Samuel L. Peoples
BBIO 393: Computational Biology
Project 1

Beta Diversity of Human Gut Microbiome Between Lean and Obese Twins

1. Introduction

This project was completed based on Jeff Gordon's *A Core Gut Microbiome in Obese and Lean Twins* which was used to determine if there exists a noticeable dissimilarity when comparing the Bray-Curtis-Faith Beta Diversity between Lean-vs-Lean, Obese-vs-Obese, and Lean-vs-Obese Twins.

This metric is defined as "a non-phylogenetic β diversity metric" where $BC_{jk} = \frac{\Sigma_i |X_{ij} - X_{ik}|}{\Sigma_i (X_{ij} - X_{ik})}$, and i represents OTUs, and j,k are the samples being compared. This is completed using the *beta-diversity.py* script from within a *Qiime2* virtual machine. The annotated script is as-follows.

2. Import the appropriate packages.

The *Biom* package is used to load the table, which may be used further in Project 2, but a majority of the preprocessing is completed with separate scripts, so this may be removed if it is ultimately not used. *Matplotlib, Scipi, Pylab, Numpy,* and *Random* are used to facilitate any math and visualization that needs to be done, while *OS* and *CSV* are used to save and load different files throughout the process.

The main function is displayed first, so to provide a blueprint for the progression of the script. The *filepath* is defined so to keep track of the absolute path to the project folder location, which contains all the files necessary for this project. The original OTU table is passed into a rarefaction function, and the associated tables are created, ultimately to be visualized as distributions and boxplots for visual comparison.

```python
from biom import load_table
import matplotlib.pyplot as plt
import scipy.stats as stats
import matplotlib as mpl
import pandas as pd
import pylab as pl
import numpy as np
import random
import csv
import os

def main():
    """
    Define the path to the script, which should be in
    the same directory as study_77_011618-113533
    """
    filepath = '/home/qiime/Desktop/Project1'

    # Rarefaction of biom table, assigns path as variable
    rarefaction = single_rarefaction(filepath)

    # Parses the tables and labels
    table = parse_table(filepath,rarefaction)
    betas = betaDiversity(filepath)
    lean, obese = label_tables(filepath, betas)

    # Lean-vs-Lean, Obese-vs-Obese, and Lean-vs-Obese tables
    LL_stats, OO_stats, LO_stats = break_table(filepath,betas, lean, obese)

    # Create the boxplots
    boxplots(filepath, LL_stats,OO_stats,LO_stats)

    # Plot the Distributions
    distribution(filepath,LL_stats,OO_stats,LO_stats)
```

3. Define the helper functions.

The *single_rarefaction* function is used to access the original *.biom* file, and calls the *single_rarefaction.py* script with a rarefaction sample-size of 1000. This makes sure that each column sums to the same value, and will ensure that our results are more accurate. The filepath of the new *biom* file is returned.

```
def single_rarefaction(filepath):
    """
    Runs single_rarefaction.py and returns output file location
    Sample size set to 1000
    """
    input = '/study_77_011618-113533/processed_data/219_otu_table.biom'
    output = '/study_77_011618-113533/processed_data/219_otu_table_even1000.biom'
    sample_size = '1000'
    os.system('single_rarefaction.py'+
              ' -i '+filepath+input+
              ' -o '+filepath+output+
              ' -d '+sample_size)
    return output
```

This function is used to ensure that the counts of the columns in the new table sum to 1000, and saves the new *.biom* file to a variable, which is left unused during this project, but may be accessed during the second project.

```
def parse_table(filepath,rarefaction):
    """
    Passes the rarified table to be parsed
    Returns parsed table
    """
    # Load the table
    biom = load_table(filepath+rarefaction)

    # Define the observations and samples
    otus = biom.ids('observation')
    samples = biom.ids('sample')
    m = biom.matrix_data
    data = [pd.SparseSeries(m[i].toarray().ravel()) for i in np.arange(m.shape[0])]
    table = pd.SparseDataFrame(data, index=otus, columns = samples)
    counts(table)
    return table

def counts(table):
    """
    Counts the sum of the columns to verify that the rarefaction
    had been completed properly.
    """
    for sum in table.sum(axis=0):
        if sum != 1000:
            print("single_rarefaction.py failure.")
    print("Rarefaction check completed.")
```

The *betaDiversity* function is used to calculate the dissimilarity matrix, which is saved to the *stats* directory. The Bray-Curtis-Faith metric is defined here, where a Uni-Frac metric will be implemented in Project 2 to improve accuracy with the introduction of a phylogenetic tree. The dissimilarity matrix is loaded from the output csv file, and returned as a DataFrame.

```
def betaDiversity(filepath):
    """
    Runs beta_diversity.py with a bray_curtis_faith metric,
    and returns the beta diversity matrix as a DataFrame
```

```python
    """
    input = '/study_77_011618-113533/processed_data/219_otu_table_even1000.biom'
    output_dir = '/stats'
    output = '/bray_curtis_faith_219_otu_table_even1000.txt'
    metric = 'bray_curtis_faith'
    os.system('beta_diversity.py'
              +' -i '+filepath+input
              +' -o '+filepath+output_dir
              +' -m '+metric)
    return pd.DataFrame.from_csv(filepath+output_dir+output, sep ='\t')
```

The *label_tables* function is used to associate the sample IDs with their obesity category from the mapping file; a simpler mapping was used after removing excess feature columns. Two lists are returned containing sample IDs that are verified to be in the dissimilarity matrix for each of the respective categories.

```python
def label_tables(filepath, betas):
    """
    Returns label lists of sample IDs which
    correspond to their values in the mapping file
    """
    ez_path = '/study_77_011618-113533/mapping_files/easy_mapping.csv'
    lean=[]
    obese=[]
    lines = []
    with open(filepath+ez_path,'r') as mapping:
        for line in mapping:
            lines.append(line.split(","))
        mapping.close()
    for line in lines:
        if str(line[2]).__contains__("Lean"):
            if line[0] in betas.keys():
                lean.append(line[0])
        elif str(line[2]).__contains__("Obese"):
            if line[0] in betas.keys():
                obese.append(line[0])
    return lean,obese
```

The *break_table* function is used to separate the comparisons into three categories, Lean-vs-Lean, Obese-vs-Obese, and Lean-vs-Obese. The stats are populated into separate lists, verifying that the distance is not equal to zero, which prevents self-comparisons from being considered in the distributions. Summary statistics printed to the console and saved to a text file, while all distances are also saved to their respective *CSV* files. Separating the values in this manner allows us to visualize the box-plots and distributions, making comparisons about the dissimilarities when comparing the different microbial communities.

```python
def break_table(filepath, betas, lean, obese):
    """
    Breaks the beta diversity matrix into Lean-vs-Lean,
    Obese-vs-Obese, and Lean-vs-Obese. The function will
    not allow comparison of self (no distances of zero),
    and will return lists of distances for each value.
    Each list is saved to its own csv file.
    """
    LL_stats = []
    OO_stats = []
    LO_stats = []

    # Separate distances to comparisons of
    # Lean-Lean, Obese-Obese, Lean-Obese
    for i in range(len(lean)):
        for j in range(len(betas[lean[i]].keys())):
            if betas[lean[i]].keys()[j] in lean:
                if betas[lean[i]][j] != 0:
                    LL_stats.append(betas[lean[i]][j])
            elif betas[lean[i]].keys()[j] in obese:
                LO_stats.append(betas[lean[i]][j])
    for i in range(len(obese)):
        for j in range(len(betas[obese[i]].keys())):
            if betas[obese[i]].keys()[j] in obese:
                if betas[obese[i]][j]!= 0:
                    OO_stats.append(betas[obese[i]][j])

    # Print the stats of the three categories
    print("Lean-vs-Lean:\n "+
          "Low: "+str(min(LL_stats))+          # .345
          "\n Mean: "+str(np.mean(LL_stats))+  # .845
          "\n Std: "+str(np.std(LL_stats))+    # .072
          "\n High: "+str(max(LL_stats))+      # .982
          "\n Count: "+str(len(LL_stats)))     # 3422
    print("")
    print("Obese-vs-Obese:\n "+
          "Low: "+str(min(OO_stats))+          # .337
          "\n Mean: "+str(np.mean(OO_stats))+  # .851
          "\n Std: "+str(np.std(OO_stats))+    # .071
          "\n High: "+str(max(OO_stats))+      # .998
          "\n Count: "+str(len(OO_stats)))     # 34782
    print("")
    print("Lean-vs-Obese:\n "+
          "Low: "+str(min(LO_stats))+          # .582
          "\n Mean: "+str(np.mean(LO_stats))+  # .857
          "\n Std: "+str(np.std(LO_stats))+    # .064
          "\n High: "+str(max(LO_stats))+      # .994
          "\n Count: "+str(len(LO_stats)))     # 11033

    # Save the Bray-Curtis-Faith distances to output csvs,
    with open(filepath+"/stats/LL_stats.csv", "wb") as f:
        wr = csv.writer(f,quoting=csv.QUOTE_ALL)
        wr.writerow(LL_stats)
        f.close()
    with open(filepath+"/stats/OO_stats.csv", "wb") as f:
        wr = csv.writer(f,quoting=csv.QUOTE_ALL)
        wr.writerow(OO_stats)
        f.close()
    with open(filepath+"/stats/LO_stats.csv", "wb") as f:
        wr = csv.writer(f,quoting=csv.QUOTE_ALL)
        wr.writerow(LO_stats)
        f.close()

    return LL_stats, OO_stats, LO_stats
```

4. Visualize the output.

The boxplot figure is created from the respective stats lists, allowing the visualization of the different comparison categories. The data is defined as list of samples from the respective categories, which is a simple random sample of 1000 from each. This visualization is discussed in detail in the conclusion.

```
def boxplots(filepath, LL_stats, OO_stats, LO_stats):
    """
    Creates three boxplots based on the LL, OO, and LO
    categories, and saves the boxplots to boxplot.png
    """
    mpl.use('agg')
    data = sample(LL_stats, OO_stats, LO_stats)
    fig = plt.figure(1, figsize=(9,6))
    ax = fig.add_subplot(111)
    bp = ax.boxplot(data)
    ax.set_title('Comparison of Beta Diversity Dissimilarity\n Sample of 1000')
    ax.set_xticklabels(['Lean vs Lean','Obese vs Obese', 'Lean vs Obese'])
    [flier.set(marker='o',color='#e7298a', alpha=.3) for flier in bp['fliers']]
    fig.savefig(filepath+"/figures/boxplot.png", bbox_inches='tight')
```

```python

def sample(LL_stats, OO_stats, LO_stats):
    """
    Returns a random sample of the three categories
    """
    return [random.sample(LL_stats, 1000),
            random.sample(OO_stats, 1000),
            random.sample(LO_stats, 1000)]
```

Individual distributions are plotted to better distinguish the differences in dissimilarity. Each visualization is discussed in detail in the conclusion.

```python
def distribution(filepath,LL_stats,OO_stats,LO_stats):
    """
    Visualize the Distributions of the
    Lean-vs-Lean, Obese-vs-Obese, and Lean-vs-Obese
    categories. Saves figures to output file
    """
    LL = sorted(LL_stats)
    OO = sorted(OO_stats)
    LO = sorted(LO_stats)
    LL_fit = stats.norm.pdf(LL,np.mean(LL),np.std(LL))
    OO_fit = stats.norm.pdf(OO,np.mean(OO),np.std(OO))
    LO_fit = stats.norm.pdf(LO,np.mean(LO),np.std(LO))

    # Clear any figures that may be present.
    pl.clf()

    # Plot the Lean vs Lean
    pl.plot(LL,LL_fit,'-o')
    pl.hist(LL,normed=True)
    pl.title("Bray-Curtis-Faith Dissimilarity of Lean vs Lean Twins")
    pl.savefig(filepath+"/figures/LL_Distribution.png")
    pl.clf()

    # Plot the Obese vs Obese
    pl.plot(OO,OO_fit,'-o')
    pl.hist(OO,normed=True)
    pl.title("Bray-Curtis-Faith Dissimilarity of Obese vs Obese Twins")
    pl.savefig(filepath+"/figures/OO_Distribution.png")
    pl.clf()

    #Plot the Lean vs Obese
    pl.plot(LO,LO_fit,'-o')
    pl.hist(LO,normed=True)
    pl.title("Bray-Curtis-Faith Dissimilarity of Lean vs Obese Twins")
    pl.savefig(filepath+"/figures/LO_Distribution.png")
    pl.clf()
```

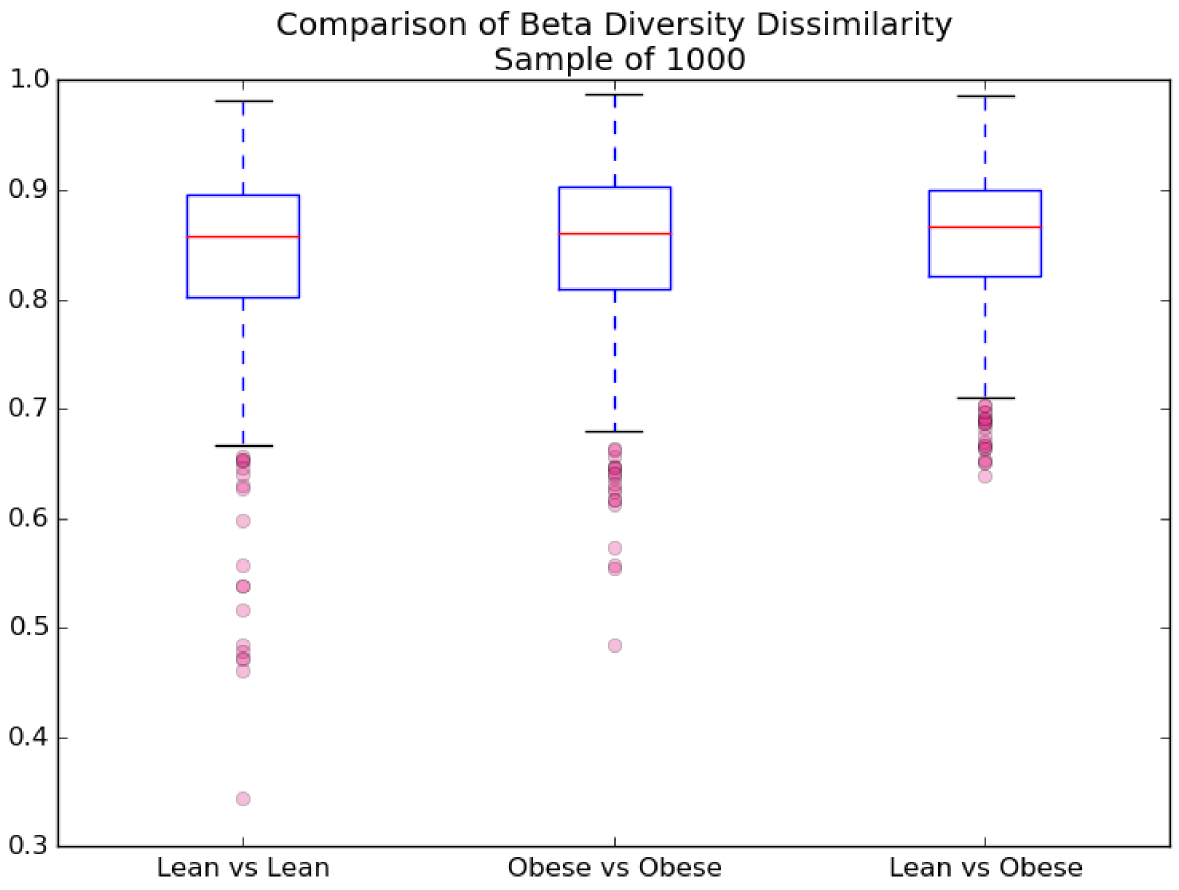
5. Conclusion

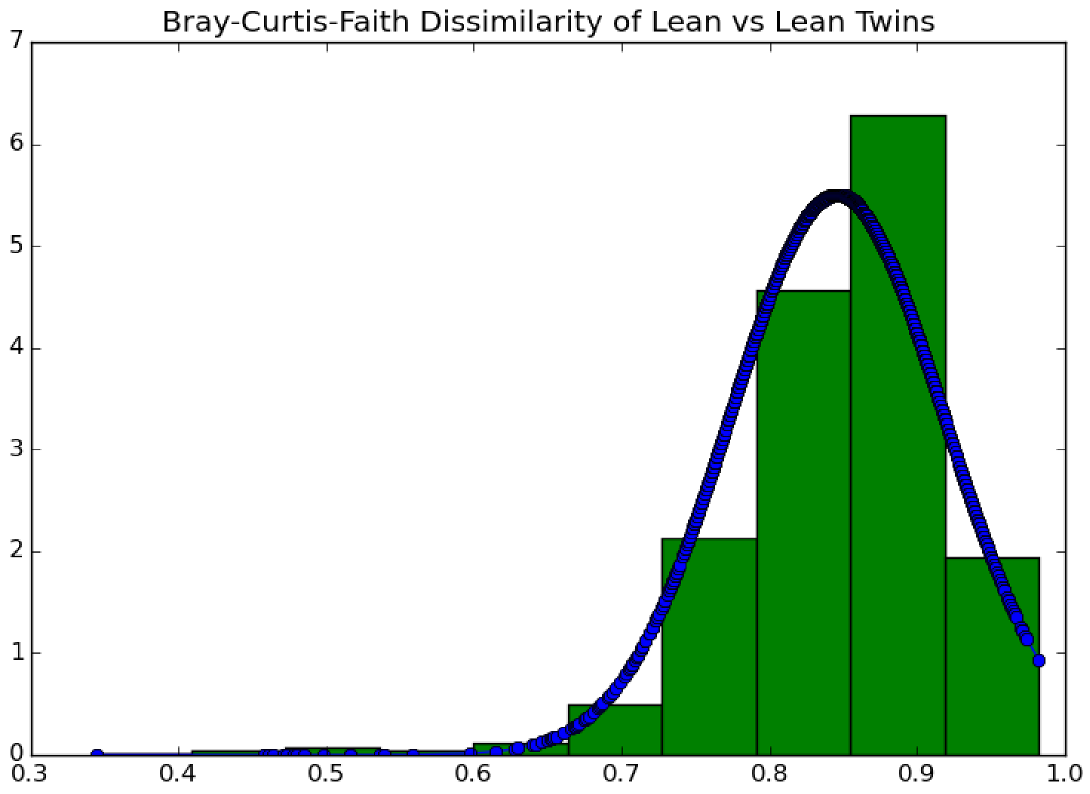
When viewing the boxplots and distributions, we can see that the distribution is reduced when comparing dislike obesity categories, where Lean-vs-Obese has no samples below roughly 0.60, and Lean-vs-Lean, Obese-vs-Obese each have samples with dissimilarities as low as roughly 0.30.

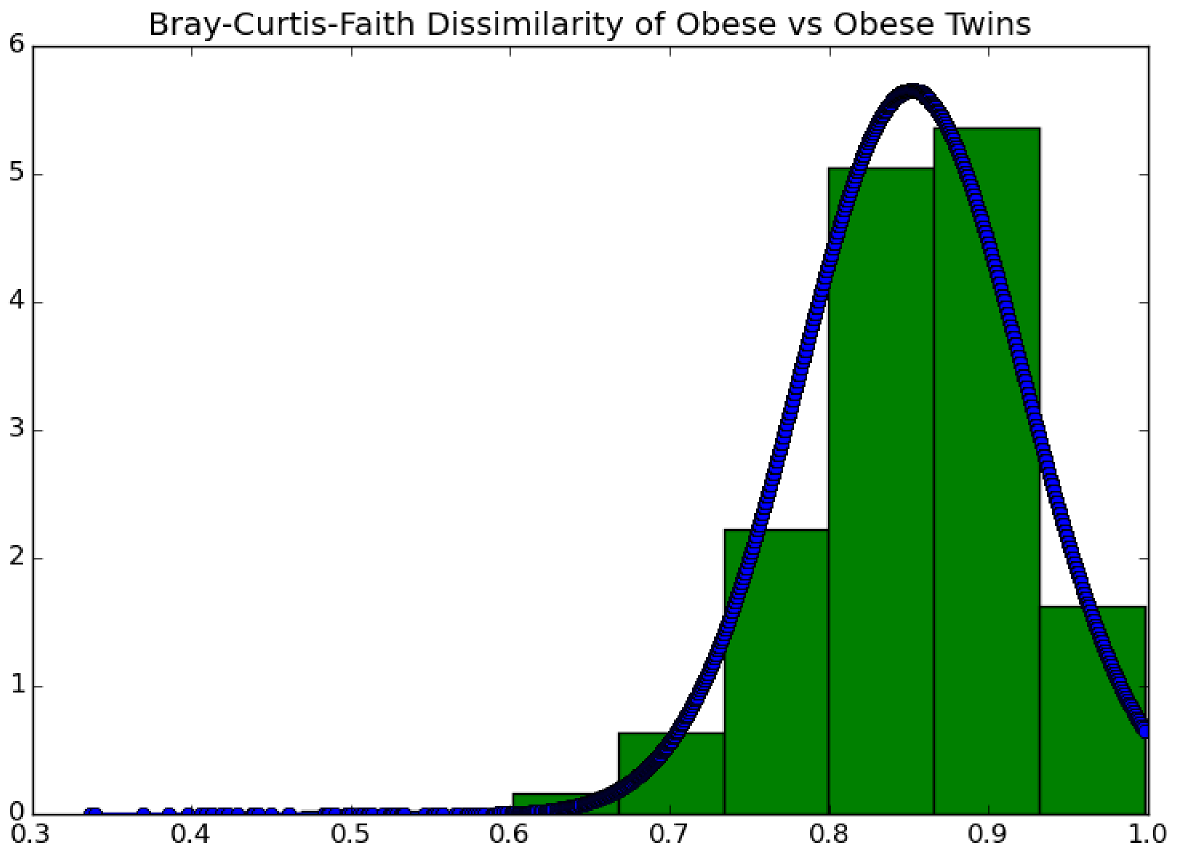
This difference opens up many questions which can be answered in Project 2, which include the statistical significance of this difference, and comparison to the same analysis under different diversity metrics. The increasing trend of higher dissimilarity in the boxplots suggests that there may be a relationship between the diversity of the gut microbiome and obesity, and could be answered with more investigation.

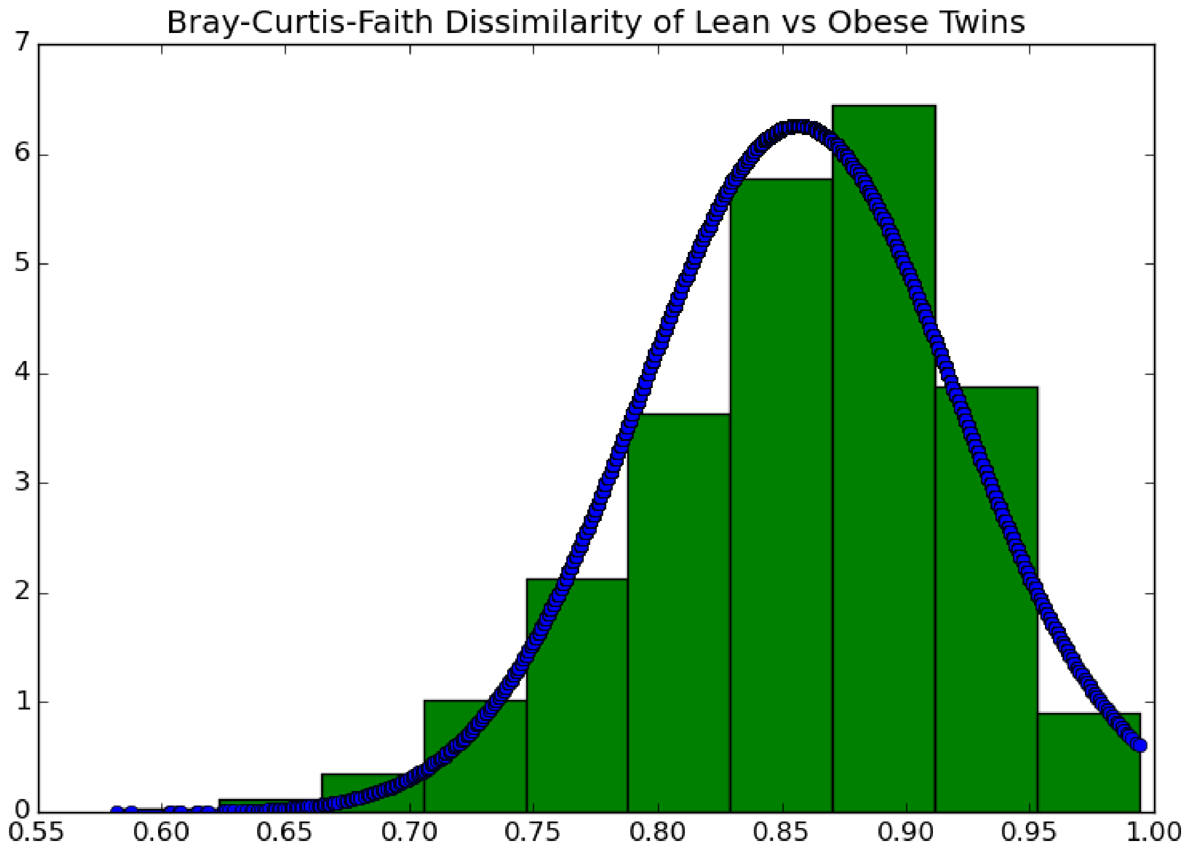
There is also room for improvement, where accuracy of the analysis can be increased with the use of a phylogeny-based beta diversity metric (Uni-Frac), which would affect the distances in the dissimilarity matrix. This evaluation can also be improved by implementing a Monte-Carlo (or similar) simulation with many samples of 1000 comparisons. The script could also be generalized following Project 2, allowing for more parameterized functionality, which would be applicable to more datasets.

Being able to visually recognize the dissimilarity reveals the potential differences that exist within the microbial communities of lean vs obese individuals, providing inspiration for further investigation and insights.






Bray-Curtis-Faith Dissimilarity of Obese vs Obese Twins
0
1
2
3
4
5
6
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0


Bray-Curtis-Faith Dissimilarity of Lean vs Obese Twins
0
1
2
3
4
5
6
7
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00

6. References

Qiime: http://qiime.org/
Qiita: https://qiita.ucsd.edu/
Bray-Curtis-Faith: http://readiab.org/book/0.1.3/3/1#4.1.1
Gut Microbiome Dataset: https://qiita.ucsd.edu/study/description/77
Biom-Format: http://biom-format.org/documentation/biom_format.html
Lozupone/ Knight's UniFrac: http://aem.asm.org/content/71/12/8228.full
Qualitative/ Quantitative: https://www.ncbi.nlm.nih.gov/pubmed/17220268
Weighted UniFrac: https://liorpachter.wordpress.com/2013/09/18/unifrac-revealed/