Fur seal chemical fingerprints encode colony membership, mother-offspring similarity, relatedness and genetic quality

Stoffel, M.A., Caspers, B.A., Forcada, J., Giannakara, A., Baier, M.C., Eberhart-Phillips, L.J. , Müller, C. & Hoffman, J.I.

This document provides the code for all major analysis from our paper. Duplicate analyses that have not been part of an argument (e.g. most analysis for pups) were strapped out for readability. The Rmarkdown file as well as the data are stored on GitHub. For any questions just contact me: martin.adam.stoffel@gmail.com

We wrote a package for doing some of the inbreeding related analysis, such as calculating g2 or sMLH which is hosted on this GitHub repository. The inbreedR package provides functions for measuring inbreeding from molecular data (SNPs and microsatellites) and will soon be published. To download packages from GitHub repositories, one needs to install the `devtools` package.

For running the complete code you need two additional functions (`get_pairdiff` and `multiplot`) which are outsourced as well as the `files` folder as a subfolder.

```
# install.packages("devtools")
library(devtools)
# install_github("mastoffel/inbreedR")
library(inbreedR)
```

See `?inbreedR` for further information on the functions.

Loading data, standardisation and transformation

Loading the

- raw chemical data (scent_raw, called scent data from now), which is the output of Gas-chromatography peak detection was done in Xcalibur 2.0.5 (talk about preprocessing here)
- and a data frame containing identities for colony membership (colony), mother-offspring pairs (family) and mothers and pups, respectively (age)

```
scent_raw <- as.data.frame(t(read.csv(".\\files\\scent_raw.csv", row.names = 1)))
factors <- read.csv(".\\files\\factors.csv",row.names=1)
head(factors)
#>     colony family age
#> M10      2     10   1
#> M12      2     12   1
#> M14      2     14   1
#> M15      1     15   1
#> M16      2     16   1
#> M17      2     17   1
```

Standardising observations by total, such that within every observation compounds add up to 100 % (Thus averaging out absolute concentration differences between samples)

```r
scent_stand <- as.data.frame(t(apply(scent_raw, 1, function(x) (x/sum(x)) * 100)))
```

Log(x+1) transformation of the standardised scent data.

```r
scent <- log(scent_stand + 1)
```

The scent matrix contains 82 observations and 213 compounds (retention times of chemicals are column names, values are relative concentrations) in total.

```r
dim(scent)
#> [1]  82 213
head(scent[1:6])
#>     8.061111111     8.23 8.307142857 8.394  8.47375 8.516153846
#> M10    0.000000 0.000000   0.0000000     0 0.000000   0.6562090
#> M12    0.000000 0.000000   0.4864961     0 0.000000   0.0000000
#> M14    3.222626 1.665421   0.0000000     0 0.000000   0.0000000
#> M15    0.000000 0.000000   0.0000000     0 0.000000   0.0000000
#> M16    0.000000 0.000000   0.6849915     0 1.008018   0.5654895
#> M17    2.330450 0.000000   0.0000000     0 0.000000   0.0000000
```

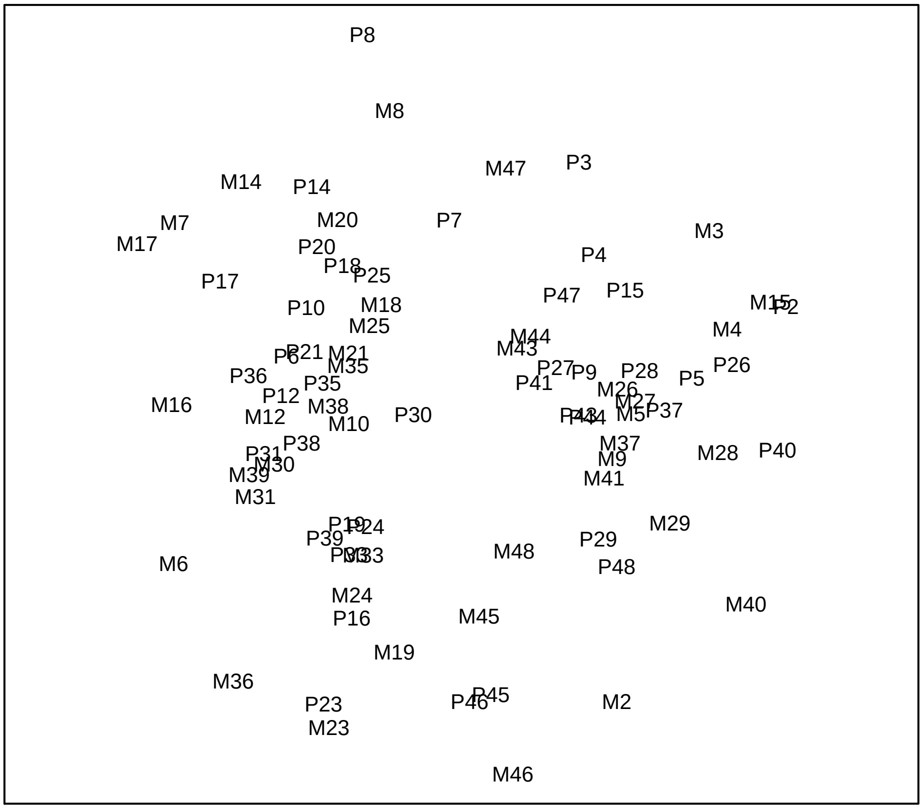
Colony differences in chemical fingerprints

```r
library(vegan)
library(MASS)
```

Non-metric multidimensional scaling (nMDS) visualizes a distance matrix (Bray-Curtis similarity). The nMDS algorithm aims to place each individual in a 2-dimensional space such that the between-individual distances are preserved as well as possible. Axis coordinates are arbitrary and not shown. The plot is better visualized with colours (see paper) and is shown here for the purpose of demonstration. Mother-offspring pairs can be identified by labels (e.g. M14, P14).

```r
scent_mds <- MASS::isoMDS(vegdist(scent))
#> initial  value 28.002906
#> iter   5 value 21.594484
#> final  value 21.345037
#> converged
```

```r
vegan::ordiplot(scent_mds, type = "t", ylab = "", xlab = "",axes=FALSE, frame.plot=TRUE)
```



Analysis of Similarities (ANOSIM) is a non-parametric test for group differences based on a Bray-curtis (or any other) similarity matrix. We use the vegan package (Oksanen et al. 2015) for ANOSIM and several other functions. Most analysis are done for the whole sample as well as for mothers and pups seperately to avoid pseudoreplication. ANOSIM is based on a permutation test, which is why results can slightly differ from the paper.

Dissimilarity between the two colonies.

```r
vegan::anosim(dat = scent, grouping = factors$colony,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent, grouping = factors$colony, permutations = 1000,      distance = "bray")
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.5691
#>       Significance: 0.000999
#>
```

```
#> Permutation: free
#> Number of permutations: 1000
```

Dissimilarity between mothers from the two colonies.

```
vegan::anosim(dat = scent[factors$age == 1, ], grouping = factors$colony,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$age == 1, ], grouping = factors$colony,      permutations = 1000,
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.5748
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Dissimilarity between pups from the two colonies.

```
vegan::anosim(dat = scent[factors$age == 2, ], grouping = factors$colony,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$age == 2, ], grouping = factors$colony,      permutations = 1000,
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.556
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Genetic differentiation of the two colonies was assesed through bayesian structure analysis, with the software "Structure" (Pritchard, Stephens, and Donnelly 2000)

Mother offspring similarity in chemical fingerprints.

Full sample

```
vegan::anosim(dat = scent, grouping = factors$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent, grouping = factors$family, permutations = 1000,      distance = "bray")
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.6723
#>       Significance: 0.000999
```

```
#>
#> Permutation: free
#> Number of permutations: 1000
```

Mother offspring similarity within colony 1 (Special study beach)

```
vegan::anosim(dat = scent[factors$colony == 1, ],
       grouping = factors[factors$colony == 1, ]$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$colony == 1, ], grouping = factors[factors$colony ==      1, ]$fam
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.5339
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Mother offspring similarity within colony 2 (Freshwater beach)

```
vegan::anosim(dat = scent[factors$colony == 2, ],
       grouping = factors[factors$colony == 2, ]$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$colony == 2, ], grouping = factors[factors$colony ==      2, ]$fam
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.4532
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Chemical similarity vs. geographic distance on special study beach

- location data in meters is available for this population as the special study beach on Bird Island provides an aerial walkway

Loading X-Y coordinates of each individual.

```
coord  <- read.csv(".\\files\\coordinates_beach1.csv", row.names=1)
head(coord)
#>      X  Y
#> M15 10  8
#> M19 10 12
```

```
#> M2  25 15
#> M26 23 13
#> M27 26 18
#> M28 26 18
```

Converting coordinates to pairwise euclidian distance matrix.

```
dist_mat <- as.matrix(dist(coord, method = "euclidian"))
```

Constructing a bray curtis similarity matrix (from chemical fingerprints) of all individuals from beach 1 (special study beach). We constantly used spearman rank correlation in mantel tests.

```
scent_bc <- as.matrix(vegan::vegdist(as.matrix(scent[factors$colony == 1, ])),
                      method = "bray")
```

Geographic distance vs. chemical similarity in mothers

```
geo_mum <- dist_mat[1:20, 1:20]
scent_mum <- scent_bc[1:20, 1:20]
vegan::mantel(geo_mum, scent_mum, method = "spearman")
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = geo_mum, ydis = scent_mum, method = "spearman")
#>
#> Mantel statistic r: 0.008091
#>       Significance: 0.478
#>
#> Upper quantiles of permutations (null model):
#>   90%   95% 97.5%   99%
#> 0.194 0.251 0.290 0.332
#> Permutation: free
#> Number of permutations: 999
```

Geographic distance vs. chemical similarity in pups

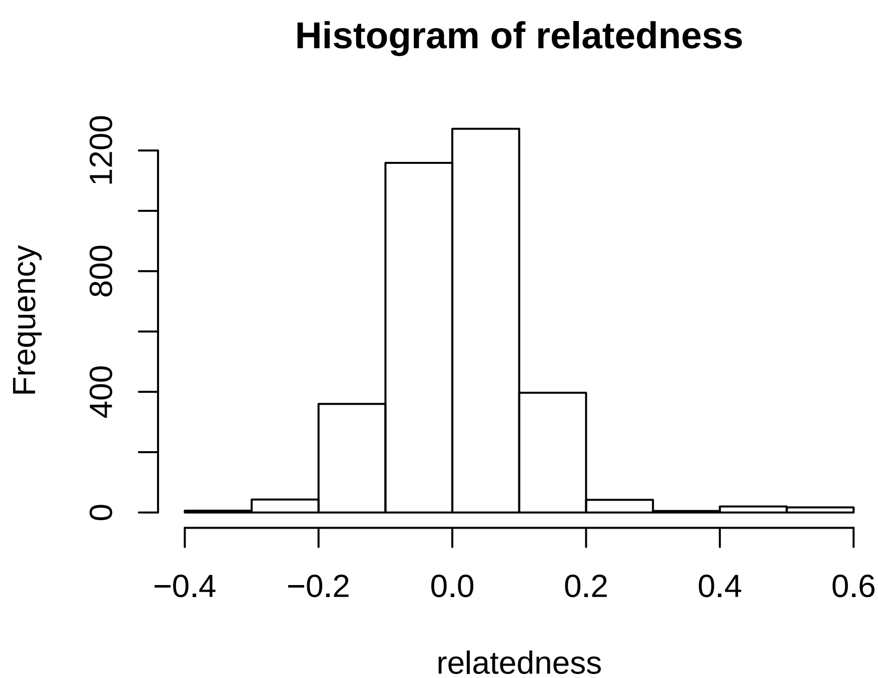
```
geo_pup <- dist_mat[21:40, 21:40]
scent_pup <- scent_bc[21:40, 21:40]
vegan::mantel(geo_pup, scent_pup, method = "spearman")
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = geo_pup, ydis = scent_pup, method = "spearman")
#>
#> Mantel statistic r: 0.06039
#>       Significance: 0.297
#>
#> Upper quantiles of permutations (null model):
#>   90%   95% 97.5%   99%
#> 0.150 0.208 0.261 0.307
```

```
#> Permutation: free
#> Number of permutations: 999
```

Correlation between genotype and overall chemical fingerprints.

Relatedness and overall chemical similarity Load pairwise relatedness (Queller and Goodnight 1989) based on 41 microsatellite markers.

```
relatedness <- as.matrix(read.csv(".\\files\\relatedness.csv",row.names=1))
head(relatedness[1:6, 1:6])
#>             M10         M12         M14         M15         M16 M17
#> M10          NA          NA          NA          NA          NA  NA
#> M12 -0.09578940          NA          NA          NA          NA  NA
#> M14 -0.10861601 -0.16464236          NA          NA          NA  NA
#> M15 -0.03246021 -0.11981456 -0.12591268          NA          NA  NA
#> M16  0.07639825  0.13027995  0.01176970 -0.02336469          NA  NA
#> M17  0.04367833 -0.09591802 -0.06258925  0.03730164 -0.08061711  NA
hist(relatedness)
```



Pairwise bray curtis similarity in chemical fingerprints of all individuals.

```
scent_bc <- 1-(as.matrix(vegan::vegdist(as.matrix(scent)), method = "bray"))
head(scent_bc[1:6, 1:6])
#>           M10       M12       M14        M15        M16       M17
#> M10 1.0000000 0.4913098 0.3029950 0.22560196 0.35280730 0.4108673
```

```
#> M12 0.4913098 1.0000000 0.3552582 0.22453963 0.48401295 0.4058738
#> M14 0.3029950 0.3552582 1.0000000 0.12185666 0.38121502 0.4885025
#> M15 0.2256020 0.2245396 0.1218567 1.00000000 0.09922127 0.2028706
#> M16 0.3528073 0.4840130 0.3812150 0.09922127 1.00000000 0.3725530
#> M17 0.4108673 0.4058738 0.4885025 0.20287064 0.37255296 1.0000000
```

Mantel test between genetic relatedness and bray curtis similarity in chemical fingerprints of all individuals.

```
vegan::mantel(relatedness, scent_bc, method = "spearman", permutation = 1000)
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = relatedness, ydis = scent_bc, method = "spearman",      permutations = 1000)
#>
#> Mantel statistic r: 0.07231
#>       Significance: 0.003996
#>
#> Upper quantiles of permutations (null model):
#>    90%    95%  97.5%    99%
#> 0.0369 0.0464 0.0547 0.0662
#> Permutation: free
#> Number of permutations: 1000
```

We find a significant relationship between the overall chemical fingerprints and genetic relatedness. However, we are likely to have a problem of pseudoreplication here. For that reason, we are analysing mothers and pups seperately.

Fur seal mothers: mantel test between genetic relatedness and bray curtis similarity of olfactory fingerprints.

```
vegan::mantel(relatedness[factors$age == 1, factors$age == 1],
              scent_bc[factors$age == 1, factors$age == 1],
              method = "spearman", permutation = 1000)
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = relatedness[factors$age == 1, factors$age ==      1], ydis = scent_bc[factors$a
#>
#> Mantel statistic r: 0.05938
#>       Significance: 0.10789
#>
#> Upper quantiles of permutations (null model):
#>    90%    95%  97.5%    99%
#> 0.0604 0.0757 0.0915 0.1035
#> Permutation: free
#> Number of permutations: 1000
```

Fur seal pups: mantel test between genetic relatedness and bray curtis similarity of olfactory fingerprints.

```
vegan::mantel(relatedness[factors$age == 2, factors$age == 2],
              scent_bc[factors$age == 2, factors$age == 2],
              method = "spearman", permutation = 1000)
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = relatedness[factors$age == 2, factors$age ==      2], ydis = scent_bc[factors$a
#>
#> Mantel statistic r: 0.02985
#>       Significance: 0.25974
#>
#> Upper quantiles of permutations (null model):
#>    90%    95%  97.5%    99%
#> 0.0575 0.0750 0.0900 0.1128
#> Permutation: free
#> Number of permutations: 1000
```

Correlation between heterozygosity (sMLH) and diversity (number of compounds) of chemical fingerprints

- The function `sMLH` is part of the `inbreedR` package, currently available on GitHub. Install with: (decomment the whole section for actually installing from github)

```
# install.packages("devtools")
library(devtools)
# install_github("mastoffel/inbreedR")
library(inbreedR)
# ?inbreedR
```

Loading raw genotypes and calculating standardised multilocus heterozygosity (sMLH) based on 41 markers. * `inbreedR`package requires a special format, see `?convert_raw` for more information‘*

```
genotypes <- read.table(".\\files\\genotypes.txt", row.names=1)
genotypes_formatted <- inbreedR::convert_raw(genotypes, miss_val = NA)
heterozygosity <- inbreedR::sMLH(genotypes_formatted)
```

Number of compounds per individual.

```
num_comp <- as.vector(apply(scent, 1, function(x) length(x[x>0])))
```

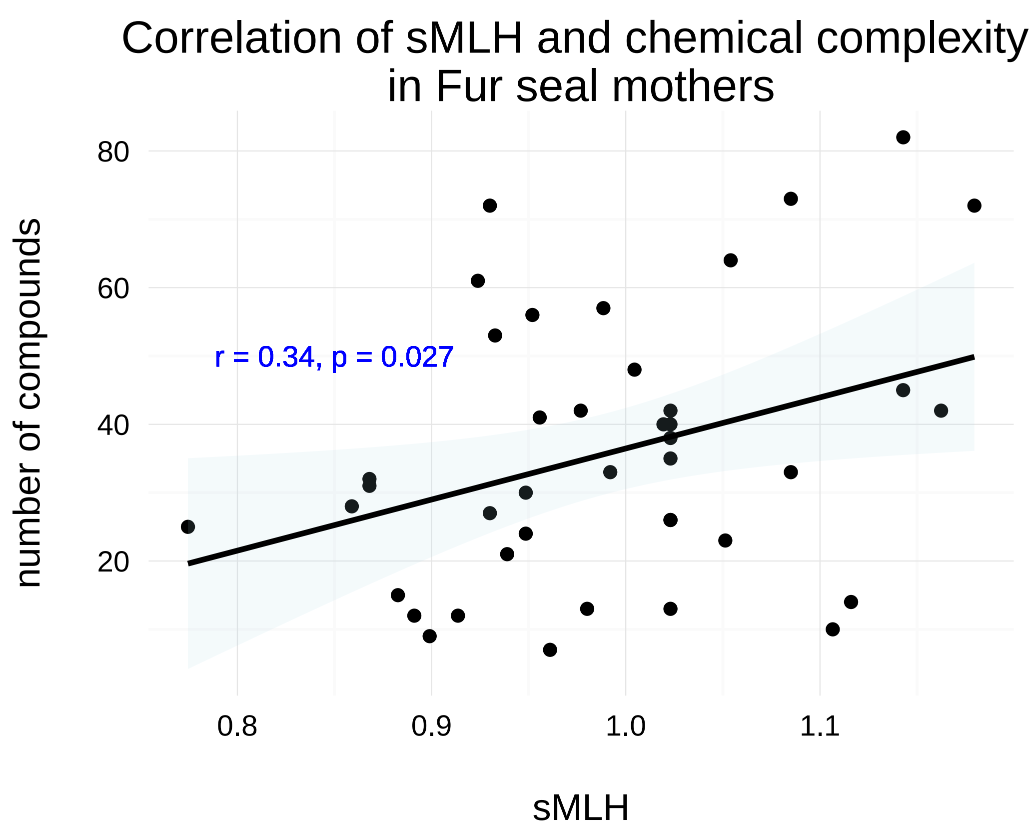
Linear model of heterozygosity on number of compounds in mothers

A clear association between sMLH and chemical complexity in mothers but not pups.

```
het_mum <- heterozygosity[factors$age == 1]
num_comp_mum <- num_comp[factors$age==1]
summary(lm(het_mum ~ num_comp_mum))
#>
```

```
#> Call:
#> lm(formula = het_mum ~ num_comp_mum)
#>
#> Residuals:
#>       Min        1Q    Median        3Q       Max
#> -0.199148 -0.049220  0.005588  0.047729  0.161623
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.9339147  0.0282437  33.066   <2e-16 ***
#> num_comp_mum 0.0015914  0.0006936   2.294   0.0272 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.08633 on 39 degrees of freedom
#> Multiple R-squared:  0.1189, Adjusted R-squared:  0.09633
#> F-statistic: 5.264 on 1 and 39 DF,  p-value: 0.02724
```

Plotting is done with ggplot2, an implementation of the grammar of graphics (Wickham 2009)



Linear model of heterozygosity on number of compounds in pups

```r
het_pup <- heterozygosity[factors$age == 2]
num_comp_pup <- num_comp[factors$age==2]
summary(lm(het_pup ~ num_comp_pup))
#>
#> Call:
#> lm(formula = het_pup ~ num_comp_pup)
#>
#> Residuals:
#>       Min        1Q    Median        3Q       Max
#> -0.199070 -0.067303 -0.001971  0.050500  0.152902
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.9940863  0.0235812  42.156   <2e-16 ***
#> num_comp_pup 0.0003928  0.0005542   0.709    0.483
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.08035 on 39 degrees of freedom
#> Multiple R-squared:  0.01272,    Adjusted R-squared:  -0.0126
#> F-statistic: 0.5025 on 1 and 39 DF,  p-value: 0.4826
```

Strength of correlation between sMLH and number of compounds increases with an increasing number of genetic markers in mothers.

The `resample_loci()` function samples an increasing subset of loci, calculates sMLH and correlates with a vector y (here: number of compounds in chemical fingerprints).

```r
resample_loci <- function(genotypes, y, num_iter = 1000) {
# genotypes in inbreedR format. See ?inbreedR
# y is a vector to correlate with sMLH
# num_iter is the number of resamplings per added locus
        # calculate number of loci
        num_loci <- ncol(genotypes)
        results <- data.frame(matrix(nrow = num_iter, ncol = num_loci))
        for (i in seq_along((1: num_loci))){
                for (k in seq_along(1:num_iter)) {
                loci_ind <- sample(1:num_loci, i, replace = FALSE)
                het <- inbreedR::sMLH(genotypes[, loci_ind])
                results[k, i] <- cor(het[1:41],y) # heterozygosity subsetted for mothers
                }
        }
results
}

# Converting genotypes into the right format
genotypes_formatted <- inbreedR::convert_raw(genotypes, miss_val = NA)
# Resampling 1 - 40 loci each 1000 times, compute sMLH and correlate with number of compounds
resample_mums <- resample_loci(genotypes_formatted, num_comp_mum, num_iter = 1000)
```

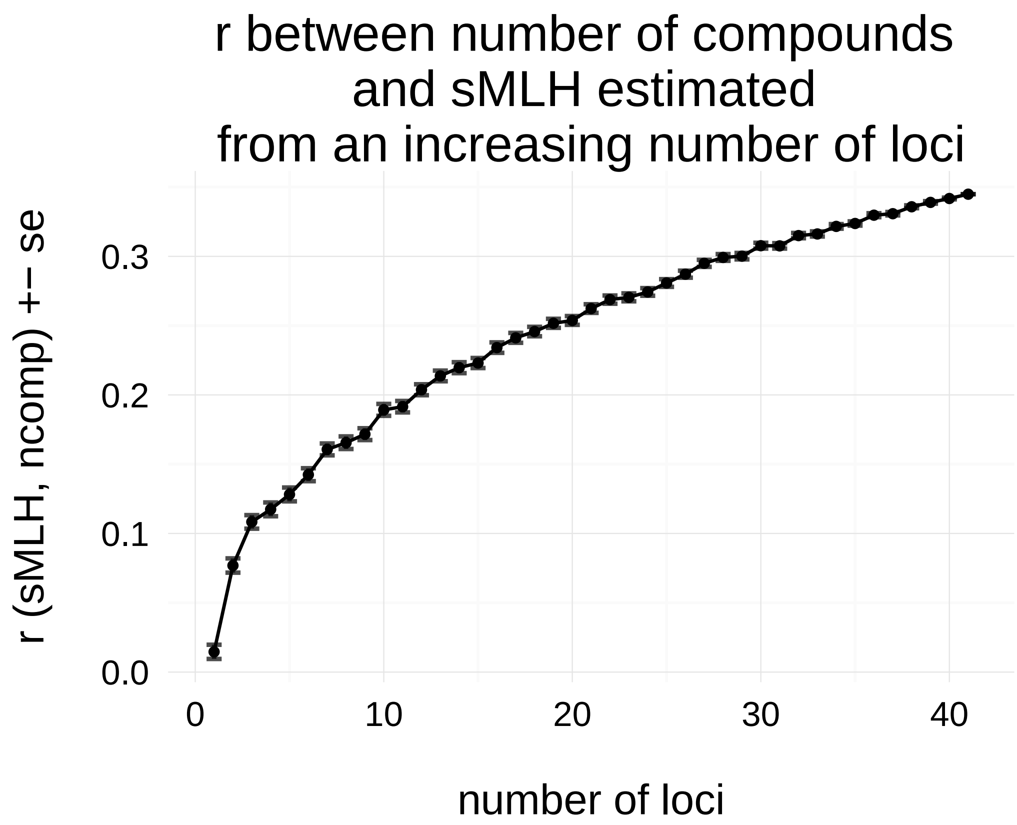
Calculating summary statistics for the resampling output: mean, sd, se of the correlations per subset of markers.

```
sum_results <- function(resampling_output) {
        mean_cor <- apply(resampling_output,2,mean, na.rm=T)
        sd_cor <- apply(resampling_output,2,sd, na.rm=T)
        se_cor <- sd_cor/(sqrt(nrow(resampling_output)))
        sum_results <- data.frame(locnum = 1:ncol(resampling_output),
                                  cormean = mean_cor, corsd = sd_cor, corse = se_cor)
}

results_mums <- sum_results(resample_mums)
```

Plotting mean correlation of heterozygosity (estimated by an increasing number of markers) with number of compounds in chemical fingerprints for Fur seal mothers.

Pups are not shown here for simplicity and to avoid code replication. For the full figure see the results section of the paper

```
# plotting
library(grid)
ggplot2::ggplot(results_mums, aes(x = locnum, y = cormean)) +
        geom_line(size = 0.6, colour = "black") +
        geom_errorbar(aes(ymin = cormean-corse, ymax = cormean+corse),
                      width=0.8, alpha=0.7, size = 0.8, colour = "black") +
        geom_point(size = 2, shape = 16) +
        theme_minimal(base_size = 16) +
        theme(axis.title.x = element_text(vjust= -2 ,size = 16),
              axis.title.y = element_text(vjust=3,size = 16),
              axis.ticks.x = element_blank(),
              axis.ticks.y = element_blank(),
              plot.margin = (unit(c(.5, .5, 2, 2), "cm"))) +
        #geom_hline(yintercept=0.305) +
        ylab("r (sMLH, ncomp) +- se") +
        xlab("number of loci") +
        labs(title = "r between number of compounds \nand sMLH estimated \nfrom an increasing number of
```



Estimation of identity disequilibrium g2 with the inbreedR package. (can diverge slightly from the RMES program)

Instead to just finding a correlation between heterozygosity and a trait such as chemical complexity, one can ask whether variation in inbreeding (so called-general effects) is a potential cause. This can be measured with a parameter called g2 (David et al. 2007), that assesses identity disequilibrium through quantification of excess double heterozygote loci. We are currently working on the inbreedR package, which provides functions for calculation g2 with both microsatellites and SNPs.

Calculate g2.

```
g2 <- inbreedR::g2_microsats(genotypes_formatted, nperm = 1000, nboot = 1000, CI = 0.95)
```

```
#>
#> Data: 82 observations at 41 markers
#> Function call = inbreedR::g2_microsats(genotypes = genotypes_formatted, nperm = 1000,     nboot = 10
#>
#> g2 = 0.00241214, se = 0.001397673
#>
#> confidence interval
#>         2.5%        97.5%
#> 2.015313e-05 5.380258e-03
```

```
#>
#> p (g2 > 0) = 0.025 (based on 1000 permutations)
```

potentially make figure for increasing markers here

Factor analysis on the chemical compounds data with the package HDMD.

HDMD (McFerrin 2013) allows for doing a Factor analysis with high dimensional data(where the number of variables exceeds the number of observations) by calculating a general inverse matrix.

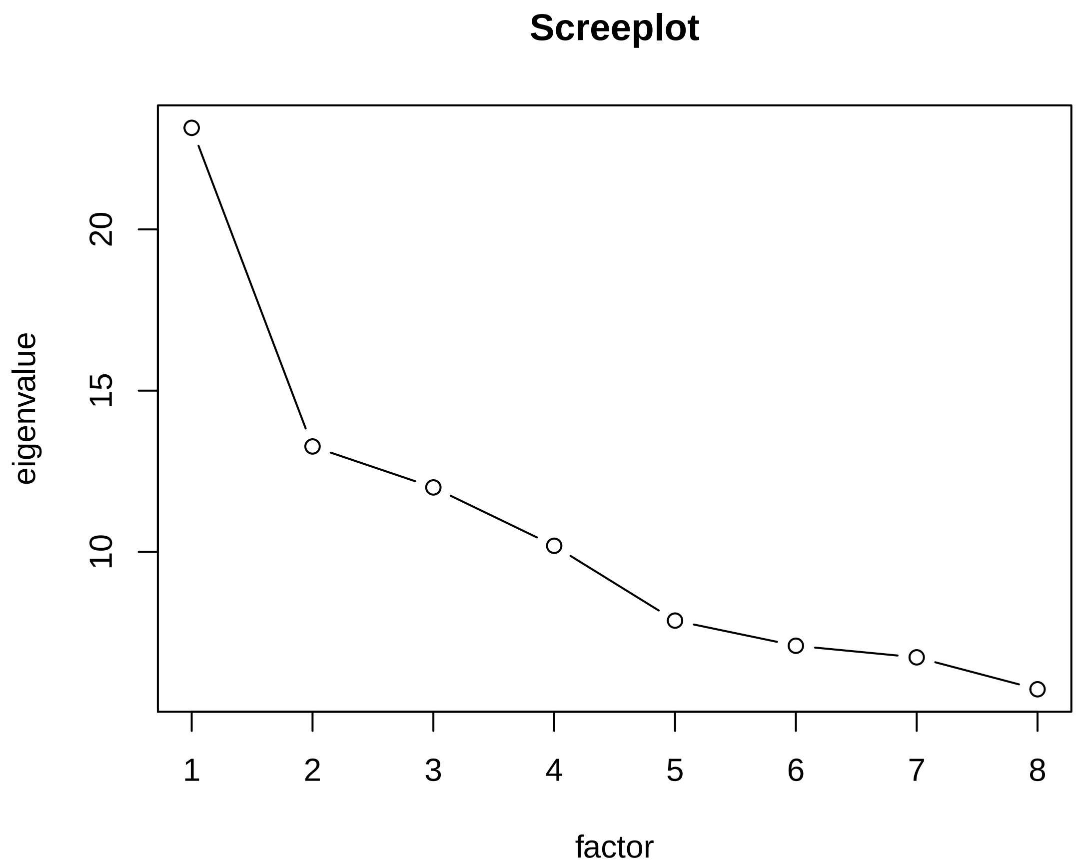
```r
library(HDMD)
library(minmodelr)
source("get_pairdiff.R")
```

Factor analysis and extraction of factor scores for the first 4 factors. Promax rotation of the factors allows them to be non-orthogonal and thus correlated. After FA, the factor scores for each individual on all 4 factors are extracted.

```r
# factor analysis with 4 factors, promax rotation ------------------------------
scent_fa <- HDMD::factor.pa.ginv(scent, nfactors = 4,
                           prerotate = T,rotate = "promax",
                           scores = T, m = 3)
#> Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
#> done
#> Could not solve for inverse correlation.  Using general inverse ginv(r)
fa_scores <- as.data.frame(scent_fa$scores)
head(fa_scores)
#>              F1         F2         F3          F4
#> M10  0.21501860 -0.8384633 0.02809373  0.41819811
#> M12  0.00144057 -0.4989538 0.19093716  0.63217100
#> M14 -0.32163481 -0.7076035 0.16407820 -0.18508919
#> M15 -0.34857816 -0.5187462 0.12530472 -0.93750096
#> M16 -0.38997409 -0.1759757 0.53663454  1.18679542
#> M17 -0.10514763 -0.6294724 0.18209318  0.02876237
```

The eigenvalue course seen in the screeplot allows for decisions on the number of factors to retain.

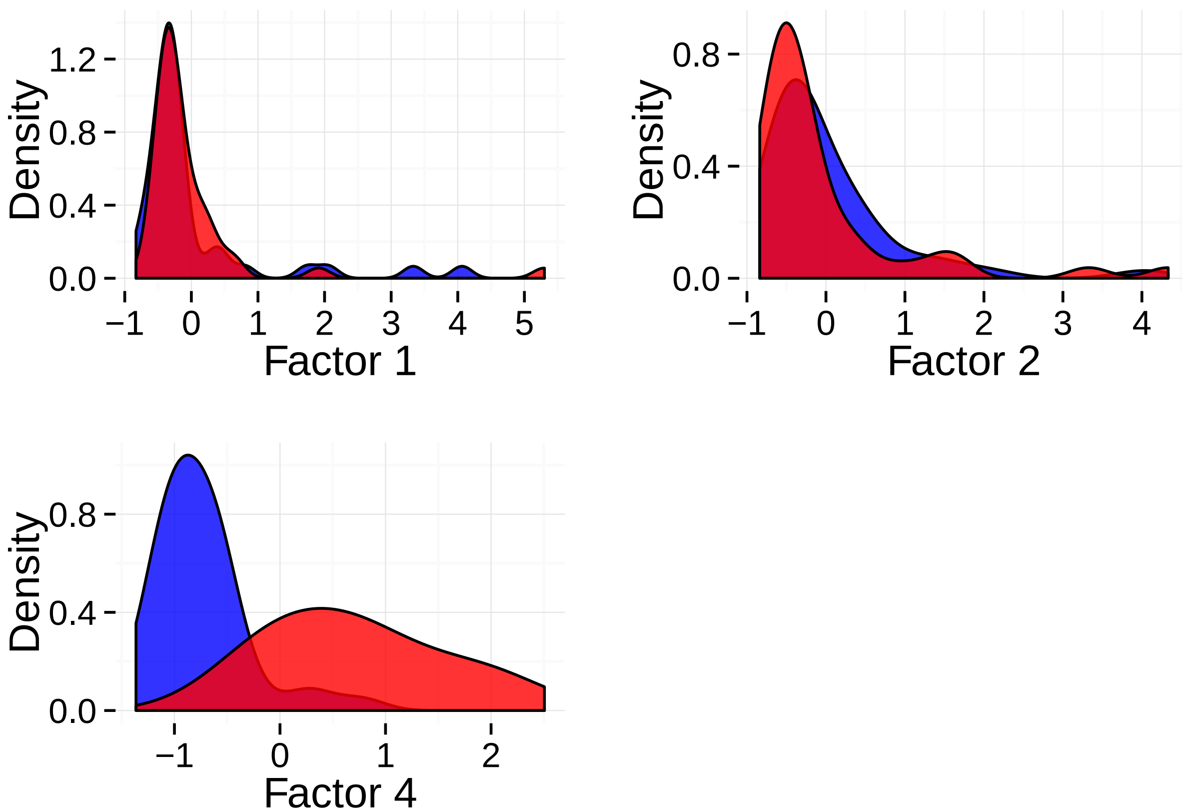
```r
# screeplot, 4 factors left to the "scree"
plot(scent_fa$values[1:8], type="b", ylab = "eigenvalue", xlab = "factor",
     main = "Screeplot")
```



Plotting the distribution of factor scores seperately for each colony. Similar distributions suggest the compounds which are represented by a given factor to be similarly distributed across colonies and could thus be of potential genetic origin, while different distributions as in factor 4 suggest this factor to represent environmentally influenced compounds.

```
# distribution of factor scores
df <- cbind(fa_scores, factors["colony"])
df$colony <- as.factor(df$colony)

for (i in c(1,2,4)) {
plot_all <- ggplot(df, aes_string(x = paste("F", i, sep = ""), fill = "colony")) +
        geom_density(alpha=0.8, size=0.5, aes(fill = colony),adjust=1.5) +
        scale_fill_manual(values = c("blue","red")) +
        guides(fill=guide_legend(title=NULL)) +
        theme_minimal(base_size = 16) +
        theme(legend.position="none") +
        scale_x_continuous(breaks = c(seq(from = -1, to = 6, by = 1))) +
        scale_y_continuous(breaks = c(seq(from = 0, to = 1.4, by = 0.4))) +
        xlab(paste("Factor", i, sep = " ")) +
        ylab("Density")

assign(paste("f", i, "_plot", sep = ""), plot_all)
}
```

```r
# using multiplot function from cookbook-r.com for plotting multiple ggplots
source("multiplot.R")
multiplot(f1_plot, f2_plot, f4_plot, cols = 2)
```



Linear model of heterozygosity on factors (factor scores) as explanatory variables in mothers.

```r
# bind heterozygosity and the factor scores in one data.frame and subset mothers
het_df <- cbind(heterozygosity, fa_scores)[factors$age == 1, ]
het_model <- lm(heterozygosity ~., data=het_df)
summary(het_model)
#>
#> Call:
#> lm(formula = heterozygosity ~ ., data = het_df)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -0.19439 -0.06271  0.01210  0.04769  0.14674
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.987971   0.013037  75.780   <2e-16 ***
#> F1           0.029393   0.012288   2.392   0.0221 *
#> F2           0.027521   0.011898   2.313   0.0265 *
#> F3           0.004033   0.012964   0.311   0.7575
#> F4          -0.009617   0.014180  -0.678   0.5020
```

```
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.08315 on 36 degrees of freedom
#> Multiple R-squared:  0.2455, Adjusted R-squared:  0.1616
#> F-statistic: 2.928 on 4 and 36 DF,  p-value: 0.03406
```

While Factor1 and Factor2 seem to represent substances that are accociated with heterozygosity, Factor 3 and Factor 4 clearly don´t. To simplify the model we used deletion testing (Crawley, Statistics). The minmodelr package contains some helper functions for this task. See ?MinMod, ?DelTestVar. *We don´t generally recommend a deletion testing procedure. In our case, results are clear and we use it for simplicity rather than for fishing significant results.*

```r
library(devtools)
# install_github("mastoffel/minmodelr")
library(minmodelr)
```

```r
het_reduced <- minmodelr::MinMod(het_df)
#>
#> Call:
#> glm(formula = depVar ~ ., family = family, data = bestmodeldf)
#>
#> Deviance Residuals:
#>       Min         1Q     Median         3Q        Max
#> -0.191202  -0.060032   0.009789   0.057485   0.155757
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.98814    0.01278  77.328   <2e-16 ***
#> F1           0.02783    0.01167   2.385   0.0222 *
#> F2           0.02809    0.01164   2.414   0.0207 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for gaussian family taken to be 0.006649744)
#>
#>     Null deviance: 0.32986  on 40  degrees of freedom
#> Residual deviance: 0.25269  on 38  degrees of freedom
#> AIC: -84.303
#>
#> Number of Fisher Scoring iterations: 2
# extract data frame
het_reduced_df <- het_reduced[[1]]
# extract reduced model
het_reduced_mod <- het_reduced[[2]]
# deletion testing for both variables in the reduced model. See ?DelTestVar
table <- minmodelr::DelTestVar(het_reduced_df)
#>               Estimate Deviance Explained        F P (F-test)
#> (Intercept) 0.98814218                 NA       NA         NA
#> F1          0.02783316           11.46936 5.689346 0.02215967
#> F2          0.02808759           11.74642 5.826780 0.02071002
#>             P (Chisquared-test)
#> (Intercept)                  NA
```

```r
#> F1                   0.01706822
#> F2                   0.01578399
# deviance explained by the reduced model
dev_expl <- (het_reduced_mod$null.deviance - het_reduced_mod$deviance) / het_reduced_mod$null.deviance
summary(het_reduced_mod)
#>
#> Call:
#> glm(formula = depVar ~ ., family = family, data = bestmodeldf)
#>
#> Deviance Residuals:
#>       Min         1Q     Median         3Q        Max
#> -0.191202  -0.060032   0.009789   0.057485   0.155757
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.98814    0.01278  77.328   <2e-16 ***
#> F1           0.02783    0.01167   2.385   0.0222 *
#> F2           0.02809    0.01164   2.414   0.0207 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for gaussian family taken to be 0.006649744)
#>
#>     Null deviance: 0.32986  on 40  degrees of freedom
#> Residual deviance: 0.25269  on 38  degrees of freedom
#> AIC: -84.303
#>
#> Number of Fisher Scoring iterations: 2
```

Creating a new variable F1F2 which is the sum of the two factor scores and using this variable as predictor in a linear model of heterozygosity.

```r
# sum of factors as variable
het_df$F1F2 <- het_df$F1 + het_df$F2
table <- minmodelr::DelTestVar(as.data.frame(cbind(het_df$heterozygosity, het_df$F1F2)))
#>               Estimate Deviance Explained        F  P (F-test)
#> (Intercept) 0.98813169                 NA       NA          NA
#> V2          0.02796073           23.39391 11.90979 0.001356642
#>             P (Chisquared-test)
#> (Intercept)                  NA
#> V2                 0.0005583969
summary(lm(heterozygosity ~ F1F2, data = het_df))
#>
#> Call:
#> lm(formula = heterozygosity ~ F1F2, data = het_df)
#>
#> Residuals:
#>       Min        1Q    Median        3Q       Max
#> -0.191204 -0.060060  0.009766  0.057466  0.155764
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 0.988132   0.012596  78.449  < 2e-16 ***
```

```
#> F1F2        0.027961   0.008102   3.451  0.00136 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.08049 on 39 degrees of freedom
#> Multiple R-squared:  0.2339, Adjusted R-squared:  0.2143
#> F-statistic: 11.91 on 1 and 39 DF,  p-value: 0.001357
```

Linear model of genetic relatedness on factor scores as explanatory variables for mothers. Pairwise genetic relatedness is represented as a matrix. To model the relationship between relatedness and factor scores we created a matrix for each factor, whereby each pairwise value represents the difference in factor scores for a pair of seals.

`get_pairdiff()` creates these matrices. We based these analysis on mothers and pups seperately.

Creation of 4 pairwise distance matrices for each factor.

```
source("get_pairdiff.R")
fa_diff_mums <- get_pairdiff(relatedness[factors$age == 1, factors$age == 1],
                             fa_scores[factors$age == 1, ], df = F)

# assign pairwise difference factor matrices to names
for (i in seq_along(1:4)) {
        assign(paste("f", i, "_diff", sep=""), fa_diff_mums[, i+1])
}
```

The ecodist package (Goslee and Urban 2007) can handle multiple distance matrices by doing a partial mantel test.

Every partial mantel test just tests for the association with the first response, while the other are permutated

```
rel_dist <- as.dist(relatedness[factors$age == 1, factors$age == 1])
ecodist::mantel(rel_dist ~ f1_diff + f2_diff + f3_diff + f4_diff, mrank = T, nperm = 1000)
#>     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
#> -0.12256667  0.97700000  0.02400000  0.04000000 -0.16665271 -0.07260009
ecodist::mantel(rel_dist ~ f2_diff + f1_diff + f3_diff + f4_diff, mrank = T)
#>      mantelr        pval1        pval2        pval3    llim.2.5%
#> -0.047920124  0.792000000  0.209000000  0.409000000 -0.087732180
#>   ulim.97.5%
#> -0.005925897
ecodist::mantel(rel_dist ~ f3_diff + f2_diff + f1_diff + f4_diff, mrank = T)
#>    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
#> 0.08900545 0.05900000 0.94200000 0.12100000 0.05002336 0.13034462
ecodist::mantel(rel_dist ~ f4_diff + f3_diff + f2_diff + f1_diff, mrank = T)
#>     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
#> 0.052327208 0.122000000 0.879000000 0.256000000 0.001209259 0.093079036
```

```


fa_diff_pups <- get_pairdiff(relatedness[factors$age == 2, factors$age == 2],
                             fa_scores[factors$age == 2, ], df = F)

for (i in seq_along(1:4)) {
```

```r
        assign(paste("f", i, "_diff", sep=""), fa_diff_pups[, i+1])
}

rel_dist <- as.dist(relatedness[factors$age == 2, factors$age == 2])
ecodist::mantel(rel_dist ~ f1_diff + f2_diff + f3_diff + f4_diff, mrank = T)
#>      mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
#>  0.02435999  0.31900000  0.68200000  0.65100000 -0.01590521  0.05874250
ecodist::mantel(rel_dist ~ f2_diff + f1_diff + f3_diff + f4_diff, mrank = T)
#>      mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
#>  0.01340994  0.41000000  0.59100000  0.80400000 -0.02748921  0.05010192
ecodist::mantel(rel_dist ~ f3_diff + f2_diff + f1_diff + f4_diff, mrank = T)
#>    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
#> 0.08487309 0.07900000 0.92200000 0.16500000 0.03726978 0.13467527
ecodist::mantel(rel_dist ~ f4_diff + f3_diff + f2_diff + f1_diff, mrank = T)
#>      mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
#> -0.06239643  0.89800000  0.10300000  0.20400000 -0.11248160 -0.02709051
```

Linear model for associations between genetic relatedness and factor scores as explanatory variables for pups.

```r
col_df <- cbind(factors["colony"], fa_scores)
col_reduced <- minmodelr::MinMod(col_df)
#>
#> Call:
#> glm(formula = depVar ~ ., family = family, data = bestmodeldf)
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -0.8127  -0.2569  -0.1038   0.2719   0.7243
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  1.51220    0.03696   40.91  < 2e-16 ***
#> F4           0.38454    0.03791   10.14 5.08e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for gaussian family taken to be 0.1120233)
#>
#>     Null deviance: 20.4878  on 81  degrees of freedom
#> Residual deviance:  8.9619  on 80  degrees of freedom
#> AIC: 57.179
#>
#> Number of Fisher Scoring iterations: 2
col_reduced_df <- col_reduced[[1]]
# dev_expl <- (col_reduced_df$null.deviance - col_reduced_df$deviance) / col_reduced_df$null.deviance
table <- minmodelr::DelTestVar(col_reduced[[1]])
#>              Estimate Deviance Explained        F   P (F-test)
#> (Intercept) 1.5121951                 NA       NA           NA
#> F4          0.3845353           56.25756 102.8887 5.082064e-16
```

```
#>             P (Chisquared-test)
#> (Intercept)                  NA
#> F4                 3.545071e-24
```

Colony differences in factor scores: Just factor 4 shows significant differences.

Identification of substance subsets.

```r
# subsets and identification
library(vegan)
library(ggplot2)
library(dplyr)
library(magrittr)
library(vegan)
library(reshape2)
```

Similarity percentages analysis (`simper`) identifies the contribution of a specific compound to group similarity / dissimilarity. ANOSIM was used to test whether a small subset of the compounds with the highest contributions shows significant patterns.

Identification of best substances encoding mother-offspring similarity. For this analysis we have 41 groups (mother-offspring pairs) and want to look at withing group similarities rather then between group dissimilarities. This was done in Primer-E, as the `simper` function from the vegan package computes discriminating compounds, rather then compounds that make a mother-pup pair unique (although both sets overlap of course) .

```r
# results from simper analysis in Primer-E
mp_simp <- read.csv(".\\files\\simper_mp_results.csv", colClasses = c("character", "numeric"))
mp_simp
#>           comp contrib
#> 1  19.72268293   15.54
#> 2  15.45769231   12.25
#> 3  26.78859155   11.97
#> 4   16.3974359   11.30
#> 5  19.52538462   10.87
#> 6       21.405    8.49
#> 7  37.56363636    6.48
#> 8  15.62272727    6.48
#> 9  33.63655172    6.28
#> 10 30.80365385    6.03
#> 11   20.361875    5.34
#> 12 17.40942623    4.79
```

Mother offspring similarity based on a Bray-curtis similarity matrix which was computed from just the subset of 12 top compounds from the SIMPER analysis is highly significant, both overall, as well as within colonies.

Full sample

```r
vegan::anosim(dat = scent[mp_simp$comp], grouping = factors$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[mp_simp$comp], grouping = factors$family,      permutations = 1000, distan
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.6787
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Within colony 1 (Special study beach)

```r
vegan::anosim(dat = scent[factors$colony == 1, mp_simp$comp],
       grouping = factors[factors$colony == 1, ]$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$colony == 1, mp_simp$comp],      grouping = factors[factors$colony
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.5304
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Within colony 2 (Freshwater beach)

```r
vegan::anosim(dat = scent[factors$colony == 2, mp_simp$comp],
       grouping = factors[factors$colony == 2, ]$family,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> vegan::anosim(dat = scent[factors$colony == 2, mp_simp$comp],      grouping = factors[factors$colony
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.3066
#>       Significance: 0.003996
#>
#> Permutation: free
#> Number of permutations: 1000
```

Identification of best substances encoding colony dissimilarity. Using `simper` from the `vegan` package to find the important substances for discriminating between the two colonies. And sorting them subsequently in order of contribution to colony dissimilarity.

```r
# simper analysis
simp_colony <- vegan::simper(scent, factors$colony)
```

```
# getting 15 best substances and their contribution to colony dissimilarity
simp_colony_names <- rownames(summary(simp_colony, ordered = TRUE)[[1]])[1:15]
contribution <- summary(simp_colony, ordered = TRUE)[[1]]$contr[1:15]

# indices of colony substances (58,62,68,74,86,89,90,98,106,107,110,164,181,189,211)
ind_col <- paste(which(names(scent)%in%simp_colony_names), collapse = ",")

# connect to data frame and compute contribution in percent
col_simp <- data.frame(comp = simp_colony_names, contrib = contribution*100, stringsAsFactors = FALSE)
col_simp
#>           comp  contrib
#> 1  15.45769231 3.006785
#> 2   16.3974359 2.419310
#> 3  26.78859155 2.069715
#> 4  19.52538462 1.965096
#> 5       21.405 1.894868
#> 6  21.34820513 1.671251
#> 7  19.72268293 1.669923
#> 8  30.80365385 1.482931
#> 9   38.5183871 1.440400
#> 10 17.40942623 1.330158
#> 11 20.51086207 1.288827
#> 12 33.63655172 1.266612
#> 13 21.57529412 1.208037
#> 14 15.74219178 1.180692
#> 15 19.66514286 1.128883
```

Colony dissimilarity based on 15 compounds.

```
# overall (number of permutations is 1000 instead of 10,000 in the paper)
anosim(dat = scent[col_simp$comp], grouping = factors$colony,
       distance = "bray", permutations = 1000)
#>
#> Call:
#> anosim(dat = scent[col_simp$comp], grouping = factors$colony,      permutations = 1000, distance = "
#> Dissimilarity: bray
#>
#> ANOSIM statistic R: 0.7726
#>       Significance: 0.000999
#>
#> Permutation: free
#> Number of permutations: 1000
```

Identification of substanced encoding relatedness.

All the following analyses are shown for the subset of mothers.

The core of the idea is to use a bootstrapping procedure on the `BIO-ENV` function, originally by Clarke (Clarke and Warwick 2001), which was modified (Taylor 2014) to work with a bray curtis similarity matrix. For details see the methods part of the paper. The function is built to run on parallel with snowfall (Knaus 2013) on a server or similar, but still takes a couple of days to finish.

Additional packages used are Hadley Wickham´s dplyr (Wickham and Francois 2015) and stringr (Wickham 2015).

```r
################ BIO-ENV bootstrap procedure ###################################
#### run seperately on multicore server #####
#### aim: resampling test for finding the substances associated with genetic
#### relatedness. Basic assumption: Each variable will be tested in many different
#### environments (individuals, other variables), which will prevent spurious
#### correlations, as the really important substances will occur in best subsets
#### in many different constellations. (see methods section)

# parallel computing using 40 cores, takes some days nevertheless and is just
# shown here.

library(vegan)
library(stringr)
library(dplyr)
library(snow)
library(snowfall)
source("bio.env.R")

# number of cores
ncores <- 2
# subset
scent_mum <- filter(scent, factors$age == 1)
relate_mum <-  relatedness[factors$age == 1, factors$age == 1]

# initialise results vector
all_best <- vector()

# initialise cluster
snowfall::sfInit(parallel=TRUE, cpus=ncores, type="SOCK")

# export libraries and main function to all cores
snowfall::sfSource("bio.env.R")
snowfall::sfLibrary(vegan)
snowfall::sfLibrary(stringr)
snowfall::sfLibrary(dplyr)

bootstrap <- function(iter_comp) { # main resampling function
        for (i in 1:500) {
                # sample 20 out of 41 mothers, indices
                ind_obs <- sort(sample(1:41, size = 20, replace = F))
                # subset relate_mum and scent_mum
                reltemp <- 1-as.dist(relate_mum[ind_obs, ind_obs])
                abundtemp <- scent_mum[ind_obs, ]
                for (i in iter_comp) {
                        # sample 10 compounds
                        index_comps <- sort(sample(1:213, size = 10, replace = F))
                        abundtemp_sub <- abundtemp[, index_comps]
                        # get vector with 0 for null-column and 1 for non-null column
                        nullcomps <- apply(abundtemp_sub, 2, function(x) sum(x>0))
                        abundtemp_sub <- subset(abundtemp_sub,
                                                subset = c(rep(TRUE,nrow(abundtemp_sub))),
                                                select = (nullcomps >= 2))
                        # new iteration if too less substances left
```

```
                        if (ncol(abundtemp_sub) <= 2) next
                        # main function: bio.env finds subset that mostly correlates
                        # with relatedness
                        results <- bio.env(reltemp, abundtemp_sub,
                                           var.dist.method = "bray",
                                           scale.fix = F, scale.var = F)
                        mods <- results$best.model.vars
                        best <- unlist(str_split(mods, ","))
                        all_best <- append(all_best, best)
                        # write(best, file = "best.txt", append = TRUE, sep = " ")
                }
        }
        return(all_best)
}
# export objects
snowfall::sfExportAll(except = NULL, debug = FALSE)
snowfall::sfClusterEval(ls())
# create list of 500 iterations for all cores
vals <- list()
for (i in 1:ncores) {
        vals[[i]] <- 1:500
}
# run analysis
# best is a list of all best subsets
best <- snowfall::sfLapply(vals, bootstrap)
# stop cluster
sfStop()
# bring all results
results <- unlist(best)

############################### END ##############################################
```

Analysing results from the BIO-ENV bootstrap analysis.

best_mums is a data frame containing the number of occurences of each variable in the best subset from the BIO-ENV bootstrap analysis. Substances, that were retained more often are therefore likely to be genuinly associated with genetic relatedness.

```
# substance occurences are sorted in the table
best_mums <- read.csv("files/bootstrap_mums.csv",row.names=1)
```

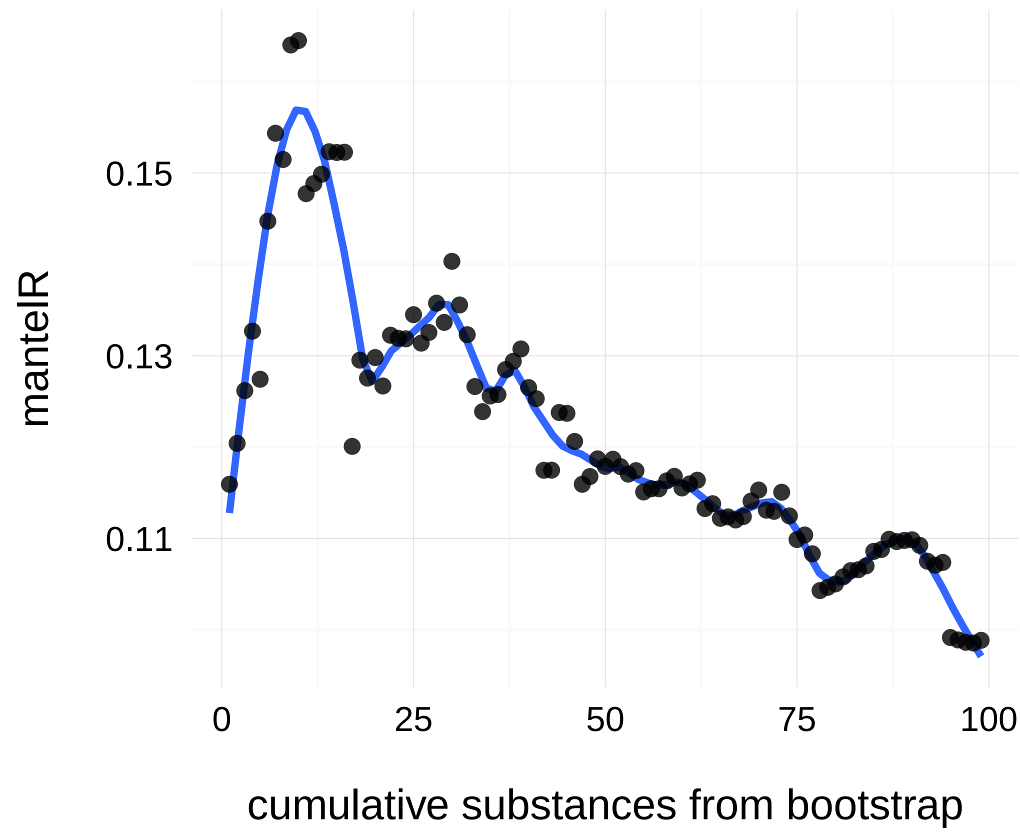
To analyse how many of these compounds are really important, the idea is to take an increasing number of "best" compounds and compute a mantel test with relatedness for each of the subsets. The subsequent plot shows a nice peak, which could be seen as the optimal number of chemicals encoding relatedness.

```
# subset mothers
scent_mum <- dplyr::filter(scent, factors$age == 1)
relate_mum <-  1-relatedness[factors$age == 1, factors$age == 1]

sub_names_mums <- row.names(best_mums)

statm <- vector()
sigm <- vector()
```

```r
# compute mantelR for an increasing set of best substances
for (i in 2:100) {
        bc_dist <- vegan::vegdist(scent_mum[, sub_names_mums[1:i]], method = "bray")
        mod <- vegan::mantel(relate_mum, bc_dist, na.rm = T, method = "spearman")
        statm <- append(statm, mod$statistic)
        sigm <- append(sigm, mod$sig)
}

stat_df <- data.frame(num_comps = 1:length(statm), mantelR = statm)
```

Plotting mantelR for an increasing number of best substances.

```r
library(grid)
# simple plot
ggplot2::ggplot(stat_df, aes(x = num_comps, y = mantelR)) +
        stat_smooth(se = FALSE, span = 0.16, size = 1.3, method = "loess") +
        geom_point(colour = "black", size = 3, alpha = 0.8) +
        theme_minimal(base_size = 16) +
        theme(strip.text.x = element_text(vjust=1,size = 16),
              axis.title.x = element_text(vjust= -2 ,size =16),
              axis.title.y = element_text(vjust=3,size = 16),
              axis.ticks.x = element_blank(),
              axis.ticks.y = element_blank(),
              plot.margin = (unit(c(.5, .5, 2, 2), "cm"))) +
        # scale_x_continuous(breaks=c(seq(from = 0.8, to = 1.20, by = 0.1))) +
        #geom_text(aes(0.85,80, label="(a) r = 0.34, p = 0.027"),size=4) +
        xlab("cumulative substances from bootstrap") +
        ylab("mantelR")
```



The plot peaks at 10 substances. We now want to do a single mantel test for chemical bray-curtis similarities based on these 10 compounds and genetic relatedness. As already shown in the plot, the mantelR is 0.164 and is highly significant.

```
# indices of the 10 best compounds associated with relatedness ------------------
comp_ind_m <- c(36,52,86,88,96,103,110,203,206,207)

# bray curtis similarity matrix based on this 10 compounds
scent_bc <- 1-(as.matrix(vegan::vegdist(as.matrix(scent[factors$age == 1, comp_ind_m])),
                      method = "bray"))
# relatedness matrix
rel_m <- relatedness[1:41, 1:41]

# mantel test for association between both
vegan::mantel(rel_m, scent_bc, method = "spearman", permutation = 1000, na.rm = TRUE)
#>
#> Mantel statistic based on Spearman's rank correlation rho
#>
#> Call:
#> vegan::mantel(xdis = rel_m, ydis = scent_bc, method = "spearman",      permutations = 1000, na.rm =
#>
#> Mantel statistic r: 0.164
#>       Significance: 0.002997
```

```
#>
#> Upper quantiles of permutations (null model):
#>    90%    95%  97.5%    99%
#> 0.0720 0.0906 0.1131 0.1246
#> Permutation: free
#> Number of permutations: 1000
```


Clarke, K Robert, and Richard Martyn Warwick. 2001. *PRIMER V5: User Manual/tutorial*. PRIMER-E Limited.

David, Patrice, Benoit Pujol, Frederique Viard, Vincent Castella, and JEROMe Goudet. 2007. "Reliable Selfing Rate Estimates from Imperfect Population Genetic Data." *Molecular Ecology* 16 (12). Wiley Online Library: 2474–87.

Goslee, Sarah C., and Dean L. Urban. 2007. "The Ecodist Package for Dissimilarity-Based Analysis of Ecological Data." *Journal of Statistical Software* 22 (7): 1–19.

Knaus, Jochen. 2013. *Snowfall: Easier Cluster Computing (Based on Snow)*. http://CRAN.R-project.org/package=snowfall.

McFerrin, Lisa. 2013. *HDMD: Statistical Analysis Tools for High Dimension Molecular Data (HDMD)*. http://CRAN.R-project.org/package=HDMD.

Oksanen, Jari, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, and Helene Wagner. 2015. *Vegan: Community Ecology Package*. http://CRAN.R-project.org/package=vegan.

Pritchard, J K, M Stephens, and P Donnelly. 2000. "Inference of population structure using multilocus genotype data." *Genetics*.

Queller, D C, and K F Goodnight. 1989. "Estimating relatedness using genetic markers." *Evolution*.

Taylor, Marc. 2014. *Sinkr: A Collection of Functions Featured on the Blog 'Me Nugget'*. https://github.com/menugget/sinkr.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer New York. http://had.co.nz/ggplot2/book.

———. 2015. *Stringr: Simple, Consistent Wrappers for Common String Operations*. http://CRAN.R-project.org/package=stringr.

Wickham, Hadley, and Romain Francois. 2015. *Dplyr: A Grammar of Data Manipulation*. http://CRAN.R-project.org/package=dplyr.