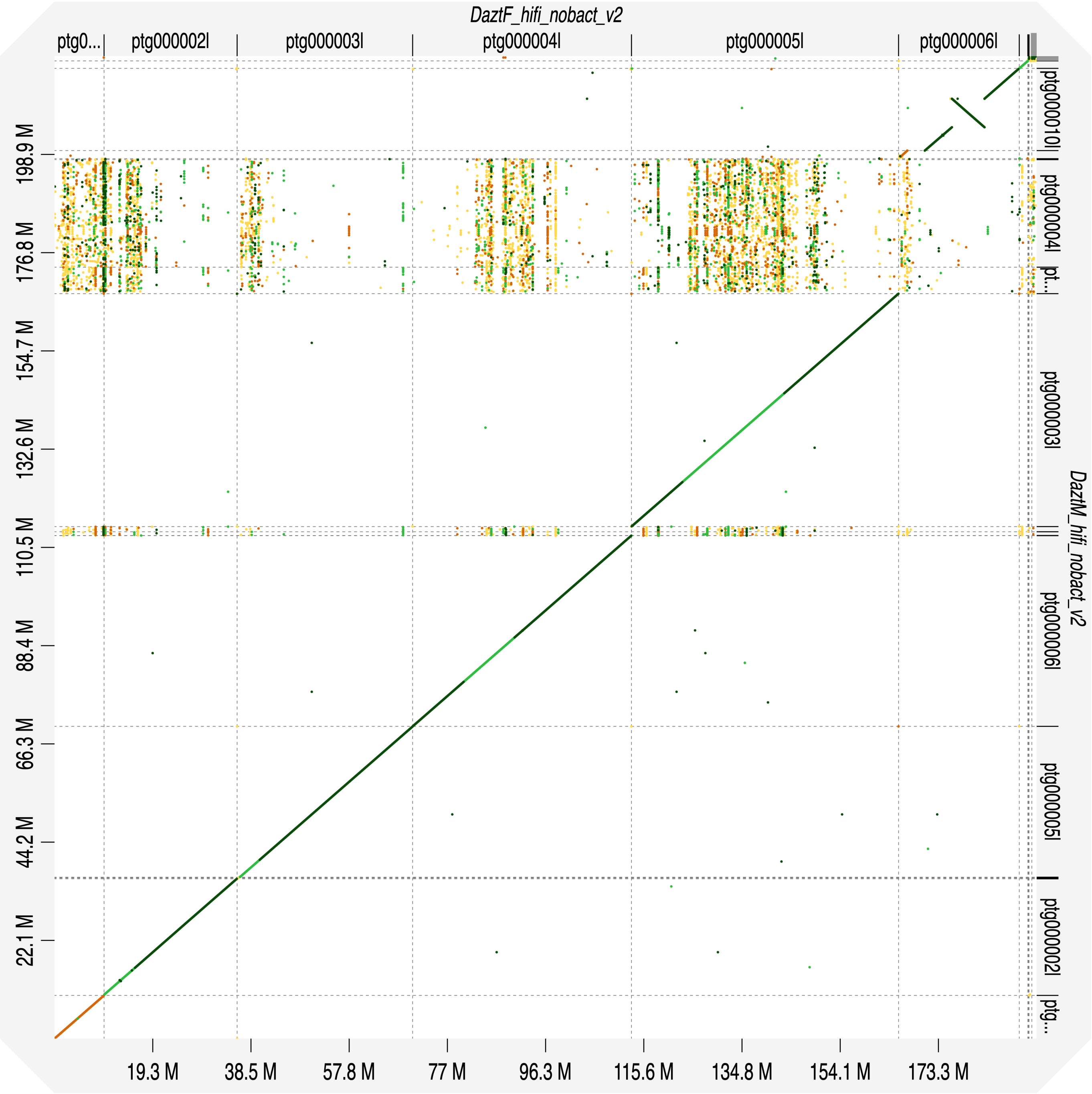
D. azteca

Assembly



quast, busco, and other data I will undoubtedly be asked for...

Species (sex)	Length	# of Scaffolds	N50	BUSCO (complete)	BUSCO (single)	BUSCO (dup)
D. azteca v2 (male)	~221Mb	38	33.9Mb	97.5%	96.8%	0.7%
D. azteca v2 (female)	~192Mb	30	34.4Mb	99.3%	98.7%	0.6%

Coverage

Method 1

Make .bam files with minimap and plot coverage using chromosome quotient method

Code:

```
# on the linux
# align female longreads to male reference
minimap2 -t 8 -ax map-hifi /hdd/Taylor/data/DaztM_hifi_nobact_v2.fa
/hdd/Taylor/data/D-azteca_F_HiFi.fastq.gz |samtools view -bS >
/hdd/Taylor/data/DaztF_hifi.bam

# align male longreads to male reference
minimap2 -t 8 -ax map-hifi /hdd/Taylor/data/DaztM_hifi_nobact_v2.fa
/hdd/Taylor/data/D-azteca_M_HiFi.fastq.gz |samtools view -bS >
/hdd/Taylor/data/DaztM_hifi.bam

# sort bams and get coverage
samtools sort DaztM_hifi.bam >DaztM_hifi_sorted.bam
samtools coverage DaztM_hifi_sorted.bam >DaztM_hifi.cov
samtools sort DaztF_hifi.bam >DaztF_hifi_sorted.bam
samtools coverage DaztF_hifi_sorted.bam >DaztF_hifi.cov
```

Using RStudio:

```
# Load required libraries
library(ggplot2)
library(cowplot)
library(ggrepel)

# Load data
aztmale <-
read.delim("/Users/conway/Desktop/CurrentWorkingDatasets/DaztM_hifi.cov",
header = TRUE)
aztfem <-
read.delim("/Users/conway/Desktop/CurrentWorkingDatasets/DaztF_hifi.cov",
header = TRUE)

# Rename columns for clarity
colnames(aztmale) <- c("scaffold", "startpos", "endpos", "numreads",
"covbases", "coverage", "meandepth", "meanbaseq", "meanmapq")
colnames(aztfem) <- c("scaffold", "startpos", "endpos", "numreads",
"covbases", "coverage", "meandepth", "meanbaseq", "meanmapq")

# Normalize data
aztmale$normalized <- aztmale$numreads / sum(aztmale$numreads)
aztfem$normalized <- aztfem$numreads / sum(aztfem$numreads)
```

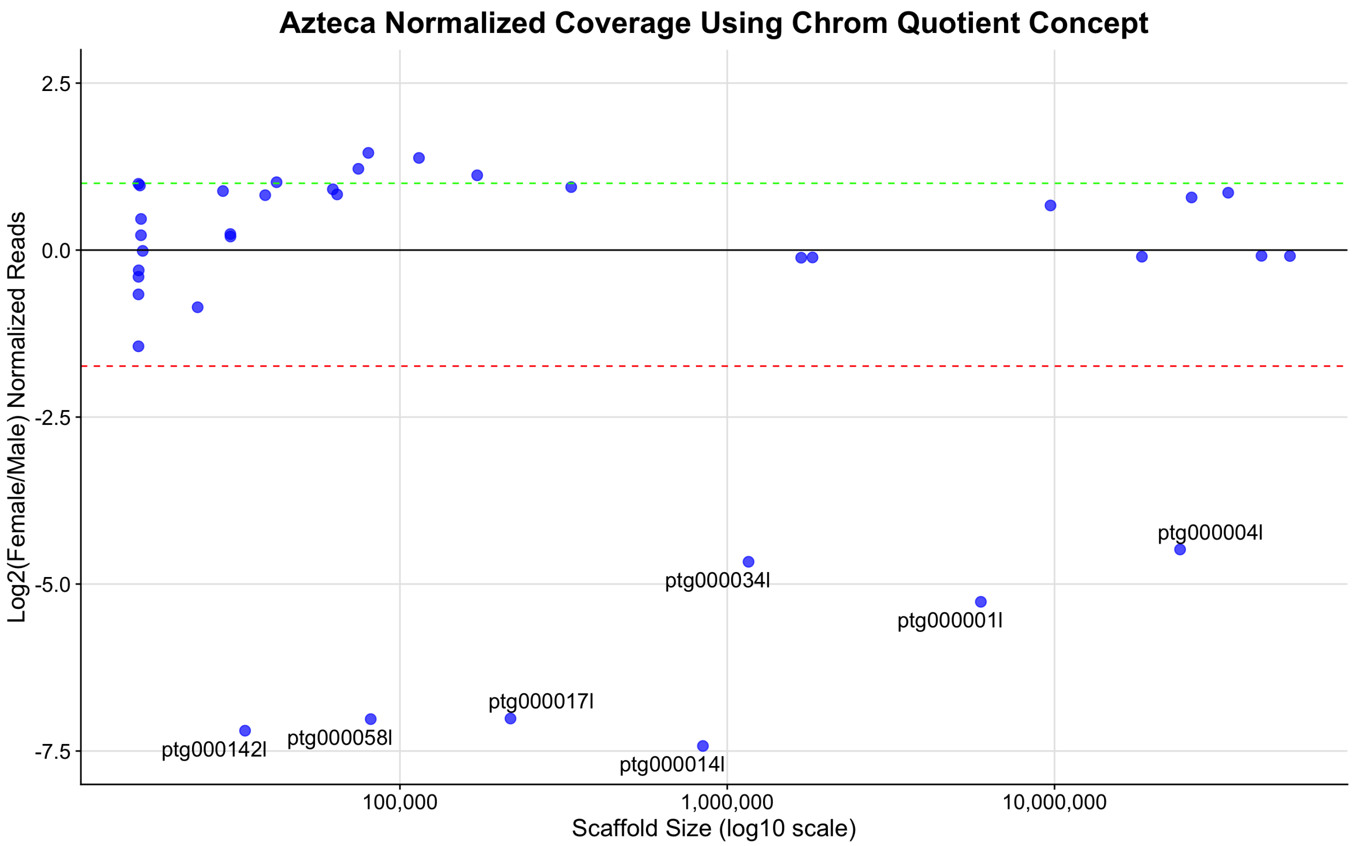
```
# Calculate scaffold sizes
aztmale$size <- aztmale$endpos - aztmale$startpos
aztfem$size <- aztfem$endpos - aztfem$startpos

# Compute log2 ratio of female/male normalized reads
aztmale$log2fem_male <- log2(aztfem$normalized / aztmale$normalized)

# List of scaffolds to highlight
#highlight_scaffolds <- c("ptg000009l", "ptg000018l", "ptg000019l",
#                         "ptg000022l", "ptg000028l", "ptg000029l",
#                         "ptg000038l", "ptg000046l", "ptg000098l")

# Add a column to indicate if a scaffold should be highlighted
#aztmale$highlight <- ifelse(aztmale$scaffold %in% highlight_scaffolds,
"yes", "no")

ggplot(aztmale[aztmale$size > 10000, ], aes(x = size, y = log2fem_male)) +
  # Add points with conditional coloring
  geom_point(aes(color = highlight), size = 3, alpha = 0.7) +
  # Use a log10 scale for the x-axis
  scale_x_continuous(trans = "log10", labels = scales::comma_format()) +
  # Add horizontal reference lines
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  geom_hline(yintercept = log2(0.3), color = "red", linetype = "dashed") +
  geom_hline(yintercept = log2(2), color = "green", linetype = "dashed") +
  # Label only points below the red line
  geom_text_repel(aes(label = ifelse(log2fem_male < log2(0.3), scaffold,
"")),
                  size = 5,
                  box.padding = 0.4,
                  point.padding = 0.4,
                  max.overlaps = Inf) +
  # Customize color scale
  scale_color_manual(values = c("no" = "blue", "yes" = "blue"), guide =
"none") +
  # Axis labels
  labs(
    x = "Scaffold Size (log10 scale)",
    y = "Log2(Female/Male) Normalized Reads",
    title = "Azteca Normalized Coverage Using Chrom Quotient Concept"
  ) +
  # Set y-axis limits
  ylim(-7.5, 2.5) +
  # Apply clean theme
  theme_cowplot(font_size = 14) +
  theme(
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5),  #
Center and bold title
    axis.title = element_text(size = 16),  # Larger axis titles
    axis.text = element_text(size = 14),   # Larger axis text
    panel.grid.major = element_line(color = "grey90", size = 0.5),  #
Subtle grid lines
    legend.position = "none"               # Remove legend
  )
```



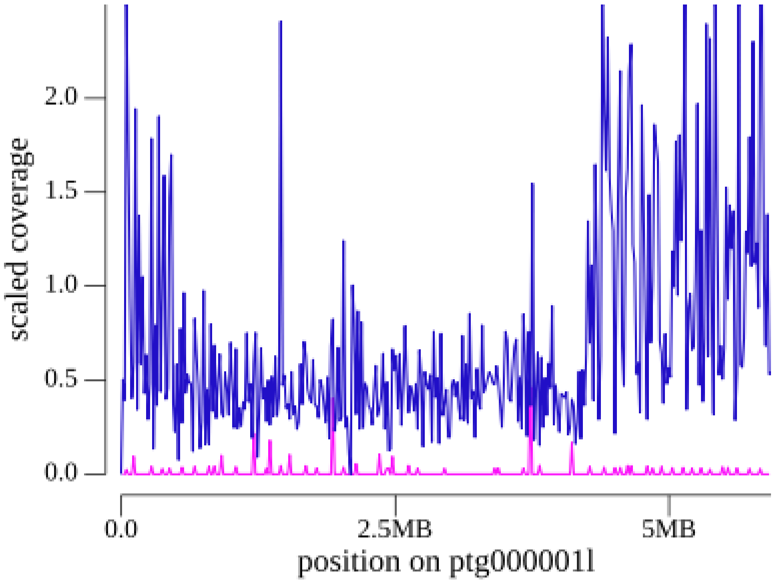
Points aligning around 1 should be X-linked, and points aligning around 0 should be autosomal. Anything under the red line is putative Y-linked. This is because females have 2 Xs when males have 1, and males have 1 Y while females have zero. You expect (when log2 tranformed) for the autosomal reads to cancel out and end up around 0.

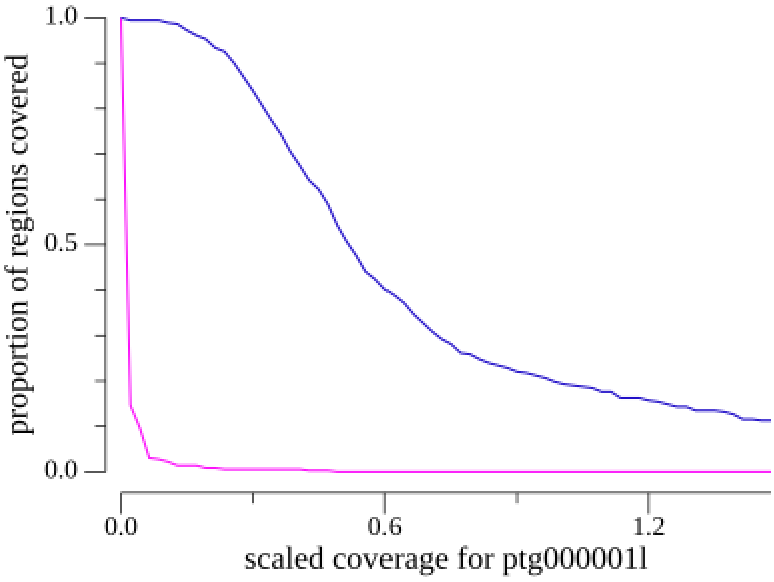
Method 2

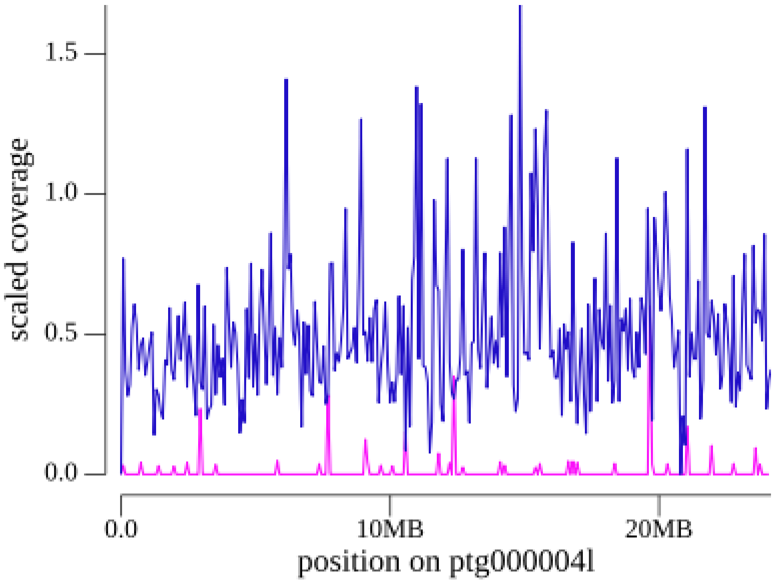
Indexcov

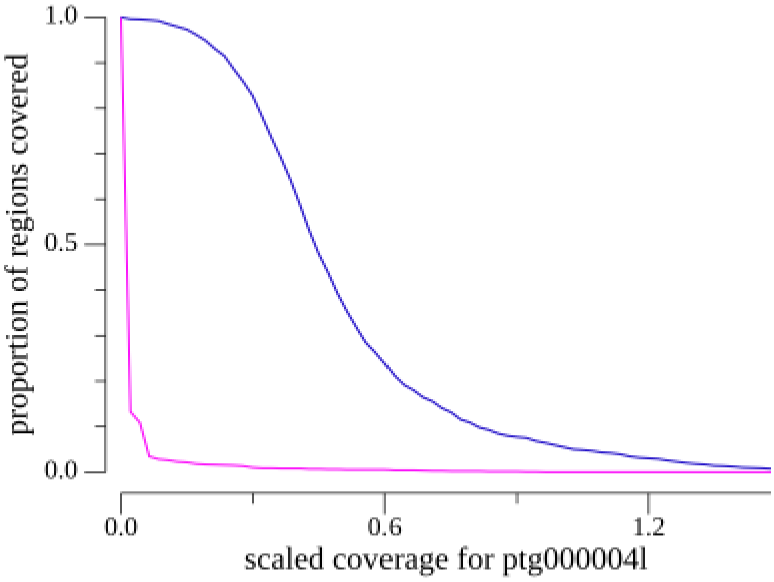
Code:

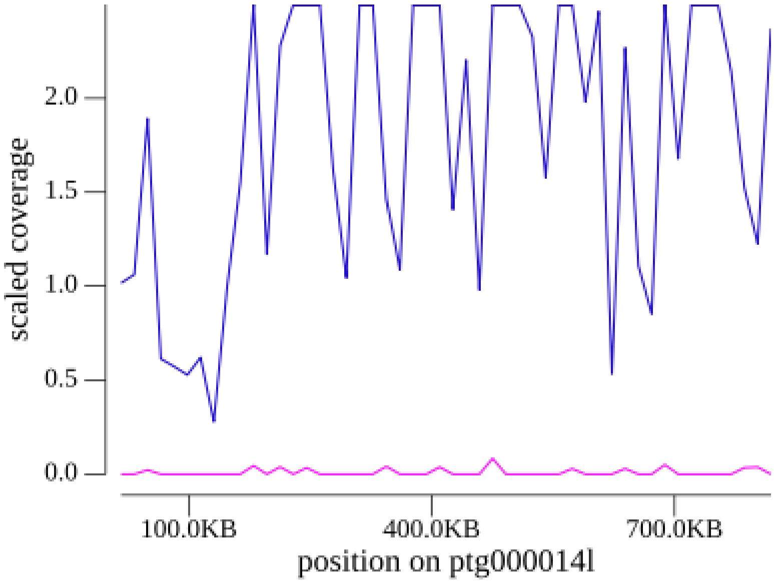
```
goleft indexcov --directory ../indexcov_Dazt *.bam
```

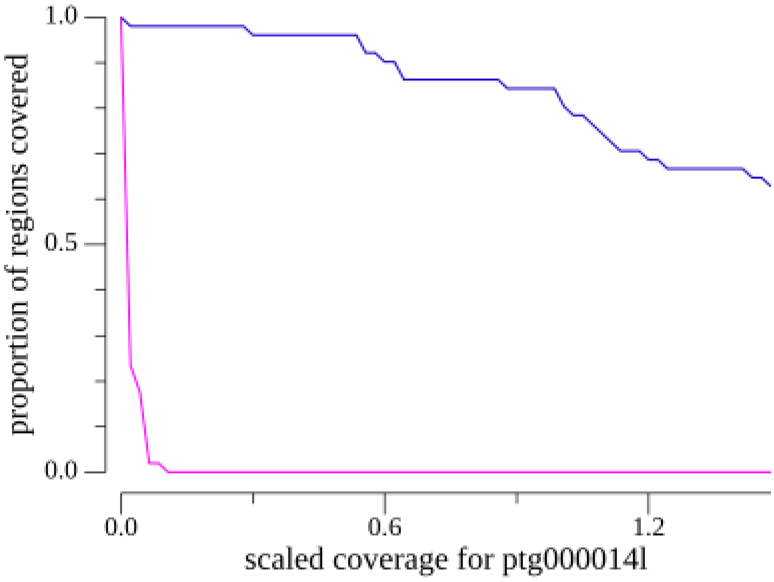

scaled coverage
2.0
1.5
1.0
0.5
0.0
0.0
2.5MB
5MB
position on ptg000001l

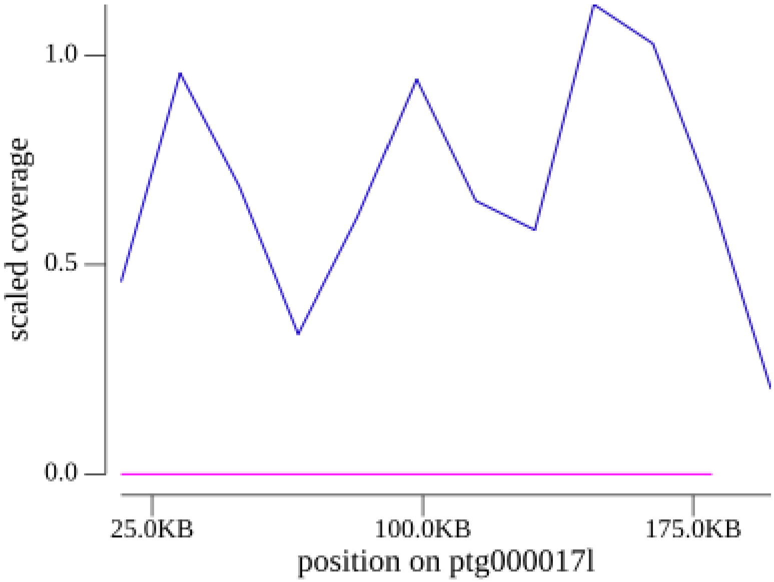

proportion of regions covered
1.0
0.5
0.0
0.0
0.6
1.2
scaled coverage for ptg000001l

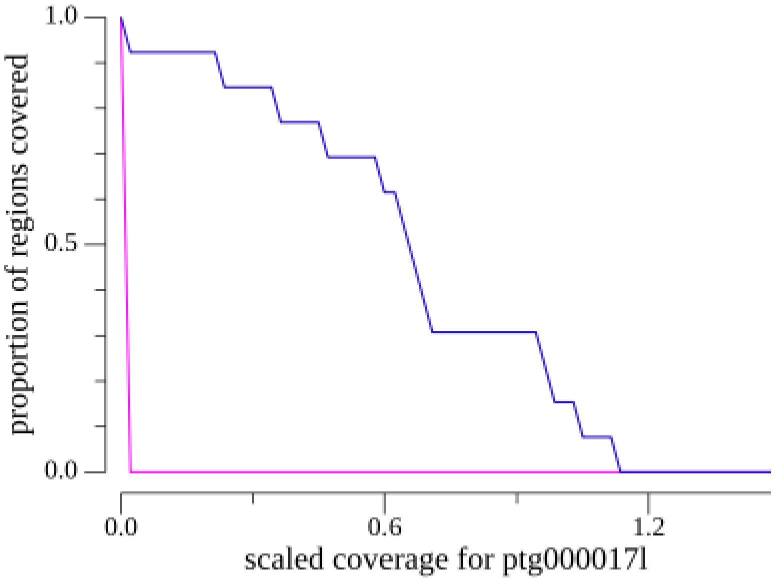

scaled coverage
1.5
1.0
0.5
0.0
0.0
10MB
20MB
position on ptg000004l

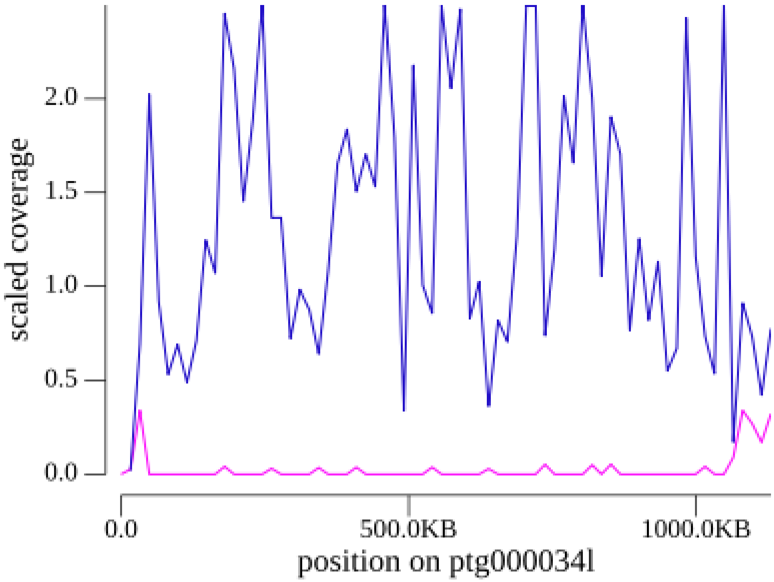

1.0
0.5
0.0
proportion of regions covered
0.0
0.6
1.2
scaled coverage for ptg000004l

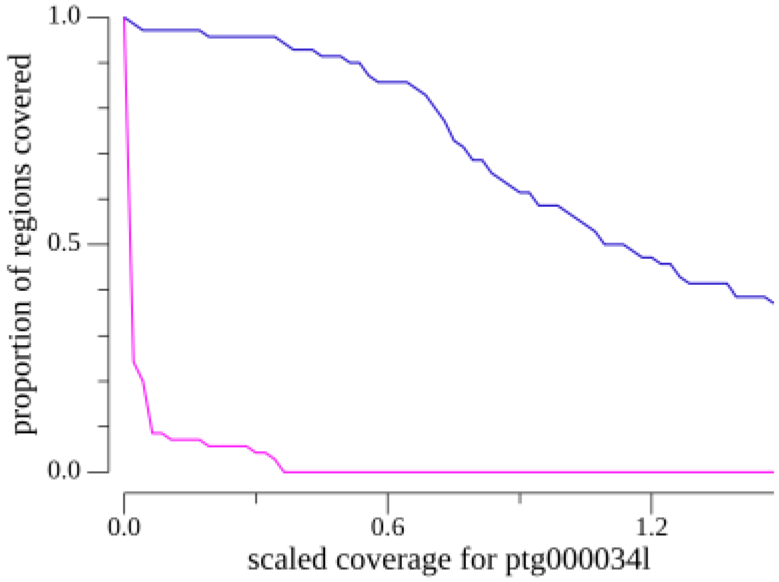

2.0
1.5
1.0
0.5
0.0
scaled coverage
100.0KB
400.0KB
700.0KB
position on ptg000014l

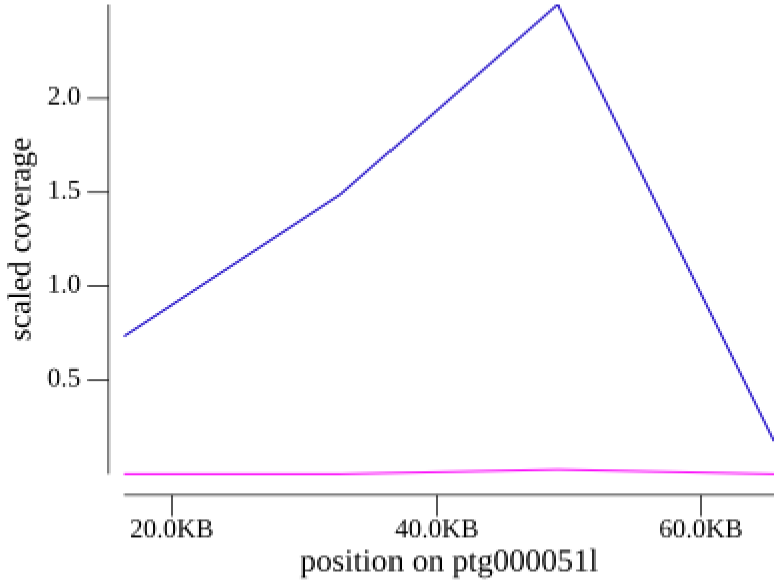

1.0
0.5
0.0
proportion of regions covered
0.0
0.6
1.2
scaled coverage for ptg000014l

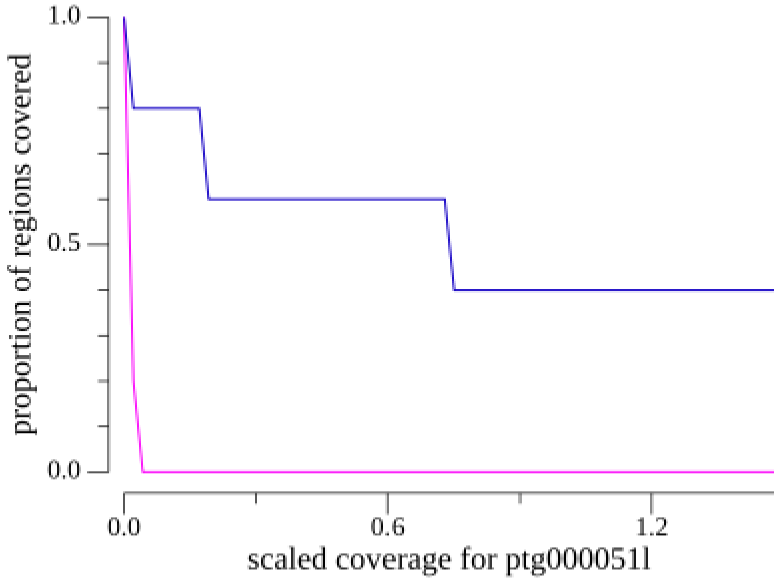

scaled coverage
1.0
0.5
0.0
25.0KB
100.0KB
175.0KB
position on ptg000017l

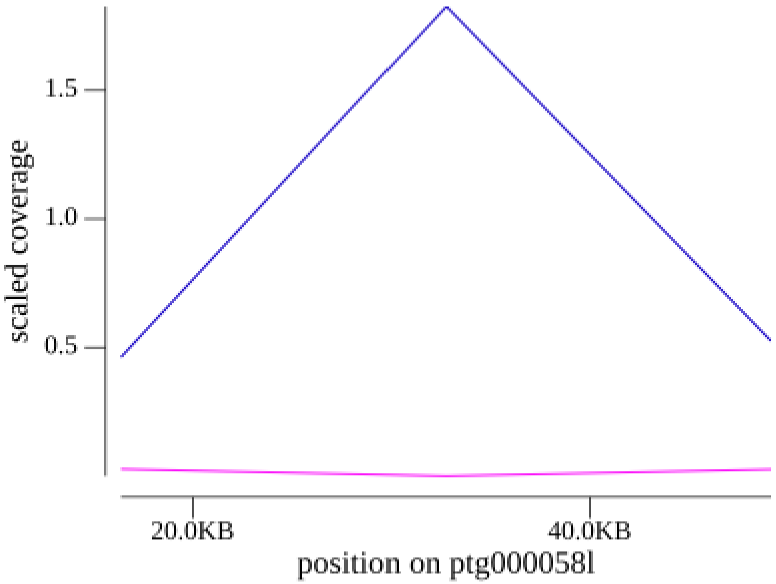

proportion of regions covered
1.0
0.5
0.0
0.0
0.6
1.2
scaled coverage for ptg000017l

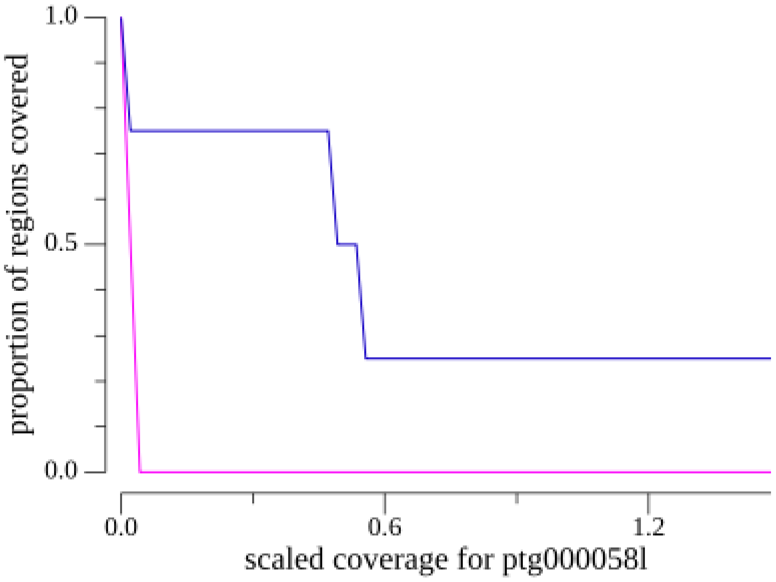

scaled coverage
2.0
1.5
1.0
0.5
0.0
0.0
500.0KB
1000.0KB
position on ptg000034l


1.0
0.5
0.0
proportion of regions covered
0.0
0.6
1.2
scaled coverage for ptg000034l


2.0
1.5
1.0
0.5
scaled coverage
20.0KB
40.0KB
60.0KB
position on ptg000051l


1.0
0.5
0.0
proportion of regions covered
0.0
0.6
1.2
scaled coverage for ptg000051l





Confirmation of putative Y scaffolds

Scaffold	Length	# of genes (unmasked)	# of genes (masked)	samtools coverage	indexcov	PCR
ptg000142l	33636			y	n	
ptg000058l	81416			y	y	
ptg000017l	217656			y	y	
ptg000014l	842711			y	y	
ptg000034l	1161669			y	y	
ptg000001l	5955038			y	y	
ptg000004l	24186116			y	y	
ptg000051l	97880			n	y	

Locate and mask repeats with repeatmodelor and repeatmasker on the cluster

```
#!/bin/bash
#SBATCH --job-name=RMaffM  # Job name
#SBATCH --partition=kucg      # Partition Name (Required)
#SBATCH --mail-type=END,FAIL,BEGIN     # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=tconway@ku.edu   # Where to send mail
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1          # Run on a single CPU
#SBATCH --mem=64gb           # Job memory request
#SBATCH --time=4-00:00:00        # Time limit days-hrs:min:sec
#SBATCH --output=RMaffM_%j.log  # Standard output and error log

module load repeatmodeler
module load repeatmasker/4.0.9

#usage: sbatch RepeatMasker.args.job <fasta> <prefix>

cd $SCRATCH
mkdir RMazteca

echo "STARTING"
cd RMazteca
cp $HOME/$1 .

BuildDatabase -name $2 -engine ncbi $1

RepeatModeler -engine ncbi -pa 8 -database $2

RepeatMasker -pa 8 -gff -lib $2-families.fa -dir MaskerOutput$2 $1

echo done
```

- With this data, you can look at Y-linked repeat families.

Annotate with helixer

- Go to https://www.plabipd.de/helixer_main.html
- Input fasta
- Change "Select Lineage-specific mode" to invertebrate
- Enter GFF label name and email address
- Submit job and wait
- grep gene foo.gff > genes.txt
- Import genes.txt into spreadsheet
- Convert gff to fasta using gffread (see below for code)
- blastx Y_transcripts.fa
- look up each gene on flybase and fill out spreadsheet

Renaming Transcripts

Using Nilanjan's method

Step 1: On the cluster

```
# gffread
gffread your_transcripts.gff -g genomic_reference.fasta -w
your_transcripts.fasta

# make a database
makeblastdb -in Dmel_translation_clean.fasta -dbtype prot -out
dmel_protein_database

# Run blastx
blastx -query DaztM_masked_fixednames_transcripts.fa -db
dmel_protein_database -outfmt 6 -evalue 1e-5 -max_target_seqs 1 -
num_threads 4 -out blast_azt_transcripts.txt

# retrieve protein sequences
cut -f2 blast_azt_transcripts.txt | sort | uniq >
aztM_best_hit_proteins.list
seqtk subseq ../Dmel_translation_clean.fasta aztM_best_hit_proteins.list >
aztM_best_hit_proteins.fa

makeblastdb -in DaztM_masked_fixednames_transcripts.fa -dbtype nucl -out
DaztM_transcripts_db

tblastn -query aztM_best_hit_proteins.fa -db DaztM_transcripts_db -outfmt
6 -evalue 1e-5 -max_target_seqs 1 -out aztM_blast_reciprocal.txt

awk '{print $1"\t"$2}' aztM_blast_reciprocal.txt > aztM_forward_hits.txt
awk '{print $2"\t"$1}' aztM_blast_reciprocal.txt >
aztM_reciprocal_hits.txt
sort aztM_forward_hits.txt aztM_reciprocal_hits.txt | sed  's/-
P[ABCDEFGHIJKLMNOPQRSTUVWXYZ]//g'  | uniq > reciprocal_best_hits.txt
awk '{if(a[$2]++){print $1"\t"$2"."a[$2]}else{print $0}}'
reciprocal_best_hits.txt > aztM_RBH.txt
awk '{print $0 ".1"}' aztM_RBH.txt > aztM_RBH2.txt
awk '{sub(/\.1$/, "", $1); print}' RBH.txt  >RBH3.txt

gawk 'NR==FNR { mapping[$1] = $2; next } { for (key in mapping) gsub(key,
mapping[key]) } 1' RBH2.txt ../DaztM_masked_helixer.gff > aztM_temp.gff

gawk 'NR==FNR { mapping[$1] = $2; next } { for (key in mapping) gsub(key,
mapping[key]) } 1' RBH3.txt temp.gff > renamed.gff
```

Primers for Y validation

azt_ptg4_1_F - CTGCATCATCTGGGTAAGTCG azt_ptg4_1_R - TTTCGCACCGGAAAGTTTTGG

For azteca, we may only be able to find primers for ptg4.