MICROBIOME ANALYSIS - MOTHUR

16S and metagenomics

By: Dr. SUDEEP D. GHATE

HANDS ON!

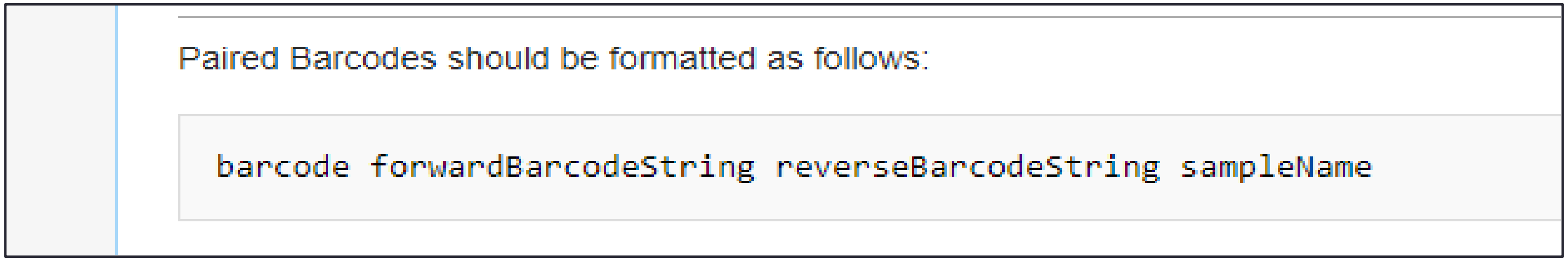
Let’s make science happen

Workflow

1. Retrieve data
2. Cluster sequences
3. Taxonomic classification
4. Phylogenetic tree construction
5. OTU table creation
6. Downstream visualization / analysis

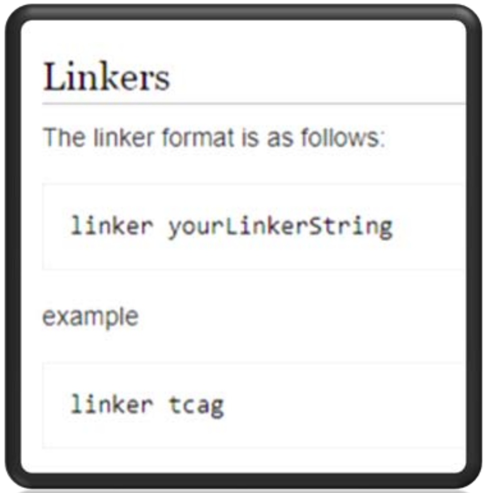
Oligos File

- The oligos file is used to provide barcodes and primers to mothur. Mothur removes items in the following order: linkers, barcodes, spacers then primers.



Paired Barcodes should be formatted as follows:

```
barcode forwardBarcodeString reverseBarcodeString sampleName
```



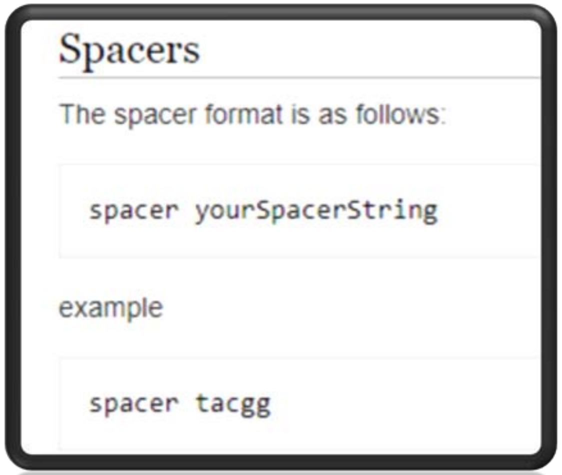
Linkers

The linker format is as follows:

```
linker yourLinkerString
```

example

```
linker tcag
```



Spacers

The spacer format is as follows:

```
spacer yourSpacerString
```

example

```
spacer tacgg
```

Make.contigs

- The make.contigs command parameters are file, ffastq, rfastq, ffasta, rfasta, fqfile, rqfile, findex, rindex, oligos, format, tdiffs, bdiffs, pdiffs, align, match, mismatch, gapopen, gapextend, insert, deltaq, maxee, allfiles and processors.
- if the base has a quality score below the threshold we eliminate it. Default=20.

Trim.seqs

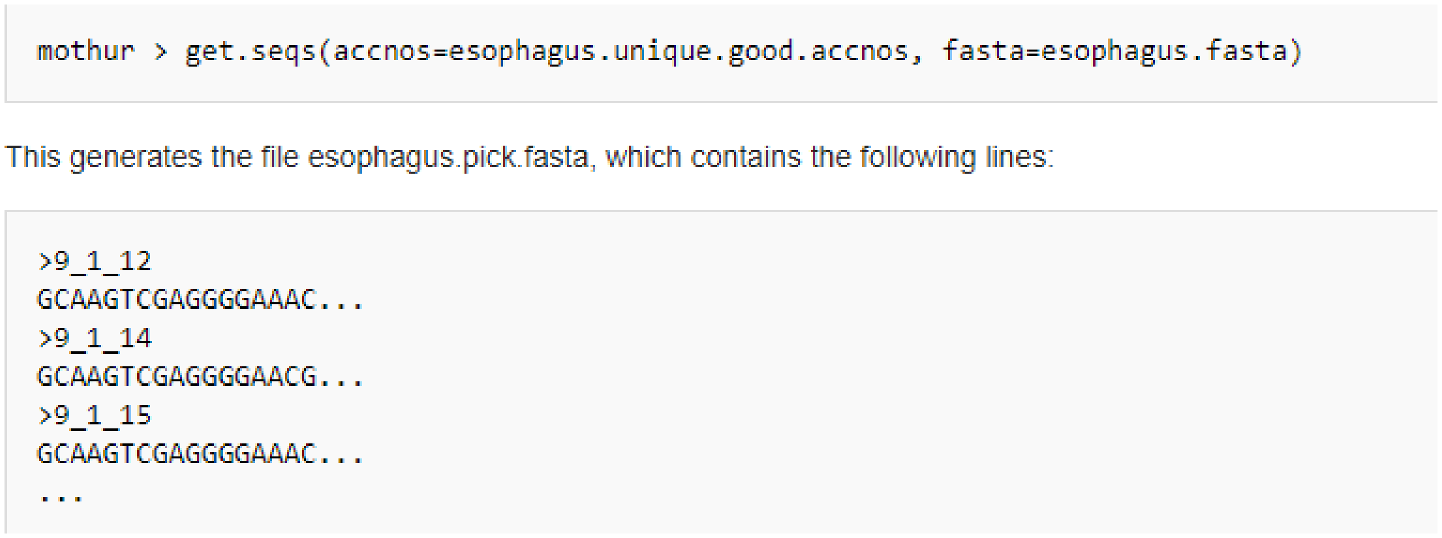
- The outputted sequences will be trimmed to remove the user-provided primers, barcodes, and sequences that drop below a quality threshold.
- **oligos**
- The oligos option takes a file that can contain the sequences of the forward and reverse primers and barcodes and their sample identifier.
- The forward primer is best thought of as the forward sequencing primer. So if you are using the 16S rRNA primers 27f and 338r to generate sequencing substrate, but you are sequencing off of the 338r end of the fragment, you would list 338r as the forward primer and 27f as the reverse.

List.seqs

- The **list.seqs** command will write out the names of the sequences found within a fastq, fasta, name, group, count, list, or align.report file. This could be useful for using the get.seqsand remove.seqs commands as well as to generate a group file.

Get.seqs

- The **get.seqs** command takes a list of sequence names (.accnos file) and either a fastq, fasta, name, group, list, count or align.report file to generate a new file that contains only the sequences in the list.



```
mothur > get.seqs(accnos=esophagus.unique.good.accnos, fasta=esophagus.fasta)
```

This generates the file esophagus.pick.fasta, which contains the following lines:

```
>9_1_12
GCAAGTCGAGGGGAAAC...
>9_1_14
GCAAGTCGAGGGGAACG...
>9_1_15
GCAAGTCGAGGGGAAAC...
...
```

Screen.seqs

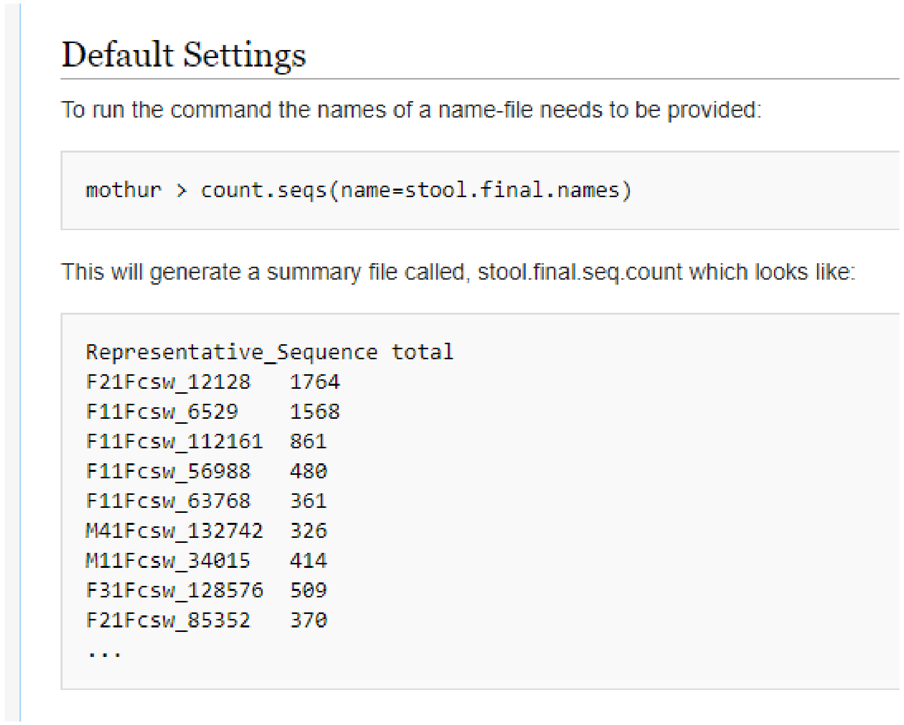
- The **screen.seqs** command enables you to keep sequences that fulfill certain user defined criteria. Furthermore, it enables you to cull those sequences not meeting the criteria from a names, group, contigsreport, alignreport and summary file.

unique.seqs

- The command returns only the unique sequences found in a fasta-formatted sequence file and a file that indicates those sequences that are identical to the reference sequence. Often times a collection of sequences will have a significant number of identical sequences. It sucks up considerable processing time to have to align, calculate distances, and cluster each of these sequences individually.

Count.seqs

- The **count.seqs** / make.table command counts the number of sequences represented by the representative sequence in a name file. If a group file is given, it will also provide the group count breakdown.



Default Settings

To run the command the names of a name-file needs to be provided:

```
mothur > count.seqs(name=stool.final.names)
```

This will generate a summary file called, stool.final.seq.count which looks like:

```
Representative_Sequence total
F21Fcsw_12128   1764
F11Fcsw_6529    1568
F11Fcsw_112161  861
F11Fcsw_56988   480
F11Fcsw_63768   361
M41Fcsw_132742  326
M11Fcsw_34015   414
F31Fcsw_128576  509
F21Fcsw_85352   370
...
```

Align.seqs

- The **align.seqs** command aligns a user-supplied fasta-formatted candidate sequence file to a user-supplied fasta-formatted template alignment.
- gotoh, and needleman - needleman is the default setting:

Filter.seqs

- **filter.seqs** removes columns from alignments based on a criteria defined by the user.
- **vertical**
- By default vertical option is set to T, and any column that only contains gap characters (i.e. '-' or '.') is ignore
- **trump**
- The trump option will remove a column if the trump character is found at that position in any sequence of the alignment.
- SILVA aligners will precede the first base of the sequence with a string of periods

Pre.cluster

- The **pre.cluster** command implements a pseudo-single linkage algorithm with the goal of removing sequences that are likely due to sequencing errors.

- **method**
- The method parameter allows you to specify the algorithm to use to complete the preclustering step. Possible methods include simple, tree, unoise, and deblur. Default=simple.

Chimera removal

- The chimera.uchime command reads a fasta file and reference file and outputs potentially chimeric sequences.

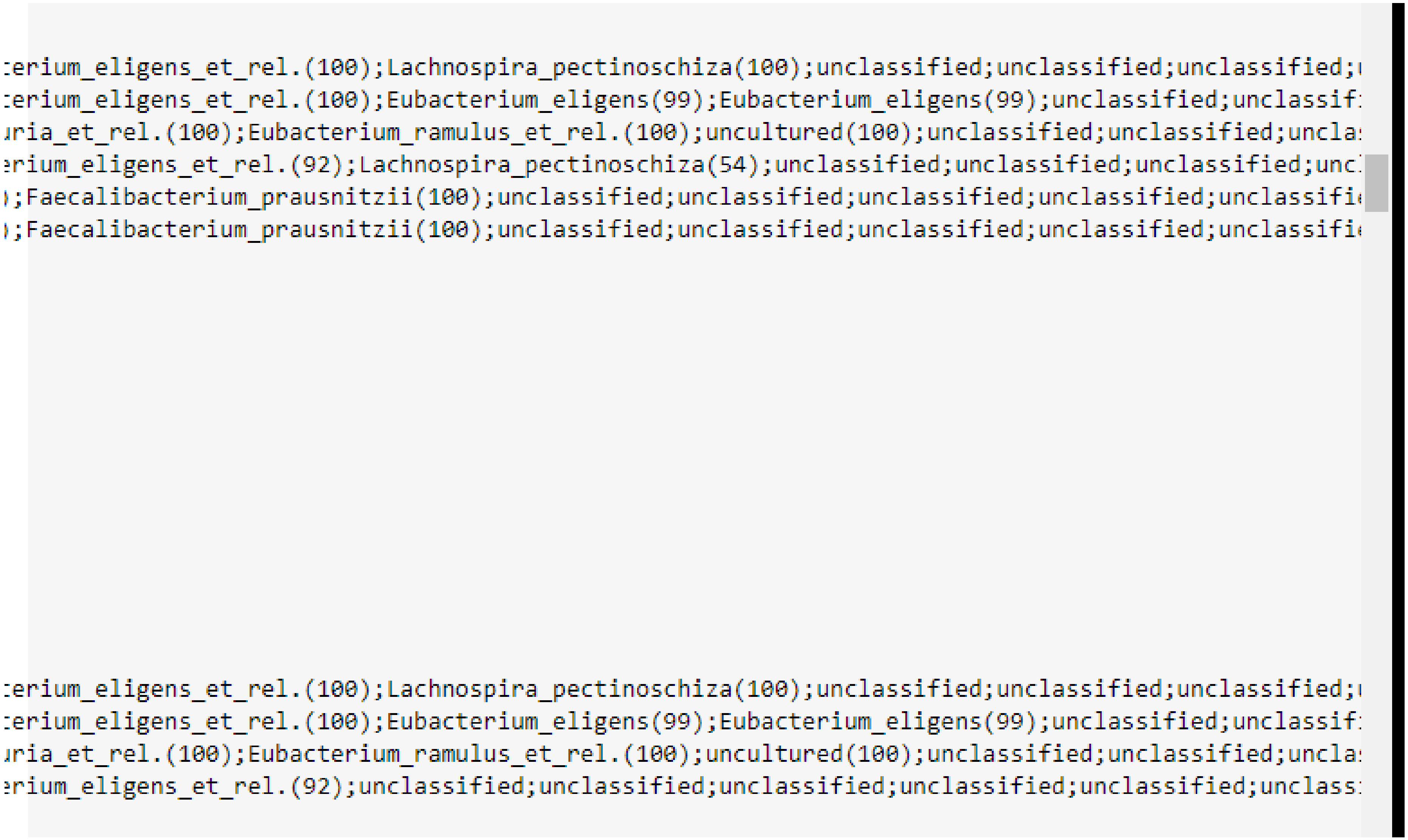
- Chimeric reads occur when one sequencing read aligns to two distinct portions of the genome with little or no overlap. Chimeric reads are indicative of structural variation. Chimeric reads are also called split reads.
- **dereplicate**
- The dereplicate parameter can be used when checking for chimeras by group. If the dereplicate parameter is false, then if one group finds the sequence to be chimeric, then all groups find it to be chimeric, default=f. If you set dereplicate=t, and then run remove.seqs with dups=f you can remove only the redundant chimeric sequences.

Classify.seqs

- The **classify.seqs** command allows the user to use several different methods to assign their sequences to the taxonomy outline of their choice. Current methods include using a k-nearest neighbor consensus and Wang approach

- method=wang
- the wang method looks at the query sequence kmer by kmer. The method looks at all taxonomies represented in the template, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then calculates the probability a query sequence would be in a given taxonomy based on the kmers it contains, and assign the query sequence to the taxonomy with the highest probability.

- method=knn
- The k-Nearest Neighbor algorithm involves identifying the k-most similar sequences in a database that are similar to your sequence. By default, mothur will find the 10 most similar sequences in the database. Once mothur has identified the k-most similar sequences, she will use the taxonomy information for each sequence to determine the consensus taxonomy. mothur gives you the ability to determine the method that is used to find the closest matches, the value of k



```
terium_eligens_et_rel.(100);Lachnospira_pectinoschiza(100);unclassified;unclassified;unclassified;
terium_eligens_et_rel.(100);Eubacterium_eligens(99);Eubacterium_eligens(99);unclassified;unclassif
uria_et_rel.(100);Eubacterium_ramulus_et_rel.(100);uncultured(100);unclassified;unclassified;uncla
erium_eligens_et_rel.(92);Lachnospira_pectinoschiza(54);unclassified;unclassified;unclassified;unc
);Faecalibacterium_prausnitzii(100);unclassified;unclassified;unclassified;unclassified;unclassifi
);Faecalibacterium_prausnitzii(100);unclassified;unclassified;unclassified;unclassified;unclassifi
```

```
terium_eligens_et_rel.(100);Lachnospira_pectinoschiza(100);unclassified;unclassified;unclassified;
terium_eligens_et_rel.(100);Eubacterium_eligens(99);Eubacterium_eligens(99);unclassified;unclassif
uria_et_rel.(100);Eubacterium_ramulus_et_rel.(100);uncultured(100);unclassified;unclassified;uncla
erium_eligens_et_rel.(92);unclassified;unclassified;unclassified;unclassified;unclassified;unclass
```

Dist.seqs

- The **dist.seqs** command will calculate uncorrected pairwise distances between aligned DNA sequences.
- Once a distance matrix gets read into mothur, the **cluster** command can be used to assign sequences to OTUs.

- **output**
- The output option allows you specify the form of the matrix generated by dist.seqs. By default, dist.seqs will generate a column-formatted matrix. You can set the output to "lt", for a phylip formatted lower triangle matrix, or to "square" for a phylip formatted square matrix. If output is set to lt or square the cutoff option is ignored.

Make.shared

- The make.shared command reads a list and group file or biom file and creates a .shared file as well as a rabund file for each group.

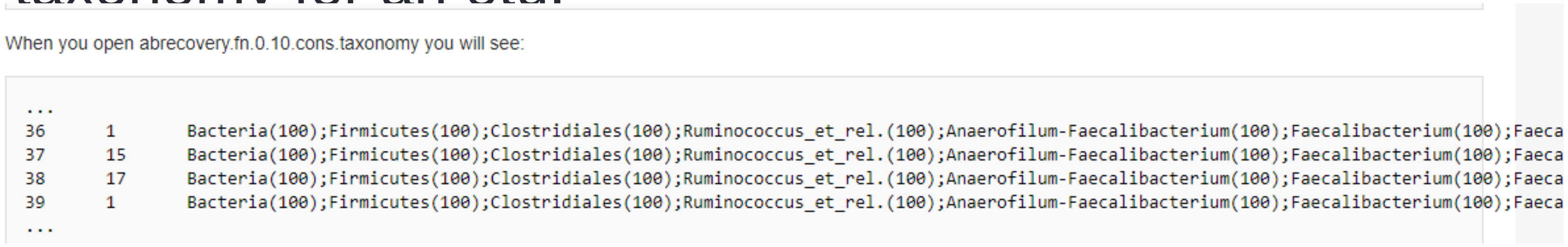
Shared file

- A shared file is analogous to an rabund file. The data in a shared file represent the number of times that an OTU is observed in multiple samples.

```
label	Group	numOtus	Otu01	Otu02	Otu03	Otu04	Otu05	Otu06	Otu07	Otu08	Otu09	Otu10
0.10	forest	55	0	5	2	3	1	1	3	3	1	0
0.10	pasture	55	7	2	5	1	3	2	0	0	1	2
```

Classify.otu

- The classify.otu command is used to get a consensus taxonomy for an otu.



When you open abrecovery.fn.0.10.cons.taxonomy you will see:

```
...
36      1       Bacteria(100);Firmicutes(100);Clostridiales(100);Ruminococcus_et_rel.(100);Anaerofilum-Faecalibacterium(100);Faecalibacterium(100);Faeca
37      15      Bacteria(100);Firmicutes(100);Clostridiales(100);Ruminococcus_et_rel.(100);Anaerofilum-Faecalibacterium(100);Faecalibacterium(100);Faeca
38      17      Bacteria(100);Firmicutes(100);Clostridiales(100);Ruminococcus_et_rel.(100);Anaerofilum-Faecalibacterium(100);Faecalibacterium(100);Faeca
39      1       Bacteria(100);Firmicutes(100);Clostridiales(100);Ruminococcus_et_rel.(100);Anaerofilum-Faecalibacterium(100);Faecalibacterium(100);Faeca
...
```

- The first column is the otu number, the second column is the number of sequences in the otu and the third column is the consensus taxonomy.

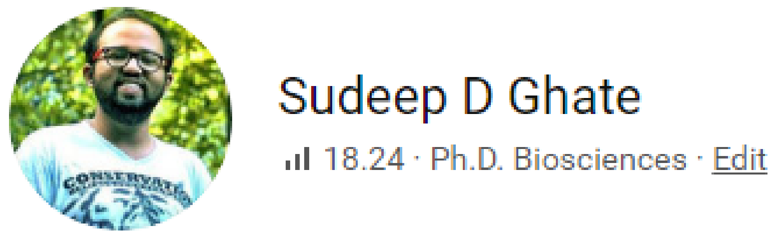
Contact details

DR. SUDEEP D. GHATE

MICROBIOME DIVISION

YENEPOYA RESEARCH CENTRE

- Ph – 9742352321
- **sudeep1129@gmail.com**



Sudeep D Ghate

18.24 · Ph.D. Biosciences · Edit