# R-code for 'Heterozygosity at neutral and immune loci does not influence neonatal mortality due to microbial infection in Antarctic fur seals'

*Vivienne Litzke, Meinolf Ottensmann, Jaume Forcada & Joseph I. Hoffman*

*December 7, 2018*

### Preface

This document provides all the `R code` used for our paper. Both the Rmarkdown file and the data can be downloaded from the accompanying GitHub repository on (URL TO GITHUB) as a zip archive containing all the files. Our data originates from samples collected from a colony of Antarctic fur seals (*Arctocephalus gazella*) at Bird Island, South Georgia between the years of 2000 and 2014. We investigated the effects of neutral and immune gene heterozygosity on early mortality due to bacterial infection using the `inbreedR` package.[1]

---

### Download packages and libraries

In order to repeat analyses presented in this manuscript a number of packages that extend the functionalities of base `R` are required. These can be installed using the code shown below.

```
install.packages('inbreedR')
install.packages("Rcpp")
install.packages("readxl")
install.packages("ggplot2")
install.packages("gridExtra")
install.packages("stringi", repos="http://cran.rstudio.com/", dependencies=TRUE)
install.packages("fansi")
install.packages("adegenet")
install.packages("AICcmodavg")
install.packages("raster")
install.packages("reshape2")
source("https://bioconductor.org/biocLite.R")
biocLite("qvalue")
```

```
library(inbreedR)
library(readxl)
library(magrittr)
library(ggplot2)
library(grid)
library(gridExtra)
library(AICcmodavg)
library(Matrix)
library(lme4)
library(qvalue)
```

[1] Stoffel, M. A., Esser, M., Kardos, M., Humble, E., Nichols, H., David, P., & Hoffman, J. I. (2016). inbreedR: an R package for the analysis of inbreeding based on genetic markers. Methods in Ecology and Evolution, 7(11), 1331-1339.

```
library(adegenet)
library(reshape2)
```

In order to use `inbreedR`, the working format is typically an *individual x loci* matrix, where rows represent individuals and every two columns represent a single locus. If an individual is heterozygous at a given locus, it is coded as 1, whereas a homozygote is coded as 0, and missing data are coded as NA.

The first step is to read the data from an excel file. Our original table includes, plate number, well number, species, id, year, health status (represented by a binomial with 0 for healthy and 1 for infected), birth weight, and the following markers (a and b for alleles).

```
## reda data
seals <- readxl::read_excel("data/genotypes_raw.xlsx", skip = 1)[1:78,]
## express alleles as numerals
seals[8:ncol(seals)] <- lapply(seals[8:ncol(seals)], as.numeric)
```

Here is an example of what the data frame looks like:

```
head(seals[1:6,4:12])
```

```
## # A tibble: 6 x 9
##   ID    Year  `Health status` Birthweight Agt47.a Agt47.b Agt10.a Agt10.b
##   <chr> <chr> <chr>           <chr>         <dbl>   <dbl>   <dbl>   <dbl>
## 1 AGP0~ 2000  0               5.09999999~     237     245     213     213
## 2 AGP0~ 2000  1               4.8             241     245     213     213
## 3 AGP0~ 2001  1               4.8             241     241     213     213
## 4 AGP0~ 2001  0               4.45            237     241     213     215
## 5 AGP0~ 2002  0               4.59999999~     237     241     213     213
## 6 AGP0~ 2002  1               4.05            245     245     213     213
## # ... with 1 more variable: Agi11.a <dbl>
```

Since demographic data is present in the beginning of the data frame, we will start our new genotype file from the 8th column onwards. The function `convert_raw` converts a common format for genetic markers (two columns per locus) into the `inbreedR` working format. Afterwards, `check_data` allowes to test whether the genotype data frame has the correct format for subsequent analyses using `inbreedR` functions.

```
seals_geno <- convert_raw(seals[8:ncol(seals)])
check_data(seals_geno, num_ind = 78, num_loci = 61)
```

---

### Separate loci by marker type

Divide the neutral and immune markers from their respective columns in the adjusted inbreedR format, and compute standard multilocus heterozygosity (sMLH).[2]

```
immune_markers <- seals_geno[, 1:13]
neutral_markers <- seals_geno[, 14:61]

all_het <- sMLH(seals_geno)
neutral_het <- sMLH(neutral_markers)
immune_het <- sMLH(immune_markers)
```

[2]Coltman, D. W. and J. Slate. 2003. Microsatellite measures of inbreeding: a meta-analysis. Evolution 57:971–983.

### Create and reshape dataframe

Take out id, health, marker types, and birth weight as variables.

```
birthweight <- as.numeric(as.character(seals[["Birthweight"]]))

sealdata <- data.frame(id = seals[[4]],  health = factor(seals[[6]]),
                       All = all_het, Neutral = neutral_het, Immune = immune_het)

sealdataweight <- data.frame(id = seals[[4]],  health = factor(seals[[6]]), birthweight,
                             All = all_het, Neutral = neutral_het, Immune = immune_het)

sealdata_reshaped <- reshape2::melt(sealdata)
sealdataframe_plusyear <- cbind(sealdataweight, year = as.numeric(seals[[5]]))
```

---

### Calculate $g_2$

$g_2$ is a proxy for identity disequilibrium. It is a measure of two-locus disequilibrium, which quantifies the extent to which heterozygosities are correlated across pairs of loci.[3] This allows us to take a look at our neutral marker heterozygosity to determine if there is variation in inbreeding in the population.

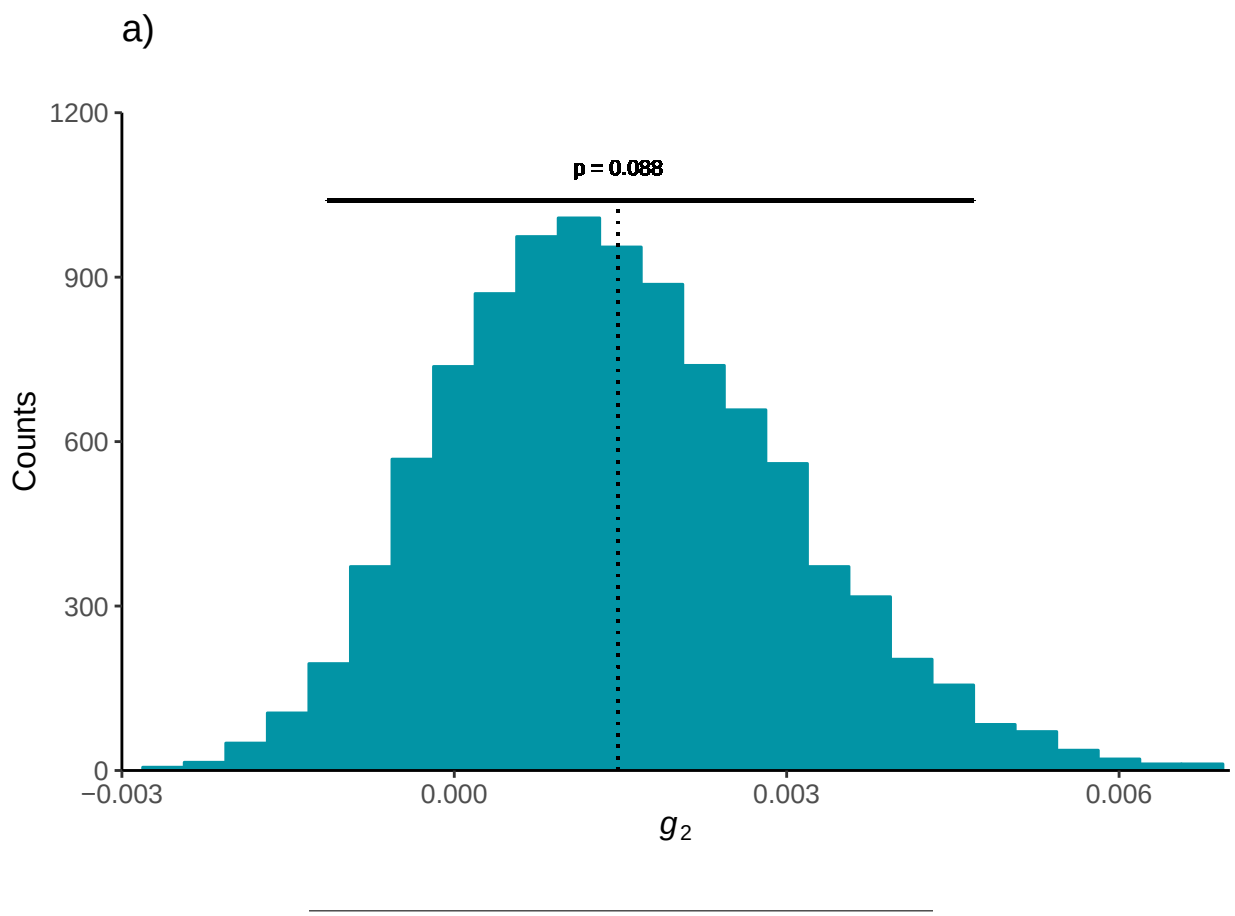
```
g2_neutral <- g2_microsats(neutral_markers, nperm = 9999, nboot = 9999)
g2_neutral_bs <- data.frame(bs = g2_neutral$g2_boot,
                            lcl = g2_neutral$CI_boot[[1]],
                            ucl = g2_neutral$CI_boot[[2]],
                            g2  = g2_neutral$g2,
                            p = g2_neutral$p_val)
```

Plot the distribution of g2 estimates:

```
g2_neutral_bs_histogram <-
  ggplot2::ggplot() +
  theme_classic() +
  geom_histogram(binwidth = 0.000375, data = g2_neutral_bs, aes(x = bs),
                 color = "#0294A5",
                 fill = "#0294A5") +
  geom_errorbarh(data = g2_neutral_bs,
                 aes(y = 1040, x = g2, xmin = lcl, xmax = ucl),
                 color = "black", size = 0.7, linetype = "solid") +
  geom_linerange(data = g2_neutral_bs,
                 aes(ymin = 0, ymax = 1040, x = g2),
                 linetype = 'dotted') +
  theme(text = element_text(size = 12),
        panel.border = element_blank(),
        strip.background =element_rect(fill = "white", colour = "white"),
        strip.text = element_text(colour = 'white'),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  facet_wrap(~p) +
  ylab("Counts") +
  labs(x = expression(italic(g)["2"])) +
  ggtitle("a)") +
```

[3] David, P., Pujol, B., Viard, F., Castella, V., & Goudet, J. (2007). Reliable selfing rate estimates from imperfect population genetic data. Molecular ecology, 16(12), 2474-2487.

```
  scale_y_continuous(expand = c(0,0), limits = c(0,1200)) +
  scale_x_continuous(limits = c(-0.003, 0.007),
                     breaks = seq(-0.003, 0.009, 0.003),
                     expand = c(0,0)) +
  annotate("text", x = g2_neutral_bs$g2, y = 1100,
           label = paste0('p = ', round(g2_neutral_bs$p, 3)),
           family = theme_get()$text[["family"]],
           size = theme_get()$text[["size"]]/4)
plot(g2_neutral_bs_histogram)
```

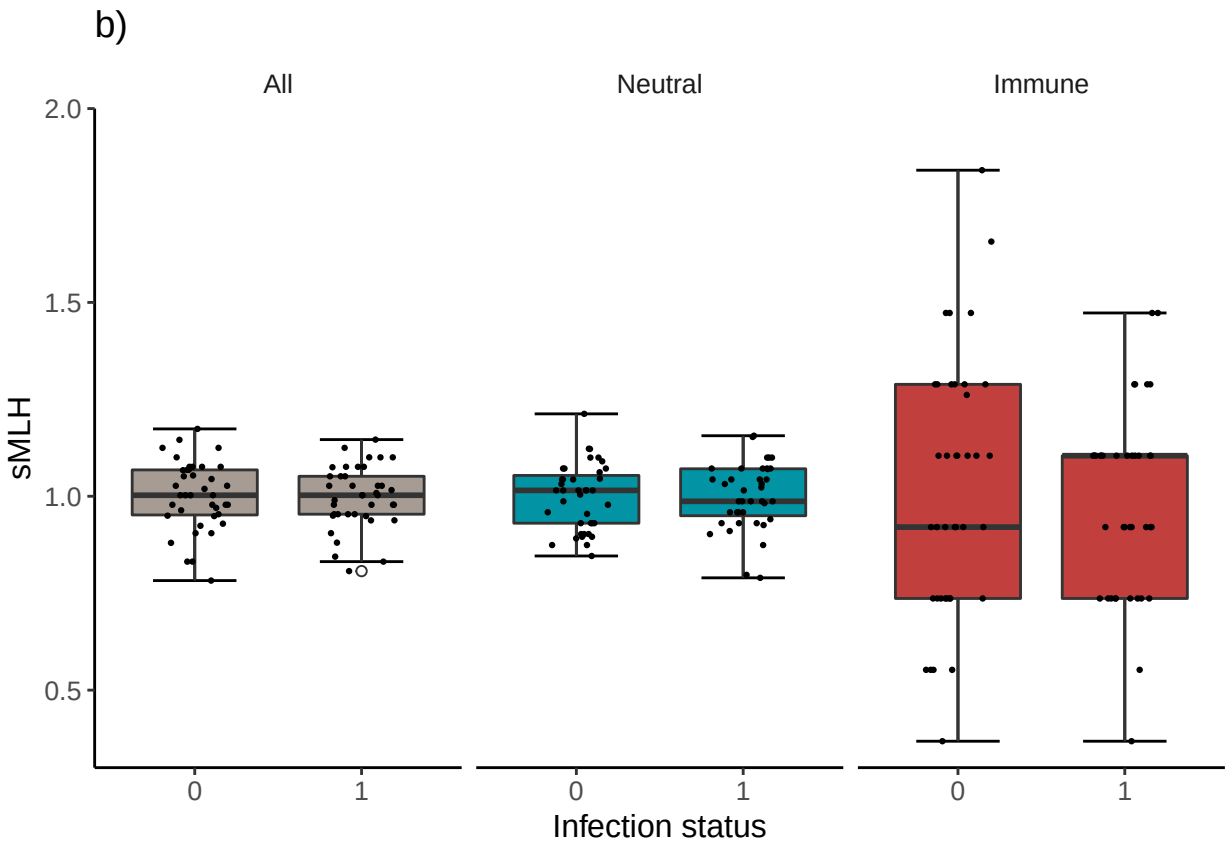



**Plot heterozygosity among marker sets**

In order to visualize sMLH for all, neutral, and immune markers, create the following box-plot:

```
het_plot <-
  ggplot(data = sealdata_reshaped, aes(x = health, y = value, fill = variable)) +
  stat_boxplot(aes(x = health, y = value),
               geom = 'errorbar', linetype = 1, width = 0.5) +
  geom_boxplot( aes(x = health, y = value), outlier.shape = 1) +
  geom_jitter(shape = 16, position = position_jitter(0.2), size = .8) +
  theme_classic() +
  theme(legend.position = "none",
        panel.border = element_blank(),
        strip.background = element_blank(),
```

```
        text = element_text(size = 12),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  xlab("Infection status") +
  ylab("sMLH") +
  ggtitle("b)") +
  scale_fill_manual(values =  c("#A79C93", "#0294A5", "#C1403D")) +
  facet_wrap(~variable, nrow = 1) +
  scale_y_continuous(limits = c(0.3, 2),
                     expand = c(0,0))
plot(het_plot)
```




---

**Calculate heterozygosity for each individual locus**

As we have previously looked at genome-wide effects, it may be of interest to look for local effects. Therefore, we wanted to examine the heterozygosity for each locus. First, define a function tomput the confidence interval:

```
confidence_interval <- function(vector) {
  vec_sd <- sd(vector)        # standard deviation of sample
  n <- length(vector)         # sample size
  vec_mean <- mean(vector)    # mean of sample
  error <- qt((.95 + 1)/2, df = n - 1) * vec_sd / sqrt(n)              # error according to t distribu
  result <- c("lower" = vec_mean - error, "upper" = vec_mean + error)   # confidence interval as a vect
```

```
  return(result)
}
```

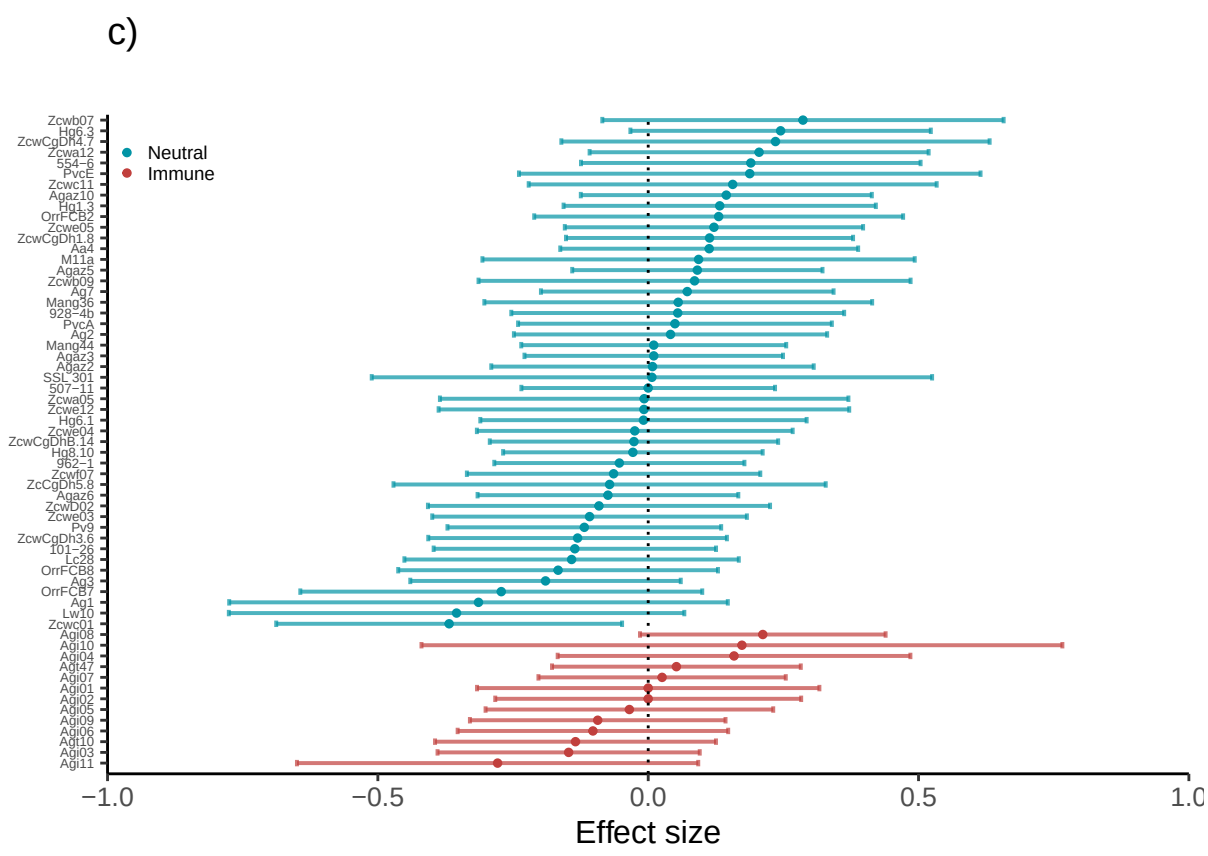
Calculate the heterozygosity for each locus, and use a regression on infection status:

```
## calcaute sMLH
het_per_locus <- apply(seals_geno, 2, sMLH)
## add factors
df <-  cbind(sealdataframe_plusyear, seals_geno)
## add marker type as names to the data.frame
names(df)[6:66] <- c(paste0("Immune", 1:13), paste0("Neutral", 1:48))

lm_by_loc <- lapply(1:61, function(x) {
  value <- df[,x + 7]                                   # extract data of given marker x
  # res <- summary(lme4::lmer(as.numeric(df$health) ~ value + (1|df$year)))   # do regression
  # conf <- confint(lme4::lmer(as.numeric(df$health) ~ value + (1|df$year)))
  res <- summary(lm(as.numeric(df$health) ~ value))   # do regression
  conf <- confint(lm(as.numeric(df$health) ~ value))
  f <- res$fstatistic
  pf(f[1], f[2], f[3], lower=FALSE)
  out <- data.frame(beta = res$coefficients[2,1],
                    lcl = conf[2,1],
                    ucl = conf[2,2])
}) %>%
  do.call("rbind",.) %>%
  cbind(., data.frame(names = colnames(seals)[seq(8, ncol(seals), 2)] %>%
                        substring(., first = 1, last = nchar(.) - 2),
                      type = c(rep("Immune", 13),rep("Neutral", 48)),
                      dummy = ""))

 # order by effect size
lm_by_loc <- lm_by_loc[with(lm_by_loc, order(type, beta, decreasing = F)),]
lm_by_loc$num <- 1:61

## create data frame to label effects
names_df <-  data.frame(label = lm_by_loc$names,
                        num = lm_by_loc$num)
```

Create a plot to feature each loci and their relevant effect sizes:

```
het_by_loci_plot <- ggplot(lm_by_loc, aes(x = num, y = beta, col = type)) +
  geom_errorbar(aes(ymin = lcl, ymax = ucl),
                width = 0.6, alpha = 0.7, size = 0.7) +
  geom_point(size = 1) +
  scale_x_continuous(expand = c(0,0), breaks = 1:61, labels = names_df$label) +
  scale_y_continuous(expand = c(0,0)) +
  geom_hline(yintercept = 0, linetype = "dotted") +
  coord_flip(xlim = c(0, 61.5), ylim = c(-1,1)) +
  scale_color_manual(values = c("#C1403D","#0294A5"),
                     name = "",
                     breaks = c("Neutral", "Immune"),
                     labels = c("Neutral", "Immune")) +
  theme_classic() +
  xlab("") +
  ylab("Effect size") +
```

```
  theme(legend.justification = c(0,1),
        legend.position = c(0,1.05),
        legend.background = element_rect(fill = NA),
        text = element_text(size = 12),
        axis.text.y = element_text(size = 5),
        legend.text = element_text(size = 7),
        panel.border = element_blank(),
        strip.background = element_rect(fill = "white", colour = "white"),
        strip.text = element_text(colour = 'white'),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  guides(color = guide_legend(
    keywidth = 0.05,
    keyheight = 0.05,
    default.unit = "inch")) +
  facet_wrap(~dummy) +
  ggtitle("c)")
het_by_loci_plot
```




To look for local effects between effect sizes of the neutral and immune loci, use a Wilcoxon test:

```
wilcox.test(lm_by_loc$beta[1:13],lm_by_loc$beta[14:61])
```

```
##
##  Wilcoxon rank sum test
##
## data:  lm_by_loc$beta[1:13] and lm_by_loc$beta[14:61]
## W = 285, p-value = 0.6445
```

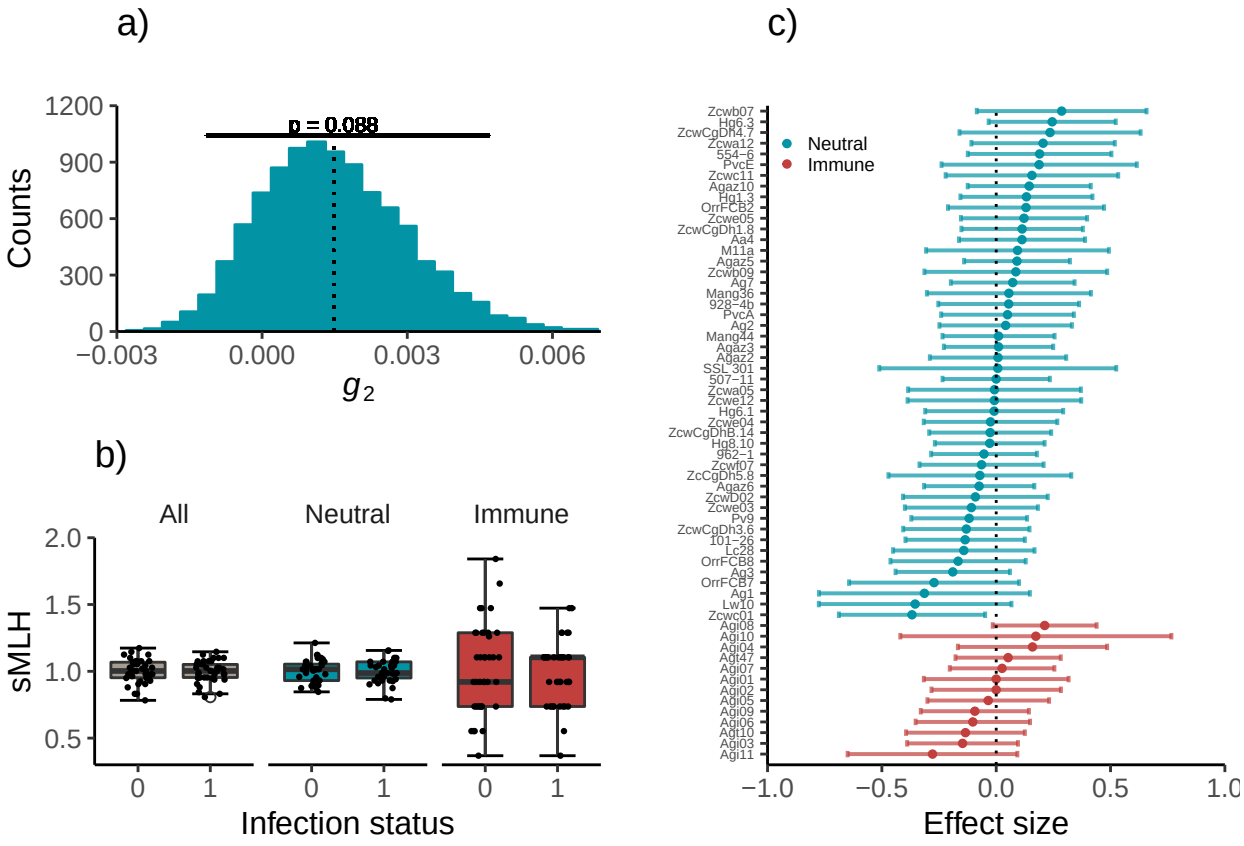
```
## alternative hypothesis: true location shift is not equal to 0
```

---

### Create the final combined figure

To create a combination plot of all figures (as in the manuscript):

```
lay <- rbind(c(1,3),
             c(2,3))

combo_plot <- grid.arrange(g2_neutral_bs_histogram,
                           het_plot,
                           het_by_loci_plot, ncol = 3, layout_matrix = lay)
```




```
combo_plot
```

---

### Modeling

To test for associations between microsatellite heterozygosity and death from bacterial infection, we constructed several alternative generalized linear mixed-models (GLMMs) incorporating relevant predictor variables and quantified their relative support using AICc weights within a multi-model inference framework. All of the models had pup survival as a binary response variable (coded as 0 = alive and 1 = dead) and included year as

a random effect to statistically control for any variation in survivorship attributable to inter-annual variation. The following GLMMs were considered:

```
models <- list(
  glmer(health ~  1 + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  All + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  Immune + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  Neutral + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  1 + birthweight + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  All + birthweight + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  Immune + birthweight + (1|year), data = sealdataframe_plusyear, family = 'binomial'),
  glmer(health ~  Neutral + birthweight + (1|year), data = sealdataframe_plusyear, family = 'binomial'))
names(models) <- paste0("m", 1:length(models))

## model selection
pander::pandoc.table(AICcmodavg::aictab(models, second.ord = T))
```

```
##
## ------------------------------------------------------------------------------------
##       Modnames    K    AICc    Delta_AICc    ModelLik    AICcWt       LL      Cum.Wt
## -------- ---------- --- ------- ------------ ---------- --------- -------- --------
##  **1**        m1       2    112.3        0            1        0.3441    -54.07    0.3441
##
##  **5**        m5       3    114.1      1.777        0.4113     0.1415    -53.87    0.4856
##
##  **3**        m3       3    114.3      1.985        0.3707     0.1276    -53.98    0.6131
##
##  **2**        m2       3    114.4      2.112        0.3478     0.1197    -54.04    0.7328
##
##  **4**        m4       3    114.4      2.151        0.3411     0.1174    -54.06    0.8502
##
##  **7**        m7       4     116       3.708        0.1566     0.05388   -53.73    0.904
##
##  **6**        m6       4    116.2      3.884        0.1434     0.04935   -53.81    0.9534
##
##  **8**        m8       4    116.3      3.997        0.1355     0.04662   -53.87      1
## ------------------------------------------------------------------------------------
```

These included 'null models' without any genetic effects (models i and v) as well as models that included sMLH combined over all loci or calculated separately for the neutral versus immune loci. Models v to viii also included pup birth weight (in kg) to incorporate any potential effects of body size on survivorship. All of the models were specified using the glmer function of the package "lme4" with a binomial error structure.[4] Using the R package `AICcmodavg`, the most parsimonious model was selected based on the delta AICc value, which compares weights as a measure of the likelihood of a particular model.[5] The best supported model has $\Delta$ AICc = 0 and a difference of two or more units was applied as a criterion for choosing one model over a competing model.[6]

Apply a false discovery rate correction for a table of p-values.

```
pval <- read.table("pvalues.txt",header = F, sep = ",") %>% as.vector() %>% .[[1]]
qobj <- qvalue(pval)
```

[4] Bates, D., Mächler, M., Bolker, B., & Walker, S. (2014). Fitting linear mixed-effects models using lme4. arXiv preprint arXiv:1406.5823.

[5] Mazerolle, M. J., & Mazerolle, M. M. J. (2017). Package 'AICcmodavg'. R package.

[6] Anderson, D. R., & Burnham, K. P. (2002). Avoiding pitfalls when using information-theoretic methods. The Journal of Wildlife Management, 912-918.

```
qvalues <- qobj$qvalues
pi0 <- qobj$pi0
lfdr <- qobj$lfdr
summary(qobj)

df <- data.frame(p = qobj$pvalues,
                 q = qobj$qvalues)
#view(df)
```

---

# Supplementary Data

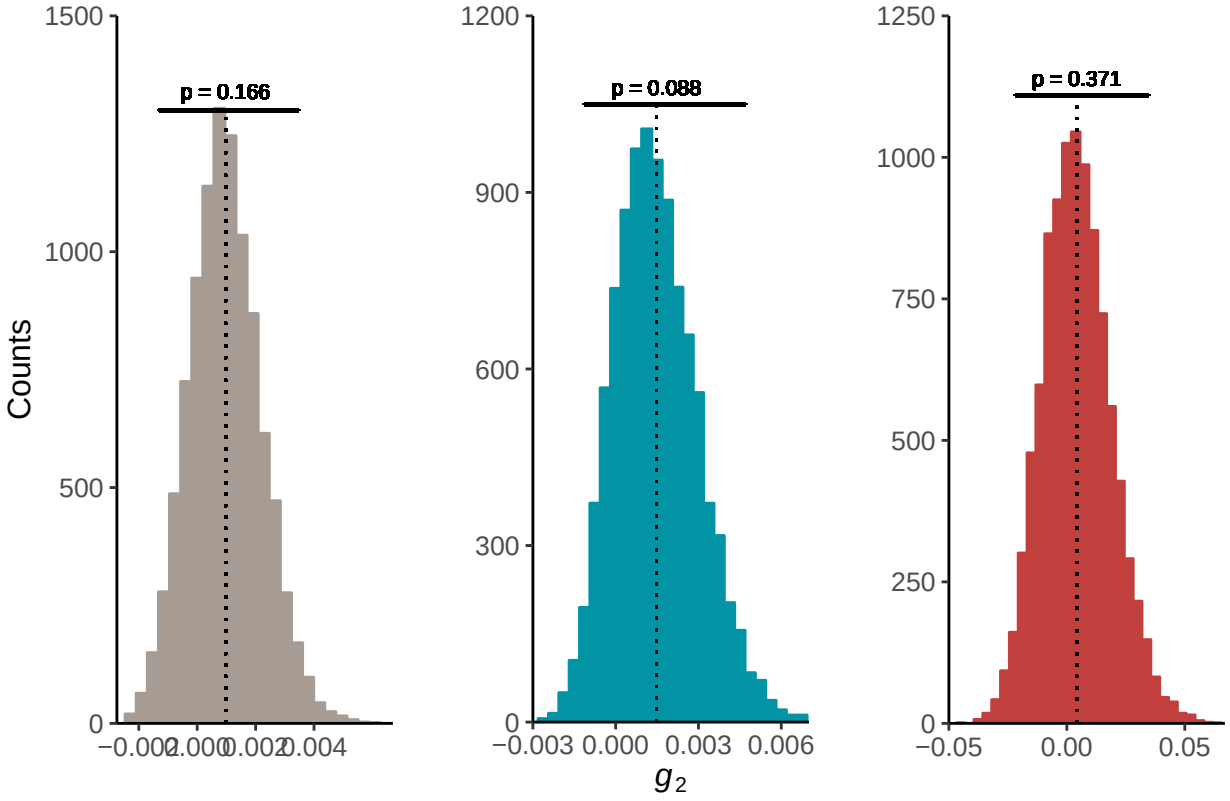
### (A) $g_2$ for all marker sets.

If there is interest to see if a variation in inbreeding can be captured among different marker sets, calculate $g_2$ for all and immune microsats and create histograms:

```
g2_all <- g2_microsats(cbind(neutral_markers, immune_markers), nperm = 9999, nboot = 9999)
g2_all_bs <- data.frame(bs = g2_all$g2_boot,
                        lcl = g2_all$CI_boot[[1]],
                        ucl = g2_all$CI_boot[[2]],
                        g2  = g2_all$g2,
                        p = g2_all$p_val)

g2_immune <- g2_microsats(immune_markers, nperm = 9999, nboot = 9999)
g2_immune_bs <- data.frame(bs = g2_immune$g2_boot,
                           lcl = g2_immune$CI_boot[[1]],
                           ucl = g2_immune$CI_boot[[2]],
                           g2  = g2_immune$g2,
                           p = g2_immune$p_val)

all_graphs_g2_neutral_bs_histogram <-
  ggplot2::ggplot() +
  theme_classic() +
  geom_histogram(binwidth = 0.000375, data = g2_neutral_bs, aes(x = bs),
                 color = "#0294A5",
                 fill = "#0294A5") +
  geom_errorbarh(data = g2_neutral_bs,
                 aes(y = 1050, x = g2, xmin = lcl, xmax = ucl),
                 color = "black", size = 0.7, linetype = "solid") +
  geom_linerange(data = g2_neutral_bs,
                 aes(ymin = 0, ymax = 1050, x = g2),
                 linetype = 'dotted') +
  theme(text = element_text(size = 12),
        panel.border = element_blank(),
        strip.background = element_rect(fill = "white", colour = "white"),
        strip.text = element_text(colour = 'white'),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  facet_wrap(~p) +
  ylab(" ") +
  labs(x = expression(italic(g)["2"])) +
  scale_y_continuous(expand = c(0,0), limits = c(0,1200)) +
```

```
  scale_x_continuous(limits = c(-0.003, 0.007),
                     breaks = seq(-0.003, 0.009, 0.003),
                     expand = c(0,0)) +
  annotate("text", x = g2_neutral_bs$g2, y = 1079,
           label = paste0('p = ', round(g2_neutral_bs$p, 3)),
           family = theme_get()$text[["family"]],
           size = theme_get()$text[["size"]]/4)

all_graphs_g2_all_bs_histogram <-
  ggplot2::ggplot() +
  theme_classic() +
  geom_histogram(binwidth = 0.00038, data = g2_all_bs, aes(x = bs),
                 color = "#A79C93",
                 fill = "#A79C93") +
  geom_errorbarh(data = g2_all_bs,
                 aes(y = 1300, x = g2, xmin = lcl, xmax = ucl),
                 color = "black", size = 0.7, linetype = "solid") +
  geom_linerange(data = g2_all_bs,  aes(ymin = 0, ymax = 1300, x = g2),
                 linetype = 'dotted') +
  theme(text = element_text(size = 12),
        panel.border = element_blank(),
        strip.background = element_rect(fill = "white", colour = "white"),
        strip.text = element_text(colour = 'white'),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  facet_wrap(~p) +
  ylab("Counts") +
  xlab(" ") +
  scale_y_continuous(expand = c(0,0), limits = c(0,1500)) +
  scale_x_continuous(limits = c(-0.00275, 0.0067),
                     breaks = c(-0.002, 0.000, 0.002, 0.004),
                     labels = c("-0.002","0.000","0.002","0.004"),
                     expand = c(0,0)) +
  annotate("text", x = g2_all_bs$g2, y = 1340,
           label = paste0('p = ', round(g2_all_bs$p, 3)),
           family = theme_get()$text[["family"]],
           size = theme_get()$text[["size"]]/4)

all_graphs_g2_immune_bs_histogram <-
  ggplot2::ggplot() +
  theme_classic() +
  geom_histogram(binwidth = 0.00375, data = g2_immune_bs, aes(x = bs),
                 color = "#C1403D",
                 fill = "#C1403D") +
  geom_errorbarh(data = g2_immune_bs,
                 aes(y = 1110, x = g2, xmin = lcl, xmax = ucl),
                 color = "black", size = 0.7, linetype = "solid") +
  geom_linerange(data = g2_immune_bs,  aes(ymin = 0, ymax = 1110, x = g2),
                 linetype = 'dotted') +
  theme(text = element_text(size = 12),
        panel.border = element_blank(),
        strip.background = element_rect(fill = "white", colour = "white"),
        strip.text = element_text(colour = 'white'),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
```

```
  facet_wrap(~p) +
  ylab(" ") +
  xlab(" ") +
  scale_y_continuous(expand = c(0,0), limits = c(0,1250)) +
  scale_x_continuous(limits = c(-0.05, 0.067),
                     breaks = seq(-0.05, 0.05, 0.05),
                     expand = c(0,0)) +
  annotate("text", x = g2_immune_bs$g2, y = 1139,
           label = paste0('p = ', round(g2_immune_bs$p, 3)),
           family = theme_get()$text[["family"]],
           size = theme_get()$text[["size"]]/3.8)
```




### (B) Calculate g2 for subsets of the data

Here, we repeat the estimation of g2 for each marker type and for the entire dataset

```
g2_neutral_resampled <- pbapply::pblapply(seq(4, 48, 4), function(x) { -->
  subs <- lapply(1:100, function(y) {
    rand <- sample(1:48, x, replace = FALSE)
    loci <- neutral_markers[, rand]
    g2 <- g2_microsats(loci, nperm = 0, nboot = 9999, verbose = F)
    df <- data.frame(bs = g2$g2_boot,
                     lcl = g2$CI_boot[[1]],
                     ucl = g2$CI_boot[[2]],
                     g2  = g2$g2,
                     p = g2$p_val)
```

```
    return(df[1,])
  }) %>% do.call("rbind", .)
  return(data.frame(g2 = mean(subs$g2),
                    lcl = confidence_interval(subs$g2)[1],
                    ucl = confidence_interval(subs$g2)[2]))
}) %>%  do.call("rbind", .)
g2_neutral_resampled$loci <- seq(4, 48, 4)
save(g2_neutral_resampled, file = "data/g2_neutral_resampled.RData")    # save the data

g2_neutral_resampled_plot <-
  ggplot(data = g2_neutral_resampled, aes(x = loci, y = g2)) +
  geom_line() +
  geom_point(size = 1.5) +
  geom_errorbar(aes(ymin = lcl,
                    ymax = ucl),
                width = 0.8, alpha = 0.7, size = 0.8, colour = "black") +
  geom_hline(yintercept = 0, linetype = "dotted") +
  theme_classic() +
  theme(legend.position = "none",
        panel.border = element_blank(),
        strip.background = element_blank(),
        text = element_text(size = 12),
        aspect.ratio = 1,
        axis.title.y = element_text(face = "italic"),
        plot.margin = grid::unit(c(2,2,2,2), 'mm')) +
  xlab("Number of loci") +
  labs(y = expression(italic(g)["2"])) +
  scale_x_continuous(expand = c(0,0), limits = c(0, 50))
```

### (C) Heat map

Next, we test for patterns in allelic richness among markers (i.e. immune vs neutral), developmental source (i.e. designed for Antarctic fur seals, phocids or otariids). Secondly, we evaluate the cross-amplification success of loci in two other species of pinnipeds, namely Grey seal and Nothern Elephant seal.

```
# Read and format genotypes
heatmap_df <- readxl::read_xlsx("data/genotypes_raw.xlsx", skip = 1)[, c(3, 8:ncol(seals))]

# Randomly select six individuals per species
heatmap_df <- heatmap_df[c(sample(which(heatmap_df[["Species"]] == "Fur seal"), size = 6, replace = F),
           sample(which(heatmap_df[["Species"]] == "Grey seal"), size = 6, replace = F),
           sample(which(heatmap_df[["Species"]] == "Northern Elephant seal"), size = 6, replace = F)),]

# Extract geno
marker_geno <- apply(heatmap_df[,-1], 2, as.character)

# Get names of loci
loci_names <- colnames(marker_geno)[seq(1, ncol(marker_geno), 2)] %>%
  substring(., first = 1, last = nchar(.) - 2)

# Define a vector of Immune marker names
immune_marker_names <- c("Agi01", "Agi02", "Agi03", "Agi04",
                         "Agi05", "Agi06", "Agi07", "Agi08",
```

```
                          "Agi09", "Agi10", "Agi11", "Agt10", "Agt47")

# Collapse information for each locus in one column
marker_geno <- lapply(seq(1, ncol(marker_geno), 2), function(x) {
  marker_geno[,x:(x + 1)] %>%
    apply(., 1, paste0, collapse = "/")
}) %>%
  do.call("cbind",.) %>%
  ## rename loci
  set_colnames(x = ., value = paste0("Locus", 1:61))

## set missing data to NA
marker_geno[which(marker_geno == "NA/NA")] <- NA

## convert to GENIND object
genind <- adegenet::df2genind(marker_geno, ploidy = 2, sep = "/", pop = heatmap_df[["Species"]] %>% as.
```

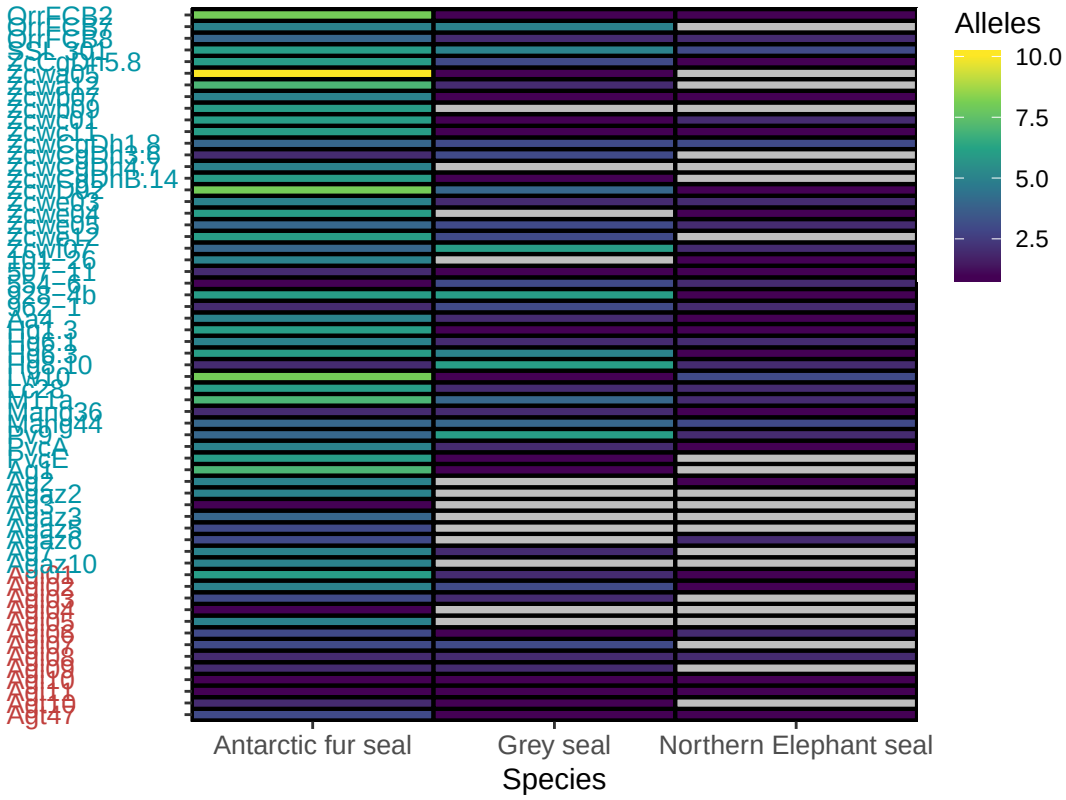
```
## Convert to GENPOP
genpop <- adegenet::genind2genpop(genind)
```

```
##
##  Converting data from a genind to a genpop object...
##
## ...done.
```

```
heatmap_df <- lapply(levels(genpop@loc.fac), function(i) {
  df.temp <- genpop@tab[,which(genpop@loc.fac == i)]   ## fetch data
  if (is.null(dim(df.temp))) {
    df.temp[df.temp > 0] <- 1
    df.temp[df.temp == 0] <- 0

  } else {
    df.temp <-   apply(df.temp, 2, function(x) ifelse(x > 0, 1, 0)) %>% ## presence/absence of allele
      rowSums(na.rm = T) ## count alleles
  }
  # return results
  return(data.frame(Species = names(df.temp),
                    Locus = i,
                    Alleles = df.temp))
}) %>%
  do.call("rbind", .)

## set zero to NA
heatmap_df[["Alleles"]][which(heatmap_df[["Alleles"]] == 0)] <- NA

heatmap_df[["Locus"]] <- factor(heatmap_df[["Locus"]], labels = loci_names)
heatmap_df[["Type"]] <- 'Neutral'
heatmap_df[["Type"]][which(heatmap_df[["Locus"]] %in% immune_marker_names)] <- 'Immune'

## sort by species
heatmap_df[["Species"]] <- factor(heatmap_df[["Species"]],
                          levels = c("Fur seal", "Grey seal", "Northern Elephant seal"),
                          labels = c("Antarctic fur seal", "Grey seal", "Northern Elephant seal"))
```

```
## define colours for marker types
col_key <- ifelse(levels(heatmap_df[["Locus"]]) %in% immune_marker_names, "#C1403D", "#0294A5")
```

```
plot <- ggplot(data = heatmap_df, aes(x = Species, y = Locus, fill = Alleles)) +
  theme_classic() +
  geom_tile(colour = "Black", size = .75) +
  scale_fill_viridis_c(name = "Alleles", na.value = "Grey75") +
  scale_x_discrete(expand = c(0,0)) +
  theme(
    plot.margin = margin(t = 5, r = 25, b = 5, l = 15, unit = "mm"),
    legend.position = c(1,1),
    legend.justification = c(0, 1),
    legend.direction = "vertical",
    legend.margin = margin(0,0,0,5, "mm"),
    axis.text.y = element_text(hjust = 0, colour = col_key, size = 10),
    axis.line.x = element_blank(),
    axis.text.x = element_text(size = 10)) +
  xlab("Species") +
  ylab("")
plot
```




```
ggsave(plot,
       filename = 'HeatmapLoci.tiff',
       width = 6,
       height = 9,
       units = "in",
```

```
       dpi = 300)
```

### Compare allelic richness

The heatmap above shows several patterns which are tested statistically next.

```
## get raw data again
genotypes_raw <- readxl::read_xlsx("data/genotypes_raw.xlsx", skip = 1)[, c(3, 8:ncol(seals))]

# Extract genotypes
marker_geno <- apply(genotypes_raw[,-1], 2, as.character)

# Collapse information for each locus in one column
marker_geno <- lapply(seq(1, ncol(marker_geno), 2), function(x) {
  marker_geno[,x:(x + 1)] %>%
    apply(., 1, paste0, collapse = "/")
}) %>%
  do.call("cbind",.) %>%
  ## rename loci
  set_colnames(x = ., value = paste0("Locus", 1:61))

## set missing data to NA
marker_geno[which(marker_geno == "NA/NA")] <- NA

## Create GENIND for Antarctic fur seal alone
genind_afs <- adegenet::df2genind(marker_geno[1:78,], ploidy = 2, sep = "/")

## extract allele numbers for both marker types
immune_afs <- genind@loc.n.all[1:13]
mean(immune_afs)
```

```
## [1] 4.461538
```

```
sd(immune_afs)
```

```
## [1] 2.43637
```

```
neutral_afs <- genind@loc.n.all[14:61]
mean(neutral_afs)
```

```
## [1] 7.375
```

```
sd(neutral_afs)
```

```
## [1] 2.573391
```

```
## compare marker types
wilcox.test(immune_afs, neutral_afs, paired = F)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  immune_afs and neutral_afs
## W = 129, p-value = 0.001215
## alternative hypothesis: true location shift is not equal to 0
```

```
## compare neutral markers by origin
neutral_afs <- genind@loc.n.all[14:22]
mean(neutral_afs)
```

```
## [1] 4.777778
```

```
sd(neutral_afs)
```

```
## [1] 2.048034
```

```
neutral_others <- genind@loc.n.all[23:61]
mean(neutral_others)
```

```
## [1] 7.974359
```

```
sd(neutral_others)
```

```
## [1] 2.311153
```

```
## compare by marker
wilcox.test(neutral_afs, neutral_others, paired = F)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  neutral_afs and neutral_others
## W = 52.5, p-value = 0.001104
## alternative hypothesis: true location shift is not equal to 0
```

```
## cross-amplification
immune <- dplyr::filter(heatmap_df, Species != "Antarctic fur seal", Type == "Immune")[["Alleles"]]
immune <- ifelse(is.na(immune), 0, 1) # check if amplified
mean(immune) ## cross-amplification rate
```

```
## [1] 0.6923077
```

```
neutral <- dplyr::filter(heatmap_df, Species != "Antarctic fur seal")[27:44, "Alleles"]
neutral <- ifelse(is.na(neutral), 0, 1) # check if amplified
mean(neutral) ## cross-amplification rate
```

```
## [1] 0.2222222
```

```
wilcox.test(neutral, immune, paired = F)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  neutral and immune
## W = 124, p-value = 0.00255
## alternative hypothesis: true location shift is not equal to 0
```

## References