Computational Methods for Human Microbiome Analysis

Matthieu J. Miossec,[1] Sandro L. Valenzuela,[1] Katterinne N. Mendez,[1] and Eduardo Castro-Nallar[1]

[1]Center for Bioinformatics and Integrative Biology, Faculty of Biological Sciences, Universidad Andrés Bello, Santiago, Chile

As the field of microbiomics advances, the burden of computational work that scientists need to perform in order to extract biological insight has grown accordingly. Likewise, while human microbiome analyses are increasingly shifting toward a greater integration of various high-throughput data types, a core number of methods form the basis of nearly every study. In this unit, we present step-by-step protocols for five core stages of human microbiome research. The protocols presented in this unit provide a base case for human microbiome analysis, complete with sufficient detail for researchers to tailor certain aspects of the protocols to the specificities of their data. © 2017 by John Wiley & Sons, Inc.

Keywords: alpha and beta diversity • differential abundance testing • human microbiome • metagenome decontamination • metagenome reads • read mapping

How to cite this article:
Miossec, M. J., Valenzuela, S. L., Mendez, K. N., & Castro-Nallar, E. (2017). Computational methods for human microbiome analysis. *Current Protocols in Microbiology*, *47*, 1E.14.1–1E.14.17. doi: 10.1002/cpmc.41

INTRODUCTION

The study of the human microbiome is critical to gaining a full understanding of human health. Given the strong microbial diversity that already exists among healthy individuals, studying microbiome-based disorders requires a carefully crafted analysis that leads to reproducible insights (The Human Microbiome Project, 2012). While a clear research question, sound sampling, and sequencing protocols are key to the success of a study, they must be accompanied by a robust computational downstream analysis. In this unit, we present the core components of such an analysis as five basic protocols using tools tailored to human microbiome research, such as Pathoscope 2.0 (Hong et al., 2014) and PathoStat (Manimaran et al., 2016). In this unit, we work through each protocol using an example dataset originating from a study of asthma-associated microbial communities (Castro-Nallar et al., 2015). The 14 metagenome samples used for this unit were collected from eight children and adolescents with asthma and six healthy controls, extracted from the inferior turbinate of each nare.

In the Basic Protocol 1, we revisit a crucial element of all sequencing projects: quality control. For the best quality control we combine the strengths of FastQC (Andrews, 2010) and PRINSEQ (Schmieder & Edwards, 2011). In Basic Protocols 2 and 3, we perform read mapping, in each case with a different purpose. In Basic Protocol 2, the aim is the decontamination of sequence data. Human microbiome sequence data is cluttered with host DNA. By mapping reads to a human genome reference using Bowtie2 (Langmead

Current Protocols in Microbiology 1E.14.1–1E.14.17, November 2017
Published online November 2017 in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpmc.41

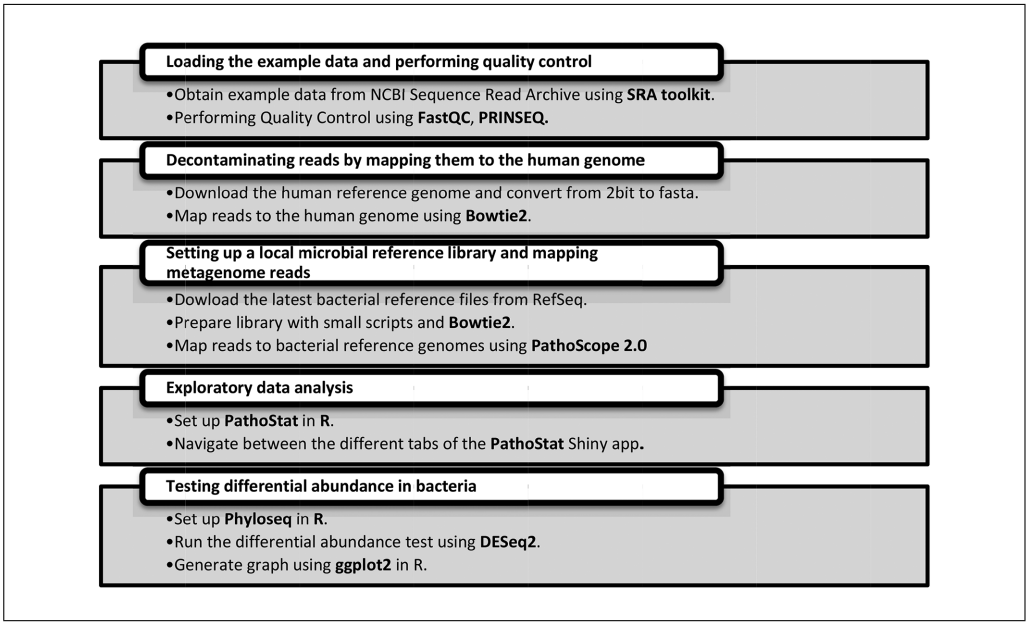


Figure 1E.14.1 Overview of each basic protocol contained in this unit.

& Salzberg, 2012), we can remove such reads. Using Pathoscope 2.0 (Hong et al., 2014) in Basic Protocol 3, we map the remaining reads to microbial libraries, thus providing these with taxonomic classifications. This mapping will in turn allow the researcher to determine the abundance and diversity of taxonomic units. This protocol will also present a way of building the necessary microbial library locally for Pathoscope 2.0 to use.

In Basic Protocols 4 and 5, we explore the composition of the microbial communities of each of our patients using an array of statistical and enumerative tools supported by graphical representations. We divide the data across a number of dimensions, such as taxonomic class or the host health status. From the gathered information, we can draw inferences about the content of our samples and how these relate to one another, particularly with regard to disease. The majority of the statistical analysis will be done using PathoStat, as delineated in the Basic Protocol 4, with the testing of differential abundance of bacteria performed using Phyloseq (McMurdie & Holmes, 2013) in Basic Protocol 5.

Together, these protocols, and the corresponding tools, embody the complete set of computational methods necessary for a successful human microbiome analysis. Figure 1E.14.1 provides an overview of the unit.

NOTE: For each of our terminal instructions, line numbers (in light gray) have been added for clarity. Each line number corresponds to a new line in the terminal.

BASIC PROTOCOL 1

LOADING THE EXAMPLE DATA AND PERFORMING QUALITY CONTROL

Basic Protocol 1 lays out the first stage of the analysis along with essential pre-processing steps. The example data provided for this analysis, which can be substituted with the researcher's data, must first undergo quality control. We collect and visualize quality-control metrics using FastQC (v. 0.11.5). In order to act upon these metrics, we use the read trimming and filtering options provided by PRINSEQ (v. 0.20.4).

Materials

To run programs that are essential to later protocols (i.e., Bowtie2, Pathoscope 2.0) a high-performance computer (HPC) cluster running on Linux is strongly

recommended. Programs must be run from the command line terminal corresponding to the login node of the cluster. For this protocol, FastQC and PRINSEQ must be installed. The tools and their installation guidelines are available at the following URLs:
FastQC, v.0.11.5+:
https://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc
PRINSEQ Lite v.0.20.4+:
https://sourceforge.net/projects/prinseq/files/standalone/

Additionally, to acquire the 14 examples stored at NCBI's Sequence Read Archive (SRA), NCBI SRA Toolkit must be installed and configured. The toolkit [SRA Toolkit (v.2.8.2-1)] and its documentation are available at *https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software*

1. Acquire the 14 example samples recommended to perform the whole unit using SRA Toolkit.

   We recommend that users first complete this unit with the example data provided through NCBI's SRA before applying it to their own sequence data. The data consists of 14 fastq files, which are available at https://www.ncbi.nlm.nih.gov/bioproject/PRJNA255523/

   To download the fastq files, in the terminal, type:

   ```
   1 [path/to/]fastq-dump [AccessionNumber]
   ```

   Input the following accession numbers:

   ```
   SRR1528344
   SRR1528346
   SRR1528348
   SRR1528420
   SRR1528426
   SRR1528430
   SRR1528434
   SRR1528456
   SRR1528458
   SRR1528460
   SRR1528462
   SRR1528464
   SRR1528466
   SRR1528468.
   ```

   This will produced the corresponding files in fastq format.

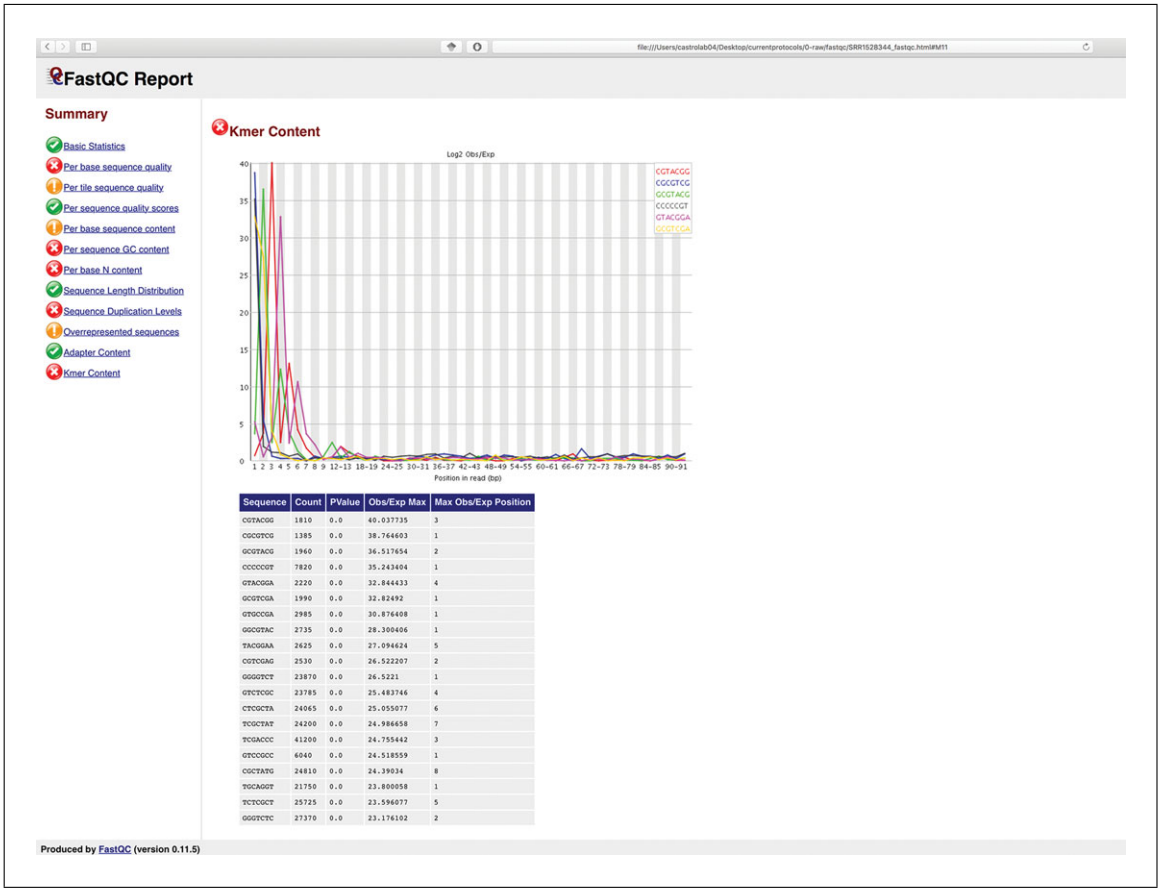
2. Run FastQC on each of the 14 sample files.

   FastQC is a small program that collects a series of quality metrics from fastq files (Andrews, 2010). The program provides a summary of quality metrics and, for each, assigns pass (potentially with warnings) or fail status. FastQC provides reports where metrics are reproduced graphically.

   In the folder containing the files, for each of the 14 sample files, type:

   ```
   1 [path/to/]fastqc -t num_threads [filename].fastq
   ```

   This will produce several files, most importantly `fastqc_report.html`*, an example of which is shown in Figure 1E.14.2. The user must set the number of threads used by the programs to generate its statistics (*`-t num_threads`*) based on the constraints of the cluster being used.*



Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGTACGG	1810	0.0	40.037735	3
CGCGTCG	1385	0.0	38.764603	1
GCGTACG	1960	0.0	36.517654	2
CCCCCGT	7820	0.0	35.243404	1
GTACGGA	2220	0.0	32.844433	4
GCGTCGA	1990	0.0	32.82492	1
GTGCCGA	2985	0.0	30.876408	1
GGCGTAC	2735	0.0	28.300406	1
TACGGAA	2625	0.0	27.094624	5
CGTCGAG	2530	0.0	26.522207	2
GGGGTCT	23870	0.0	26.5221	1
GTCTCGC	23785	0.0	25.483746	4
CTCGCTA	24065	0.0	25.055077	6
TCGCTAT	24200	0.0	24.986658	7
TCGACCC	41200	0.0	24.755442	3
GTCCGCC	6040	0.0	24.518559	1
CGCTATG	24810	0.0	24.39034	8
TGCAGGT	21750	0.0	23.800058	1
TCTCGCT	25725	0.0	23.596077	5
GGGTCTC	27370	0.0	23.176102	2

Figure 1E.14.2 What the FastQC report should look like for `SRR1528344.fastq`. The kmer content is problematic at the start of many reads and will therefore affect what filters we apply with PRINSEQ in the following step.

3. Open `fastqc_report.html` in your default browser and survey the different quality metrics.

   The quality metrics reported by FastQC suggest a number of issues that can be resolved by trimming and filtering reads. In particular, we see an excess of k-mers at the start of many reads. FastQC does not offer the necessary tools for trimming and filtering our reads. We therefore now use PRINSEQ. In Figure 1E.14.3, we provide a list of criteria used to assess trimming and filtering requirements.

4. Perform quality control trimming and filtering on each of the 14 sample files using PRINSEQ.

   As with FastQC, PRINSEQ provides tools to assess the quality of sequence data from both genomic and metagenomic datasets (Schmieder & Edwards, 2011). Of particular interest to us here is that it provides utilities for trimming the ends of reads and filtering those of dubious quality.

   To do this, in the terminal type:

```
1 perl [path/to/]prinseq-lite.pl -fastq
  [filename].fastq
-out_good [filename].fastq.pass -out_bad null
-trim_left 5 -trim_qual_right 20 -min_len 75
-trim_qual_window 15 -trim_qual_step 5
-ns_max_n 0
```

   As indicated by the line number, this command must be typed as a single line.

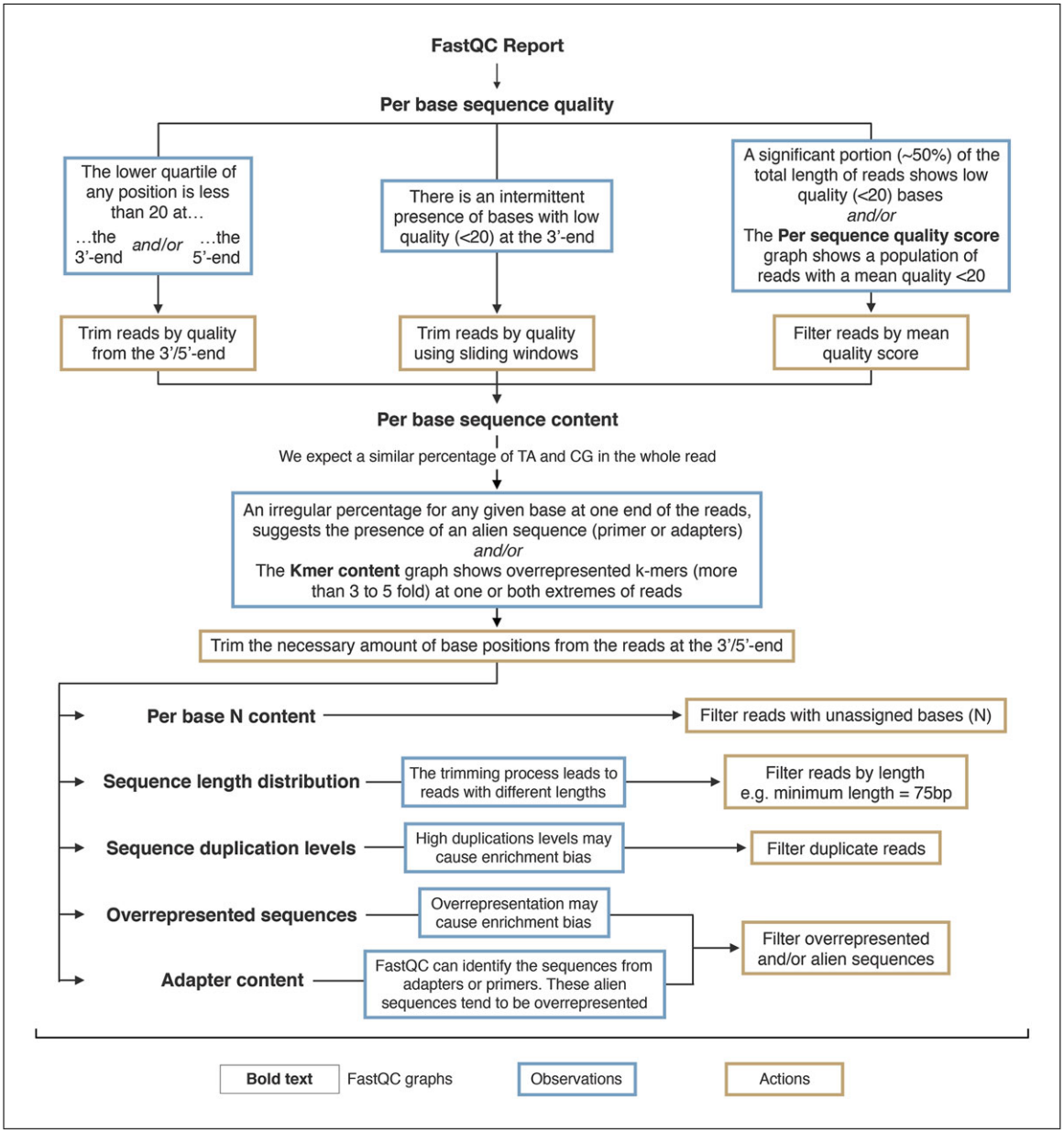


Figure 1E.14.3 Decision tree outlining the quality control steps applied to raw sequencing reads. We highlight observations (blue) and the recommended actions (gold). FastQC graphs are highlighted in bold.

We apply both trimming and filtering to the reads of each of our fastq files (`-fastq`*) and output only those reads that pass the threshold we have set (*`-out_bad null`*). Given the high clustering of k-mers across the start end of reads in our data, we trim this end (*`-trim_left`*).*

Note that this step will vary strongly for other experimental data and will depend very much on the output of FastQC.

We also trim our reads according to the average base quality in subsets of the read starting from the right. An average quality score is determined within a series of windows. The size of the window and steps between windows are manually set (`-trim_qual_window`* and* `-trim_qual_step`*). The score for trimming is set to 20 (*`-trim_qual_right`*). If any of the reads have a length <75 bp after trimming, they are filtered (*`-min_length`*). Additionally, any sequence that contains unassigned bases (N) is removed (*`-ns_max_n`*).*

DECONTAMINATING READS BY MAPPING THEM TO THE HUMAN GENOME USING BOWTIE2

Basic Protocol 2 describes the mapping of the quality-controlled reads to a human reference genome. The goal of the mapping process is to identify, and ultimately discard, host DNA. This step can be understood as an in silico decontamination step. The reads are mapped to the human genome reference using the sequence aligner Bowtie2 (v2.2.9). With the current example data, this protocol is not absolutely required for PathoScope 2.0 to perform its mapping correctly in Basic Protocol 3. However, it will be necessary for many other human microbiome studies. We therefore consider it an important protocol and provide it here for the benefit of other analyses.

Materials

To run Bowtie2, a high-performance computer (HPC) cluster running on Linux is strongly recommended. The user must install Bowtie2 and the tool twoBitToFa, following the installation guidelines available at:
Bowtie2, v.2.2.9+: *http://bowtie-bio.sourceforge.net/bowtie2/index.shtml*
twoBitToFa:
https://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/twoBitToFa

1. Download the human genome reference.

   To do this, in the terminal type:

```
1 mkdir human
2 cd human/
3 wget http://hgdownload.cse.ucsc.edu/goldenPath/
  hg19/bigZips/hg19.2bit
```

2. Convert the 2-bit file into fasta using twoBitToFasta.

   To do this, in the terminal type:

```
1 [path/to/]twoBitToFa hg19.2bit hg19.fasta
```

3. Finally, index the human genome.

```
1 [path/to/]bowtie2-build -large-index --threads
  [CPU]x2 hg19.fasta hg19
2 cd ..
```

4. Run Bowtie2 on each of the 14 files containing quality controlled reads.

   Bowtie2 is a fast and sensitive sequence aligner with a relatively small memory footprint, which supports several different type of alignments (Langmead & Salzberg, 2012). We use Bowtie2 to map reads to the human genome reference, leading to distinct outputs: a SAM file containing reads that align to the reference and reads in fastq format that do not align. The latter are the reads we want to analyze going forward.

   To run Bowtie2, in the terminal type:

```
1 [Path/to/]bowtie2 -x human/hg19 --end-to-end -q-U
  [filename].fastq.pass -p [CPU]x2 -S
  [filename].human.sam --un [filename].filtered.fastq
```

   We map unpaired reads in fastq format (`-U` and `-q`) against the human genome (`-x`) in order to remove as much as possible of the host DNA. The reads that map to the reference are given in the SAM format (`-S`). The reads that do not, the filtered reads that we are interested in going forward, are provided in a in fastq format (`-un`). The mapping must be end to end (`--end-to-end`), i.e., the entire read must align to the reference genome

to be mapped. We recommend setting the number of threads the program uses (`-p`) to one that fits the constraints of the cluster being used.

SETTING UP A LOCAL MICROBIAL REFERENCE LIBRARY AND MAPPING METAGENOME READS TO THE REFERENCES USING PATHOSCOPE 2.0

BASIC PROTOCOL 3

Basic Protocol 3 covers the whole process of mapping reads to multiple microbial references contained in a library. The aim is to match each read to the bacterial genome that best corresponds to the original organism. From this pairing of reads and reference genomes, we will thus be able in later protocols to ascertain the abundance of different organisms and the diversity of the human microbiome being studied. Following decontamination, we use PathoScope 2.0 to map reads to microbial genome references which are gathered in a library reproduced locally. The PathoScope 2.0 mapping is done in two distinct steps: First, reads are mapped using Bowtie2 internally in PathoMap. Second, a number of reads are re-assigned using a penalized statistic mixture model in PathoID. Reads mapped to organisms with common taxon IDs are regrouped across a single organism. Reads that can potentially map to several references are re-assigned to the reference that most likely corresponds to the source organism based on other mappings. For detailed information regarding the algorithm implemented in PathoScope, see Francis et al. (2013) and Hong et al. (2014).

Materials List

To run PathoScope 2.0 a high-performance computer (HPC) cluster running on Linux is strongly recommended. To build a local library of bacterial reference genomes, the user must install pasteTaxID and Bowtie2 (if it is not already the case) and PathoScope 2.0 (for mapping) using the installation guidelines available at the following URLs:
pasteTaxID: *https://github.com/microgenomics/pasteTaxID*
Bowtie2, v.2.2.9+: *http://bowtie-bio.sourceforge.net/bowtie2/index.shtml*
PathoScope 2.0: *https://github.com/PathoScope/PathoScope*

The mapping of metagenome reads will be performed against reference sequences downloaded from NCBI Reference Sequence Database (RefSeq). Below are the steps necessary to download and format the required files.

1. Begin by downloading the latest bacterial reference fasta files from RefSeq using their FTP server.

   There are many files to download; therefore we recommend using wget with regular expression.

   In the terminal type:

```
1 mkdir bacteria
2 cd bacteria/
3 wget -r -nd -A '*.genomic.fna.gz'
  ftp://ftp.ncbi.nlm.nih.gov/refseq/release/bacteria/*
```

   This command will download all bacterial genomic references to the local folder. Files must then be unzipped with the following command:

```
4 gunzip *.gz
```

2. Run the pasteTaxID utility to restore taxonomic or genomic ID.

   Run `pasteTaxID` *in order to find and restore the taxonomic ID, or genomic ID if the former is unavailable, in the header of each bacterial reference. This will be useful in later protocols for PathoStat to correctly label the organisms identified through PathoScope.*

To do this, in the terminal type:

```
1 bash pasteTaxID.bash -workpath [path/to/bacteria/]
```

The workpath must be set to the directory containing the bacterial reference genomes.

3. Concatenate the reference sequences.

   While still in the folder containing all the bacterial references, type:

```
1 cat bacteria*fna > all_bacteria.fna
```

4. Build an index of the concatenated reference sequences using Bowtie2.

   To do this, in the terminal type:

```
1 bowtie2-build --large-index --threads [CPU]x2
  all_bacteria.fna all_bacteria
```

   Here again, the number of threads should correspond to the user's available resources.

   This index file is used for rapid mapping of reads to the bacterial reference without having to resort to the original fasta files.

   Having now prepared the bacterial reference library, the mapping process can begin.

5. Map filtered reads of each of the 14 samples to the bacterial reference genomes using the PathoScope 2.0 module PathoMap.

   PathoScope 2.0 is composed of a number of independent modules. In this protocol we use two of these modules, PathoMap and PathoID. PathoMap provides a first-pass mapping of reads to bacterial references.

   To use PathoMap, in the terminal type:

```
1 [path/to/]pathoscope2.py MAP
-U [filename].filtered.fastq
-indexDir [path/to/bacteria]
-targetIndexPrefixes all_bacteria
-outDir. -outAlign [filename].sam
-expTag MAPPED [filename] -numThreads [CPU]x2
```

   We take the filtered unpaired reads (`-U`) that did not align to the human genome reference and map them to bacterial reference genomes using the indices that were built in steps 1 to 4 of this protocol (defined using `-indexDir` and `-targetIndexPrefixes`). The results are given in the SAM format (`-out_align`).

6. Re-assign ambiguous reads using the PathoScope 2.0 module PathoID.

   Type the following commands:

```
1 [path/to]pathoscope2.py ID -alignFile
  [filename].sam -fileType sam -outDir. -expTag
  MAPPED_[filename]
```

   The SAM file containing the mapped reads (`-alignFile` and `-fileType`) are submitted to PathoID and a new mapping is assigned to a number of ambiguous reads based on a penalized statistic mixture model. The export tag is set to reflect the name of the original fastq file. For each of the 14 samples, a new SAM file is produced which accounts for the read re-assignments. Additionally, a tab-delimited file (`.tsv`) is generated. This file contains a report which is necessary for the protocols that follow.

EXPLORATORY DATA ANALYSIS USING PathoStat

BASIC PROTOCOL 4

Having mapped the metagenome reads to bacterial sequences, we can now begin to analyze the resulting data. The reports, generated for each sample by PathoID, contain the results necessary to carry out in-depth statistical analyses of the data. Of particular interest is the relative abundance and diversity of organisms at different taxonomic levels and which taxonomic classes of organisms tend to co-occur. For our example sample set, an important question is whether we see stark differences in the composition of microbial communities between healthy and asthmatic patients. To carry out all the enumerative and statistical analyses in a few simple steps, we use the program PathoStat. PathoStat is an R Shiny application that takes data generated by PathoScope 2.0 and outputs a large range of statistics in a browser window (Manimaran et al., 2016).

Materials

The user must install the statistical software environment R and from the console thus provided download the PathoStat package available through Bioconductor. The R package and installation guidelines for PathoStat are available at the following URLs:
R (v.3.3.1):
https://www.r-project.org/
PathoStat:
https://www.bioconductor.org/packages/release/bioc/html/PathoStat.html

1. Download `report.tsv`.

   To do this, in the terminal type:

   ```
   wget 'report.tsv'
   https://github.com/microgenomics/MicrobiomeAnalysisData/blob/master/report.tsv
   ```

   This file is a tab-delimited list of the 14 samples post-filtering, with run number, host condition (healthy or asthma), and run number. The most important information for PathoStat is the host condition.

   Depending on research requirements, different metrics will be of more interest than others. To consult the full gamut of metrics available through PathoStat, open R, load the PathoStat library, and, in the R console, type:

   ```
   vignette("PathoStatUserManual",
   package='PathoStat')
   ```

   To enter the R console, in the terminal type R. To close the R console, type `q()`. *Once you open the R console, do not close it until you have finished your analysis.*

2. Run PathoStat in R.

   In the R console, type:

   ```
   library(PathoStat)
   pstat<-createPathoStat(input_dir =
   "[path/to/report.tsv]", sample_data_file =
   "report.tsv")
   runPathoStat(pstat)
   ```

   With the first and second commands, you call up the PathoStat library and set up a working directory. With the third command, you create a PathoStat object that takes the report you have just downloaded, which you then use to run the program with the final command. A graphical user interface should now automatically open in your default browser, displaying the relative abundance of organisms in the 14 samples.

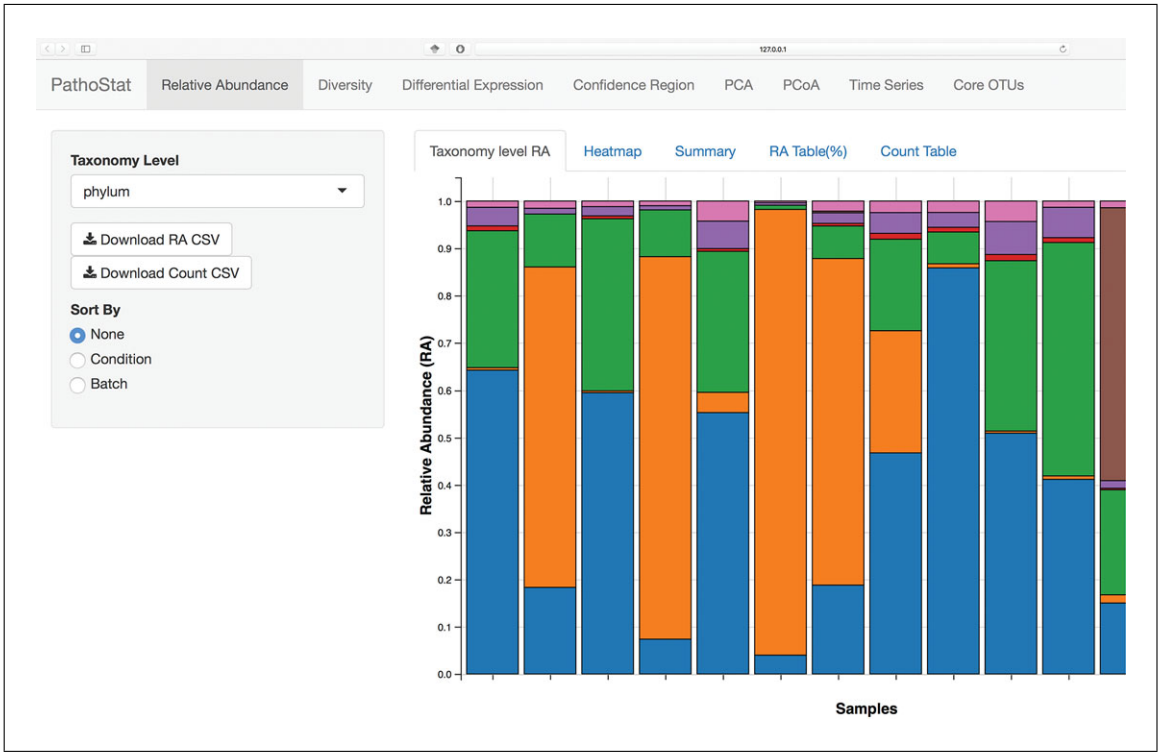


Figure 1E.14.4 Relative abundance of different taxa at the phylum level. What is striking is the high abundance of Proteobacteria in patients with asthma (first 8 columns) compared with healthy patients.

3. Visualize relative abundance by selecting different taxonomic levels.

   When the application is launched, you will see a graph representing the relative abundance of different organisms in the 14 samples, classified according to a taxonomic level that can be set by the user. Control the taxonomic level displayed by using the 'Taxonomy Level' drop-down menu in the left control panel. By setting the taxonomic level to phylum, for example, we notice a strikingly low level of proteobacteria in healthy patients compared with some asthmatic patients. An example of this is provided in Figure 1E.14.4.

   Additional information appears when hovering over a sample column such as the sample name, the taxonomic group to which a particular color code refers, or relative abundance as a ratio of the sample. Note that relative abundance is also delivered in text format—either as a percentage or as absolute counts of reads per taxon. These can be found by clicking the subtabs marked 'RA Table' and 'Count Table'.

4. Select the Diversity tab in the main menu and explore alpha and beta diversity.

   Selecting the diversity tab will automatically open the window on the alpha diversity subtab, showing different measures of species richness within the samples. Alpha diversity is determined by the number of taxonomic units present and whether only a few dominate (homogeneity). Beta diversity in the following subtab shows variation between samples, shown as a heatmap.

5. In the subtab menu, select co-occurrence.

6. Use the drop-down menus to assign taxonomic levels to the colors, shapes, and labels of the co-occurrence graph.

   Through the resulting graph, you will be able to see which bacterial taxa have numbers that rise and fall together. Figure 1E.14.5 provides one example of this. The strictness with which co-occurrence is determined can be adjusted with the maximum distance slider. This representation can be particularly helpful in teasing out co-dependence between organisms. Strong co-occurrence between several organisms often hints at a biological relationship between those, such as shared resources.

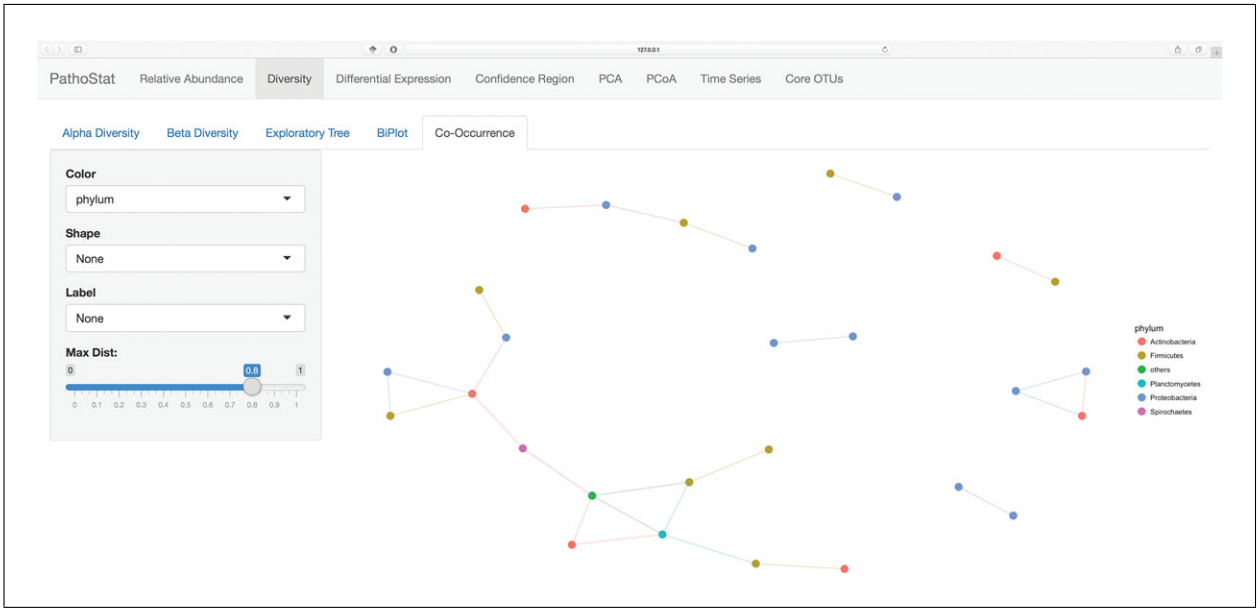


Figure 1E.14.5 Representation of co-occurrence of different taxa at the phylum level.

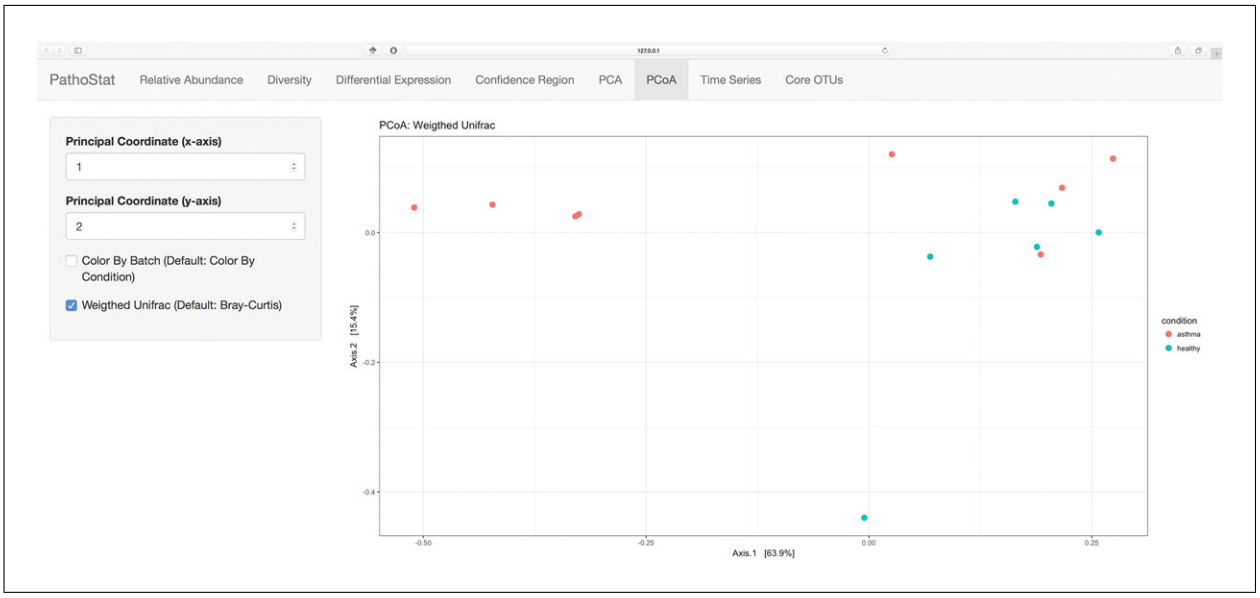


Figure 1E.14.6 Principal Coordinates Analysis with weighted Unifrac. We see one cluster that only contains patients with asthma.

7. In the main menu, select the PcoA (Principal Coordinates Analysis) tab.

8. In the control panel, select Weighted Unifrac.

   Bray-Curtis and UniFrac are distance metrics that show the level of dissimilarity between different microbial communities. The data are color-coded on individual health—healthy or with asthma. An example is shown in Figure 1E.14.6. With our example data, we see three clusters appear. It is worth noting that one of those clusters corresponds to asthma patients. It therefore appears that the microbiota of as subset of asthmatic patients differ significantly from that of controls.

9. Select the tab corresponding to Core OTUs

   While many of the statistics we have seen so far are about teasing out differences between our samples or categories of samples, it can be useful to find out what samples have in common at different taxonomic levels. This is what the core OTUs (for Operational Taxonomic Unit) graph shows. As with other tools, the taxonomic level being studied can

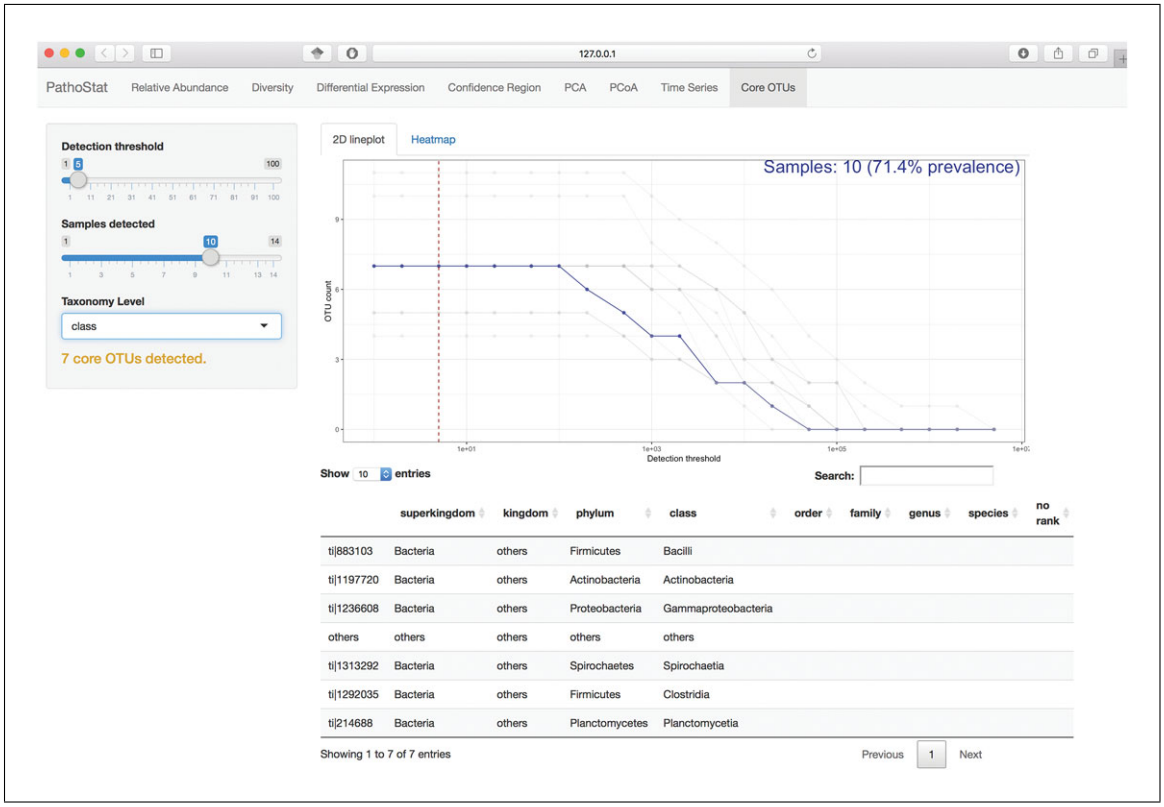


Figure 1E.14.7 Core OTUs found in at least 10 samples when a detection threshold of 5% is set.

be set via a drop-down menu. The threshold at which a taxon is considered part of a sample and the number of samples at which it will be considered a core OTU is set with two sliders in the left-hand panel. Figure 1E.14.7 shows one possible configuration.

Researchers should be aware that a number of other statistical tools exist within PathoStat that can be useful in specific contexts. For example, the time series, while of no use with the example data, is an important metric for studies that involve sampling of a given microbiome over time.

BASIC PROTOCOL 5

TESTING DIFFERENTIAL ABUNDANCE IN BACTERIA USING PHYLOSEQ

Testing the differential abundance in species of bacteria between healthy and asthmatic patients allows us to begin teasing out a correlation between particular bacteria and disease profile. This statistical method can be used to confirm what some of PathoStat's methods suggested. Using DEseq2 via Phyloseq, we are able to distinguish which species have increased presence in one set of patients over another.

Materials

The user must install the statistical software environment R and, from the console thus provided, download the DEseq2, Phyloseq, and ggplot2 packages available through Bioconductor and R-CRAN at the following URLs:
R (v.3.4.0):
https://www.r-project.org/
Independently of R, the user must also download parseMethods available at:
parseMethods:
https://github.com/microgenomics/parseMethods

1. Download 'metadata.txt' from the provided repository.

   To do this, in the terminal type:

```
1 wget 'metadata.txt'
  https://github.com/microgenomics/MicrobiomeAnalysis
  Data/blob/master/metadata.txt
```

The file is nearly identical to `report.tsv` *with the exception of the ID column. In this case, the ID column refers back to the names of the final reports produced by PathoScope for each sample.*

2. Parse together all the reports from PathoScope using parseMethods tool.

   To do this, in the terminal type:

```
1 bash parseMethods.bash --workpath [path/to/reports]
--method PATHOSCOPE
```

   The workpath must be set to the location where all the reports are contained so that these can be parsed together. We also signify that the report is formatted according to PathoScope's specification (`--method`*). This will lead to a comma-separated file* `pathoscope_table.csv` *containing the aggregate of all reports. This file is necessary for the following steps.*

3. Open R and load the libraries for phyloseq, DESeq2 and ggplot.

   To do this, in the R console type:

```
1 library(DESeq2)
2 library(phyloseq)
3 library(ggplot2)
```

   We first need to load our data into a phyloseq object. We can then convert the resulting object into a DESeq2 object. The result will then be shown as a graph using the graphical library ggplot2.

4. Format `pathoscope_table.csv` and `metadata.txt` and create a phyloseq object.

   To do this, in the R console type:

```
1 df<-read.csv("pathoscope_table.csv",header =
  T,check.names = F)
2 metad<-read.table("metadata.txt",header =
  T,row.names = 1,sep = "\t",check.names = F)
```

   The two tables are loaded into R variables through these two commands.

```
3 metad<-sample_data(metad)
4 tax<-tax_table(as.matrix(df[,c(1:7)]))
5 otu<-otu_table(as.matrix(df[,c(9:ncol(df))]),
  taxa_are_rows = T)
6 phyob<-phyloseq(otu,tax,metad)
```

   The relevant sample data are collected using the information provided in the metadata table. Taxon name and quantities are extracted from the PathoScope table that was generated in the previous steps. The information is then incorporated to a phyloseq object.

5. Convert the phyloseq object to DESeq2 object and estimate size factors.

   To do this, in the R console type:

```
1 sample_data(phyob)$host_disease <-
  relevel(sample_data(phyob)$host_disease, "healthy")
2 diagdds = phyloseq_to_deseq2(phyob, ~ host_disease)
```

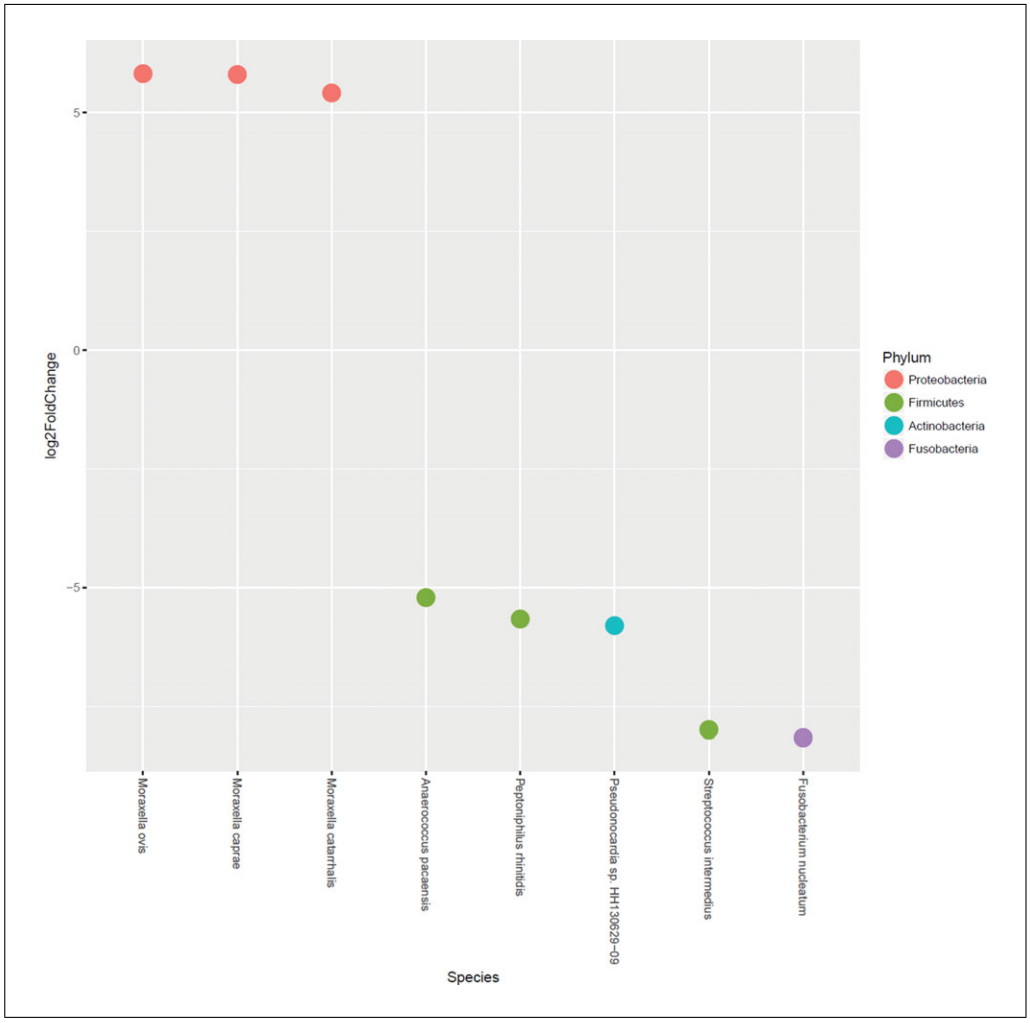


Figure 1E.14.8 Differential abundance test for bacteria using Phyloseq and DESeq2. The graph represents log-fold change in the abundance of certain species in asthmatic patients compared to healthy controls. A positive value signifies increased presence in patients with asthma. Three proteobacteria; *Moraxella catarrhalis*, *M. caprae* and *M. ovis*, show a much stronger presence in asthmatic patients than in controls.

```
3 diagdds = DESeq(diagdds, test="Wald",
  fitType="local")
```

This will create the DESeq2 object that will be used to run a test on the samples, divided by health status.

6. Check results, removing NA (not available) values, and order by the adjusted *p* value.

   To do this, in the R console type:

```
1 res = results(diagdds, cooksCutoff = FALSE)
2 alpha = 0.01
```

alpha represents our p value cutoff threshold.

```
3 sigtab = res[which(res$padj < alpha),]
```

Only show results where the p value is below the cutoff threshold.

```
4 sigtab = cbind(as(sigtab, "data.frame"),
  as(tax_table(phyob)[rownames(sigtab),], "matrix"))
5 head(sigtab)
```

Save the data frame `sigtab` *as a matrix and check content with the head command.*

7. Order phylum and genus taxonomic classes

   To do this, in the R console type:

```
x = tapply(sigtab$log2FoldChange, sigtab$Phylum,
function(x) max(x))
x = sort(x, TRUE)
sigtab$Phylum = factor(as.character(sigtab$Phylum),
levels=names(x))
```

   This series of functions orders the phylum.

```
x = tapply(sigtab$log2FoldChange, sigtab$Species,
function(x) max(x))
x = sort(x, TRUE)
sigtab$Species = factor(as.character
(sigtab$Species), levels=names(x))
```

   This series of functions orders the species genus.

8. Save the results to disk and plot the results using ggplot2.

   To do this, in the R console first type:

```
write.table(sigtab,"differential_species_abundance
.tsv",quote = F,sep = "\t",row.names = F)
```

   This saves the data in a tab delimited file for future reference.

   Then type:

```
ggplot(sigtab, aes(x=Species, y=log2FoldChange,
  color=Phylum)) + geom_point(size=6) +
theme(axis.text.x = element_text(angle = -90, hjust =
  0, vjust=0.5))
```

   We input the table created in the previous steps followed by a number of parameters that determine the appearance of our final graph. The graph thus obtained should look like Figure 1E.14.8.

COMMENTARY

Background Information

The role that microbiota—the microbial communities inhabiting the human body—can play in disease has long been overlooked, coming to prominence only in the last decade (Young, 2017). This renewed interested in the study of the human microbiome also coincides with the development of high-throughput sequencing platforms that enable the analysis of whole microbial communities, including those that are unculturable (Bik, 2016). The number of computational methods for turning microbiome data into biological insight has also increased accordingly.

With so many tools and methodologies now at the disposal of researchers, it is important to provide some clear examples of end-to-end analysis of human microbiome data. This is what we aimed to provide with the above series of protocols. Additionally, it is important to be able to rely on a concise set of tools rather than having several mutually incompatible tools that each execute a small task. Both PathoScope 2.0 and PathoStat were developed with the objective of aggregating several of the steps necessary to the study of microbiota (Hong et al., 2014; Manimaran et al., 2016). In this unit, PathoScope 2.0 delivers two important read mapping stages. Researchers are able to match reads to the most appropriate taxonomic unit, providing an initial alignment of reads to microbial references followed by a re-assessment of some of the ambiguous mappings (Hong et al., 2014). Likewise, PathoStat provides a single source for an entire range of statistical and enumerative descriptions of the data (Manimaran et al., 2016). Graphical and textual information is accessed in a single

browser window once PathoStat is launched. Researchers can jump between statistical representations of the data in their pursuit of biologically meaningful relations between microbiota and disease. There are plans in the future to further integrate PathoScope 2.0 and PathoStat into a single bundle (pers. comm.).

It is important to keep in mind the current limitations of the tools that we presented in this unit. While PathoStat has most of the elements necessary to test differential abundance in microbes, we have deemed it prudent to introduce this final step as a separate basic protocol, using the well-established Phyloseq (McMurdie & Holmes, 2013). However, it is expected that future iterations of PathoStat will better incorporate differential abundance testing, thus offering the entire gamut of useful statistical tests for human microbiome analysis.

Critical Parameters

The example sequences used in these protocols have a relatively low storage footprint. This might not always be the case with human microbiome data. In order to use the tools described in this section with larger datasets, researchers are encouraged to apply digital normalization to the data between quality control and the first round of read mapping (Basic Protocols 1 and 2).

As noted in Basic Protocol 4, the relevance of additional statistical tests in PathoStat will depend largely on the research question a particular study is attempting to elucidate. While following the protocol, researchers should therefore choose which additional tests to explore that might not be present in this protocol.

A number of programs mentioned in the protocol have an option for setting the number of threads the program uses. We suggest that researchers set the number of threads to correspond to the constraints of the HPC cluster they are using (e.g., 2 × number of CPUs).

Troubleshooting

Every program used in this unit comes with documentation that can be retrieved from the installation sources provided at the start of every protocol. Additionally, any tool loaded into R comes with documentation, which can be invoked in the R console. The commands necessary to read documentation are given as part of the installation process.

Anticipated Results

Provided that the different protocols are followed faithfully when executed on the example data, the results should always be the same for each protocol. Files corresponding to the input and output of different stages of protocols are provided for comparison. Results will obviously differ if researchers elect to use their own data.

Time Considerations

While most protocols can be comfortably completed within an hour, completion of protocols involving one or more read mapping procedures—Basic Protocol 2 and 3—will depend largely on the capacity of the elected HPC cluster to handle these tasks. The unit should therefore be performed intermittently over a few days.

Acknowledgements

We would like to thank George Washington University's high-performance computing facility, Colonial One, for providing data storage, support, and computing power for analyses (*http://colonialone.gwu.edu/*). ECN was funded by FONDECYT de Iniciación and PAI grants number 11160905 and 82140008, respectively.

Literature Cited

Andrews, S. (2010). FastQC: A quality control tool for high throughput sequence data. Available at *http://www.bioinformatics.babraham.ac.uk/projects/fastqc*.

Bik, E. M. (2016). The hoops, hopes, and hypes of human microbiome research. *Yale Journal of Biology and Medicine*, *89*, 363–373.

Castro-Nallar, E., Shen, Y., Freishtat, R. J., Pérez-Losada, M., Manimaran, S., Liu, G., ... Crandall, K. A. (2015). Integrating microbial and host transcriptomics to characterize asthma-associated microbial communities. *BMC Medical Genomics*, *8*, 50. doi: 10.1186/s12920-015-0121-1.

Francis, O. E., Bendall, M., Manimaran, S., Hong, C., Clement, N. L., Castro-Nallar, E., ... Johnson, W. E. (2013). Pathoscope: Species identification and strain attribution with unassembled sequencing data. *Genome Research*, *23*, 1721–1729. doi: 10.1101/gr.150151.112.

Hong, C., Manimaran, S., Shen, Y., Perez-Rogers, J. F., Byrd, A. L., Castro-Nallar, E., ... Johnson, W. E. (2014). PathoScope 2.0: A complete computational framework for strain identification in environmental or clinical sequencing samples. *Microbiome*, *2*, 33. Available at *https://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4164323&tool=pmcentrez&rendertype=abstract* doi: 10.1186/2049-2618-2-33.

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, *9*, 357–359. doi: 10.1038/nmeth.1923.

Manimaran, S., Bendall, M., Valenzuela, S., Castro, E., Faits, T., & Johnson, W. E. (2016). PathoStat: PathoStat statistical microbiome analysis

package. *R package version* 1.2.0. Available at http://bioconductor.org/packages/release/bioc/html/PathoStat.html.

McMurdie, P. J., & Holmes, S. (2013). Phyloseq: An R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One*, *8* doi: 10.1371/journal.pone.0061217.

Schmieder, R., & Edwards, R. (2011). Quality control and preprocessing of metagenomic datasets. *Bioinformatics*, *27*, 863–864. doi: 10.1093/bioinformatics/btr026.

The Human Microbiome Project. (2012). Structure, function and diversity of the healthy human microbiome. *Nature*, *486*, 207–214. Available at *https://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3564958&tool=pmcentrez&rendertype=abstract* doi: *10.1038/nature11234*.

Young, V. B. (2017). The role of the microbiome in human health and disease: An introduction for clinicians. *BMJ*, *356*, j831. Available at *https://www.bmj.com/lookup/doi/10.1136/bmj.j831* doi: 10.1136/bmj.j831.