
Supplementary Script S2: Molecular analysis of the fungal diversity in *Pinus nigra* subsp *laricio* needles

Supplementary Script S2: Molecular analysis of the fungal diversity in *Pinus nigra* subsp *laricio* needles

- Software used

- Join paired ends

- Quality and primer artefact filtering

 - Remove primer artefacts

 - Special cases: remove sequences with primer fw and primer rev joined.

 - Reorient all sequences in 5'-3'

 - Discard sequences with bad primer in 3'-5' included in the sequences

 - Discard low quality sequences using sickle

 - Quality check

- Demultiplexing

 - Pool files and remove primers and labels

- Dereplication using code from swarm github readme pages

 - Linearization to simplify processes

 - Dereplication

- Detecting chimera and non fungal ITS sequences

 - Chimera detection using the Unite database as reference

 - Disard non fungal ITS sequences using ITSx

- Clustering sequences into OTUs

 - Cluster with Swarm (d=1)

 - Closed reference

 - Open reference

 - Uparse

 - Uparse with default parameter

Assign taxonomy

Assign taxonomy to the representative sequences of each OTU for

Uparse

Assign taxonomy to the representative sequences of each closed-references OTUs

Assign taxonomy to the representative sequences of each open-references OTUs

Assign taxonomy to the representative sequences of each swarm

Create OTU table

Map reads (including singletons) back to OTUs

Uparse

Closed Reference

Open Reference

Swarm

Convert Uparse result into a Biom table.

Convert Closed result into a Biom table

Convert Open result into a Biom table

Convert Swarm result into a Biom table

Krona Visualization

Compare taxonomic distribution of number of sequences and number of OTUs using Uparse

Compare taxonomic distribution of number of sequences and number of OTUs using swarm

Compared different taxonomy assignment methods

RDP

blast

mothur

uclust

Final figure

Compared different clustering methods (usearch/uclust/swarm)

uparse

swarm

Final figure

Assignment of ecological guild using FUN guild

Software used

- [Qiiime](#)
- [fqgrep](#)
- [sickle](#)
- [fastx_toolkit](#)
- [Krona](#)
- [swarm](#)
- [Usearch and Uclust](#)
- [RDP](#)
- [ITSx](#)
- [biom format](#)

Join paired ends

```
qiime
join_paired_ends.py -f ../all_R1.fastq.gz -r ../all_R2.fastq.gz -
o ../join_paired

fastx_trimmer -f 1 -l 281 -i ../join_paired/fastqjoin.join.fastq
-o ../join_paired_quality.fastq

fastx_trimmer -m 180 -t 1 -i ../join_paired_quality.fastq -o
../join_paired_quality2.fastq
```

Quality and primer artefact filtering

Remove primer artefacts

```
python ../remove_multiprimer.py -i ../join_paired_quality2.fastq
-o ../j_p_qual_removeMultiPrim1.fastq -f TTGAACGCACATTGCGC -r
CTCGGATCAGGTAGGGAT
```

Special cases: remove sequences with primer fw and primer rev joined.

```
fqgrep -p 'TTGAACGCACATTGCGC*CTCGGATCAGGTAGGGAT'  
.../j_p_qual_removeMultiPrim1.fastq -e -v >  
.../j_p_qual_removeMultiPrim.fastq
```

Reorient all sequences in 5'-3'

```
fqgrep -p "TTGAACGCACATTGCGC" -e  
.../j_p_qual_removeMultiPrim.fastq > .../j_p_qual_rMP_5-  
3only.fastq  
  
fqgrep -p "GCGCAATGTGCGTTCAA" -e  
.../j_p_qual_removeMultiPrim.fastq > .../j_p_qual_rMP_3-5.fastq  
  
fastx_reverse_complement -i .../j_p_qual_rMP_3-5.fastq >  
.../j_p_qual_rMP_3-5_reversecomp.fastq  
  
cp .../j_p_qual_rMP_5-3only.fastq .../j_p_qual_rMP_5-3.fastq  
cat .../j_p_qual_rMP_3-5_reversecomp.fastq >> .../j_p_qual_rMP_5-  
3.fastq
```

Discard sequences with bad primer in 3'-5' included in the sequences

```
fqgrep -v -p "GCGCAATGTGCGTTCAA" -e .../j_p_qual_rMP_5-3.fastq >  
.../j_p_qual_rMP_5-3_clean.fastq
```

Discard low quality sequences using sickle

```
sickle se -f .../j_p_qual_rMP_5-3_clean.fastq -t sanger -o  
.../j_p_qual_rMP_5-3_very_clean.fastq -q 33
```

Quality check

```
fastqc ../join_paired_quality2.fastq ../j_p_qual_rMP_5-3_very_clean.fastq -o ../fastqc/
```

Demultiplexing

demultiplex.sh, *rename.pl* are available in supplementary materials of Balint et al., 2014 (“An Illumina metabarcoding pipeline for fungi”)

```
bash ../demultiplex.sh ../bcd_fw_sameL.csv  
../bcd_rev_sameL_Inv.csv ../j_p_qual_rMP_5-3_very_clean.fastq
```

Pool files and remove primers and labels

Renaming each sequence using its sample names

```
perl rename.pl
```

Pooling all files from which forward and reverse tags are congruent

```
cat renamed_*.fastq > ../combined_sampled.fastq
```

Then I trim primers and tags. I choose to cut 31 bp because primer + tags have a length between 29 and 31. So I prefer to cut more bp to avoid a tendency to similarity between sequences of the same site. I also remove all sequences under 120 bp.

```
fastx_trimmer -f 32 -i ../combined_sampled.fastq -o  
../head_trimmed.fastq  
fastx_trimmer -m 120 -t 31 -i ../head_trimmed.fastq -o  
../all_trimmed.fastq
```

Dereplication using code from swarm github readme pages

Linearization to simplify processes

In short, this code transform each couple of line corresponding to one per sequences in one line per sequences.

```
awk 'NR==1 {print ; next} {printf /^>/ ? "\n"$0"\n" : $1} END
{printf "\n"}' ../all_trimmed.fasta >
../all_trimmed_linearized.fasta
```

Dereplication

```
grep -v "^>" ../all_trimmed_linearized.fasta | \
sort -d | uniq -c | \
while read abundance sequence ; do
    hash=$(printf "${sequence}" | shasum)
    hash=${hash:0:40}
    printf ">%s_%d_%s\n" "${hash}" "${abundance}" "${sequence}"
done | sort -t "_" -k2,2nr -k1.2,1d | \
sed -e 's/\_/\n/2' > ../all_trimmed_dereplicated_withchim.fasta
```

Detecting chimera and non fungal ITS sequences

Chimera detection using the Unite database as

reference

Replace “_” by “;size=” for Uparse.

```
cp ../all_trimmed_dereplicated_withchim.fasta
../all_trimmed_dereplicated_withchim_uparse.fasta
sed -i -e "s/_/;size=/g"
../all_trimmed_dereplicated_withchim_uparse.fasta

../usearch8.0.1623_i86linux32 -uchime_ref
../all_trimmed_dereplicated_withchim_uparse.fasta -db
../unite_its2.fa -nonchimeras
../all_trimmed_dereplicated_uparse.fasta -chimeras
../chimeras_Ref.fasta -strand plus
```

Replace “;size=” by “_” after Uparse.

```
cp ../all_trimmed_dereplicated_uparse.fasta
../all_trimmed_dereplicated_to_chim_denovo.fasta
sed -i -e "s/;size=/_/g" ../all_trimmed_dereplicated_uparse.fasta
```

Disard non fungal ITS sequences using ITSx

```
perl ITSx -i ../all_trimmed_dereplicated_uparse.fasta -o
../ITSx/ITSx -t all --anchor 40 -N 1 --reset T --table T --
detailed_results T --save_raw T --cpu 4 --complement F

pullseq -i ../ITSx/ITSx.ITS2.fasta -g 'F|ITS2' -l 250 | cut -d'|' -
f1 > ../ITS_fungi.fasta

sed -i -e "s/_/;size=/g" ../ITS_fungi.fasta
```

Clustering sequences into OTUs

Cluster with Swarm (d=1)

```
cp ../ITS_fungi.fasta ../ITS_fungi_swarm.fasta
sed -i -e "s/;size=/_/g" ../ITS_fungi_swarm.fasta

cd ../swarm/bin
./swarm -f -t 4 ../ITS_fungi_swarm.fasta -w
../OTU_representatives_D1.fasta -o ../OTU_myfile_D1.swarm
```

Closed reference

```
#-----
#params_97.txt
pick_otus:enable_rev_strand_match TRUE      #from tutorial on
fungi illumina
pick_otus:max_accepts 0
pick_otus:max_rejects 0
pick_otus:stepwords 8                      #from tutorial on fungi illumina
pick_otus:word_length 8                    #from tutorial on fungi illumina
#-----

pick_closed_reference_otus.py -i ../ITS_fungi.fasta -o
../qiime_closedref -r
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta -p
../params_97.txt -s -a
```

Open reference

```
pick_open_reference_otus.py -i ../ITS_fungi.fasta -o
../qiime_openref -r
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta -p
../params_97.txt
```

Uparse


```
.../usearch8.0.1623_i86linux32 -sortbysize .../ITS_fungi.fasta -
fastaout .../Uparse_sorted.fasta -minsize 2
```

```
.../usearch8.0.1623_i86linux32 -sortbysize .../ITS_fungi.fasta -
fastaout .../Uparse_sorted_min1.fasta -minsize 1
```

Uparse with default parameter

```
.../usearch8.0.1623_i86linux32 -cluster_otus
.../Uparse_sorted.fasta -uparseout .../UPARSE_results -otus
.../UPARSE.fasta -sizeout -relabel OTU_
```

```
.../usearch8.0.1623_i86linux32 -cluster_otus
.../Uparse_sorted_min1.fasta -uparseout .../UPARSE_results -otus
.../UPARSE.fasta -sizeout -relabel OTU_
```

Assign taxonomy

Assign taxonomy to the representative sequences of each OTU for Uparse

```
export RDP_JAR_PATH=$HOME/app/rdp_classifier_2.2/rdp_classifier-
2.2.jar
```

```
qiime
assign_taxonomy.py -i .../UPARSE_min2.fasta --reference_seqs_fp
.../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
.../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m rdp
--rdp_max_memory 8000 -o .../taxo_rdp/
```

```
assign_taxonomy.py -i .../UPARSE.fasta --reference_seqs_fp
.../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
.../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m rdp
```

```

--rdp_max_memory 8000 -o ../taxo_rdp_min1/

assign_taxonomy.py -i ../UPARSE_min2.fasta --reference_seqs_fp
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt --similarity
0.90 -m uclust -o ../taxo_uclust/

assign_taxonomy.py -i ../UPARSE_min2.fasta --reference_seqs_fp
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt --
blast_e_value 0.001 -m blast -o ../taxo_blast/

assign_taxonomy.py -i ../UPARSE_min2.fasta --reference_seqs_fp
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m
mothur -o ../taxo_mothur/

```

Assign taxonomy to the representative sequences of each closed-references OTUs

```

assign_taxonomy.py -i ../ITS_fungi_rep_set.fasta --
reference_seqs_fp ../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta
--id_to_taxonomy_fp
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m rdp
--rdp_max_memory 8000 -o ../taxo_rdp/

```

Assign taxonomy to the representative sequences of each open-references OTUs

```

assign_taxonomy.py -i ../rep_set.fna --reference_seqs_fp
../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta --
id_to_taxonomy_fp
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m rdp
--rdp_max_memory 8000 -o ../taxo_rdp/

```

Assign taxonomy to the representative sequences of each swarm

```
assign_taxonomy.py -i ../OTU_representatives_D1.fasta --  
reference_seqs_fp ../sh_refs_qiime_ver7_dynamic_s_01.08.2015.fasta  
--id_to_taxonomy_fp  
../sh_taxonomy_qiime_ver7_dynamic_s_01.08.2015.txt -c 0.80 -m rdp  
--rdp_max_memory 8000 -o ...taxo_rdp/
```

Create OTU table

Map reads (including singletons) back to OTUs

Uparse

```
../usearch8.0.1623_i86linux32 -usearch_global  
../all_trimmed_linearized.fasta -db ../UPARSE_min2.fasta -strand  
plus -id 0.97 -uc ../map.uc -top_hit_only  
  
../usearch8.0.1623_i86linux32 -usearch_global  
../all_trimmed_linearized.fasta -db ../UPARSE.fasta -strand plus  
-id 0.97 -uc ../map_min1.uc -top_hit_only
```

Closed Reference

```
../usearch8.0.1623_i86linux32 -usearch_global  
../all_trimmed_linearized.fasta -db ../ITS_fungi_rep_set.fasta -  
strand plus -id 0.97 -uc ../map.uc -top_hit_only
```

Open Reference

```
.../usearch8.0.1623_i86linux32 -usearch_global  
.../all_trimmed_linearized.fasta -db .../rep_set.fna -strand plus -  
id 0.97 -uc .../map.uc -top_hit_only
```

Swarm

```
.../usearch8.0.1623_i86linux32 -usearch_global  
.../all_trimmed_linearized.fasta -db  
.../OTU_representatives_D1.fasta -strand plus -id 0.97 -uc  
.../map.uc -top_hit_only
```

Convert Uparse result into a Biom table.

uc2otutab.py is available on the site of [Uparse](#).

```
python .../uc2otutab.py .../map.uc > .../otu_table.txt  
biom convert --table-type="OTU table" -i .../otu_table.txt -o  
.../otu_table.biom --to-json  
  
python .../uc2otutab.py .../map2.uc > .../otu_table_complete.txt  
biom convert --table-type="OTU table" -i .../otu_table_complete.txt  
-o .../otu_table_complete.biom --to-json  
  
python .../uc2otutab.py .../map_min1.uc > .../otu_table_min1.txt  
biom convert --table-type="OTU table" -i .../otu_table_min1.txt -o  
.../otu_table_min1.biom --to-json  
  
python .../uc2otutab.py .../map2_min1.uc >  
.../otu_table_complete_min1.txt  
biom convert --table-type="OTU table" -i  
.../otu_table_complete_min1.txt -o  
.../otu_table_complete_min1.biom --to-json
```

Convert Closed result into a Biom table

```
python .../uc2otutab.py .../map.uc > .../otu_table.txt  
biom convert --table-type="OTU table" -i .../otu_table.txt -o
```

```
.../otu_table.biom --to-json
```

Convert Open result into a Biom table

```
python .../uc2otutab.py .../map.uc > .../otu_table.txt  
biom convert --table-type="OTU table" -i .../otu_table.txt -o  
.../otu_table.biom --to-json
```

Convert Swarm result into a Biom table

```
python .../uc2otutab.py .../map.uc > .../otu_table.txt  
biom convert --table-type="OTU table" -i .../otu_table.txt -o  
.../otu_table.biom --to-json
```

Krona Visualization

Compare taxonomic distribution of number of sequences and number of OTUs using Uparse

```
cat .../UPARSE_tax_assignments.txt | sed  
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed  
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |  
awk '{ $NF=""; print $0 }' | sed "s/;\t/g" | sed "s/k__//g" |  
sed "s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g"  
| sed "s/g__//g" | sed "s/s__//g" > .../RDP_interm.txt  
  
cat .../UPARSE_tax_assignments.txt | sed "s/;\t/g" | cut -f 1 >  
.../OTUnames_RDP.txt  
  
paste .../RDP_interm.txt .../OTUnames_RDP.txt > .../RDP_interm2.txt
```

```

printf '1\n%.0s' {1..662} > ../RDP_interm3.txt

paste ../RDP_interm3.txt ../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../OneSeqBy0tu.txt

cat ../taxo_rdp/UPARSE_tax_assignments.txt | sed "s/;/\t/g" | cut
-f 2 | sed "s/size=//g" > ../RDP_interm4.txt

paste ../RDP_interm4.txt ../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../NumberOfSequences.txt

ktImportText ../NumberOfSequences.txt -o
../NumberOfSequences.html

ktImportText ../OneSeqBy0tu.txt -o ../OneSeqBy0tu.html

ktImportKrona ../NumberOfSequences.html ../OneSeqBy0tu.html -
o ../TaxoRDP.html

```

Compare taxonomic distribution of number of sequences and number of OTUs using swarm

```

cat ../taxo_rdp/D1_withoutsingletons.txt | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > ../RDP_interm.txt

cat ../taxo_rdp/UPARSE_tax_assignments.txt | sed "s/;/\t/g" | cut
-f 1 > ../OTUnames_RDP.txt

paste ../RDP_interm.txt ../OTUnames_RDP.txt > ../RDP_interm2.txt

printf '1\n%.0s' {1..6832} > ../RDP_interm3.txt

paste ../RDP_interm3.txt ../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../OneSeqBy0tu_swarm.txt

cat ../taxo_rdp/D1_withoutsingletons.txt | sed "s/_/;size=g" |
sed "s/;/\t/g" | cut -f 2 | sed "s/size=//g" >

```

```

.../RDP_interm4.txt

paste .../RDP_interm4.txt .../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g">
.../NumberOfSequences_swarm.txt

ktImportText .../NumberOfSequences_swarm.txt -o
.../NumberOfSequences_swarm.html

ktImportText .../OneSeqBy0tu_swarm.txt -o
.../OneSeqBy0tu_swarm.html

ktImportKrona .../NumberOfSequences_swarm.html
.../OneSeqBy0tu_swarm.html -o .../TaxoSwarm.html

```

Compared different taxonomy assignment methods

RDP

```

cat .../taxo_rdp/UPARSE_tax_assignments.txt | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > .../RDP_interm.txt

cat .../taxo_rdp/UPARSE_tax_assignments.txt | sed "s/;/\t/g" | cut
-f 1 > .../OTUnames_RDP.txt

paste .../RDP_interm.txt .../OTUnames_RDP.txt > .../RDP_interm2.txt

printf '1\n%.0s' {1..662} > .../RDP_interm3.txt

paste .../RDP_interm3.txt .../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> .../RDP.txt

ktImportText .../RDP.txt -o .../RDP.html

```

blast

```

cat ../taxo_blast/UPARSE_tax_assignments.txt | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > ../blast_interm.txt

cat ../taxo_blast/UPARSE_tax_assignments.txt | sed "s/;/\t/g" |
cut -f 1 > ../OTUnames_blast.txt

paste ../blast_interm.txt ../OTUnames_blast.txt >
../blast_interm2.txt
printf '1\n%.0s' {1..662} > ../blast_interm3.txt
paste ../blast_interm3.txt ../blast_interm2.txt | sed "s/;/\t/g"
| sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../blast.txt

ktImportText ../blast.txt -o ../blast.html

```

mothur

```

cat ../taxo_mothur/UPARSE_tax_assignments.txt | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > ../mothur_interm.txt

cat ../taxo_mothur/UPARSE_tax_assignments.txt | sed "s/;/\t/g" |
cut -f 1 > ../OTUnames_mothur.txt

paste ../mothur_interm.txt ../OTUnames_mothur.txt >
../mothur_interm2.txt
printf '1\n%.0s' {1..662} > ../mothur_interm3.txt
paste ../mothur_interm3.txt ../mothur_interm2.txt | sed
"s/;/\t/g" | sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../mothur.txt

ktImportText ../mothur.txt -o ../mothur.html

```

uclust

```

cat ../taxo_uclust/UPARSE_tax_assignments.txt | sed

```



```

"s/Unassigned\s/k__Fungi;unassigned\t/g" | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed
"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > ../uclust_interm.txt

cat ../taxo_uclust/UPARSE_tax_assignments.txt | sed "s/;/\t/g" |
cut -f 1 > ../OTUnames_uclust.txt

paste ../uclust_interm.txt ../OTUnames_uclust.txt >
../uclust_interm2.txt

printf '1\n%.0s' {1..662} > ../uclust_interm3.txt

paste ../uclust_interm3.txt ../uclust_interm2.txt | sed
"s/;/\t/g" | sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../uclust.txt

ktImportText ../uclust.txt -o ../uclust.html

```

Final figure

```

ktImportKrona ../RDP.html ../blast.html ../mothur.html
../uclust.html -o ../TaxoSoftware.html

```

Compared different clustering methods (usearch/uclust/swarm)

uparse

Previously compute using RDP.

```

ktImportText ../RDP_uparse.txt -o ../RDP_uparse.html

```

swarm

```

cat ../taxo_rdp/D1_withoutsingletons.txt | sed
"s/k__Fungi\s/k__Fungi;unassigned\t/g" | sed

```

```

"s/p__Ascomycota\s/p__Ascomycota;unassigned\t/g" | cut -f 2- |
awk '{ $NF=""; print $0 }' | sed "s/;/\t/g" | sed "s/k__//g" | sed
"s/p__//g" | sed "s/c__//g" | sed "s/o__//g" | sed "s/f__//g" |
sed "s/g__//g" | sed "s/s__//g" > ../RDP_interm.txt

cat ../taxo_rdp/D1_withoutsingletons.txt | sed "s/;/\t/g" | cut -f
1 > ../OTUnames_RDP.txt

paste ../RDP_interm.txt ../OTUnames_RDP.txt > ../RDP_interm2.txt

printf '1\n%.0s' {1..6832} > ../RDP_interm3.txt

paste ../RDP_interm3.txt ../RDP_interm2.txt | sed "s/;/\t/g" |
sed "s/^\s*//g" | sed "s/\s*\t/\t/g"> ../RDP_swarm.txt

ktImportText ../RDP_swarm.txt -o ../RDP_swarm.html

```

Final figure

```

ktImportKrona ../RDP_uparse.html ../RDP_swarm.html -o
../TaxoClustering.html

```

Assigation of ecological guild using FUN guild

```

sed 's/;size=/_/' ../UPARSE_tax_assignments.txt | sed 's/;/\t1\t1/'
| cut -f1-4 > ../funguild_interm.txt

echo -e "OTU_ID\tS1\tS2\tTaxonomy\n$(cat ../funguild_interm.txt
)" > ../FUNGUILD.txt

python ../Guilds_v1.0.py -otu ../FUNGUILD.txt -db fungi -m -u

```