Investigating associations between MHC diversity and scent and colony membership

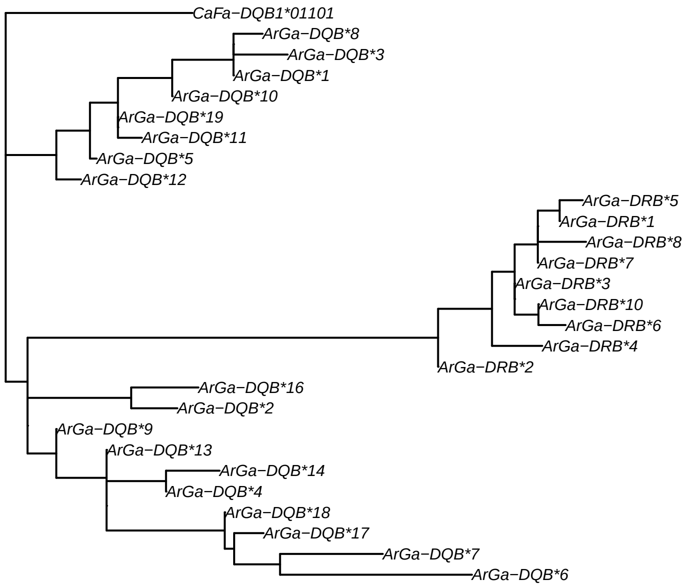
Meinolf Ottensmann

```
library(GCalignR)
library(ggplot2)
library(magrittr)
library(vegan)
library(phyloseq)
```

Calculate pairwise UniFrac distances

Pairiwse estimates of MHC similariy are calculated by UniFrac distances derived from a maximum likelihood gene tree that was constructed in MEGA7 and exported as *.nwk* file.

```
## load ML tree
tree <- ape::read.tree(file = "Phylogeny/Trees/nwk/arga_beta_exon.nwk")
plot(tree, cex = 0.5)
```



Next, genotypes are loaded and subsetted to individuals that have been successfully genotyped for both loci. Furthermore, this analysis is restricted to individuals that possess only those alleles that were classified as genuine with high confidence (i.e. present in more than one individual).

```
## load genotypes:
load("miseq_reads/DQB-Pool/RData/genotypes_dqb.RData")
load("miseq_reads/DRB-Pool/RData/genotypes_drb.RData")

## find samples in both datasets
samples <- names(genotypes_dqb)[which(names(genotypes_dqb) %in% names(genotypes_drb))]

## order samples
genotypes_drb <- genotypes_drb[samples]
genotypes_dqb <- genotypes_dqb[samples]

## change Zotu to correct allele prefix (i.e. ArGa-DQB and ArGa-DRB)
for (i in 1:length(genotypes_dqb)) {
  genotypes_dqb[[i]] <- stringr::str_replace(genotypes_dqb[[i]], "Zotu", "ArGa-DQB*")
  genotypes_drb[[i]] <- stringr::str_replace(genotypes_drb[[i]], "Zotu", "ArGa-DRB*")
}

genotypes <- genotypes_drb
for (i in 1:length(genotypes)) {
  genotypes[[i]] <- c(genotypes[[i]], genotypes_dqb[[i]])
}

## remove samples with uncertain alleles

fx <- function(x) any(!(x %in% tree$tip.label))
remove <- which(!(unlist(lapply(genotypes,fx))) == FALSE)

samples <- samples[-as.numeric(remove)]
genotypes <- genotypes[-as.numeric(remove)]
```

Now, a matrix of OTUs (= alleles) is constructed in which the presence/absence of alleles within indiviudal samples is coded by 1 and 0 respectively. Samples represent columns and rows are alleles

```
## get otus = alleles
otus <-  unique(unlist(genotypes))

## create a matrix of otus, code presence/absence per individual
otumat <- matrix(0, nrow = length(otus), ncol = length(genotypes))
rownames(otumat) <- otus
colnames(otumat) <- names(genotypes)
for (i in 1:length(genotypes)) {
  temp <- genotypes[[i]]
  otumat[which(rownames(otumat) %in% temp),i] <- 1
}
## overview
otumat[1:5, 1:5]
>             M12 P12 P27 P30 P28
> ArGa-DRB*1    1   1   1   1   0
> ArGa-DRB*2    1   1   0   0   1
> ArGa-DQB*1    1   0   1   1   1
> ArGa-DQB*7    1   0   0   0   0
> ArGa-DQB*10   1   1   0   0   0
```

In order to calculate UniFrac distances, a *phseq* objects needs to be constructed first, in which the gene tree and the OTU table are combined.

```
## create physeq object
physeq <- phyloseq::otu_table(otumat, taxa_are_rows = TRUE)
physeq <- phyloseq::merge_phyloseq(physeq, tree)

## calculate pairwise unifrac distances
ufrac <- phyloseq::UniFrac(physeq)
```

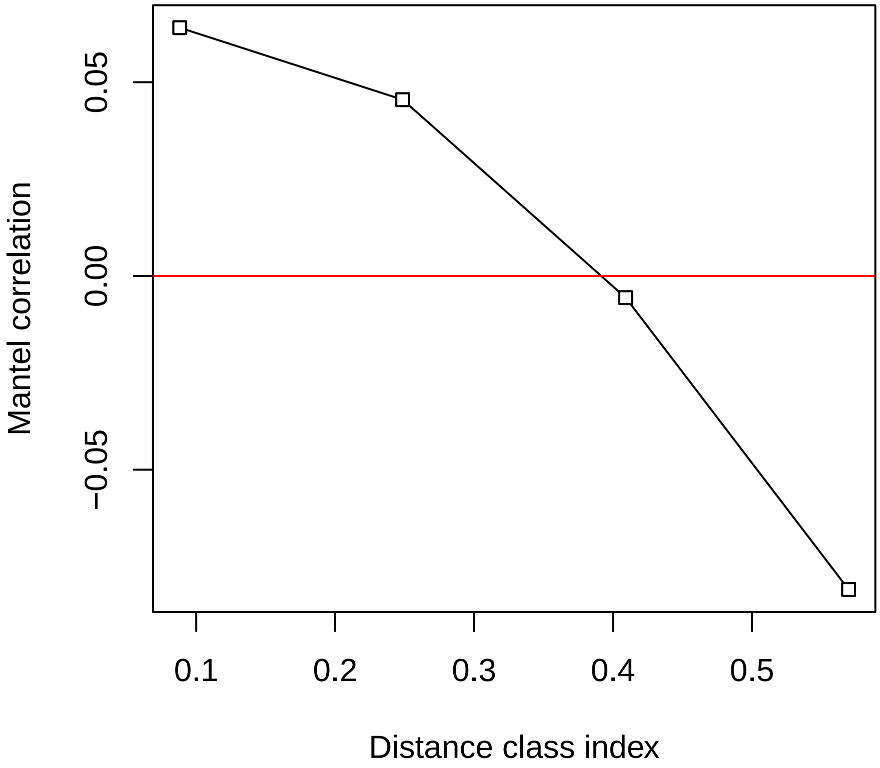
Process raw chemical data from (**???**). As the processing requires considerable execution time, the resulting data are available within this repository.

```
## not run
# load("RData/peak_data.RData")
# aligned_peak_data <-
#   GCalignR::align_chromatograms(data = peak_data,
#                                 rt_col_name = "time",
#                                 rt_cutoff_low = 8,
#                                 blanks = c("C2","C3"),
#                                 delete_single_peak = T)
load(file =  "RData/aligned_peak_data.RData")

## normalise
scent <-
  GCalignR::norm_peaks(aligned_peak_data,
                    conc_col_name = "area",
                    rt_col_name = "time",
                    out = "data.frame")
scent <- log(scent + 1)

## subset data with respect to MHC data
scent <- scent[match(colnames(otumat), rownames(scent)),]

## calculate distance
scent_bc <- (as.matrix(vegan::vegdist(as.matrix(scent)), method = "bray")) %>%
  as.dist()
```

Perform statistics

```
## Mantel test
vegan::mantel(ydis = ufrac, xdis = scent_bc, method = "spearman", permutation = 9999)
>
> Mantel statistic based on Spearman's rank correlation rho
>
> Call:
> vegan::mantel(xdis = scent_bc, ydis = ufrac, method = "spearman",      permutations = 9999)
>
> Mantel statistic r: 0.1148
>       Significance: 0.0447
>
> Upper quantiles of permutations (null model):
```

```
>    90%    95%  97.5%    99% 
> 0.0817 0.1103 0.1383 0.1717 
> Permutation: free
> Number of permutations: 9999

x <- vegan::mantel.correlog(D.geo = ufrac, D.eco =  scent_bc, n.class = 4, r.type = "spearman")
plot(x)
```

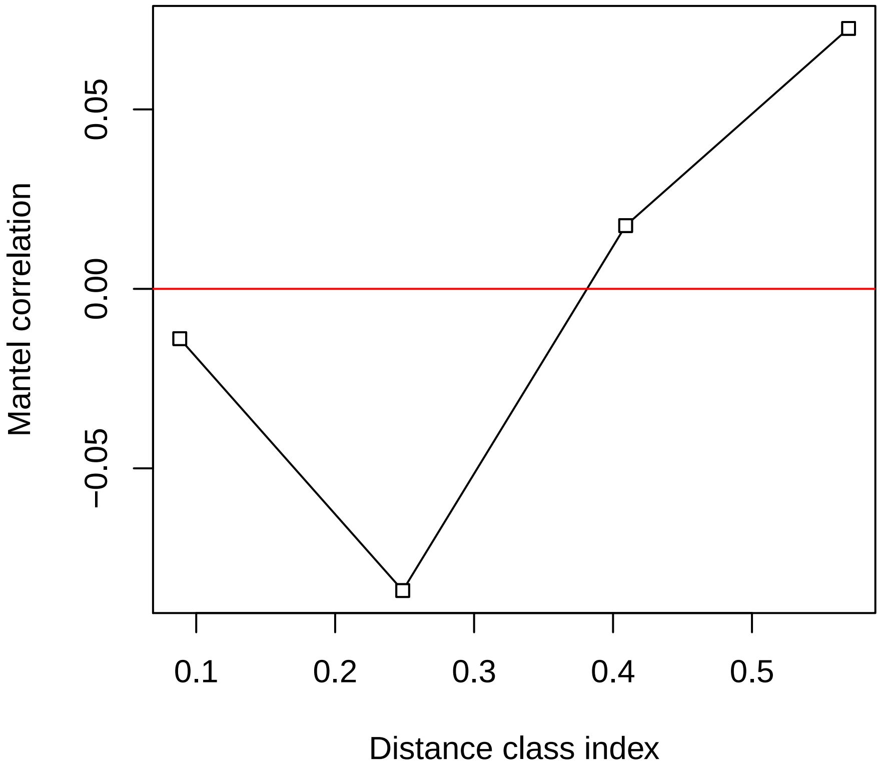


Load estimates of genome-wide Queller & Goodnight relatedness (**???**) and subset the data with respect to MHC data

```
microsats <- read.table("data/relatedness_microsats.txt")

# create matrix to store data of interest
microsats_sub <- matrix(NA, length(samples), length(samples))
rownames(microsats_sub) <- colnames(otumat)
colnames(microsats_sub) <- colnames(otumat)
## fill with values
for (r in samples) {
  for (c in samples) {
    relatedness <- ifelse(
      is.na(microsats[which(rownames(microsats) == r),
                             which(rownames(microsats) == c)]),
                   microsats[which(rownames(microsats) == c),
                             which(rownames(microsats) == r)],
                   microsats[which(rownames(microsats) == r),
                             which(rownames(microsats) == c)])
    microsats_sub[r,c] <- relatedness
  }
}
```

Explore relationship betwee UniFrac distances and relatedness. For ease of interpretation relatedness estimates are converted to distances by substracting from 1.

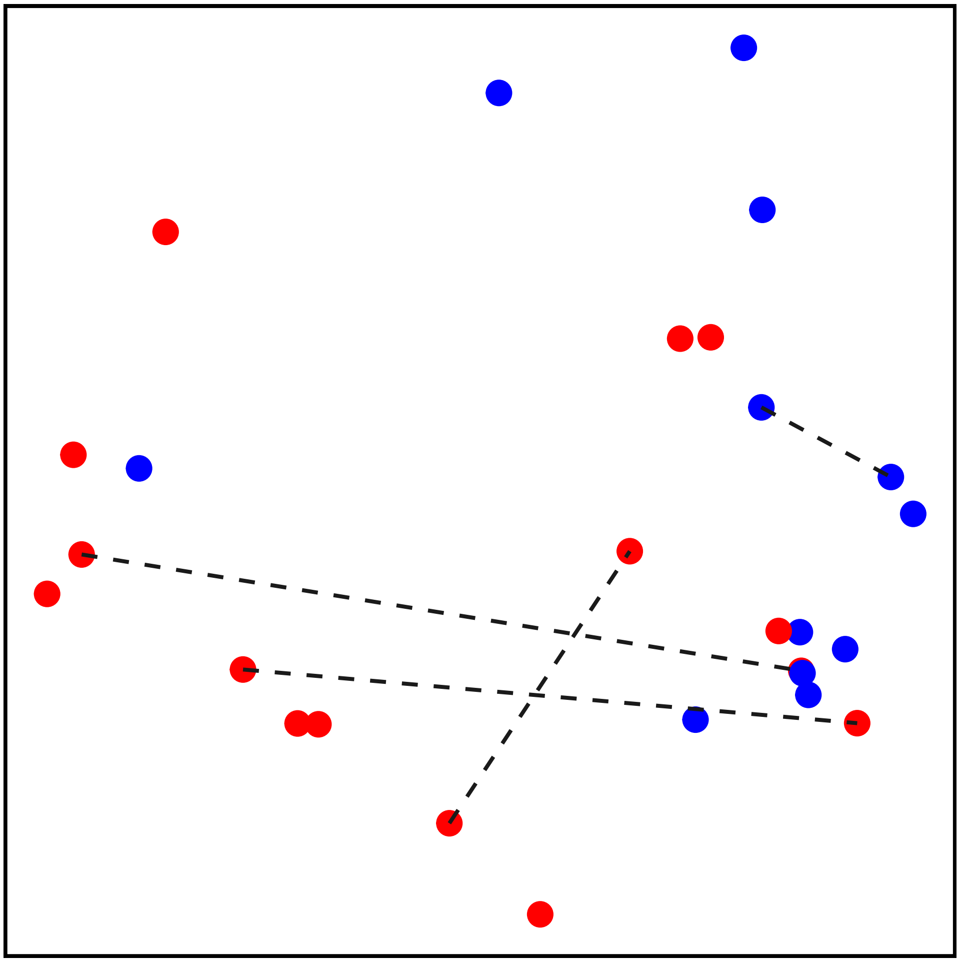
```
## Mantel test
vegan::mantel(ydis = ufrac, xdis = 1 - as.dist(microsats_sub), method = "spearman", permutation = 9999)
>
> Mantel statistic based on Spearman's rank correlation rho
>
> Call:
> vegan::mantel(xdis = 1 - as.dist(microsats_sub), ydis = ufrac,      method = "spearman", permutations
>
> Mantel statistic r: -0.07048
>       Significance: 0.8631
>
> Upper quantiles of permutations (null model):
>   90%   95% 97.5%   99%
> 0.082 0.103 0.123 0.144
> Permutation: free
> Number of permutations: 9999

x <- vegan::mantel.correlog(D.geo = ufrac, D.eco =  1 - as.dist(microsats_sub), n.class = 4, r.type = "
plot(x)
```



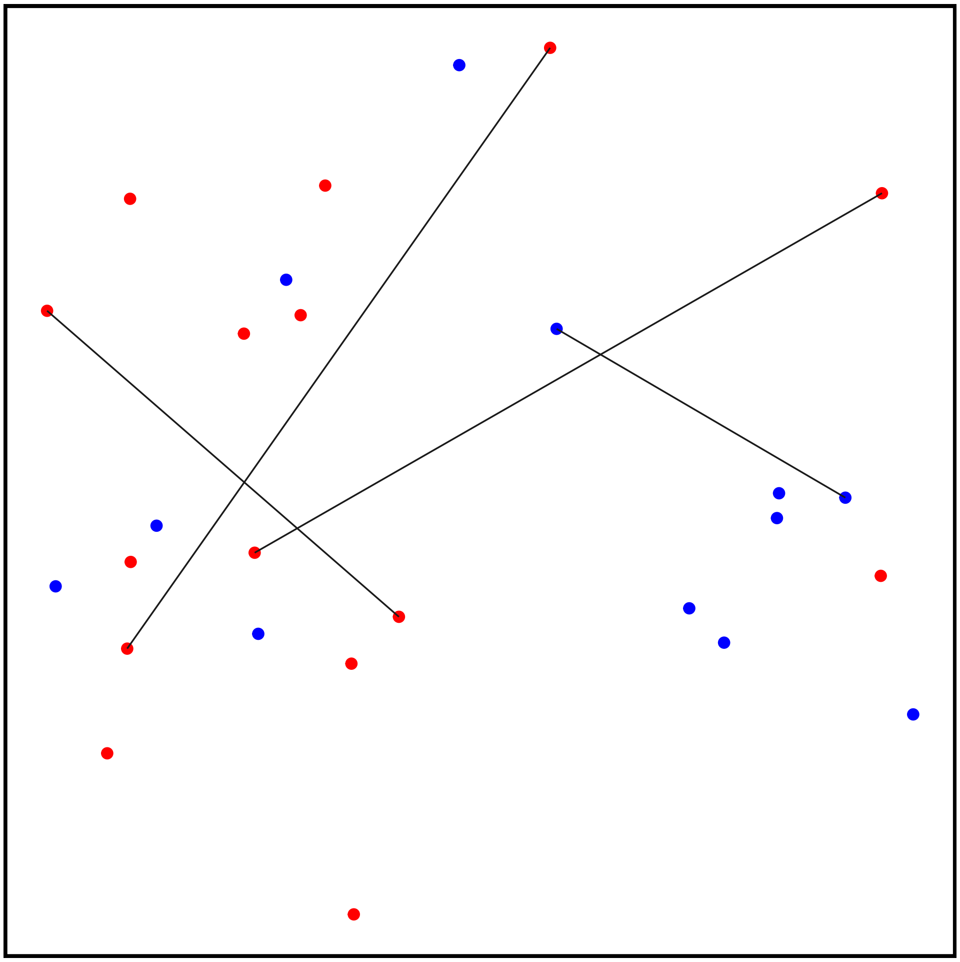
Nonmetric multidimensional scaling (NMDS)

```
## load and wrangle metadatatata
load("RData/metadata_scent.RData")
metadata_scent <- metadata_scent[which(rownames(metadata_scent) %in% rownames(scent)),]
metadata_scent$age <- as.factor(metadata_scent$age)
metadata_scent$age <- factor(metadata_scent$age, labels = c("adult", "pub"))
metadata_scent$family <- as.factor(metadata_scent$family)
metadata_scent$ind <- rownames(metadata_scent)
head(metadata_scent)
>     colony family   age ind
> M12    FWB     12 adult M12
> M14    FWB     14 adult M14
> M15    SSB     15 adult M15
> M16    FWB     16 adult M16
> M19    SSB     19 adult M19
> M23    FWB     23 adult M23

## add to physeq object
sample_data(physeq) <- metadata_scent
sample_data(physeq) <- peak_factors
ordu = ordinate(physeq, "NMDS", "unifrac")
> Run 0 stress 0.1438507
> Run 1 stress 0.140313
> ... New best solution
> ... Procrustes: rmse 0.04613077  max resid 0.1405379
> Run 2 stress 0.152869
> Run 3 stress 0.1393084
> ... New best solution
> ... Procrustes: rmse 0.02076157  max resid 0.06915865
> Run 4 stress 0.140313
> Run 5 stress 0.1393084
> ... Procrustes: rmse 2.062554e-06  max resid 6.097536e-06
> ... Similar to previous best
> Run 6 stress 0.1396491
> ... Procrustes: rmse 0.06067956  max resid 0.2860723
> Run 7 stress 0.152869
> Run 8 stress 0.152869
> Run 9 stress 0.1630032
> Run 10 stress 0.1744331
> Run 11 stress 0.140313
> Run 12 stress 0.1842888
> Run 13 stress 0.1396491
> ... Procrustes: rmse 0.06066725  max resid 0.2860117
> Run 14 stress 0.1396491
> ... Procrustes: rmse 0.06064694  max resid 0.2859126
> Run 15 stress 0.1396491
> ... Procrustes: rmse 0.06068085  max resid 0.2860801
> Run 16 stress 0.1393084
> ... New best solution
> ... Procrustes: rmse 2.925486e-06  max resid 8.511809e-06
> ... Similar to previous best
> Run 17 stress 0.1393084
> ... Procrustes: rmse 2.421932e-06  max resid 7.995825e-06
> ... Similar to previous best
> Run 18 stress 0.162529
> Run 19 stress 0.152869
> Run 20 stress 0.1998472
> *** Solution reached
p1 <- plot_ordination(physeq,
                      ordu,
                      color = "colony")

p1 <- p1 +
  geom_point(size = 4) +
  theme_void() +
  scale_color_manual(values = c("blue","red")) +
  theme(panel.background = element_rect(colour = "black", size = 1.25,
                                        fill = NA),
        aspect.ratio = 1,
        legend.position = "b") +
  geom_line(aes(group = family), size = 0.7, color = "grey10",
            linetype = "dashed")
```

```
p1
```



```
scent_nmds <- vegan::metaMDS(comm = scent, distance = "bray")
> Run 0 stress 0.1882077
> Run 1 stress 0.1911386
> Run 2 stress 0.2558383
> Run 3 stress 0.1825182
> ... New best solution
> ... Procrustes: rmse 0.0442241  max resid 0.1622274
> Run 4 stress 0.1911386
> Run 5 stress 0.2675943
> Run 6 stress 0.1825185
> ... Procrustes: rmse 0.0002897934  max resid 0.001140754
> ... Similar to previous best
> Run 7 stress 0.1825182
> ... Procrustes: rmse 6.448387e-05  max resid 0.000220168
> ... Similar to previous best
> Run 8 stress 0.1911386
> Run 9 stress 0.1884093
> Run 10 stress 0.2381474
> Run 11 stress 0.2178516
> Run 12 stress 0.2136452
> Run 13 stress 0.2176605
> Run 14 stress 0.2723225
> Run 15 stress 0.1911386
> Run 16 stress 0.1824748
> ... New best solution
> ... Procrustes: rmse 0.02360522  max resid 0.09916996
> Run 17 stress 0.1824749
> ... Procrustes: rmse 8.354673e-05  max resid 0.0002778932
> ... Similar to previous best
> Run 18 stress 0.2394441
> Run 19 stress 0.2143889
> Run 20 stress 0.1911386
> *** Solution reached
## get x and y coordinates
scent_nmds <- as.data.frame(scent_nmds[["points"]])
## add the colony as a factor to each sample
scent_nmds <- cbind(scent_nmds, colony = metadata_scent[["colony"]], ind = rownames(metadata_scent), fa

ggplot(data = scent_nmds,aes(MDS1,MDS2,color = colony)) +
  geom_point() +
  theme_void() +
  scale_color_manual(values = c("blue","red")) +
  theme(panel.background = element_rect(colour = "black", size = 1.25,
                                        fill = NA), aspect.ratio = 1,
        legend.position = "n") +
  geom_line(data = scent_nmds, aes(group = family), size = 0.3, color = "grey10")
```



Test for colony effect

```
## colony effect
vegan::adonis(scent ~ metadata_scent$colony,permutations = 9999)
>
> Call:
> vegan::adonis(formula = scent ~ metadata_scent$colony, permutations = 9999)
>
> Permutation: free
> Number of permutations: 9999
>
> Terms added sequentially (first to last)
>
>                       Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
> metadata_scent$colony  1    0.3004 0.30036  1.3408 0.0509 0.1847
> Residuals             25    5.6004 0.22402         0.9491
> Total                 26    5.9008                 1.0000
vegan::adonis(ufrac ~ metadata_scent$colony ,permutations = 9999)
>
> Call:
> vegan::adonis(formula = ufrac ~ metadata_scent$colony, permutations = 9999)
```

```
>
> Permutation: free
> Number of permutations: 9999
>
> Terms added sequentially (first to last)
>
>                       Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
> metadata_scent$colony  1   0.17281 0.172813   2.122 0.07824  0.105
> Residuals             25   2.03598 0.081439         0.92176
> Total                 26   2.20879                  1.00000
```

```
sessionInfo()
```

References