Curation of the *Callithrix Penicillata* draft Genome

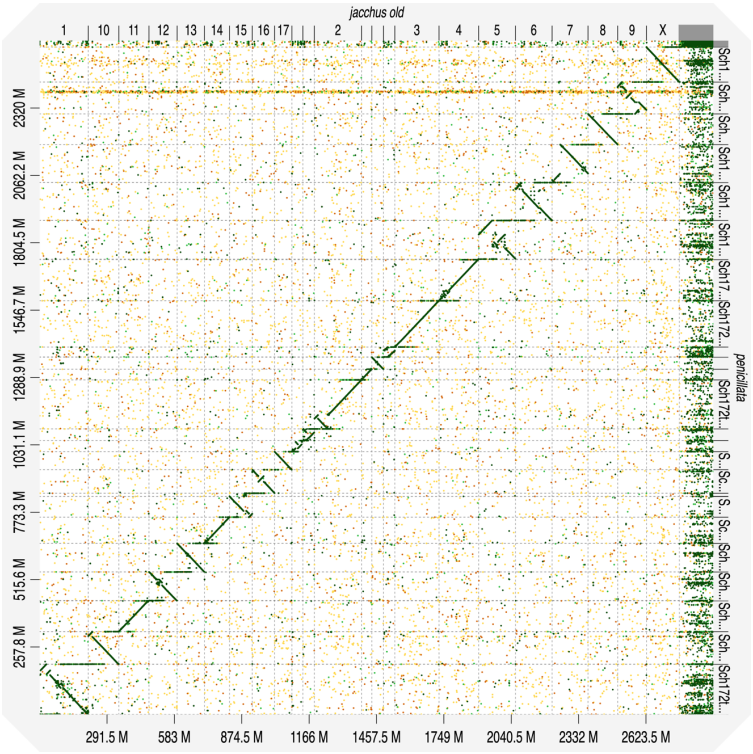
For organizing and labelling chromosomes for *Callithrix penicillata jj1*:

Run Dgenies.sh to run minimap on target and query file and to also index for input of D-genies.

Script: /thesis/scripts/Dgenies.sh

```
HS_REF=/storage/reference_genomes/human/1k_genomes_GRCh38/GRCh38_full_analysis_set_plus_decoy_hla.fa
CJ_REF=/storage/reference_genomes/callithrix/jacchus/Callithrix_jacchus.ASM275486v1.dna.nonchromosomal.fa
CP_REF=../callithrix_penicillata_dt1.fasta

/home/jgjohns6/minimap2-2.12_x64-linux/minimap2 -t 20 -x asm20 $(HS_REF) $(CP_REF) > $@

/home/jgjohns6/minimap2-2.12_x64-linux/minimap2 -t 20 -x asm10 $(CJ_REF) $(CP_REF) > $@

echo "Callithrix penicillata" > $@
cut -f 1,2 $(CP_REF).fai >> $@


echo "Callithrix jacchus" > $@
cut -f 1,2 $(CJ_REF).fai >> $@


echo "Homo sapiens" > $@
cut -f 1,2 $(HS_REF).fai >> $@
```

Run D-genies to get dot plot. Below is the graph of the 2014 *callithrix jacchus* versus the dovetail2 original *callithrix penicillata* genome.



R script for renaming and sorting chromosomes based on association table gained from D-genies graph. Chromosomes are sorted by chromosomes 1-22 then X, then by length of scaffold. Scaffolds are named as scaffold_md5sum using only the first 8 characters of md5sum.
R script is named: thesis/scripts/chr_labelling.R

```
# DGenies association table with penicillata as query and old jacchus as target
assoc = "Callithrix_penicillata_jacchus_old_assoc.tsv"
# Input Fasta file
dt1 = "callithrix_penicillata_dt1.fasta"
# Output Fasta File
output = "callithrix_penicillata_jj1.fasta"
# Sequence Dictionary
dict = "callithrix_penicillata_dt1.dict"
```

```
library(tidyr)
library(gtools)
library(readr)
library(dplyr)
library(stringr)

# read the sequence dictionary, modify the name, length, and md5 columns
# sort by longest scaffolds first.
dat_raw = read_tsv(dict,col_names=FALSE,skip=1)
dat = dat_raw %>% transmute(Name = str_sub(X2,4), Length = as.integer(str_sub(X3,4)), Id =
str_c("scaffold_",str_sub(X4,4,4+8-1))) %>%
                arrange(desc(Length))

# QC: Check if the first 8 characters of each MD5 are unique
# length(unique(dat$Id)) == length(dat$Id)

# Read association table and sort by Query length
assoc_dat = read_tsv(assoc,col_types="ccciiiiii",na="na") %>%
          arrange(desc(`Q-len`))

# QC: Check that the first 25 are sorted the same in both tables
all(assoc_dat$Query[1:25] == dat$Name[1:25])

# The longest 25 scaffolds consist of Chrs 1-22, X, and pieces of Chrs 4 and 15.
# Everything below that has a length less that 500kb and starts matching various
# scaffolds.

# rename the chromosome
dat$Id[1:23] = str_c("chr",assoc_dat$Target[1:23])

# rename the two chromosome fragments
```

```
dat$Id[24:25] = str_c("chr",assoc_dat$Target[24:25],"un_",dat$Id[24:25])

# Resort the chromosomes by name
dat[1:23,] = dat[mixedorder(assoc_dat$Target[1:23]),]

# Save mapping dat
write_tsv(dat,"dt1_new_names.tsv")

################################################################################
#
# Read fasta and sort
library(Biostrings)

fasta = readDNAStringSet(dt1)
new_fasta = fasta[dat$Name]
#QC: all(names(new_fasta) == dat$Name)

# rename fasta
names(new_fasta) = dat$Id

# write it out
writeXStringSet(new_fasta, output)
```

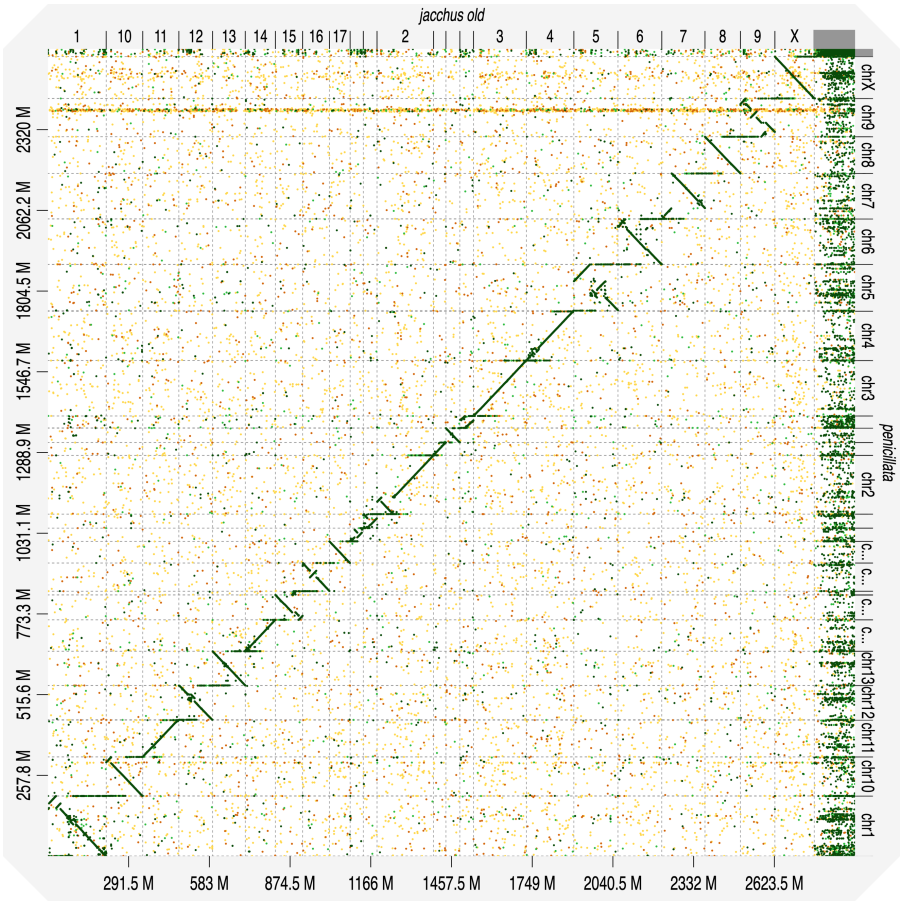
Creating the dictionary file of jj1 fasta to compare against old fasta to ensure no changes were made to sequences, just to order and names:

```
picard CreateSequenceDictionary R=callithrix_penicillata_jj1.fasta
O=Callithrix_penicillata_jj1.dict
```

General script for comparing dictionary files:

```
Cut -f 4 file.dict | sort >/tmp/jj1.dict
Cut -f 4 file2.dict | sort > /tmp/dt1.dict
Diff -u /tmp/jj1.dict /tmp/dt1.dict
```

Run Dgenies.sh with jj1 *Callitrhix penicillata* to use for D-genies against the 2014 *callithrix jacchus*.



Reverse complimenting to create *Callithrix penicillata* jj2 version:

2. Run R script that determines positive and negative hits, or which chromosomes should be reverse complemented on jj1. Anything that was below 0.5 was reversed complimented.

Script = thesis/scripts/determine_reverse_compliment.R

paf = "jacchus_old_penicillatajj2.paf"

```
library(tidyr)
library(gtools)
library(readr)
library(dplyr)
library(stringr)

dat = read_tsv(paf,col_names=FALSE)

res = dat %>% filter(str_detect(X1,"^chr") & X1 == str_c("chr",X6)) %>%
   group_by(X1,X6,X5) %>% summarize(s = sum(X11))

strand = res %>% spread(X5,s) %>% mutate(pos=`+`/(`+`+`-`)) %>%
   arrange(pos)
```

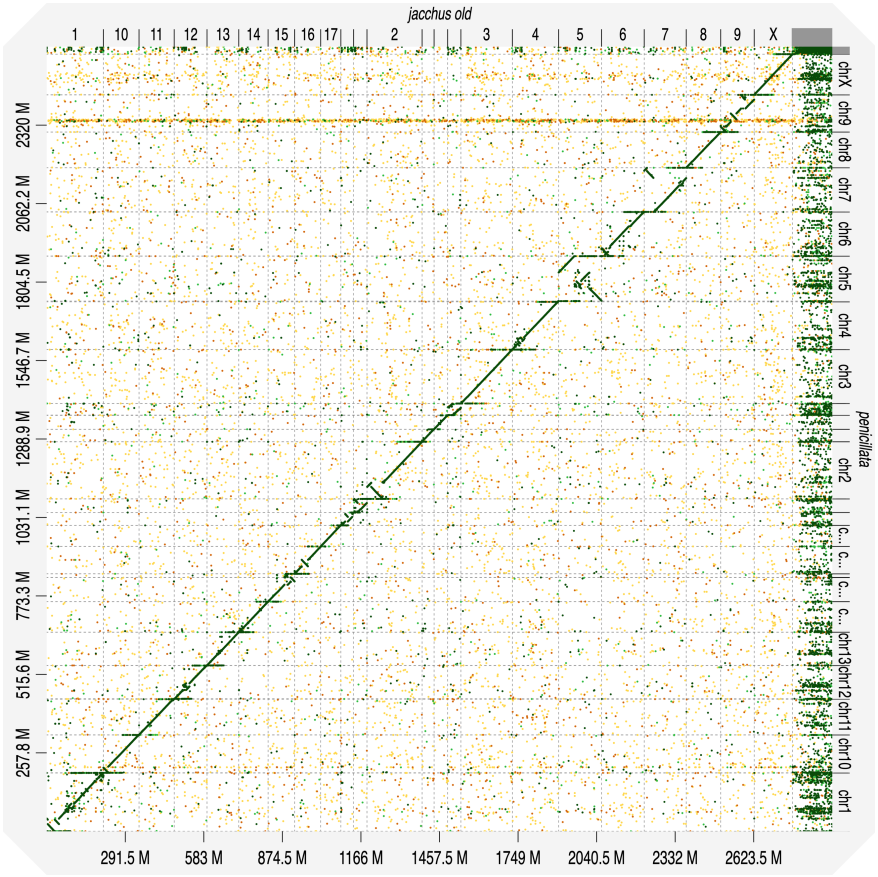
(3) Run R script to reverse complement the chromosomes to be switched:

```
library(Biostrings)
dna = readDNAStringSet("callithrix_penicillata_jj1.fasta")
chr = c("chr17", "chr8", "chrX", "chr13", "chr12", "chr21", "chr6", "chr7", "chr15", "chr16",
"chr1", "chr9", "chr10")

for (s in chr) {
  dna[s] = reverseComplement(dna[s])
}

writeXStringSet(dna, "callithrix_penicillata_jj2.fasta")
```

Create dict file for jj2 and compare it against jj1 same as above.

Create paf file using minimap2 and rerun the script (2) to ensure the chromosomes reversed.

Create new D-genies graph with jj2 against the old jacchus genome.



Referee Quality score program:

Run minimap2 against dovetail2 fastq files.

/home/jgjohns6/minimap2-2.12_x64-linux/minimap2 -ax sr -t10 -2
/work/jgjohns6/penicillata/callithrix_penicillata_jj2.fasta
/storage/reference_genomes/callithrix/penicillata/Denovo/DTG-SG-156_R1_001.fastq.gz
/storage/reference_genomes/callithrix/penicillata/Denovo/DTG-SG-156_R2_001.fastq.gz >
jj2_DT1_DT2.sam

Create Bam file to Sam file:

samtools view -S -b jj2_DT1_DT2.sam > jj2_DT1_DT2.bam

Sort Bam file:

samtools sort jj2_DT1_DT2.bam -o jj2_DT1_DT2_sorted.bam

Create mpilup file to use for referee:

samtools mpileup -f callithrix_penicillata_jj2.fasta jj2_DT1_DT2_sorted.bam >
jj2_DT1_DT2.mpileup

zipmpileup for use of tabix:

bgzip jj2_CT1_CT2.mpileup

index mpileup for use of tabix:

tabix -s1 -b2 -e2 jj2_DT1_DT2.mpileup.gz

Split up mpileup file by chromosome to use for referee:

Script: /thesis/scripts/tabix.sh

tabix jj2_DT1_DT2.mpileup.gz chr2 > jj2_DT1_DT2_chr2.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr3 > jj2_DT1_DT2_chr3.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr4 > jj2_DT1_DT2_chr4.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr5 > jj2_DT1_DT2_chr5.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr6 > jj2_DT1_DT2_chr6.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr7 > jj2_DT1_DT2_chr7.mpileup

tabix jj2_DT1_DT2.mpileup.gz chr8 > jj2_DT1_DT2_chr8.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr9 > jj2_DT1_DT2_chr9.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr10 > jj2_DT1_DT2_chr10.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr11 > jj2_DT1_DT2_chr11.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr12 > jj2_DT1_DT2_chr12.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr13 > jj2_DT1_DT2_chr13.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr14 > jj2_DT1_DT2_chr14.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr15 > jj2_DT1_DT2_chr15.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr16 > jj2_DT1_DT2_chr16.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr17 > jj2_DT1_DT2_chr17.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr18 > jj2_DT1_DT2_chr18.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr19 > jj2_DT1_DT2_chr19.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr20 > jj2_DT1_DT2_chr20.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr21 > jj2_DT1_DT2_chr21.mpileup
tabix jj2_DT1_DT2.mpileup.gz chr22 > jj2_DT1_DT2_chr22.mpileup
tabix jj2_DT1_DT2.mpileup.gz chrX > jj2_DT1_DT2_chrX.mpileup

Running referee on the mpileup files in parallal

parallel python /home/jgjohns6/penicillata/referee/referee.py --pileup -p10 -gl {} -ref

callithrix_penicillata_jj2.fasta ::: *chr*.mpileup

Finding counts of each individual chromosome file to run in parallel

Script= /thesis/scripts/count.sh

awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51aIPnqp.txt | sort | uniq -c | sort -nr > referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51AlRmID.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51BPpgXd.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51cCyJoY.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51FrKkpc.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51FXmTbz.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51HvkwEl.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51ibrQKU.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51ieclpa.txt | sort | uniq -c | sort -nr >> referee_test_counts_2.txt

```
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51JCzuFu.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51MIgYvh.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51MkhmrK.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51odXLnp.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51qDcMrd.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51QIBRHu.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51QYvanm.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51rWDkTn.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51sOPzpU.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51syUcIM.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51wPTjiH.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51xyJMHT.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-25-2019.01-51-51ZsiMks.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
awk -F '\t' '{print $3}' referee-out-09-27-2019.11-42-33oqACbe.txt | sort | uniq -c | sort -nr >>
referee_test_counts_2.txt
```

*Note: referee_test_counts_2.txt is now referee_finalcounts.txt

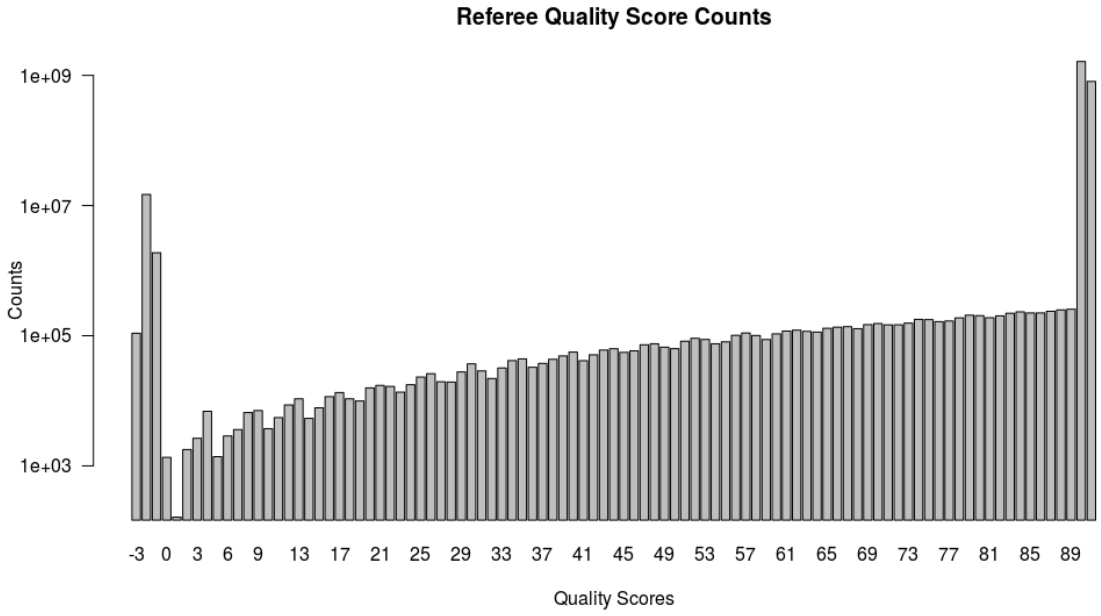
Taking out extra spaces at beginning of the line to make into dataframe:

```
cat referee_test_counts_2.txt | sed -e 's/^[ \t]*//' > referee_test_counts_3.txt
```

R script to create barplot of log of counts

Script = /thesis/scripts/referee_barplot.R

```
data=read.csv("../referee_test_counts_3.txt",header=FALSE, sep=" ")
dt2=data[order(data$V2),]
dt3=aggregate(.  ~ V2, dt2, sum)

barplot(dt3$V1, log = 'y', main = "Referee Quality Score Counts",
      xlab = "Quality Scores", ylab = "Counts", names.arg = dt3$V2, las=1)
```



Comparing Mapping Quality Scores between 2017 *Callithrix jacchus* and *Callithrix penicillata* jj2

Subsampling random 1,000,000 reads from original fastq files

/work/jgjohns6/penicillata/seqtk/seqtk sample -s100

/storage/reference_genomes/callithrix/penicillata/Denovo/DTG-SG-156_R2_001.fastq.gz

1000000 > sub_DTG_2.fq

Running minimap2 to compare new jacchus vs jj2 reads:

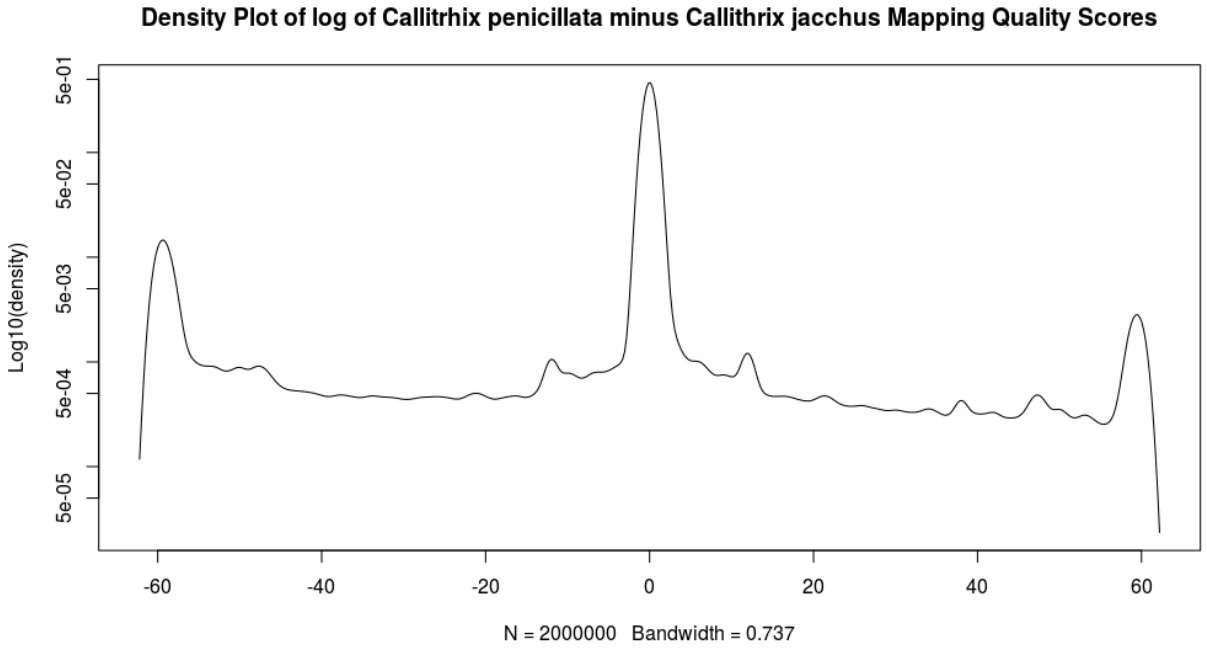
/home/jgjohns6/minimap2-2.12_x64-linux/minimap2 -ax sr -t10 -2
/work/jgjohns6/penicillata/Callithrix_jacchus.ASM275486v1.dna.nonchromosomal.fa
sub_DTG_1.fq sub_DTG_2.fq > jj2_
sub_DT1_DT2.sam

Making density plot to compare the mapping quality scores:

Script: /thesis/scripts/density.R

```
MQ = read.csv("../jj2_sub_DT1_DT2_jj2_135.txt", header =FALSE, sep ="\t")
MQjac = read.csv("../jj2_sub_DT1_DT2_new_jac_135.txt", header = FALSE, sep = "\t")
```

```
MQ2= (MQ$V3 - MQjac$V3)
den=density(MQ2)
plot(den, log ='y', main = "Density Plot of log of Callitrhix penicillata minus Callithrix jacchus Mapping Quality Scores", ylab = "Log10(density)")
```



Using denovo gear to calculate heterzygosity:

bcftools mpileup -a 'AD,DP' -T sites_to_mutate_jj2.bed -q3 -Q13 -Ou -f callithrix_penicillata_jj2.fasta jj2_DT1_DT2_sorted.bam > bcftools.pileup.txt

bcftools call bcftools.pileup.txt -m -A -p 0 -P 0 -O z -o bcftools_jj2_2.vcf.gz

Rscript optimize_parameters.R dng.txt

Where dng.txt is:

dng loglike --input=bcftools_jj2_2.vcf.gz --ped=jj2.ped

jj2.ped is:

##PEDNG v1.0
#Individual Father Mother Sex Samples
jj2_DT1_DT2_sorted.bam . . female =

Output of denovo:

theta lib-error ref-bias-hom ref-bias-het
0.0034137405 0.0005070511 1.0000000000 0.9999999710
lib-overdisp-hom lib-overdisp-het
0.0018241117 0.2517488035