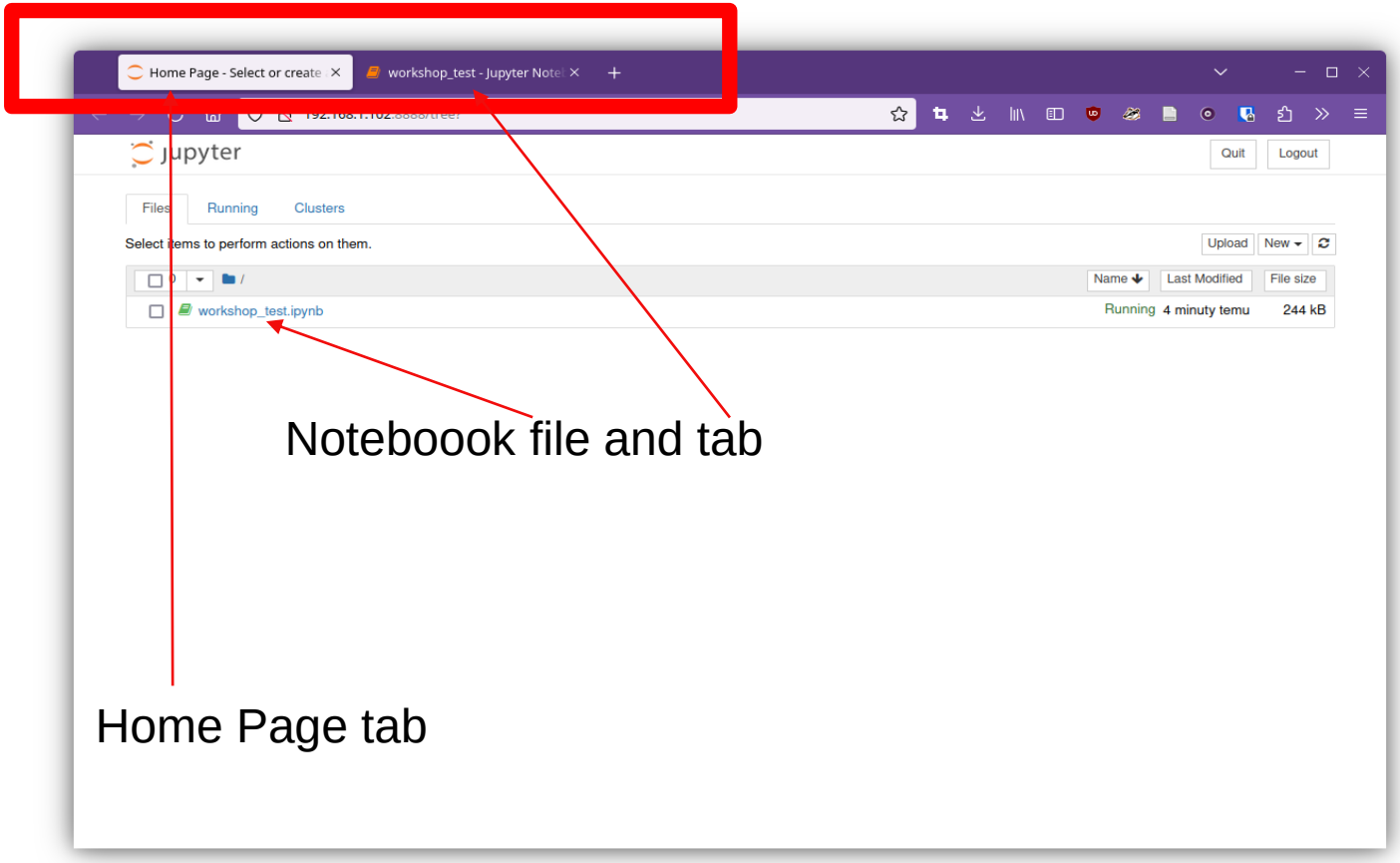
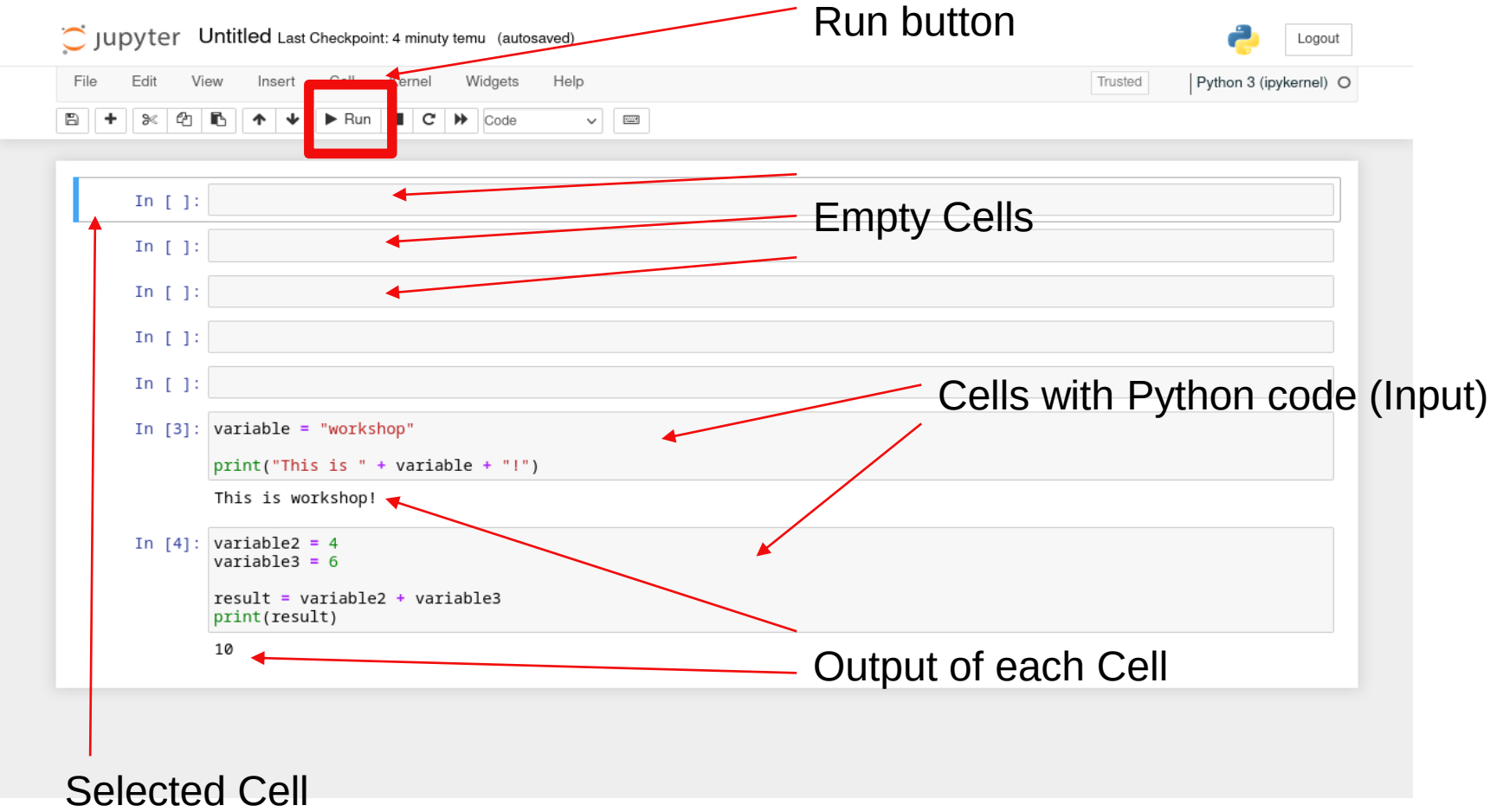
Introduction to Jupyter Notebook

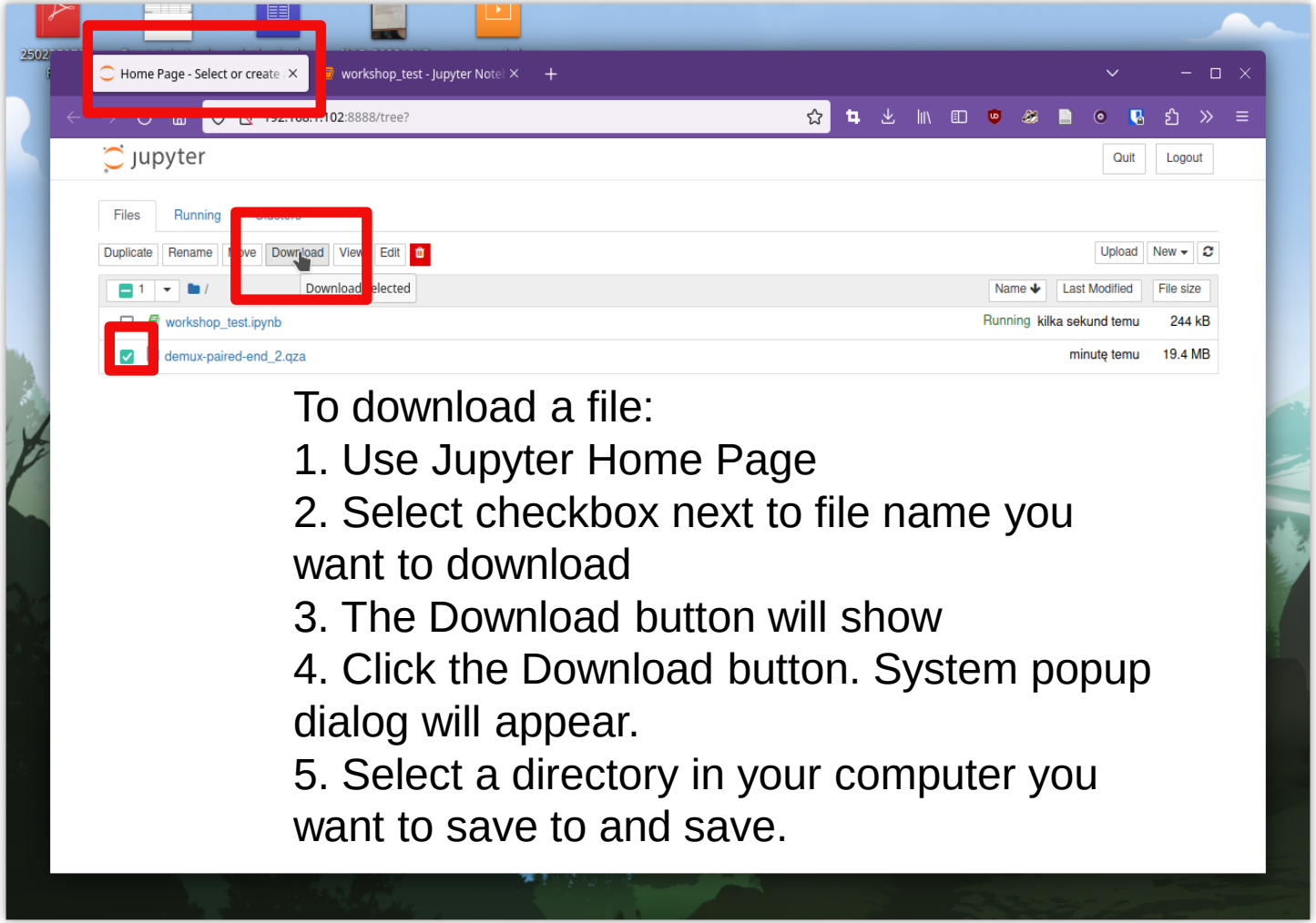
Jupyter Home Page



Jupyter Notebook



How to download file from server to your computer



To download a file:

1. Use Jupyter Home Page
2. Select checkbox next to file name you want to download
3. The Download button will show
4. Click the Download button. System popup dialog will appear.
5. Select a directory in your computer you want to save to and save.

view.qiime2.org website

https://view.qiime2.org

qiime2view

This interface can view .qza and .qzv files directly in your browser without uploading to a server. Click here to learn more.

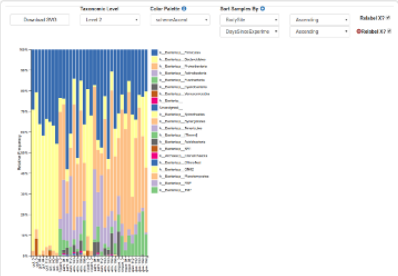
Drag and drop or click here

to view a QIIME 2 Artifact or Visualization (.qza/.qzv) from your computer.

You can also provide a link to a file on Dropbox or a file from the web.

Gallery

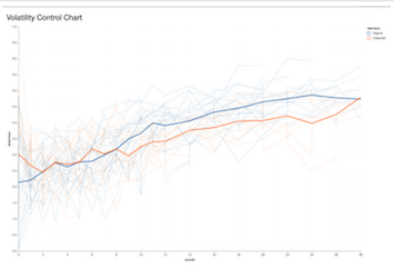
Don't have a QIIME 2 result of your own to view? Try one of these!



Taxonomic Bar Plots

Explore the taxonomy of samples in the Moving Pictures Tutorial. Try selecting different taxonomic levels and metadata-based sample sorting.

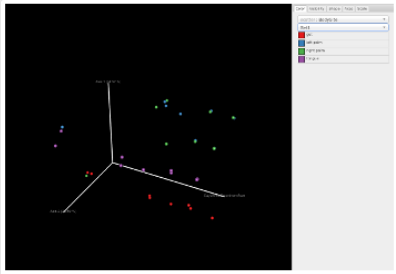
Try it!



Volatility Control Chart

Explore interactive line plots to assess how volatile a dependent variable is over a continuous, independent variable in one or more groups.

Try it!



Emperor Ordination

View the differences between sample composition using unweighted UniFrac in ordination space. Color the samples by different metadata columns.

Try it!

SIREN WORKSHOP

OLSZTYN
2024

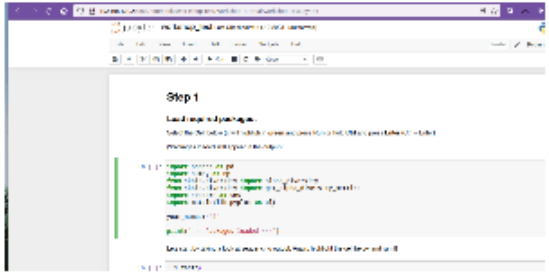
STEP 1

Step 1

Load required packages:

Select the **Cell** below (it will highligh in **blue or green** and press **Run** or hold **Ctrl** and press **Enter** (Ctrl + Enter)

('Packages loaded' will appear in the output):



Click to enlarge

In [1]:
```
import pandas as pd
import numpy as np
from skbio.diversity import alpha_diversity
from skbio.diversity import get_alpha_diversity_metrics
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

your_number='2'

print('>>> Packages loaded <<<')
```

```
>>> Packages loaded <<<
```

STEP 1

```
your_number='2'

print('>>> Packages loaded <<<')
```

```
>>> Packages loaded <<<
```

Lets start by taking a look at sequencing output.

Again, highlight the cell below and run it!

```
In [2]: !ls ../fastq/
```

```
25k-V3-02-22-R24_11_L001_R1_001.fastq.gz
25k-V3-02-22-R24_28_L001_R2_001.fastq.gz
25k-V3-05-22-R23_10_L001_R1_001.fastq.gz
25k-V3-05-22-R23_27_L001_R2_001.fastq.gz
25k-V3-07-21-R23_0_L001_R1_001.fastq.gz
25k-V3-07-21-R23_17_L001_R2_001.fastq.gz
```

Exclamation mark (!) allows us to run Bash (Linux terminal) commands (to interact directly with a server), not Python

Here we are running >ls< command (!ls) which lists the contents of a given directory

STEP 2

Step 2

Command to import .fastq files you've seen above into QIIME2

Import the sequence data to QIIME2

Now lets start the analysis using qiime2. We need to import the sequences data files into a QIIME 2 artifact using the qiime tools import plugin.

Run the cell below.

Using this sign we are able to write command in multiple lines

These are parameters, which gives QIIME2 various details.
Here, parameters concern input-path where .fastq files are loacated

In [2]:
```
!qiime tools import \
  --input-path ../fastq/ \
  --type 'SampleData[PairedEndSequencesWithQuality]' \
  --input-format CasavaOneEightSingleLanePerSampleDirFmt \
  --output-path demux-paired-end_{your_number}.qza | batch

print('>>> finished <<<')
print('>>> Files has been imported. Your project is now saved as demux-paired-end.qza<<<')
```

After running this command, you will obtain .qza

```
warning: commands will be executed using /bin/sh
job 207 at Tue Jan  9 13:17:00 2024
>>> finished <<<
>>> Files has been imported. Your project is now saved as demux-paired-end.qza<<<
```

Now please switch your broser tab to "Home Page - Select or create a notebook" tab. Do you see your just-created file there?

Again we are using Bash to run QIIME2

STEP 3

Step 3

Step 3.1

Generate a visualization file to examine the sequence quality

You will get the file with a .qza extension, which is internal file used by qiime2 and is not accesible to us.

After importing the demultiplex sequence data into an artifact, we will generate a summary with the plugin **qiime demux summarize**. This summary provides us with visual information of the distribution of sequence qualities at each position in the sequence data for the next step of the pipeline. The sequence qualities inform the choices for some of the sequence-processing parameters, such as the truncation parameters of the DADA2 denoising step. This summary also tells us about how many sequences were obtained per sample.

To convert .qza file into .qzv file, please run the cell below.

In [62]:
```
!time qiime demux summarize \
--i-data demux-paired-end_{your_number}.qza \
--o-visualization demux-paired-end_{your_number}.qzv | batch


print('>>> finished <<<')
print('>>> You have successfully converted .qza to .qzv file! <<<')
```

```
warning: commands will be executed using /bin/sh
job 243 at Tue Jan 16 10:25:00 2024

real    0m39,704s
user    0m40,051s
sys     0m2,238s
>>> finished <<<
>>> You have successfully converted .qza to .qzv file! <<<
```

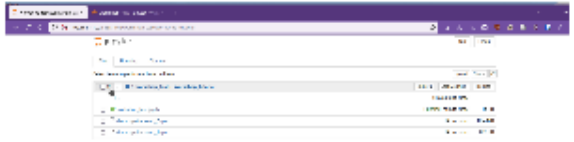
We can put >time< command, which will measure for us time that it took to finish running the command

Step 3.2

Now you have your .qzv file, which can be viewed using qiime2 website.

To do it, please follow the steps below:

1. Switch your browser tab from notebook to the Home Page (Broser tab named: Home Page - select or create a notebook).
2. Select and download the demux-paired-end_{your_number}.qzv file to your computer.
3. Go to https://view.qiime2.org/ (click to open in a new tab)
4. Upload your file by clicing on a gray box and selecting .qzv file from your computer.



Click to enlarge

STEP 3

Demultiplexed sequence counts summary

	forward reads	reverse reads
Minimum	NaN	NaN
Median	NaN	NaN
Mean	NaN	NaN
Maximum	NaN	NaN
Total	NaN	NaN
Minimum	24761.0	NaN
Median	24846.0	NaN
Mean	24826.666667	NaN
Maximum	24873.0	NaN
Total	74480.0	NaN
Minimum	NaN	24761.0
Median	NaN	24846.0
Mean	NaN	24826.666667
Maximum	NaN	24873.0
Total	NaN	74480.0

Forward Reads Frequency Histogram

Reverse Reads Frequency Histogram

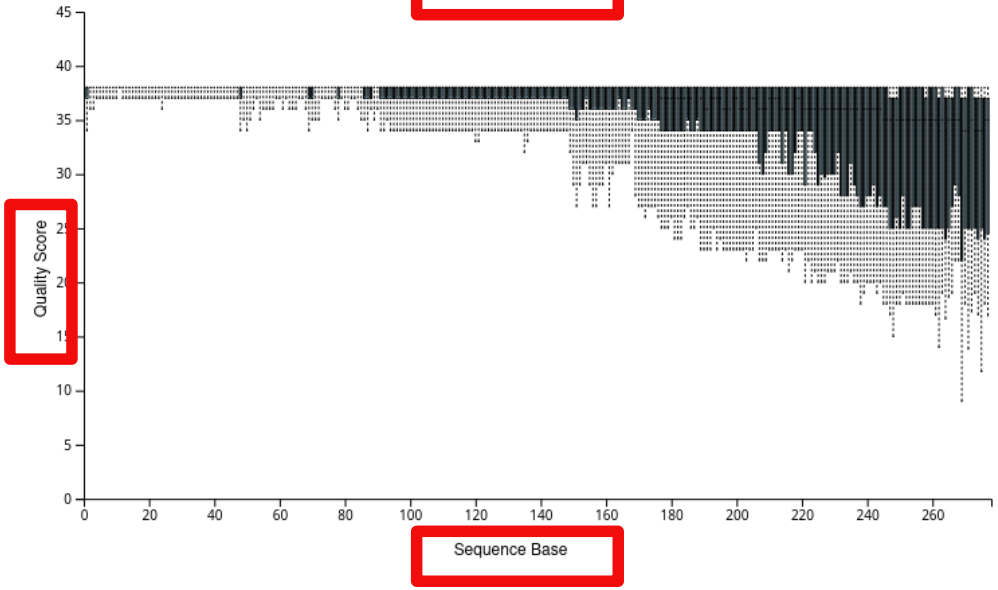
STEP 3

qiime2view File: **demux-paired-end.qzv** Visualization Details Provenance

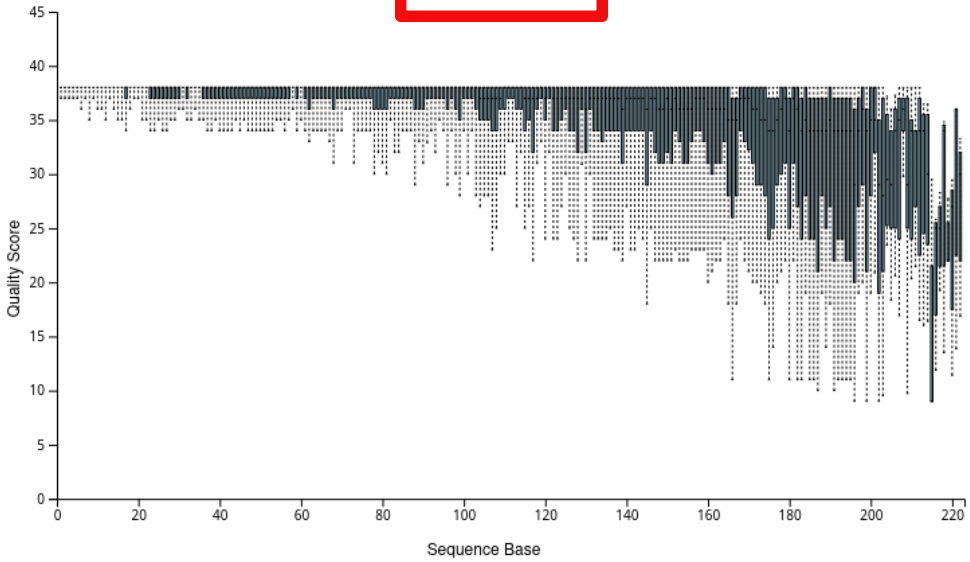
Overview Interactive Quality Plot

Click and drag on plot to zoom in. Double click to zoom back out to full size. Hover over a box to see the parametric seven-number summary of the quality scores at the corresponding position.

Forward Reads



Reverse Reads



Step 4

STEP 4

Denoise sequences, selecting sequence variants and feature table construction

QIIME 2 offers Illumina sequence denoising via **DADA2** among others.

DADA2 is an accurate, high-resolution sample inference from amplicon sequencing data.

The DADA2 package infers exact **amplicon sequence variants (ASVs)** from high-throughput amplicon sequencing data, replacing the coarser and less accurate **OTU** clustering approach (e.g. Mothur). The dada2 pipeline takes as input demultiplexed fastq files, and outputs the sequence variants and their sample-wise abundances after removing substitution and chimera errors.

For this procedure we will use the **dada2 denoise-paired** QIIME2 plugin, which will both **merge** and **denoise** paired-end reads. This method will also allow us to **remove the low-quality regions** of the sequences. Additionally, it also allows us to **remove our adapters** in the sequences before denoising. DADA2 requires the adapters to be removed to prevent false positive detection of chimeras as a result of degeneracy in the adapters.

The DADA2 denoise-paired method requires four parameters:

```
--p-trunc-len-f n truncates each forward read sequence at position n
--p-trunc-len-r n truncates each reverse read sequence at position n
--p-trim-left-f m trims off the first m bases of each forward read sequence
--p-trim-left-r m trims off the first m bases of each reverse read sequence
```

To determine what values to use for these parameters, we need to look at the Interactive Quality Plot tab in the demux-paired-end.qzv file that was generated by **qiime demux summarize**.

When viewing the quality plot look for the point in the forward and reverse reads where **quality scores decline below 25-30**. We will need to trim reads around this point to create high quality sequence variants.

Also, we will be removing the adapter sequences. We will set the optional --p-trim-left-f and --p-trim-left-r parameters to the length of the primer sequences to remove them before denoising.

STEP 4

This step is crucial. Please carefully examine „demux-paired-end.qzv” file in view.qiime2.org website

Before running the cell below, please replace n's and m's!

In [63]:
```
!time qiime dada2 denoise-paired \
--i-demultiplexed-seqs demux-paired-end_{your_number}.qza \
--p-trunc-len-f n \
--p-trunc-len-r n \
--p-trim-left-f m \
--p-trim-left-r m \
--o-representative-sequences rep-seqs_{your_number}.qza \
--o-table table_{your_number}.qza \
--o-denoising-stats stats_{your_number}.qza \
--p-n-threads 1 | batch

print('>>> finished <<<')
print('Your reads are now trimmed, truncaded and denoised!')
```

Replace n’s and m’s with values you deduced from „demux-paired-end.qzv” file.

Output artifact with summary statistics (needed in step 5)

Artifact with the main results of this step. Contains cleaned sequences. It will be used in step 6.1

```
warning: commands will be executed using /bin/sh
job 244 at Tue Jan 16 10:26:00 2024

real    0m47,977s
user    0m47,734s
sys     0m2,228s
>>> finished <<<
Your reads are now trimmed, truncaded and denoised!
```

Step 5.1

Converting stats.qza to .qzv

By convering stats.qza to .qzv we will be able to observe statistics of Step 4. Again, we need to convert .qza file into .qzv file.

Note, that Step 4 produced 3 output files. For visualization we need only *stats.qzv* file.

To convert .qza file, please run the cell below.

In [64]:
```
!qiime metadata tabulate \
--m-input-file stats_{your_number}.qza \
--o-visualization stats_{your_number}.qzv | batch

print('>>> finished <<<')
```

```
warning: commands will be executed using /bin/sh
job 245 at Tue Jan 16 10:27:00 2024
>>> finished <<<
```

Step 5.2

Now, please download stats.qzv file (using Home Page tab) to your computer.

If unsure how to download and upload your .qzv file, please follow the guide from step #3.2.

STEP 6

Step 6.1

Assign taxonomy

ASVs are of limited usefulness by themselves. We are often more interested in what type of bacterial strains are present in our samples, not just the **diversity** of the samples. So, to identify these sequence variants, we require:

```
(1) a reference database
(2) an algorithm for identifying the sequence using the database.
```

In the following section we begin exploring the bacterial taxonomic composition of the samples and relate that to our sample metadata.

We will now start to assign the taxonomy to the sequences in our FeatureData[Sequence] QIIME 2 artifact (i.e. **rep-seqs.qza** file).

We will use a pre-trained Naive Bayes classifier already provided by QIIME 2 project and the **q2-feature-classifier plugin**.

The pre-trained Naive Bayes classifier that we will use in this tutorial was trained on the SILVA database.

In [65]:

```
!time qiime feature-classifier classify-sklearn \
--i-classifier ../db/silva-138-99-515-806-nb-classifier.qza \
--i-reads rep-seqs_{your_number}.qza \
--o-classification taxonomy_{your_number}.qza \
--p-n-jobs 2 | batch

print('>>> finished <<<')
print('>>> You have successfully assigned taxonomy! <<<')
```

(2) (1) Output of step 4

```
warning: commands will be executed using /bin/sh
job 246 at Tue Jan 16 10:27:00 2024

real    6m2,544s
user    7m52,297s
sys     0m15,813s
>>> finished <<<
>>> You have successfully assigned taxonomy! <<<
```

STEP 6

Step 6.2

Create Taxa Barplot

We are almost done with the QIIME2 analysis, we have all the information we need to begin visualization of our data. QIIME2 offers plugins to calculate and visualize various metrics. However, to make things a little bit more interesting, we will use Python to visualize our data.

But first, to show that QIIME2 also can be used as a visualization software, we will create "Taxa Barplot" which is commonly added to the results of outsourced bioinformatic analysis.

Run the cell below. This will produce *taxa-bar-plots.qzv* file. Download this file from Home Page (as previously) and upload to https://view.qiime2.org

In [67]:
```
!time qiime taxa barplot \
    --i-table table_{your_number}.qza \
    --i-taxonomy taxonomy_{your_number}.qza \
    --o-visualization taxa-bar-plots_{your_number}.qzv

print('>>> finished <<<')
print('>>> You have successfully created .qzv file! <<<')
```

```
Saved Visualization to: taxa-bar-plots_2.qzv

real    0m5,506s
user    0m5,459s
sys     0m1,104s
>>> finished <<<
>>> You have successfully created .qzv file! <<<
```

What do you think?

Now, as we have got everything we needed, we are slowly leaving QIIME2 software. The one last thing is to export QIIME2 results into .tsv file.

Step 7

STEP 7

Export assigned taxonomy (from QIIME2 artifact (.qza) to text file, that could be used in subsequent steps)

If you run the cell below, you will convert QIIME2 .qza file with taxonomy assignment to .tsv file (Tab Separated Values, which can be opened in Notapad or Excel)

In [20]:
```
!qiime taxa collapse \
    --i-table table_{your_number}.qza \   #
    --i-taxonomy taxonomy_{your_number}.qza \   #
    --p-level 6 \   # taxonomic level to export
    --o-collapsed-table collapsed-table_{your_number}.qza | batch

!qiime tools export \
    --input-path collapsed-table_{your_number}.qza \
    --output-path exported-feature-table_{your_number} | batch

!biom convert -i exported-feature-table_2/feature-table.biom -o table.from_biom.tsv --to-tsv

print('>>> finished <<<')
print('>>> You have now coverted QIIME2 output to the "table.from_biom.tsv" file!<<<')
```

```
warning: commands will be executed using /bin/sh
job 230 at Wed Jan 10 12:35:00 2024
warning: commands will be executed using /bin/sh
job 231 at Wed Jan 10 12:35:00 2024
>>> finished <<<
>>> You have now coverted QIIME2 output to the "table.from_biom.tsv" file!<<<
```

Congratulations! We are now done with qiime2 analysis. Your results are now safely saved in a text file. Lets proceed to analyze your data.

STEP 8

Step 8

Data analysis using Python

First, lets inspect our file. Run the cell below, to use Linux 'head' command, which will display first few lines of specified file:

In [21]:
```
!head table.from_biom.tsv
```

```
# Constructed from biom file
#OTU ID 10k-V3-02-22-R24        10k-V3-05-22-R23        10k-V3-07-21-R23
d__Bacteria;p__Actinobacteriota;c__Acidimicrobiia;o__Microtrichales;f__Microtrichaceae;g__Candidatus_Microthrix 44
5.0     384.0   178.0
d__Bacteria;p__Bacteroidota;c__Bacteroidia;o__Chitinophagales;f__Chitinophagaceae;__    100.0   210.0   124.0
d__Bacteria;p__Actinobacteriota;c__Actinobacteria;o__Micrococcales;__;__        95.0    109.0   135.0
d__Bacteria;p__Chloroflexi;c__Chloroflexia;o__Chloroflexales;__;__       41.0    31.0    214.0
d__Bacteria;p__Proteobacteria;c__Gammaproteobacteria;o__Burkholderiales;f__Rhodocyclaceae;__    92.0    201.0   67.
0
d__Bacteria;p__Proteobacteria;c__Gammaproteobacteria;o__Competibacterales;f__Competibacteraceae;g__Candidatus_Compe
tibacter        132.0   176.0   177.0
d__Bacteria;p__Actinobacteriota;c__Actinobacteria;o__Micrococcales;f__Intrasporangiaceae;g__Lapillicoccus       77.
0       121.0   90.0
d__Bacteria;p__Chloroflexi;c__Anaerolineae;o__Caldilineales;f__Caldilineaceae;g__uncultured     158.0   220.0   23
7.0
```

Step 8.2

Next, load the text file form the step above, containing qiime2 results into "pandas":

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Run the cell below, your file will be loaded into pandas "DataFrame" and its contents will appear as an output ¶

In [49]:
```
taxonomic_df = pd.read_csv('table.from_biom.tsv', sep='\t', skiprows=1)
taxonomic_df
```

Out[49]:

	#OTU ID	10k-V3-02-22-R24	10k-V3-05-22-R23	10k-V3-07-21-R23
0	d__Bacteria;p__Actinobacteriota;c__Acidimicrob...	445.0	384.0	178.0
1	d__Bacteria;p__Bacteroidota;c__Bacteroidia;o__...	100.0	210.0	124.0
2	d__Bacteria;p__Actinobacteriota;c__Actinobacte...	95.0	109.0	135.0
3	d__Bacteria;p__Chloroflexi;c__Chloroflexia;o__...	41.0	31.0	214.0
4	d__Bacteria;p__Proteobacteria;c__Gammaproteoba...	92.0	201.0	67.0
...	...	...	...	...
194	d__Bacteria;p__Cyanobacteria;c__Sericytochroma...	0.0	6.0	0.0
195	d__Bacteria;p__Actinobacteriota;c__Acidimicrob...	0.0	3.0	0.0
196	d__Bacteria;p__Desulfobacterota;c__Desulfuromo...	0.0	0.0	3.0
197	d__Bacteria;p__Chloroflexi;c__KD4-96;o__KD4-96...	2.0	0.0	0.0
198	d__Bacteria;p__Spirochaetota;c__Leptospirae;o_...	2.0	0.0	0.0

199 rows × 4 columns

STEP 8

Step 8.3

As you can see, the #OTU ID column contains all taxonomic information. We don't need all that information right now. Lets make it simpler, by running the cell below:

In [50]:
```
split = taxonomic_df['#OTU ID'].str.split(';',expand=True)
taxonomic_df = taxonomic_df.join(split[5])
taxonomic_df.drop(columns='#OTU ID', inplace=True)
taxonomic_df.rename(columns={5:"#OTU ID"}, inplace=True)
taxonomic_df.set_index('#OTU ID', inplace=True)
taxonomic_df
```

Out[50]:

#OTU ID	10k-V3-02-22-R24	10k-V3-05-22-R23	10k-V3-07-21-R23
g__Candidatus_Microthrix	445.0	384.0	178.0
__	100.0	210.0	124.0
__	95.0	109.0	135.0
__	41.0	31.0	214.0
__	92.0	201.0	67.0
...	...	...	...
g__Sericytochromatia	0.0	6.0	0.0
__	0.0	3.0	0.0
g__Pseudopelobacter	0.0	0.0	3.0
g__KD4-96	2.0	0.0	0.0
g__Leptospiraceae	2.0	0.0	0.0

199 rows × 3 columns

Much better! With our table clean and ready, lets proceed with the analysis!

Step 9.1

Now we can calculate the shannon alpha diversity metrics

Run the cell below to calculate shannon diversity metrics!

In [52]:
```
counts = []
for x in taxonomic_df.columns: counts.append(list(taxonomic_df[x].values))

ids = []
for x in list(taxonomic_df.columns): ids.append(x)


adiv_shannon = alpha_diversity('shannon', counts, ids)
adiv_shannon
```

Out[52]:
```
10k-V3-02-22-R24    6.201760
10k-V3-05-22-R23    6.091653
10k-V3-07-21-R23    6.045206
dtype: float64
```

Thats it! So easy!

STEP 9

Step 9.2

Now calculate other alpha diversity metrics by yourself. To list all available metrics, plese run, the following command, and proceed:

In [45]:
```
get_alpha_diversity_metrics()
```

Out[45]:
```
['ace',
 'berger_parker_d',
 'brillouin_d',
 'chao1',
 'chao1_ci',
 'dominance',
 'doubles',
 'enspie',
 'esty_ci',
 'faith_pd',
 'fisher_alpha',
 'gini_index',
 'goods_coverage',
 'heip_e',
 'kempton_taylor_q',
 'lladser_ci',
 'lladser_pe',
 'margalef',
 'mcintosh_d',
 'mcintosh_e',
 'menhinick',
 'michaelis_menten_fit',
 'observed_otus',
 'osd',
 'pielou_e',
 'robbins',
 'shannon',
 'simpson',
 'simpson_e',
 'singles',
 'strong']
```

Please note that the results of some metrics are not single values. If you get an error in any of the following steps, please change your diversity metrics.

Step 9.3

What will you choose? Please write your metric of choice in the cell below:

In [66]:
```
#                        _please write metric between brackets '' and run the cell
metric_of_your_choice = 'heip_e'
adiv_yourchoice = alpha_diversity(metric_of_your_choice, counts, ids)
adiv_yourchoice
```

Out[66]:
```
10k-V3-02-22-R24    0.545913
10k-V3-05-22-R23    0.516906
10k-V3-07-21-R23    0.565540
dtype: float64
```

Great!

Put your metric between the `` and run the Cell

STEP 9

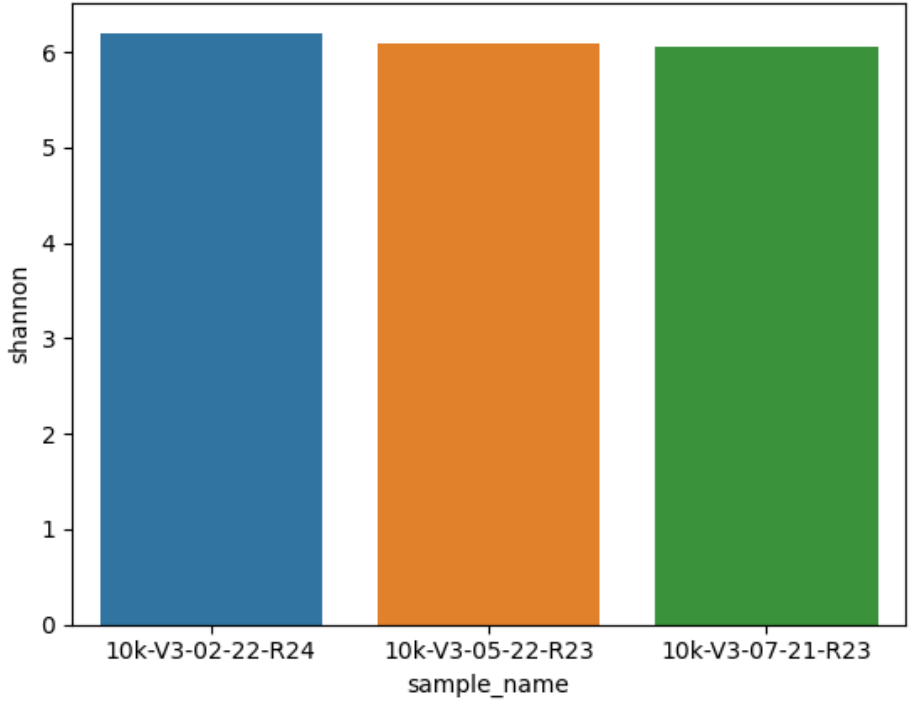
Step 9.4

Now lets visualize our results!

Running the cell below, will show shannon diversity metrics as a simple barplot. Go ahead and run it!

In [63]:
```
adiv_shannon_df = adiv_shannon.to_frame().reset_index()
adiv_shannon_df.rename(columns = {'index':'sample_name', 0:'shannon'}, inplace=True)
%matplotlib inline
sns.barplot(data=adiv_shannon_df, x='sample_name', y='shannon')
```

Out[63]:
```
<AxesSubplot: xlabel='sample_name', ylabel='shannon'>
```



Step 9.5

Great! But what about your other metric? Now we will merge results of both metrics in a single table and then plot them toghether.

Running the cell below will merge shannon metrics with metric of your choice (that you choose previously)

In [67]:
```
adiv_yourchoice_df = adiv_yourchoice.to_frame().reset_index()

adiv_yourchoice_df.rename(columns= {'index':'sample_name', 0:metric_of_your_choice}, inplace=True)

adiv_shannon_and_yourchoice_df = adiv_shannon_df.merge(adiv_yourchoice_df, on='sample_name')
adiv_shannon_and_yourchoice_df
```

Out[67]:

	sample_name	shannon	heip_e
0	10k-V3-02-22-R24	6.201760	0.545913
1	10k-V3-05-22-R23	6.091653	0.516906
2	10k-V3-07-21-R23	6.045206	0.565540

STEP 9

Step 9.6

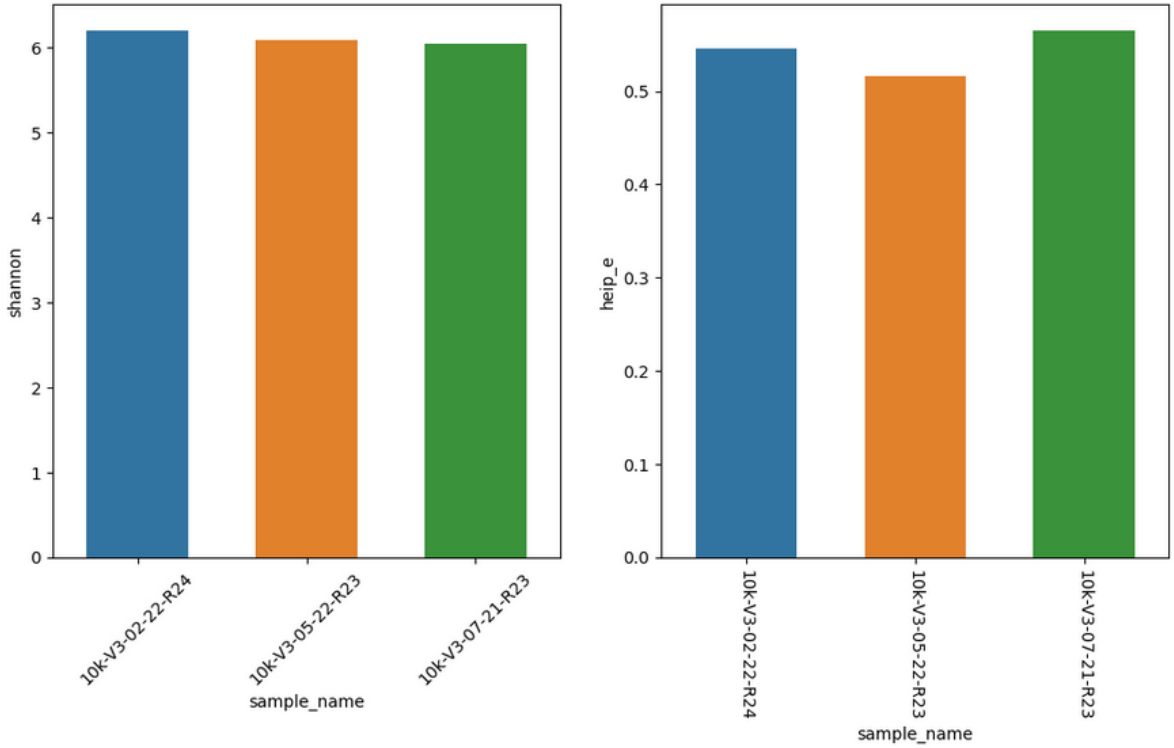
Now we can plot both results on a single figure

In [70]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))


sns.barplot(data=adiv_shannon_and_yourchoice_df, x='sample_name', y='shannon', ax=ax1, width=0.6)
sns.barplot(data=adiv_shannon_and_yourchoice_df, x='sample_name', y=metric_of_your_choice, ax=ax2, width=0.6)

ax1.tick_params(axis='x', labelrotation=45)
ax2.tick_params(axis='x', labelrotation=-90)


# ZMIANA KOLORÓW JESZCZE NP
```



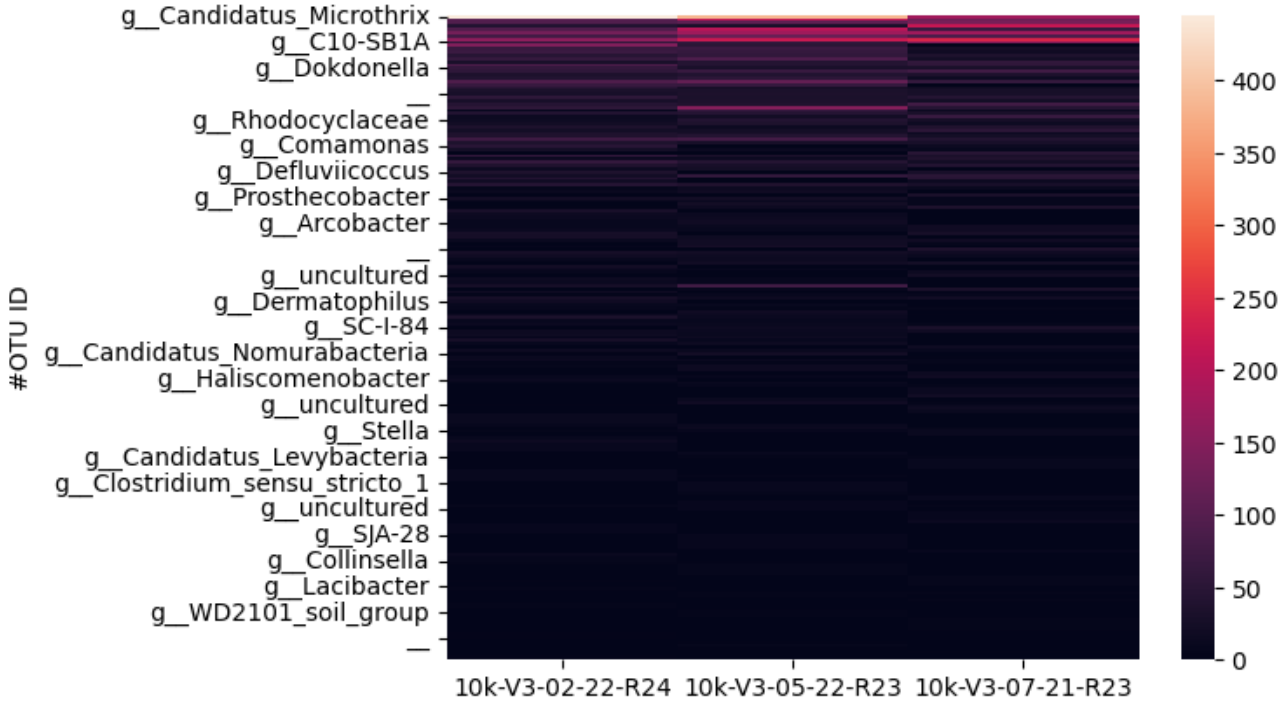
STEP 10

Step 10.1

Heatmap

Run the Cell below to draw a simple heatmap from our data

In [72]:
```
%matplotlib inline
heatmap = sns.heatmap(taxonomic_df)
```



STEP 10

Step 10.2

Its not realy readible, lets select top 20 species:

In [74]:
```
# df.iloc lets us select

taxonomic_df_top = taxonomic_df.iloc[:20]
taxonomic_df_top
```

Out[74]:

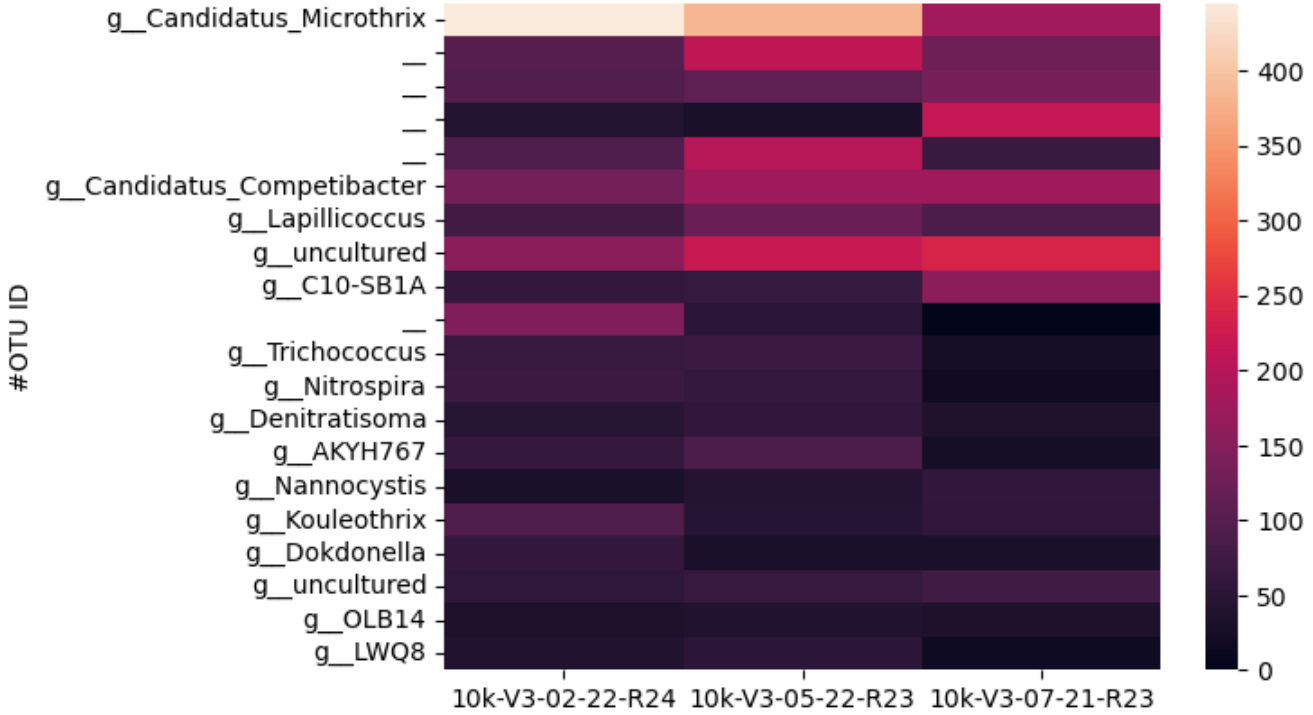
#OTU ID	10k-V3-02-22-R24	10k-V3-05-22-R23	10k-V3-07-21-R23
g__Candidatus_Microthrix	445.0	384.0	178.0
__	100.0	210.0	124.0
__	95.0	109.0	135.0
__	41.0	31.0	214.0
__	92.0	201.0	67.0
g__Candidatus_Competibacter	132.0	176.0	177.0
g__Lapillicoccus	77.0	121.0	90.0
g__uncultured	158.0	220.0	237.0
g__C10-SB1A	60.0	64.0	156.0
__	143.0	50.0	0.0
g__Trichococcus	65.0	69.0	23.0
g__Nitrospira	69.0	62.0	19.0
g__Denitratisoma	46.0	58.0	35.0
g__AKYH767	62.0	89.0	23.0
g__Nannocystis	30.0	42.0	58.0
g__Kouleothrix	91.0	44.0	57.0
g__Dokdonella	60.0	30.0	30.0
g__uncultured	54.0	66.0	75.0
g__OLB14	34.0	39.0	34.0
g__LWQ8	39.0	51.0	17.0

STEP 10

Step 10.3

And replot our heatmap

In [77]:
```
%matplotlib inline
heatmap_top = sns.heatmap(taxonomic_df_top)
```



Congratulations! Now we are done!