Creation of a plant reference database of meta-barcoding genes

Jiajian You
2576169Y
Supervised by Dr Umer Zeeshan Ijaz
A thesis submitted in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE IN COMPUTER SYSTEM ENGINEERING

`

Contents

`

Abstract

Biologists invented DNA barcoding technology, which uses specific DNA regions to identify plant species. Currently, researchers use plant DNA barcodes include two core barcodes matK and rbcL, two complementary barcodes psbA trnH gene spacer (ptigs) and ITS2.

This study is to develop a plant database for short amplicons. Download matK and ITS2 region sequence data from NCBI, generate fasta file and taxonomy file for those sequences. Create databases of matK and ITS2, train their classifier, and Using classifier to identify plant species and generate summary statistics of how many unique taxonomic groups (Genus, Family, Species etc) are found.

Keywords: matK, ITS2, NCBI, Meta-barcoding, Plant reference database, Taxonomy

`

Acknowledgements

I would like to express my deepest thankfulness to Dr Umer Zeeshan Ijaz and Dr Ciara Keating for providing me with the opportunity to research this project. I really appreciate Dr Umer spent lots of time meeting with Fan Zou and me every week to discuss progress and problems we met on creating a plant reference database, he answers my questions patiently and gave me advice on project planning. Dr Ciara Keating always responded to my messages no matter how late it is and she provided guidance to help me to go smoothly in the progress. What's more, when plant reference database is in trouble, she always set up a meeting to help us solve the problems.

`

1.Backgrounds and aims

1.1 Background of DNA barcode

Traditionally, plant taxonomic identification has relied upon morphological characteristic. In the last two decades, molecular tools based on DNA sequences of short standardised gene fragments, termed DNA barcodes, have been developed for species discrimination. The most common DNA barcode used in animals is a fragment of the cytochrome c oxidase (COI) mitochondrial gene, while for plants, two chloroplast gene fragments from the RuBisCo large subunit (rbcL) and maturase K (matK) genes are widely used. (Živa Fišer Pečnikar et al., 2013). Information gathered from DNA barcodes can be used beyond taxonomic studies and will have far-reaching implications across many fields of biology, including ecology (rapid biodiversity assessment and food chain analysis), conservation biology (monitoring of protected species), biosecurity (early identification of invasive pest species), medicine (identification of medically important pathogens and their vectors) and pharmacology (identification of active compounds). However, it is important that the limitations of DNA barcoding are understood and techniques continually adapted and improved as this young science matures (Živa Fišer Pečnikar et al., 2013).

The ability of DNA barcoding to distinguish species from a range of taxa and to reveal species has, nowadays, been well documented. DNA barcoding has proved useful in the study of taxonomically difficult taxa (Rivera and Currie 2009). Moreover, this technique helped to recognize different developmental life stages of a single species, which was impossible by using morphological characters alone (Živa Fišer Pečnikar et al., 2013).

1.2 Background of matK

The matK gene, formerly known as orfK, is emerging as yet another gene with potential contributions to plant molecular systematics and evolution (Johnson and Soltis, 1994, 1995; Steele and Vilgalys, 1994; Liang and Hilu, 1996; Gadek, Wilson, and Quinn, in press). The gene, 1500 base pairs (bp), is located within the intron of the chloroplast gene trnK, on the large single-copy section adjacent to the inverted repeat. The matK gene in the chloroplast DNA has evolved at a higher rate than several other genes currently used in systematic studies (Matsumoto et al., 1998). Olmstead and Palmer (1994) reported that among 20 genes used in molecular systematics, the matK gene has the highest overall nucleotide substitution rate.

Strong phylogenetic signal from matK has rendered it an invaluable gene in plant systematic and evolutionary studies at various evolutionary depths. Further, matK is proposed as the only chloroplast-encoded group II intron maturase, thus implicating MATK in chloroplast posttranscriptional processing. For a protein-coding gene, matK has an unusual evolutionary mode and tempo, including relatively high substitution rates at both the nucleotide and amino acids levels(MICHELLE M et al., 2010). These evolutionary features have

`

raised questions about matK function. In one study, it examined matK RNA and protein from representative land plant species to provide insight into functional aspects of this unusual gene(MICHELLE M et al., 2010). The study reports the first evidence of a transcript for matK separate from the trnK precursor and demonstrate that a full-length MATK protein exists in five angiosperm species. The study also shows that matK RNA and protein levels are regulated by light and developmental stage, suggesting functional roles for this putative maturase. Specifically, matK expression increased after etiolation and decreased at 4 weeks after germination. The study provides evidence for the expression of the only putative chloroplast-encoded group II intron maturase and insight into regulation mechanisms relating to plant development and, indirectly, to photosynthesis. (MICHELLE M et al., 2010)

1.3 Background of ITS2 in plants

The internal transcribed spacer 2 (ITS2) region of nuclear ribosomal DNA is regarded as one of the candidate DNA barcodes because it possesses a number of valuable characteristics, such as the availability of conserved regions for designing universal primers, the ease of its amplification, and sufficient variability to distinguish even closely related species (Hui Yao et al., 2010). However, a general analysis of its ability to discriminate species in a comprehensive sample set is lacking. The ITS2 region unveiled a different ability to identify closely related species within different families and genera. The secondary structure of the ITS2 region could provide useful information for species identification and could be considered as a molecular morphological characteristic (Hui Yao et al., 2010).

The Consortium for the Barcode of Life (CBOL) recommends the two-locus rbcL–matK combination as the universal plant DNA barcode. rbcL can be reasonably amplified across a diverse set of plants but was not variable enough to discriminate species (Claire-Iphanise Michel et al., 2016). MatK was challenging to amplify given that primers were not widely applicable and was too variable to be used solely as a universal DNA barcode. In contrast, the ITS2 barcode alone can pinpoint the taxonomic identity of majority of the species tested. ITS2 had the highest barcoding success rate of the three markers investigated in this study, and was also found to be less variable than matK but variable enough to discriminate among species (Claire-Iphanise Michel et al., 2016).

1.4 DNA sequencing and Amplicon sequence variant

DNA sequencing is the process of determining the nucleic acid sequence – the order of nucleotides in DNA. It includes any method or technology that is used to determine the order of the four bases: adenine, guanine, cytosine, and thymine. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery. (Behjati S al., 2013)

Knowledge of DNA sequences has become indispensable for basic biological research, and in numerous applied fields such as medical diagnosis, biotechnology, forensic biology,

`

virology and biological systematics. Comparing healthy and mutated DNA sequences can diagnose different diseases including various cancers, characterize antibody repertoire, and can be used to guide patient treatment.[5] Having a quick way to sequence DNA allows for faster and more individualized medical care to be administered, and for more organisms to be identified and cataloged.(Abate AR, et al.,2013)

The rapid speed of sequencing attained with modern DNA sequencing technology has been instrumental in the sequencing of complete DNA sequences, or genomes, of numerous types and species of life, including the human genome and other complete DNA sequences of many animals, plant, and microbial species. Following the development of fluorescence-based sequencing methods with a DNA sequencer, DNA sequencing has become easier and orders of magnitude faster. (Pettersson E. et al., 2009)

Amplicon sequence variant (ASV) is a term used to refer to single DNA sequences recovered from a high-throughput marker gene analysis. These amplicon reads are created following the removal of erroneous sequences generated during PCR and sequencing. This allows ASVs to distinguish sequence variation by a single nucleotide change. ASVs are utilized to classify groups of species based on DNA sequences, finding biological and environmental variation and to determine ecological patterns.

1.5 QIIME 2

QIIME 2 is a powerful, extensible, and decentralized microbiome analysis package with a focus on data and analysis transparency. QIIME 2 enables researchers to start an analysis with raw DNA sequence data and finish with publication-quality figures and statistical results. QIIME 2, a completely reengineered and rewritten system that is expected to quality control from different sequencing platforms (DADA2 and Deblur), taxonomy assignment and phylogenetic insertion, which quantitatively improve the results over QIIME 1 and other tools.

The plugins also support qualitatively new functionality, including microbiome paired-sample and time-series analysis (which are critical for studying the effects of treatments on the microbiome), and machine learning. Trained machine learning models can be saved for application to new data and interrogated to identify important microbiome features. Several recently released plugins, including q2-cscs, q2-metabolomics, q2-shogun, q2-metaphlan2 and q2-picrust2, provide initial support for analysis of metabolomics and shotgun metagenomics data. Additionally, many of the existing ‘downstream’ analysis tools, such as q2-sample-classifier, can already work with these data types individually or in combination if they are provided in a feature table. Thus, QIIME 2 has the potential to serve not only as a marker-gene analysis tool but also a multidimensional and powerful data science platform that can be rapidly adapted to analyze diverse microbiome features (E Bolyen et al., 2019).

QIIME 2 provides a software development kit that can be used to integrate it as a component of other systems (such as Qiita or Illumina BaseSpace) and to develop interfaces targeted toward users with different levels of computational sophistication. QIIME 2 provides the QIIME 2 Studio graphical user interface and QIIME 2 View, interfaces designed for end-user biologists, clinicians and policy-makers; the QIIME 2 application programming interface, designed for data scientists who want to automate workflows or work interactively in Jupyter

`

Notebooks; and q2cli and q2cwl, providing a command-line interface and CWL wrappers for QIIME 2, designed for experts in high-performance computing (E Bolyen et al., 2019).

The tools in QIIME 2 are all interoperable through plugins, exchange of files in standard formats or using multi-language environments, such as Jupyter Notebooks. For example, the BIOM format is supported by all of them. A diverse ecosystem of interoperable software is beneficial for the field, because it allows both experienced users to obtain multiple perspectives on their data and novice bioinformaticians to work in the programming environments that they are most comfortable with (for example, phyloseq allows users to work in R, whereas QIIME 2 allows users to work in Python). QIIME 2 can import data from microbiome data-sharing platforms such as Qiita, the European Bioinformatics Institute (EBI) European Read Archive and the National Center for Biotechnology Information (NCBI) Sequence Read Archive (E Bolyen et al., 2019).

1.6 Aims and Objectives

In microbiology we have reference databases e.g. silva, midas, greengenes, there is few similar database for plants. In particular, at present there's no maintained database for matK or ITS2 region. So, the overall objective of this thesis is to develop a plant database for matK or ITS2.

When we work on plant Illumina paired-end sequencing ~250bp - 300bp, we need to know which species it is and what is the taxonomy of it? A database which can identify plant species would be developed by finding known sequences that are already deposited at NCBI.

After an Amplicons processing of plant sample, and make a Qiime2 work flow for creating a new database in order to identify plant. Within the database, we use DNA based approaches to identify a plant species which ASV sequences are available.

The following is a simple workflow of my project.

1. Download matK and ITS2 sequence data from NCBI Nucleotide, generate a FASTA file of all known sequences for matK and ITS2 region, and then generate a taxonomy for those sequences, create a database.
2. After creating the databases, we can then follow up taxonomic approach to compare taxonomy assignment.
3 After the taxonomy is obtained, we can generate summary statistics of how many unique taxonomic groups (Genus, Family, Species etc) are found and generate a 2D graph of x-axis (total number of sequences) y-axis (total number of unique taxonomic groups).
4 The new plant database not only help us to classify plants and identify the species, but also help us to analyze diet of ungulates and do plant diversity surveys.

`

2. Methods

In order to develop a plant database for ITS2 and matK region, to classify plants and identify the species using the plant database classifier. I need to get four data files which are accession taxonomy file, database classifier file, representative sequences file of studies on matK or ITS2 region, and taxonomy file for each studies.

First, download a fasta file of all known sequences for matK or ITS2 region, and generate an accession taxonomy for the region. Second, use fasta file and accession taxonomy to train a new database classifier. Third, download the meta-data sequence of studies and create their representative sequences file. Forth, use database classifier file and representative sequences file to generate taxonomy file of plants.

So, in order to create reference database and taxonomy file, I make a workflow of creating database, including four steps. The workflow is shown as Figure 1.

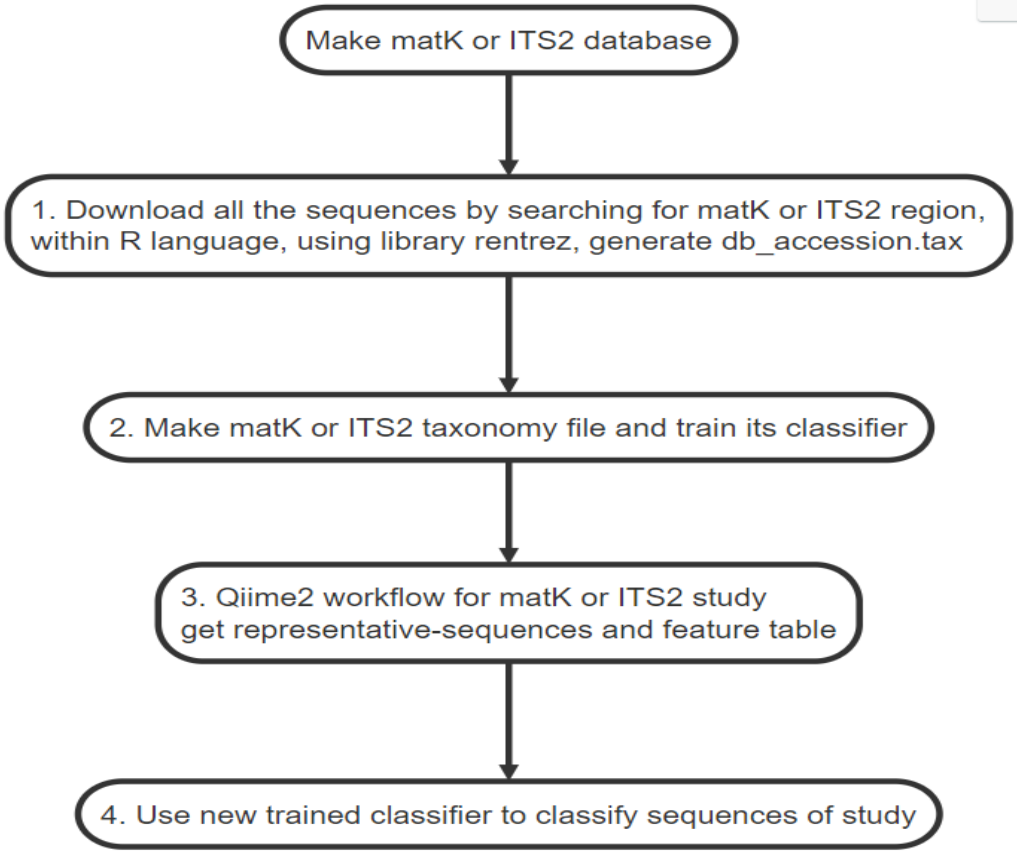


Figure 1: workflow of plant reference database creation

2.1 Make matK and ITS2 database

In this thesis, we need to build a plant barcode database. We need the plants barcode genes

`

data which contains the plant taxonomy information. The National Center for Biotechnology Information (NCBI) Nucleotide database has been an important resource for genomic, genetic, and proteomic research. This project's provision of curated and stable annotated reference genomes, and the project's provision of curated and stable annotated reference genomes, transcripts, and proteins for selected viruses, microbes, organelles, and eukaryotic organisms, has allowed researchers to focus on the best representative sequence data in contrast to the redundant data in GenBank, and to unambiguously reference specific genetic sequences (Nuala A. O'Leary et al 2016).

I download all the sequences by searching for matk or ITS2 region, with R language, using library rentrez, generated db_accession.tax for the sequences. Database has two components, fasta file and taxonomy file. In order to get a fasta file, I download all the sequences by searching for a particular region, save that as a fasta file (db.fasta), and remove everything from the name except accession IDs. To get taxonomy file of database, I extract accession IDs and store it in a txt file, named IDs_accession.txt. Within R language, I use library rentrez to generate db_accession.tax by IDs_accession.txt. It shown as Figure 2

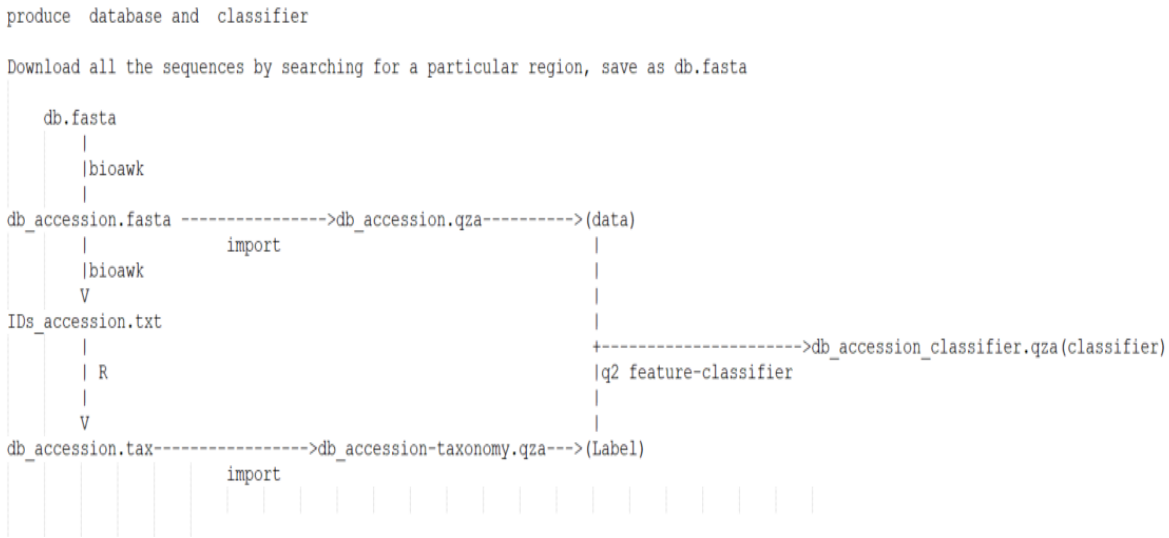


Figure 2: Flow of creating database and classifier

2.2 Make database accesion-taxonomy file and train classifier

Using qiime2 tools, I import db_accession.fasta as format of DNASequencesDirectoryFormat to file db_accession.qza, and import db_accession.tax to file db_accession-taxonomy.qza. With file db_accession.qza and file db_accession-taxonomy.qza, I use qiime2 feature-classifier to train a new classifier, and save taxonomic classifier to a file named db_accession_classifier.qza, as shown in Figure 2

2.3 Qiime2 workflow for study of a particular region

I search the literature to find studies that have researched matK or ITS2 region. I download the data and create a meta-data file.

`

In google scholar, I find studies that include "amplicon sequencing" "paired end" and "ncbi" as search terms for the genes of matK or ITS2 region. I found studies that have matK, ITS2 regions and the data uploaded to NCBI Sequence Read Archive. Then I can create a meta-data file. I record the sequencing platform, targeted regions, author name and sample names. Some of this can be taken from the paper, the rest by looking the project up on NCBI.
After searching studies on matK or ITS2 region. I find data with PRJN number, and download the paired-end reads of ASV sequences.

As a result of my literature research, I make an excel file with the author's name, paper title, BioProject number (PRJN), SRA Accession_Number, meta-barcode, type of file, size of file for each study. The excel file is shown as Table 1

Author	Paper title	BioProject	SRA Accessior	Metabarcode_1	Metabarcode_2	File Type	Size(MB)
Kingsly C. Beng	Amplicon sequencing dataset of soil fungi and associated environmental variables collected in karst and non-karst sites across Yunnan province, southwest China	PRJNA486218	SRP158134	ITS2		fastq	1374
Jana Batovska	Using Next-Generation Sequencing for DNA Barcoding: Capturing Allelic Variation in ITS2	PRJNA343434		ITS2		fastq	58
Rosemary J. Moorhouse-Gann	New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones	PRJNA393998	SRP136381	ITS2		fastq	3645
R. Scott Cornman	Taxonomic Characterization of Honey Bee (Apis mellifera) Pollen Foraging Based on Non-Overlapping Paired-End Sequencing of Nuclear Ribosomal Loci	PRJNA295334		ITS2		fastq	7141
Grace Moore	Paleo-metagenomics of North Ameican fossil packrat middens: Past biodiversity revealed by ancient DNA	PRJNA488629		ITS2		fastq	60039
Nicole A. Fahner	Large-Scale Monitoring of Plants through Environmental DNA Metabarcoding of Soil:Recovery, Resolution, and Annotation of Four DNA Markers	PRJNA318025	SRP073252	ITS2	matK	fastq	10048
Jinxin Liu	The Species Identification in Traditional Herbal Patent Medicine, Wuhu San, Based on Shotgun Metabarcoding	PRJNA663116		ITS2	matK	fastq	38459

Table 1: result of literature research

Within qiime2 workflow as shown in Figure 3, I generate file one by one.

`

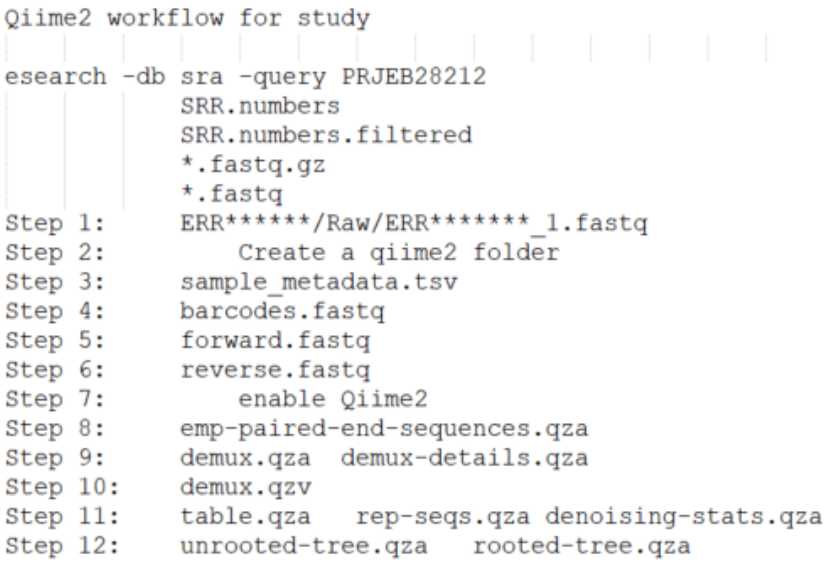


Figure 3: Qiime2 workflow for study

2.4 Use new trained classifier to classify ASV sequences of study

Once the classifier is obtained, I can classify plants using our plant database classifier, and identify species. Additionally, I can use database classifier and rep-seqs.qza file to generate taxonomy file named taxonomy.qza. The taxonomy file contains taxonomic information about species that I can visualize taxonomy file by converting the file into a visual format file named taxonomy.qzv. Finally, I create bar-plots from my taxonomy file and sample_metadata.tsv for each region to get bar-plot for phylum-level, class-level, etc.

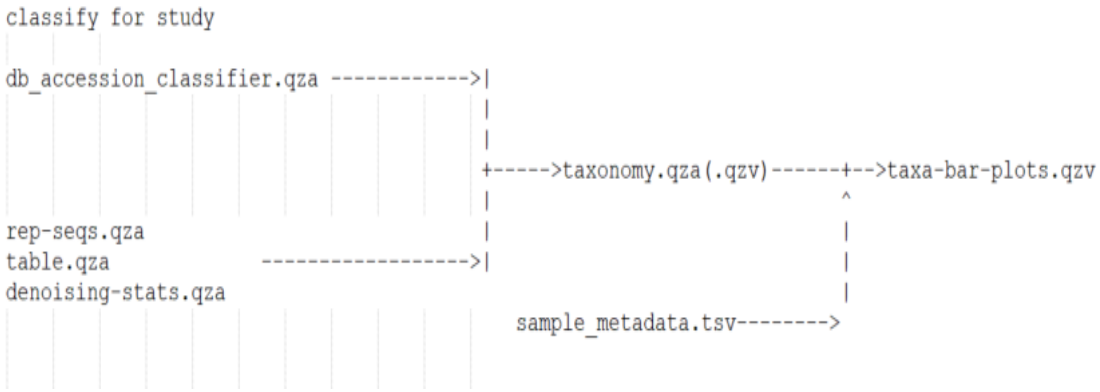


Figure 4: classify ASV sequences in study

`

3 Results

3.1 Summary of database and taxonomy

3.1.1 matK database

For the matK database, it is found 203,200 matK gene reads, and of these gene data, about 184,533 gene taxonomies can be read because of IP restrictions in NCBI.

3.1.2 Taxonomy of matK database

Using workflow in Appendix A, I generate file db_accession.fasta and IDs_accession.txt. After getting IDs_accession.txt, I use R workflow to generate db_accession.tax that I can get the summary statistics of the database at different taxonomic ranks and find amount of type in each level shown in Table 2.

Level	Type amount	Reads
Kingdom	1	184,533
phylum	1	184,533
class	21	190,175
order	147	186,738
family	558	190,146
genus	9,828	189,682
species	1,161	1,161
Unassigned	1	18,667
Total		203,200

Table 2: summary statistics of the matk databases at different taxonomic rank

3.1.3 ITS2 database

For the ITS2 database, it is found 441,100 ITS2 gene reads, and of these gene data, about 424,860 gene taxonomies can be read because of IP restrictions in NCBI.

3.1.4 Taxonomy of ITS2 database

Level	Type amount	Reads
kingdom	1	424,860
phylum	3	424,849
class	35	423,733

`

order	158	417,677
family	670	420,338
genus	9,889	420,999
species	11,548	1,790
Unassigned	1	16,240
Total	22305	441,100

Table 3: summary statistics of the ITS2 databases at different taxonomic rank

Using workflow in Appendix A, I generate file db_accession.fasta and IDs_accession.txt.

After getting IDs_accession.txt, I use R workflow to generate db_accession.tax that I can get the summary statistics of the database at different taxonomic ranks and find amount of type in each level shown in Table 3.

3.2 Qiime2 results

3.2.1 Qiime2 results for matK study

In order to get meta-sequence of matK, I use one of my searching studies which is in the result of literature research shown in Table 1. I choose one of the matK studies, which is *Large-Scale Monitoring of Plants through Environmental DNA Metabarcoding of Soil: Recovery, Resolution, and Annotation of Four DNA Markers* and data of this study is available with BioProject number, PRJNA318025. Using qiime2 workflow shown in Appendix A, I get gene data in this study, and its SRR numbers is 140 and demultiplexed sequence amounts is 41,025,444.

Item	matK
PRJN	PRJNA318025
SRR numbers	140
Demultiplexed sequence amounts	41,025,444

Table 4: Statistics of the matK study

After getting demux.qzv, I use Qiime2 viewer (https://view.qiime2.org) to analysis this file, and manually figure out the thresholds, in forward reads at point 160 (Figure 5), and in reverse reads at point 120 (Figure 6) where the quality drops down significantly.
In Figure 7, we can see that the total number of gene sequences included in this study is 41025444, and we can see 140 representative sequences. In Figure 8. Demultiplexed sequence length summary in matK study.

`

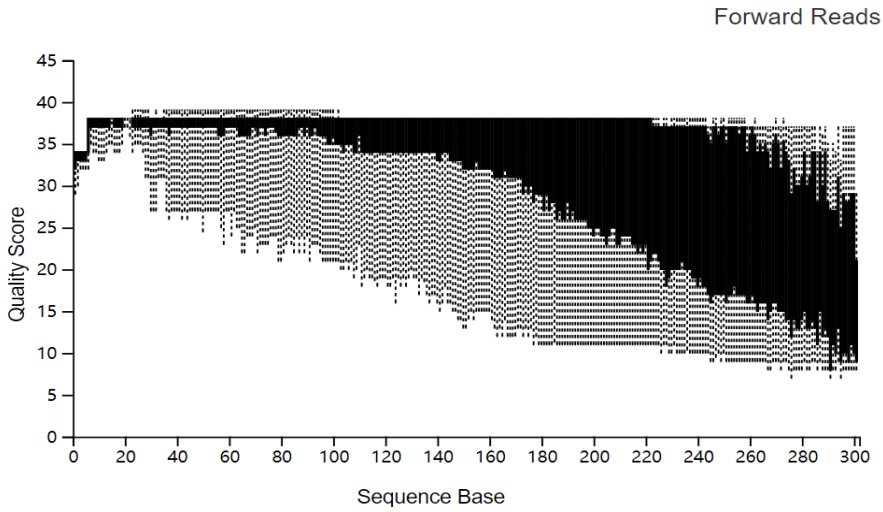


Figure 5: Demultiplexed sequence of Forward Reads

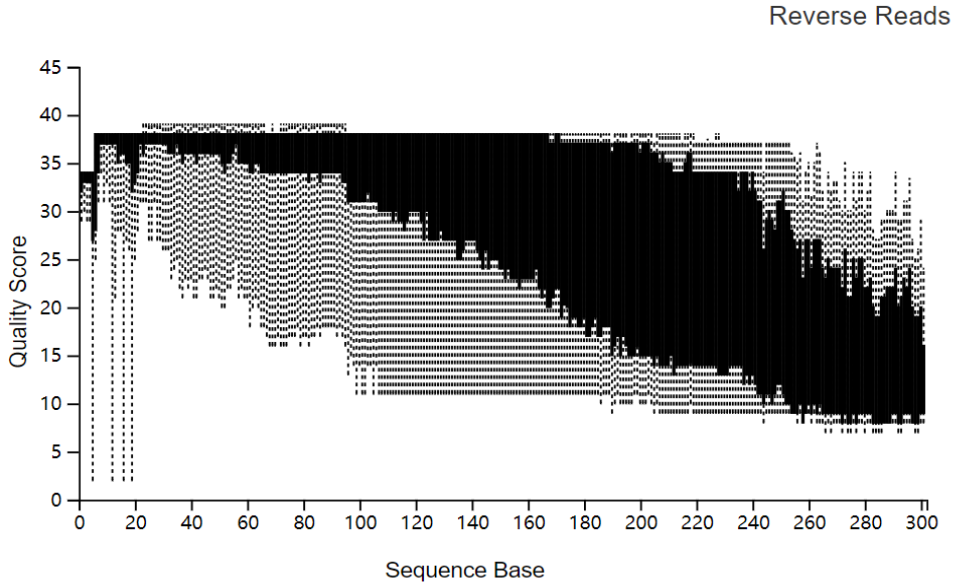


Figure 6: Demultiplexed sequence of Reverse Reads

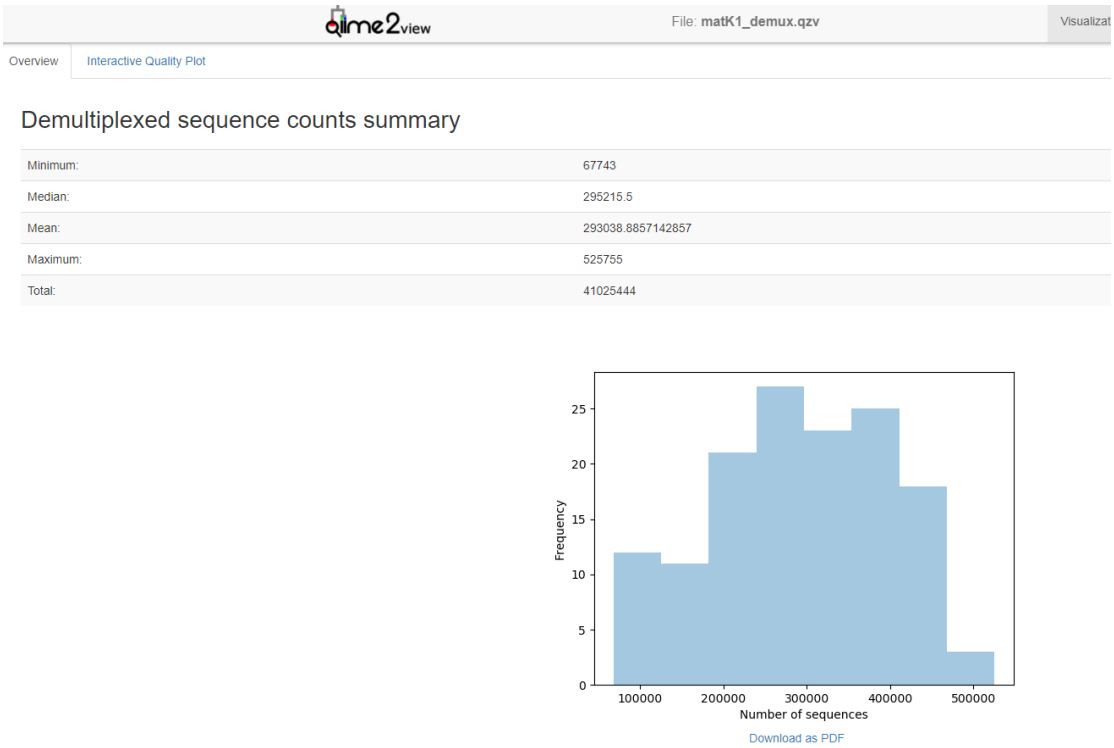


Figure 7: Demultiplexed sequence counts summary

Demultiplexed sequence length summary

Forward Reads

Total Sequences Sampled	10000
2%	45 nts
9%	49 nts
25%	93 nts
50% (Median)	162 nts
75%	301 nts
91%	301 nts
98%	301 nts

Reverse Reads

Total Sequences Sampled	10000
2%	45 nts
9%	49 nts
25%	93 nts
50% (Median)	162 nts
75%	301 nts
91%	301 nts
98%	301 nts

Figure 8: Demultiplexed sequence length summary in matK study

I use DADA2 algorithm which will produce table.qza as an abundance table and rep-seqs.qza will contain the ASV sequences. When I get rep-seqs.qza file, its visualization data as followings: qiime feature-table tabulate-seqs --i-data rep-seqs.qza --o-visualization rep-seqs.qzv

`

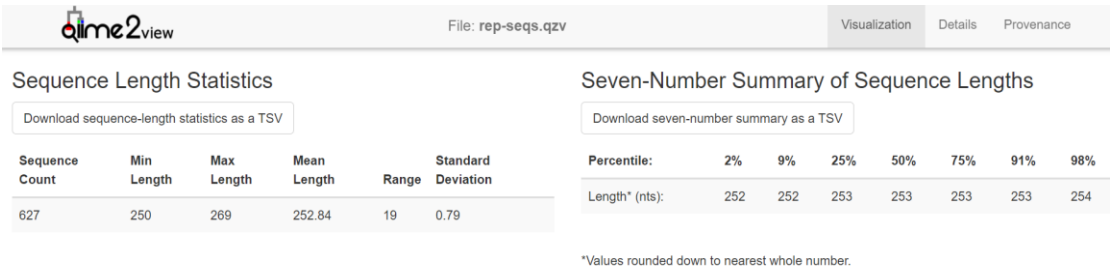


Figure 9: rep-seqs.qzv and Sequence Length Statistics

Statistic	Value
count	627
min	250
max	269
mean	252.839
range	19
std	0.790044

Table 5. descriptive_stats.tsv

After the denoising is completed, I get the representative sequence file rep-seqs.qza. Figures 9 show the visual analysis of the rep-seqs. Figure 10 shows statistical results of the denoising process of rep-seqs in matK study. Figure 11 captures some of the statistical results of the denoising process.

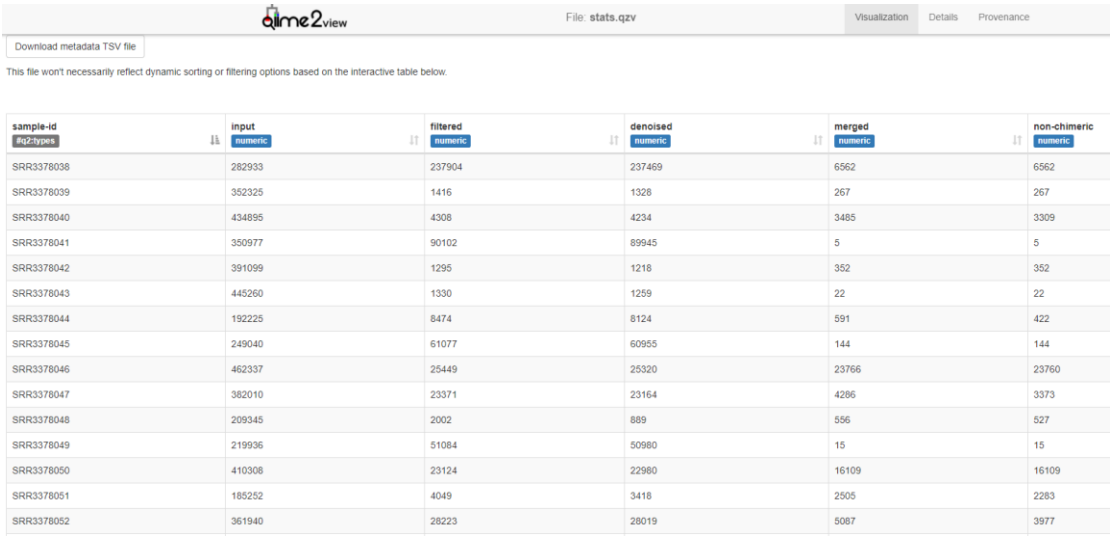


Figure 10: statistical results of the denoising process of rep-seqs in matK study

I create a phylogenetic tree(Figure 12), export rooted-tree.qza as NewickDirectoryFormat to directory output, using online tool http://etetoolkit.org/treeview/ to visualise newick tree file, tree.nwk

`

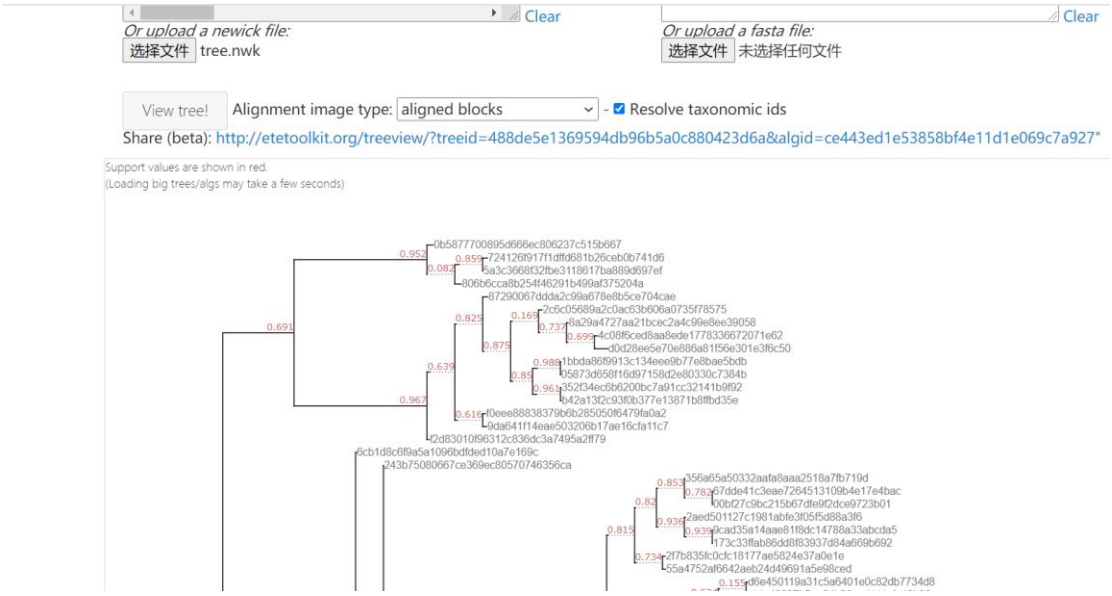


Figure 12: Partial viewtree in matK study

3.2.2 matK taxonomy result

In one of my matK study paper, *Large-Scale Monitoring of Plants through Environmental DNA Metabarcoding of Soil:Recovery, Resolution, and Annotation of Four DNA Markers*, I get the rep-seqs.qza and then classify rep-seqs.qza to get the taxonomy file named taxonomy.qza. Through visualizing taxonomy.qza to taxonomy.qzv(Figure 13), bar-plot of taxonomy(Figure 15) can be generated. Figure 14 shows provenance of taxa-bar-plots. After classifying matk study, I can get the summary statistics (Figure 16) of the database at different taxonomic ranks, thus to find type number of each level. (Table 6)

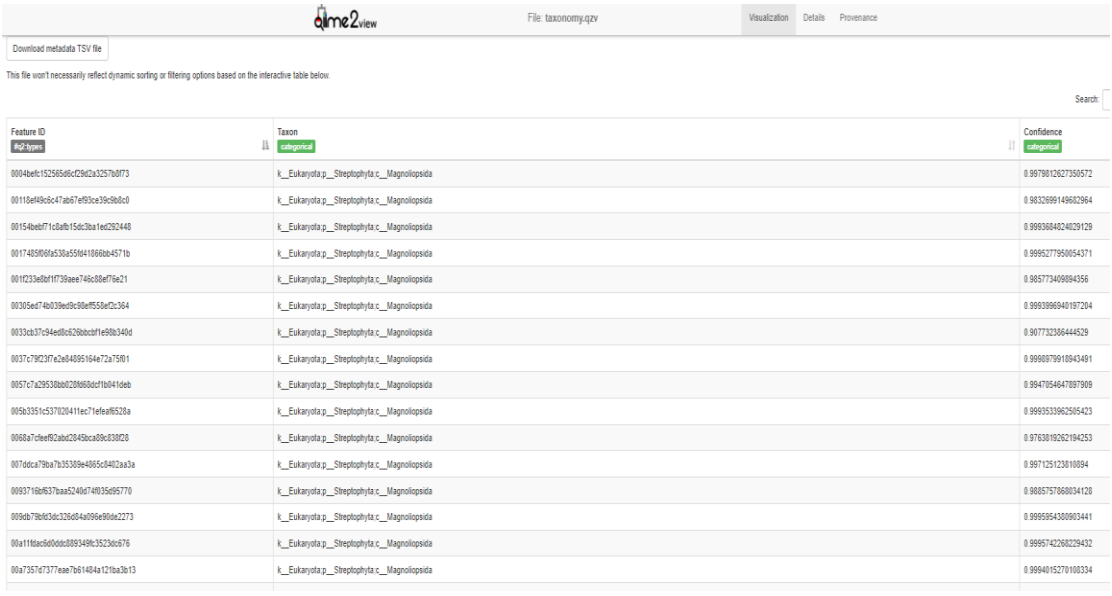


Figure 13: taxonomy result

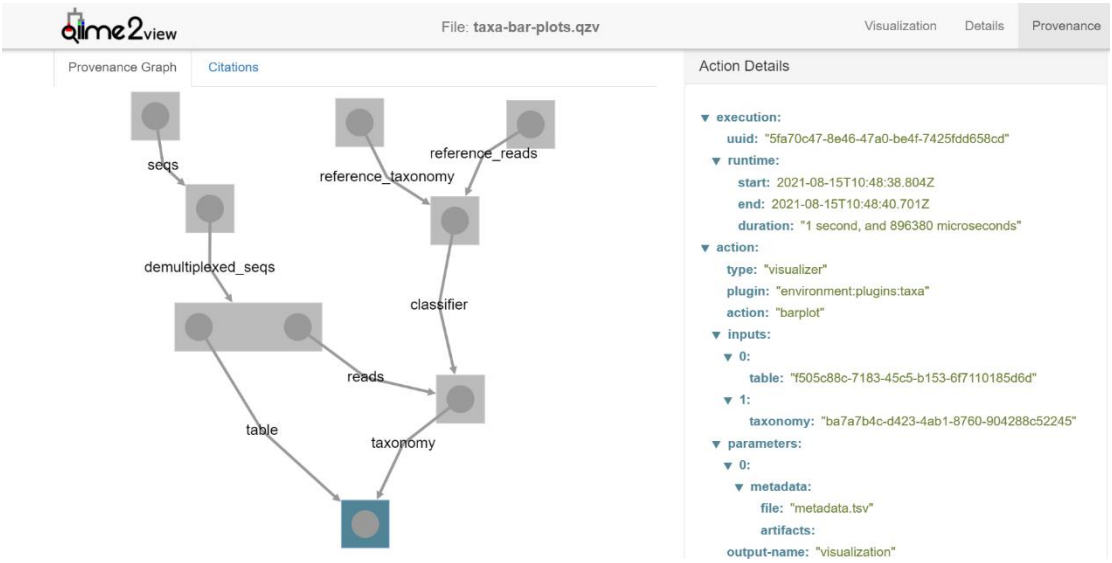


Figure 14: Provenance of taxa-bar-plots

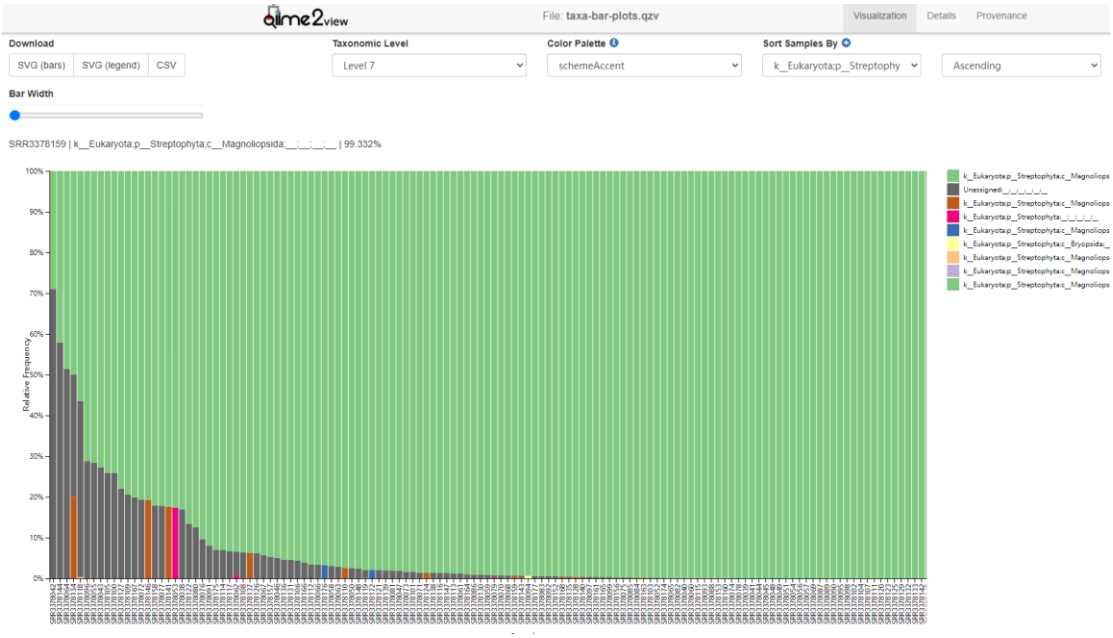


Figure 15: taxa-bar-plots

```
[studentprojects@howe /shared5/studentprojects/YOU/new_database/new_matK]$ tr ";" "\t" < metadata.tsv | awk '{
print $2}' | sort | uniq -c
      1 categorical
      1 ID
   4680 k__Eukaryota
    187 Unassigned
```

Figure 16: summary of metadata.tsv

Level	Type amounts	reads

kingdom	1	4680
phylum	1	4680
class	2	4616
order	3	8
family	4	5
genus	2	2
species	2	2
Unassigned	1	187
Total	16	4867

Table 6. summary statistics of matk metadata.tsv

3.2.3 Qiime2 workflow ITS2 study

In order to get meta-sequence of ITS2, I use one of my searching studies which is in the result of literature research shown in Table 1. I choose one of the ITS2 studies, which is *New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones* and data of this study is available with BioProject number, PRJNA393998. Using qiime2 workflow shown in Appendix A, I get gene data about the study, SRR number is 1, but demultiplexed sequence amounts is 12,592,989 (Table 7)

Item	ITS2
PRJN	PRJNA393998
SRR numbers	1
Demultiplexed sequence amounts	12,592,989

Table 7: Statistics of the matK study

After getting demux.qzv, I use Qiime2 viewer (https://view.qiime2.org) to analysis this file, and manually figure out the thresholds, in forward reads at point 210 (Figure 17), and in reverse reads at point 180 (Figure 18) where the quality drops down significantly.
In Figure 19, we can see that the total number of gene sequences included in this study is 12,592,989, and we can see 1 representative sequences.

I set thresholds as followings:

qiime dada2 denoise-paired --i-demultiplexed-seqs demux.qza --p-trim-left-f 0 --p-trim-left-r 0 --p-trunc-len-f 210 --p-trunc-len-r 180 --p-n-threads 0 --o-table table.qza --o-representative-sequences rep-seqs.qza --o-denoising-stats denoising-stats.qza --verbose

`

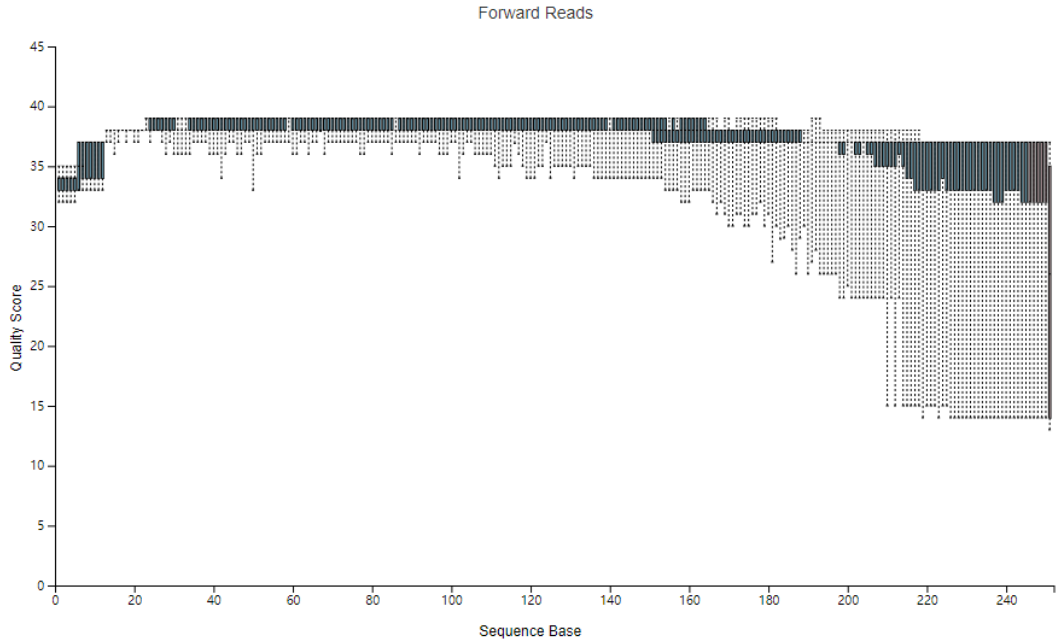


Figure 17: Demultiplexed sequence of Forward Reads

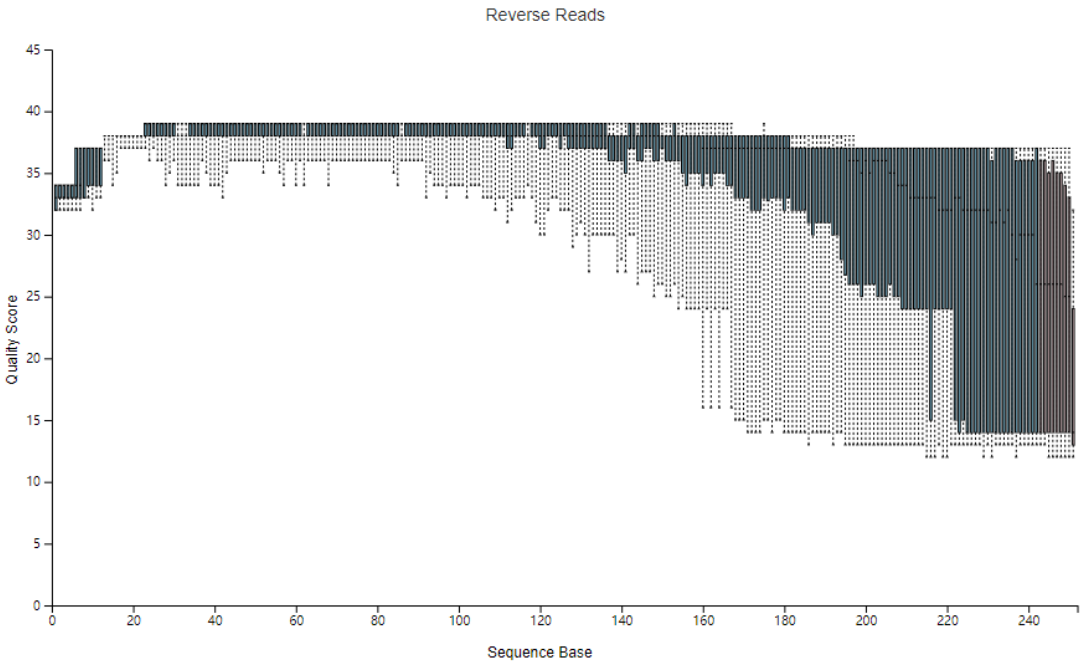


Figure 18: Demultiplexed sequence of Reverse Reads

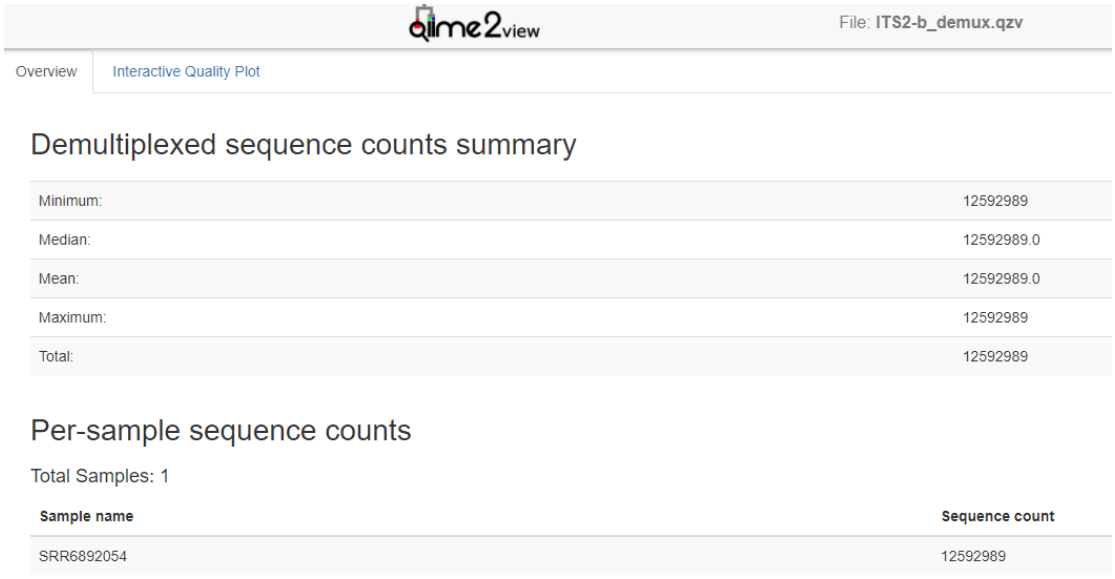
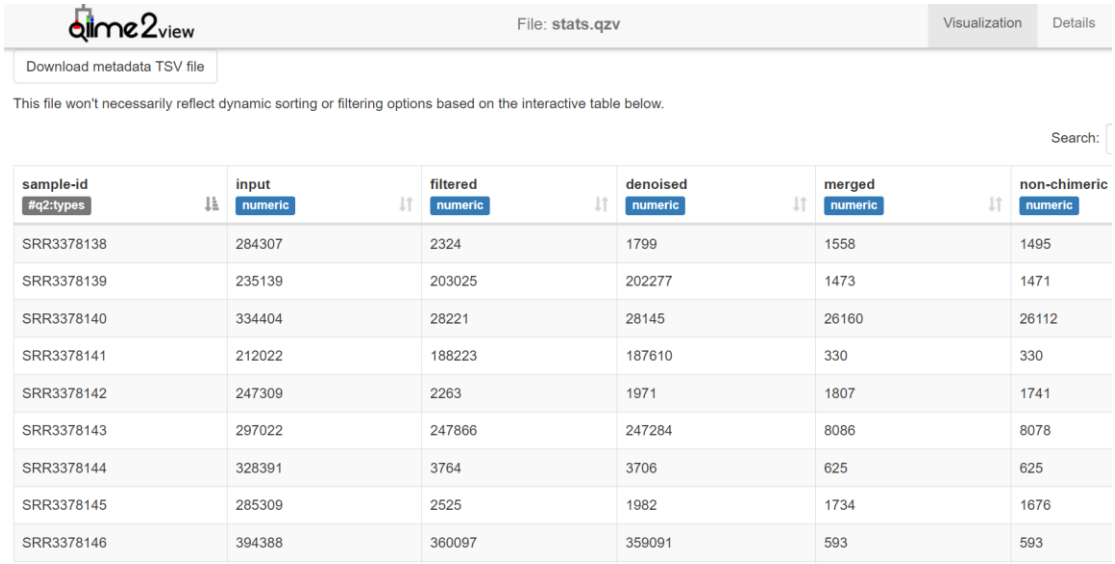


Figure 19: Demultiplexed sequence counts summary

I run DADA2 algorithm which will produce table.qza as an abundance table and rep-seqs.qza will contain the ASV sequences (Figure 20).



sample-id #q2:types	input numeric	filtered numeric	denoised numeric	merged numeric	non-chimeric numeric
SRR3378138	284307	2324	1799	1558	1495
SRR3378139	235139	203025	202277	1473	1471
SRR3378140	334404	28221	28145	26160	26112
SRR3378141	212022	188223	187610	330	330
SRR3378142	247309	2263	1971	1807	1741
SRR3378143	297022	247866	247284	8086	8078
SRR3378144	328391	3764	3706	625	625
SRR3378145	285309	2525	1982	1734	1676
SRR3378146	394388	360097	359091	593	593

Figure 20: data in stats.qzv

3.2.4 ITS2 taxonomy result

In one of my ITS2 study paper, *New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones*, I get the rep-seqs.qza and then classify rep-seqs.qza to get the taxonomy file named taxonomy.qza.

`

Through visualizing taxonomy.qza to taxonomy.qzv, bar-plot of taxonomy can be generated. After classifying ITS2 study, I can get the summary statistics of the database at different taxonomic ranks, thus to find type number of each level.

The classification of the representative sequences, using the classifier trained from our database gives a taxonomy file which is shown as Figure 21, the command is as following: *qiime feature-classifier classify-sklearn --i-classifier db_accession_classifier.qza --i-reads /shared5/studentprojects/YOU/ITS2-study/qiime2/rep-seqs.qza --o-classification taxonomy.qza.* In Figure 22, it shows provenance of taxa-bar-plots.

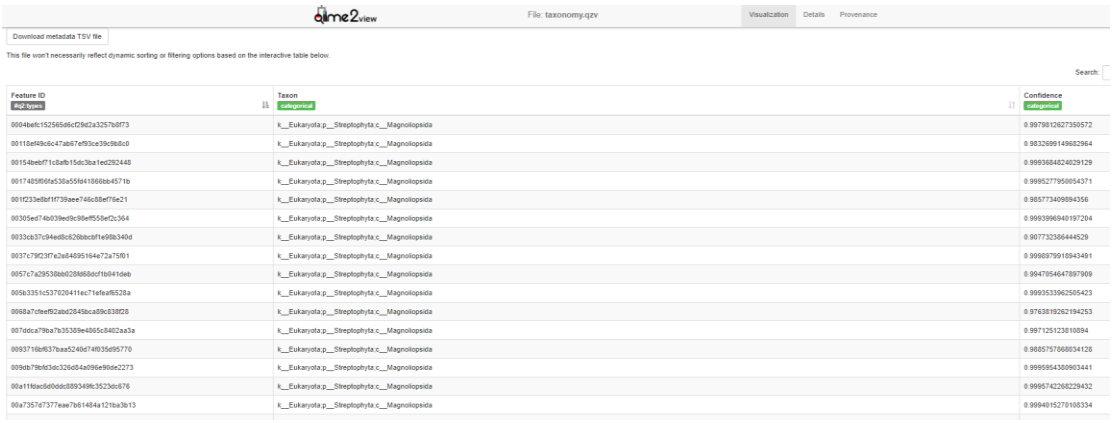
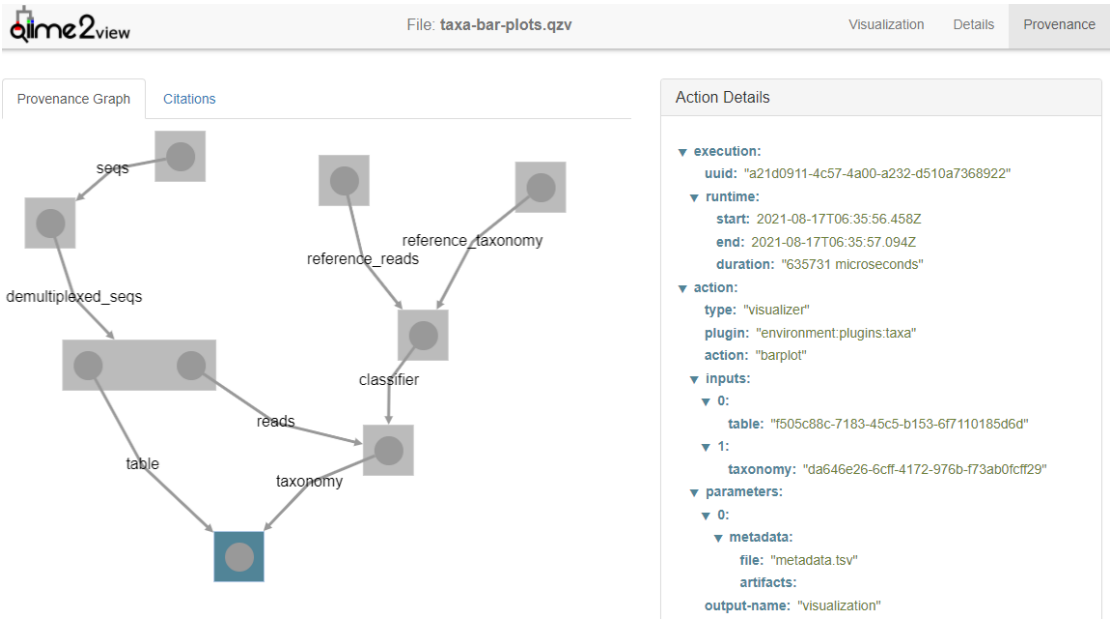


Figure 21: taxonomy of rep-seqs in ITS2 study

Taxonomy barplot (Figure 23) using command as following:

qiime taxa barplot \
--i-table /shared5/studentprojects/YOU/ITS2-study/qiime2/table.qza \
--i-taxonomy taxonomy.qza \
--m-metadata-file /shared5/studentprojects/YOU/ ITS2-study/qiime2/sample_metadata.tsv \
--o-visualization taxa-bar-plots.qzv



`

Figure 22: Provenance of taxa-bar-plots

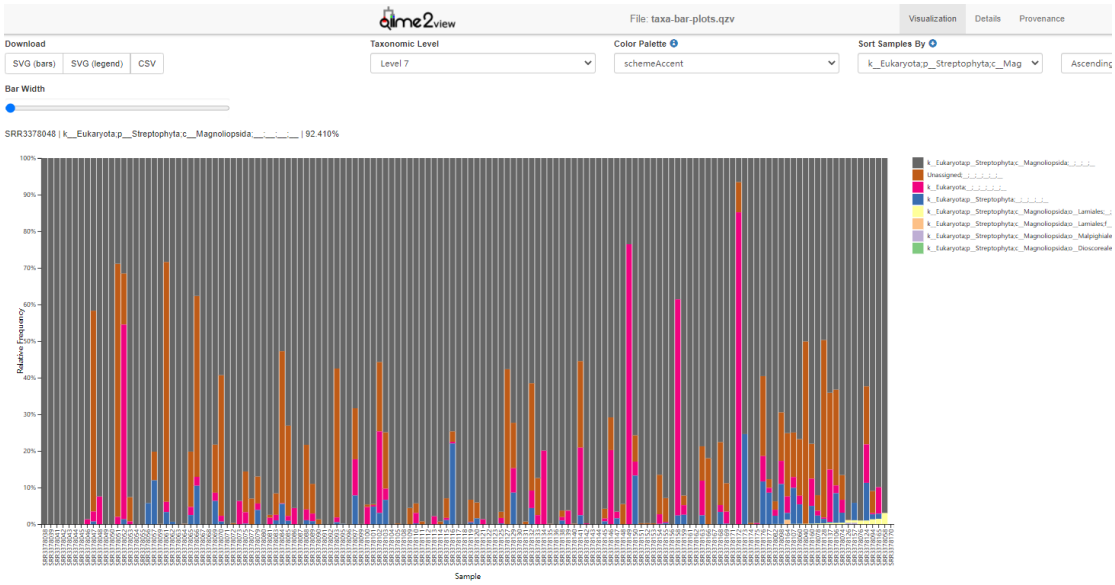


Figure 23: taxa-bar-plots

4 Discussion

4.1 Data download time consuming

The Entrez Programming Utilities (E-utilities) are a set of nine server-side programs that provide a stable interface into the Entrez query and database system at the National Center for Biotechnology Information (NCBI). The E-utilities use a fixed URL syntax that translates a standard set of input parameters into the values necessary for various NCBI software components to search for and retrieve the requested data. The E-utilities are therefore the structured interface to the Entrez system, which currently includes 38 databases covering a variety of biomedical data, including nucleotide and protein sequences, gene records, three-dimensional molecular structures, and the biomedical literature.
(https://www.ncbi.nlm.nih.gov/books/)

To access these data, a piece of software first posts an E-utility URL to NCBI, then retrieves the results of this posting, after which it processes the data as required. The software can thus use any computer language that can send a URL to the E-utilities server

`

and interpret the XML response; examples of such languages are Perl, Python, Java, and C++. Combining E-utilities components to form customized data pipelines within these applications is a powerful approach to data manipulation.
(https://www.ncbi.nlm.nih.gov/books/)

When it comes to the creation of the database by using R library rentrez, I find that each reads takes about 2 seconds, matK study reads totaled 232300, it will take estimated 110 hours to have work done. For ITS2 study, it will take 245 hours, about 10 days. In the process of generating the taxonomic data file of matK, this step is very time-consuming. I wonder what is the reason for it and after some research I find the reason in Usage Guidelines and Requirements in NCBI. In order not to overload the E-utility servers, NCBI recommends that users post no more than three URL requests per second.

In order to improve the processing speed, I divided IDs_accession.txt into 10 groups, each group run on 10 different server or PC. Because it takes 2 second to process each data, so I will take 11 hours in all. Taking matK as an example, IDs_accession.txt file can be divided into 10 group and each file contains 20320 lines.

4.2 matK study

There are 4867 ASV sequences, and 4565 of them were assigned, about 96.16%, with all classifications having a confidence of over 75 percent. Using new trained matK classifier to classify the sequences in the PRJNA318025 project, it is found that the level kingdom mainly contains Eukaryota, level phylum mainly contains Streptophyta plant, and level class mainly contains Andreaeopsida and Bryopsida plant. Table 8 show taxonomy statistics of the matK study of PRJNA318025. Among them, there is 3.84% unassigned, and 96.16% assigned. In Figure 24, bar-plot shows taxonomy of matk study

Level	amount of type	reads
kingdom	1	4680
phylum	1	4680
class	2	4616
order	3	8
family	4	5
genus	2	2
species	2	2
Unassigned	1	187
Total	16	4867

Table 8. taxonomy statistics of the matK study

`

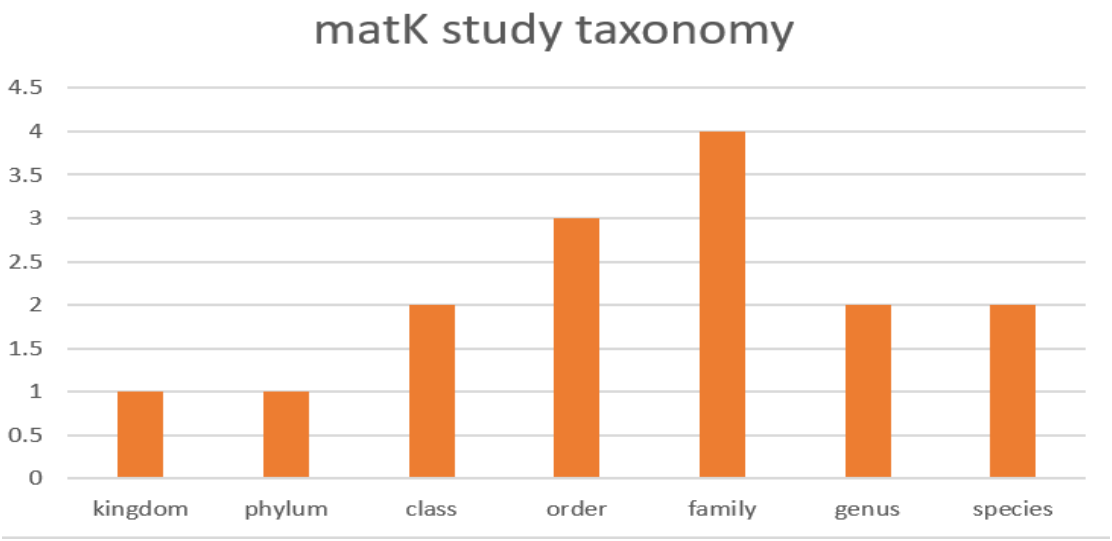


Figure 24: taxonomy of matk study

4.3 ITS2 study

In ITS2 study, using new trained ITS2 classifier to classify the sequences in the PRJNA393998 project, it is found that level kingdom contains 1 type that is Eukaryota, level phylum contains 1 type that is Streptophyta and so on, which shown in Table 9. After doing the classification, it is found that the amount of type are 12, and about 80.51% of ASV sequences were assigned and about 19.49% unassigned. In Figure 25, bar-plot shows taxonomy of ITS2 study

Level	amount of type	Plant species name
kingdom	1	Eukaryota
phylum	1	Streptophyta
class	1	Magnoliopsida
order	3	Dioscoreales Lamiales Malpighiales

`

family	3	Burmanniaceae Salicaceae Scrophulariaceae
genus	2	Bennettiodendron Burmannia
species	0	
Unassigned	1	
Total	12	

Table 9. taxonomy statistics of the ITS2 study

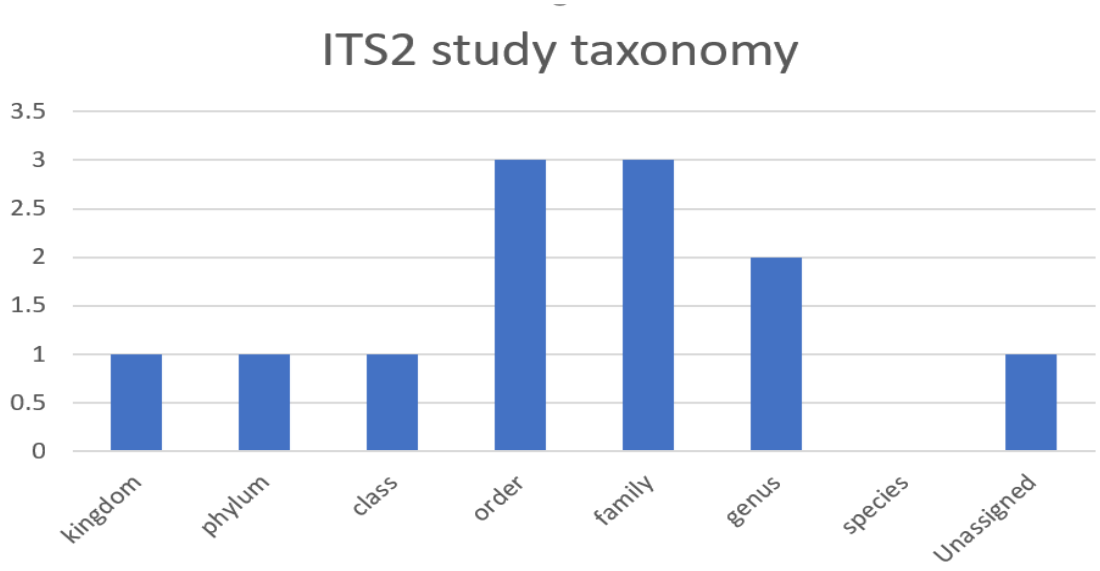


Figure 25: taxonomy of ITS2 study

5 Conclusions

Based on the Qiime2 workflow, this project use data download from NCBI to train classifier and use this classifier to classify meta-sequence of matK and ITS2 related studies, and then generate taxonomy file of these studies. According to the taxonomy file, it can help us to figure out how many unique plant taxonomic groups in particularly area. Furthermore, the new database can also help us to Identify plant species, analyze diet of ungulates, do plant diversity surveys, and Identify abundance values of biological samples.

`

6 References

1. Fazekas A, Kuzmina ML, Newmaster SG et al (2012) DNA barcoding methods for land plants
2. Natasha de Vere et al, DNA Barcoding for Plants, Plant Genotyping, pp 101-118
3 Kress WJ, Erickson DL (eds). Springer protocols methods in molecular biology 858 DNA barcodes methods and protocols. Springer, New York, pp 223–252
4. Lynch M, Milligan BG (1994) Analysis of poplulation genetic-structure with RAPD markers. Mol Ecol 3:91–99
5. Vere N, Rich TCG, Ford CR et al (2012) DNA barcoding the native flowering plants and conifers of Wales. PLoS One 7:e37945
6. Fazekas AJ, Burgess KS, Kesanakurti PR et al (2008) Multiple multilocus DNA barcodes from the plastid genome discriminate plant species equally well. PLoS One 3:e2802
7. Davey JW, Hohenlohe PA, Etter PD, Boone JQ, Catchen JM, Blaxter ML (2011) Genome-wide genetic marker discovery and genotyping using next-generation sequencing. Nat Rev Genet 12:499–510
8. Webb CO (2000) Exploring the phylogenetic structure of ecological communities: an example for rain forest trees. Am Nat 156:145–155
9. R. Scott Cornman et al. (2021) Taxonomic Characterization of Honey Bee (Apis mellifera) Pollen Foraging Based on Non-Overlapping Paired-End Sequencing of Nuclear Ribosomal Loci
10. Harvey PH, Leigh Brown AJ, Maynard SJ, NeeS (2006) New uses for new phylogenies. Oxford University Press, Oxford
11. Živa Fišer Pečnikar et al. (2013) years since the introduction of DNA barcoding: from theory to application. Epub 2013 Nov 8.
12. Nicole A. Fahner et al. (2016) Large-Scale Monitoring of Plants through Environmental DNA Metabarcoding of Soil:Recovery, Resolution, and Annotation of Four DNA Markers
13. Rosemary J. et al. (2018) New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones
14. Jinxin Liu et al. (2021) The Species Identification in Traditional Herbal Patent Medicine, Wuhu San, Based on Shotgun Metabarcoding
15. Ciara Keating, et al. (2020) Circular Economy of Anaerobic Biofilm Microbiomes: A Meta-Analysis Framework for Re-exploration of Amplicon Sequencing Data
16. Stephanie A. Coghlan et al. (2021) Development of an environmental DNA metabarcoding assay for aquatic vascular plant communities
17. Kingsly C. Beng et al. (2019) Amplicon sequencing dataset of soil fungi and associated environmental variables collected in karst and non-karst sites across Yunnan province, southwest China
18. Jana Batovska et al. (2017) Using Next-Generation Sequencing for DNA Barcoding: Capturing Allelic Variation in ITS2
19. Rosemary J. et al. (2018) New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones
20. R. Scott Cornman et al. (2015) Taxonomic Characterization of Honey Bee (Apis mellifera)

`

Pollen Foraging Based on Non-Overlapping Paired-End Sequencing of Nuclear Ribosomal Loci
21. Grace Moore et al. (2020) Paleo-metagenomics of North Ameican fossil packrat middens: Past biodiversity revealed by ancient DNA

`

Appendix：

Appendix A, Qiime2 Workflow

1 Prepare data

Read article and find studies with accession number PRJN number， and download the paired-end reads.

esearch database of SRA by PRJN number
gunzip file which type is fastq.gz to fastq

2 workflow

Step 1: We are going to organize our data in such a manner that for every sample we have the folder name extracted from the paired-end files, and we are going to dump the raw sequences in a “Raw” folder:

Step 2: Create a qiime2 folder

Step 3: Create sample_metadata.tsv

Step 4: Generate barcodes for each read

Step 5: Collate all the forward reads from all the folders together in a single forward.fastq file

Step 6: Collate all the reverse reads from all the folders together in a single reverse.fastq file

Step 7: Zip all the FASTQ files and move them to emp-paired-end-sequences folder
Enable Qiime2 environment Qiime2

Step 8: Import the sequences to qiime2

Step 9: Demultiplex the sequences in Qiime2， Generate file， demux.qza and demux-details.qza

Step 10: Depends on the quality, fine tune Dada2 algorithm by specifying the thresholds
Export demux.qza to demux.qzv for visualization drag and drop the file demux.qzv on the Qiime2 viewer https://view.qiime2.org and manually Figure out the thresholds, where the quality drops down significantly

`

Step 11: Run DADA2 algorithm which will produce table.qza as an abundance table and rep-seqs.qza will contain the ASV sequences

Step 12: Create a phylogenetic tree.

`

Appendix B. My Workflow logs

1 Download data
Find a study of matK with accession number, PRJNA318025, so that we are able to download the paired-end reads.

The study:
Large-Scale Monitoring of Plants through Environmental DNA Metabarcoding of Soil:Recovery, Resolution, and Annotation of Four DNA Markers.

1.1 setup environment parameters

[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ pwd
/shared5/studentprojects/YOU/matk1
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ export PATH=/home/opt/sratoolkit.2.9.0-centos_linux64/bin:$PATH
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ export PATH=/home/opt/edirect:$PATH

1.2 download data of PRJNA318025
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ esearch -db sra -query PRJNA318025 | efetch --format runinfo |cut -d "," -f 1 > SRR.numbers
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ awk '/SRR|ERR/' SRR.numbers > SRR.numbers.filtered
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ for i in $(cat SRR.numbers.filtered); do echo Processing $i; fastq-dump --split-files --origfmt --gzip $i ; done

.....
Processing SRR3378058
Read 80667 spots for SRR3378058
Written 80667 spots for SRR3378058
Processing SRR3378059
Read 328589 spots for SRR3378059
Written 328589 spots for SRR3378059
Processing SRR3378060
Read 243486 spots for SRR3378060
Written 243486 spots for SRR3378060

1.3 Move all the sequences in a subdirectory called "sequences"
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ mkdir sequences
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ mv *.fastq.gz sequences/.
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ ls

`

sequences SRR.numbers SRR.numbers.filtered

[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ cd sequences
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ gunzip *
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ ls
SRR3378038_1.fastq SRR3378054_1.fastq SRR3378070_1.fastq SRR3378086_1.fastq
SRR3378102_1.fastq SRR3378118_1.fastq SRR3378134_1.fastq SRR3378150_1.fastq
SRR3378166_1.fastq
SRR3378038_2.fastq SRR3378054_2.fastq SRR3378070_2.fastq SRR3378086_2.fastq
SRR3378102_2.fastq SRR3378118_2.fastq SRR3378134_2.fastq SRR3378150_2.fastq
SRR3378166_2.fastq

2 Qiime2 workflow

Step 1:
We are going to organize our data in such a manner that for every sample we have the folder name extracted from the paired-end files, and we are going to dump the raw sequences in a "Raw" folder

[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ for i in $(awk -F"_" '{print $1}' <(ls *.fastq) | sort | uniq); do mkdir $i; mkdir $i/Raw; mv $i*.fastq $i/Raw/.; done

[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ ls | wc -l
140

[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ ls */Raw/*
SRR3378038/Raw/SRR3378038_1.fastq SRR3378066/Raw/SRR3378066_1.fastq
SRR3378094/Raw/SRR3378094_1.fastq SRR3378122/Raw/SRR3378122_1.fastq
SRR3378150/Raw/SRR3378150_1.fastq
SRR3378038/Raw/SRR3378038_2.fastq SRR3378066/Raw/SRR3378066_2.fastq
SRR3378094/Raw/SRR3378094_2.fastq SRR3378122/Raw/SRR3378122_2.fastq
SRR3378150/Raw/SRR3378150_2.fastq
SRR3378039/Raw/SRR3378039_1.fastq SRR3378067/Raw/SRR3378067_1.fastq
SRR3378095/Raw/SRR3378095_1.fastq SRR3378123/Raw/SRR3378123_1.fastq
SRR3378151/Raw/SRR3378151_1.fastq
SRR3378039/Raw/SRR3378039_2.fastq SRR3378067/Raw/SRR3378067_2.fastq
SRR3378095/Raw/SRR3378095_2.fastq SRR3378123/Raw/SRR3378123_2.fastq
SRR3378151/Raw/SRR3378151_2.fastq

Step 2:
Create a qiime2 workflow folder named qiime2
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ cd ..
[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ mkdir qiime2

`

[studentprojects@becker /shared5/studentprojects/YOU/matk1]$ cd qiime2
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
total 0

Step 3:
Get the path of sequences folder assigned to a variable d

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ cd ../sequences
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ pwd
/shared5/studentprojects/YOU/matk1/sequences

[studentprojects@becker
/shared5/studentprojects/YOU/matk1/sequences]$ d="/shared5/studentprojects/YOU/matk
1/sequences";
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ t=$(ls $d | wc -
l);
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ echo $d
/shared5/studentprojects/YOU/matk1/sequences
[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ echo $t
140

[studentprojects@becker /shared5/studentprojects/YOU/matk1/sequences]$ cd ../qiime2/
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ paste <(ls $d)
<(perl -le 'sub p{my $l=pop @_;unless(@_){return map [$_],@$l;}return map { my $ll=$_; map
[@$ll,$_],@$l} p(@_);} @a=[A,C,G,T]; print join("", @$_) for p(@a,@a,@a,@a,@a,@a,@a,@a);'
| awk -v k=$t 'NR<=k{print}') | awk 'BEGIN{print "sample-id\tbarcode-
sequence\n#q2:types\tcategorical"}1' > sample_metadata.tsv

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
sample_metadata.tsv

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ cat
sample_metadata.tsv
sample-id barcode-sequence
#q2:types categorical
SRR3378038 AAAAAAAA
SRR3378039 AAAAAAAC
SRR3378040 AAAAAAAG
SRR3378041 AAAAAAAT
SRR3378042 AAAAAACA

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ wc -l
sample_metadata.tsv

`

142 sample_metadata.tsv

Step 4:
Generate barcodes for each read using the file as above

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ (for i in $(ls $d); do bc=$(awk -v k=$i '$1==k{print $2}' sample_metadata.tsv); bioawk -cfastx -v k=$bc '{print "@"$1" "$4"\n"k"\n+";for(i=0;i< length(k);i++){printf "#"};printf "\n"}' $d/$i/Raw/*_1.fastq ; done) > barcodes.fastq

Step 5:
Collate all the forward reads from all the folders together in a single forward.fastq file

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ (for i in $(ls $d); do cat $d/$i/Raw/*_1.fastq ; done) > forward.fastq

Step 6:
Collate all the reverse reads from all the folders together in a single reverse.fastq file

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ (for i in $(ls $d); do cat $d/$i/Raw/*_2.fastq ; done) > reverse.fastq
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
barcodes.fastq forward.fastq reverse.fastq sample_metadata.tsv

Sanity check: see if all the numbers match
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ bioawk -cfastx 'END{print NR}' forward.fastq
41025444
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ bioawk -cfastx 'END{print NR}' reverse.fastq
41025444
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ bioawk -cfastx 'END{print NR}' barcodes.fastq
41025444

Step 7:
Zip all the FASTQ files and move them to emp-paired-end-sequences folder

[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ gzip *.fastq
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
barcodes.fastq.gz forward.fastq.gz reverse.fastq.gz sample_metadata.tsv
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ mkdir emp-paired-end-sequences; mv *.gz emp-paired-end-sequences/.
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls

`

emp-paired-end-sequences sample_metadata.tsv

Next, Enable Qiime2 on the Orion cluster
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ export PATH=/home/opt/miniconda2/bin:$PATH
[studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ source activate qiime2-2019.7

Step 8:
Import the sequences with Qiime2 tools

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime tools import --type EMPPairedEndSequences --input-path emp-paired-end-sequences --output-path emp-paired-end-sequences.qza
Imported emp-paired-end-sequences as EMPPairedEndDirFmt to emp-paired-end-sequences.qza
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
emp-paired-end-sequences emp-paired-end-sequences.qza sample_metadata.tsv

Step 9:
Demultiplex the sequences in Qiime2

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime demux emp-paired --p-no-golay-error-correction --i-seqs emp-paired-end-sequences.qza --m-barcodes-file sample_metadata.tsv --m-barcodes-column barcode-sequence --o-per-sample-sequences demux.qza --o-error-correction-details demux-details.qza
Saved SampleData[PairedEndSequencesWithQuality] to: demux.qza
Saved ErrorCorrectionDetails to: demux-details.qza
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ls
demux-details.qza demux.qza emp-paired-end-sequences emp-paired-end-sequences.qza sample_metadata.tsv

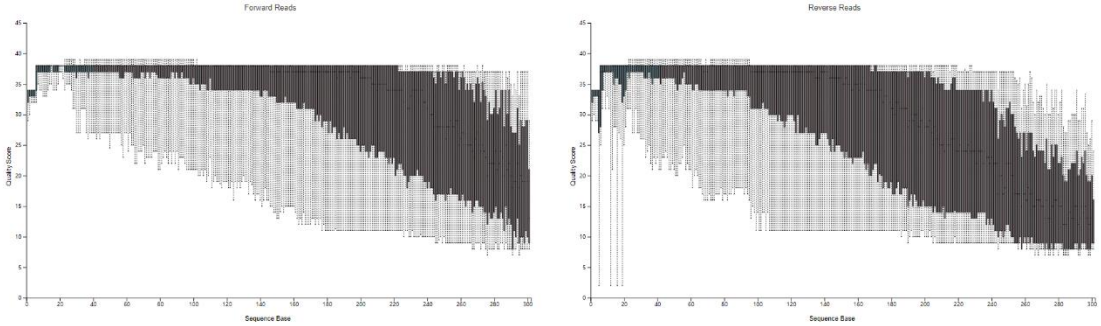
Step 10:
Depends on the quality of our run, we want to fine tune Dada2 algorithm by specifying the thresholds

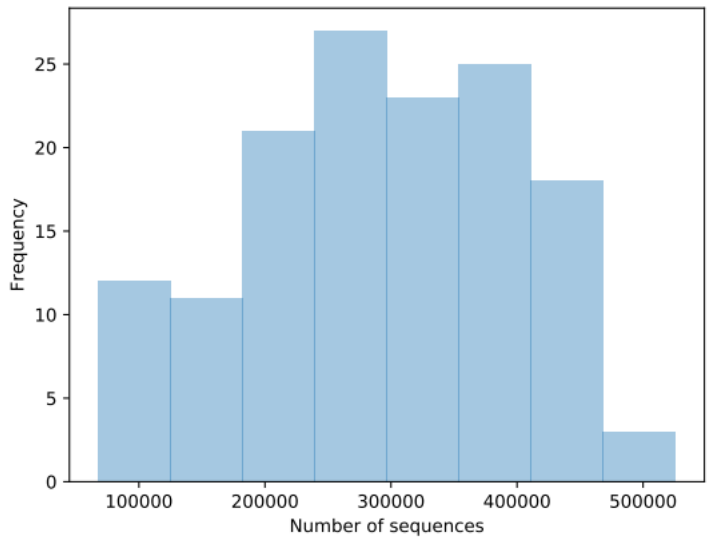
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime demux summarize --i-data ./demux.qza --o-visualization ./demux.qzv
Saved Visualization to: ./demux.qzv
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ ll

`

total 17308993
-rw-rw-r--. 1 studentprojects studentprojects 210870401 Jul 28 08:25 demux-details.qza
-rw-rw-r--. 1 studentprojects studentprojects 8589957796 Jul 28 08:25 demux.qza
-rw-rw-r--. 1 studentprojects studentprojects   307084 Jul 28 09:15 demux.qzv
drwxrwxr-x. 2 studentprojects studentprojects   5 Jul 27 13:23 emp-paired-end-sequences
-rw-rw-r--. 1 studentprojects studentprojects 8914822740 Jul 27 13:38 emp-paired-end-sequences.qza
-rw-rw-r--. 1 studentprojects studentprojects   2849 Jul 26 03:24 sample_metadata.tsv
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ cp demux.qzv matK1_demux.qzv

Next drag and drop the file which name is matK1_demux.qzv on the Qiime2 viewer https://view.qiime2.org and manually Figure out the thresholds, i.e., where the quality drops down significantly





Step 11:
Run DADA2 algorithm which will produce table.qza as an abundance table and rep-seqs.qza will contain the ASV sequences

`

In Forward Reads, the quality scores start to diminish somewhere in the middle of 160, chose 160
In Reverse Reads, the quality scores start to diminish somewhere in the middle of 120, chose 120

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime dada2 denoise-paired --i-demultiplexed-seqs demux.qza --p-trim-left-f 0 --p-trim-left-r 0 --p-trunc-len-f 160 --p-trunc-len-r 120 --p-n-threads 0 --o-table table.qza --o-representative-sequences rep-seqs.qza --o-denoising-stats denoising-stats.qza --verbose
Running external command line application(s). This may print messages to stdout and/or stderr.
The command(s) being run are below. These commands cannot be manually re-run as they will depend on temporary files that no longer exist.

Command: run_dada_paired.R /tmp/tmpw2je1m6p/forward /tmp/tmpw2je1m6p/reverse /tmp/tmpw2je1m6p/output.tsv.biom /tmp/tmpw2je1m6p/track.tsv /tmp/tmpw2je1m6p/filt_f /tmp/tmpw2je1m6p/filt_r 160 120 0 0 2.0 2.0 2 consensus 1.0 0 1000000

R version 3.5.1 (2018-07-02)
Loading required package: Rcpp
DADA2: 1.10.0 / Rcpp: 1.0.2 / RcppParallel: 4.4.3
1) Filtering .....
2) Learning Error Rates
194275840 total bases in 1214224 reads from 25 samples will be used for learning the error rates.
145706880 total bases in 1214224 reads from 25 samples will be used for learning the error rates.
3)Denoise remaining samples .....
4) Remove chimeras (method = consensus)
5) Write output
Saved FeatureTable[Frequency] to: table.qza
Saved FeatureData[Sequence] to: rep-seqs.qza
Saved SampleData[DADA2Stats] to: denoising-stats.qza
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$

Step 12:
Create a phylogenetic tree
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ unset MAFFT_BINARIES
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime phylogeny align-to-tree-mafft-fasttree --i-sequences rep-seqs.qza --o-alignment aligned-rep-seqs.qza --o-masked-

`

alignment masked-aligned-rep-seqs.qza --p-n-threads 0 --o-tree unrooted-tree.qza --o-rooted-tree rooted-tree.qza

Saved FeatureData[AlignedSequence] to: aligned-rep-seqs.qza
Saved FeatureData[AlignedSequence] to: masked-aligned-rep-seqs.qza
Saved Phylogeny[Unrooted] to: unrooted-tree.qza
Saved Phylogeny[Rooted] to: rooted-tree.qza
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$

3 Export data that is produced by qiime2 in qza/qzv format

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime tools export --input-path table.qza --output-path output
Exported table.qza as BIOMV210DirFmt to directory output

The table is exported as BIOM file (https://biom-format.org/)

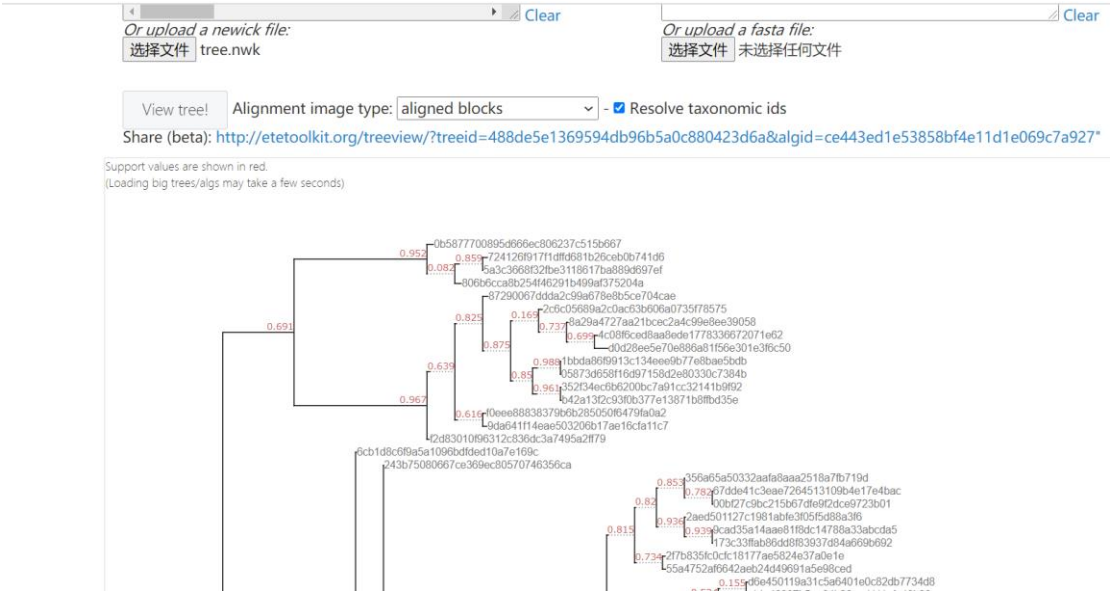
Produce dna-sequences.fasta in the output folder

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime tools export --input-path rep-seqs.qza --output-path output
Exported rep-seqs.qza as DNASequencesDirectoryFormat to directory output

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/matk1/qiime2]$ qiime tools export --input-path rooted-tree.qza --output-path output
Exported rooted-tree.qza as NewickDirectoryFormat to directory output

Visualise newick tree files using an online tool http://etetoolkit.org/treeview/

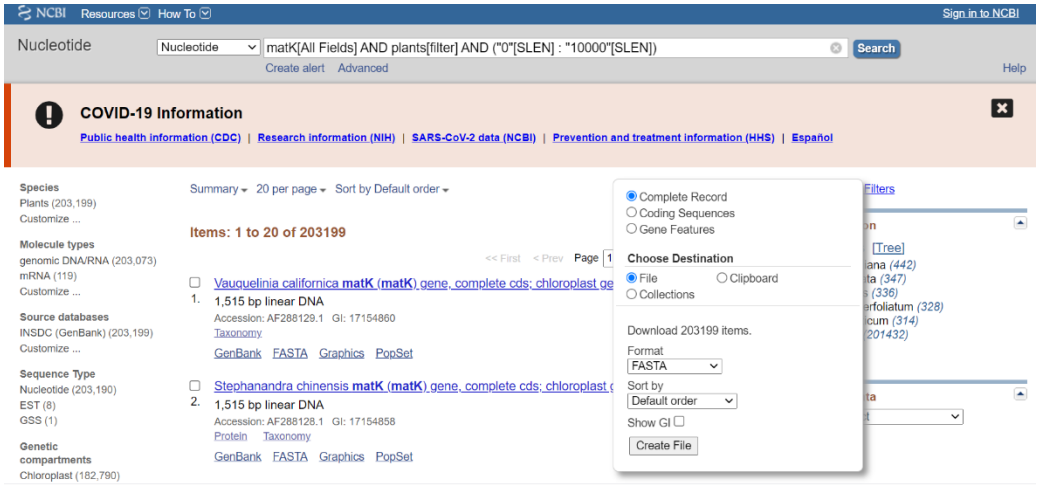
`



4 Training classifier logs

4.1 Download all the sequences by searching for matk region, save that as a fasta file (sequence.fasta)

In NCBI website, searching for matk region using:
matK[All Fields] AND plants[filter] AND ("0"[SLEN] : "10000"[SLEN])
Download file which is named sequence.fasta



(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ cp sequence.fasta db.fasta

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ ll db.fasta -h
-rw-rw-r--. 1 studentprojects studentprojects 228M Jul 6 15:33 db.fasta

`

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ head db.fasta
>AF288129.1 Vauquelinia californica matK (matK) gene, complete cds; chloroplast gene for chloroplast product
ATGGAAGAATTTCAAGGATATTTAGAACTAGATAGATATCAGCAACATGACTTCCTATACCCACT
TATCT
TTCGGGAGTATATTTATGCACTTGCTCATGATCATGGTTTAAATAGATCGATTTTGTTGGATAATG
TAGG
TTATGACACTAAATATAGTTTACTAATTATAAAACGTTTAATTAGTCGAATGTATCAACAGAATCA
TTTG
ATTATTTCCGCTAATGATTCTAACCAAAATAAATTTTTTGGGTACAACAAAAATTTGTATTCTCAA
ATGA
TGTCGGAGGGATTTGCAGTCATTGTGGAAATTCCGTTTTCCCTACGATTAGTATCTTCCTTAGAG
GCGAC
AGAAATCGTAAAATCTTATAATTTACGATCAATTCATTCAATATTTCCTTTTTTAGAGGACAAATT
CCCA
CATTTAAATTATGTATCAGATGTACTAATACCCTACCCCATTCATCTGGAAATCTTGGTTCAAACC
CTTC
GCTATTGGGTGAAAGATCCCTCTTCTTTACATTTATTACGACTCCTTCTTCACGAGTATTATAATT
GGAA
TAGTCTTATTACTACAAAAAAAGTGATTTTTTCAAAAAGTAATCCACGATTATTCTTGCTCCTATA
TAAT
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ bioawk -cfastx '{print ">"$1"\n"$2}' db.fasta > db_accession.fasta

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ head -n 4 db_accession.fasta
>AF288129.1
ATGGAAGAATTTCAAGGATATTTAGAACTAGATAGATATCAGCAACATGACTTCCTATACCCACT
TATCTTTCGGGAGTATATTTATGCACTTGCTCATGATCATGGTTTAAATAGATCGATTTTGTTGGA
TAATGTAGGTTATGACACTAAATATAGTTTACTAATTATAAAACGTTTAATTAGTCGAATGTATCA
ACAGAATCATTTGATTATTTCCGCTAATGATTCTAACCAAAATAAATTTTTTGGGTACAACAAAAA
TTTGTATTCTCAAATGATGTCGGAGGGATTTGCAGTCATTGTGGAAATTCCGTTTTCCCTACGATT
AGTATCTTCCTTAGAGGCGACAGAAATCGTAAAATCTTATAATTTACGATCAATTCATTCAATATT
TCCTTTTTTAGAGGACAAATTCCCACATTTAAATTATGTATCAGATGTACTAATACCCTACCCCAT
TCATCTGGAAATCTTGGTTCAAACCCTTCGCTATTGGGTGAAAGATCCCTCTTCTTTACATTTATT
ACGACTCCTTCTTCACGAGTATTATAATTGGAATAGTCTTATTACTACAAAAAAAGTGATTTTTTC
AAAAAGTAATCCACGATTATTCTTGCTCCTATATAATTCTCATGTATGTGAATACGAATCCATTTT
ACTTTTTCTTCGTAATCAATCTTCTCATTTACGATTAACCTCTTCGGGTATCTTTTTTGAGCGAATA
CATTTCTATGAAAAAAAAAAATATCCTGTAGAAGAAGTCTTCGTTAATGATTTTCCGGCCGCCAT
CTTATGGTTCTTCAAGGATCCTTTTATGCATTATGTTAGATATCAAGGAAAATCAATTCTGTCTTC

`

GAAGGATACCCCTCTTCTGATGAATAAGTGGAAATATTATCTTGTCAATTTATGGCAATGTCATT
CTTATGTGTGGTCTCAACCAGGAAGGATTTATATAAACCAATTATCCAAGCATTCCCTTGATTTTT
TGGGTTATTTTTCAAGTATGCGACCAAACCTTTCGGTGGTACGGAGTCAAATGCTAGAAAATTCA
TTTCTAATGGATAATGCTATGAAGAAGCTTGATACATTAGTTCCAATTATTCCTTTGATTGGATCA
TTGGCTAAAGTGAAATTTTGTAACGCATTAGGGCATCCTATTAGTAAGTCCACCTGGGCAGATTC
GTCGGATTTTGATATTATCGACCGATTTGTGCATATATGCAGAAATCTTTCTCATTATTACAGTGG
ATCCTCAAGAAAAAAGAGTTTGTATCGAATAAAATATATACTTCGACTTTCTTGTGTTAAAACTTT
GGCTCGTAAACACAAAAGTACTGTACGAACTTTTTTGAAAAGATTAGGTTATAAATTATTGGACG
AATTCTTTACGGAAGAAGAACAGATTCTTTCTTTAATCTTCCCAAGAGCTTCTTATACTTTGAAGA
AGTTTTATAGAGGTCGAATTTGGTATTTGGATATTTTTTGCATCAATGATCTAGTCAATCATGAAT
A

4.2 Remove everything from the name except accession IDs.
First extract accession IDs and store it in a txt file

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ bioawk -cfastx '{print $1}' db_accession.fasta > IDs_accession.txt

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ head IDs_accession.txt
AF288129.1
AF288128.1
AF288127.1
AF288126.1
AF288125.1
AF288124.1
AF288123.1
AF288122.1
AF288121.1

4.3 Enable R-environment and Run R

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ export PATH=/home/opt/miniconda2/bin:$PATH
[studentprojects@becker /shared5/studentprojects/YOU/0706]$ source activate r-environment
(r-environment)[studentprojects@becker /shared5/studentprojects/YOU/0706]$

Run R
(r-environment)[studentprojects@becker /shared5/studentprojects/YOU/0706]$ R

in R, run the following commands:

library(rentrez)
#Load the mapping table up

`

```
mapping_table<-read.csv("IDs_accession.txt",header=FALSE)
#extract gids
gids<-as.character(mapping_table$V1)
taxa_levels<-NULL
for(i in seq(1:length(gids))){
  print(paste("Processing",i,"/",length(gids)))
  tmp<-
tryCatch(paste(XML::xpathSApply(entrez_fetch(db="taxonomy",id=entrez_summary(db="nu
cleotide", id=gids[i])$taxid,rettype="xml", parsed=TRUE),
"//LineageEx/Taxon/ScientificName", XML::xmlValue),collapse=";"),error=function(e) "")
  tmp2<-
tryCatch(paste(XML::xpathSApply(entrez_fetch(db="taxonomy",id=entrez_summary(db="nu
cleotide", id=gids[i])$taxid,rettype="xml", parsed=TRUE), "//LineageEx/Taxon/Rank",
XML::xmlValue),collapse=";"),error=function(e) "")

  #From the XML returned extract the taxonomy
  tmp1_df<-strsplit(tmp,";")[[1]]
  #From the XML returned extract the levels
  tmp2_df<-strsplit(tmp2,";")[[1]]

  #Now assemble the whole taxonomy
  tmp<-paste(paste("k__",tmp1_df[tmp2_df=="superkingdom"],sep=""),";",
  paste("p__",tmp1_df[tmp2_df=="phylum"],sep=""),";",
  paste("c__",tmp1_df[tmp2_df=="class"],sep=""),";",
  paste("o__",tmp1_df[tmp2_df=="order"],sep=""),";",
  paste("f__",tmp1_df[tmp2_df=="family"],sep=""),";",
  paste("g__",tmp1_df[tmp2_df=="genus"],sep=""),";",
  paste("s__",tmp1_df[tmp2_df=="species"],sep=""),";",sep="")


  if(is.null(taxa_levels)){taxa_levels<-tmp}else{taxa_levels<-c(taxa_levels,tmp)}
}
data_to_write<-data.frame(ID=mapping_table[,1],Taxa=taxa_levels)
write.table(data_to_write,"db_accession.tax",sep="\t",row.names=F,col.names=F,quote=F)
quit()
```

4.4 Deactivate r-environment and import the sequences in qiime2 format

```
(r-environment) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ source
deactivate
DeprecationWarning: 'source deactivate' is deprecated. Use 'conda deactivate'.


[studentprojects@becker /shared5/studentprojects/YOU/0706]$ head -n 4 db_accession.tax
AF288129.1                                                                  cellular
```

`

organisms;Eukaryota;Viridiplantae;Streptophyta;Streptophytina;Embryophyta;Tracheophyta;Euphyllophyta;Spermatophyta;Magnoliopsida;Mesangiospermae;eudicotyledons;Gunneridae;Pentapetalae;rosids;fabids;Rosales;Rosaceae;Amygdaloideae;Maleae;Vauquelinia
AF288128.1 cellular organisms;Eukaryota;Viridiplantae;Streptophyta;Streptophytina;Embryophyta;Tracheophyta;Euphyllophyta;Spermatophyta;Magnoliopsida;Mesangiospermae;eudicotyledons;Gunneridae;Pentapetalae;rosids;fabids;Rosales;Rosaceae;Amygdaloideae;Neillieae;Neillia
AF288127.1 cellular organisms;Eukaryota;Viridiplantae;Streptophyta;Streptophytina;Embryophyta;Tracheophyta;Euphyllophyta;Spermatophyta;Magnoliopsida;Mesangiospermae;eudicotyledons;Gunneridae;Pentapetalae;rosids;fabids;Rosales;Rosaceae;Amygdaloideae;Spiraeeae;Spiraea
AF288126.1 cellular organisms;Eukaryota;Viridiplantae;Streptophyta;Streptophytina;Embryophyta;Tracheophyta;Euphyllophyta;Spermatophyta;Magnoliopsida;Mesangiospermae;eudicotyledons;Gunneridae;Pentapetalae;rosids;fabids;Rosales;Rosaceae;Amygdaloideae;Maleae;Sorbus

[studentprojects@becker /shared5/studentprojects/YOU/0706]$ export PATH=/home/opt/miniconda2/bin:$PATH
[studentprojects@becker /shared5/studentprojects/YOU/0706]$ source activate qiime2-2019.7

4.5 Import data to Qiime2 qza format

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ qiime tools import --type 'FeatureData[Sequence]' --input-path db_accession.fasta --output-path db_accession.qza
Imported db_accession.fasta as DNASequencesDirectoryFormat to db_accession.qza

(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ qiime tools import --type 'FeatureData[Taxonomy]' --input-format HeaderlessTSVTaxonomyFormat --input-path db_accession.tax --output-path db_accession-taxonomy.qza

4.6 Train classifier
(qiime2-2019.7) [studentprojects@becker /shared5/studentprojects/YOU/0706]$ qiime feature-classifier fit-classifier-naive-bayes --i-reference-reads db_accession.qza --i-reference-taxonomy db_accession-taxonomy.qza --o-classifier db_accession_classifier.qza
Saved TaxonomicClassifier to: db_accession_classifier.qza

4.7 Using classifier
qiime feature-classifier classify-sklearn --i-classifier db_accession_classifier.qza --i-reads /shared5/studentprojects/YOU/matk1-study/qiime2/rep-seqs.qza --o-classification taxonomy.qza

4.8 visualization

`

```
qiime metadata tabulate --m-input-file taxonomy.qza --o-visualization taxonomy.qzv

4.9 produce taxa barplot
qiime taxa barplot \
  --i-table /shared5/studentprojects/YOU/matk1-study/qiime2/table.qza \
  --i-taxonomy taxonomy.qza \
  --m-metadata-file /shared5/studentprojects/YOU/matk1-study/qiime2/sample_metadata.tsv \
  --o-visualization taxa-bar-plots.qzv
```

Appendix C. R language workflow

```
library(rentrez)
#Load the mapping table up
mapping_table<-read.csv("IDs_accession.txt",header=FALSE)
#extract gids
gids<-as.character(mapping_table$V1)
taxa_levels<-NULL
for(i in seq(1:length(gids))){
  print(paste("Processing",i,"/",length(gids)))
  tmp<-
tryCatch(paste(XML::xpathSApply(entrez_fetch(db="taxonomy",id=entrez_summary(db="nucleotide", id=gids[i])$taxid,rettype="xml", parsed=TRUE), "//LineageEx/Taxon/ScientificName", XML::xmlValue),collapse=";"),error=function(e) "")
  tmp2<-
tryCatch(paste(XML::xpathSApply(entrez_fetch(db="taxonomy",id=entrez_summary(db="nucleotide", id=gids[i])$taxid,rettype="xml", parsed=TRUE), "//LineageEx/Taxon/Rank", XML::xmlValue),collapse=";"),error=function(e) "")

  #From the XML returned extract the taxonomy
  tmp1_df<-strsplit(tmp,";")[[1]]
  #From the XML returned extract the levels
  tmp2_df<-strsplit(tmp2,";")[[1]]

  #Now assemble the whole taxonomy
  tmp<-paste(paste("k__",tmp1_df[tmp2_df=="superkingdom"],sep=""),";",
  paste("p__",tmp1_df[tmp2_df=="phylum"],sep=""),";",
  paste("c__",tmp1_df[tmp2_df=="class"],sep=""),";",
  paste("o__",tmp1_df[tmp2_df=="order"],sep=""),";",
  paste("f__",tmp1_df[tmp2_df=="family"],sep=""),";",
  paste("g__",tmp1_df[tmp2_df=="genus"],sep=""),";",
  paste("s__",tmp1_df[tmp2_df=="species"],sep=""),";",sep="")
```

`

```
  if(is.null(taxa_levels)){taxa_levels<-tmp}else{taxa_levels<-c(taxa_levels,tmp)}
}
data_to_write<-data.frame(ID=mapping_table[,1],Taxa=taxa_levels)
write.table(data_to_write,"db_accession.tax",sep="\t",row.names=F,col.names=F,quote=F)
quit()
```

Appendix D. LIST OF ABBREVIATIONS

Abbreviation	Explanation
DNA	Deoxyribonucleic acid
FASTA	Fast-all
NCBI	National Centre for Biotechnology Information
SSH	Secure shell
UA	Unassigned
ITS	Internal Transcribed Spacer

`