Week 8: Classifying taxonomy of short reads with mothur

Rika Anderson

Abstract

This week we're going to take a closer look at the taxonomy of your datasets using mothur. We'll classify taxonomy and diversity using the metagenomic reads that mapped to 16S ribosomal RNA. The Tara Oceans folks have already pulled out the reads that match 16S ribosomal RNA, so we're going to use those datasets.

Citation: Rika Anderson Week 8: Classifying taxonomy of short reads with mothur. **protocols.io**
dx.doi.org/10.17504/protocols.io.g7tbznn
Published: 04 May 2017

Protocol

Asking questions about taxonomy

Step 1.

Today we're going to learn how to use sequence data to assess diversity across samples. We'll learn how to:

1) classify the distribution of different taxa in different datsets (i.e. what % are Procholorococcus, what % are SAR11, what % are Thaumarchaeota, and so on)

2) calculate diversity metrics to determine which sample sites had higher richness or evenness.

The Tara Ocean people have already pulled out all of the metagenomic reads that matched 16S ribosomal RNA from their datasets. Recall that 16S ribosomal RNA is highly conserved, and this gene is often used as a 'barcode' for indicating the taxonomy of a specific genome.

However, we're going to do things a little differently this week: rather than have you ask a question or test a hypothesis at the **end** of the lab today, you're going to ask your question or write a hypothesis at the **beginning**.

Take a moment to develop a question or hypothesis about taxonomic abundance and/or diversity across different sample sets.

For example:

1) I hypothesize that there will be a higher abundance of SUP05 in the North Pacific mesopelagic zone compared to the South Pacific mesopelagic zone because (xyz).

2) I hypothesize that the taxonomic diversity will be highest in the mesopelagic zone compared to all other depths in the North Atlantic sample site because (xyz).

You can refine these questions or hypotheses later on, but you should at least have a general idea of what you'd like to investigate prior to continuing because it will determine which sample sets you will investigate for the rest of lab today.

For example, if you were to test hypothesis #1, you'd want to compare the 16S rRNA datasets for the North Pacific and the South Pacific mesopelagic zones. If you were to test hypotheis #2, you'd want to compare the 16S rRNA datasets for all three sample depths in the North Atlantic sample site.

Using mothur to profile the taxonomy of your datase

Step 2.

Once you have come up with a general question or hypothesis, log on to liverpool using ssh on your Terminal. Make a new directory and change into it.

cmd COMMAND

```
mkdir ~/project_directory/taxonomy
cd project_directory/taxonomy
```

Copy over dat

Step 3.

There are 16S data files for each of your project regions in your project dataset folders. Copy over all of the relevant 16S rRNA files that you will need to address your question or hypothesis into your new taxonomy folder.

cmd COMMAND

```
cp /usr/local/data/Tara_datasets/[project sample site of interest]/[16S rRNA data file of i
nterest] .
```

For example, if you wanted to copy over all of the 16S rRNA datasets from the Arabian Sea, you could use the asterisk (wildcard) and type: cp /usr/local/data/Tara_datasets/Arabian_Sea/*16S* .

Using mothur

Step 4.

Now, we're going to use a program called "mothur" to analyze these sequences. mothur is a bit different from other programs that we've used in that we can enter a mothur interface and type commands that are specific to mothur. To enter the mothur program, simply type "mothur".

Fun fact: The first version of this computer program was called "DOTUR". The next version was called "SONS." The final version was called "mothur" and it's been that way ever since.

cmd COMMAND
```
mothur
```

Create the groups file

Step 5.

Right now, the 16S rRNA-matching reads from each sample site are in separate files. Pretty soon we are going to merge all of your FASTA files together in order to compare them. Before we do that, we need to tell mothur how to tell them apart once they are merged. We do that with the make.group command. Replace the file names below with your 16S rRNA fasta file names. Substitute the group names with a suitable name for your sample so that you can recognize it, like 'NPacific_DCM' or 'Arabian_surface.'

Note: any time you see something in [brackets] in a command, it means you have to substitute that with your own data.

cmd COMMAND
```
make.group(fasta=[file1.fasta]-[file2.fasta]-[file3.fasta], groups=[group1]-[group2]-
[group3])
```
For example: make.group(fasta=ERR598944_MERGED_FASTQ_16SrRNA_10000.fasta-ERR599001_MERGED_FASTQ_16SrRNA_10000.fasta-ERR599078_MERGED_FASTQ_16SrRNA_10000.fasta, groups=meso-transect-surface)

Create the groups file

Step 6.

This command should generate a file that ends in ".groups.” Take a look at it with less. You will see that each sequence name is identified with the group name that you provided. This file will be essential for allowing mothur to compare groups later on.

Merge the FASTA files together

Step 7.

Now you can merge all of your FASTA files together, and the .groups file will record which sequences came from which file.The output here will be a file called merged.fa. Again, substitute "file-1.fa" and so on with the names of your 16S rRNA fasta files.

Hint: use the up arrow on your keyboard to call up the last command you typed, and then edit that command instead of retyping all of the filenames again.

cmd COMMAND
```
merge.files(input=[file1.fa]-[file2.fa]-[file3.fa], output=merged.fa)
```
For example: merge.files(input=ERR598944_MERGED_FASTQ_16SrRNA_10000.fasta-

ERR599001_MERGED_FASTQ_16SrRNA_10000.fasta-
ERR599078_MERGED_FASTQ_16SrRNA_10000.fasta, output=merged.fa)

Align the sequences to a reference alignment

Step 8.

Everything we've done so far is basically record-keeping. Now it's time to do real science stuff. In order to compare the sequences to each other, we need to align them. There are many different ways to do this. For this very large 16S rRNA dataset, we are going to align them to the SILVA alignment. (It's a very good, well-curated 16S rRNA database made by Germans.) I've already downloaded it for you, you just need to point the path to the right file. The "flip=true" parameter means that mothur will also try to check the reverse complement if the original sequence doesn't make a good alignment.

This alignment step will take 5-6 minutes.

cmd COMMAND

```
align.seqs(fasta=merged.fa, reference=/usr/local/data/silva.seed_v119.align, flip=true)
```

Examine the alignment

Step 9.

Whenever you make a gigantic alignment, the result is likely to contain some oddly aligned sequences. It is not practical to view the alignment in an alignment editor and manually correct the mistakes. But with mothur, we can view a statistical summary of the alignment

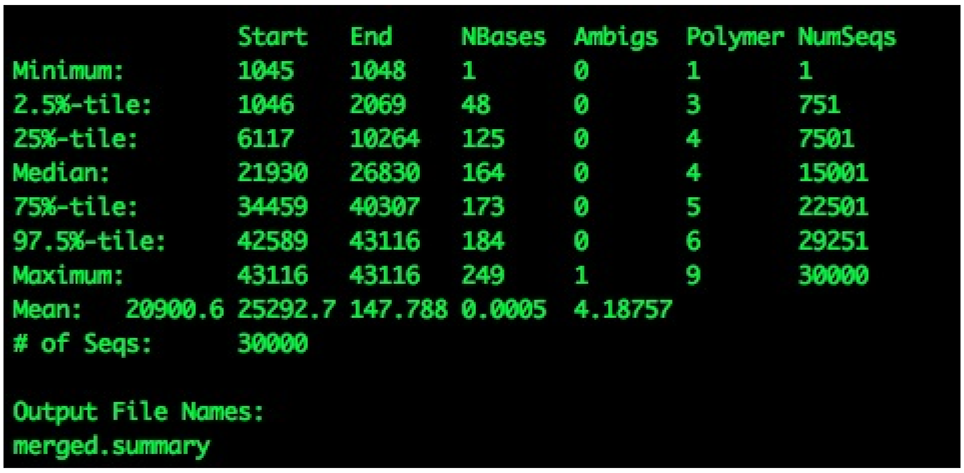
cmd COMMAND

```
summary.seqs(fasta=merged.align)
```

Examine the alignment

Step 10.

You should get an output that looks something like this:



```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1045    1048    1       0       1       1
2.5%-tile:      1046    2069    48      0       3       751
25%-tile:       6117    10264   125     0       4       7501
Median:         21930   26830   164     0       4       15001
75%-tile:       34459   40307   173     0       5       22501
97.5%-tile:     42589   43116   184     0       6       29251
Maximum:        43116   43116   249     1       9       30000
Mean:   20900.6 25292.7 147.788 0.0005  4.18757
# of Seqs:      30000

Output File Names:
merged.summary
```

This output reports that the smallest sequence in the dataset is 1 bases, and the longest is 249 bases. At least 97.5% of the sequences are longer than 184 bases.

The Start and End columns report where on the reference alignment the sequences were placed. This output reports that 97.5% of the sequences started the alignment at or before position 1046 and ended their alignment after position 2069.

The polymer column reports the longest stretch of a repeated single base (eg. AAAAAA). 97.5% of the sequences have a max homopolymer length of less than 6. The longest is 9, which is very unlikely to occur in a real 16S rRNA molecule and probably indicates that this sequence is probably low-quality. Remember, however, that this was a much bigger problem for 454 sequences, and these were sequenced with Illumina. We aren't going to worry too much about homopolymers here.

Trim the alignment

Step 11.

Let's use the screen.seqs command to exclude any sequences that start after the 97.5% tile start position or end before the 2.5% tile end position and also exclude any sequences shorter than the median number of bases. You will have to substitute those numbers in the command below. These criteria should eliminate the short sequences and the sequences that don't align in the correct region. You could choose more stringent or more relaxed criteria if you wish. I'd recommend playing around with this if you end up with too few sequences or if you get results that look like they have arisen from bad sequences that didn't get removed.

The groups file will reflect the name of all the sample fasta files and will end in '.groups.'

cmd COMMAND

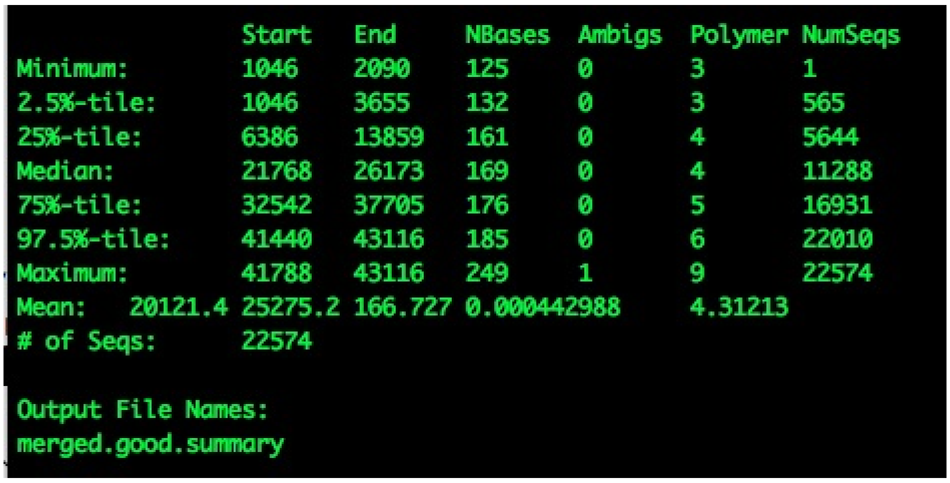
```
screen.seqs(fasta=merged.align, group=[groups file], start=[97.5% tile], end=[2.5% tile], minlength=[25% tile])
```

For example: screen.seqs(fasta=merged.align, name=merged.names, group=ERR598944_MERGED_FASTQ_16SrRNA_10000.ERR599001_MERGED_FASTQ_16SrRNA_10000.ERR599078_MERGED_FASTQ_16SrRNA_10000.groups, start=42589, end=10264, minlength=162) This eliminates any sequences that started after 4258, ended before 10264, or were shorter than 162.

Trim the alignment

Step 12.

Now if you view the statistical summary again (note that we now have new filenames), you will see that the total number of sequences has gone down. It might make you sad to throw away data, but the criteria we used were fairly forgiving, so the data that we dumped was probably bad data. (You should have started with 10,000 sequences for every input fasta file you started with. For example, I started with 3 fasta files, so I started with 30,000. If you lose more than half your data, however, you might reconsider your screening parameters.) Here is what my summary looked like:


```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1046    2090    125     0       3       1
2.5%-tile:      1046    3655    132     0       3       565
25%-tile:       6386    13859   161     0       4       5644
Median:         21768   26173   169     0       4       11288
75%-tile:       32542   37705   176     0       5       16931
97.5%-tile:     41440   43116   185     0       6       22010
Maximum:        41788   43116   249     1       9       22574
Mean:   20121.4 25275.2 166.727 0.000442988     4.31213
# of Seqs:      22574

Output File Names:
merged.good.summary
```

cmd COMMAND

```
summary.seqs(fasta=merged.good.align)
```

Trim the alignment

Step 13.

Next we need to get rid of overhangs and all of the unnecessary gaps in the alignment with the filter.seqs command.

The vertical=T option removes all columns that contain only gaps. Basically, this just cleans up and simplifies the alignment, making it much easier to work with.

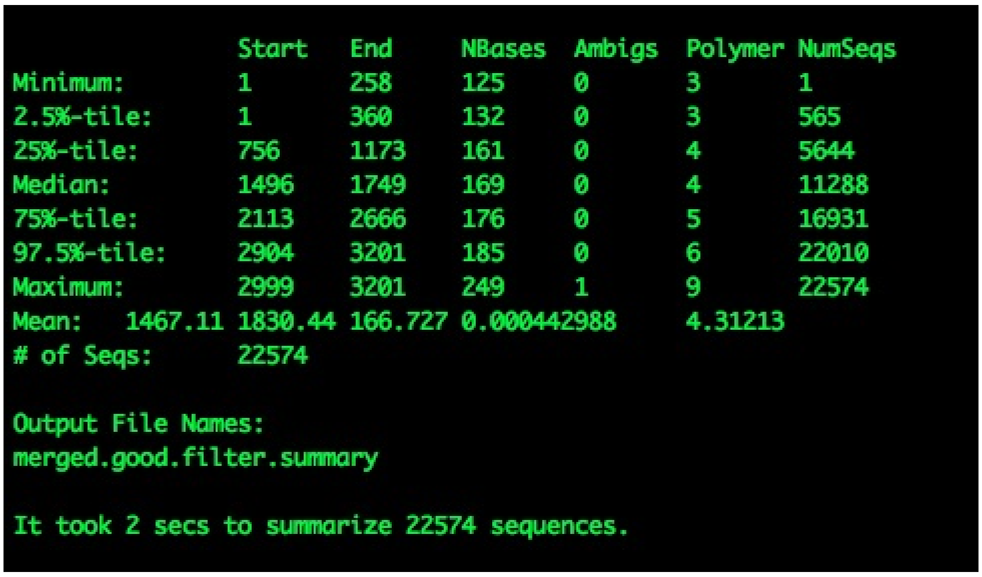
cmd COMMAND

```
filter.seqs(fasta=merged.good.align, vertical=T)
```

Trim the alignment

Step 14.

Let's see what our alignment looks like now. Here's what my output for summary.seqs looks like:


```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       258     125     0       3       1
2.5%-tile:      1       360     132     0       3       565
25%-tile:       756     1173    161     0       4       5644
Median:         1496    1749    169     0       4       11288
75%-tile:       2113    2666    176     0       5       16931
97.5%-tile:     2904    3201    185     0       6       22010
Maximum:        2999    3201    249     1       9       22574
Mean:   1467.11 1830.44 166.727 0.000442988     4.31213
# of Seqs:      22574

Output File Names:
merged.good.filter.summary

It took 2 secs to summarize 22574 sequences.
```

cmd COMMAND

```
summary.seqs(fasta=merged.good.filter.fasta)
```

Trim the alignment

Step 15.

You will probably need to try a variety of criteria with the screen.seqs and filter.seqs command to get a result that you like for your dataset. There is no “correct” solution to deciding exactly where to trim or how many sequences to keep, and a lot of trial and error may be required before you get a satisfying result. Just be sure to keep your filenames straight. If you go back and repeat the screen.seqs command, move all of the previous folder to a new folder with a descriptive name like “screened.end2410.length234” so you know which commands generated which data.

Before we go any further, let's rename our current files to make things simpler: You can use the "system" command in mothur to run programs outside of mothur without leaving the mothur environment.

We're going to rename “merged.good.filter.fasta” as “final.fasta”
and your 'xxx.good.groups' file to “final.groups” (the actual name of your original groups file will depend on the fasta files you put into it)

cmd COMMAND

```
system(mv merged.good.filter.fasta final.fasta)
system(mv [some filename].good.groups final.groups)
```

For the group file, here is an example: system(mv ERR598944_MERGED_FASTQ_16SrRNA_10000.ERR599001_MERGED_FASTQ_16SrRNA_10000.ERR599078_MERGED_FASTQ_16SrRNA_10000.good.groups final.groups)

Compute distances among aligned sequences

Step 16.

Now that we have a good alignment, we can choose to use that alignment in a variety of different ways. First, though, we're going to take eliminate redundancy in the final file by eliminating identical sequences. Reducing redundancy will make mothur do things faster. To do this we will use the unique.seqs command. This will generate a file called final.unique.fasta.

cmd COMMAND

```
unique.seqs(fasta=final.fasta)
```

Compute distances among aligned sequences.

Step 17.

One thing we can do now is compare membership of specific sequences among the groups. First, we have to calculate the distances among the sequences based on their percent similarity. This will generate pairwise distances for every combination of sequences in your alignment up to a maximum distance of 0.50. This creates a large distance matrix of all of your sequences. Distances larger than 0.5 (or 50% different) won't be meaningful or useful to us, and it will save some computation time to ignore them.

This will take about 15 minutes, so I'd recommend that you let this run and open up a new Terminal window while you move on to step 21 and come back here when it's done.

cmd COMMAND
```
dist.seqs(fasta=final.unique.fasta, cutoff=0.2)
```

Compute distances among aligned sequences.

Step 18.

When dist.seqs is done, we will cluster all of the sequences into operational taxonomic units (OTUs) according to the distances we just computed. OTUs are another way of designating closely related groups of organisms, or "species," but as we've seen, there is a lot of dispute as to what a microbial "species" really is. This will take about 5 minutes to complete.

cmd COMMAND
```
cluster(column=final.unique.dist, name=final.names)
```

Compute distances among aligned sequences.

Step 19.

We can now compare the membership of these OTUs among different samples. First, we have to make a shared file to compare shared sequences among each of our samples.

cmd COMMAND
```
make.shared(list=final.unique.an.list, group=final.groups)
```

Compute distances among aligned sequences.

Step 20.

We can visualize this information with a Venn diagram illustrating how many OTUs are shared among samples.

This will output a .svg file that you can open with a web browser or graphics program. **Save this as Figure 1. (Save the "0.01" svg file, not the "unique" file.)**

cmd COMMAND
```
venn()
```

Classify your sequences

Step 21.

Now we will classify our sequences by comparing them to a reference database. We will once again use the SILVA database to compare these sequences.

You may get some 'could not be classified' errors. You can safely ignore these. Because these are randomly sequenced metagenomic reads, some of these reads may not have lined up particularly well with the 16S rRNA reference sequences and so mothur is throwing errors and will ignore them, but we still have enough data from other sequences to get a result that is meaningful.

cmd COMMAND
```
classify.seqs(fasta=final.fasta, group=final.groups, reference=/usr/local/data/silva.seed_v
   119.align, taxonomy=/usr/local/data/silva.seed_v119.tax)
```

Classify your sequences

Step 22.

Use FileZilla or scp to transfer your files over to your local desktop. Open the file that is called 'final.seed_v119.wang.tax.summary' in Excel. (You may have to change the name so the file ends in '.txt' or Excel won't recognize it as a valid file to open.) Here is the definition of the columns, from left to right?

- Taxonomic level is in the farthest left-hand column. The lower the number, the larger the phylogenetic classification. For example, Archaea, Bacteria, Eukarya, and 'unknown' are taxonomic level 1. Taxonomic level 2 goes down a bit deeper: it classifies different phyla of Archaea, Bacteria, and Eukaryotes. Taxonomic level 3 classifies different classes of those phyla, and so on.
- The rankID provides a means of keeping track of where that particular organism falls. For example, the SAR_11 clade is rankID 0.2.17.2.9, which means it is a clade within the Alphaproteobacteria (0.2.17.2), which are a clade within the Proteobacteria (rankID 0.2.17), which is a clade within the Bacteria (rankID 0.2).
- The taxon column tells you the name of the taxon.
- The daugther level tells you how may levels down you are in the phylogeny.
- The 'total' tells you how many total sequences are within that taxonomic category.
- Each of the following columns gives you the taxonomic breakdown for that sample.

I'd recommend sorting your Excel spreadsheet by taxlevel. Make a pie chart for each sample depth based on the taxonomic distribution at taxonomic level 2 to compare the taxonomic breakdown between each of your samples. **Save these pie charts as Figure 2(abc) and provide a figure caption for this figure.**

Calculate the diversity of your sequences

Step 23.

Now we're going to calculate the diversity at each of your sample sites. These will will be pure calculations in Excel rather than on the command-line. mothur can do this using OTUs, but the OTUs we generated in mothur were at too fine a resolution to be useful for our purposes-- we want to compare taxonomic diversity at a higher taxonomic level, so we're calculating it by hand based on our taxonomic classifications.

We'll calculate diversity using two measures: species richness (r) and the Shannon-Weiner Index (H'). The Shannon-Weiner Index (H') is meant to take into account both the taxon richness (how many different taxa there are) and evenness (does one taxon dominate or are they evenly distributed?)

We are going to calculate the diversity of each of your sample sites at taxonomic level 2. This means that each entry at level 2 (i.e. 'Euryarchaeota,' 'Thaumarchaeota,' 'Proteobacteria') will count as one taxon. The number of sequences for that taxon in your sample represents the total number counted in that sample for that taxon. Calculate the following for each of your sites (i.e. you should have one H' value for the deep chlorophyll max, one for the surface, and one for the mesopelagic zone).

Species richness = r = number of taxa in your sample

The equation for the Shannon-Weiner index is:

$H' = -\Sigma (P_i \ln(P_i))$

H'= index of taxonomic diversity, the Shannon-Weiner Index

P_i = proportion (percent) of total sample belonging to the i^{th} taxon

ln= natural log (log base e = not the same as log!)

This index takes into account more than just the number of taxa (richness) in a sample, but also how evenly distributed the taxa are (evenness) within the sample. The index increases either by having more richness or by having greater evenness.

Hint: $-\Sigma (P_i \ln(P_i)) = -((P_{taxon1}*\ln(P_{taxon1})) + (P_{taxon2}*\ln(P_{taxon2})) + (P_{taxon3}*\ln(P_{taxon3})) + \ldots)$

I suggest that you start by calculating the total number of sequences for each sample site for all of taxonomic level 2. Then, for each taxon within taxlevel2, calculate the total proportion of seqences belonging to that taxon (P_i). Then calculate H'.

Excel command for natural log: LN()

Note that in Excel, LN(0) = error, so skip the cells with a value of 0.

Pay attention to your use of parentheses!

Please do these calculations in a way that is clear so that I can track your calculations. You will be submitting these Excel spreadsheets as part of your lab assignment for this week. Please compile the total number of sequences, the species richness, and the Shannon-Weiner index for each of your sample depths in a table. **Save this as Table 1 and provide a caption.**

Calculate the diversity of your sequences

Step 24.

In your Excel spreadsheet, make 5 columns containing metadata for each of your samples (1 column each for temperature, chlorophyll, nitrate, oxygen, and salinity). Make a scatterplot for each of the metadata versus your Shannon-Weiner index (H') for each of the samples. For example, one plot will have temperature on the x axis, and H' on the y axis, one will have chlorophyll on the x axis, and H' on the y axis, and so on. For each plot, plot a trendline through the data and include the equation and the R-squared value. **Save these 5 plots as Figure 3(abcde).**

Calculate the diversity of your sequences

Step 25.

When you are done calculating the diversity index for each of your samples, create a Word document in which you compile Figure 1, Figure2(abc), Figure 3(abcde) and Table 1.

Please also include your Excel spreadsheet when you submit your lab questions this week so that I can see your calculations.

Return to the question or hypothesis you wrote at the beginning of the lab. Based on the data you generated today, what is the answer to your question, or did you prove or disprove your hypothesis? Write a discussion (1-2 paragraphs) explaining your results based on what we learned in class about marine taxa and diversity in various ocean basins and at different depths.

Please also answer the following questions:

1. How many OTUs were shared among all of your sample sites? Why do you think this number was relatively low? If you had created a Venn diagram at a higher taxonomic level (such as taxonomic level 2, which is what we used to calculate the Shannon-Weiner index), would you have had a higher or lower number of shared sequences between sites? Explain why.

2. What is the difference between richness and the Shannon-Weiner index? Describe a situation in which you might have a high richness but a relatively low Shannon-Weiner index.

3. Does your taxonomic diversity, as calculated by the Shannon-Weiner index, correlate with any of the metadata for your sample (temperature, chlorophyll, nitrate, oxygen, salinity)? (The R squared value should vary between 0 and 1; the stronger the correlation, the closer the R-squared value is to 1. We did not calculate p-values or conduct a more rigorous statistical analysis, but the R-squared

value will tell you how closely the variables are correlated.) Write a short paragraph speculating on any correlations you find based on what we learned in class.