Tetrahymena Pipeline

Reference Locations:

References
Reference genome of Tlr location:
/storage/datasets/Tetrahymena_thermophila/analyses/virus/data

Mac reference location: /storage/reference_genomes/tetrahymena_thermophila/mac/mac.genome.fasta

Mic reference location:
/storage/reference_genomes/tetrahymena_thermophila/mic/mic.genome.fasta

Illumina:
Sample location: /storage/datasets/Tetrahymena_thermophila/ancestor_ges/fastq

Pacbio:
Pacbio location: /storage/datasets/Tetrahymena_thermophila/SRX2635099/fastq

Combing R1 and R2 for all 40 ancestors: (MAKE BASH SCRIPT IN FOLDER)
Found in: /work/jgjohns6/Tetrahymena/illumina/scripts/Ancestors_R1.sh (and Ancestors_R2.sh)

```
folder='/storage/datasets/Tetrahymena_thermophila/ancestor_ges/fastq/Anc*'
DIR='Ancestors'
mkdir -p $DIR
for subfolder in $folder;
do
oldfilename=${subfolder##*/}
oldfilename=${oldfilename}.R2.fastq
#echo $subfolder "$DIR/$subfolder_R2.fastq";
find $subfolder -name "*R2*" -exec cat {} \; >> "$DIR/$oldfilename";
Done
```

Make ultra-reference using cat command on Mic + Mac + Tlr references and create reference index using bowtie2 build
Ultra-reference found in: /work/jgjohns6/Tetrahymena/illumina/ultra_reference
Referenced from Bowtie2 Manual: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-build-indexer

Aligning Ancestors R1 and R2 individually using bowtie2, reference bash script, ultra_RG.sh:
Found in: /work/jgjohns6/Tetrahymena/illumina/scripts/bowtie2_Anc.sh

```
bowtie2 --rg-id Anc_10_A -x ultra_reference.index -1 /home/jgjohns6/Ancestors/Anc_10_A-32273406.R1.fastq -2 /home/jgjohns6/Ancestors/Anc_10_A-32273406.R2.fastq -S Anc_10_A_ultra.sam
```

Run full script in parallel using:

```
Parallel -j 20 < bowtie2.sh
```

Merging all 40 ancestor files into one file, Anc_ultra.sam:
Found in /work/jgjohns6/Tetrahymena/illumina/Anc_ultra/Anc_ultra.sam

```
samtools merge Anc_ultra.sam *.sam
```

Convert Sam file to Bam file:
Found in /work/jgjohns6/Tetrahymena/illumine/Anc_ultra/Anc_ultra.bam

```
samtools view -b -o Anc_ultra.bam Anc_ultra.sam
```

Filter on SAM flag 3852 and mapping quality 10 to create filtered bam file:

SAM flag 3852 filters read unmapped, mate unmapped, not primary alignment, read fails platform/vendor quality checks, read is PCR or optical duplicate, and supplementary alignment. Used this website as reference: https://broadinstitute.github.io/picard/explain-flags.html

```
samtools view -h -F3852 -q10 Anc_RG_ultra.bam > Anc_RG_ultra_3852_10.bam
```

Sort bam file

```
Samtools sort Anc_ultra_3852_10.bam > Anc_ultra_3852_10_sorted.bam
```

Create an Index for the bam in order to run the filter:

```
samtools index Anc_RG_ultra.bam Anc_RG_ultra.bai
```

Filter on AF45 (Tlr chromosome)

```
samtools view -h -b Anc_RG_ultra.bam AF451863.1 > Anc_RG_ultra_AF45v2.bam
```

Count Plot:

SAM flag is not actually needed to make count plot, yet is shown in these scripts, count plot is made from RG and header

Convert Bam file back into Sam

Create text file of RG and SAM flag columns using the filtered bam

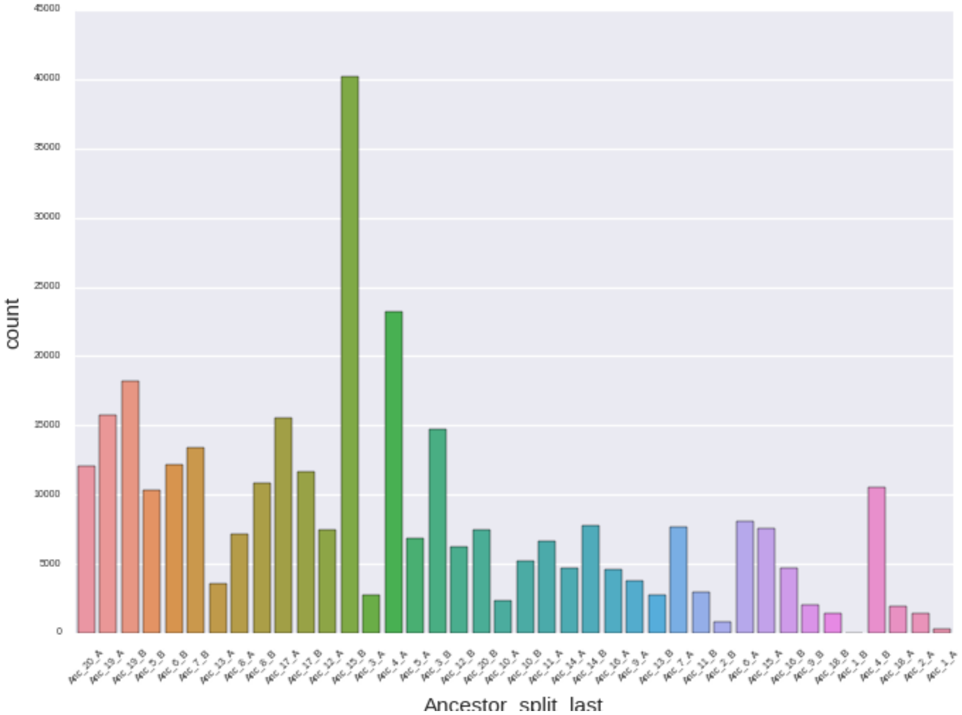
```
awk '{for(i=1;i<=NF;i++) if(i==2 || $i~"RG") printf $i" ";print ""}' Anc_ultra_3852_10.sam >
Anc_RG_ultra_3852.txt
```

Create a png file on the count plot using the text file

```
import pandas as pd
import matplotlib
matplotlib.use('Agg')
from matplotlib import pyplot as plt
import seaborn as sns
df = pd.read_csv('Anc_ultra_3852_10_AF45.txt',sep=" ",index_col=False, header=None,
names=['Ancestor'], engine=
'python')

#df = df.iloc[4:]
#df.head()
#df['Ancestor']

df = df[~df['Ancestor'].astype(str).str.startswith('VN')]
df = df[~df['Ancestor'].astype(str).str.startswith('SN')]
df = df[~df['Ancestor'].astype(str).str.startswith('ID')]
df = df[~df['Ancestor'].astype(str).str.startswith('ID')]
df = df[df['Ancestor'].astype(str).str.startswith('RG')]


df['Ancestor_split'] = df['Ancestor'].str.split(':')
```

```
df['Ancestor_split_last'] = df['Ancestor_split'].str[-1]
#print(df)
#df2 = df.iloc[10:]
x = df['Ancestor_split_last'].value_counts()
#print(x)
my_figure = sns.countplot(x='Ancestor_split_last', hue=None, data=df) #no y input needed, counts
occurances for y axis
plt.xticks(rotation=45)
my_figure.tick_params(labelsize=5)
#my_figure.savefig('Ancestorplot.png')
fig = my_figure.get_figure()
fig.savefig('Ancestorplot_ultra_AF39_3852_3.png')
```

Must download png to home computer to view, hines does not allowing viewing of images



Coverage Histogram

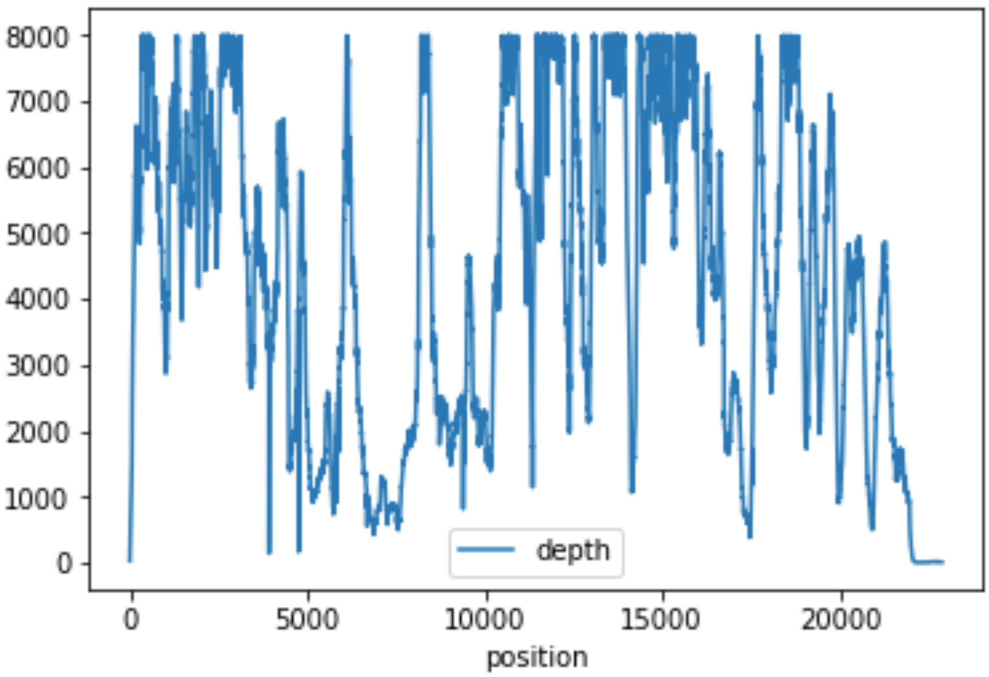
Create coverage file of filtered bam

```
samtools depth deduped_MA605.bam > deduped_MA605.coverage
```

Plot Histogram in python

```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
import numpy as np
df=pd.read_csv("ancestoralign_sorted.coverage",header=None, sep="\t", names=['contig', 'position',
'depth'])
print(df)
df.plot(x='position', y='depth')
```

```
plt.show()
```


8000
7000
6000
5000
4000
3000
2000
1000
0
0
5000
10000
15000
20000
position
depth