Museomics Workshop 2025: Bioinformatics Pipeline

This document describes the bioinformatics workflow for processing and analyzing sequencing data for the Museomics Workshop 2025. The slides to this workshop can be found here.

---

1. Preparation

(a) System Requirements

- **Linux/Unix (macos) system** with a **BASH terminal**. Note that this workshop is **not compatible with Windows systems**, but you can use the Windows Subsystem for Linux (WSL) to run the commands. Moreover, please note that the macos terminal is now by default a ZSH terminal, so you need to change the shell to BASH by running `chsh -s /bin/bash` in the terminal.
- app. **10GB** free space.
- **>= 16GB RAM / >= 4 cores**
- **VScode** and Markdown Preview Enhanced installed to view this document and execute the code snippets.
- **Conda installed and in PATH** so that you can run `conda` commands in the terminal. If you don't have conda installed, you can install it by following the instructions here.

(b) Install required software

```
## set Working directory
WD=</path/to/your/working/directory>
#WD=/media/inter/mkapun/projects/MuseomicsWorkshop2025

## make sure conda and mamba are installed on your system
sh ${WD}/shell/requirements.sh ${WD}
```

(c) Download and decompress the raw input FASTQ data

```
## create and change to data directory
cd ${WD}/data && mkdir ${WD}/data/raw_reads

## download raw reads
wget -O raw_reads.tar.gz "https://filesender.aco.net/download.php?token=18816b43-8209-46cf-a
```

```
## untar the compressed folder within ${WD}/data
tar -xzf ${WD}/data/raw_reads.tar.gz -C ${WD}/data/raw_reads

## optionally download mapDamage2 results
cd ${WD}/data && mkdir -p ${WD}/results/mapDamage
wget -O mapDamage2.tar.gz "https://filesender.aco.net/download.php?token=30903ed5-4981-4060-
tar -xzf ${WD}/data/mapDamage2.tar.gz -C ${WD}/results/mapDamage
```

Datasets

Library	Name	Age	City	Country	Wolbachia	Type	SRA
DGRP370	DGRP370	2003	Raleigh	USA	wMel	recent	SRR834539
DGRP338	DGRP338	2003	Raleigh	USA	wMelCS	recent	SRR834513
ZI268	ZI268	2003	Ziawonga	Zambia	wMel	recent	SRR189425
HG_15	19SL15	1933	Lund	Sweden	unknown	historic	SRR23876580
HG0027	19SL3	1933	Lund	Sweden	unknown	historic	SRR23876574
HG0029	18DZ5	1899	Zealand	Denmark	unknown	historic	SRR23876565

2. Preview Raw FASTQ Data

Preview the first 10 lines of the forward read file:

```
gunzip -c ${WD}/data/raw_reads/ZI268_1.fastq.gz | less
gunzip -c ${WD}/data/raw_reads/18DZ5_1.fastq.gz | less
```

> *QUESTIONS*:
> A) What is the structure of the datasets?
> B) How do the two files differ?

---

3. Trim Reads with *fastp*

In our first analysis, we will focus on a subset of 1,000,000 reads from each of the datasets and test how trimming and adapter removal will influence the read lengths and base quality of the datasets.

```
mkdir -p ${WD}/data/trimmed_reads && cd ${WD}/data/trimmed_reads
conda activate ${WD}/scripts/programs
```

Loop through each library in datasets.csv and trim reads:

```
while IFS=$"," read -r Library Name Age City Country Wolb Type SRA; do
    if [[ "${Library}" != "Library" ]]; then
```

```
        echo "Processing library ${Name}"
        fastp \
            -i ${WD}/data/raw_reads/${Name}_1.fastq.gz \
            -I ${WD}/data/raw_reads/${Name}_2.fastq.gz \
            -o ${WD}/data/trimmed_reads/${Name}_1_trimmed.fastq.gz \
            -O ${WD}/data/trimmed_reads/${Name}_2_trimmed.fastq.gz \
            --merge \
            --merged_out ${WD}/data/trimmed_reads/${Name}_merged.fastq.gz \
            --length_required 25 \
            --dedup \
            --trim_poly_g \
            --html ${WD}/data/trimmed_reads/${Name}.html \
            --json ${WD}/data/trimmed_reads/${Name}.json \
            --detect_adapter_for_pe
    fi
done <${WD}/data/datasets.csv
```

Let's have a look at the HTML output files:

```
open ${WD}/data/trimmed_reads/18DZ5.html
open ${WD}/data/trimmed_reads/ZI268.html
```

> *QUESTIONS*:
> A) What do the parameters mean?
> B) How do the read length distributions differ between the datasets?

---

4. Testing for eukaryotic contamination

In the next analysis, we will again focus on the trimmed subset of 1,000,000 reads from each of the datasets and test if we find evidence for contamination with exogenous DNA. Here we will use BLAST to search for sequence similarity against a database of mitochondrial genomes. This is a common approach to screen for contamination in ancient DNA studies, as mitochondrial DNA is often well-preserved and can be easily amplified. If you want to test for contamination with prokaryotic DNA, you can use Kraken2 with a database of bacterial genomes (e.g. here), which we unfortunately cannot cover in this workshop.

Download mitochondrial genomes for contamination screening. We will focus on the focal species *Drosophila melanogaster* and other likely contaminants such as *Homo sapiens*, the carpet or museum beetle *Anthrenus verbasci*, a common pest in museums, and in our case *Gryllus bimaculatus*.

```
mkdir -p ${WD}/data/refseq/contamination && cd ${WD}/data/refseq/contamination

## Gryllus bimaculatus
wget "https://www.ncbi.nlm.nih.gov/sviewer/viewer.fcgi?id=PP230540.1&db=nuccore&report=fasta
```

```
sed -i ’’ ’1s/.*/>Gryllus_bimaculatus/’ Gryllus_bimaculatus_mito.fasta

## Homo sapiens
wget "https://www.ncbi.nlm.nih.gov/sviewer/viewer.fcgi?id=NC_012920.1&db=nuccore&report=fast
sed -i ’’ ’1s/.*/>Homo_sapiens/’ Homo_sapiens_mito.fasta

## Drosophila melanogaster
wget "https://www.ncbi.nlm.nih.gov/sviewer/viewer.fcgi?id=NC_024511.2&db=nuccore&report=fast
sed -i ’’ ’1s/.*/>Drosophila_melanogaster/’ Drosophila_melanogaster_mito.fasta

## Anthrenus verbasci
wget "https://www.ncbi.nlm.nih.gov/sviewer/viewer.fcgi?id=PV608434.1&db=nuccore&report=fasta
sed -i ’’ ’1s/.*/>Anthrenus_verbasci/’ Anthrenus_verbasci_mito.fasta

## now concatenate all files into one
cat *_mito.fasta > mitochondrial.fasta
```

Next, we will build a BLAST database based on these genomes to test for sequence similarity with our sequencing data.

```
cd ${WD}/data/refseq/contamination

## build nucleotide database
makeblastdb -in mitochondrial.fasta -dbtype nucl -out mitoDB
```

After that, we convert the trimmed FASTQ files to FASTA format and run BLAST against the mitochondrial database. We will only retain the top hit for each read if the e-value is lower than 1e-10 and the sequence similarity is larger than 90%.

```
## make a results folder
mkdir -p ${WD}/results/contamination

## make empty output file
>${WD}/results/contamination/BLAST.tsv

## loop through all files in dataset and convert the FASTQ files, then BLAST
while IFS=$"," read -r Library Name Age City Country Wolb Type SRA; do
    if [[ "${Library}" != "Library" ]]; then
        gunzip -c ${WD}/data/trimmed_reads/${Name}_1_trimmed.fastq.gz | \
            awk ’NR % 4 == 1 {print ">" substr($0, 2)} NR % 4 == 2 {print $0}’ >${WD}/data/t

        gunzip -c ${WD}/data/trimmed_reads/${Name}_merged.fastq.gz | \
            awk ’NR % 4 == 1 {print ">" substr($0, 2)} NR % 4 == 2 {print $0}’ >>${WD}/data/

        blastn \
            -num_threads 4 \
            -evalue 1e-10 \
            -max_target_seqs 1 \
            -outfmt ’6 qseqid sseqid slen qlen pident length mismatch evalue bitscore’ \
            -db ${WD}/data/refseq/contamination/mitoDB \
            -query ${WD}/data/trimmed_reads/${Name}_trimmed.fasta | \
            awk -v Name=${Name} ’$5>90 {print Name"\t"$0}’ >>${WD}/results/contamination/BLA
    fi
done <${WD}/data/datasets.csv
```

OK, let’s have a look at the tabular output file.

```
less  ${WD}/results/contamination/BLAST.tsv
```

> *QUESTIONS*:
> A) What do the columns mean?
> B) Do you see non-*Drosophila* hits?

Finally, we plot the proportion of reads mapped to mitochondrial genomes in R using ggplot2.

```
${WD}/scripts/programs/bin/Rscript -e "
library(tidyverse)
df <- read_tsv(’${WD}/results/contamination/BLAST.tsv’, col_names = FALSE)
df.prop <- df %>%
    select(1,3,5,9) %>%
    rename(Library = 1, Name = 2, ReadLength = 3, Evalue = 4) %>%
    group_by(Library,Name) %>%
    summarise(Count = n()) %>%
    mutate(Proportion = Count / sum(Count)) %>%
    arrange(desc(Count))
plot <- ggplot(df.prop, aes(x = Library, y = Proportion, fill = Name)) +
    geom_bar(stat = ’identity’, position = ’fill’) +
    labs(title = ’Proportion of Reads Mapped to Mitochondrial Genomes’,
         x = ’Library’,
         y = ’Proportion of Reads’) +
    theme_bw() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
ggsave(’${WD}/results/contamination/BLAST_proportion_plot.pdf’, plot, width = 10, height = 6
ggsave(’${WD}/results/contamination/BLAST_proportion_plot.png’, plot, width = 10, height = 6
"
```

> *QUESTIONS*:
> A) What is the proportion of contamination?
> B) Are there differences between historic and recent datasets?

Finally, we are testing if there are differences in the read lengths between the endogenous and exogenous (i.e. contaminant) DNA.

```
${WD}/scripts/programs/bin/Rscript -e "
library(tidyverse)
df <- read_tsv(’${WD}/results/contamination/BLAST.tsv’, col_names = FALSE)
```

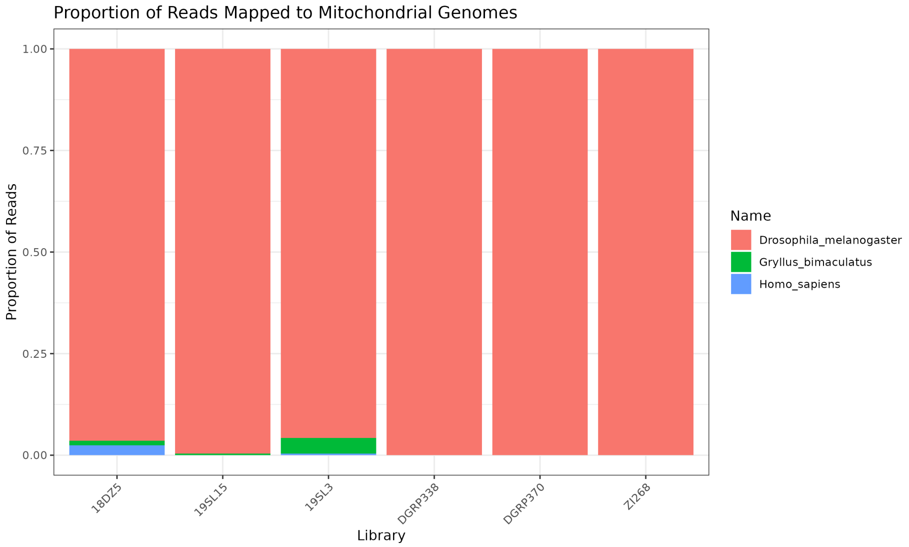


Figure 1: BLAST Contamination Plot

```
df.prop.RL <- df %>%
    select(1,3,5,9) %>%
    rename(Library = 1, Name = 2, ReadLength = 3, Evalue = 4) %>%
    group_by(Library,Name,ReadLength) %>%
    summarise(Count = n()) %>%
    mutate(Proportion = Count / sum(Count)) %>%
    arrange(desc(Count))
plot.RL <- ggplot(df.prop.RL,
    aes(x = ReadLength,
        y = Count,
        fill = Name)) +
    geom_bar(stat = 'identity',
        position = 'dodge') +
    labs(title = 'Proportion of Reads Mapped to Mitochondrial Genomes by Read Length',
         x = 'Read Lengths',
         y = 'Read Counts') +
         facet_grid(Library~Name, scales = 'free_y') +
    guides(fill = guide_legend(title = 'Mitochondrial Genome')) +
    theme_bw() +
    scale_y_log10() +
    theme(
        plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 45, hjust = 1)
    ) + theme(legend.position = 'bottom')
ggsave('${WD}/results/contamination/BLAST_proportion_plot_by_RL.pdf',
    plot.RL,
    width = 7,
    height = 5)
ggsave('${WD}/results/contamination/BLAST_proportion_plot_by_RL.png',
    plot.RL,
    width = 7,
    height = 5)
"
```

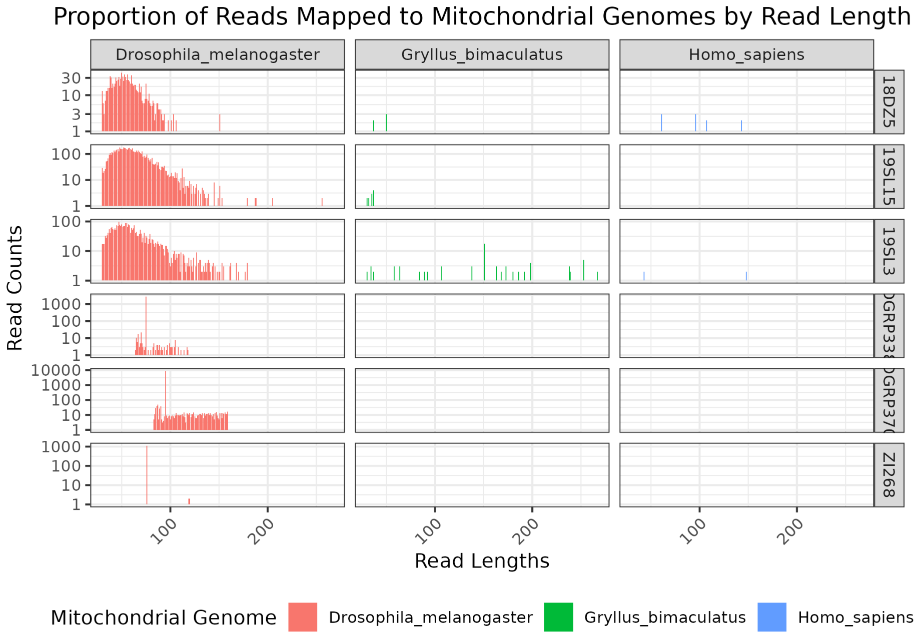


Figure 2: BLAST Contamination Plot by read lengths

QUESTIONS:
A) How to interpret the distribution of read lengths in the historic and recent samples?
B) Do the read lengths differ between the exogenous and the endogenous DNA?
C) What does this tell us about the contamination source?

5. Testing for DNA degradation

For this analysis, we will use another dataset that has been prefiltered to map to a specific genomic region (2L:1-100,000) of the *D. melanogaster* genome, to the

D. melanogaster mitochondrion and to the wMel *Wolbachia* genome. In a first step, we are mapping reads to the three aforementioned genomes using minimap2, with specific settings for short reads. Since several partially overlapping reads have been merged during the trimming, we will map these separately from the paired-end reads and merge the BAM files afterwards.

```
mkdir -p ${WD}/results/minimap2 && cd ${WD}/results/minimap2
while IFS=$"," read -r Library Name Age City Country Wolb Type SRA; do
    if [[ "${Library}" != "Library" ]]; then
        echo "Processing library ${Name}"
        minimap2 -ax sr --secondary=no -t 150 \
            ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
            ${WD}/data/raw_reads/${Name}_2L_100K_1_trimmed.fastq.gz \
            ${WD}/data/raw_reads/${Name}_2L_100K_2_trimmed.fastq.gz | \
            samtools view -bS -F 4 - | \
            samtools sort -o ${WD}/results/minimap2/${Name}_PE.bam
        samtools index ${WD}/results/minimap2/${Name}_PE.bam

        minimap2 -ax sr --secondary=no -t 150 \
            ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
            ${WD}/data/raw_reads/${Name}_2L_100K_merged.fastq.gz | \
            samtools view -bS -F 4 - | \
            samtools sort -o ${WD}/results/minimap2/${Name}_merged.bam
        samtools index ${WD}/results/minimap2/${Name}_merged.bam

        samtools merge -f ${WD}/results/minimap2/${Name}.bam \
            ${WD}/results/minimap2/${Name}_PE.bam \
            ${WD}/results/minimap2/${Name}_merged.bam
        samtools index ${WD}/results/minimap2/${Name}.bam

        rm ${WD}/results/minimap2/${Name}_PE.bam*
        rm ${WD}/results/minimap2/${Name}_merged.bam*
    fi
done <${WD}/data/datasets.csv
```

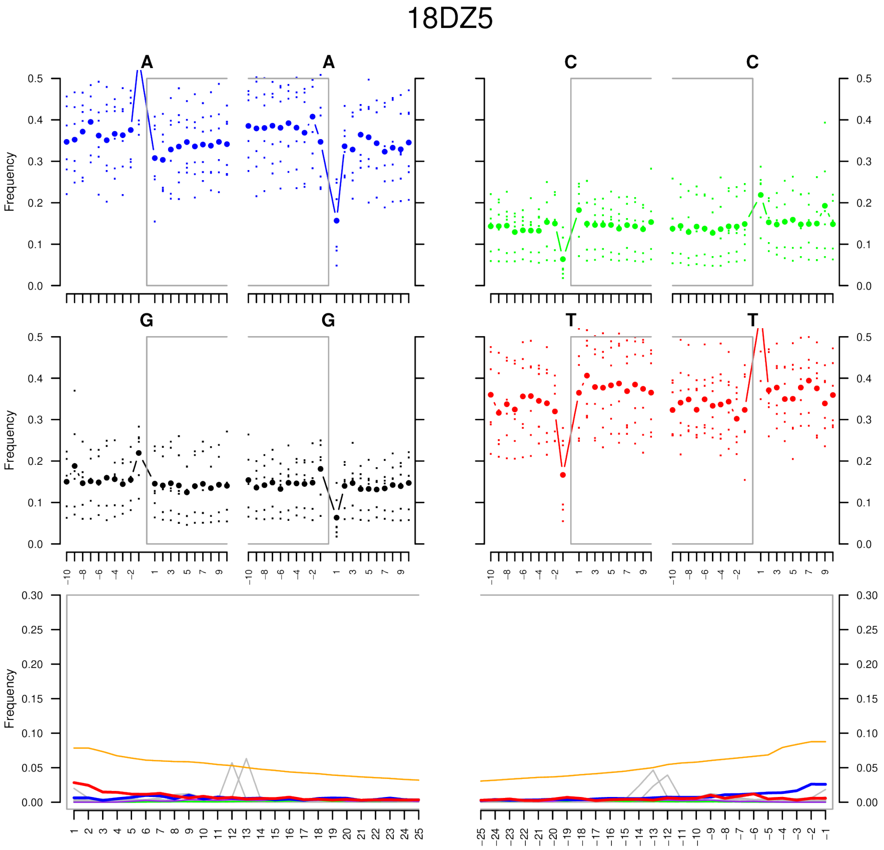
In the next step, we will employ MapDamage2 to investigate deamination patterns with respect to the fragment position. MapDamage2 is a tool designed to analyze DNA damage patterns in ancient DNA sequences, particularly focusing on the deamination of cytosines to uracils at the 5' and 3' end of fragments, which is a common form of damage in ancient samples. It provides insights into the age and preservation state of the DNA by modeling the expected damage patterns.

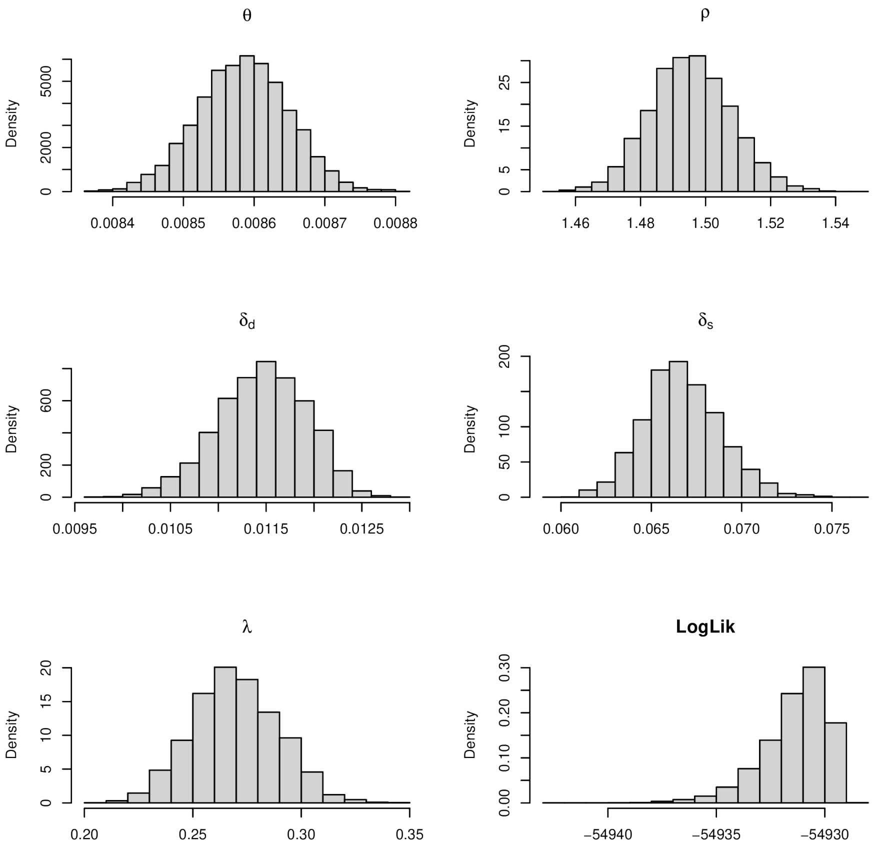
```
## change conda environment
conda deactivate
conda activate ${WD}/scripts/mapdamage2


## make output folder
```

```
mkdir -p ${WD}/results/mapDamage && cd ${WD}/results/mapDamage

## loop through libraries in datasets.csv
while IFS=$"," read -r Library Name Age City Country Wolb Type SRA; do
    if [[ "${Library}" != "Library" ]]; then
        echo "Processing library ${Name}"

        # run mapDamage2
        mapDamage -i ${WD}/results/minimap2/${Name}.bam \
            -r ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
            --rescale \
            --folder=${WD}/results/mapDamage/${Name}

        ## convert PDFs to PNGs (only works on Linux systems)
        for pdf in ${WD}/results/mapDamage/${Name}/*.pdf; do
            png=${pdf%.pdf}.png
            convert -density 300 $pdf -quality 90 $png
        done

    fi
done <${WD}/data/datasets.csv

## change environment
conda deactivate
conda activate ${WD}/scripts/programs
```

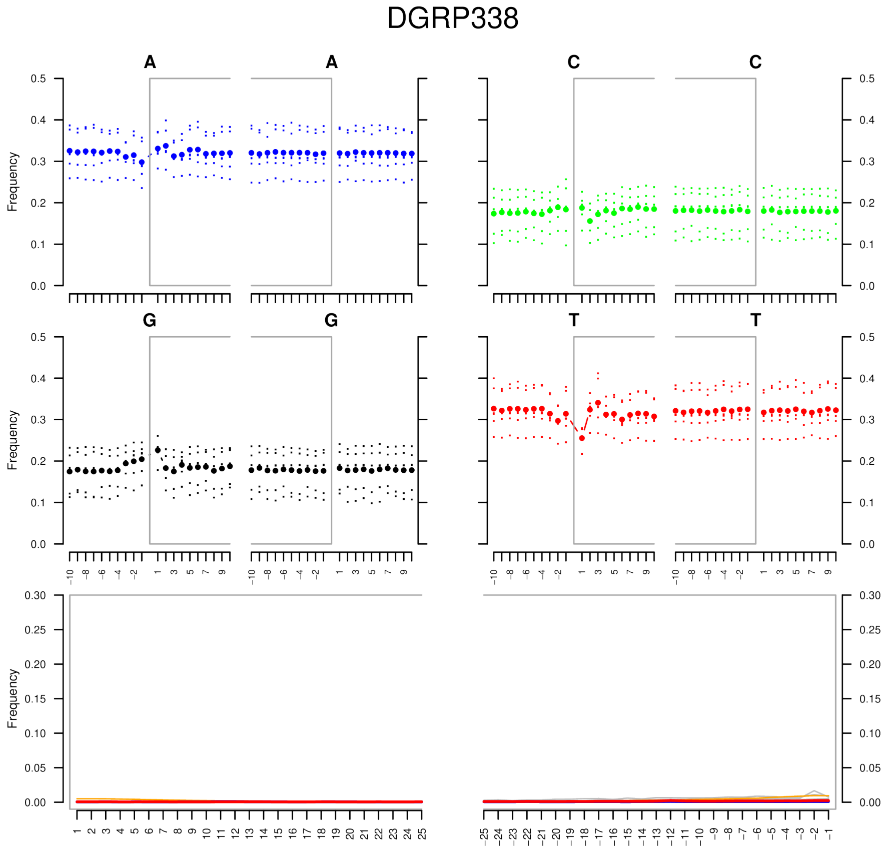
Now let's have a look at the results for a historic (18DZ5) and a recent (DGRP338) library.

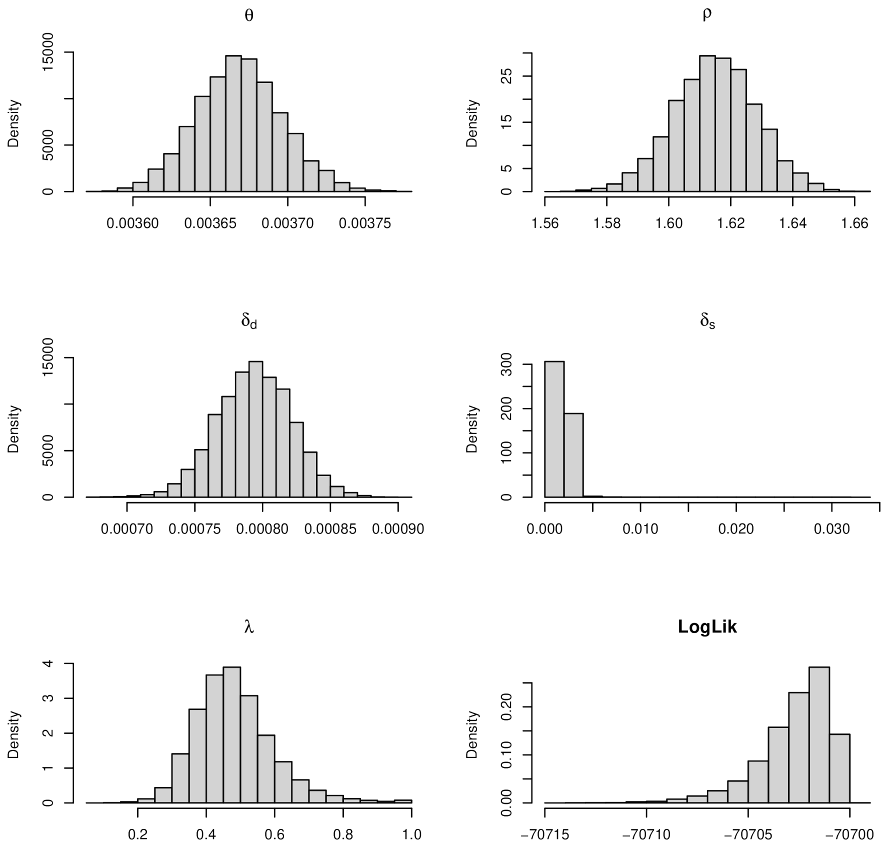
18DZ5 (historic)


18DZ5
A
A
C
C
G
G
T
T
Frequency
Frequency
Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−10
−8
−6
−4
−2
1
3
5
7
9
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
−25
−24
−23
−22
−21
−20
−19
−18
−17
−16
−15
−14
−13
−12
−11
−10
−9
−8
−7
−6
−5
−4
−3
−2
−1



DGRP338 (recent)


DGRP338
A
A
C
C
G
G
T
T
Frequency
Frequency
Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−10
−8
−6
−4
−2
1
3
5
7
9
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
−25
−24
−23
−22
−21
−20
−19
−18
−17
−16
−15
−14
−13
−12
−11
−10
−9
−8
−7
−6
−5
−4
−3
−2
−1



Let's summarize the results in a table across all libraries based on the `Stats_out_MCMC_iter_summ_stat.csv` files

```
## make output folder
mkdir -p ${WD}/results/mapDamage && cd ${WD}/results/mapDamage
## make empty output file
echo "Name Theta DeltaD DeltaS Lambda">${WD}/results/mapDamage/MapDamage_summary.txt

## loop through libraries in datasets.csv
while IFS=$"," read -r Library Name Age City Country Wolb Type SRA; do
    if [[ "${Library}" != "Library" ]]; then
        echo "Processing library ${Name}"
        awk -F "," -v "V"=${Name} '/Mean/ {print V, $2, $3, $4, $5}' ${WD}/results/mapDamage
    fi
done <${WD}/data/datasets.csv
```

Now plot the mean values across all libraries

```
${WD}/scripts/programs/bin/Rscript -e "
library(tidyverse)

## Load the data
data <- read.table('${WD}/results/mapDamage/MapDamage_summary.txt', header = TRUE, sep = '
data.long<-reshape(data,
    direction='long',
    varying=list(colnames(data)[2:ncol(data)]),
    timevar='Parameter',
    times=colnames(data)[2:ncol(data)])

# plot the coverage distribution by region and library
ggplot(data.long, aes(x = Name, y = Theta, fill = Name)) +
    geom_bar(stat = 'identity', position = 'dodge') +
    labs(title = 'Briggs Paramters',
         x = 'Name',
         y = 'Value') +
    theme_bw() +
    facet_wrap(.~Parameter , scales = 'free_y') +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Save the plot as PDF and PNG
ggsave('${WD}/results/mapDamage/MapDamage_summary.pdf', width = 8, height = 4)
ggsave('${WD}/results/mapDamage/MapDamage_summary.png', width = 8, height = 4)
"
```

Briggs Parameters

Briggs parameters

> *QUESTIONS*:
> A) How to interpret these plots?
> B) What are the differences between historic and recent samples?

6. Read Depth and Coverage

Before we carry out the formal analysis of our dataset, we will at last calculate read depths (i.e. the average number of reads mapping to a given position) and coverage (the proportion of the genome covered by at least one read) for each genomic region and library using samtools. Furthermore, we will calculate the ratio of *Wolbachia* to *Drosophila* reads to obtain an estimate of the prevalence and relative bacterial titer in infected flies.

```
## create output folder
mkdir -p ${WD}/results/ReadDepths && cd ${WD}/results/ReadDepths
```

```
## Make header line
printf "library\trname\tstartpos\tendpos\tnumreads\tcovbases\tcoverage\tmeandepth\tmeanbaseq
    >${WD}/results/ReadDepths/ReadDepths.txt

## Loop through libraries and calculate summary statistics for each library and genomic reg
for i in ${WD}/results/minimap2/*.bam; do

    ## get the library name
    base=$(basename $i .bam)

    ## calculate statistics
    samtools coverage \
        --reference ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
        ${i} |
        awk -v base=${base} 'BEGIN{OFS="\t"} NR >1 {print base, $0}' \
            >>${WD}/results/ReadDepths/ReadDepths.txt
done
```

Now, we can plot and compare read depths and coverages in R.

```
## plot read depths, coverages and relative bacterial titer
${WD}/scripts/programs/bin/Rscript -e "

library(tidyverse)

## Load the data
data <- read.table('${WD}/results/ReadDepths/ReadDepths.txt', header = TRUE, sep = '\t')
# Convert the library and rname columns to factors with specific levels
data\$library <- factor(data\$library, levels = c('18DZ5', '19SL15', '19SL3', 'DGRP338', 'DG
# Convert the rname column to a factor with specific levels
data\$rname <- factor(data\$rname, levels = c('2L', 'wMel', 'mitochondrion_genome'))
# rename the rname column to Region
data\$rname <- recode(data\$rname, '2L' = '2L', 'wMel' = 'Wolbachia', 'mitochondrion_genome'

# plot the coverage distribution by region and library
ggplot(data, aes(x = rname, y = coverage, fill = library)) +
    geom_bar(stat = 'identity', position = 'dodge') +
    labs(title = 'Coverage Distribution by Region and Library',
         x = 'Region',
         y = 'Coverage') +
    theme_bw() +
    facet_grid(.~library , scales = 'free_y') +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))+
    theme(legend.position = 'bottom')

# Save the plot as PDF and PNG
```

```
ggsave('${WD}/results/ReadDepths/coverage_distribution.pdf', width = 8, height = 6)
ggsave('${WD}/results/ReadDepths/coverage_distribution.png', width = 8, height = 6)

# plot the read depths and coverages and the ratio or read depths for 2L and wMel
ggplot(data, aes(x = rname, y = numreads, fill = library)) +
    geom_bar(stat = 'identity', position = 'dodge') +
    labs(title = 'Read Depth Distribution by Region and Library',
         x = 'Region',
         y = 'Read Depth') +
    theme_bw() +
    facet_grid(.~library , scales = 'free_y') +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))+
    theme(legend.position = 'bottom')

# Save the plot as PDF and PNG
ggsave('${WD}/results/ReadDepths/read_depth_distribution.pdf', width = 8, height = 6)
ggsave('${WD}/results/ReadDepths/read_depth_distribution.png', width = 8, height = 6)

## plot the ratio of read depths for Wolbachia and 2L
# First, calculate the ratio of read depths for Wolbachia and 2L
data\$ratio <- ifelse(data\$rname == 'Wolbachia', data\$numreads / data[data\$rname == '2L',

# plot the ratio of read depths for Wolbachia and 2L
ggplot(data, aes(x = library, y = ratio, fill = library)) +
    geom_bar(stat = 'identity', position = 'dodge') +
    labs(title = 'Ratio of Read Depths for Wolbachia and 2L',
         x = 'Library',
         y = 'Ratio of Read Depths') +
    theme_bw() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))+
    theme(legend.position = 'bottom')

# Save the plot as PDF and PNG
ggsave('${WD}/results/ReadDepths/read_depth_ratio.pdf', width = 8, height = 6)
ggsave('${WD}/results/ReadDepths/read_depth_ratio.png', width = 8, height = 6)
"
```

Read Depths

Coverage

Ratio *Wolbachia* / 2L

QUESTIONS:

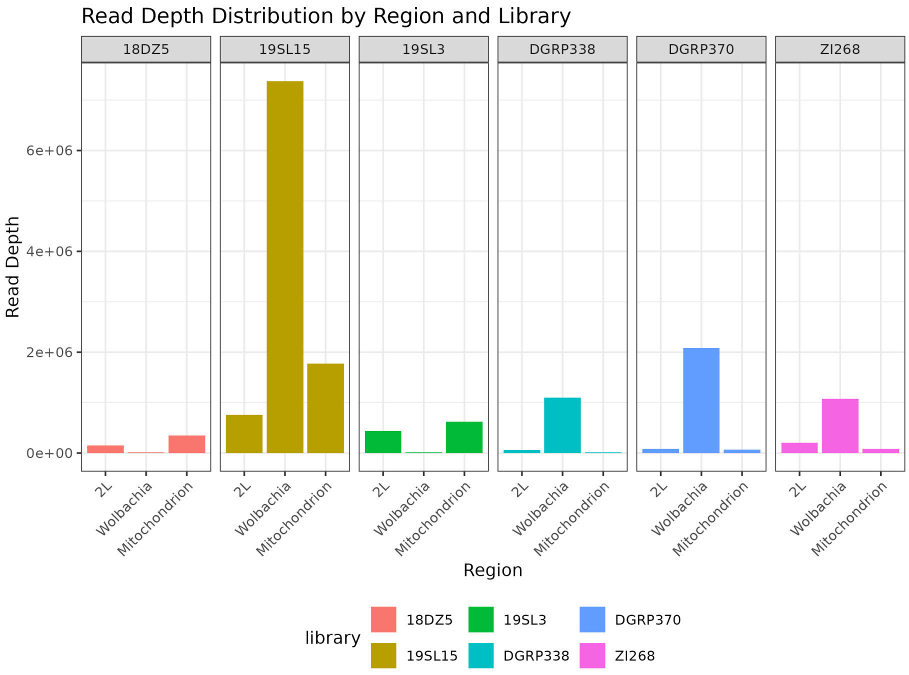


Figure 3: Read Depths

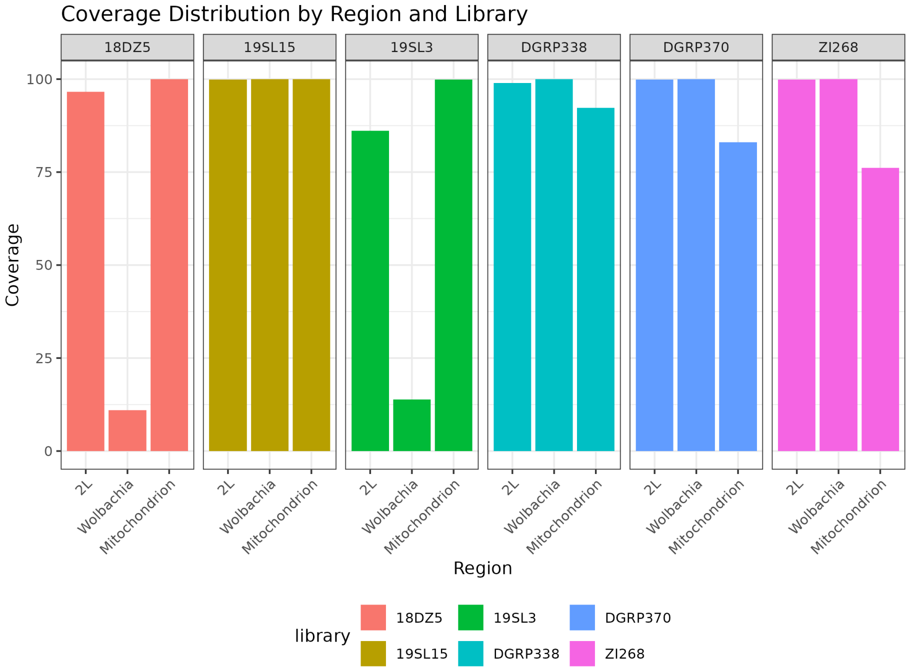


Figure 4: Coverage

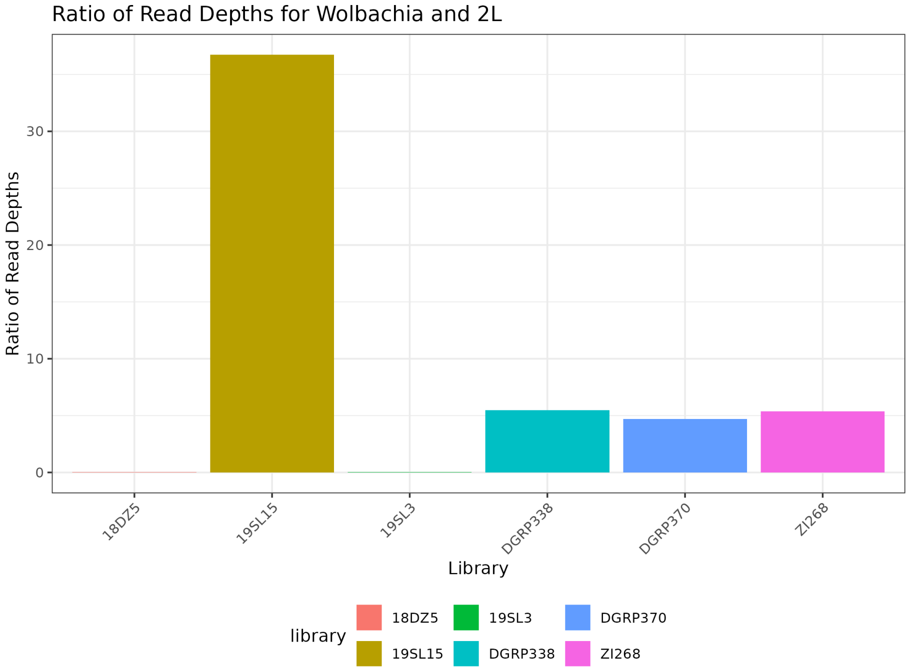


Figure 5: Ratio

A) Are there differences in the read depths and coverages between recent and historic samples?
B) How does the read depth of *Wolbachia* compare to the nuclear genome (2L)?
C) What can we infer about the *Wolbachia* infection status?

---

7. SNP Calling and Phylogenetic Analysis

In our final analysis, we will assess if the historic sample which shows relatively high bacterial titer is likely infected with the most common *wMel* or the rare *wMelCS* variant. We will therefore call SNP variants for each of the genomic regions based on the BAM files with rescaled base-quality scores from mapDamage using bcftools, which accounts for the ploidy of the corresponding contigs (*Wolbachia* and Mitochondrion: haploid; 2L: diploid). We will use a custom script to convert the SNPs in VCF file format to the Phylip format. For the diploid genome, we will randomly draw one allele in heterozygous positions to create a pseudo-haploid haplotype. We will then employ a very simple phylogenetic approach to compare the datasets.

At first we are liberal and even include the historic sample with very low coverage (18DZ5) for the SNP calling. In a second step, we will exclude this sample and repeat the analysis.

```
## Use rescaled BAM files for SNP calling
>${WD}/results/minimap2/scaled_bam.list
for i in ${WD}/results/mapDamage/*/*.rescaled.bam; do
    echo $i >>${WD}/results/minimap2/scaled_bam.list

    ## make index for scaled BAM files
    samtools index $i
done

## make output folder
mkdir -p ${WD}/results/SNPs && cd ${WD}/results/SNPs

## Diploid region (2L)
bcftools mpileup -Ou \
    -f ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
    -r 2L \
    -a AD,DP \
    -b ${WD}/results/minimap2/scaled_bam.list |
    bcftools call -mv --ploidy 2 -Ou |
    bcftools view -v snps -Oz -o ${WD}/results/SNPs/2L.diploid.vcf.gz
```

```
python3 ${WD}/scripts/DiploVCF2Phylip.py \
    --input ${WD}/results/SNPs/2L.diploid.vcf.gz \
    --MaxPropGaps 0.1 \
    --MinCov 30 \
    >${WD}/results/SNPs/2L.phy

# Haploid region (mitochondrion_genome)
bcftools mpileup -Ou \
    -f ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
    -r mitochondrion_genome \
    -a AD,DP \
    -b ${WD}/results/minimap2/scaled_bam.list |
    bcftools call -mv --ploidy 1 -Ou |
    bcftools view -v snps -Oz -o ${WD}/results/SNPs/mito.haploid.vcf.gz

python3 ${WD}/scripts/HaploVCF2Phylip.py \
    --input ${WD}/results/SNPs/mito.haploid.vcf.gz \
    --MinAlt 1 \
    --MaxPropGaps 0.7 \
    --MinCov 10 \
    >${WD}/results/SNPs/mito.phy

# Haploid region (wMel)
bcftools mpileup -Ou \
    -f ${WD}/data/refseq/dmel/dmel_wMel_2L_100K.fasta.gz \
    -r wMel \
    -a AD,DP \
    -b ${WD}/results/minimap2/scaled_bam.list |
    bcftools call -mv --ploidy 1 -Ou |
    bcftools view -v snps -Oz -o ${WD}/results/SNPs/Wolbachia.haploid.vcf.gz

python3 ${WD}/scripts/HaploVCF2Phylip.py \
    --input ${WD}/results/SNPs/Wolbachia.haploid.vcf.gz \
    --MinAlt 1 \
    --MaxPropGaps 0.5 \
    --MinCov 5 \
    >${WD}/results/SNPs/Wolbachia.phy
```

Visualize phylogenetic trees in R.

```
## Make UPGMA trees based on genetic distance
${WD}/scripts/programs/bin/Rscript -e "
library(ggtree)
library(phangorn)
library(phytools)
library(ape)
library(tidyverse)
library(patchwork)

## read input data
input_files <- c(
  '${WD}/results/SNPs/2L.phy',
  '${WD}/results/SNPs/mito.phy',
  '${WD}/results/SNPs/Wolbachia.phy'
)

## adjust titles
titles <- c('2L 100K Region', 'Mitochondrion', 'Wolbachia')
plots <- list()
for (i in seq_along(input_files)) {
  phylo <- read.phyDat(input_files[i], format = 'phylip')
  tree <- upgma(dist.ml(phylo))
  tree <- midpoint.root(tree)
  Xmax <- max(nodeHeights(tree))
  p <- ggtree(tree, layout = 'roundrect') +
    geom_tiplab(size = 3) +
    ggplot2::xlim(0, Xmax + Xmax/3) +
    ggtitle(titles[i]) +
    theme_tree2() +
    theme_bw()
  plots[[i]] <- p
}
combined_plot <- plots[[1]] + plots[[2]] + plots[[3]] + plot_layout(ncol = 3)
ggsave('${WD}/results/SNPs/combined_phylo_trees.pdf', combined_plot, width = 12, height = 6)
ggsave('${WD}/results/SNPs/combined_phylo_trees.png', combined_plot, width = 12, height = 6)
"
```

QUESTIONS:
A) Are there differences between the nuclear, the mitochondrial and the *Wolbachia* tree?
B) How to interpret the long branch in the *Wolbachia* tree?
C) What does the tree based on 2L depict?

Finally, we exclude the potentially contaminated sample and re-plot phylogenetic trees:

```
python3 ${WD}/scripts/HaploVCF2Phylip.py \
    --input ${WD}/results/SNPs/Wolbachia.haploid.vcf.gz \
    --MinAlt 1 \
    --MaxPropGaps 0.5 \
    --MinCov 5 \
    --exclude 18DZ5.rescaled,19SL3.rescaled \
```

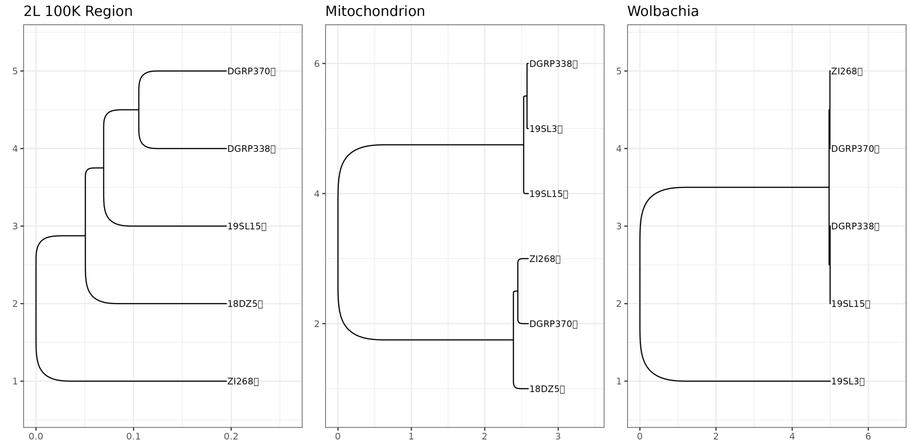


Figure 6: Phylogenetic Trees

```
    >${WD}/results/SNPs/Wolbachia_no19SL3.phy
```

```
${WD}/scripts/programs/bin/Rscript -e "
library(ggtree)
library(phangorn)
library(phytools)
library(ape)
library(tidyverse)
library(patchwork)
input_files <- c(
  '${WD}/results/SNPs/2L.phy',
  '${WD}/results/SNPs/mito.phy',
  '${WD}/results/SNPs/Wolbachia_no19SL3.phy'
)
titles <- c('2L 100K Region', 'Mitochondrion', 'Wolbachia')
plots <- list()
for (i in seq_along(input_files)) {
  phylo <- read.phyDat(input_files[i], format = 'phylip')
  tree <- upgma(dist.ml(phylo))
  #tree <- midpoint.root(tree)
  Xmax <- max(nodeHeights(tree))
  p <- ggtree(tree, layout = 'roundrect') +
    geom_tiplab(size = 3) +
    ggplot2::xlim(0, Xmax + Xmax/3) +
    ggtitle(titles[i]) +
    theme_tree2() +
    theme_bw()
  plots[[i]] <- p
}
combined_plot <- plots[[1]] + plots[[2]] + plots[[3]] + plot_layout(ncol = 3)
ggsave('${WD}/results/SNPs/combined_phylo_trees_no19SL3.pdf', combined_plot, width = 12, hei
ggsave('${WD}/results/SNPs/combined_phylo_trees_no19SL3.png', combined_plot, width = 12, hei
"
```

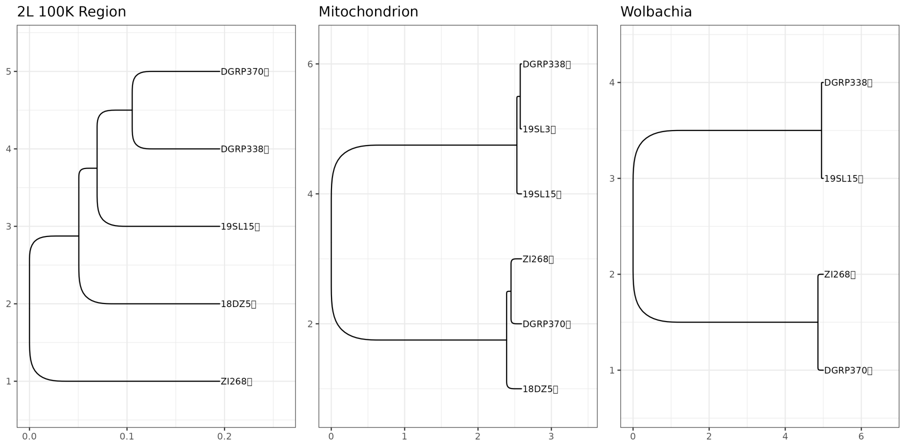


Figure 7: Phylogenetic Trees clean

QUESTIONS:
A) Why are the mitochondrial and the *Wolbachia* trees so similar?
B) What can we infer about the *Wolbachia* variants in the historic sample(s)?

Thank you for participating in the Museomics Workshop 2025! I hope you found this bioinformatics pipeline useful for your own research. If you have any questions or feedback, please feel free to reach out to me