Transcriptomics - Walkthrough

Notebook:	Thesis Methods		
Created:	15/09/2020 15:48	**Updated:**	15/09/2020 17:41
Author:	Jess Friedersdorff		

Copying data to HPC:

The data from the Belfast sequencing hub who had already done some quality control.

I set up a tunnel from localhost to HPC to copy over all the fastq.gz files:

```
tunnel
scp -P 10022 *.gz user@localhost:~/
```

CHECK MD5s:
(note that 49906_3 sample was unzipped and rezipped locally before moving).

original_md5s.txt:

```
MD5 (007_1_S9_L001_R1_001.fastq.gz) = cd30b755abb1315c507903146555b063
MD5 (007_2_S10_L001_R1_001.fastq.gz) = a5c4271a45ba8a3f200f55826e4b205d
MD5 (007_3_S11_L001_R1_001.fastq.gz) = 4a2219a3caaa04f46585182481994385
MD5 (007_4_S12_L001_R1_001.fastq.gz) = ef2c1ff0d0e57d92f63017f3caaaa7dd
MD5 (12662_1_S17_L001_R1_001.fastq.gz) = 1abf3ca1ded43e61ed026ea11b77fd06
MD5 (12662_2_S18_L001_R1_001.fastq.gz) = 53afe96eb9540ae4988f1ecb38ac7e6e
MD5 (12662_3_S19_L001_R1_001.fastq.gz) = a2223b0664f6f6863e2f2e89f3500b7b
MD5 (12662_4_S20_L001_R1_001.fastq.gz) = 82e84ebf6d6d0ca30cc45330da77b42a
MD5 (49906_1_S13_L001_R1_001.fastq.gz) = 9e5d99da9cc30f4f9d6c5f6a6e44476b
MD5 (49906_2_S14_L001_R1_001.fastq.gz) = 5aca8fd11f81a410b6ff27d1a783954e
MD5 (49906_3_S15_L001_R1_001.fastq.gz) = 45f4851d536d0602ba2f43f865d2b0f4
MD5 (49906_4_S16_L001_R1_001.fastq.gz) = 71268b764a6aa81ba62e34aea75e6855
MD5 (Isol6_1_S21_L001_R1_001.fastq.gz) = fb0f2e229e180f6257493a924b6dc20f
MD5 (Isol6_2_S22_L001_R1_001.fastq.gz) = 3200047e817d36110464ef4ff0b30c8e
MD5 (Isol6_3_S23_L001_R1_001.fastq.gz) = 4dedc6e170a5634bbae205c0e171786e
MD5 (Isol6_4_S24_L001_R1_001.fastq.gz) = abdafe4c7d58bb9de186f9989e838628
MD5 (S85_1_S5_L001_R1_001.fastq.gz) = 2a0f6a30d0115d8c399b8eb65d1704b8
MD5 (S85_2_S6_L001_R1_001.fastq.gz) = 35366e38f3c298ec31cd1d842b822735
MD5 (S85_3_S7_L001_R1_001.fastq.gz) = f6a2e37e4c302ae6095681736f4fa232
MD5 (SY3_1_S1_L001_R1_001.fastq.gz) = 657109fb619de9a34962d892c6aaa6c0
MD5 (SY3_2_S2_L001_R1_001.fastq.gz) = 884a8b37e5e12e107caf21e171460fd6
MD5 (SY3_3_S3_L001_R1_001.fastq.gz) = 56a1d5ee89fed4d85fbae3bef26c8c95
MD5 (SYC_4_S4_L001_R1_001.fastq.gz) = 3e35d38bf76168c2e6707c0f251a5b9b
MD5 (s85_4_S8_L001_R1_001.fastq.gz) = 0f6f5ff2486bceb8527446c03eb4df9c
```

My_md5s.txt:

```
for i in *.gz; do md5sum ${i} >> My_md5s.txt; done
cat My_md5s.txt
cd30b755abb1315c507903146555b063  007_1_S9_L001_R1_001.fastq.gz
a5c4271a45ba8a3f200f55826e4b205d  007_2_S10_L001_R1_001.fastq.gz
4a2219a3caaa04f46585182481994385  007_3_S11_L001_R1_001.fastq.gz
ef2c1ff0d0e57d92f63017f3caaaa7dd  007_4_S12_L001_R1_001.fastq.gz
1abf3ca1ded43e61ed026ea11b77fd06  12662_1_S17_L001_R1_001.fastq.gz
53afe96eb9540ae4988f1ecb38ac7e6e  12662_2_S18_L001_R1_001.fastq.gz
a2223b0664f6f6863e2f2e89f3500b7b  12662_3_S19_L001_R1_001.fastq.gz
82e84ebf6d6d0ca30cc45330da77b42a  12662_4_S20_L001_R1_001.fastq.gz
9e5d99da9cc30f4f9d6c5f6a6e44476b  49906_1_S13_L001_R1_001.fastq.gz
5aca8fd11f81a410b6ff27d1a783954e  49906_2_S14_L001_R1_001.fastq.gz
16244cd2a8bc7bbc3ac3ad3a5ccbab68  49906_3_S15_L001_R1_001.fastq.gz
71268b764a6aa81ba62e34aea75e6855  49906_4_S16_L001_R1_001.fastq.gz
fb0f2e229e180f6257493a924b6dc20f  Isol6_1_S21_L001_R1_001.fastq.gz
3200047e817d36110464ef4ff0b30c8e  Isol6_2_S22_L001_R1_001.fastq.gz
4dedc6e170a5634bbae205c0e171786e  Isol6_3_S23_L001_R1_001.fastq.gz
abdafe4c7d58bb9de186f9989e838628  Isol6_4_S24_L001_R1_001.fastq.gz
2a0f6a30d0115d8c399b8eb65d1704b8  S85_1_S5_L001_R1_001.fastq.gz
35366e38f3c298ec31cd1d842b822735  S85_2_S6_L001_R1_001.fastq.gz
4a28342f9ef32abbaee36254fd299d48  S85_3_S7_L001_R1_001.fastq.gz
0f6f5ff2486bceb8527446c03eb4df9c  s85_4_S8_L001_R1_001.fastq.gz
657109fb619de9a34962d892c6aaa6c0  SY3_1_S1_L001_R1_001.fastq.gz
884a8b37e5e12e107caf21e171460fd6  SY3_2_S2_L001_R1_001.fastq.gz
56a1d5ee89fed4d85fbae3bef26c8c95  SY3_3_S3_L001_R1_001.fastq.gz
3e35d38bf76168c2e6707c0f251a5b9b  SYC_4_S4_L001_R1_001.fastq.gz

for i in $(cat My_md5s.txt | cut -d " " -f1); do grep ${i} original_md5s.txt ; done
MD5 (007_1_S9_L001_R1_001.fastq.gz) = cd30b755abb1315c507903146555b063
MD5 (007_2_S10_L001_R1_001.fastq.gz) = a5c4271a45ba8a3f200f55826e4b205d
MD5 (007_3_S11_L001_R1_001.fastq.gz) = 4a2219a3caaa04f46585182481994385
MD5 (007_4_S12_L001_R1_001.fastq.gz) = ef2c1ff0d0e57d92f63017f3caaaa7dd
MD5 (12662_1_S17_L001_R1_001.fastq.gz) = 1abf3ca1ded43e61ed026ea11b77fd06
MD5 (12662_2_S18_L001_R1_001.fastq.gz) = 53afe96eb9540ae4988f1ecb38ac7e6e
MD5 (12662_3_S19_L001_R1_001.fastq.gz) = a2223b0664f6f6863e2f2e89f3500b7b
MD5 (12662_4_S20_L001_R1_001.fastq.gz) = 82e84ebf6d6d0ca30cc45330da77b42a
MD5 (49906_1_S13_L001_R1_001.fastq.gz) = 9e5d99da9cc30f4f9d6c5f6a6e44476b
MD5 (49906_2_S14_L001_R1_001.fastq.gz) = 5aca8fd11f81a410b6ff27d1a783954e
MD5 (49906_4_S16_L001_R1_001.fastq.gz) = 71268b764a6aa81ba62e34aea75e6855
MD5 (Isol6_1_S21_L001_R1_001.fastq.gz) = fb0f2e229e180f6257493a924b6dc20f
MD5 (Isol6_2_S22_L001_R1_001.fastq.gz) = 3200047e817d36110464ef4ff0b30c8e
MD5 (Isol6_3_S23_L001_R1_001.fastq.gz) = 4dedc6e170a5634bbae205c0e171786e
MD5 (Isol6_4_S24_L001_R1_001.fastq.gz) = abdafe4c7d58bb9de186f9989e838628
MD5 (S85_1_S5_L001_R1_001.fastq.gz) = 2a0f6a30d0115d8c399b8eb65d1704b8
MD5 (S85_2_S6_L001_R1_001.fastq.gz) = 35366e38f3c298ec31cd1d842b822735
MD5 (s85_4_S8_L001_R1_001.fastq.gz) = 0f6f5ff2486bceb8527446c03eb4df9c
MD5 (SY3_1_S1_L001_R1_001.fastq.gz) = 657109fb619de9a34962d892c6aaa6c0
MD5 (SY3_2_S2_L001_R1_001.fastq.gz) = 884a8b37e5e12e107caf21e171460fd6
MD5 (SY3_3_S3_L001_R1_001.fastq.gz) = 56a1d5ee89fed4d85fbae3bef26c8c95
```

```
MD5 (SYC_4_S4_L001_R1_001.fastq.gz) = 3e35d38bf76168c2e6707c0f251a5b9b
```

Using Trimmomatic version 0.39:

```
for i in 007*.fastq; do output=$(echo ${i} | cut -d "." -f1) ; java -jar ~/bin/Trimmomatic-0.39/trimmomatic-0.39.jar SE -threads 4 -
phred33 ${i} ${output}_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 007_1_S9_L001_R1_001.fastq 007_1_S9_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 20129894 Surviving: 19800467 (98.36%) Dropped: 329427 (1.64%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 007_2_S10_L001_R1_001.fastq 007_2_S10_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 20741639 Surviving: 19457619 (93.81%) Dropped: 1284020 (6.19%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 007_3_S11_L001_R1_001.fastq 007_3_S11_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 18968644 Surviving: 18774212 (98.97%) Dropped: 194432 (1.03%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 007_4_S12_L001_R1_001.fastq 007_4_S12_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 20987815 Surviving: 20573172 (98.02%) Dropped: 414643 (1.98%)
TrimmomaticSE: Completed successfully
```

```
for i in 12662*.fastq; do output=$(echo ${i} | cut -d "." -f1) ; java -jar ~/bin/Trimmomatic-0.39/trimmomatic-0.39.jar SE -threads 4 -
phred33 ${i} ${output}_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 12662_1_S17_L001_R1_001.fastq 12662_1_S17_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 26333518 Surviving: 25506921 (96.86%) Dropped: 826597 (3.14%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 12662_2_S18_L001_R1_001.fastq 12662_2_S18_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 19946188 Surviving: 19228499 (96.40%) Dropped: 717689 (3.60%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 12662_3_S19_L001_R1_001.fastq 12662_3_S19_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 22916892 Surviving: 21439585 (93.55%) Dropped: 1477307 (6.45%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 12662_4_S20_L001_R1_001.fastq 12662_4_S20_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 23755207 Surviving: 22682024 (95.48%) Dropped: 1073183 (4.52%)
TrimmomaticSE: Completed successfully
```

```
for i in 49906*.fastq; do output=$(echo ${i} | cut -d "." -f1) ; java -jar ~/bin/Trimmomatic-0.39/trimmomatic-0.39.jar SE -threads 4 -
phred33 ${i} ${output}_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 49906_1_S13_L001_R1_001.fastq 49906_1_S13_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 17150102 Surviving: 16142405 (94.12%) Dropped: 1007697 (5.88%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 49906_2_S14_L001_R1_001.fastq 49906_2_S14_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 16271250 Surviving: 15653485 (96.20%) Dropped: 617765 (3.80%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 49906_3_S15_L001_R1_001.fastq 49906_3_S15_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 18138770 Surviving: 17741124 (97.81%) Dropped: 397646 (2.19%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 49906_4_S16_L001_R1_001.fastq 49906_4_S16_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 20096428 Surviving: 14700541 (73.15%) Dropped: 5395887 (26.85%)
TrimmomaticSE: Completed successfully
```

```
for i in Isol6*.fastq; do output=$(echo ${i} | cut -d "." -f1) ; java -jar ~/bin/Trimmomatic-0.39/trimmomatic-0.39.jar SE -threads 4 -
phred33 ${i} ${output}_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 Isol6_2_S22_L001_R1_001.fastq Isol6_2_S22_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 25034845 Surviving: 22801850 (91.08%) Dropped: 2232995 (8.92%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 Isol6_3_S23_L001_R1_001.fastq Isol6_3_S23_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 19792084 Surviving: 18655653 (94.26%) Dropped: 1136431 (5.74%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 Isol6_4_S24_L001_R1_001.fastq Isol6_4_S24_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 25372732 Surviving: 24307970 (95.80%) Dropped: 1064762 (4.20%)
TrimmomaticSE: Completed successfully

TrimmomaticSE: Started with arguments:
-threads 4 -phred33 Isol6_1_S21_L001_R1_001.fastq Isol6_1_S21_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 19041759 Surviving: 18159348 (95.37%) Dropped: 882411 (4.63%)
TrimmomaticSE: Completed successfully
```

```
for i in [sS]85*.fastq; do output=$(echo ${i} | cut -d "." -f1) ; java -jar ~/bin/Trimmomatic-0.39/trimmomatic-0.39.jar SE -threads 4 -
phred33 ${i} ${output}_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 S85_1_S5_L001_R1_001.fastq S85_1_S5_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 28944057 Surviving: 28802773 (99.51%) Dropped: 141284 (0.49%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 S85_2_S6_L001_R1_001.fastq S85_2_S6_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 22565532 Surviving: 22458082 (99.52%) Dropped: 107450 (0.48%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 s85_4_S8_L001_R1_001.fastq s85_4_S8_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 21420358 Surviving: 21191847 (98.93%) Dropped: 228511 (1.07%)
TrimmomaticSE: Completed successfully

TrimmomaticSE: Started with arguments:
-threads 4 -phred33 S85_3_S7_L001_R1_001.fastq S85_3_S7_L001_R1_001_trimmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Input Reads: 19724993 Surviving: 19504529 (98.88%) Dropped: 220464 (1.12%)
TrimmomaticSE: Completed successfully
```

```
for i in SY[3C]*.fastq; do output=$(echo ${i} | cut -d "." -f1) ;mmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 MINLEN:50; done
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 SY3_1_S1_L001_R1_001.fastq SY3_1_S1_L001_R1_001_trimmed.fastq ILLUMINACL
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0
Input Reads: 27321811 Surviving: 27076897 (99.10%) Dropped: 244914 (0.90%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 SY3_2_S2_L001_R1_001.fastq SY3_2_S2_L001_R1_001_trimmed.fastq ILLUMINACL
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0
Input Reads: 21551740 Surviving: 21301504 (98.84%) Dropped: 250236 (1.16%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 SY3_3_S3_L001_R1_001.fastq SY3_3_S3_L001_R1_001_trimmed.fastq ILLUMINACL
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0
Input Reads: 20263625 Surviving: 20167672 (99.53%) Dropped: 95953 (0.47%)
TrimmomaticSE: Completed successfully
TrimmomaticSE: Started with arguments:
-threads 4 -phred33 SYC_4_S4_L001_R1_001.fastq SYC_4_S4_L001_R1_001_trimmed.fastq ILLUMINACL
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0
Input Reads: 20108794 Surviving: 20065692 (99.79%) Dropped: 43102 (0.21%)
TrimmomaticSE: Completed successfully
```

Bowtie align trimmed reads to genes

```
for i in *.ffn; do bowtie2-build ${i} ${i}_index; done
```

Then run this command for each of the replicates and for each sample:

```
bowtie2 --threads 4 --sensitive-local -x Acetoanaerobium_sticklandii_12662.ffn_index -U 12662_1_S17_L001_R1_001_trimmed.fastq -S
12662_1_S17_L001_R1_001_bowtie.sam
```

```
for i in *.output; do echo -e "\n-------\n\n"${i}"\n"; cat ${i}; done


Acetoanaerobium_sticklandii_12662_bowtie.output


25506921 reads; of these:
  25506921 (100.00%) were unpaired; of these:
    431984 (1.69%) aligned 0 times
    24918071 (97.69%) aligned exactly 1 time
    156866 (0.61%) aligned >1 times
98.31% overall alignment rate
19228499 reads; of these:
  19228499 (100.00%) were unpaired; of these:
    392757 (2.04%) aligned 0 times
    18722337 (97.37%) aligned exactly 1 time
    113405 (0.59%) aligned >1 times
97.96% overall alignment rate
21439585 reads; of these:
  21439585 (100.00%) were unpaired; of these:
    430066 (2.01%) aligned 0 times
    20889397 (97.43%) aligned exactly 1 time
    120122 (0.56%) aligned >1 times
97.99% overall alignment rate
22682024 reads; of these:
  22682024 (100.00%) were unpaired; of these:
    437870 (1.93%) aligned 0 times
    22069084 (97.30%) aligned exactly 1 time
    175070 (0.77%) aligned >1 times
98.07% overall alignment rate


-------


Clostridium_aminophilum_F_bowtie.output


16142405 reads; of these:
  16142405 (100.00%) were unpaired; of these:
    11940858 (73.97%) aligned 0 times
    4025312 (24.94%) aligned exactly 1 time
    176235 (1.09%) aligned >1 times
26.03% overall alignment rate
15653485 reads; of these:
  15653485 (100.00%) were unpaired; of these:
    11530026 (73.66%) aligned 0 times
    3990025 (25.49%) aligned exactly 1 time
    133434 (0.85%) aligned >1 times
26.34% overall alignment rate
17741124 reads; of these:
  17741124 (100.00%) were unpaired; of these:
    13014445 (73.36%) aligned 0 times
    4515287 (25.45%) aligned exactly 1 time
    211392 (1.19%) aligned >1 times
26.64% overall alignment rate
14700541 reads; of these:
  14700541 (100.00%) were unpaired; of these:
    10098623 (68.70%) aligned 0 times
    4347685 (29.57%) aligned exactly 1 time
    254233 (1.73%) aligned >1 times
31.30% overall alignment rate


-------


Eubacterium_pyruvativorans_isolate6_bowtie.output

22801850 reads; of these:
  22801850 (100.00%) were unpaired; of these:
    599500 (2.63%) aligned 0 times
    21942069 (96.23%) aligned exactly 1 time
    260281 (1.14%) aligned >1 times
97.37% overall alignment rate
18655653 reads; of these:
  18655653 (100.00%) were unpaired; of these:
    406143 (2.18%) aligned 0 times
    18033829 (96.67%) aligned exactly 1 time
    215681 (1.16%) aligned >1 times
97.82% overall alignment rate
24307970 reads; of these:
  24307970 (100.00%) were unpaired; of these:
    632545 (2.60%) aligned 0 times
    23372801 (96.15%) aligned exactly 1 time
    302624 (1.24%) aligned >1 times
97.40% overall alignment rate
18159348 reads; of these:
  18159348 (100.00%) were unpaired; of these:
    447697 (2.47%) aligned 0 times
    17495470 (96.34%) aligned exactly 1 time
```

```
    216181 (1.19%) aligned >1 times
97.53% overall alignment rate


-------


Fibrobacter_succinogenes_subsp_bowtie.output


28802773 reads; of these:
  28802773 (100.00%) were unpaired; of these:
    1471520 (5.11%) aligned 0 times
    6065956 (21.06%) aligned exactly 1 time
    21265297 (73.83%) aligned >1 times
94.89% overall alignment rate
22458082 reads; of these:
  22458082 (100.00%) were unpaired; of these:
    1010195 (4.50%) aligned 0 times
    4898172 (21.81%) aligned exactly 1 time
    16549715 (73.69%) aligned >1 times
95.50% overall alignment rate
21191847 reads; of these:
  21191847 (100.00%) were unpaired; of these:
    1349403 (6.37%) aligned 0 times
    4771594 (22.52%) aligned exactly 1 time
    15070850 (71.12%) aligned >1 times
93.63% overall alignment rate
19504529 reads; of these:
  19504529 (100.00%) were unpaired; of these:
    1029658 (5.28%) aligned 0 times
    5013974 (25.71%) aligned exactly 1 time
    13460897 (69.01%) aligned >1 times
94.72% overall alignment rate


-------


Ruminococcus_albus_SY3_bowtie.output


27076897 reads; of these:
  27076897 (100.00%) were unpaired; of these:
    2677041 (9.89%) aligned 0 times
    22371477 (82.62%) aligned exactly 1 time
    2028379 (7.49%) aligned >1 times
90.11% overall alignment rate
21301504 reads; of these:
  21301504 (100.00%) were unpaired; of these:
    2883573 (13.54%) aligned 0 times
    16802522 (78.88%) aligned exactly 1 time
    1615409 (7.58%) aligned >1 times
86.46% overall alignment rate
20167672 reads; of these:
  20167672 (100.00%) were unpaired; of these:
    2209854 (10.96%) aligned 0 times
    16486494 (81.75%) aligned exactly 1 time
    1471324 (7.30%) aligned >1 times
89.04% overall alignment rate
20065692 reads; of these:
  20065692 (100.00%) were unpaired; of these:
    2217272 (11.05%) aligned 0 times
    16468360 (82.07%) aligned exactly 1 time
    1380060 (6.88%) aligned >1 times
88.95% overall alignment rate


-------


Ruminococcus_flavefaciens_007c_bowtie.output


19800467 reads; of these:
  19800467 (100.00%) were unpaired; of these:
    1341184 (6.77%) aligned 0 times
    18300867 (92.43%) aligned exactly 1 time
    158416 (0.80%) aligned >1 times
93.23% overall alignment rate
19457619 reads; of these:
  19457619 (100.00%) were unpaired; of these:
    1243911 (6.39%) aligned 0 times
    18054462 (92.79%) aligned exactly 1 time
    159246 (0.82%) aligned >1 times
93.61% overall alignment rate
18774212 reads; of these:
  18774212 (100.00%) were unpaired; of these:
    1175954 (6.26%) aligned 0 times
    17430691 (92.84%) aligned exactly 1 time
    167567 (0.89%) aligned >1 times
93.74% overall alignment rate
20573172 reads; of these:
  20573172 (100.00%) were unpaired; of these:
    1349659 (6.56%) aligned 0 times
    19074139 (92.71%) aligned exactly 1 time
    149374 (0.73%) aligned >1 times
93.44% overall alignment rate
```

To have a rough idea of the amount of ribosomal rRNA reads there are (also how many aligned to the ribosomal proteins):

```
grep -E "S ribosomal protein | ribosomal RNA" *.ffn | cut -d ">" -f2 | cut -d " " -f1 > ribosomal_gene_list.txt
```

```
for i in *.sam; do
    lines=$(wc -l ${i} | cut -d " " -f1);
    riboreads=$(grep -cf ribosomal_gene_list.txt ${i});
    echo -e ${i}"\t"${lines}"\t"${riboreads} | awk '{$4=$3/$2*100; print $0}';
done
```

I then converted all of the Sam Bowtie alignment files into Bam files (to save on space) and made the indeces using samtools in a new submission file. Using the -F flag option, I removed any alignments that either did not match (flag 4) or were not primary alignments (256).

```
for i in *.sam; do samtools view -bS -F260 ${i} | samtools sort -o ${i}_bowtie.bam; samtools index ${i}_bowtie.bam; done
```

Using Bowtie to align trimmed reads to fasta file of16S genes to determine likely culture in sample:

```
Fasta file:
cat RNA_genes_HAPs_NAPs.ffn
>Acetoanaerobium_sticklandii_12662__ANGDMIAC_02609 16S ribosomal RNA (partial)
GTAACGGCTCACCAAGGCAACGATCAGTAGCCGACCTGAGAGGGTGATCGGCCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGAAAGCCTGATGCAGCAACGCCGCGTGAGCGATGA
>Eubacterium_pyruvativorans_isolate6__EAINEGJO_01918 16S ribosomal RNA
TATCAAGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCGTGCCTAACACATGCAAGTCGAGCGGGAAGCTCACAAATGATTCTTCGGATGAATGCGTGAGTGGACAGCGGCGGACGGGTGAGTAACGCGTGGGCAACCTGCCCCTTACTGAA
>Fibrobacter_succinogenes_subsp._succinogenes_S85__DBPGGDCG_01077 16S ribosomal RNA
ATGAAGAGTTTGATCCTGGCTCAGAACGAACGCTGGTGGCGTGTCTTATACATGCAAGTCGAGCGAGACAGCAATGTCAAGCGGCGAACGGGTGAGTAACGCGTAAGCAATCTGCCCCATATCAGGAAATACCCGTGCCAACGCGCGGTTAATGT
--
>Fibrobacter_succinogenes_subsp._succinogenes_S85__DBPGGDCG_01896 16S ribosomal RNA
ATGAAGAGTTTGATCCTGGCTCAGAACGAACGCTGGTGGCGTGTCTTATACATGCAAGTCGAGCGAGGCAGCAATGCCGAGCGGCGAACGGGTGAGTAACGCGTAAGCAATCTGCCCCATATCAGGAAATACCCGTGCCAACGCGCGGTTAATGT
--
>Fibrobacter_succinogenes_subsp._succinogenes_S85__DBPGGDCG_02374 16S ribosomal RNA
ATGAAGAGTTTGATCCTGGCTCAGAACGAACGCTGGTGGCGTGTCTTATACATGCAAGTCGAGCGAGGCAGCAATGCCGAGCGGCGAACGGGTGAGTAACGCGTAAGCAATCTGCCCCATATCAGGAAATACCCGTGCCAACGCGCGGTTAATGT
>Ruminococcus_albus_SY3__OKNHPEEE_02463 16S ribosomal RNA
AATTAAGAGTTTGATCCTGGCTCAGGACGAACGCTGGCGGCACGCTTAACACATGCAAGTCGAACGAGCGAAAGAGTGCTTGCACTCTCTAGCTAGTGGCGGACGGGTGAGTAACACGTGAGCAATCTGCCTTTCGGTGGGGGATACCAATTGGA
>Ruminococcus_flavefaciens_007c__ECADDMPC_02129 16S ribosomal RNA
ATAAAGAGTTTGATCCTGGCTCAGGACGAACGCTGGCGGCACGCTTAACACATGCAAGTCGAACGGAGTTATTTTGAGTTTACTTAGAATAACTTAGTGGCGGACGGGTGAGTAACACGTGAGCAACCTACCTTAGAGAGAGGGATAGCTTCTGC
>Clostridium_aminophilum_F__gnl|X|LEMCAHCD_1:0-158
GTTATCCCCCGGTACAGGGCAGGTTACCCACGTGTTACTCACCCGTCCGCCACTAAAATTACAAAATTCCATCCGAAAACTTCCTTCTGCAATTTCCGTTCGACTTGCATGTGTTAGGCACGCCGCCAGCGTTCATCCTGAGCCAGGATCGAACT
>Clostridium_aminophilum_F__gnl|X|LEMCAHCD_12:103156-103314
GAGAGTTCGATCCTGGCTCAGGATGAACGCTGGCGGCGTGCCTAACACATGCAAGTCGAACGGAAATTGCAGAAGGAAGTTTTCGGATGGAATTTTGTAATTTTAGTGGCGGACGGGTGAGTAACACGTGGGTAACCTGCCCTGTACCGGGGGAT
>Clostridium_aminophilum_F__gnl|X|LEMCAHCD_16:81672-82101
GATAAGCCCTCGACCTATTAGTAACAGTCAGCTGCATGCATTACTGCACTTCCACCTCTGCCCTATCAACCTCATACTCTCTAAGGGGTCTTACTTCTTACGAATGGGATATCTCATCTTGAGGGGGGCTTCACGCTTAGATGCCTTCAGCGTTT
>Clostridium_aminophilum_F__gnl|X|LEMCAHCD_9:0-328
CACATCGACGGAGTGGTTTGGCACCTCGATGTCGGCTCATCGCATCCTGGGGCTGTAGCAGGTCCCAAGGGTTGGGCTGTTCGCCCATTAAAGCGGTACGCGAGCTGGGTTCAGAACGTCGTGAGACAGTTCGGTCCCTATCCGGCGCGGGCGCA
```

Example:

```
bowtie2 --threads 4 --sensitive-local -x RNA_genes_HAPs_NAPs -U 007_1_S9_L001_R1_001_trimmed.fastq -S
007_1_S9_L001_R1_001_trimmed.fastq_ribosomal.sam
```

Then run this:

```
for i in *.sam; do awk '!/^#/{if ($3 != "*") print $0}' ${i} | cut -f3 | cut -d "_" -f1,2,3 | sort | uniq -c > ${i}_counts; done
```

output:

```
for i in *_counts; do echo ${i}; cat ${i} | sort -nrk1 | head -6; echo -e "\n----\n"; done
```

007_1_S9_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1197418 Ruminococcus_flavefaciens_007c
510994 Clostridium_aminophilum_F
126489 Ruminococcus_albus_SY3
  700 Acetoanaerobium_sticklandii_12662
  178 Eubacterium_pyruvativorans_isolate6
  122 Fibrobacter_succinogenes_subsp.

007_2_S10_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1229042 Ruminococcus_flavefaciens_007c
440061 Clostridium_aminophilum_F
137282 Ruminococcus_albus_SY3
  758 Acetoanaerobium_sticklandii_12662
  214 Eubacterium_pyruvativorans_isolate6
  160 Fibrobacter_succinogenes_subsp.

007_3_S11_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1270852 Ruminococcus_flavefaciens_007c
409628 Clostridium_aminophilum_F
146092 Ruminococcus_albus_SY3
  518 Acetoanaerobium_sticklandii_12662
  157 Eubacterium_pyruvativorans_isolate6
   94 Fibrobacter_succinogenes_subsp.

007_4_S12_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1203271 Ruminococcus_flavefaciens_007c
444494 Clostridium_aminophilum_F
130704 Ruminococcus_albus_SY3
  844 Acetoanaerobium_sticklandii_12662

225 Eubacterium_pyruvativorans_isolate6
196 Fibrobacter_succinogenes_subsp.

12662_1_S17_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,041,318 Acetoanaerobium_sticklandii_12662
804,324 Clostridium_aminophilum_F
18,926 Ruminococcus_flavefaciens_007c
9277 Eubacterium_pyruvativorans_isolate6
720 Ruminococcus_albus_SY3
131 Fibrobacter_succinogenes_subsp.

12662_2_S18_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
764,943 Acetoanaerobium_sticklandii_12662
464,263 Clostridium_aminophilum_F
14862 Ruminococcus_flavefaciens_007c
4435 Eubacterium_pyruvativorans_isolate6
429 Ruminococcus_albus_SY3
66 Fibrobacter_succinogenes_subsp.

12662_3_S19_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
818,526 Acetoanaerobium_sticklandii_12662
543,162 Clostridium_aminophilum_F
16739 Ruminococcus_flavefaciens_007c
4511 Eubacterium_pyruvativorans_isolate6
511 Ruminococcus_albus_SY3
114 Fibrobacter_succinogenes_subsp.

12662_4_S20_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
914,522 Acetoanaerobium_sticklandii_12662
625,773 Clostridium_aminophilum_F
16836 Ruminococcus_flavefaciens_007c
8438 Eubacterium_pyruvativorans_isolate6
672 Ruminococcus_albus_SY3
148 Fibrobacter_succinogenes_subsp.

49906_1_S13_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
975,118 Clostridium_aminophilum_F
200,203 Eubacterium_pyruvativorans_isolate6
173105 Ruminococcus_flavefaciens_007c
155628 Ruminococcus_albus_SY3
87876 Acetoanaerobium_sticklandii_12662
54351 Fibrobacter_succinogenes_subsp.

49906_2_S14_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
888,989 Clostridium_aminophilum_F
236,230 Eubacterium_pyruvativorans_isolate6
185873 Ruminococcus_flavefaciens_007c
171704 Ruminococcus_albus_SY3
89189 Acetoanaerobium_sticklandii_12662
58949 Fibrobacter_succinogenes_subsp.

49906_3_S15_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,112,443 Clostridium_aminophilum_F
214,576 Eubacterium_pyruvativorans_isolate6
198273 Ruminococcus_flavefaciens_007c
189746 Ruminococcus_albus_SY3
101404 Acetoanaerobium_sticklandii_12662
66505 Fibrobacter_succinogenes_subsp.

49906_4_S16_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
718,153 Clostridium_aminophilum_F
210,723 Eubacterium_pyruvativorans_isolate6
161912 Ruminococcus_flavefaciens_007c
146724 Ruminococcus_albus_SY3
79197 Acetoanaerobium_sticklandii_12662
47250 Fibrobacter_succinogenes_subsp.

Isol6_1_S21_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
834,990 Eubacterium_pyruvativorans_isolate6

299660 Clostridium_aminophilum_F
   373 Acetoanaerobium_sticklandii_12662
   183 Ruminococcus_flavefaciens_007c
   159 Ruminococcus_albus_SY3
    54 Fibrobacter_succinogenes_subsp.

Isol6_2_S22_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,184,134 Eubacterium_pyruvativorans_isolate6
250,137 Clostridium_aminophilum_F
   295 Acetoanaerobium_sticklandii_12662
   224 Ruminococcus_albus_SY3
   218 Ruminococcus_flavefaciens_007c
   167 Fibrobacter_succinogenes_subsp.

Isol6_3_S23_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
705,413 Eubacterium_pyruvativorans_isolate6
344,989 Clostridium_aminophilum_F
   292 Ruminococcus_albus_SY3
   285 Ruminococcus_flavefaciens_007c
   282 Acetoanaerobium_sticklandii_12662
    80 Fibrobacter_succinogenes_subsp.

Isol6_4_S24_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,306,579 Eubacterium_pyruvativorans_isolate6
424465 Clostridium_aminophilum_F
   525 Acetoanaerobium_sticklandii_12662
   478 Ruminococcus_flavefaciens_007c
   380 Ruminococcus_albus_SY3
   218 Fibrobacter_succinogenes_subsp.

S85_1_S5_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
2,516,714 Fibrobacter_succinogenes_subsp.
308906 Clostridium_aminophilum_F
   104 Ruminococcus_flavefaciens_007c
    78 Ruminococcus_albus_SY3
    78 Eubacterium_pyruvativorans_isolate6
    43 Acetoanaerobium_sticklandii_12662

S85_2_S6_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
2,216,731 Fibrobacter_succinogenes_subsp.
236,164 Clostridium_aminophilum_F
    58 Ruminococcus_flavefaciens_007c
    36 Acetoanaerobium_sticklandii_12662
    35 Ruminococcus_albus_SY3
    30 Eubacterium_pyruvativorans_isolate6

S85_3_S7_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,707,683 Fibrobacter_succinogenes_subsp.
218938 Clostridium_aminophilum_F
   130 Ruminococcus_flavefaciens_007c
    61 Ruminococcus_albus_SY3
    50 Acetoanaerobium_sticklandii_12662
    38 Eubacterium_pyruvativorans_isolate6

S85_4_S8_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
1,918,344 Fibrobacter_succinogenes_subsp.
245613 Clostridium_aminophilum_F
    56 Ruminococcus_flavefaciens_007c
    55 Ruminococcus_albus_SY3
    47 Acetoanaerobium_sticklandii_12662
    35 Eubacterium_pyruvativorans_isolate6

SY3_1_S1_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
732,411 Ruminococcus_albus_SY3
267,877 Clostridium_aminophilum_F
  53465 Ruminococcus_flavefaciens_007c
   403 Acetoanaerobium_sticklandii_12662
   192 Fibrobacter_succinogenes_subsp.
   176 Eubacterium_pyruvativorans_isolate6

SY3_2_S2_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
561,670 Ruminococcus_albus_SY3
202,769 Clostridium_aminophilum_F
43858 Ruminococcus_flavefaciens_007c
218 Acetoanaerobium_sticklandii_12662
96 Fibrobacter_succinogenes_subsp.
85 Eubacterium_pyruvativorans_isolate6

SY3_3_S3_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
617,297 Ruminococcus_albus_SY3
217,617 Clostridium_aminophilum_F
50081 Ruminococcus_flavefaciens_007c
175 Acetoanaerobium_sticklandii_12662
139 Fibrobacter_succinogenes_subsp.
80 Eubacterium_pyruvativorans_isolate6

SY3_4_S4_L001_R1_001_trimmed.fastq_ribosomal.sam_counts
546,539 Ruminococcus_albus_SY3
200241 Clostridium_aminophilum_F
45778 Ruminococcus_flavefaciens_007c
152 Acetoanaerobium_sticklandii_12662
122 Fibrobacter_succinogenes_subsp.
54 Eubacterium_pyruvativorans_isolate6

FeatureCounts

```
featureCounts [options] -a <annotation_file> -o <output_file> input_file1 [input_file2]
```

default input is gff annotation and .sam files (.out from the bowtie).

I used the gffs and the bam alignment files for each sample, where all four bam files were given to feature counts.

Example:

```
featureCounts -t CDS -g ID -a Acetoanaerobium_sticklandii_12662.gff -o 12662_featurecounts 12662_1_S17_L001_R1_001_bowtie.sam.bam
12662_2_S18_L001_R1_001_bowtie.sam.bam 12662_3_S19_L001_R1_001_bowtie.sam.bam 12662_4_S20_L001_R1_001_bowtie.sam.bam
```

Matching genes to bactNOGs in emapper output:
I need to check which part of the 10th column (containing the nog group code in the form of number@NOG) contains the bactNOG, as it's not always the first one. I can use a loop in awk to check each column.

input:

```
head Acetoanaerobium_sticklandii_12662.emapper.annotations
# emapper version: emapper-2.0.1 emapper DB: 2.0
# command: ./emapper.py  --cpu 1 -i Acetoanaerobium_sticklandii_12662/Acetoanaerobium_sticklandii_12662.faa --output
Acetoanaerobium_sticklandii_12662 -m diamond
# time: Fri Jan  3 16:53:30 2020
#query_name     seed_eggNOG_ortholog    seed_ortholog_evalue    seed_ortholog_score     best_tax_level  Preferred_name
GOs                                                                                     EC       KEGG_ko
KEGG_Pathway    KEGG_Module     KEGG_Reaction   KEGG_rclass     BRITE   KEGG_TC CAZy    BiGG_Reaction  taxonomic
scope                                                                                   eggNOG OGs       best eggNOG OG  COG Functional
cat.     eggNOG free text desc.
ANGDMIAC_00001  1511.CLOST_2037 1e-204  719.2   Clostridia
                        Bacteria        1TSH8@1239,25B05@186801,COG2720@1,COG2720@2     NA|NA|NA        V       VanW like protein
ANGDMIAC_00002  1511.CLOST_2036 2.2e-78 298.1   Peptostreptococcaceae
                                Bacteria        1VDD6@1239,24NEN@186801,25UNJ@186804,COG0454@1,COG0456@2        NA|NA|NA
(GNAT) domain
ANGDMIAC_00003  1511.CLOST_2035 5.3e-189        666.8   Clostridia      hutG            3.5.3.8
ko:K01479       ko00340,ko01100,map00340,map01100                                                       M00045   R02285  RC00221,RC00681
ko00000,ko00001,ko00002,ko01000                         Bacteria        1TP2A@1239,2493V@186801,COG0010@1,COG0010@2
E       Belongs to the arginase family
```

First I need to edit the mapper file slightly, such that instead of a two columned list of all genes and the COGs, it should be a two columned list with the second list comma separated if there is more than one COG for that gene:

```
awk 'BEGIN{prevgene=""; prevcog=""} {if ($1 != prevgene) {print prevgene, prevcog; prevgene=$1; prevcog=$2} else {prevcog=prevcog","$2}}
END{print prevgene, prevcog}' gene_to_bactOG_map.list > gene_to_bactOG_mapper_edited.list
```

output:

```
ANGDMIAC_01066 COG0448@2
ANGDMIAC_01068 COG2195@2
ANGDMIAC_01069 COG1288@2
ANGDMIAC_01070 COG0664@2
ANGDMIAC_01071 COG1143@2,COG2006@2
```

```
for i in *featurecounts; do echo "OG_family "$(head -2 ${i} | tail -1 ) > ${i}_OGfamilies.table; awk 'BEGIN{while (getline <
"gene_to_bactOG_mapper_edited.list") array[$1]=$2}{if ($1 in array) {print array[$1]"\t" $0} else {print "noOG\t"$0}}' ${i} >>
${i}_OGfamilies.table ; done
```

output:

```
==> Sy3_featurecounts_OGfamilies.table <==
OG_family Geneid Chr Start End Strand Length SY3_1_S1_L001_R1_001_bowtie.sam.bam SY3_2_S2_L001_R1_001_bowtie.sam.bam
SY3_3_S3_L001_R1_001_bowtie.sam.bam SY3_4_S4_L001_R1_001_bowtie.sam.bam
COG4991@2,COG4886@2     OKNHPEEE_00001  gnl|X|OKNHPEEE_1        726     2183    -       1458    1552    1358    1134    1385
COG0515@2,COG3903@2     OKNHPEEE_00002  gnl|X|OKNHPEEE_1        2556    2990    +       435     323     282     188     191
COG0639@2       OKNHPEEE_00003  gnl|X|OKNHPEEE_1        3369    4043    -       675     577     444     398     468
```

```
COG4868@2       OKNHPEEE_00004  gnl|X|OKNHPEEE_1        4112    5578    -       1467    1594    1817    1266    1328
COG0842@2       OKNHPEEE_00005  gnl|X|OKNHPEEE_1        6130    7068    -       939     535     460     456     435
COG1131@2       OKNHPEEE_00006  gnl|X|OKNHPEEE_1        7068    8390    -       1323    282     229     219     200
COG1816@2       OKNHPEEE_00007  gnl|X|OKNHPEEE_1        8387    9397    -       1011    615     514     446     387
```

Then create a new table for each sample of just the counts for each COG. This should duplicate genes.
Note that this list of bactOGs contains the COGs and OGs from HAPs, SAPs and NAPs! Reduce this to just the ones from HAPs and NAPs by cutting the first column from the featurecount files and uniquing them:

```
for i in *.table; do sed '1d' ${i} | cut -f1; done | grep -v "noOG" | tr "," "\n" | sort | uniq > list_of_bactOGs.list

for j in *table; do name=$(echo ${j} | cut -d "_" -f1); echo OG ${name}_1 ${name}_2 ${name}_3 ${name}_4 > ${name}.counts; for i in $(cat
list_of_bactOGs.list); do if grep -wq ${i} ${j}; then awk "/${i}/ {print \"$i\",\$8,\$9,\$10,\$11}" ${j}; else echo ${i} x x x x ; fi
>>${name}.counts; done ;done
```

Now add up any counts were the OG occurs more than once in the genome:

```
for i in *.counts; do head -1 ${i} > ${i}.col; sort -k1 ${i} | awk 'BEGIN{prevfam=$1;rep1=$2;rep2=$3;rep3=$4;rep4=$5}{if ($1 !=prevfam)
{print prevfam, rep1, rep2, rep3, rep4; prevfam=$1; rep1=$2;rep2=$3; rep3=$4; rep4=$5;} else {rep1=rep1+$2;rep2=rep2+$3; rep3=rep3+$4;
rep4=rep4+$5;}} END{print prevfam, rep1, rep2, rep3, rep4}' >> ${i}.col; done
```

then paste together:

```
paste *.col | tr " " "\t" > OG_families_RNA_counts.tsv
```

There are 588 OG gene families common to HAPs and NAPs.

Matching genes to OG Identifiers in emapper output
First make a count table for the numbers of genes in each of the categories for each of the genomes: (COG Functional Category)

```
INFORMATION STORAGE AND PROCESSING
[J] Translation, ribosomal structure and biogenesis
[A] RNA processing and modification
[K] Transcription
[L] Replication, recombination and repair
[B] Chromatin structure and dynamics


CELLULAR PROCESSES AND SIGNALING
[D] Cell cycle control, cell division, chromosome partitioning
[Y] Nuclear structure
[V] Defense mechanisms
[T] Signal transduction mechanisms
[M] Cell wall/membrane/envelope biogenesis
[N] Cell motility
[Z] Cytoskeleton
[W] Extracellular structures
[U] Intracellular trafficking, secretion, and vesicular transport
[O] Posttranslational modification, protein turnover, chaperones


METABOLISM
[C] Energy production and conversion
[G] Carbohydrate transport and metabolism
[E] Amino acid transport and metabolism
[F] Nucleotide transport and metabolism
[H] Coenzyme transport and metabolism
[I] Lipid transport and metabolism
[P] Inorganic ion transport and metabolism
[Q] Secondary metabolites biosynthesis, transport and catabolism


POORLY CHARACTERIZED
[R] General function prediction only
[S] Function unknown

no_identifier
```

obtained from: ftp://ftp.ncbi.nih.gov/pub/wolf/COGs/COG0303/fun.txt (accessed 06/01/2020)

For each annotations file, make a list that contains the gene and the OG identifier letter.

```
for i in *.annotations; do cut -f1,21 ${i}; done > genes_and_OG_identifier.list

==> genes_and_OG_identifier.list <==
# emapper version: emapper-2.0.1 emapper DB: 2.0
# command: ./emapper.py  --cpu 1 -i Acetoanaerobium_sticklandii_12662/Acetoanaerobium_sticklandii_12662.faa --output
Acetoanaerobium_sticklandii_12662 -m diamond
# time: Fri Jan  3 16:53:30 2020
#query_name     COG Functional cat.
ANGDMIAC_00001  V
ANGDMIAC_00002  K
ANGDMIAC_00003  E
ANGDMIAC_00004  P
ANGDMIAC_00005  S
ANGDMIAC_00007  K
```

The add this column to the front of the featurecounts files.

```
for i in *featurecounts; do awk 'BEGIN{FS=OFS="\t"; while (getline < "genes_and_COG_identifier.list") array[$1]=$2} {if ($1 in array)
print array[$1], $0; else print "no_identifier", $0}' ${i} > ${i}_COGletter; done

Input:
head genes_and_OG_identifier.list
# emapper version: emapper-2.0.1 emapper DB: 2.0
# command: ./emapper.py  --cpu 1 -i Acetoanaerobium_sticklandii_12662/Acetoanaerobium_sticklandii_12662.faa --output
Acetoanaerobium_sticklandii_12662 -m diamond
# time: Fri Jan  3 16:53:30 2020
#query_name     COG Functional cat.
ANGDMIAC_00001  V
ANGDMIAC_00002  K
```

```
ANGDMIAC_00003  E

 head *featurecounts
==> 007c_featurecounts <==
# Program:featureCounts v2.0.0; Command:"featureCounts" "-t" "CDS" "-g" "ID" "-a" "Ruminococcus_flavefaciens_007c.gff" "-o"
"007c_featurecounts" "007_1_S9_L001_R1_001_bowtie.sam.bam" "007_2_S10_L001_R1_001_bowtie.sam.bam" "007_3_S11_L001_R1_001_bowtie.sam.bam"
"007_4_S12_L001_R1_001_bowtie.sam.bam"
Geneid  Chr     Start   End     Strand  Length  007_1_S9_L001_R1_001_bowtie.sam.bam     007_2_S10_L001_R1_001_bowtie.sam.bam    007_3_S11_L001_R1_001_bowti
ECADDMPC_00001  gnl|X|ECADDMPC_2        162     1397    +       1236    47      27      65      38
ECADDMPC_00002  gnl|X|ECADDMPC_2        1399    2388    +       990     28      18      39      20
ECADDMPC_00003  gnl|X|ECADDMPC_2        2375    3418    +       1044    42      31      62      36
ECADDMPC_00004  gnl|X|ECADDMPC_3        13      2451    -       2439    11      8       5       14



Output:
head *COGletter
==> 007c_featurecounts_COGletter <==
L       ECADDMPC_00001  gnl|X|ECADDMPC_2        162     1397    +       1236    47      27      65      38
L       ECADDMPC_00002  gnl|X|ECADDMPC_2        1399    2388    +       990     28      18      39      20
L       ECADDMPC_00003  gnl|X|ECADDMPC_2        2375    3418    +       1044    42      31      62      36
U       ECADDMPC_00004  gnl|X|ECADDMPC_3        13      2451    -       2439    11      8       5       14
S       ECADDMPC_00005  gnl|X|ECADDMPC_3        2534    2917    -       384     1       0       3       0
S       ECADDMPC_00006  gnl|X|ECADDMPC_3        2938    3879    -       942     2       4       1       2
        ECADDMPC_00007  gnl|X|ECADDMPC_3        3879    4448    -       570     6       1       1       3
```

Now make a list of all the letters and the counts that correspond:

```
for j in *COGletter; do name=$(echo $j | cut -d "_" -f1); echo -e ${name}"\t"${name}"_1\t"${name}"_2\t"${name}"_3\t"${name}"_4" >
${name}_COGidentifier_counts; for i in $(cut -f1 COG_functional_identifiers.list); do if grep -wq ${i} ${j}; then grep -w ${i} ${j} | cut
-f1,8,9,10,11; else echo -e ${i} "\t0\t0\t0\t0"; fi ;done >> ${name}_COGidentifier_counts; done

inputs:
head Sy3*COGletter
T       OKNHPEEE_00001  gnl|X|OKNHPEEE_1        726     2183    -       1458    1552    1358    1134    1385
KLT     OKNHPEEE_00002  gnl|X|OKNHPEEE_1        2556    2990    +       435     323     282     188     191
T       OKNHPEEE_00003  gnl|X|OKNHPEEE_1        3369    4043    -       675     577     444     398     468
S       OKNHPEEE_00004  gnl|X|OKNHPEEE_1        4112    5578    -       1467    1594    1817    1266    1328
V       OKNHPEEE_00005  gnl|X|OKNHPEEE_1        6130    7068    -       939     535     460     456     435
V       OKNHPEEE_00006  gnl|X|OKNHPEEE_1        7068    8390    -       1323    282     229     219     200

head COG_functional_identifiers.list
J       Translation, ribosomal structure and biogenesis
A       RNA processing and modification
K       Transcription
L       Replication, recombination and repair
B       Chromatin structure and dynamics
D       Cell cycle control, cell division, chromosome partitioning

output:
==> Sy3_COGidentifier_counts <==
Sy3     Sy3_1   Sy3_2   Sy3_3   Sy3_4
J       848     200     480     246
J       372498  392401  313800  273022
J       36761   34394   29949   31771
J       17823   16406   10652   18098
J       23440   20014   12791   22689
J       388     268     281     237


for i in *COGidentifier_counts; do awk 'BEGIN{FS=OFS="\t"; prevfam=$1;rep1=$2;rep2=$3;rep3=$4;rep4=$5}{if ($1 !=prevfam) {print prevfam,
rep1, rep2, rep3, rep4; prevfam=$1; rep1=$2; rep2=$3; rep3=$4; rep4=$5;} else {rep1=rep1+$2;rep2=rep2+$3; rep3=rep3+$4; rep4=rep4+$5}}
END{print prevfam, rep1, rep2, rep3, rep4}' ${i} > ${i}.cols; done

paste *.cols -d "\t" > COG_identifier_RNA_counts.table
```

Resulting raw counts:

		12662_1	12662_2	12662_3	12662_4	49906_1	49906_2	49906_3	49906_4	Isol6_1	Isol6_2	Isol6_3	Isol6
J	Translation, ribosomal structure and biogenesis	438860	404634	390081	482776	480582	462162	518274	583404	423012	515656	395210	5235
A	RNA processing and modification	0	0	0	0	0	0	0	0	0	0	0	0
K	Transcription	236674	223577	219892	241608	229985	233355	244995	242015	107921	134527	108479	1479
L	Replication, recombination and repair	200486	192202	183894	210751	181170	189131	190659	195706	91419	114385	98488	1261
B	Chromatin structure and dynamics	0	0	0	0	0	0	0	0	0	0	0	0
D	Cell cycle control, cell division, chromosome partitioning	107132	98472	105460	106123	71806	78363	82826	72673	43702	55141	45242	5864
Y	Nuclear structure	0	0	0	0	0	0	0	0	0	0	0	0
V	Defense mechanisms	54889	52515	49014	56693	39425	43090	45876	39746	11398	12006	11993	1398
T	Signal transduction mechanisms	165634	161005	155873	152651	113242	121800	132019	102105	33237	46208	41935	4824

M	Cell wall/membrane/envelope biogenesis	311866	273486	268288	358281	167368	177965	197643	163804	820637	926380	781337	1126
N	Cell motility	146905	146810	130447	160820	35334	37108	42416	33215	1230	1235	1377	1769
Z	Cytoskeleton	0	0	0	0	0	0	0	0	0	0	0	0
W	Extracellular structures	0	0	0	0	0	0	0	0	0	0	0	0
U	Intracellular trafficking, secretion, and vesicular transport	55866	49363	48658	52161	44457	47953	52694	44435	52219	66508	62189	7239
O	Posttranslational modification, protein turnover, chaperones	121372	109113	112337	126900	96094	99716	111093	88036	104740	123746	94960	1293
C	Energy production and conversion	899001	739414	832332	838229	393545	378263	454121	408909	705765	874406	812066	9493
G	Carbohydrate transport and metabolism	277150	231956	261282	289759	136172	155633	173785	140123	63070	88790	80915	9967
E	Amino acid transport and metabolism	820672	708769	718195	881492	202912	213798	241265	174916	412103	477557	449548	5531
F	Nucleotide transport and metabolism	169541	143264	147256	170565	109425	113584	125961	102224	65235	79121	74400	8483
H	Coenzyme transport and metabolism	191289	168971	175259	200328	93597	94834	105414	98210	59388	70494	68141	7728
I	Lipid transport and metabolism	104778	90295	96565	110030	105552	101995	121776	98538	238315	275813	271372	3191
P	Inorganic ion transport and metabolism	242284	213623	227483	228192	126397	130267	146154	119426	141223	182933	139301	2038
Q	Secondary metabolites biosynthesis, transport and catabolism	34649	31280	31020	37694	20395	21250	24184	24633	15782	17570	15215	2190
R	General function prediction only	0	0	0	0	0	0	0	0	0	0	0	0
S	Function unknown	853577	783165	784411	853759	1032345	872781	1157470	1361120	216146	250440	223687	2863
no_identifier	No COG function available	78898	89080	92852	73338	207868	213552	209261	215820	60257	79031	58321	1007
	total	5511523	4910994	5030599	5632150	3887671	3786600	4377886	4309058	3666799	4391947	3834176	4945

Resulting percentages:

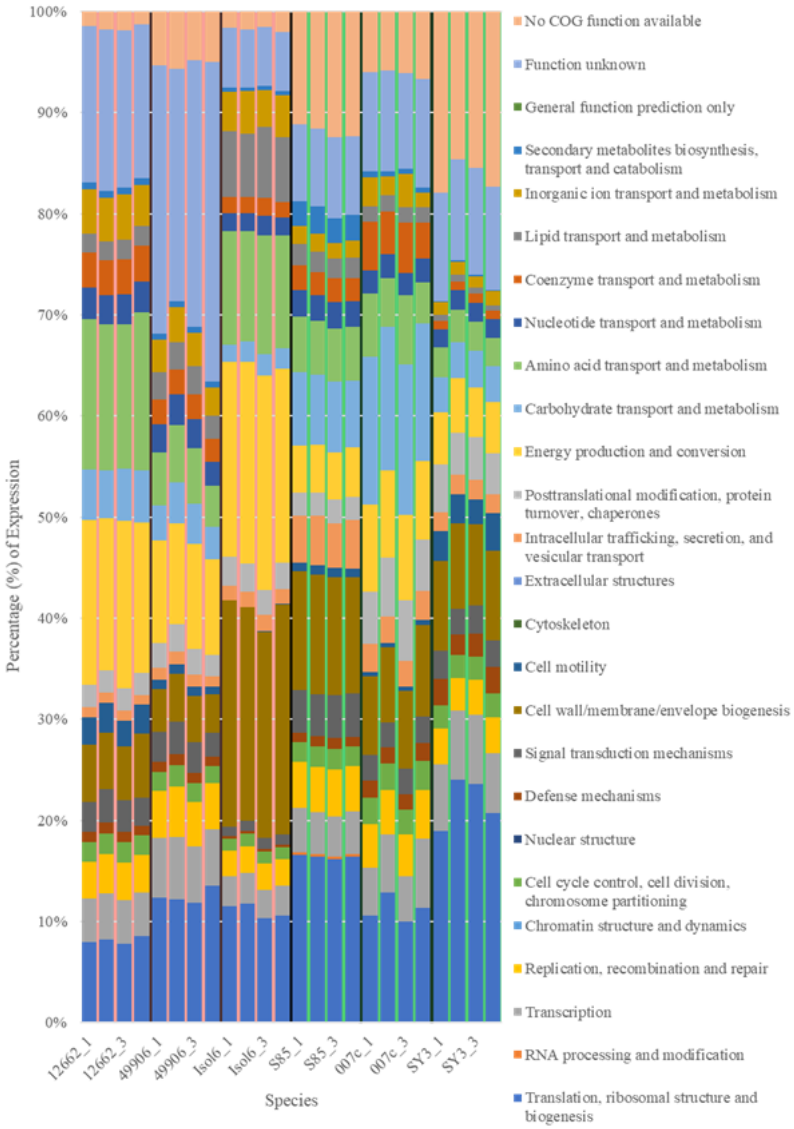
		12662_1	12662_2	12662_3	12662_4	49906_1	49906_2	49906_3	49906_4	Isol6_1	Isol6_2	Isol6_3	Is
J	Translation, ribosomal structure and biogenesis	7.96259	8.23935	7.754166	8.571789	12.36169	12.2052	11.83845	13.53901	11.53627	11.74094	10.30756	1
A	RNA processing and modification	0	0	0	0	0	0	0	0	0	0	0	0
K	Transcription	4.294167	4.552581	4.37109	4.289801	5.915753	6.162653	5.596194	5.616425	2.943194	3.063038	2.829265	2
L	Replication, recombination and repair	3.637579	3.913709	3.655509	3.741928	4.660117	4.994745	4.355047	4.541735	2.493155	2.604426	2.568688	2
B	Chromatin structure and dynamics	0	0	0	0	0	0	0	0	0	0	0	0
D	Cell cycle control, cell division, chromosome partitioning	1.943782	2.005134	2.096371	1.884236	1.847018	2.069482	1.891918	1.686517	1.19183	1.255502	1.179967	1
Y	Nuclear structure	0	0	0	0	0	0	0	0	0	0	0	0
V	Defense mechanisms	0.995895	1.069335	0.974317	1.006596	1.014103	1.13796	1.047903	0.922383	0.310843	0.273364	0.312792	0
T	Signal transduction mechanisms	3.005231	3.278461	3.098498	2.71035	2.912849	3.216606	3.015588	2.369543	0.906431	1.052107	1.093716	0
M	Cell wall/membrane/envelope biogenesis	5.658436	5.568852	5.333122	6.361354	4.305097	4.699863	4.514576	3.801388	22.3802	21.0927	20.37822	2
N	Cell motility	2.665416	2.989415	2.593071	2.855393	0.908873	0.979982	0.968869	0.770818	0.033544	0.02812	0.035914	0

Z	Cytoskeleton	0	0	0	0	0	0	0	0	0	0	0	0
W	Extracellular structures	0	0	0	0	0	0	0	0	0	0	0	0
U	Intracellular trafficking, secretion, and vesicular transport	1.013622	1.005153	0.967241	0.926129	1.143538	1.266387	1.20364	1.0312	1.424103	1.514317	1.621965	1
O	Posttranslational modification, protein turnover, chaperones	2.20215	2.221811	2.233074	2.253136	2.471763	2.633391	2.537595	2.043045	2.856442	2.817566	2.476673	2
C	Energy production and conversion	16.3113	15.0563	16.54539	14.88293	10.1229	9.989516	10.37307	9.489522	19.24744	19.9093	21.17967	1
G	Carbohydrate transport and metabolism	5.028556	4.723199	5.193855	5.144732	3.502663	4.110099	3.96961	3.251824	1.720029	2.021655	2.110362	2
E	Amino acid transport and metabolism	14.89011	14.43229	14.27653	15.65107	5.219372	5.646173	5.510993	4.059263	11.23877	10.87347	11.72476	1
F	Nucleotide transport and metabolism	3.076119	2.91721	2.927206	3.028417	2.814667	2.99963	2.877211	2.372305	1.779072	1.801502	1.940443	1
H	Coenzyme transport and metabolism	3.47071	3.440668	3.483859	3.556865	2.407534	2.504463	2.407874	2.279152	1.619614	1.605074	1.777201	1
I	Lipid transport and metabolism	1.901072	1.83863	1.919553	1.953606	2.715045	2.693577	2.781617	2.286764	6.499265	6.279971	7.077714	6
P	Inorganic ion transport and metabolism	4.395954	4.349893	4.521986	4.051597	3.251227	3.44021	3.338461	2.771511	3.851397	4.165191	3.63314	4
Q	Secondary metabolites biosynthesis, transport and catabolism	0.628665	0.636938	0.616626	0.669265	0.524607	0.561189	0.552413	0.571656	0.430403	0.40005	0.396826	0
R	General function prediction only	0	0	0	0	0	0	0	0	0	0	0	0
S	Function unknown	15.48713	15.94718	15.5928	15.15867	26.55433	23.0492	26.43902	31.58741	5.894678	5.702255	5.834031	5
no_identifier	No COG function available	1.43151	1.813889	1.845744	1.302132	5.346852	5.639677	4.779955	5.008519	1.643313	1.799452	1.521083	2
	total	100	100	100	100	100	100	100	100	100	100	100	1

Averages and standard deviations of the percentages (to 1 d.p):

		AVG_HAP	STD_HAP	AVG_NAP	STV_NAP
J	Translation, ribosomal structure and biogenesis	10.6	2.0	16.5	4.8
A	RNA processing and modification	0.0	0.0	0.1	0.1
K	Transcription	4.4	1.2	5.4	1.2
L	Replication, recombination and repair	3.6	0.9	4.1	0.5
B	Chromatin structure and dynamics	0.0	0.0	0.0	0.0
D	Cell cycle control, cell division, chromosome partitioning	1.7	0.4	2.3	0.3
Y	Nuclear structure	0.0	0.0	0.0	0.0
V	Defense mechanisms	0.8	0.4	1.7	0.6
T	Signal transduction mechanisms	2.3	1.0	3.1	0.8
M	Cell wall/membrane/envelope biogenesis	10.6	8.2	9.4	1.7
N	Cell motility	1.2	1.2	1.5	1.2
Z	Cytoskeleton	0.0	0.0	0.0	0.0
W	Extracellular structures	0.0	0.0	0.0	0.0

U	Intracellular trafficking, secretion, and vesicular transport	1.2	0.2	3.1	1.2
O	Posttranslational modification, protein turnover, chaperones	2.4	0.3	4.0	1.4
C	Energy production and conversion	15.2	4.3	6.1	1.7
G	Carbohydrate transport and metabolism	3.6	1.3	8.3	4.7
E	Amino acid transport and metabolism	10.4	4.2	4.6	1.4
F	Nucleotide transport and metabolism	2.5	0.6	2.3	0.3
H	Coenzyme transport and metabolism	2.5	0.8	2.5	1.5
I	Lipid transport and metabolism	3.7	2.2	1.4	0.6
P	Inorganic ion transport and metabolism	3.8	0.5	1.8	0.7
Q	Secondary metabolites biosynthesis, transport and catabolism	0.5	0.1	1.1	1.1
R	General function prediction only	0.0	0.0	0.0	0.0
S	Function unknown	16.1	9.2	9.4	1.2
no_identifier	No COG function available	2.8	1.8	11.5	4.4
	total	100.0	0.0	100.0	0.0



I then wanted to look into this later, so reran this command but changed it slightly:

```
for i in *featurecounts; do awk 'BEGIN{FS=OFS="\t"; while (getline < "genes_and_COG_identifier.list") array[$1]=$2} {if ($1 in array)
print array[$1], $0; else print "no_identifier", $0}' ${i} > ${i}_COGletter; done
```

Matching genes to K0s in emapper output

Make file for each species with all the kos:

```
for i in *.annotations; do cut -f1,9 ${i} > ${i}_kos.list; done
```

```
INPUT:
==> Ruminococcus_albus_SY3.emapper.annotations <==
# emapper version: emapper-2.0.1 emapper DB: 2.0
# command: ./emapper.py  --cpu 1 -i Ruminococcus_albus_SY3/Ruminococcus_albus_SY3.faa --output Ruminococcus_albus_SY3 -m diamond
# time: Fri Jan  3 16:52:39 2020
#query_name     seed_eggNOG_ortholog    seed_ortholog_evalue    seed_ortholog_score     best_tax_level  Preferred_name
GOs                                                                                        EC       KEGG_ko
KEGG_Pathway    KEGG_Module     KEGG_Reaction   KEGG_rclass     BRITE   KEGG_TC CAZy    BiGG_Reaction  taxonomic
scope                                                                          eggNOG OGs       best eggNOG OG  COG Functional
cat.     eggNOG free text desc.
OKNHPEEE_00001  697329.Rumal_1442       3.4e-59 236.1   Bacteria                        3.2.1.8 ko:K01181
       ko00000,ko01000                         Bacteria        COG3103@1,COG4886@1,COG4886@2,COG4991@2
NA|NA|NA       T                                                                                       Sh3 type 3 domain protein
OKNHPEEE_00002  697329.Rumal_3442       6e-36   157.1   Ruminococcaceae                         ko:K16247
       ko00000,ko03000                         Bacteria        1TY16@1239,24A3H@186801,3WIWE@541000,COG0515@1,COG0515@2,COG3903@1,COG3903@2
KLT     Protein kinase domain
```

```
OUTPUT:
==> Ruminococcus_albus_SY3.emapper.annotations_kos.list <==
# emapper version: emapper-2.0.1 emapper DB: 2.0
# command: ./emapper.py  --cpu 1 -i Ruminococcus_albus_SY3/Ruminococcus_albus_SY3.faa --output Ruminococcus_albus_SY3 -m diamond
# time: Fri Jan  3 16:52:39 2020
#query_name     KEGG_ko
OKNHPEEE_00001  ko:K01181
OKNHPEEE_00002  ko:K16247
OKNHPEEE_00003
```

```
OKNHPEEE_00004
OKNHPEEE_00005  ko:K21397
OKNHPEEE_00006  ko:K01990,ko:K21397
```

make list of all possible kos:

```
cat *.list | cut -f2 | tr "," "\n" | sed 's/ko://g' | awk '{if ($0 !~ /#/) print $0}' | sort | uniq | sed '$d' >
list_of_all_possible_K0s.list

OUTPUT:
head list_of_all_possible_K0s.list
K00001
K00003
K00005
K00008
K00009
K00012
K00013
K00014
K00016
```

add the KOs to the featurecounts output as a new column:

```
for i in *featurecounts; do awk 'BEGIN{while (getline < "all_annotated_kos.list") array[$1]=$2} {if ($1 in array) {print array[$1], $0}}'
${i} > ${i}_K0s; done


INPUT:
==> Sy3_featurecounts <==
# Program:featureCounts v2.0.0; Command:"featureCounts" "-t" "CDS" "-g" "ID" "-a" "Ruminococcus_albus_SY3.gff" "-o" "Sy3_featurecounts"
"SY3_1_S1_L001_R1_001_bowtie.sam.bam" "SY3_2_S2_L001_R1_001_bowtie.sam.bam" "SY3_3_S3_L001_R1_001_bowtie.sam.bam"
"SY3_4_S4_L001_R1_001_bowtie.sam.bam"
Geneid  Chr     Start   End     Strand  Length  SY3_1_S1_L001_R1_001_bowtie.sam.bam     SY3_2_S2_L001_R1_001_bowtie.sam.bam
OKNHPEEE_00001  gnl|X|OKNHPEEE_1        726     2183    -       1458    1552    1358    1134    1385
OKNHPEEE_00002  gnl|X|OKNHPEEE_1        2556    2990    +       435     323     282     188     191
OKNHPEEE_00003  gnl|X|OKNHPEEE_1        3369    4043    -       675     577     444     398     468
OKNHPEEE_00004  gnl|X|OKNHPEEE_1        4112    5578    -       1467    1594    1817    1266    1328

OUTPUT:
==> Sy3_featurecounts_K0s <==
Rate: # Program:featureCounts v2.0.0; Command:"featureCounts" "-t" "CDS" "-g" "ID" "-a" "Ruminococcus_albus_SY3.gff" "-o"
"Sy3_featurecounts" "SY3_1_S1_L001_R1_001_bowtie.sam.bam" "SY3_2_S2_L001_R1_001_bowtie.sam.bam" "SY3_3_S3_L001_R1_001_bowtie.sam.bam"
"SY3_4_S4_L001_R1_001_bowtie.sam.bam"
ko:K01181 OKNHPEEE_00001        gnl|X|OKNHPEEE_1        726     2183    -       1458    1552    1358    1134    1385
ko:K16247 OKNHPEEE_00002        gnl|X|OKNHPEEE_1        2556    2990    +       435     323     282     188     191
OKNHPEEE_00003 gnl|X|OKNHPEEE_1        3369    4043    -       675     577     444     398     468
OKNHPEEE_00004 gnl|X|OKNHPEEE_1        4112    5578    -       1467    1594    1817    1266    1328
ko:K21397 OKNHPEEE_00005        gnl|X|OKNHPEEE_1        6130    7068    -       939     535     460     456     435
ko:K01990,ko:K21397 OKNHPEEE_00006      gnl|X|OKNHPEEE_1        7068    8390    -       1323    282     229     219     200
```

For each ko, make a row and add the expression count columns:

```
for j in *K0s; do name=$(echo ${j} | cut -d "_" -f1); echo KO length ${name}_1 ${name}_2 ${name}_3 ${name}_4 > ${name}.col; for i in $(cat
list_of_all_possible_K0s.list); do if grep -wq ${i} ${j}; then awk "/${i}/ {print \"$i\",\$7,\$8,\$9,\$10,\$11}" ${j}; else echo ${i} - x
x x x ; fi >>${name}.col; done ;done
```

```
KO length Sy3_1 Sy3_2 Sy3_3 Sy3_4
K00001 - x x x x
K00003 1203 337 536 298 293
K00003 1245 3083 3245 2265 2676
K00003 270 833 1021 735 867
K00005 - x x x x
K00008 - x x x x
K00009 1455 57 46 43 48
K00012 1233 2248 2322 1813 2073
K00013 579 260 193 169 259
```

Then add up counts if there are multiple expression counts for genes in the same function:

```
for i in *col; do name=$(echo ${i} | cut -d "_" -f1); echo "K0 "${name}"_1 "${name}"_2 "${name}"_3 "${name}"_4" > ${name}_K0.counts; sort
-k1 ${i} | awk 'BEGIN{prevfam=$1;rep1=$3;rep2=$4;rep3=$5;rep4=$6}{if ($1 !=prevfam) {print prevfam, rep1, rep2, rep3, rep4; prevfam=$1;
rep1=$3;rep2=$4; rep3=$5; rep4=$6;} else {rep1=rep1+$3;rep2=rep2+$4; rep3=rep3+$5; rep4=rep4+$6;}} END{print prevfam, rep1, rep2, rep3,
rep4}' >> ${name}_K0.counts; done
```

```
==> Sy3.col_K0.counts <==
K0 Sy3.col_1 Sy3.col_2 Sy3.col_3 Sy3.col_4

K00001 x x x x
K00003 4253 4802 3298 3836
K00005 x x x x
K00008 x x x x
K00009 57 46 43 48
K00012 2248 2322 1813 2073
K00013 666 476 482 534
K00014 288 211 197 203
```

Finally paste together to make the table:

```
paste *.counts > K0_counts_RNA_data.table
```

There are 3025 total KO groups, of which 572 are common to all HAPs and NAPs.

Ranking genes using Transcripts per Kilobase million (TPM):

(https://rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/, accessed: 15/09/2020)

TPM is very similar to Reads per kilobase million and Fragments per kilobase million. The only difference is the order of operations. Here's how you calculate TPM:

1. *Divide the read counts by the length of each gene in kilobases. This gives you reads per kilobase (RPK).*
2. *Count up all the RPK values in a sample and divide this number by 1,000,000. This is your "per million" scaling factor.*
3. *Divide the RPK values by the "per million" scaling factor. This gives you TPM.*

I can just do this in excel.
Then I can also copy over the COG group info and then use the other commands to make the csv files - eventually I want a file with the TPMs or X's if the gene isn't present.

```
for i in *.csv; do sed '1d' ${i} | cut -d "," -f1; done | grep -v "noOG" | tr "," "\n" | sort | uniq  > list_of_bactOGs.list

for j in *.txt; do name=$(echo ${j} | cut -d "_" -f1); echo OG ${name}_1 ${name}_2 ${name}_3 ${name}_4 > ${name}.counts; for i in $(cat
list_of_bactOGs.list); do if grep -wq ${i} ${j}; then awk -F '\t' "/${i}/ {print \"$i\",\$17,\$18,\$19,\$20}" ${j}; else echo ${i}
x,x,x,x; fi >> ${name}.counts; done ;done

for i in *.counts; do head -1 ${i} > ${i}.col; sort -k1 ${i} | awk 'BEGIN{prevfam=$1;rep1=$2;rep2=$3;rep3=$4;rep4=$5}{if ($1 !=prevfam)
{print prevfam, rep1, rep2, rep3, rep4; prevfam=$1; rep1=$2;rep2=$3; rep3=$4; rep4=$5;} else {rep1=rep1+$2;rep2=rep2+$3; rep3=rep3+$4;
rep4=rep4+$5;}} END{print prevfam, rep1, rep2, rep3, rep4}' >> ${i}.col; done

for i in *.col; do sed -i 's/\r//g' ${i}; done

paste *.col | tr " " "\t" > nOG_RPK_counts.tsv
```

Now I can rank all the genes within a species and then look at the top 5% and see what is enriched. (in excel) and also look where those unique genes to the HAPs rank and compare the TPMs:

4 COGs for HAPs

COG0384@2 - isomerase, unknown function (CO)
COG1292@2 - Cell wall/membrane/envelope biogenesis; Belongs to the BCCT transporter (TC 2.A.15) family, M
COG2202@2 - Signal transduction mechanisms; Pas domain, T
COG5505@2 - Protein of unknown function (DUF819), S

2 for HAPs SAPs

COG1063@2 - Amino acid transport and metabolism; alcohol dehydrogenase, E
COG4122@2 - Amino acid transport and metabolism; O-methyltransferase activity, E

COG	identifier	**12662 rank**	**12662 TPM**	**49906 rank**	**49906 TPM**	**isol6 rank**	**isol6 TPM**
COG0384	CO	1214	88.79 +- 10.47	808	190.28±35.52	396	420.99±103.21
COG1292	M	1661	2.51±0.18	1431	33.65±8.72	483	322.6±334.6
COG2202	T	1491	27.4±3.91	462	446.91±42.25	1143	36.62±3.86
COG5505	S	1636	5.2±1.29	671	262.03±49.28	639	221.47±30.37
COG1063	E	1215	88.57±16.42	1568	9.39±2.76	1042	64.56±12.27
COG4122	E	1320	68.86±12.43	1114	99.35±14.73	711	178.5±19.15

I can also do the same with the K0s:

I need to work out the tpm for each gene though, which I have to do using the featurecount files first, that already have the KOs in the first column.
I opened all of these featurecount_KOs in one excel spreadsheet and calculated all the TPMs then exported these to one txt each called *_featurecount_TPMs_K0s.
Then I can do this:

```
for j in *featurecount_TPMs_K0s.txt; do name=$(echo ${j} | cut -d "_" -f1); echo K0 ${name}_TPM1 ${name}_TPM2 ${name}_TPM3 ${name}_TPM4 >
${name}.TPM.counts; for i in $(cat list_of_all_possible_K0s.list); do if grep -wq ${i} ${j}; then awk -F '\t' "/${i}/ {print
\"$i\",\$16,\$17,\$18,\$19}" ${j}; else echo ${i} x x x x; fi >> ${name}.TPM.counts; done ;done

for i in *TPM.counts; do head -1 ${i} > ${i}.col; sort -k1 ${i} | awk 'BEGIN{prevfam=$1;rep1=$2;rep2=$3;rep3=$4;rep4=$5}{if ($1 !=prevfam)
{print prevfam, rep1, rep2, rep3, rep4; prevfam=$1; rep1=$2;rep2=$3; rep3=$4; rep4=$5;} else {rep1=rep1+$2;rep2=rep2+$3; rep3=rep3+$4;
rep4=rep4+$5;}} END{print prevfam, rep1, rep2, rep3, rep4}' >> ${i}.col; done

for i in *TPM.col; do sed -i 's/\r//g' ${i}; done

paste *TPM.col | tr " " "\t" > K0_TPM_counts.tsv
```

<u>Transporters in the transcriptomes:</u>

I need to get it in a format of:
amino acid TC code gene name TPM counthead *tpm -> using the TPM count flles

```
for i in *_tophits; do name=$(echo ${i} | cut -d "." -f1); grep -f aatransporters.list ${i} | awk '{print $1, $2}' | cut -d "|" -f1,4 |
sed 's/gnl|//g' | cut -d "_" -f1,2 > ${name}_aatransporter_genes.list; done
cat *genes.list > aatransporter_genes.grep
```

copied all the .list files from HPC to local directory.
then added all of the transporter codes to the front of genes - any genes, any transporters, using the featurecounts files.

```
for i in *tpm.txt; do head -1 ${i} | sed 's/Geneid/tccode/' > ${i}_aatransporters; awk 'BEGIN{while(getline < "aatransporter_genes.grep")
array[$1]=$2} {if ($1 in array) print array[$1], $2,$3,$4,$5}' ${i}>> ${i}_aatransporters; done

for i in *aatransporters; do head -1 ${i} > ${i}_aatransporters.counts; for j in $(cat aatransporters.list); do if grep -q ${j} ${i}; then
awk "/${j}/ {print \"$j\",\$2,\$3,\$4,\$5}" ${i}; else echo ${j} 0 0 0 0; fi >>${i}_aatransporters.counts; done ;done
```

Now I just need to add them up if there are multiple counts of the same transporter code.

```
for i in *counts; do head -1 ${i} > ${i}.col; sed '1d' ${i} | sort -k1 | awk 'BEGIN{prevfam=$1;rep1=$2;rep2=$3;rep3=$4;rep4=$5}{if ($1
!=prevfam) {print prevfam, rep1, rep2, rep3, rep4; prevfam=$1; rep1=$2;rep2=$3; rep3=$4; rep4=$5;} else {rep1=rep1+$2;rep2=rep2+$3;
rep3=rep3+$4; rep4=rep4+$5;} } END{print prevfam, rep1, rep2, rep3, rep4}' >> ${i}.col; done
```

then paste together:

```
paste -d " " *.col > tpm_aatransporters.tsv
```

Then open the tsv in excel, change the text to columns using a space, then remove all the excess codes and rearrange the species and transpose.
Then make a graph in R using the script transcriptomics_aatransporters.R: