



# *prepare\_taxa\_charts.py*: A Python program to automate generation of publication ready taxonomic pie chart images from QIIME

Vijay Lakhujani, Chandan Badapanda \*

Xcelris Labs Limited, Ahmedabad, Gujarat, India



## ARTICLE INFO

### Article history:

Received 20 December 2016

Accepted 26 March 2017

Available online 29 March 2017

### Keywords:

16S rRNA amplicon/ITS analysis

Metagenome

QIIME

Python

Taxonomy

Pie chart

## ABSTRACT

QIIME (Quantitative Insights Into Microbial Ecology) is one of the most popular open-source bioinformatics suite for performing metagenome, 16S rRNA amplicon and Internal Transcribed Spacer (ITS) data analysis. Although, it is very comprehensive and powerful tool, it lacks a method to provide publication ready taxonomic pie charts. The script *plot\_taxa\_summary.py* bundled with QIIME generate a html file and a folder containing taxonomic pie chart and legend as separate images. The images have randomly generated alphanumeric names. Therefore, it is difficult to associate the pie chart with the legend and the corresponding sample identifier. Even if the option to have the legend within the html file is selected while executing *plot\_taxa\_summary.py*, it is very tedious to crop a complete image (having both the pie chart and the legend) due to unequal image sizes. It requires a lot of time to manually prepare the pie charts for multiple samples for publication purpose. Moreover, there are chances of error while identifying the pie chart and legend pair due to random alphanumeric names of the images. To bypass all these bottlenecks and make this process efficient, we have developed a python based program, *prepare\_taxa\_charts.py*, to automate the renaming, cropping and merging of taxonomic pie chart and corresponding legend image into a single, good quality publication ready image. This program not only augments the functionality of *plot\_taxa\_summary.py* but is also very fast in terms of CPU time and user friendly.

© 2017 Xcelris Labs Limited. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Microbiome analysis is widely carried out using QIIME bioinformatics suite [1]. It comes as a package containing multiple Python based programs to perform different types of analysis from the sequencing data. Additionally, it also offers powerful graphical visualization tools to find patterns of taxonomic diversity. Being very user friendly and well documented, it is most preferred tool for high-throughput sequencing microbial ecology studies. There are seven major steps executed during analysis through QIIME [4] as described below:

- i. In the first step, chimeras (DNA sequences from two or more microbial species) are filtered from the high quality stitched data using usearch61 [3] algorithm (de novo/reference based). High quality stitched data with phred score >30 is taken as input in this step.
- ii. In the next step, similar sequences (i.e. sequences coming from the same species) are clustered together into one representative taxonomic unit called as *Operational Taxonomic Unit* (OTU). The

basis of this sequence clustering is 97% sequence similarity and is implemented through UCLUST [3] algorithm.

- iii. Next step consists of picking a representative sequence for each of picked OTUs. As these OTUs are made up of a group of sequences, they need to be represented through one sequence which helps assigning a taxonomic name to the corresponding OTU and construct phylogenetic tree.
- iv. With the representative sequence in hand, in the next step, taxonomic names are assigned to these sequences. This is done using UCLUST algorithm [3], where queries are the representative sequences and subjects are the curated sequences from Greengene database [2]. Here, the minimum fraction of database hits that must have a specific taxonomic assignment to assign that taxonomy to a query is considered at 51% and similarity is considered at 90%.
- v. The following step is making a OTU table. A script assembles all the OTUs present in each of the samples with their abundances into a table known a BIOM file.
- vi. Next step consists of intra-sample analysis where the objective is to find the species richness as well species evenness in each of the samples.
- vii. In the final step, inter sample analysis is performed where the objective is to compare the samples to one another by generating

\* Corresponding author.

E-mail addresses: [chandan.badapanda@xcelrislabs.com](mailto:chandan.badapanda@xcelrislabs.com), [chandan.bioinfo@gmail.com](mailto:chandan.bioinfo@gmail.com) (C. Badapanda).

distance matrix which could be used to visualize a tree, graph of points or a network.

A key step in the analysis using QIIME is to make taxonomy summary pie charts based on taxonomy assignment. The script *plot\_taxa\_summary.py* with the option *-c* or *-chart\_type* as “pie” (for pie chart) creates a html file having all the pie charts along with an output folder using the taxonomy or category counts from another script *summarize\_taxa.py*. The output folder contains one pie chart image in *png* format and one legend image in *pdf* format (by default) for each sample. To view all the pie charts, the html file needs to be opened in a web browser. Along with the pie chart image (Fig. 1), there is a hyper-link for the legend image (Fig. 2) within the html file.

It is often required to have the pie chart and its legend in the same image for publication/report purpose. This could be achieved by executing the script *plot\_taxa\_summary.py* with the parameter *-s/-include\_html\_legend* which places the legend within the html page. However, with this option, the legend size is unequal as compared with size of the pie chart specially for taxonomic ranks like family and genus (Fig. 3). Again, for publication ready images, user needs to manually crop the image from the html. This task is laborious when it comes to multiple samples (say ~100) because each sample has 2 images (one pie chart and legend image) at each taxonomic level viz. Phylum, Class, Order, Family and Genus. For 100 samples, there are  $100 \times 2 \times 5 = 1000$  images to be merged manually. It takes hours to achieve this task manually even for the most skillful hands.

There is another option to generate the legend images in *png* format by executing the script *plot\_taxa\_summary.py* with the option *-t png* as mentioned below. However, since the images are named with random alphanumeric characters, it is nearly impossible to identify the pie chart and legend pair. Additionally, identifying the images and editing them manually is an error prone process.

```
plot_taxa_summary.py -i phylum.txt -l phylum -c pie -o
phylum_charts/ -t png
-i Input comma-separated list of summarized taxa file paths
-l Comma separated list of taxonomic levels (e.g. Phylum, Class,
Order) [default = None]
-c This is the type of chart to plot
-o Output directory
-t This is the type of image to produce (i.e. pdf, svg, png). [default:
pdf]
-d This is the resolution of the plot. [default: 80]
```

To remove all the bottlenecks as mentioned above, we present *prepare\_taxa\_charts.py*, a Python based efficient program to provide

good quality publication ready taxonomic pie chart images in significantly less time. The program performs the following operations sequentially:

- Parses the *pie\_charts.html* file to map the sample name and the corresponding pie chart images and make a Python dictionary out of it. This is important because the script *plot\_taxa\_summary.py* generates random file names.
- Create a user defined output folder to store the publication ready images. If the folder already exists, it is deleted.
- Copies the pie chart and legend images into the user defined folder.
- Rename the images in the following format:

for pie chart image: <taxonomic\_level><sample\_name><.png>  
for legend image: <taxonomic\_level><sample\_name><\_legend.png>

This is important to identify the taxonomic pie chart and legend pair and associate them with corresponding sample identifier.

- For each pie chart and legend image pair, a blank image is generated to accommodate both the images. Dimensions of the blank image are adjusted based on height and width of the pie chart and legend images. Both the images are then cropped automatically to remove unnecessary blank space and finally they are merged into a single image. The final images have a suffix “\_final”. Rest of the images are deleted automatically. Fig. 5 represents a final image generated by the program *prepare\_taxa\_charts.py* at order level by merging the pie chart (Fig. 1) and legend image (Fig. 2).
- The program also counts the number of samples and calculates the time required to execute step i. to v. and prints it in the command line.

## 2. Materials and methods

### 2.1. Requirements

#### 2.1.1. Python v2.7

Implementation of the program is based on Python v2.7. This version was chosen as it is more stable than v3.0. Table 1 contains the list of modules used in the program.

#### 2.1.2. Generation of legend images in png format

A prerequisite for *prepare\_taxa\_charts.py* is that legend files should be generated in *png* format (default is *pdf*). This could be achieved by running the script *plot\_taxa\_summary.py* with the option *-t/-type\_of\_file* as *png*. Also, for good quality images, the resolution should

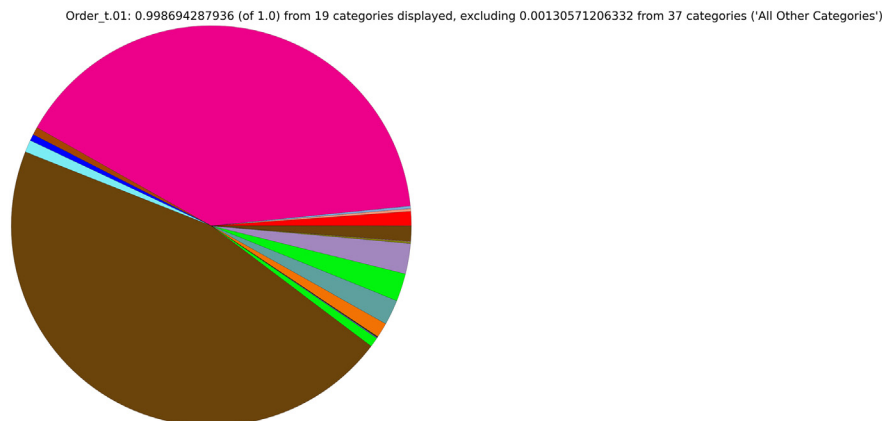


Fig. 1. Taxonomic pie chart generated from *plot\_taxa\_summary.py* summarizing taxonomy assignment at order level.

Download English Version:

<https://daneshyari.com/en/article/5590169>

Download Persian Version:

<https://daneshyari.com/article/5590169>

[Daneshyari.com](https://daneshyari.com)