Visualization of Metagenomics Data

LeAnn Lindsey, Kimberly Truong, Lourdes Valdez
Final Project, CS 6630
Fall 2020

Process Book

Table of contents:

Overview and Motivation

**Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.*

Advances in genomic sequencing technology have provided scientists with vast quantities of data to investigate scientific questions. It is now possible to obtain DNA and RNA sequence data not only for a host, but also to obtain metagenomic sequencing for all of the microbes within a specific host site, such as a mammalian gut. This metagenomic sequencing provides not only the genome sequences of the microorganisms present in the system, but also provides insight into the abundances of each species, and a phylogenetic tree of species present. It is known that the presence and abundance of specific species of microorganisms are linked to metabolic disease, autoimmune responses, pathogen detection and toxin metabolism. This metagenomic sequencing data is extremely rich but complex, and difficult to mine for information. Biologists often need to sift through this data searching for a specific gene or pathway which may be upregulated or downregulated under specific conditions and of interest in their research.

Our visualization project is to provide a useful tool to help scientists explore a metagenomic data set. Denise Dearing's laboratory studies microbial detoxification, and we have partnered with her lab to visualize a specific data set acquired last summer by Rodolfo Martinez-Mota, which explores microbial cardenolide detoxification in wild black-eared mice. Monarch butterflies have evolved a resistance to plant toxins, specifically, they feed on milkweed which has high levels of cardenolides. Monarch butterflies migrate each year to overwintering sites in the southern United States and Mexico and during this migration season, predators such as the wild black-eared mouse feed on the butterflies. The goals of the study are to determine the role of the gut microbiome of black-eared mice on detoxification of the cardenolides ingested with a monarch based diet. The Dearing lab has investigated the role of the microbiome using 16S rRNA gene marker sequencing, which provides some information about the species abundances and genes of interest, but they have not yet fully processed and analyzed the metagenomic data set acquired in the study.

While many different tools for visualizing genomic data do exist, very few are interactive enough to be useful during the research process and are more often used to create high quality images for publication. A brief list of the challenges with the current visualizations are listed below.

Challenges with Current Metagenomic Visualization Tools:

- Only a few genomic visualization tools have been extended to visualize metagenomic data, which has a higher level of complexity than genomic data.
- Do not allow a scientist to easily visualize two different experimental conditions at the same time
- Difficult to easily obtain functional pathway information from a gene
- Difficult to install and get working on various software systems, require time to learn to use and interpret output

- Many options available but most only provide one type of visualization

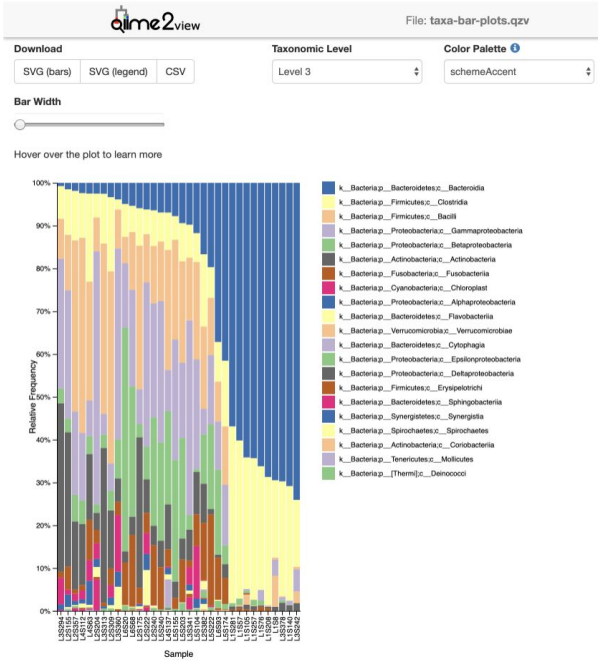
Project Objectives

1. To make an interactive tool which can be used by a researcher to explore metagenomic data from the level of family to species.
2. To connect a specific species to the genes and functional pathways present in that species.
3. To make clear visualizations of sample variation and the differences between two experimental conditions.
4. To explore and visualize interesting aspects of this specific Monarch dataset

Related Work

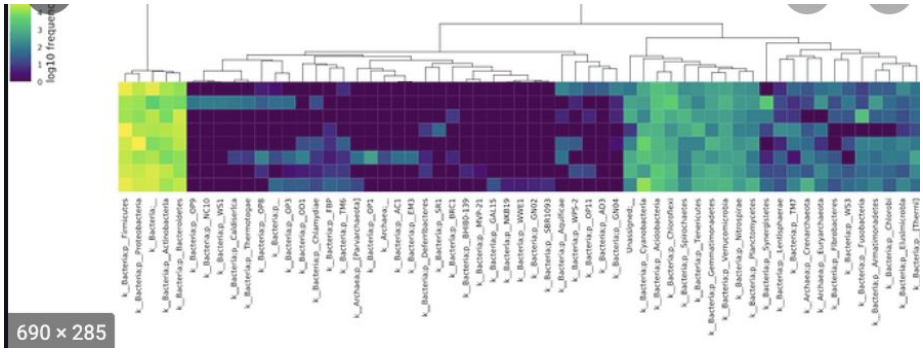
**Anything that inspired you, such as a paper, a website, visualizations we discussed in class, etc.*

This is the Qiime2 website which is commonly used by metagenomic researchers.



ColorSchemes Heatmap

For the heatmap, we looked at several color schemes, and chose this one as our favorite.

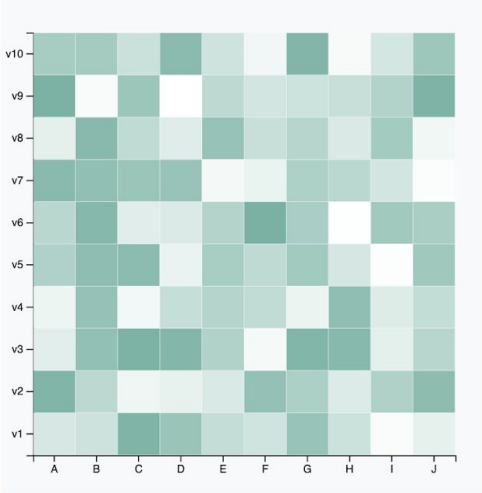


Color Schemes Stacked Bar

Heatmap

The d3 and javascript code for the heatmap was modeled after this example from the d3-graph-gallery

https://www.d3-graph-gallery.com/graph/heatmap_basic.html



Steps:

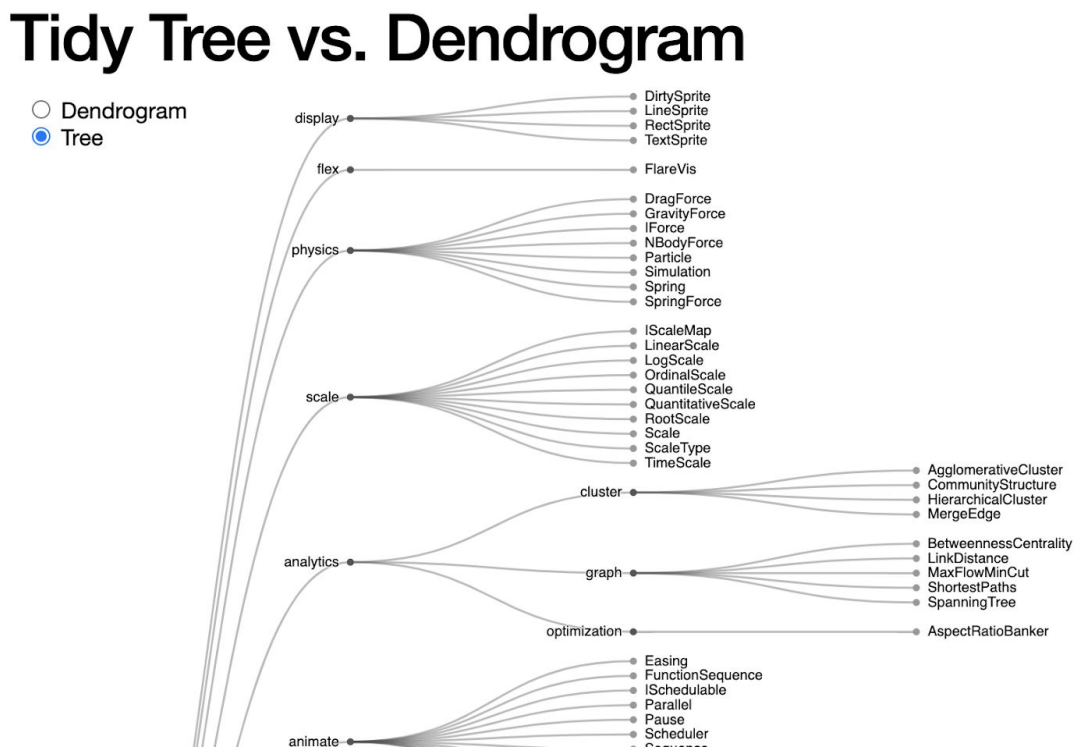
- The Html part of the code just creates a `div` that will be modified by d3 later on.
- The first part of the javascript code set a `svg` area. It

```
<!DOCTYPE html>
<meta charset="utf-8">

<!-- Load d3.js -->
<script src="https://d3js.org/d3.v4.js"></script>

<!-- Create a div where the graph will take place -->
<div id="my_dataviz"></div>
```

```
<script>

// set the dimensions and margins of the graph
var margin = {top: 30, right: 30, bottom: 30, left: 30},
  width = 450 - margin.left - margin.right,
  height = 450 - margin.top - margin.bottom;

// append the svg object to the body of the page
var svg = d3.select("#my_dataviz")
.append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
.append("g")
  .attr("transform",
        "translate(" + margin.left + "," + margin.top + ")");

// Labels of row and columns
var myGroups = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"]
var myVars = ["v1", "v2", "v3", "v4", "v5", "v6", "v7", "v8", "v9", "v10"]
```

This example is similar but adds a tooltip
https://www.d3-graph-gallery.com/graph/heatmap_tooltip.html
This example was used to improve the hover effect
https://www.d3-graph-gallery.com/graph/heatmap_style.html

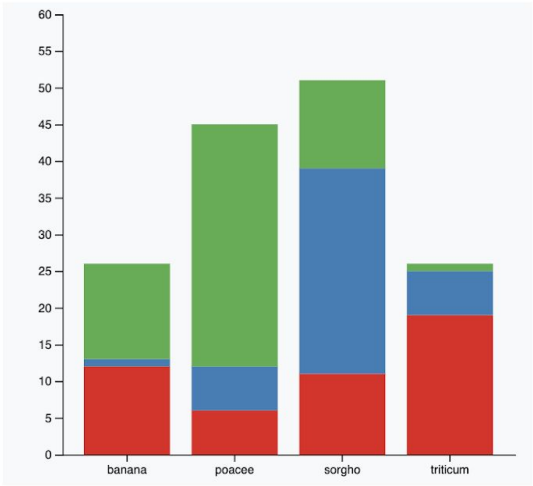
Tree



The original tree was inspired by this example from Mike Bostock
https://bl.ocks.org/mbostock/e9ba78a2c1070980d1b530800ce7fa2b

Stacked Bar Chart

The original bar chart code and data structure was inspired by the basic stacked bar in the d3-graph-gallery



Steps:

- Start by understanding the basics of barplot in d3.js.

```
<!DOCTYPE html>
<meta charset="utf-8">

<!-- Load d3.js -->
<script src="https://d3js.org/d3.v4.js"></script>

<!-- Create a div where the graph will take place -->
<div id="my_dataviz"></div>
```

```
<script>

// set the dimensions and margins of the graph
var margin = {top: 10, right: 30, bottom: 20, left: 50},
    width = 460 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// append the svg object to the body of the page
var svg = d3.select("#my_dataviz")
  .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
```

https://www.d3-graph-gallery.com/graph/barplot_stacked_basicWide.html

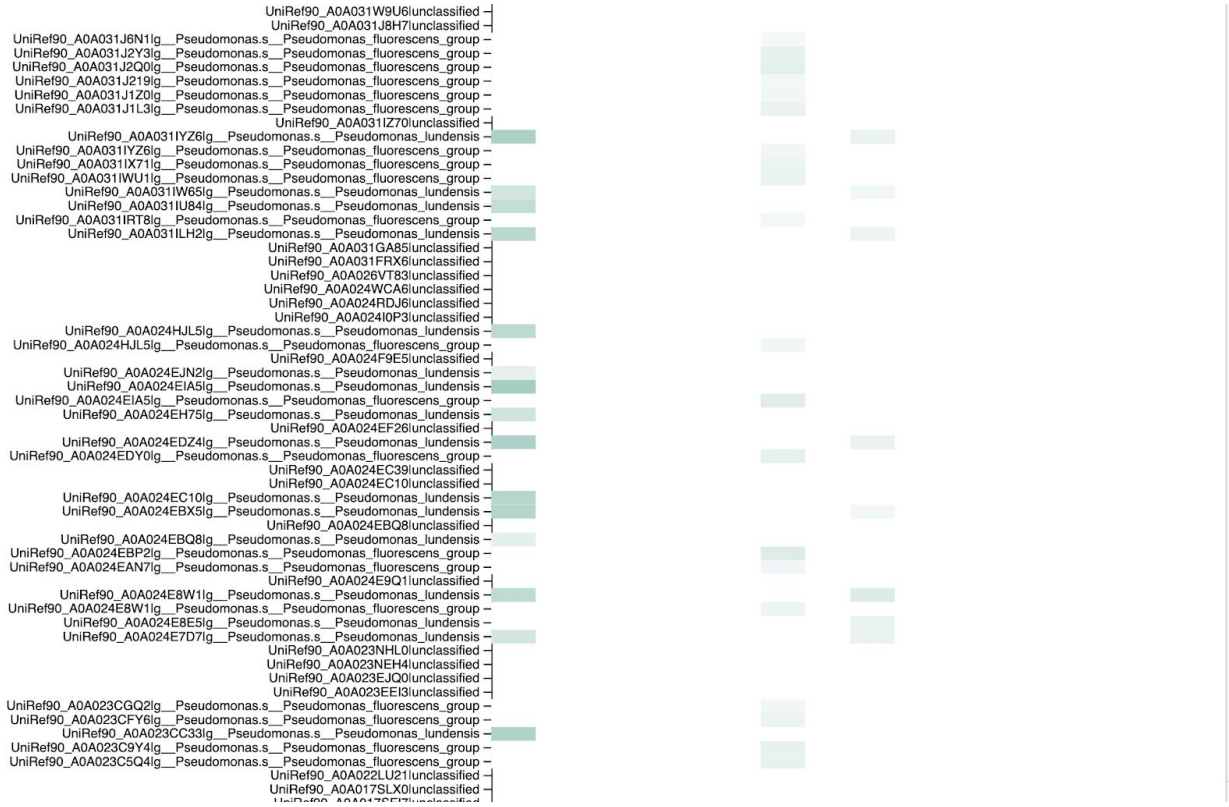
Questions

**What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?*

- What is the role of the black-eared mice gut microbiome in cardenolide detoxification?
- What are the differences in the gut microbiome of black-eared mice on a Monarch butterfly diet vs on a non-Monarch butterfly diet?
- Are any genes up- or down-regulated in mice with Monarch vs non-Monarch diets?

New questions

Why do some samples seem to have higher abundance in many species (seen by the striping below), while other samples have none of those species?


UniRef90_A0A031W9U6|unclassified
UniRef90_A0A031J8H7|unclassified
UniRef90_A0A031J6N1|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031J2Y3|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031J2Q0|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031J219|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031J1Z0|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031J1L3|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031IZ70|unclassified
UniRef90_A0A031IYZ6|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A031IYZ6|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031IX71|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031IWU1|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031IW65|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A031IU84|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A031IRT8|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A031ILH2|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A031GA85|unclassified
UniRef90_A0A031FRX6|unclassified
UniRef90_A0A026VT83|unclassified
UniRef90_A0A024WCA6|unclassified
UniRef90_A0A024RDJ6|unclassified
UniRef90_A0A024I0P3|unclassified
UniRef90_A0A024HJL5|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024HJL5|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024F9E5|unclassified
UniRef90_A0A024EJN2|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EIA5|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EIA5|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024EH75|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EF26|unclassified
UniRef90_A0A024EDZ4|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EDY0|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024EC39|unclassified
UniRef90_A0A024EC10|unclassified
UniRef90_A0A024EC10|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EBX5|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EBQ8|unclassified
UniRef90_A0A024EBQ8|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024EBP2|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024EAN7|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024E9Q1|unclassified
UniRef90_A0A024E8W1|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024E8W1|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A024E8E5|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A024E7D7|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A023NHL0|unclassified
UniRef90_A0A023NEH4|unclassified
UniRef90_A0A023EJQ0|unclassified
UniRef90_A0A023EEI3|unclassified
UniRef90_A0A023CGQ2|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A023CFY6|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A023CC33|g__Pseudomonas.s__Pseudomonas_lundensis
UniRef90_A0A023C9Y4|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A023C5Q4|g__Pseudomonas.s__Pseudomonas_fluorescens_group
UniRef90_A0A022LU21|unclassified
UniRef90_A0A017SLX0|unclassified

Data

**Source, scraping method, cleanup, etc.*
Source

Metagenomic Shotgun Sequencing Data collected from 14 samples of wild black-eared mice under the following experimental conditions:

- Monarch/Wild 6 samples
- Non-Monarch/Wild 2 samples
- Monarch/Experiment 2 samples
- Non-Monarch/Experiment 4 samples

Samples labeled “Monarch/Wild” were collected in the wild during Monarch season and are assumed to have eaten Monarch butterflies as a part of their normal foraging.

Samples labeled “Non-Monarch/Wild” were collected during the Non-Monarch season and are assumed to have not eaten Monarch butterflies during that time period.

Samples labeled “Monarch/Experiment” were taken from mice that were captured during Monarch season and fed a diet of 5 Monarch butterflies per day for a period of 48 hours.

Samples labeled “Non-Monarch/Experiment” were captured during Non-Monarch season and taken into captivity for the same period of 48 hrs but were not fed Monarch butterflies.

Data Types:

- Metagenomic Sequence data obtained from processing with Humann3 software from the Huttenhower Lab, Harvard University, to obtain gene family abundances, and functional pathway abundances (.tsv files)
- Taxonomy data obtained from MetaPhlan2 (.csv file)

Data Processing

- Taxonomy data needed to be processed into the proper format that has parent and child nodes for use in d3.
- Metagenomic data is too large to visualize, so we will perform feature reduction using Singular Value Decomposition, a numerical linear algebra technique which identifies the most important components in the data. We must reduce the data from 280k rows to 10k rows using features of interest. Update: need to reduce to 1000 rows because 10k is too large to render

Tree Data Cleaning

11/5/2020

For the Tree, we needed a specific format of the data so that it could be read in properly into the Node format.

The raw data was in this format:

```
leannmlindsey@LeAnns-MacBook-Pro raw % cat merged_abundance_table.txt
#mpa_v30_CHOCOPhlAn_201901
clade_name	NCBI_tax_id	RM-261_S63_L001_R2_001.trimmed_profile	RM-261_S63_L001_R1_001.trimmed_profile	RM-260_S55_L001_R2_001.trimmed_pr
le	RM-258_S39_L001_R1_001.trimmed_profile	RM-257_S31_L001_R2_001.trimmed_profile	RM-257_S31_L001_R1_001.trimmed_profile	RM-256_S23_L001_R
2_001.trimmed_profile	RM-254_S7_L001_R1_001.trimmed_profile	RM-253_S94_L001_R2_001.trimmed_profile	RM-253_S94_L001_R1_001.trimmed_profile
RM-250_S70_L001_R2_001.trimmed_profile	RM-250_S70_L001_R1_001.trimmed_profile	RM-249_S62_L001_R_2un.trimmed_profile	RM-249_S62_L001_R_2.t
rofile	RM-248_S54_L001_R2_001.trimmed_profile	RM-248_S54_L001_R1_001.trimmed_profile
UNKNOWN	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria	2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	1
k__Bacteria|p__Bacteroidetes	2|976	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria|p__Bacteroidetes|c__Bacteroidia	2|976|200643	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria|p__Bacteroidetes|c__Bacteroidia|o__Bacteroidales	2|976|200643|171549	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria|p__Bacteroidetes|c__Bacteroidia|o__Bacteroidales|f__Tannerellaceae	2|976|200643|171549|2005525	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria|p__Bacteroidetes|c__Bacteroidia|o__Bacteroidales|f__Tannerellaceae|g__Parabacteroides	2|976|200643|171549|2005525|375288	0
k__Bacteria|p__Bacteroidetes|c__Bacteroidia|o__Bacteroidales|f__Tannerellaceae|g__Parabacteroides|s__Parabacteroides_goldsteinii	2|976|200
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k__Bacteria|p__Firmicutes	2|1239	0	0	99.79205999999999	99.79517	100.0	100.0	14.19578	12.92446	2
100.0	100.0	100.0	0	0	0	0	0	0	0	100.0	100.0
k__Bacteria|p__Firmicutes|c__Bacilli	2|1239|91061	0	0	99.79205999999999	99.79517	100.0	100.0	14.19578	1
100.0	100.0	100.0	100.0	0	0	0	0	0	0	100.0	100.0
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales	2|1239|91061|186826	0	0	99.79205999999999	99.79517	100.0	1
100.0	77.25248	74.3981	100.0	100.0	100.0	100.0	0	0	0	0	0	0	0	100.0	100.0
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales|f__Enterococcaceae	2|1239|91061|186826|81852	0	0	0	0.10613	0
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales|f__Enterococcaceae|g__Enterococcus	2|1239|91061|186826|81852|1350	0	0	0
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales|f__Enterococcaceae|g__Enterococcus|s__Enterococcus_durans	2|1239|91061|186826|81852
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales|f__Lactobacillaceae	2|1239|91061|186826|33958	0	0	99.79205999999999
100.0	100.0	77.25248	74.3981	100.0	100.0	100.0	100.0	0	0	0	0	0	0	100.0	100.0
k__Bacteria|p__Firmicutes|c__Bacilli|o__Lactobacillales|f__Lactobacillaceae|g__Lactobacillus	2|1239|91061|186826|33958|1578	0	0	9
```

I used a stream editor in UNIX (SED) to reformat the data to CSV format with a “.” separating the Family, Genus, Species etc.

```
cat merged_abundance_table.txt | sed 's/k__//' | sed 's/|.__/./g'| sed
's/\t/,/g' | cut -f 2 --complement -d ',' > taxonomyInputFile2.csv
```

```
cat merged_abundance_table.txt       streams contents of the file to the screen
sed 's/k__//'                        select “k__” and remove it
sed 's/|.__/./g'                     select the delimiter of other levels
                                     “|s__” for example and replace with “.”
sed 's/\t/,/g'                       replace all of the tabs with commas
cut -f 2 --complement -d ','         selects all columns except column 2
```

```
[u1323098@notchpeak2:metaphlan_output1]$ cat taxonomyInputFile.txt
#mpa_v30_CHOCOPhlAn_201901
clade_name	NCBI_tax_id	RM-261_S63_L001_R2_001.trimmed_profile	RM-261_S63_L001_R1_001.trimmed_profile	RM-260_S55_L001_R2_001.trimmed_profile	RM-260_S55_L001_R
L001_R2_001.trimmed_profile	RM-259_S47_L001_R1_001.trimmed_profile	RM-258_S39_L001_R2_001.trimmed_profile	RM-258_S39_L001_R1_001.trimmed_profile	RM-257_S31_L001_
_L001_R1_001.trimmed_profile	RM-256_S23_L001_R2_001.trimmed_profile	RM-256_S23_L001_R1_001.trimmed_profile	RM-255_S15_L001_R2_001.trimmed_profile	RM-255_S15_L00
_L001_R2_001.trimmed_profile	RM-254_S7_L001_R1_001.trimmed_profile	RM-253_S94_L001_R2_001.trimmed_profile	RM-253_S94_L001_R1_001.trimmed_profile	RM-252_S86_L0
86_L001_R1_001.trimmed_profile	RM-251_S78_L001_R2_001.trimmed_profile	RM-251_S78_L001_R1_001.trimmed_profile	RM-250_S70_L001_R2_001.trimmed_profile	RM-250_S70_L0
S62_L001_R_2un.trimmed_profile	RM-249_S62_L001_R_2.trimmed_profile	RM-249_S62_L001_R2_001.trimmed_profile	RM-249_S62_L001_R_1un.trimmed_profile	RM-249_S62_
_S62_L001_R1_001.trimmed_profile	RM-248_S54_L001_R2_001.trimmed_profile	RM-248_S54_L001_R1_001.trimmed_profile
UNKNOWN	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100.0	100.0	100.0	100.0	100.0	100.0	0	0
Bacteria	2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	1
.0	100.0	0	0	100.0	100.0
Bacteria.Bacteroidetes,976	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0
Bacteria.Bacteroidetes.Bacteroidia,976|200643	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25.6019	0	0	0
Bacteria.Bacteroidetes.Bacteroidia.Bacteroidales,976|200643|171549	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	22.74752	25.6019	0	0	0	0	0	0	0	0	0
Bacteria.Bacteroidetes.Bacteroidia.Bacteroidales.Tannerellaceae,976|200643|171549|2005525	0	0	0	0	0	0	0	0	0	0	0
0	0	0	22.74752	25.6019	0	0	0	0	0	0	0	0	0	0
Bacteria.Bacteroidetes.Bacteroidia.Bacteroidales.Tannerellaceae.Parabacteroides,976|200643|171549|2005525|375288	0	0	0	0	0	0	0
0	0	0	22.74752	25.6019	0	0	0	0	0	0	0	0	0	0	0
Bacteria.Bacteroidetes.Bacteroidia.Bacteroidales.Tannerellaceae.Parabacteroides.Parabacteroides_goldsteinii,976|200643|171549|2005525|375288|328812	0	0	0
0	0	0	22.74752	25.6019	0	0	0	0	0	0	0	0	0	0	0
Bacteria.Firmicutes,1239	0	0	99.79205999999999	99.79517	100.0	100.0	14.19578	12.92446	24.34915	25.19431	8
0452	57.90693	97.34869	97.63207	100.0	100.0	77.25248	74.3981	100.0	100.0	100.0	100.0	0	0	0	0	0	0
```

We also had to shorten the sample names and add the experimental condition to the name.

11/6/2020
We realized that the original data output was too simplistic and only contained a few species. We reprocessed the data with MetaPhlan with a lower threshold to pick up less abundant species and then used the above SED command to reprocess the data. This increased the number of rows in the Tree table from 34 to 136.

11/14/2020
kingdom, phylum or division, class, order, family, genus, species

We realized after the tree was built that we needed to add two more columns to the csv file, so that it would be easier to filter the data for the stacked bar chart and tree map. We needed to either add a field that shows the clade (kingdom, phylum...species), and then the name at that level, or be able to process the data in javascript to give that information.

Stacked Bar Data Cleaning

11/14/2020
The stacked bar chart required the original data file to be transposed. We are still deciding whether it would be better to do that in pre-processing of the data and send in two different csv files, or if we can use d3.transpose to do that.

11/15/2020
After meeting with Youija we decided to transpose the data outside of javascript and then send it into javascript as a different csv file. In total we sent in three different csv files, one for each visualization (tree, stacked bar chart, and heat map).

HeatMap Data Cleaning

The two files that will be used in the heatmap are:

Combined_genefamilies_stratified.tsv	Number of Genes 280321
Combined_pathcoverage_stratified.tsv	Number of Gene Pathways 841

The genefamilies file is too large to show in the heatmap, which after some trial and error, we decided to limit to 1000 rows. We needed to make some decisions on how to filter the data and select which genes we wanted to include in the heatmap. We considered several options for limiting the data.

1. Remove all “unclassified” genes
2. Remove all genes that are not present in at least ½ of the samples

3. Use Singular Value Decomposition to determine the most important features in the data
4. Using EdgeR to identify the most differentially expressed genes

We discussed the options to select the data and these were the advantages and disadvantages of each one.

1. Remove all “unclassified” genes
   Advantages: Simple
   Disadvantages: Unclassifiied and low abundance species may be the species of interest to the researcher
2. Remove all genes that are not present in at least ½ of the samples
   Advantages: Easy to implement
   Disadvantages: This option will favor genes that are present in every sample, which may not be the best way to see differentially expressed genes, which are of most interest to researchers.
3. Use Singular Value Decomposition to determine the most important features in the data
   Advantages: Easy to implement
   Disadvantages:
4. Using EdgeR to identify the most differentially expressed genes
   Advantages: Most informative for researchers and familiar to them
   Disadvantages: More difficult and time consuming to to implement

The gene pathway file did not need to be reduced in size to be shown in the heatmap.

The format of both files had to be changed so that d3 could properly bind the data to end up with the correct number of rectangles. The format had to be changed from a matrix format with genes on the rows and samples on the columns, to a long format that has one row for each rectangle in the heatmap:

Column, Row, Value
Gene1,Sample1,Value
Gene1,Sample2,Value
Gene2,Sample1,Value
Gene2,Sample2,Value

We used python to make this change (see python notebook in github for the code).

Exploratory Data Analysis:

**What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?*

Singular Value Decomposition

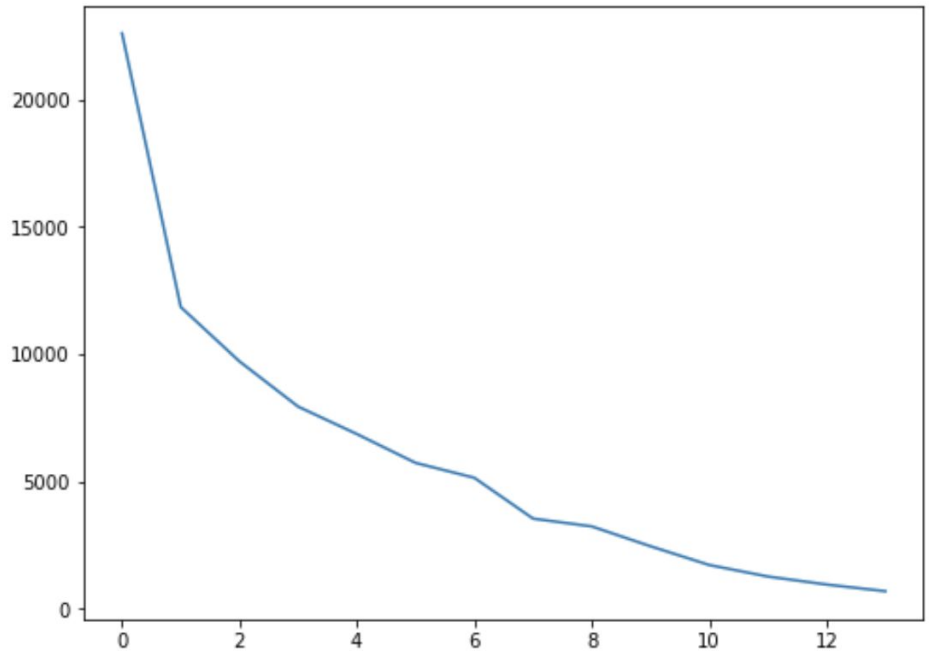
The most important data manipulation that we needed to perform was to reduce the genefamilies file to a reasonable number to visualize. We plotted the singular values from the Singular Value Decomposition to make an appropriate cutoff threshold.

Here is a graph of the flattened s diagonal vector of the SVD of the Monarch genefamilies data file.

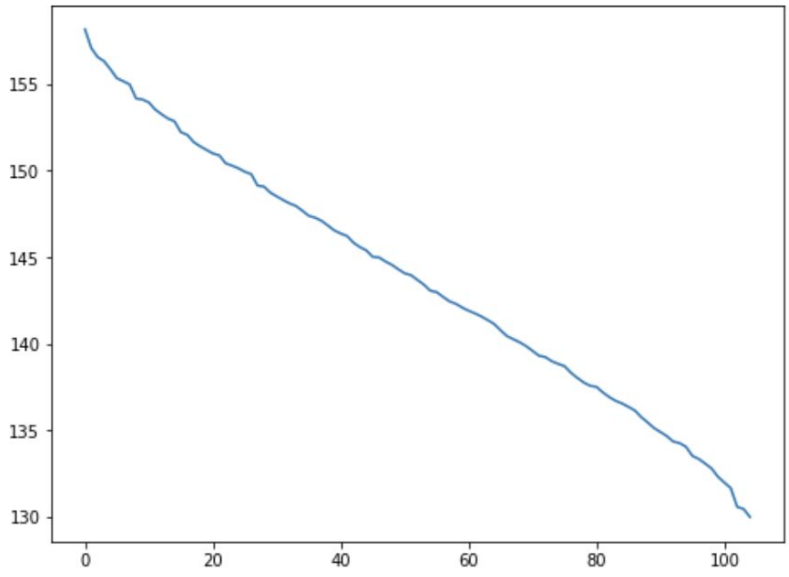
In [79]:
```
from scipy import linalg
U, s, Vh = linalg.svd(features)
U.shape,  s.shape, Vh.shape
```

Out[79]:
```
((22926, 22926), (14,), (14, 14))
```

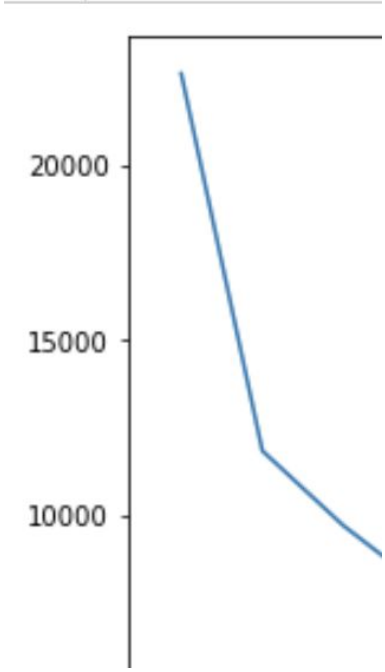
In [80]:
```
# 14 Monarch samples of mixed classes singularity graph
fig2 = plt.figure(figsize=(8,6))
#print(s)
plt.plot(s)
plt.show()
```



As a comparison, here is the same plot for a random matrix.



You can see that there is structure in the data, compared with random, with the significant columns being in this first left section of the graph.

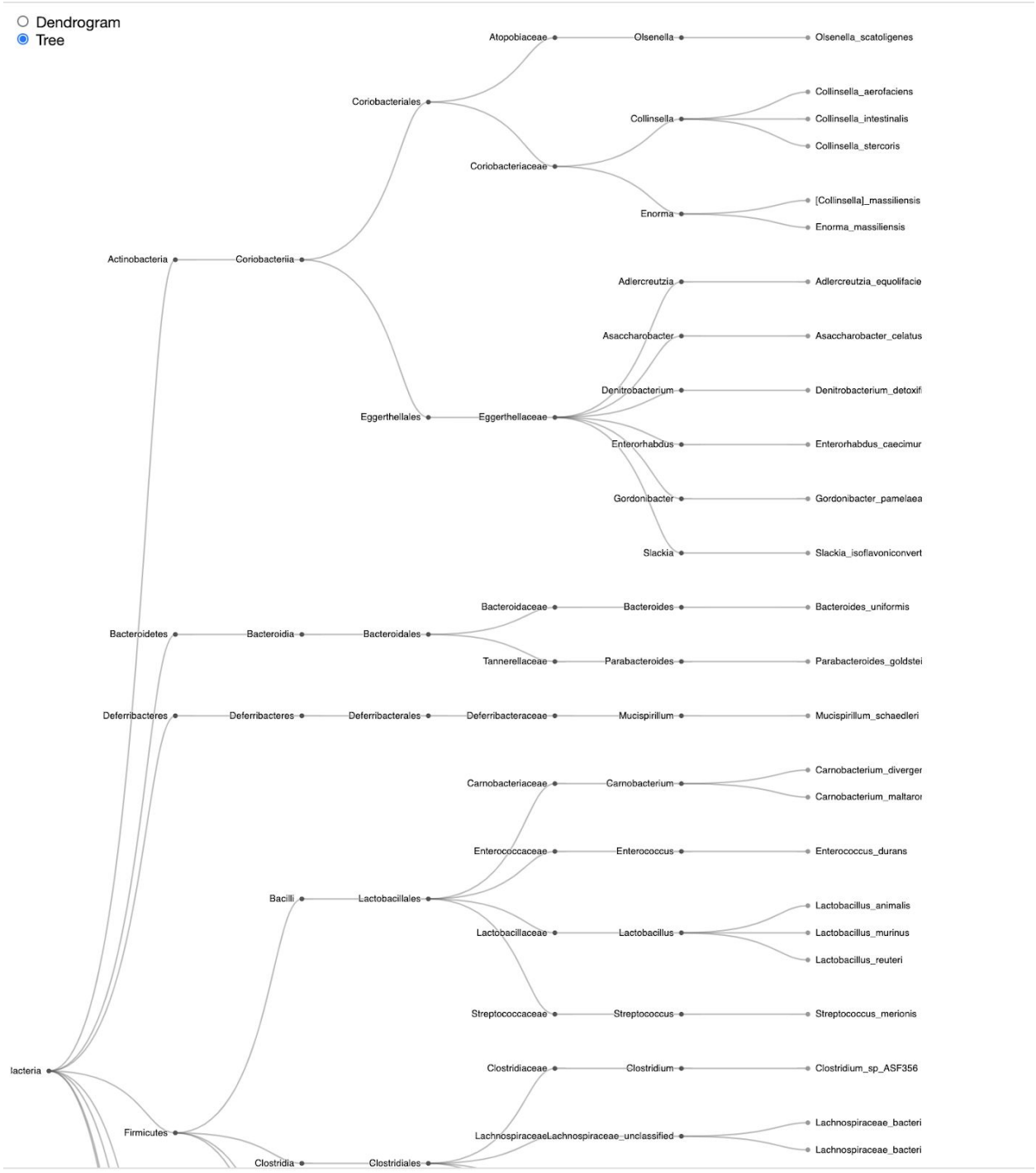


The cutoff threshold will need to be the top 1000 genes, as it is difficult to see more than that on one screen.

Design Evolution

**What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?*

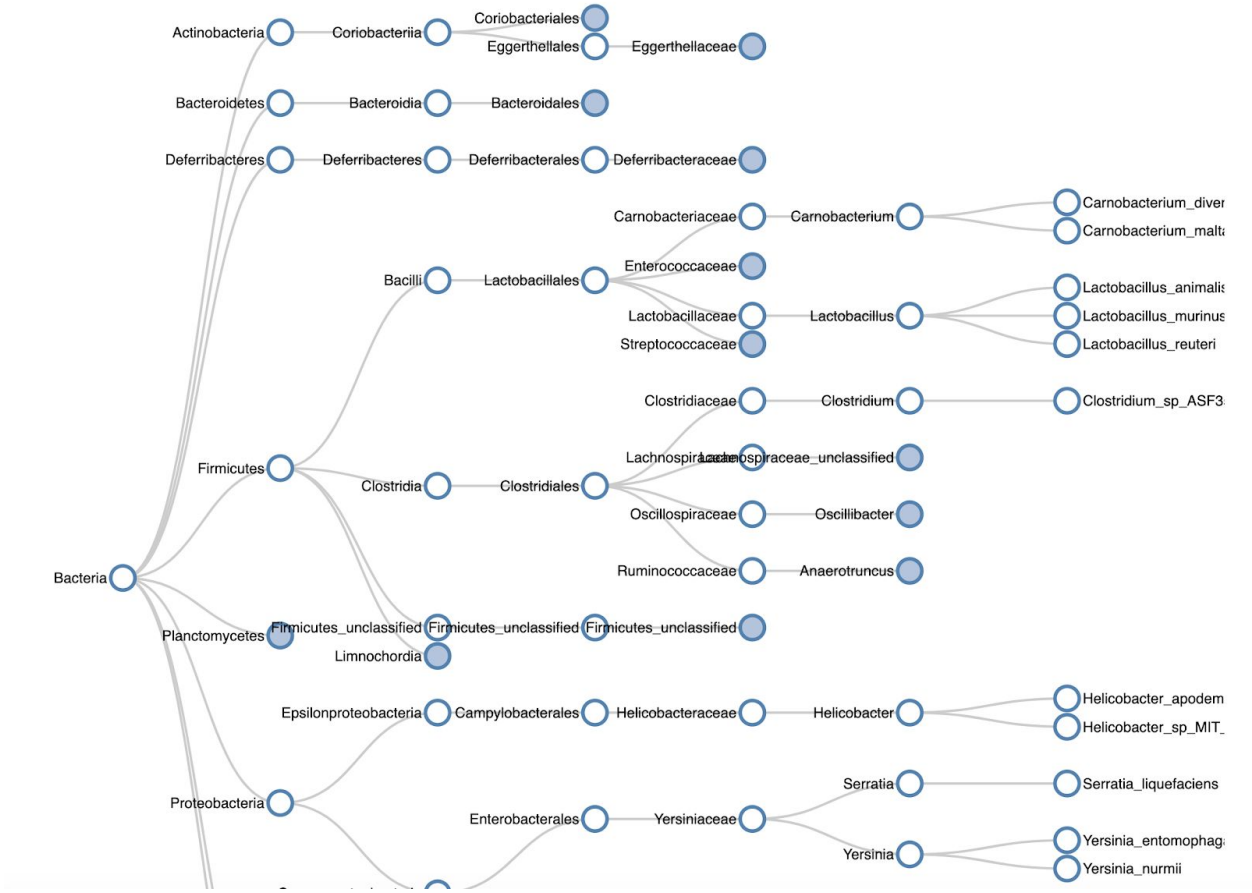
Tree v1 Flat, No interaction



Tree v2-Interactive Nodes

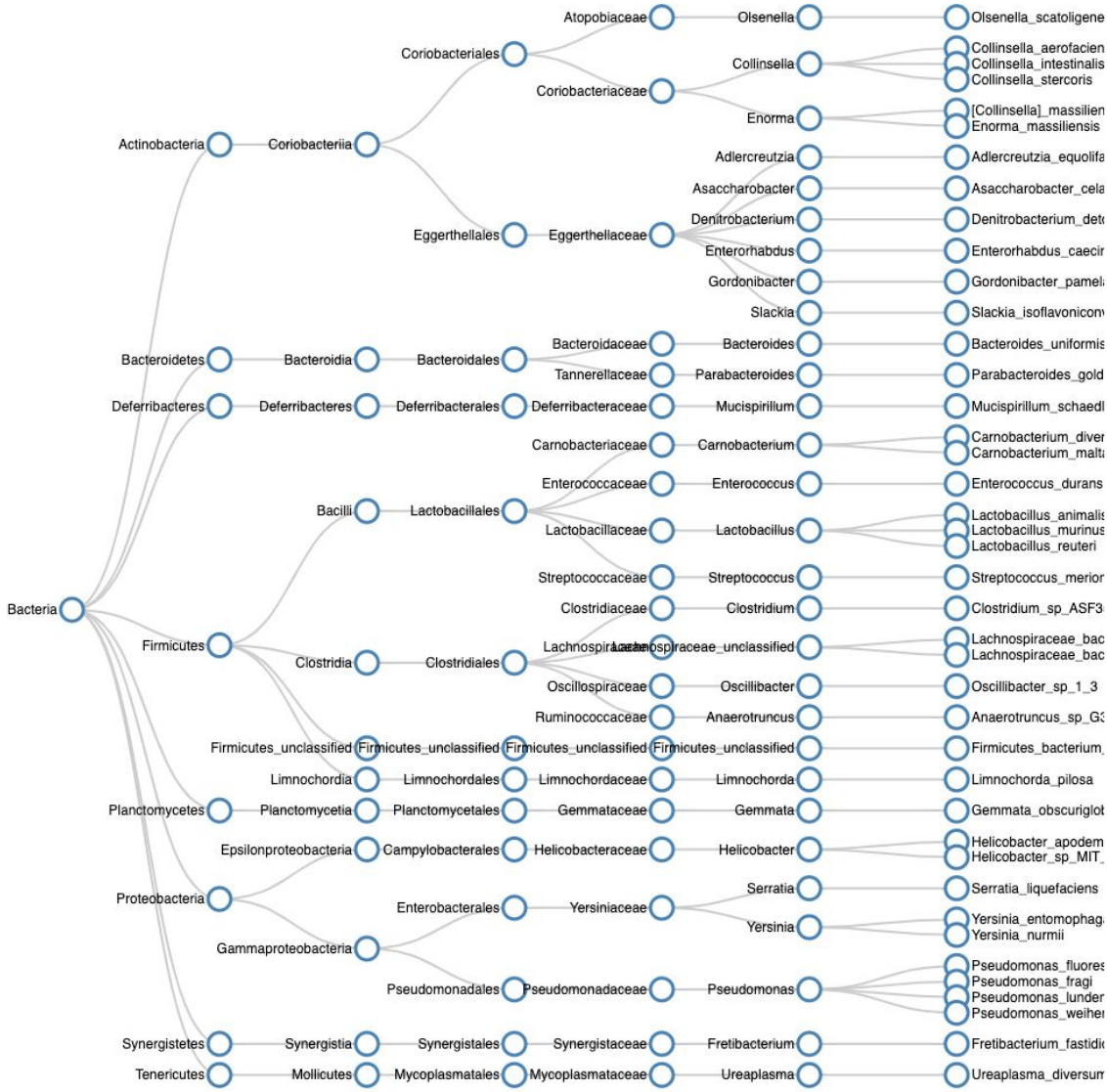
Visualization of Metagenomic Data

Authors: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez



Here, we implemented a version in which all the nodes can expand or collapse its children upon click. Nodes with hidden children are encoded with a blue fill, and nodes with their children expanded are encoded with a white fill color.

A view of the entire tree after expanding all the nodes:



Notes on Tree v2:

- It may be nice to see how many children a node has, could be printed on the circle, or may be distracting….perhaps a tooltip could show the number of children?
- SVG may need to be wider because labels at the species level are being cut off. The width of the current SVG is 1000 px. The depth of each level in the tree is fixed at 130px.
    - Many species' names are redundant i.e. they are formatted as genus_species-name. We may want to consider chopping off the genus name for most species except for **[Collinsella]_massiliensis** of the Enorma genus.
- Map at the top to tell you what level you are in may be useful (**kingdom**, phylum, class, order, family, genus, species)

- We may need an instructional message at the top to indicate that the blue nodes have children that can be expanded and the white nodes are already expanded or are leaf nodes.

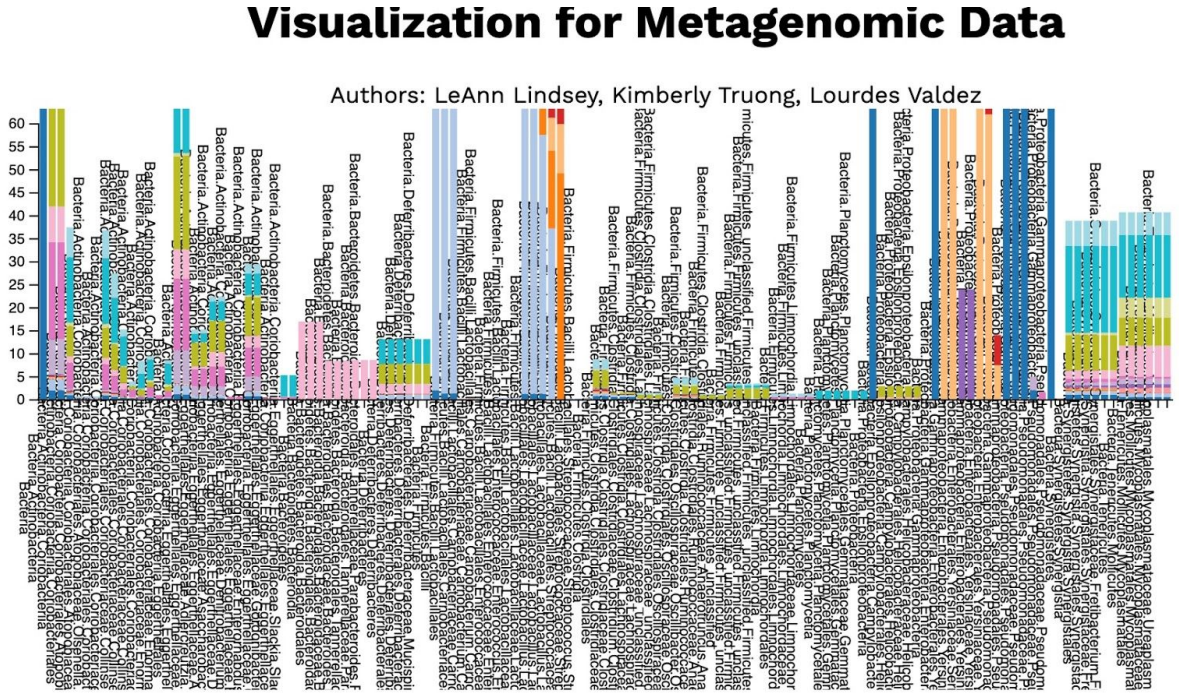
Stacked bar chart challenges:

Trying to work with the same data that was used to create the tree posed many problems. Initially we tried to have the data fit the code by moving variables around so that we didn't have to worry about transposing the data or adding another data file into our script, but despite our efforts, some data values were not being read in correctly. We then had to figure out how to transpose the data and pass in another csv file into our script.

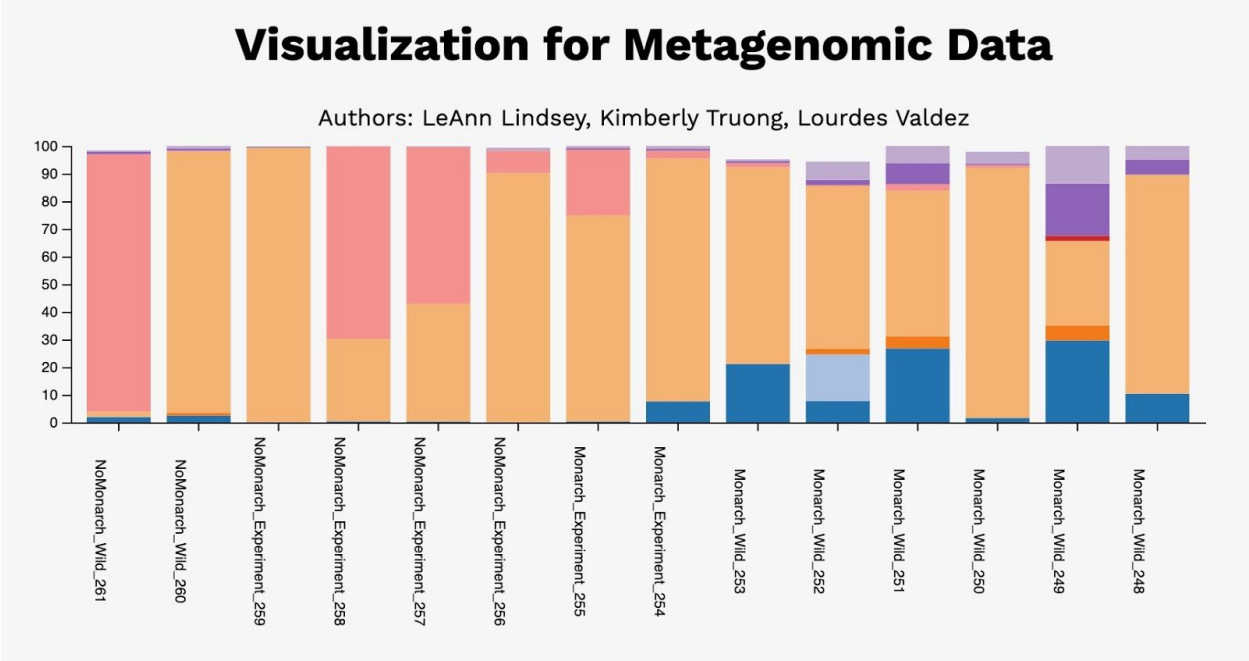
Our current challenges are:

- How to select the level of the data that we want (family, genus, species)?
- How to draw the bar chart once we have the data selected?
- How to get the tooltip to show the taxon name, not just abundance?

Before we determined how to obtain the correct selection, and before we were able to transpose the data, the heatmap looked like this.



Stacked Bar Chart v1, Flat, no interaction, hard coded data selection



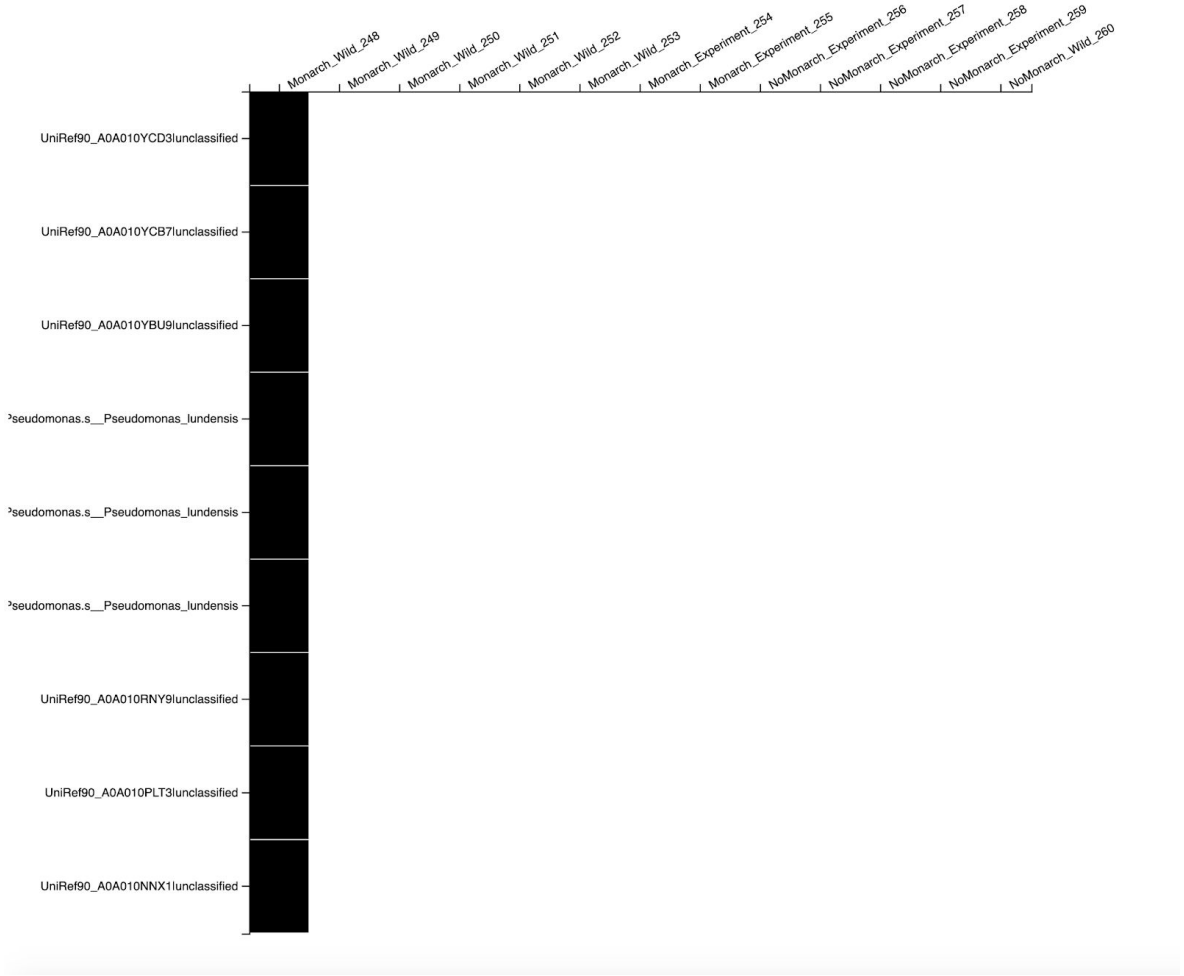
Notes on Stacked Bar Chart v1

- The original data structure will need to be changed to be able to select the appropriate rows in the data to show this visualization. This version is hardcoded with Phylum.
- In this stacked bar chart, you can see some patterns emerging between the conditions, with the blue section at the bottom being more prevalent in the Monarch condition and the purple at the top being more prevalent in Monarch, Pink being more prevalent in Non-monarch and orange being a consistently high percentage in both.
- We should add a color map legend so that it is clear what category each color maps to.
- We will also add a tooltip that allows you to see the percent abundance of each taxon shown within each column.

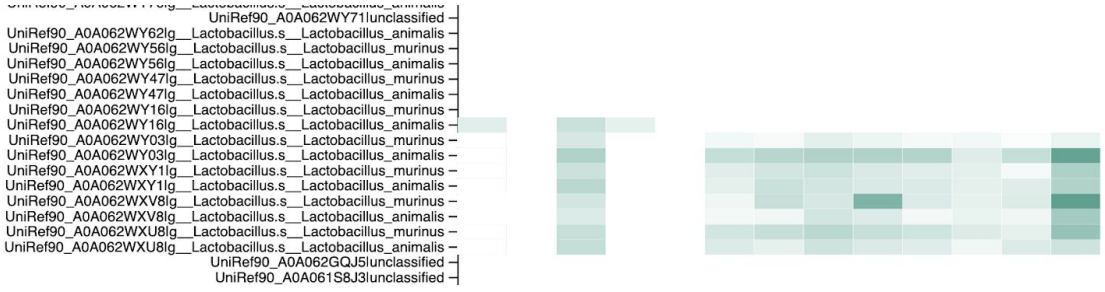
Heatmap Challenges

It turned out to be difficult to create the heatmap using our original data structure (rows were genes and columns were samples), because we realized that for a heatmap there needed to be the same number of rows in the table as the boxes in the heatmap. For this reason, we had to modify the data structure to a flat structure (described in the data section).

This is a photo of the version created with the original data structure...the rectangles are only showing up on the left.



Heatmap v1



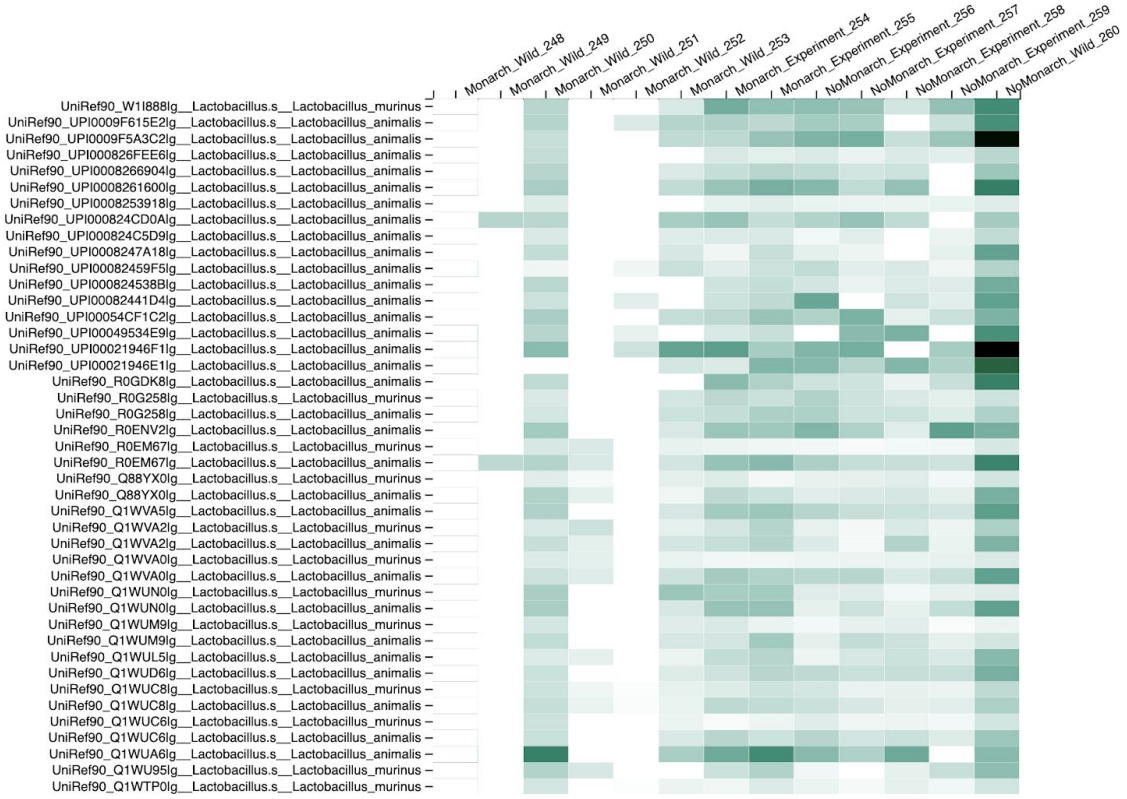
Notes on Heatmap v1:

- This was the colormap used in the sample, and it is actually working quite well. It is nice to have white be the zero value, since there are no negative values in the data matrix
- I used only a subset of the data (top 1000 rows) just to get it working. The majority of the 1000 rows had color in almost no cells in the row. From this, I noticed that it is really only interesting if cells in the row are at least ½ filled (at least 7 cells have the presence

of the bacteria). I then went back and did a subset of the data with that threshold and reduced the data from 280k to 2300 genes.

- I thought a long heatmap that scrolled off the page would not be usable, but length is not really a problem. As long as you can read the labels.

Heatmap v2



Notes on Heatmap v2

- Some samples seem to have very low abundances in all species (248, 249 and 251...need to check to see if something is wrong with the sample). One sample seems to dark in all species (260). Normalization may be necessary.
- 2300 genes are still too many to see, should subset more.
- Not seeing any patterns between No-monarch and Monarch, maybe we need to have a separation between the two conditions
- Suggestion from Youjia: filter the heatmap data to correspond with the most recently clicked node on the tree

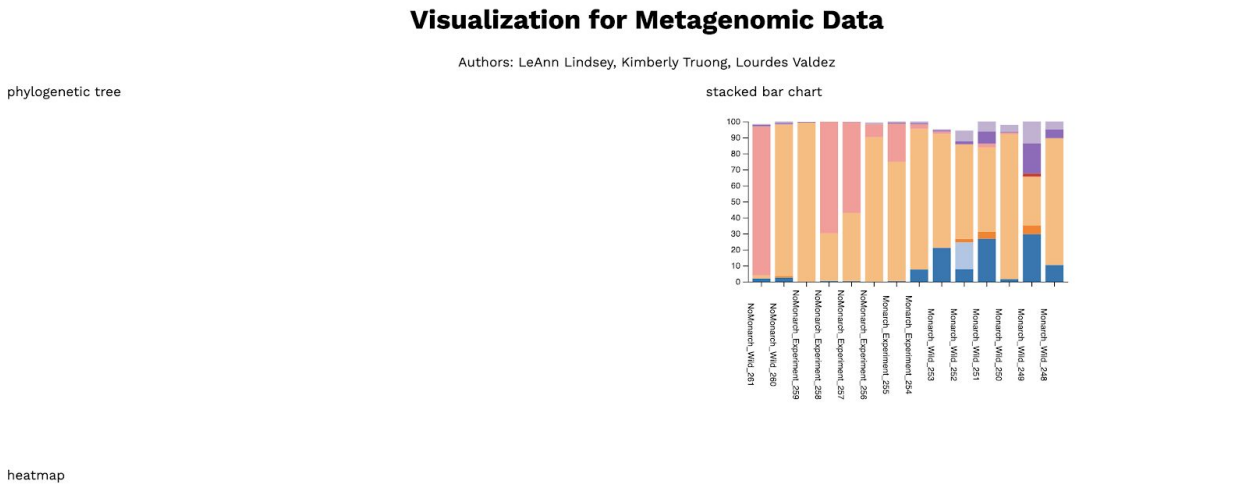
Histogram

We decided that with such few samples, we did not need the histogram. The histogram would be more useful in a situation with significantly more samples. Instead we may use the space below the stacked bar to show different clade levels in multiple stacked bar charts

Component Integration v1

We started with an initial framework .html doc, .css doc, script.js file and individual .js files for each major component.

Initial integration framework was set up by Lourdes in this format:



Notes on initial integration framework:

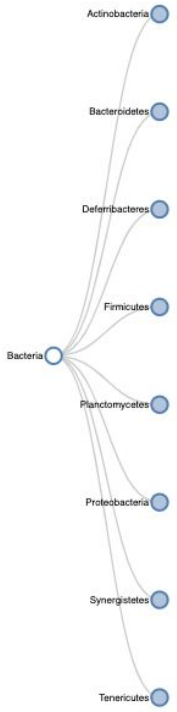
- Stacked bar chart may need to move farther to the right and expand to be taller, based on the height and width requirements of the phylogenetic tree when it is fully expanded.
- Is it possible to have the SVG resize and get wider when the tree is fully expanded? Or can the tree shrink in width to keep the SVG the same size when fully expanded?
- There is already a lot going on the page..perhaps the treemaps are not necessary? Or they could be moved to be under the stacked bar chart as in the sketch below, instead of next to the tree...

Integration v2.

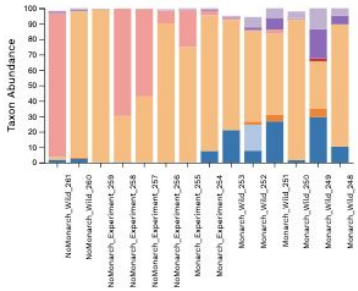
Visualization for Metagenomic Data

Authors: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez

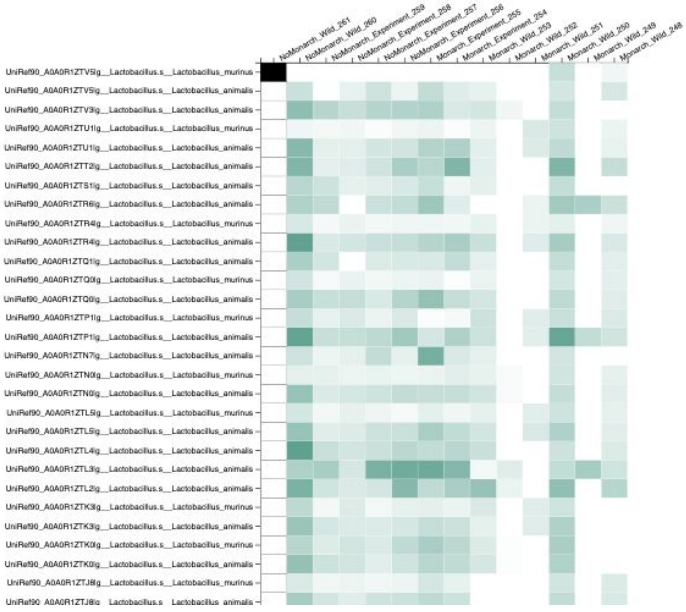
phylogenetic tree



stacked bar chart



heatmap



Implementation

**Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.*

Integration of Tree and Stacked Bar

After the tree and stacked bar were completed as individual components, we realized that there were some challenges with updating each one, and it wasn't clear exactly how we wanted the stacked bar to update based on the tree. We discussed several options to solve.

- Option 1 - Add a menu or radio buttons to the top consisting of the labels of each level of the phylogenetic tree (kingdom, phylum etc…) and the user can click on one of those to make the selection of that level of data for the stacked bar chart.
- Option 2 - Separate menu near the stacked bar chart
- Option 3 - The stacked bar chart updates automatically to a specific level when the node is clicked, for example if the lowest level of the tree is species, then the species stacked bar will be revealed.
- Option 4 - The user can select the level by option 1 or 2 and the interaction between the tree and the stacked bar is done by just highlighting, when a user highlights a node, the corresponding section in the stacked bar is highlighted in some way.
- Option 5 - Most complicated to implement. When a user clicks on a node (or otherwise identifies a node, the stacked bar updates to represent only the children of that node one level down.
- Option 6 - Also complicated to implement..there are really two variables, the clade you want to see in the stacked bar (kingdom, phylum, species etc.) and the section of the tree you want to see. This could be implemented using brushing. The user could select the level of the stacked bar they want to see using Option 1 or 2 above and then the portion of the tree they want to see they could select by brushing and that could update the stacked bar. This version, while useful, may be too complex to implement in the time available.
- Option 7: The user can select the level by option 1 or 2 and then any node at the selected level can be clicked afterwards. The stacked bar updates to show only the children of that node. This combines option 1 or 2 with option 5. Additionally, clicking anywhere on the screen that is not a node element of the tree will collapse all nodes at the level specified by the user.

After discussing the various options, we decided to attempt to implement Option 3. We decided to remove the histogram and to try to add multiple stacked bar charts on the right because you may want to see back to previous levels that you have looked at and we do have room for at least 3, with the most recent being on top.

Evaluation

**What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?*

Lesson 1 - The low abundance species are not showing up in the original Metaphlan run with default settings. Lowering the threshold slightly resulted in many more low abundance species appearing in the output table. Since it is often the low abundance species that are of most interest, this was useful to find out.

Lesson 2 - There are a few samples (248, 249, 251) that seem to have errors or look significantly different than the other samples. These need to be checked with the researchers to see if they are aberrations and need to be left out of analysis, because of some sequencing error or something like that. It is possible they could be fixed by normalization.

Lesson 3 - It turned out to be more challenging to integrate the Gene Ontology information than we had originally expected, as it is not linked in the default output of humann3, but this is also what will make the visualization very useful to researchers.

<u>Revised Schedule after the first checkpoint</u>

Goals/To Do List:

- Tree (Kimberly)
    - Mouseover children number
    - Highlight feature
    - Instructions/Labels
    - Levels at top (optional)
- Stacked Bar (Lourdes)
    - Color Legend
    - Instructions/Label at top
    - Mouseover
- Heatmap (LeAnn)
    - Instructions/Label
    - Color legend
    - Change color scheme
    - Brush to update a smaller selection
    - Click to update sidebar
    - Mouseover gives gene ontology
- Violin Chart & Statistics (LeAnn)
- Integration
    - Update heatmap based on tree (LeAnn)
    - Update bar chart based on tree (Kimberly)
    - Set up new section to the right of heatmap for violin chart & statistics (Lourdes)
- Data (LeAnn)
    - Filter data based on clade
    - SVD filter
    - GO info for selected genes

- Show to Zac in the Lab for Feedback (LeAnn)
- Live Website (LeAnn)
- ScreenCast (Lourdes)
    - Script (Lourdes/Kimberly)

Schedule:

Nov 21	Component Integration Complete
Nov 25	Screencast Recording

Initial Project Proposal & Sketches

Data Visualization Project Proposal

Basic Info

Project Title: Visualization of Metagenomic Data

Project Team:

LeAnn Lindsey	u1323098	lindsey@cs.utah.edu
Kimberly Truong	u1146244	kttruong@math.utah.edu
Lourdes Valdez	u0906219	lourdes.valdez@utah.edu

GitHub Repository:

https://github.com/leannmlindsey/dataviscourse-pr-Visualization-of-Metagenomic-Data

Background and Motivation

Advances in genomic sequencing technology have provided scientists with vast quantities of data to investigate scientific questions. It is now possible to obtain DNA and RNA sequence data not only for a host, but also to obtain metagenomic sequencing for all of the microbes within a specific host site, such as a mammalian gut. This metagenomic sequencing provides not only the genome sequences of the microorganisms present in the system, but also provides insight into the abundances of each species, and a phylogenetic tree of species present. It is known that the presence and abundance of specific species of microorganisms are linked to metabolic disease, autoimmune responses, pathogen detection and toxin metabolism. This metagenomic sequencing data is extremely rich but complex, and difficult to mine for information. Biologists often need to sift through this data searching for a specific gene or pathway which may be upregulated or downregulated under specific conditions and of interest in their research.

Our visualization project is to provide a useful tool to help scientists explore a metagenomic data set. Denise Dearing's laboratory studies microbial detoxification, and we have partnered with her lab to visualize a specific data set acquired last summer by Rodolfo Martinez-Mota, which explores microbial cardenolide detoxification in wild black-eared mice. Monarch butterflies have evolved a resistance to plant toxins, specifically, they feed on milkweed which has high levels of cardenolides. Monarch butterflies migrate each year to overwintering sites in the southern United States and Mexico and during this migration season, predators such as the wild black-eared mouse feed on the butterflies. The goals of the study are to determine the role of the gut microbiome of black-eared mice on detoxification of the cardenolides ingested with a monarch based diet. The Dearing lab has investigated the role of the microbiome using 16S rRNA gene marker sequencing, which provides some information about the species abundances and genes of interest, but they have not yet fully processed and analyzed the metagenomic data set acquired in the study.

While many different tools for visualizing genomic data do exist, very few are interactive enough to be useful during the research process and are more often used to create high quality images for publication. A brief list of the challenges with the current visualizations are listed below.

Challenges with Current Metagenomic Visualization Tools:

- Only a few genomic visualization tools have been extended to visualize metagenomic data, which has a higher level of complexity than genomic data.
- Do not allow a scientist to easily visualize two different experimental conditions at the same time
- Difficult to easily obtain functional pathway information from a gene
- Difficult to install and get working on various software systems, require time to learn to use and interpret output
- Many options available but most only provide one type of visualization

Project Objectives

5. To make an interactive tool which can be used by a researcher to explore metagenomic data from the level of family to species.
6. To connect a specific species to the genes and functional pathways present in that species.
7. To make clear visualizations of sample variation and the differences between two experimental conditions.
8. To explore and visualize interesting aspects of this specific Monarch dataset

Data

Metagenomic Shotgun Sequencing Data collected from 14 samples of wild black-eared mice under the following experimental conditions:

- Monarch/Wild 6 samples
- Non-Monarch/Wild 2 samples
- Monarch/Experiment 2 samples
- Non-Monarch/Experiment 4 samples

Samples labeled “Monarch/Wild” were collected in the wild during Monarch season and are assumed to have eaten Monarch butterflies as a part of their normal foraging.

Samples labeled “Non-Monarch/Wild” were collected during the Non-Monarch season and are assumed to have not eaten Monarch butterflies during that time period.

Samples labeled “Monarch/Experiment” were taken from mice that were captured during Monarch season and fed a diet of 5 Monarch butterflies per day for a period of 48 hours.

Samples labeled “Non-Monarch/Experiment” were captured during Non-Monarch season and taken into captivity for the same period of 48 hrs but were not fed Monarch butterflies.

Data Types:

- Metagenomic Sequence data obtained from processing with Humann3 software from the Huttenhower Lab, Harvard University, to obtain gene family abundances, and functional pathway abundances (.tsv files)
- Taxonomy data obtained from MetaPhlan2 (.csv file)

Data Processing

- Taxonomy data must be processed into the proper format that has parent and child nodes for use in d3. This data is currently in a flat csv file.
- Metagenomic data is too large to visualize, so we will perform feature reduction using Singular Value Decomposition, a numerical linear algebra technique which identifies the most important components in the data. We must reduce the data from 280k rows to 10k rows using features of interest.

Visualization Design

Step 1 Brainstorm: We brainstormed on zoom and also in our github wiki where we placed links to different observable visualization types that we wanted to consider.

Raw Brainstorm: Pictured in our sheet 1.

Filter: We realized we had redundant visualizations and decided on using the tidy tree, stacked bar chart, histogram, heatmap, and the violin plot.

Categorize: Heat map and violin plot display gene abundances and information about gene ontology, and the trees, stacked bar chart, and histogram display species abundances.

Brainstorm Metagenomic Data Visualization

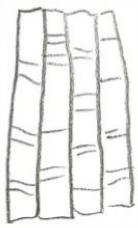
TITLE: Visualization of Metagenomic Data

AUTHORS: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez

DATE: 10/30/2020 **SHEET:** 1

interactive

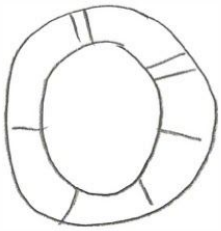
- Stacked bar chart -



Samples →

1. shows abundances in each sample
2. samples grouped by condition
3. click to drill down into Family → Phylum → Class → Species

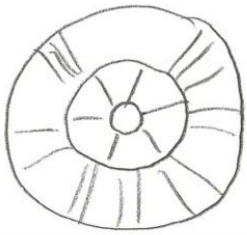
- donut chart -



clickable to drill down but - difficult to see samples compared

in general - difficult to measure/compare circles

- Zoomable Sunburst -



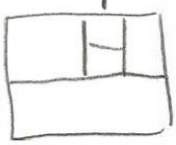
Can show more layers of hierarchy - but similar problems to donut chart

- tidy tree/dendogram -

1. traditional way to show taxonomy

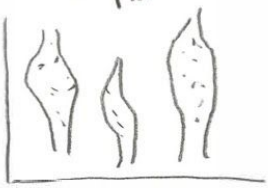
- heatmap -

- treemap -



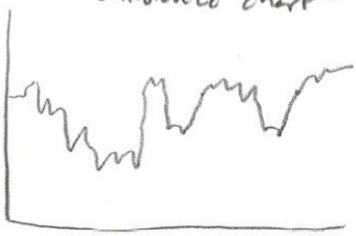
could show abundances

- Violin plot -



Violin plot shows differentially expressed genes — with advantage of showing sample distribution and sample variation

- difference chart -



genes

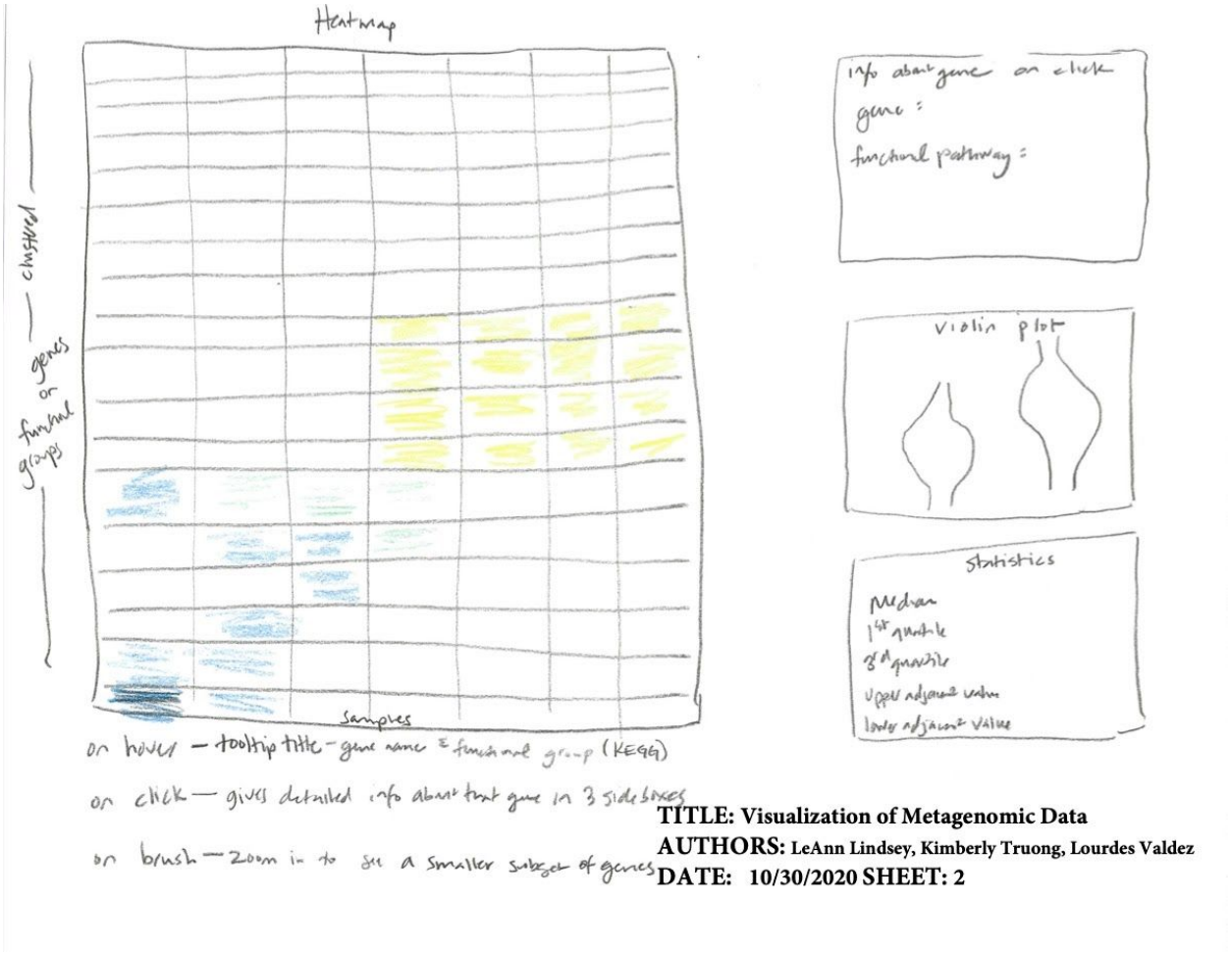
could show differentially expressed genes

- Zoomable Circle Packing -

Could show hierarchy & abundances but not familiar to readers difficult to see quantities

familiar, add interactivity to show gene/functional pathway on hover

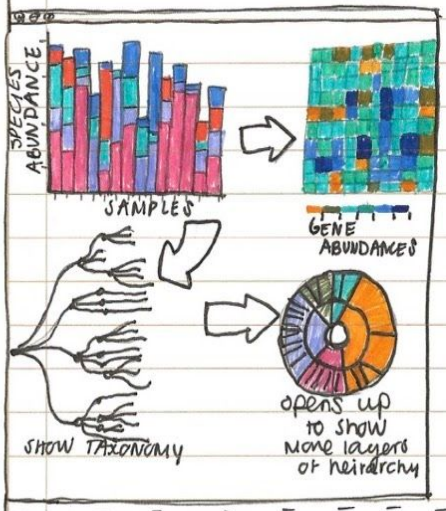
Step 2 Preliminary Designs:



TITLE: Visualization of Metagenomic Data
AUTHORS: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez
DATE: 10/30/2020 SHEET: 2

SUBJECT:

LAYOUT:



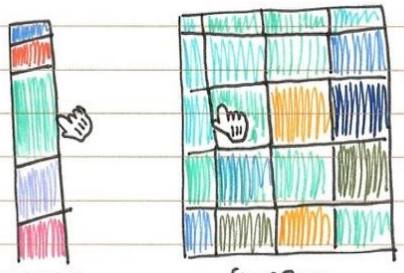
TITLE: visualization of Metagenomic Data

AUTHORS: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez

DATE: 10/30/20 SHEET: 3

TASK: Determine role of gut microbiome of black-eared mice on detox of cardenolides.

OPERATIONS:

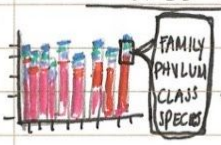


SNIPPET FROM STACKED BAR

SNIPPET FROM HEATMAP

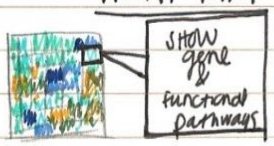
FOCUS/ZOOM:

STACKED BAR CHART



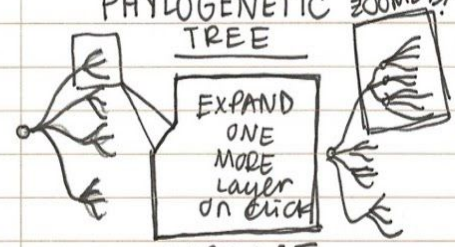
ON CLICK

HEAT MAP

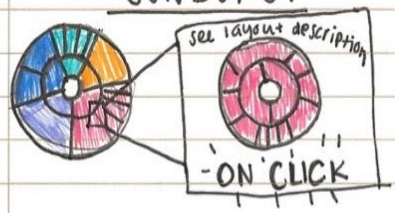


ON HOVER

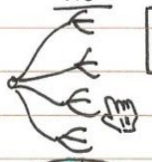
PHYLOGENETIC TREE



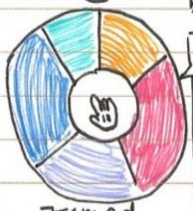
SUNBURST



UNEXPANDED VIEW



User will click on branch of interest for zoomed-in view.



INSIDE CLICK TAKES YOU BACK A LEVEL

zoomed in VIEW FROM SUNBURST

CLICKING ON A REGION LETS YOU EXPLORE IT DEEPER

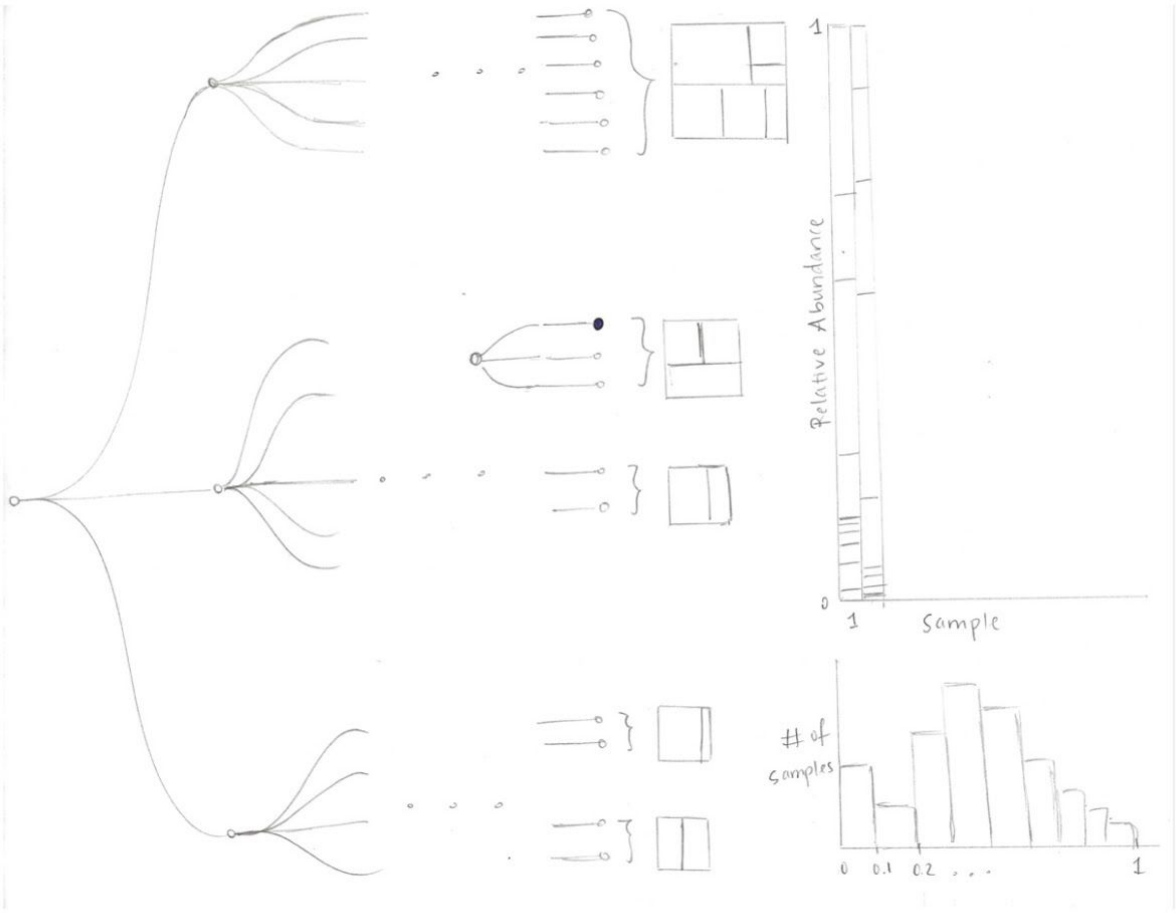
DISCUSSION:

PROS:

- Allows story to be told.
- Good use of focus allows us to give details of data w/out cluttering the webpage.

CONS

- Too much interactivity or too much data can break the page.
- Might need to simplify some visualizations.



TITLE: Visualization of Metagenomic Data
AUTHORS: LeAnn Lindsey, Kimberly Truong, Lourdes Valdez
DATE: 10/30/2020 SHEET: 4

Step 3 Realization Design:

Our final design consists of design sheet 4 -(the phylogenetic tree with the bar charts) being the home page, and design sheet 2 (the heatmap and violin plot) being a secondary view, one level down.

We decided that although the donut chart and the zoomable sunburst are visually pleasing, in general, circles do not communicate values as well as a horizontal encoding like the stacked bar chart. This is why we moved to the bar chart and the histogram. The difference chart was given up in favor of the violin chart in terms of displaying differential expression of an individual gene.

We also had a conversation with Zac Stephens from June Round's lab, who works extensively with metagenomic data, and he said that for him the most valuable information to visualize is the connection between individual gene information and gene functional pathways (gene ontology). This is why we decided to add the sidebar with individual gene information via the heatmap, and functional gene pathways via the hover of the heatmap.

Components & Data Encodings

Tidy tree

- Our dataset type is a tree consisting of nodes.
- Nodes represent taxonomic units.
- Links represent descendants of parent nodes.

Stacked Bar Chart

- Each sample is marked by horizontal position.
- Abundances of Species/Genus/Family/Phylum encoded in length of the bars.
- Species encoded by color

Histogram

- Abundance of selected species is given by position along the x-axis.
- Number of samples is encoded in the height of the bars.
- Species encoded by color

HeatMap

- Each gene is encoded by a row of squares
- Each column is a sample
- Gene abundances encoded by a diverging color map

Gene Ontology

- Gene functional pathways encoded in text

ViolinPlot

- Each sample is marked/represented by area (the violin).
- Gene abundances encoded by Y location
- Experimental condition encoded by color and X location

Statistics

- Statistics from the violin plot such as mean, max, min, 1st & 3rd quartile & standard deviation will be encoded with text and numbers.

Must Have Features

- Clickable Tidy Tree - shows the various levels of the phylogenetic tree of all species present in the host samples
- Clickable Stacked Bar Chart - shows the abundances of species at different levels (Phylum, Class, Order, Family, Genus, Species) with one bar per sample collected
- Histogram - Shows the distribution of abundances pertaining to a selected species across all samples
- Interactivity
    - Stacked Bar Chart is clickable and updates to show more specific regions of the phylogenetic table and updates the histogram of per sample abundances
    - Taxonomy chart is clickable and updates the phylogenetic table and histogram

Optional Features

- TreeMap - near Phylogenetic tree leaves, to show abundances of species within a genus
- HeatMap - Shows all genes present in a specific selection of the tidy tree and Stacked Bar Chart
    - On hover, shows small detail including Gene Name and abundance
    - On click, updates side view which includes Gene Ontology (functional pathway) information, violin plot to show abundances per sample, and statistics under the two conditions
- Violin Plot - updates on click of HeatMap to show abundances per sample of a specific gene
- Statistics comparing abundances of a specific gene in two conditions

Project Schedule

Assignments:

- Kimberly - Taxonomy Chart and Tree Map
- Lourdes - Stacked Bar Chart and Histogram
- LeAnn - Data Wrangling, Component Integration, Heatmap and Violin Plot/Statistics

Nov 1 Feature Reduction & Data Wrangling of Tree data complete

Nov 6	Check in on components & Integration design complete
Nov 10	Components Completed, Basic Integration Structure complete
Nov 15	Project Milestone due - Working Prototype
Nov 25	Component Integration Complete
Dec 1	Video Recording

Additional Material

Component Integration Design