All wombats data report

Raphael Eisenhofer

Wombat “Trifecta” metagenomic data report

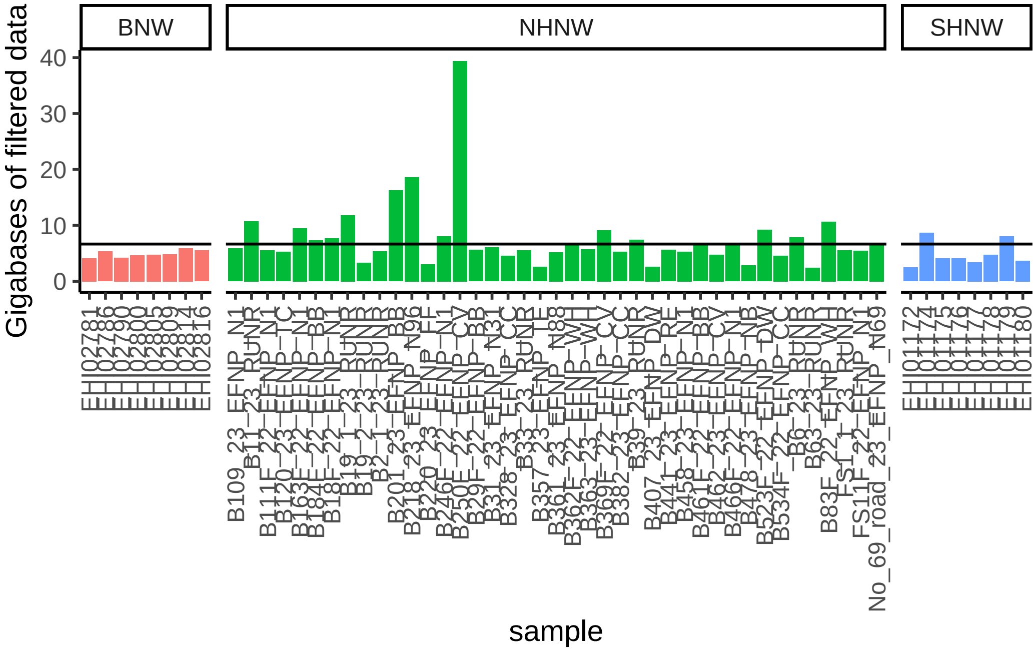
This report details the quality of the metagenomic data (from faecal samples) for all three different wombat species:

- *Lasiorhinus latifrons* (Southern Hairy-nosed Wombat;) [**SHNW**]
- *Lasiorhinus krefftii* (Northern Hairy-nosed Wombat; Yaminon) [**NHNW**]
- *Vombatus ursinus* (Bare-nosed Wombat) [**BNW**]

Preprocessing report

Here are the results for the preprocessing of the data, namely, fastp to trim/quality filter the paired reads and the mapping of filtered reads to the bare-nosed wombat genomes. (*There are currently no hairy-nosed wombat nuclear genomes as of 22/01/2025*).

```r
library(tidyverse)
library(scales)
library(ggtext)
library(ggforce)
library(ggdist)
library(patchwork)

ppr <- read_csv("../data/all_wombats/shnw_bnw_sample_info.csv") %>%
  select(EHI_number, metagenomic_bases, host_percent, project) %>%
  rename(sample = EHI_number,
         host_percentage = host_percent,
         host_species = project) %>%
  mutate(host_species = str_replace(host_species,
                                    "Lasiorhinus latifrons",
                                    "SHNW"),
         host_species = str_replace(host_species,
                                    "Vombatus ursinus",
                                    "BNW"),
         host_percentage = str_replace(host_percentage, "%", ""),
         host_percentage = as.numeric(host_percentage) / 100)
ppr_nhnw <- read_delim("../data/NHNW/preprocessing_report.tsv") %>%
  select(sample, metagenomic_bases, host_percentage) %>%
  mutate(host_species = "NHNW")

ppr_merged <- ppr %>%
  bind_rows(., ppr_nhnw)

mean_gbp <- mean(ppr_merged$metagenomic_bases) / 1e9
mean_host <- mean(ppr_merged$host_percentage)
max_host <- max(ppr_merged$host_percentage)

ppr_merged %>%
  ggplot(aes(y = metagenomic_bases / 1000000000,
             x = sample,
             fill = host_species)) +
  facet_grid(~host_species, scales = "free", space = "free") +
  geom_histogram(stat = "identity") +
  geom_hline(yintercept = mean(ppr_merged$metagenomic_bases) / 1e9) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
    legend.position = "none"
  ) +
  ylab("Gigabases of filtered data")
```

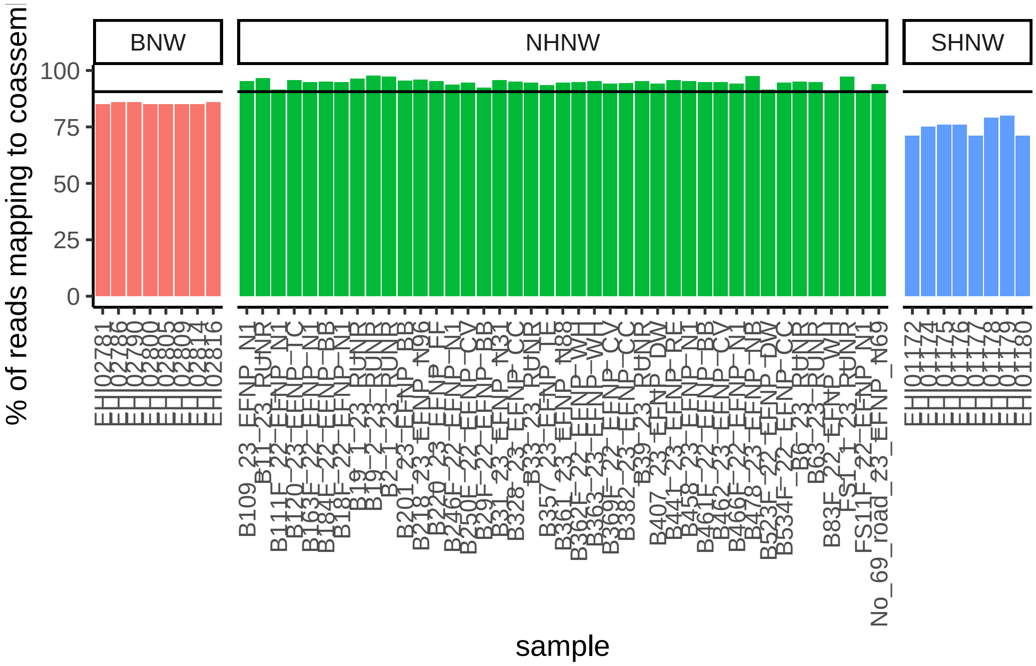


Horizontal line indicates the mean 6.7 Gbp of data for the samples. There was minimal mapping of reads to the host genome: max = 23% mean = 1.04%.

Coassembly report

Here's a summary of the coassembly and binning of all samples together by host species.

```
coasb <- read_delim("../data/all_wombats/AB_assembly_binning.csv") %>%
  select(EHI_number, num_bins, assembly_mapping_percent, Host) %>%
  rename(host_species = Host, sample = EHI_number) %>%
  mutate(host_species = str_replace(host_species,
                                    "Lasiorhinus latifrons",
                                    "SHNW"),
         host_species = str_replace(host_species,
                                    "Vombatus ursinus",
                                    "BNW"))
coasb_nhnw <- read_delim("../data/NHNW/nhnw_coassembly_summary.tsv") %>%
  select(sample, num_bins, assembly_mapping_percent) %>%
  mutate(host_species = "NHNW")

coasb_merged <- coasb %>%
  bind_rows(., coasb_nhnw) %>%
  filter(num_bins > 0)


coasb_mapping_mean <- percent(mean(coasb_merged$assembly_mapping_percent/100), accuracy = 0.
coasb_mapping_max <- percent(max(coasb_merged$assembly_mapping_percent/100), accuracy = 0.1)
coasb_mapping_min <- percent(min(coasb_merged$assembly_mapping_percent/100), accuracy = 0.1)

coasb_merged %>%
  ggplot(aes(y = assembly_mapping_percent,
             x = sample,
             fill = host_species)) +
  facet_grid(~host_species, scales = "free", space = "free") +
  geom_histogram(stat = "identity") +
  geom_hline(yintercept = mean(coasb_merged$assembly_mapping_percent)) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
    legend.position = "none"
  ) +
  ylab("% of reads mapping to coassemblies")
```

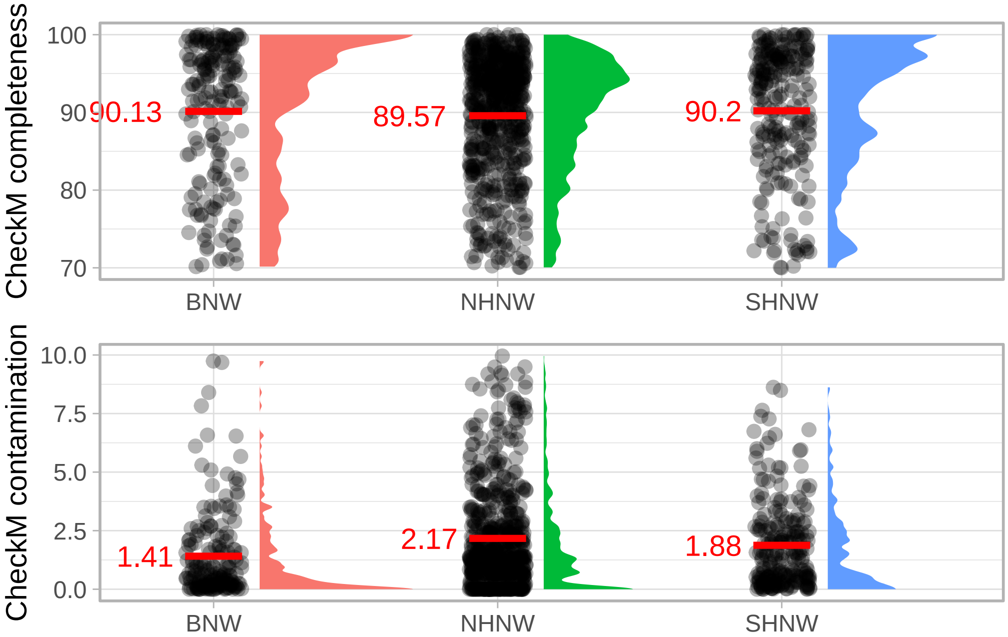


Overall the coassembly captured most of the metagenomic reads, with a mean mapping rate of 90.6% (max = 97.6%; min = 71.0%). Binning of these contigs yielded 600, 301, and 208 metagenome assembled genomes (MAGs) for the NHNW, SHNW, and BNW, respectively. Keep in mind that the microbial fractions may be different between species (more on this later).

Final MAG quality report

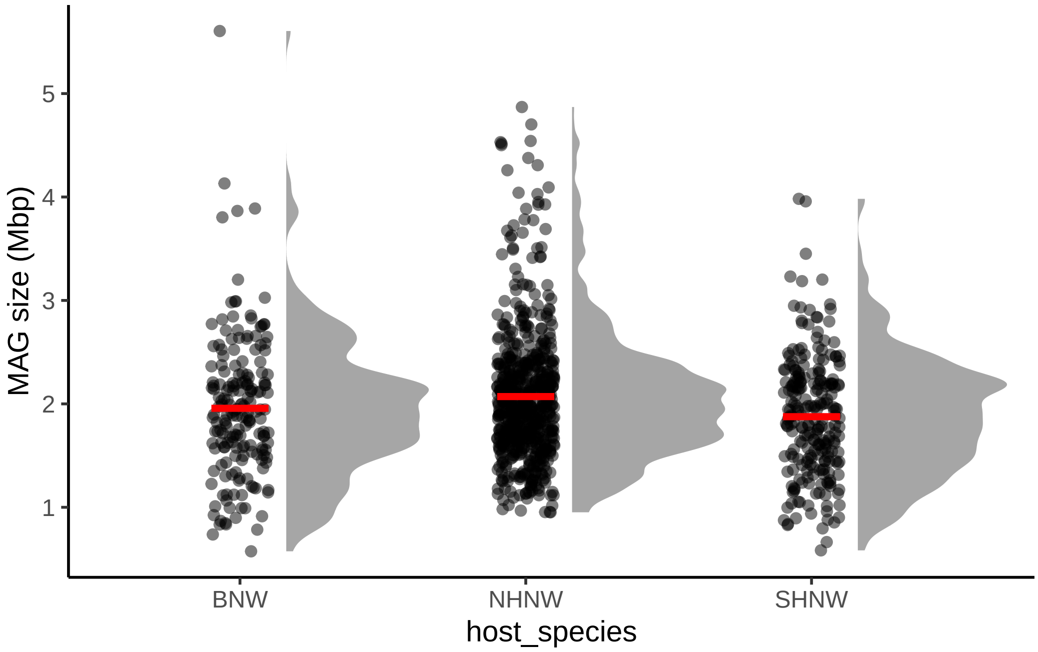
Here are some stats for the quality of the final MAGs.

```r
shnw_bnw_mags <- read_delim("../data/all_wombats/mag_info.csv") %>%
  select(mag_name, completeness, contamination, GC, size, host_species) %>%
  mutate(host_species = str_replace(host_species,
                                    "Lasiorhinus latifrons",
                                    "SHNW"),
         host_species = str_replace(host_species,
                                    "Vombatus ursinus",
                                    "BNW"),
         GC = str_replace(GC, "%", ""),
         GC = as.numeric(GC) / 100)

nhnw_mags <- read_delim("../data/NHNW/nhnw_metawrap_70_10_bins.stats") %>%
  select(bin, completeness, contamination, GC, size) %>%
  rename(mag_name = bin) %>%
  mutate(host_species = "NHNW")

mags <- shnw_bnw_mags %>%
  bind_rows(., nhnw_mags) %>%
  filter(completeness >= 70)

comp <- mags %>%
  ggplot(aes(y = completeness, x = host_species,
             fill = host_species)) +
  stat_halfeye(
    adjust = .5,
    width = .6,
    .width = 0,
    justification = -.3,
    point_colour = NA
  ) +
  geom_point(
    size = 2,
    alpha = .3,
    position = position_jitter(seed = 1, width = .1)
  ) +
  stat_summary(
    fun = "mean",
    geom = "crossbar",
    colour = "red",
    width = 0.2
  ) +
  stat_summary(
    fun = "mean",
    geom = "text",
    aes(label = round(..y.., 2)),
    hjust = 1.7,
    colour = "red"
  ) +
  theme_light() +
  theme(
    legend.box.background = element_rect(size = 0.5),
    legend.margin = margin(-5, 5, 0, 0),
    axis.title.x = element_blank(),
    legend.position = "none") +
  coord_cartesian(xlim = c(1.2, NA), clip = "off") +
  labs(y = "CheckM completeness")

cont <- mags %>%
  ggplot(aes(y = contamination, x = host_species,
             fill = host_species)) +
  stat_halfeye(
    adjust = .5,
    width = .6,
    .width = 0,
    justification = -.3,
    point_colour = NA
  ) +
  geom_point(
    size = 2,
    alpha = .3,
    position = position_jitter(seed = 1, width = .1)
  ) +
  stat_summary(
    fun = "mean",
    geom = "crossbar",
    colour = "red",
    width = 0.2,
  ) +
  stat_summary(
    fun = "mean",
    geom = "text",
    aes(label = round(..y.., 2)),
    hjust = 1.7,
    colour = "red"
  ) +
  theme_light() +
  theme(
    legend.box.background = element_rect(size = 0.5),
    legend.margin = margin(-5, 5, 0, 0),
    axis.title.x = element_blank(),
    legend.position = "none") +
  coord_cartesian(xlim = c(1.2, NA), clip = "off") +
  labs(y = "CheckM contamination")

comp / cont
```



Most MAGs are of decent quality.

```
mags %>%
  ggplot(aes(x = host_species, y = size / 1000000)) +
  geom_jitter(width = 0.1, height = 0, alpha = 0.5) +
  stat_halfeye(
    adjust = .5,
    width = .6,
    .width = 0,
    justification = -.3,
    point_colour = NA
  ) +
  stat_summary(
    fun = "mean",
    geom = "crossbar",
    colour = "red",
    width = 0.2,
  ) +
  theme_classic() +
  ylab("MAG size (Mbp)")
```



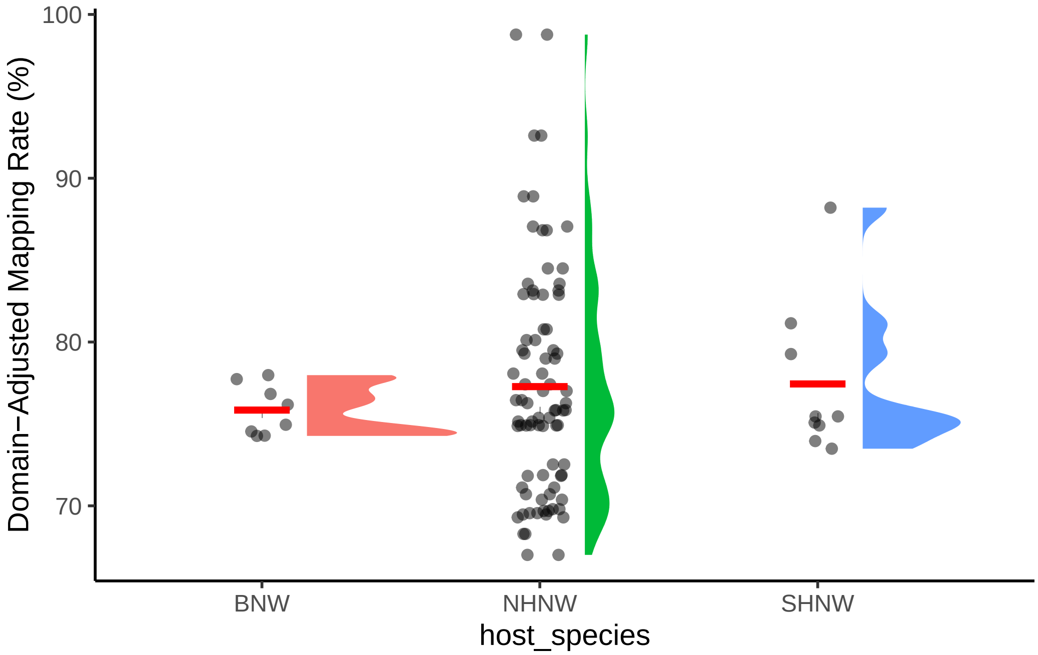
The mean MAG size is ~2 Mbp. This is pretty low, but cool, as it suggests that most of these bacteria probably can't live outside the NHNW gut!

How well did we capture the metagenomic samples?

Using ‘SingleM microbial_fraction‘, we can estimate how much prokaryotic DNA is in our metagenomes. We can then compare this to the mapping rate to calculate a **D**omain-**A**djusted **M**apping **R**ate (**DAMR**). This lets us know how well we've captured the prokaryotic community using our assembly/binning. For more info, see our paper describing the method: https://www.biorxiv.org/content/10.1101/2024.05.16.594470v1

```
shnw_bnw_smf <- read_csv("../data/all_wombats/shnw_bnw_sample_info.csv") %>%
  select(EHI_number, singlem_fraction, project) %>%
  rename(host_species = project, sample = EHI_number) %>%
  mutate(host_species = str_replace(host_species,
                                    "Lasiorhinus latifrons",
                                    "SHNW"),
         host_species = str_replace(host_species,
                                    "Vombatus ursinus",
                                    "BNW"))

shnw_bnw_mapping <- read_delim("../data/all_wombats/mag_mapping_rate.txt") %>%
  pivot_longer(., !Genome) %>%
  filter(Genome == "unmapped") %>%
  mutate(name = str_replace_all(name, " Relative Abundance \\(%\\)", ""),
           mapped = 100 - value) %>%
  inner_join(., shnw_bnw_smf, by = join_by(name == sample)) %>%
  mutate(DAMR = mapped / as.numeric(singlem_fraction))


smf_nhnw <- read_delim("../data/NHNW/smf_nhnw.tsv") %>%
  mutate(sample = str_replace_all(sample, "_M_1", "")) %>%
  rename(singlem_fraction = read_fraction) %>%
  mutate(singlem_fraction = as.numeric(singlem_fraction))

mag_mapping_nhnw <- read_delim("../data/NHNW/mag_mapping_rate.txt") %>%
  pivot_longer(., !Genome) %>%
  filter(Genome == "unmapped") %>%
    mutate(name = str_replace_all(name, " Relative Abundance \\(%\\)", ""),
           mapped = 100 - value,
           host_species = "NHNW") %>%
  inner_join(., smf_nhnw, by = join_by(name == sample)) %>%
  mutate(DAMR = mapped / as.numeric(singlem_fraction)) %>%
  select(Genome, name, value, mapped, singlem_fraction, host_species, DAMR)

damr <- shnw_bnw_mapping %>%
```

```r
  bind_rows(., mag_mapping_nhnw)

mean_damr <- percent(mean(damr$DAMR), accuracy = 0.1)

damr %>%
  ggplot(aes(x = host_species, y = DAMR * 100, fill = host_species)) +
  geom_jitter(width = 0.1, height = 0, alpha = 0.5) +
  stat_halfeye(
    adjust = .5,
    width = .6,
    .width = 0,
    justification = -.3,
    point_colour = NA
  ) +
  stat_summary(
    fun = "mean",
    geom = "crossbar",
    colour = "red",
    width = 0.2,
  ) +
  theme_classic() +
  theme(legend.position = "none") +
  ylab("Domain-Adjusted Mapping Rate (%)")
```



Overall, we've captured most of the prokaryote DNA in the samples (mean 77.2%). This is pretty decent considering the complexity of these microbial communities!