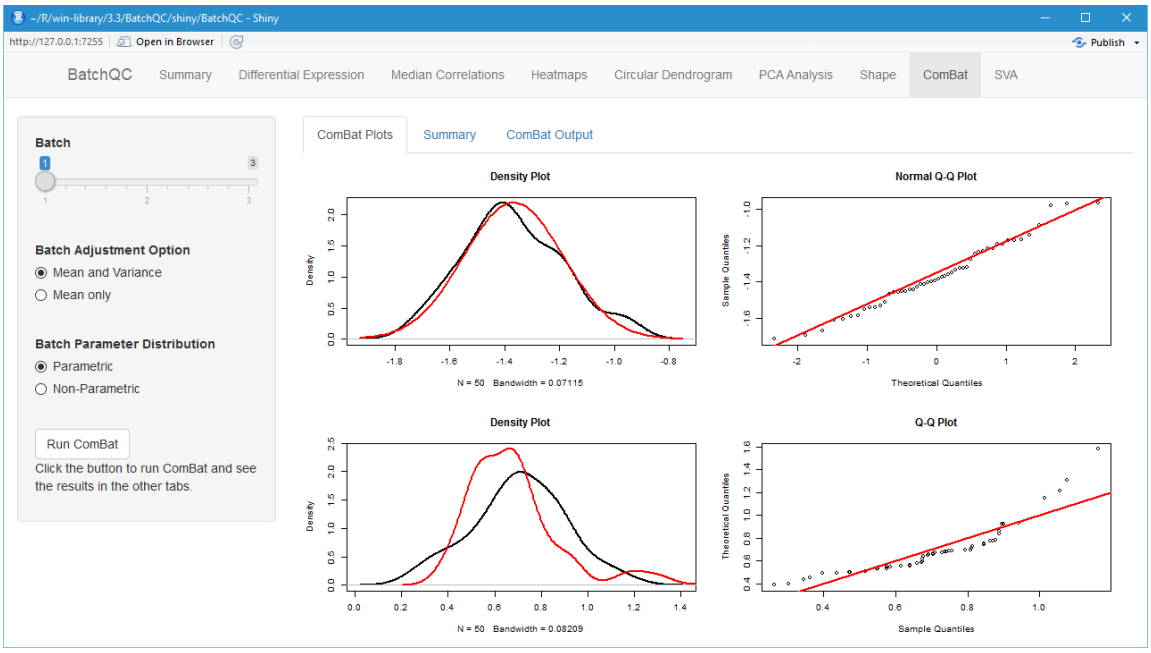



**Figure 44: ComBat diagnostics plots for the simulated dataset**

## SVA: Surrogate Variable Analysis

BatchQC also allows the user to apply Surrogate Variable Analysis (SVA) (2) to account for both known and unknown batch variables. BatchQC automatically uses a permutation-based approach to estimate the number of surrogate variables found in the data under the Summary subtab. As with ComBat, there is a 'Run SVA' button with an option to choose between the Regression Adjusted (default) and Frozen SVA options. After SVA is completed, the user can compare the SVA adjusted data with the raw and ComBat data under all of the diagnostic tabs discussed above.

## Conclusion

When performing multi-dimensional and differential expression analysis using genomic data, often times they are affected by technical variation attributable to both

observed and unobserved factors (Leek et al., 2010). Hence, batch effect is an important factor in the analysis of genomic data. When conducting genomic data analysis in two-step by first adjusting for batch using batch adjustment tools and proceeding with the downstream analysis often leads to an exaggerated significance problem. We developed an approach where the user can first use tools such as ComBat for batch adjustment and perform differential expression analysis in two-step and yet avoid the exaggerated significance problem. We have developed a toolkit called BatchQC that can help to identify whether batch effects are present in the data and adjust for batch effects in a suitable way. BatchQC is available for download as a shiny app R-package from Bioconductor at http://bioconductor.org/packages/BatchQC/. We hope you find this toolkit very useful when you are analyzing your data for batch effects.

# CHAPTER FIVE

## Conclusion

We have described three projects here, which together provide a complete set of toolkits and methodology necessary for analysis of genomic data for these applications. In the first project, we introduced methods and software for complete metagenomic analysis. We developed a complete software pipeline called PathoScope to identify microbes and pathogens to the strain level using a Bayesian mixture model with a modified pseudo likelihood model. We evaluated the accuracy of the Bayesian mixture modeling approach in comparison to other methods and also evaluated how prior information in the Bayesian mixture modeling improves these estimates. We performed a simulation study to evaluate the read coverage needed to estimate pathogen proportions to a given confidence limit. Based on this study, we recommend that for single strain identification, about 0.1X coverage of reads is sufficient to get more than 99% accuracy using PathoScope. PathoScope 2.0 is available as a python module for download from the sourceforge at https://sourceforge.net/projects/pathoscope/.

In the second project, we introduced a toolkit for microbiome variation analysis called PathoStat. We have developed a methodology for computing confidence region for the relative abundance estimates of the microbes in a sample and a module for displaying it. We have performed differential abundance analysis on a diet study dataset and used that as an example for design and development of the PathoStat toolkit. PathoStat provides a rich set of visualization modules. Some of the salient features are relative

abundance charts, diversity estimates and plots, tests of differential abundance and multi-dimensional analysis including principal component and principal coordinate analysis. The important feature of the package is the interactive feature of all the plots, allowing the user to choose various parameters and variables of interest and visualize all the dynamically generated plots customized according to the user selected criteria. The toolkit is structured so that new modules can be easily added in the future. PathoStat is developed as a shiny app R-package and is available for download from Bioconductor at http://bioconductor.org/packages/PathoStat. We hope that you will find this toolkit very useful when you want to analyze microbiome data.

When performing multi-dimensional and differential expression analysis using genomic data, often times they are affected by technical variation attributable to both observed and unobserved factors (Leek et al., 2010). We developed an approach where the user can first use tools such as ComBat for batch adjustment and perform differential expression analysis in two-step and yet avoid the exaggerated significance problem. We have developed a toolkit called BatchQC that can help to identify whether batch effects are present in the data and adjust for batch effects in a suitable way. BatchQC is available for download as a shiny app R-package from Bioconductor at http://bioconductor.org/packages/BatchQC/. We hope you find this toolkit very useful when you are analyzing your data for batch effects.

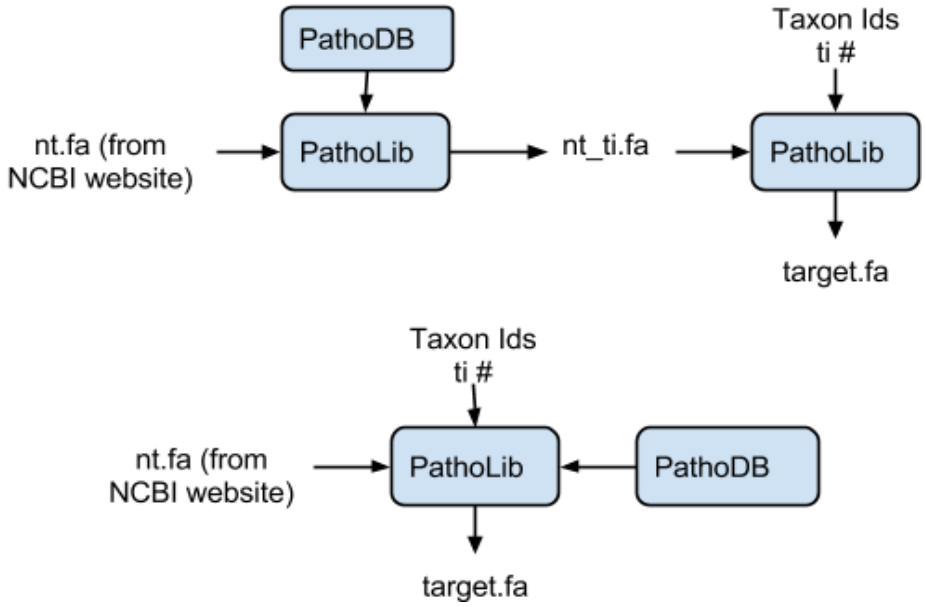
# APPENDIX

## PathoScope2 Design

## PathoLib

PathoLib prepares the target, host and other genome reference libraries of interest from the complete NT fasta file. The user inputs the taxonomy ids of interest and the PathoLib module creates a target fasta file corresponding to the given taxonomy ids from the NT fasta file.

### *Flowchart*




### *Taxonomy appended nt_ti.fa format:*

ti|<taxonomy id>|org|<organism name of the taxonomy id>|[gi|...same as in original fasta]

### *Dependency*

pathoscope.utils.seqParse

pathoscope.utils.pathoUtils

pathoscope.pathodb.dbUtils

### *Inputs:*

**Case 1: DB User information provided**

Required:

1. DB
2. DB Host
3. DB User
4. DB Passwd
5. NT file (downloaded from ftp://ftp.ncbi.nih.gov/blast/db/FASTA/nt.gz)
6. Output prefix for target

Optional:

1. DB Port (default 3306)
2. Taxon ids (Restrict to the give taxon ids)
3. Subtax flag (When present will include all taxonomy ids in the tree that comes under the list of taxon ids given above.
4. inputArgs.lib_nodesc (No description text will be appended in the fasta file that is created)

### *Outputs:*

1. *-ti_val.fa
2. *-ti_inval.fa

***Example command-line to create nt_ti_val.fa and nt_ti_inval.fa:***

pathoscope.py LIB -g /home/mani/work/data/nt_database/nt -dbhost localhost -dbuser pathoscope -dbpasswd johnsonlab -db pathodb -o nt

**Case 2: DB User information not provided**

Required:

1. NT file appended with ti information through the above case (nt_ti_val.fa)
2. Output prefix for target

Optional:

1. Taxon ids (Restrict to the give taxon ids)
2. Subtax flag (When present will include all taxonomy ids in the tree that comes under the list of taxon ids given above.
3. inputArgs.lib_nodesc (No description text will be appended in the fasta file that is created)

*Outputs:*

1. *-ti_val.fa
2. *-ti_inval.fa

***Example command-line option to create a reference library:***

pathoscope.py LIB -g /protected/projects/johnsonlab/data/innocentive_meta/nt_database/nt_ti_val.fa -t 7157 --subtax -o mosquitos

*Python functions:*

**pathoLib.append_ti_into_fasta_app(()**

*Input*:

inputArgs.lib_reference,

taxon_ids,

inputArgs.lib_subtax,

MysqlConf,

not(inputArgs.lib_nodesc),

inputArgs.lib_online_search,

inputArgs.lib_outprefix

*Output*:

*-ti_val.fa

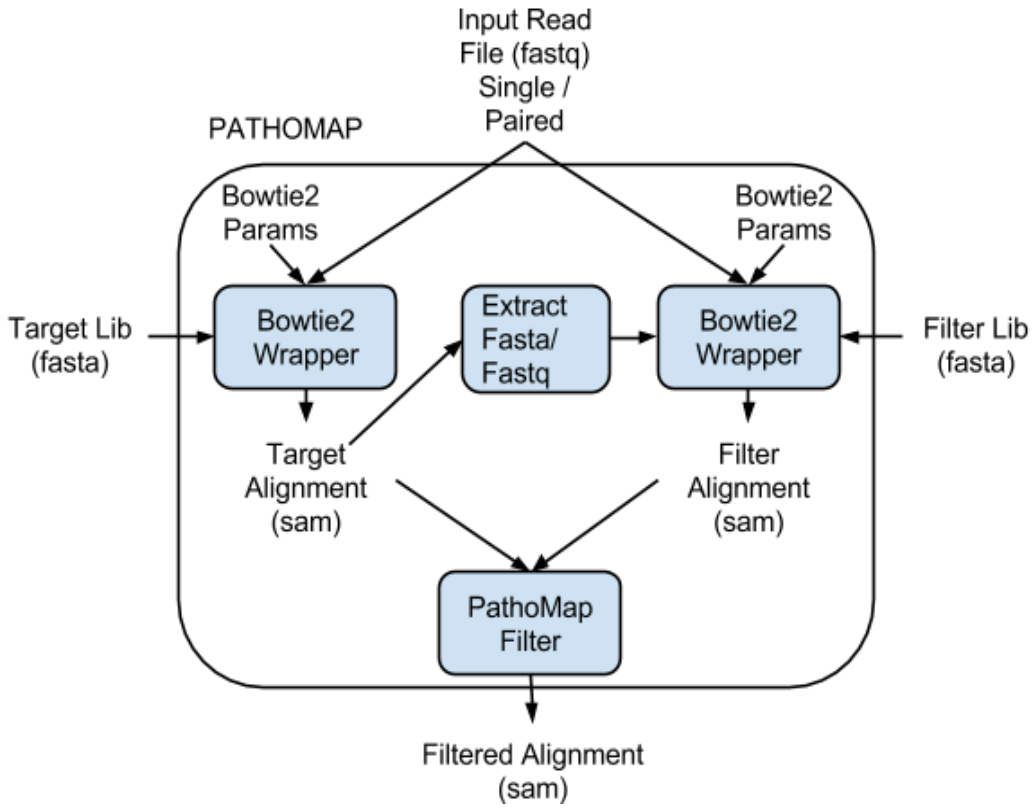
*-ti_inval.fa

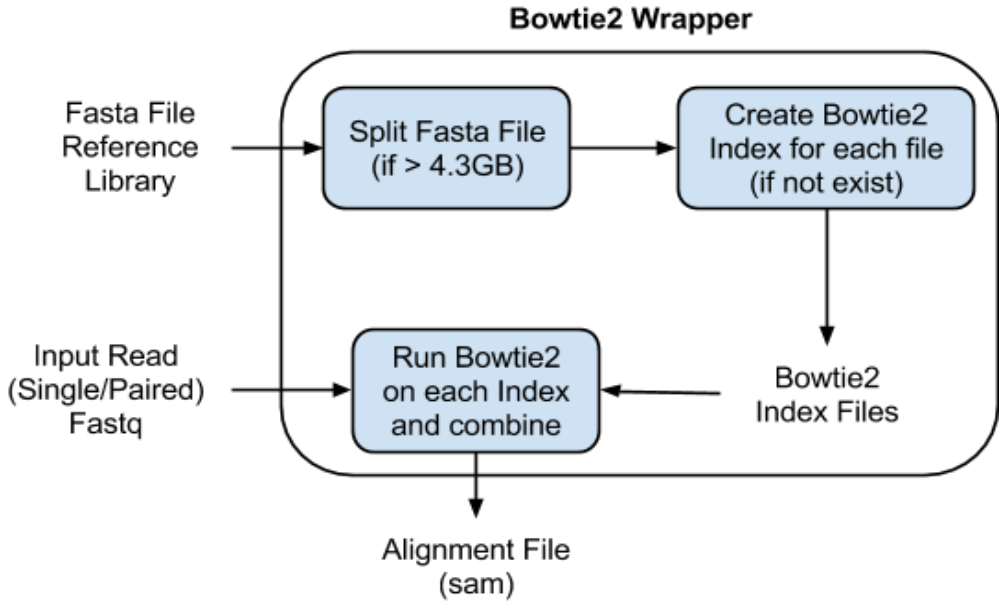
---

## PathoMap

PathoMap performs the alignment through wrappers for each type of aligners. It chooses the proper alignment parameters for the given input read fastq file taking into consideration the sequencing platform. It also splits the reference file into smaller files and combines the result at the end, if it exceeds the indexing limit of the aligner. Currently, there is a wrapper that is developed for Bowtie2 and wrappers for other aligners can be developed similarly later. PathoMap also performs the filtering from the target alignment file, all those reads that map to the filter alignment file. The result is one filtered alignment file that can be given as input to PathoID for processing.

### *Flowchart*


Input Read
File (fastq)
Single /
Paired
PATHOMAP
Bowtie2
Params
Bowtie2
Params
Target Lib
(fasta)
Bowtie2
Wrapper
Extract
Fasta/
Fastq
Bowtie2
Wrapper
Filter Lib
(fasta)
Target
Alignment
(sam)
Filter
Alignment
(sam)
PathoMap
Filter
Filtered Alignment
(sam)


Bowtie2 Wrapper
Fasta File
Reference
Library
Split Fasta File
(if > 4.3GB)
Create Bowtie2
Index for each file
(if not exist)
Input Read
(Single/Paired)
Fastq
Run Bowtie2
on each Index
and combine
Bowtie2
Index Files
Alignment File
(sam)

*Dependency:*

pathoscope.utils.pathoUtils

*Inputs:*
Required:

1. Target lib fasta
2. Input Read File (Fastq - single/paired end)

Optional:

1. Filter lib fasta
2. Bowtie2 params for Target Lib
3. Bowtie2 params for Filter Lib

*Outputs:*
Required:

1. Filtered Alignment file

Optional:

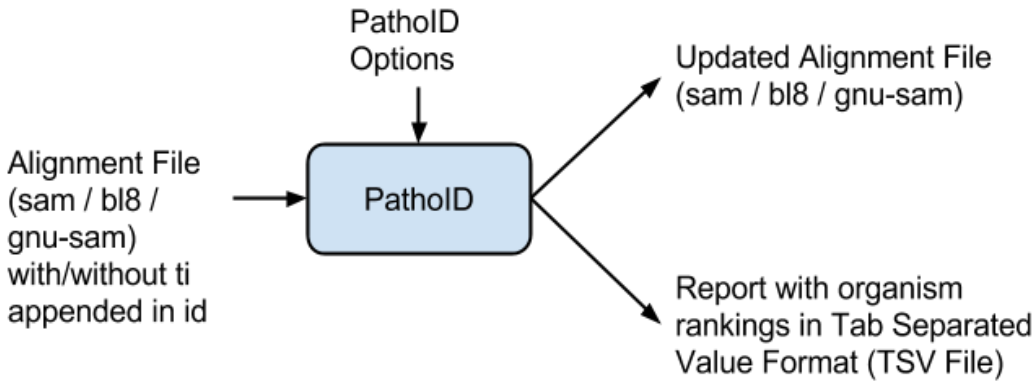
1. Keep Intermediate Split and Alignment Index Files

---

## PathoID

PathoID takes as input an alignment file in sam/blast(bl8)/gnu-sam format, which may or may not have the reference ID appended with taxon id, and produces an updated alignment file and a report with organism rankings in a Tab Separated Value Format (TSV) that can be opened in Excel. When the taxon id (ti) information is present, pathoscope will group together based on the ti and run the pathoscope reassignment algorithm to produce the results.

*Flowchart*




*Dependency:*
pathoscope.utils.pathoUtils

*Inputs:*
Required:

1. Alignment file path

Optional:

1. Alignment file format (sam/bl8/gnu-sam)
2. Output directory (where the report and updated alignment file is generated)
3. Experiment tag (File prefix for the report and other files that are generated for easy identification)
4. EM Algorithm maximum iterations
5. EM Algorithm Epsilon cutoff
6. Score Cutoff (Minimum alignment score required below which the alignment record will be dropped)
7. Output alignment matrix flag (When present outputs *-genomeId.txt and *-readId.txt files)

### *Outputs:*

Required:

1. TSV (Tab Separated Value) report; *-report.tsv file
2. Initial and Final Guess files; *-initGuess.txt, *-finGuess.txt files

Optional:

1. Updated alignment file; updated_<alignment file>
2. Output alignment matrix files; *-genomeId.txt and *-readId.txt files

### *Example command-line option to run pathoid reassignment:*

pathoscope.py -verbose ID -t=sam -f=/unprotected/projects/johnsonlab/exp/pathoscope2/mani/testset/mosquito/8_25_CTTGTA_L001.sam -o -s=.01 -e=Mosq1 -outdir=/unprotected/projects/johnsonlab/exp/pathoscope2/mani/testset/mosquito

### *Python functions:*

**PathoIdOptions:**

| | |
|---|---|
| ali_file = "" | : Alignment File (Required) |
| verbose = False | : Verbose flag to print additional information during execution |
| score_cutoff = 0.01 | : Score cutoff below which the alignment record will not be included |
| exp_tag = "" | : Experiment tag - Output file prefix for easy identification |
| ali_format = "sam" | : Alignment format (Default: sam) |
| outdir = "" | : Output directory where reports and all output files are created |

emEpsilon = 0.01 : EM Algorithm epsilon cutoff - the change below which the iteration stops

maxIter = 50 : Maximum number of iterations to run EM algorithm

out_matrix_flag = True : Creates *-genomeId.txt and *-readId.txt files, when True

noalign = False : No updated alignment file will be created when this flag is set to True

**PathoID.pathoscope_reassign()**

*Input:*
PathoIdOptions mentioned above

*Output:*
finalReport (tsv file)

reAlignfile (updated alignment file)

*Returns:*
(finalReport, x2, x3, x4, x5, x1, x6, x7, x8, x9, x10, x11, reAlignfile)

x2: Genome

x3: Initial Guess

x4: Initial Best Hit

x5: Initial Best Hit Read Numbers

x1: Final Guess

x6: Final Best Hit

x7: Final Best Hit Read Numbers

x8: Initial High Confidence Hits

x9: Initial Low Confidence Hits

x10: Final High Confidence Hits

x11: Final Low Confidence Hits

**Functions called by PathoID.pathoscope_reassign():**

**PathoID.conv_align2GRmat()**

Finds the Genome Read alignment matrix from the given alignment file in the following format given in the output below.

Input:

ali_file : Input alignment file

scoreCutoff : Score cutoff below which the alignment record will not be included

aliFormat : Alignment format (sam/gnu-sam/bl8)

Output:

U : Unique Read Index to Genome Index hash mapping

NU : Non-Unique Read Index to custom list hash mapping. The custom list has 3 components. The first one is a list of Genome Indices. The second one is a list of score (float) of the mapping. The third one is a list of normalized score in integer of the mapping.

genomes : List of Genome Ids.

reads : List of Read Ids.

**PathoID.pathoscope_em()**

Runs the PathoScope EM Algorithm and returns the result from the pathoscope EM Algorithm reassignment.

Input:

U : Unique Read Index to Genome Index hash mapping

NU : Non-Unique Read Index to custom list hash mapping. The custom list has 3 components. The first one is a list of Genome Indices. The second one is a list of score (float) of the mapping. The third one is a list of normalized score in integer of the mapping.

genomes : List of Genome Ids.

maxIter : Maximum number of iterations to run the EM Algorithm

emEpsilon : EM Algorithm epsilon value to check for change and stop

verbose : Flag to display verbose information of the execution steps

Output:

initPi, pi, theta, NU

initPi : Initial value of the genome proportions

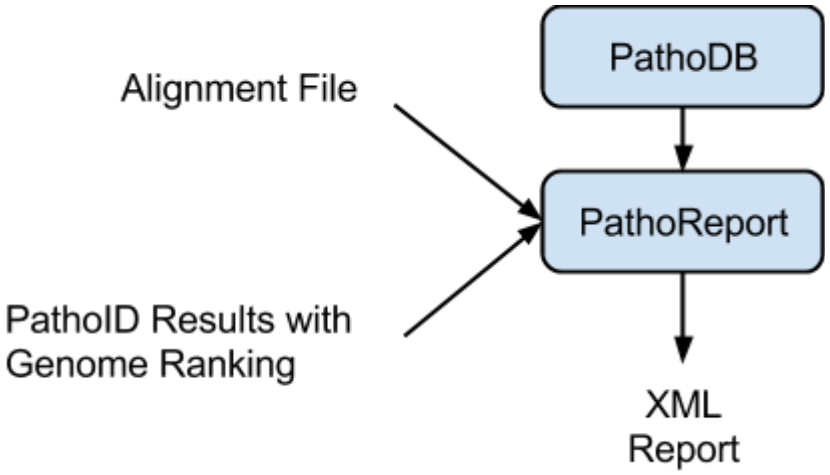
pi : Final value of the genome proportions after reassignment.

theta : Final theta value of the EM algorithm.

NU : Updated NU after EM algorithm reassignment.

## Pathoreport

*Flowchart*




*Dependency:*
pathoscope.utils.samUtils
pathoscope.pathodb.dbUtils

*Inputs:*
Required:
1. Alignment file path
2. mySQL Configuration

Optional:
1. PathoID.pathoscope_reassign() return elements
2. PathoScope2 run parameters and other information to be included in the report
3. Output directory (where the XML report will be generated)

*Python functions:*
**pathoreport.xmlReport.writePathoXML()**

*Input:*
Alignment File
mySQL Configuration
output File name

*Output:*
XML Report

### *XML Report Specification:*

<Organisms>
[List of Organism]
<Organism>
    <relativeAmount>
    <taxonomy> (including lineage information)
    <organismName>
    <genus>
    <species>
    <strain>
    <genes>
    <reads>
    <contigs>

---

## BIBLIOGRAPHY

Aickin, M. & Gensler, H. 1996. Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. American Journal of Public Health, 86(5):726-728.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17):3389-3402.

Benjamini, Y. & Hochberg, Y. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society. Series B (Methodological), 57(1):289-300.

Bhaduri, A., Qu, K., Lee, C.S., Ungewickell, A. & Khavari, P.A. 2012. Rapid identification of non-human sequences in high-throughput sequencing datasets. Bioinformatics, 28(8):1174-1175.

Blaxter, M., Mann, J., Chapman, T., Thomas, F., Whitton, C., Floyd, R. & Abebe, E. 2005. Defining operational taxonomic units using DNA barcode data. Philosophical Transactions of the Royal Society B: Biological Sciences, 360(1462):1935-1943.

Brady, A. & Salzberg, S.L. 2009. Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. Nature Methods, 6(9):673-676.

Bray, J.R. & Curtis, J.T. 1957. An Ordination of the Upland Forest Communities of Southern Wisconsin. Ecological Monographs, 27(4):325-349.

Byrd, A.L., Perez-Rogers, J.F., Manimaran, S., Castro-Nallar, E., Toma, I., McCaffrey, T., Siegel, M., Benson, G., Crandall, K.A. & Johnson, W.E. 2014. Clinical PathoScope: rapid alignment and filtration for accurate pathogen identification in clinical samples using unassembled sequencing data. BMC Bioinformatics, 15:262.

Castro-Nallar, E., Shen, Y., Freishtat, R.J., Perez-Losada, M., Manimaran, S., Liu, G., Johnson, W.E. & Crandall, K.A. 2015. Integrating microbial and host

transcriptomics to characterize asthma-associated microbial communities. BMC Medical Genomics, 8:50.

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M.W., Gaffney, D.J., Elo, L.L., Zhang, X. & Mortazavi, A. 2016. A survey of best practices for RNA-seq data analysis. Genome Biology, 17(1):13.

Dempster, A.P.L., N.M.; Rubin, D.B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1-38.

Dereniowski, D. & Kubale, M. 2004. Cholesky Factorization of Matrices in Parallel and Ranking of Graphs. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M. & Waśniewski, J. (eds.). Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, Czestochowa, Poland, September 7-10, 2003. Revised Papers. Berlin, Heidelberg: Springer Berlin Heidelberg.

Dunn, O.J. 1961. Multiple Comparisons among Means. Journal of the American Statistical Association, 56(293):52-64.

Durbin, R., Eddy, S.R., Krogh, A. & Mitchison, G. 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

Francis, O.E., Bendall, M., Manimaran, S., Hong, C., Clement, N.L., Castro-Nallar, E., Snell, Q., Schaalje, G.B., Clement, M.J., Crandall, K.A. & Johnson, W.E. 2013. Pathoscope: Species identification and strain attribution with unassembled sequencing data. Genome Research, 23(10):1721-1729.

Frank, C., Werber, D., Cramer, J.P., Askar, M., Faber, M., an der Heiden, M., Bernard, H., Fruth, A., Prager, R., Spode, A., Wadl, M., Zoufaly, A., Jordan, S., Kemper, M.J., Follin, P., Muller, L., King, L.A., Rosner, B., Buchholz, U., Stark, K. & Krause, G. 2011. Epidemic profile of Shiga-toxin-producing Escherichia coli O104:H4 outbreak in Germany. New England J Medicine, 365(19):1771-1780.

Gerlach, W. & Stoye, J. 2011. Taxonomic classification of metagenomic shotgun sequences with CARMA3. Nucleic Acids Research, 39(14):e91.

Grabherr, M.G., Haas, B.J., Yassour, M., Levin, J.Z., Thompson, D.A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B.W., Nusbaum, C., Lindblad-Toh, K., Friedman, N. & Regev, A. 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nature Biotechnology, 29(7):644-652.

Harnett, J., Myers, S.P. & Rolfe, M. 2016. Probiotics and the Microbiome in Celiac Disease: A Randomised Controlled Trial. Evidence-Based Complementary and Alternative Medicine, 2016:9048574.

Hastie, T., Tibshirani, R. & Friedman, J.H. 2009. The elements of statistical learning : data mining, inference, and prediction, 2nd. New York, NY: Springer.

Holtgrewe, M. 2010. Mason: A Read Simulator for Second Generation Sequencing Data. Fachbereich Mathematik und Infromatik. Berlin: Freie Universitat Berlin.

Hong, C., Manimaran, S. & Johnson, W.E. 2014a. PathoQC: Computationally Efficient Read Preprocessing and Quality Control for High-Throughput Sequencing Data Sets. Cancer Informatics, 13(Suppl 1):167-176.

Hong, C., Manimaran, S., Shen, Y., Perez-Rogers, J.F., Byrd, A.L., Castro-Nallar, E., Crandall, K.A. & Johnson, W.E. 2014b. PathoScope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples. Microbiome, 2:33.

Huson, D.H., Auch, A.F., Qi, J. & Schuster, S.C. 2007. MEGAN analysis of metagenomic data. Genome Research, 17(3):377-386.

Johnson, W.E., Li, C. & Rabinovic, A. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics, 8(1):118-127.

Jolliffe, I.T. & Cadima, J. 2016. Principal component analysis: a review and recent developments. Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences, 374(2065):20150202.

Jost, L. 2006. Entropy and diversity. Oikos, 113(2):363-375.

Kostic, A.D., Ojesina, A.I., Pedamallu, C.S., Jung, J., Verhaak, R.G., Getz, G. & Meyerson, M. 2011. PathSeq: software to identify or discover microbes by deep sequencing of human tissue. Nature Biotechnology, 29(5):393-396.

Kuczynski, J., Stombaugh, J., Walters, W.A., Gonzalez, A., Caporaso, J.G. & Knight, R. 2012. Using QIIME to analyze 16S rRNA gene sequences from microbial communities. Current Protocols in Microbiology, Chapter 1:Unit 1E 5.

Langmead, B. & Salzberg, S.L. 2012. Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4):357-359.

Leek, J.T. 2016. bladderbatch: Bladder gene expression data illustrating batch effects.

Leek, J.T., Scharpf, R.B., Bravo, H.C., Simcha, D., Langmead, B., Johnson, W.E., Geman, D., Baggerly, K. & Irizarry, R.A. 2010. Tackling the widespread and critical impact of batch effects in high-throughput data. Nature Reviews. Genetics, 1(10):733-739.

Leonard, M.M. & Fasano, A. 2016. The microbiome as a possible target to prevent celiac disease. Expert Review of Gastroenterology & Hepatology, 10(5):555-556.

Levine, D.M. 1977. Nonmetric multidimensional scaling and hierarchical clustering: procedures for the investigation of the perception of sports. Res Q, 48(2):341-348.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G. & Durbin, R. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics, 25(16):2078-2079.

Lozupone, C.A., Hamady, M., Kelley, S.T. & Knight, R. 2007. Quantitative and qualitative beta diversity measures lead to different insights into factors that structure microbial communities. Applied and Environmental Microbiology, 73(5):1576-1585.

Manimaran, S., Selby, H.M., Okrah, K., Ruberman, C., Leek, J.T., Quackenbush, J., Haibe-Kains, B., Bravo, H.C. & Johnson, W.E. 2016. BatchQC: interactive software for evaluating sample and batch effects in genomic data. Bioinformatics.

McGee, V.E. 1968. Multidimensional Scaling Of N Sets Of Similarity Measures: A Nonmetric Individual Differences Approach. Multivariate Behavioral Research, 3(2):233-248.

McHardy, A.C., Martin, H.G., Tsirigos, A., Hugenholtz, P. & Rigoutsos, I. 2007. Accurate phylogenetic classification of variable-length DNA fragments. Nature Methods, 4(1):63-72.

Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E.M., Kubal, M., Paczian, T., Rodriguez, A., Stevens, R., Wilke, A., Wilkening, J. & Edwards, R.A. 2008. The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC Bioinformatics, 9:386.

Monzoorul Haque, M., Ghosh, T.S., Komanduri, D. & Mande, S.S. 2009. SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. Bioinformatics, 25(14):1722-1730.

Morris, E.K., Caruso, T., Buscot, F., Fischer, M., Hancock, C., Maier, T.S., Meiners, T., Müller, C., Obermaier, E., Prati, D., Socher, S.A., Sonnemann, I., Wäschke, N., Wubet, T., Wurst, S. & Rillig, M.C. 2014. Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories. Ecology and Evolution, 4(18):3514-3524.

Murdoch, D.J., Tsai, Y.-L. & Adcock, J. 2008. P-Values are Random Variables. The American Statistician, 62(3):242-245.

Naeem, R., Rashid, M. & Pain, A. 2013. READSCAN: a fast and scalable pathogen discovery program with accurate genome relative abundance estimation. Bioinformatics, 29(3):391-392.

Navas-Molina, J.A., Peralta-Sanchez, J.M., Gonzalez, A., McMurdie, P.J., Vazquez-Baeza, Y., Xu, Z., Ursell, L.K., Lauber, C., Zhou, H., Song, S.J., Huntley, J., Ackermann, G.L., Berg-Lyons, D., Holmes, S., Caporaso, J.G. & Knight, R.

2013. Advancing our understanding of the human microbiome using QIIME. Methods in Enzymology, 531:371-444.

Nomenclature., I.C.o.Z., Ride, W.D.L., Nomenclature., I.T.f.Z., Sciences., I.U.o.B. & Museum, N.H. 1999. International code of zoological nomenclature = Code international de nomenclature zoologique. London :: International Trust for Zoological Nomenclature, c/o Natural History Museum.

Patil, K.R., Roune, L. & McHardy, A.C. 2012. The PhyloPythiaS web server for taxonomic assignment of metagenome sequences. PLoS One, 7(6):e38581.

Pavoine, S., Dufour, A.B. & Chessel, D. 2004. From dissimilarities among species to dissimilarities among communities: a double principal coordinate analysis. Journal of Theoretical Biology, 228(4):523-537.

Pearson, K. 1901. LIII. On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6, 2(11):559-572.

Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W. & Smyth, G.K. 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research, 43(7):e47.

Rohde, H., Qin, J., Cui, Y., Li, D., Loman, N.J., Hentschke, M., Chen, W., Pu, F., Peng, Y., Li, J., Xi, F., Li, S., Li, Y., Zhang, Z., Yang, X., Zhao, M., Wang, P., Guan, Y., Cen, Z., Zhao, X., Christner, M., Kobbe, R., Loos, S., Oh, J., Yang, L., Danchin, A., Gao, G.F., Song, Y., Li, Y., Yang, H., Wang, J., Xu, J., Pallen, M.J., Wang, J., Aepfelbacher, M. & Yang, R. 2011. Open-source genomic analysis of Shiga-toxin-producing E. coli O104:H4. New England Journal of Medicine, 365(8):718-724.

Schatz, M.C., Delcher, A.L. & Salzberg, S.L. 2010. Assembly of large genomes using second-generation sequencing. Genome Research, 20(9):1165-1173.

Scher, J.U. 2016. The microbiome in celiac disease: Beyond diet-genetic interactions. Cleveland Clinic Journal of Medicine, 83(3):228-230.

Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O. & Huttenhower, C. 2012. Metagenomic microbial community profiling using unique clade-specific marker genes. Nature Methods, 9(8):811-814.

Shannon, C.E. 1948. A mathematical theory of communication. The Bell System Technical Journal, 27(4):623-656.

Simpson, E.H. 1949. Measurement of Diversity. Nature, 163:688-688.

Speeckaert, R., Lambert, J., Grine, L., Van Gele, M., De Schepper, S. & van Geel, N. 2016. The many faces of interleukin-17 in inflammatory skin diseases. British Journal of Dermatology, 175(5):892-901.

Thomas, T., Gilbert, J. & Meyer, F. 2012. Metagenomics - a guide from sampling to data analysis. Microbial Informatics and Experimentation, 2:3-3.

Turner, M. 2011. Microbe outbreak panics Europe. Nature, 474(7350):137.

Whittaker, R.H. 1972. Evolution and Measurement of Species Diversity. Taxon, 21(2/3):213-251.

Woo, P.C.Y., Lau, S.K.P., Teng, J.L.L., Tse, H. & Yuen, K.Y. Then and now: use of 16S rDNA gene sequencing for bacterial identification and discovery of novel bacteria in clinical microbiology laboratories. Clinical Microbiology and Infection, 14(10):908-934.

Wood, D.E. & Salzberg, S.L. 2014. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biology, 15(3):R46.

Wroblewski, L.E., Peek, R.M., Jr. & Coburn, L.A. 2016. The Role of the Microbiome in Gastrointestinal Cancer. Gastroenterology Clinics of North America, 45(3): 543-556.

Yeung, K.Y. & Ruzzo, W.L. 2001. Principal component analysis for clustering gene expression data. Bioinformatics, 17(9):763-774.

Zhou, J., Xia, B., Treves, D.S., Wu, L.Y., Marsh, T.L., O'Neill, R.V., Palumbo, A.V. & Tiedje, J.M. 2002. Spatial and resource factors influencing high microbial diversity in soil. Applied and Environmental Microbiology, 68(1):326-334.

**CURRICULUM VITAE**

# Solaiappan Manimaran, MA, M.Eng, MBA

77 Mason Terrace Apt 22, Brookline, MA 02446
Phone: (860) 997-2762; Email: mani2012@bu.edu

## SUMMARY OF QUALIFICATIONS

I would bring 12 years of industry experience in system diagnostics and five years of research and development experience in Computational Biomedicine. I am a US citizen and authorized to work. My career goal is to lead and execute challenging projects involving statistical analysis by integrating inputs from diverse sources, incorporating scientific and business perspectives to achieve commercial success.

## RESEARCH INTEREST

Machine Learning and Statistical Analysis of Next Generation Sequencing data for cancer biomarkers and microbiome analysis of pathogen proportions in clinical mixed sample data towards improved diagnostics and personalized medicine.

## EXPERIENCE

**Sep 2012 – current** ***Boston University – Dr. Evan Johnson's Lab***
**Research Assistant in Computational Biomedicine**

- Lead software architect and development of complete Metagenomic analysis software pipeline called PathoScope, Batch Effects analysis pipeline called BatchQC and Microbiome variation analysis pipeline called PathoStat R Shiny App package.
- Bayesian modeling and analysis of high dimensional genomic data for Biomarkers and correlated data analysis including Pathway analysis.

**Aug 2016 – Sep 2016** ***Boston Biomedical Associates (BBA) - A Clinical Research Organization (CRO),*** *Marlborough, MA*
**Biostatistician**

- Helped with **clinical trials data analysis and automatic generation of custom reports** in word and excel formats using SAS programs.

**Feb 2012 – Aug 2012** ***SS&C Technologies,*** *Glastonbury, CT*
**Senior** ***Software Engineer – BenefitsXML Group***

- Helped with **data analysis and software engineering** for functional enhancements to web based software BRIX - Benefits Real-time Information eXchange.

**Feb 2000 – Oct 2011** ***Qualtech Systems, Inc. (QSI),*** *East Hartford, CT*
**Member of Technical Staff,** ***Software R&D Engineer & Quality Manager***
Technical Responsibilities:

- **Research, Statistical analysis of data**, Modeling and Diagnostics of complex systems including Medical devices using **Machine Learning techniques.**
- **Integrate data from multiple platforms and electronic records including those of medical devices** to enhance QSI's TEAMS (Testability Engineering And Maintenance System) functionality.
- **Data analysis** and implementation of algorithm for the web-based TEAMATE diagnostic reasoning engine within TEAMS-RDS (Remote Diagnosis Server) toolset.
- **Proposal, Report preparation and Project Lead** for execution of awarded projects, in collaboration with the Formal Systems Laboratory at the University of Illinois and the University of Connecticut:
  - Integrated Tool-Supported Framework for Integrated Vehicle Health Management (IVHM) Monitoring, Control and Verification, NASA, 2008-2011.
  - Scalable Formal Methods for Distributed Systems, Air Force STTR project, 2007-2008.
  - Automated Network Health Management, US Navy, 2008-2010.

Processes and Tools:

- Management Representative responsible for managing the company's Quality Management System and maintaining ISO9001:2008 certification.
- Process Lead responsible for maintaining the company's Microsoft Gold Certified Member status and later transitioning into the Silver Competency for ISV (Independent Software Vendor) Microsoft Partner Network Program.

**Co-Innovator** at QSI for Software Innovation towards automatic data analysis and development of OWL (Web Ontology Language) compliant models and translation of TEAMS diagnostic tree into PRL (Procedure Representation Language) and PLEXIL (Plan Execution Interchange Language) as part of **NASA SBIR project** "Automation of Health Management, Troubleshooting and Recovery in Lunar Outpost"

2009: Received the **"Mr. Perfect"** award at QSI – In recognition for performing the most thorough and meticulous work and completion within schedule and budget**.**

2003: Received the Achievement award for **ISO9001 certification** at QSI – In recognition of my successful efforts to prepare QSI for ISO9001:2000 certification.

**1999-2000** ***Hughes Software Systems****, Bangalore, India.*
**Software Engineer**

- Responsible for developing a VOIP (Voice Over IP) networking product called GateKeeper using the H323 protocol.
- Responsible for the web design of an e-commerce advertisement server product.

**1998-1999** ***Enterprise Component Technology****, Bangalore, India*.
**Software Programmer** (Part Time)

- Developed a framework and web-based GUI for an e-Commerce product.

## EDUCATION

**Ph.D.** Candidate - **Biostatistics** (current graduate student)
**Boston University**, September, 2012 – expected graduation in December, 2016.
**Research Assistantship** in Dr. Evan Johnson's Lab.

**MA** - **Biostatistics**
**Boston University**, Boston, September, 2014.

**MBA** - **Finance and Marketing**,
**Advanced Business Certificate** (ABC) in **Project Management**,
**University of Connecticut, 2009 - February 2012**. CGPA: 4.14/4.0

**M.Eng**. - **Computer Science & Engineering** (Integrated 4-year program)
**Indian Institute of Science (IISc)**, Bangalore, India, September, 1999.
**IISc scholarship** while pursuing Masters degree.

**BS (Honors)** - **Mathematics**,
**Indian Institute of Technology**, Kharagpur, India, June, 1995

## COMPUTER/SOFTWARE SKILLS

### Programming Languages:

- R, SAS (11 years) **(BatchQC and PathoStat R-package in Bioconductor)**
- Python, Matlab, Simulink, LabView, Perl, Tcl/Tk, Awk, Sed (11 years)
- Java, Java Servlets, JSP, JNI, Javascript (15 years)
- C, C++, VC++ (13 Years)
- XSL, XML, HTML (12 years)
- SQL database (DB) language, ODBC, JDBC (12 years)

### Operating Systems:

- Windows 9X, NT, ME, 2000, XP, Vista, 7, 8, 10 (20 years)
- Linux, UNIX (20 years); Android, iOS (8 years)

- Cloud Computing: Amazon Web Services, HPC Server Clusters (8 years)

### Software Development:

- Machine Learning/Statistical and Numerical Methods based software (15 years)
- Web Services, Cloud Computing, Diagnostics & Prognostics Software (12 years)
- Networking applications, Distributed Systems, VOIP (12 years)
- DB Applications – Oracle, MySQL, SQL Server, Access, DB2, H2 (12 years)
- Next Generation Sequencing (NGS), Microbiome Analysis Toolkit (4 years)

### Software Tools:

- Eclipse IDE, R-Studio and Visual Studio (15 years)
- Microsoft Office – Word, Excel, Powerpoint, Access, Project (15 years)
- Revision Control System GitHub, Subversion, CVS (15 years)
- Variety of Java Packages – JFreeChart, JFreeReport, Ant, Maven (15 years)

## PUBLICATIONS

- **Manimaran,** Selby, Okrah, Ruberman, Leek, Quackenbush, Kains, Bravo, Johnson "BatchQC: Interactive software for evaluating sample and batch effects in genomic data" **(2016: Oxford Bioinformatics, Applications Note)**
- Hong*, **Manimaran***, Shen, Perez-Rogers, Byrd, Castro-Nallar, Crandall, Johnson, "PathoScope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples" **(2014: Microbiome 2 (1), 1-15) *Co-First author**
- Francis, Bendall, **Manimaran**, Hong, Clement, Castro-Nallar, Snell, Schaalje, Clement, Crandall, and Johnson, "Pathoscope: Species identification and strain attribution with unassembled sequencing data" **(2013: Genome research 23 (10), 1721-1729)**
- Byrd, Perez-Rogers, **Manimaran**, Castro-Nallar, Toma, McCaffrey, Siegel, Benson, Crandall, Johnson, "Clinical PathoScope: rapid alignment and filtration for accurate pathogen identification in clinical samples using unassembled sequencing data" **(2014: BMC bioinformatics 15 (1), 262)**
- Hong, **Manimaran**, Johnson, "PathoQC: Computationally Efficient Read Preprocessing and Quality Control for High-Throughput Sequencing Data Sets" **(2015: Cancer Informatics 2014:Suppl. 1 167-176)**
- Castro-Nallar, Shen, Freishtat, Perez-Losada, **Manimaran**, Liu, Johnson and Crandall, "Integrating microbial and host transcriptomics to characterize asthma-associated microbial communities" **(2015: BMC Medical Genomics)**

## RELEVANT SKILLS/TRAINING

### Statistical Genomics:

- Analysis of Microarray and Next Gen. Sequencing Data BS830 (BU) Grade: A
- Bayesian modeling for biomedical res. and public health BS855 (BU) Grade: A
- Statistical Genetics I BS858 (BU) Grade: A-
- Statistical Genetics II BS860 (BU) Grade: Audit
- Generalized Linear Models with applications BS853 (BU) Grade: A

### Clinical Trials:

- Applied Statistical Methods for Clinical Trials I BS851 (BU) Grade: A-
- Applied Statistical Methods for Clinical Trials II BS861 (BU) Grade: A
- Bayesian Methods in Clinical Trials BS854 (BU) Grade: Audit
- Practical Skills for Biostatistics Collaboration BS715 (BU) Grade: A

### Mathematical Statistics:

- Mathematical Statistics MA582 (BU) Grade: A
- Hypothesis Testing MA782 (BU) Grade: A
- Stochastic Processes MA583 (BU) Grade: A

- Applied Time Series STAT 380 (UConn) Grade: A
- Statistical Computing STAT 5361 (UConn) Grade: A

### Project and Financial Management:

- Project Management OPIM 5270 (UConn) Grade: A+
- Project Risk and Cost Management OPIM 5668 (UConn) Grade: A+
- Financial Management FNCE 5101 (UConn) Grade: A+
- Corporate Finance FNCE 5209 (UConn) Grade: A+
- Cost Analysis & Control ACCT 5123 (UConn) Grade: A+

### Business and Marketing Management:

- Business, Law & Ethics BLAW 5175 (UConn) Grade: A+
- Managing Organizations MGMT 5138 (UConn) Grade: A
- Market Driven Management MKTG 5115 (UConn) Grade: A
- New Product & Innovation Management MKTG 5230 (UConn) Grade: A+

## HONORS, INTERESTS & ACTIVITIES

- Member of Mu Sigma Rho National Statistical Honor Society, Member of American Statistical Association (ASA), Member of Beta Gamma Sigma, Member of Golden Key International Honour Society, Member of IEEE (2000-Current)
- Received Oral Presentation Award from BU Genome Science Institute for "BatchQC: Interactive software framework for evaluating sample and batch effects in genomic data", GSI Research Symposium, November 10, 2015
- Chess; Yoga; Ancient Scriptures

## MAJOR PROJECTS

- **PathoScope** software in **python** for metagenomics analysis at Dr. Evan Johnson's Lab – Lead software architect, Statistical analysis of data and development of complete metagenomics software package that takes next-generation sequencing reads from a mixture sample and predicts which genomes are present using a **Bayesian statistical analysis framework**.
- **PathoStat:** A Comprehensive Toolkit for Microbiome Variation Analysis developed at Dr. Evan Johnson's Lab – Lead software architect, Statistical analysis of data and development of a **Shiny App R-package**
- **BatchQC:** interactive software for evaluating sample and batch effects in genomic data, **Shiny App R-package** developed at Dr. Evan Johnson's Lab – Lead software architect, Statistical analysis of datasets including **proteomics** dataset
- Generalized Linear Models to characterize the occurrence of clinical outcomes related to Inferior Vena Cava (IVC) injury
- Analysis of microarray data from sickle-cell disease (SCD) anemic subjects with pulmonary hypertension (PHT+) and without pulmonary hypertension (PHT-) to find a biomarker using differentially expressed genes between the two groups.
- Single-cell RNA-Seq Differential Expression Analysis using Bayesian modeling
- Biomarker identification of gene set using longitudinal DNA methylation data for differential methylation level corresponding to different drug treatments of cancer
- Genome-Wide Association Study (GWAS) with genotype association of Fasting Plasma Glucose (FPG) in the diagnosis of type 2 diabetes
- Survival analyses of the association between smoking and Coronary Heart Disease (CHD) and overall survival
- **Co-Investigator and Project Lead of NRA (NASA Research Announcement) Project** - "An Integrated Tool-Supported Framework for IVHM Monitoring, Control and Verification" in collaboration with Prof. Grigore Rosu, Formal Systems Laboratory, University of Illinois: Development of a Monitoring-Oriented Programming (MOP) plugin for TEAMS (Testability Engineering And Maintenance System)

software for monitoring IVHM (Integrated Vehicle Health Management) Systems.

- **Co-Investigator and Project Lead of US Air Force STTR Phase I Project** – "Scalable Formal Methods for Distributed Systems" in collaboration with Prof. Grigore Rosu, Formal Systems Laboratory, University of Illinois: Development of algorithms for comparing models to help in the design of distributed systems.
- **Project Lead of US Navy SPAWAR SBIR Project** "Automated Network Health Management" in collaboration with University of Connecticut, Lockheed Martin and Ridgetop Group Inc: Development of a complete Network Monitoring System for ship-board networks.
- **NASA SBIR project** "Automation of Health Management, Troubleshooting and Recovery in Lunar Outpost": Automate a part of Verification and Validation (V&V) of procedures, Health Management and recovery decision support-related activities through QSI's TEAMS-based approach.
- **NASA STTR project** "System Health and Impact Assessment Environment Demonstrated on Advanced Diagnostic and Prognostic Testbed (ADAPT)": Integrating QSI's software with ADAPT at NASA ARC (Ames Research Center).
- **Data analysis and Development of Web services** for integrating third party applications with QSI's TEAMS (Testability Engineering And Maintenance System) software suite
- **US Air Force SBIR Project** "Active Bus Analysis and Failure Forecasting": Monitoring, testing and analysis of aircraft busses - Used LabView for analysis.
- M.Eng Master's thesis titled "Better ways for matching XSL patterns": Development of efficient Tree pattern matching algorithms and techniques.
- Modeling the development of Viral Diseases as a system of differential equations and solving the system of differential equations through Numerical Techniques and Linear Programming Techniques.

## CONFERENCE PRESENTATION

- **2016 Joint Statistical Meeting (JSM), Chicago:** "BatchQC: Interactive software framework for evaluating sample and batch effects in genomic data", Contributed Paper, Section on Statistics in Genetics and Genomics.
- **2015 11th International Conference on Health Policy Statistics (ICHPS) - Providence, Rhode Island**, "Pathoscope 2.0: Statistical and computational methods for accurate characterization of microbes in sequencing samples", Contributed Presentation.
- **2015 7th Annual Genome Science Institute (GSI) Research Symposium** , "BatchQC: Interactive software framework for evaluating sample and batch effects in genomic data", **Graduate Speaker Award**.
- **2015 The 29th New England Statistical Symposium (NESS), University of Connecticut,** "Pathoscope 2.0: Statistical and computational methods for accurate characterization of microbes in sequencing samples."
- **2014 Joint Statistical Meeting (JSM), Boston: "**Statistical and computational methods for accurate characterization of microbes in clinical and environmental sequencing samples", Biometrics Section, Track: Large Scale Hypothesis Testing Biomarker Evaluation

## POSTER PRESENTATION

- **2016 8th Annual Genome Science Institute (GSI) Research Symposium** , "PathoStat: A Comprehensive Toolkit for Microbiome Variation Analysis"
- **2016 Evans Research Days Poster Session**, "PathoStat: A Comprehensive Toolkit for Microbiome Variation Analysis"
- **2016 Inaugural MIT-Harvard Symposium on Health & Ventures in the Microbiome,** "Pathoscope 2.0: Statistical and computational methods for accurate

characterization of microbes in sequencing samples."

- **2016 Boston University Data Science (BUDS) Day**, "BatchQC: Interactive software framework for evaluating sample and batch effects in genomic data"
- **2015 Evans Research Days Poster Session**, "BatchQC: Software pipeline to identify batch effects in sequencing and microarray data and adjust for it in the best possible way"
- **2014 Fourth Annual Boston University (BU) Clinical & Translational Science Institute (CTSI) Translational Science Symposium - Research on Disparities in Health Care** "Pathoscope 2.0: Statistical and computational methods for accurate characterization of microbes in sequencing samples"
- **2014 Evans Research Days Poster Session**, "Pathoscope 2.0: Statistical and computational methods for accurate characterization of microbes in sequencing samples"
- **2014 The 28th New England Statistical Symposium (NESS), Harvard School of Public Health** "Statistical and computational methods for accurate characterization of microbes in clinical and environmental sequencing samples
- **2014 Tufts Clinical and Translational Science Institute (CTSI) Translational Research Day**, "Pathoscope 2.0: Statistical and Computational Methods for Accurate Characterization of Microbes in Sequencing Samples"
- **2014 6th Annual Genome Science Institute (GSI) Research Symposium** , Pathoscope 2.0: Statistical and computational methods for accurate characterization of microbes in sequencing samples"
- **2013 Evans Research Days Poster Session**, "PathoScope: Methods for accurate identification and estimation of genome proportions in metagenomic samples"

## STATUS

US Citizen