1. Reads assembled and depth filtered (Linux system):

The script “1gene\_assemble\_and\_depth\_filter.py” was used, and you need to prepare a list of your sample ID, see the example “1seqFile.txt”.

Run command:

python3 1gene\_assemble\_and\_depth\_filter.py (name of the python script)

Note: you need to install Trinity software and Python3 software in advance in Linux system.

In this step we will get the document “Trinity\_dep\_filtered.fas” and “\*.depth“” document.

1. Gene name annotation

Blastn is used in this step, enquiry data blast against local reference database (you need to install BLAST software in advance in Linux system).

Run command:

makeblastdb -in local\_database.fas -dbtype nucl -title intermediate\_generated\_file -parse\_seqids -out intermediate\_generated\_file -logfile AI

blastn -query Trinity\_depth\_filter.fas -out seq.blast -db intermediate\_generated\_file -outfmt 6 -evalue 1e-5 -max\_target\_seqs 3

Note: local\_database.fas, reference data; Trinity\_depth\_filter.fas, enquiry data; seq.blast, out put file.

After this we are going to get seq.blast file, and then run python script “2addGene.py” base on “seq.blast” and “Trinity\_dep\_filtered.fas” file in Windows system, we will get gene annotation file “Trinity\_dep\_filtered\_addGene.fas”.

1. Five gene sorted by the python script “3splitGene.py” base on “Trinity\_dep\_filtered\_addGene.fas” file.

In this step we will get the \*.fas file of five gene.

1. Length filter

Cut off 20bp in the both end of the gene, and sequence length within the expected range (the lengths of *COI*, *ITS* and *rbcL* are 500bp–900bp, *matK* is 800bp–1100bp, *trnL* is 400bp–800bp) were retained for later analysis. “4piltLen.py python” script was used.

Sometimes two fragments of two barcodes can be spliced together, causing splicing errors. These can be found easily by original \*.depth text (generated by SAMTOOLS). To solve this problem, one should check the depth of the single nucleobase one by one, if the depth was suddenly reduced in the middle of the sequence, it was generally recognized as a splicing error, and these errors could be rectified manually, usually the two fragments were two barcodes.

1. Sequences that <80% cover and <98% identification were filtered out of four plant barcodes base on NCBI blast. In this step we need to prepare the files generated in the previous step and the “\*.XML” files after blast on NCBI. “5blastSta.py” were used in this step. Finally, we will get gene\_depth\_blast.fas files.
2. Species identification:

Each gene (*COI*, *ITS*, *rbcL*, *matK* and *trnL*) were blast again with local reference database (BLAST software in Linux system), the reverse sequence was corrected to forward sequence using “6reverse\_complement.py” python script base on \*.blast files and \*.fas files. Finally, we get all the forward sequence.

1. All the blast result of five gene (*COI*, *ITS*, *rbcL*, *matK* and *trnL*) were merged by “7merge.py python” script, in this step ID list of the samples “7seqFile.txt” and the blast result of each barcode were needed.

Note: Species that do not exist in local reference database were blast on BOLD Systems (<http://www.boldsystems.org/>) (forward sequence were needed) or NCBI (https://www.ncbi.nlm.nih.gov/).