R script

Preamble

We made extensive use of several online resources that explain how to implement animal models using the MCMCglmm R package ([1]) and how to estimate narrow sense heritability (h^2) and genetic correlations (r_G) of traits. We want to acknoweledge these resources and, at the same time, encourage readers interested in using the animal model to estimate quantitative genetic parameters of biological traits to consult them.
Tutorial 1 "Estimation of a biological trait heritability using the animal model - How to use the MCMCglmm R package" by Pierre de Villemereuil.
Tutorial 2 How to implement univariate ,bivariate and repeated measures animal models in MCMCglmm. This tutorial represent the Supplementary file 5 in [2].
Course Notes MCMCglmm Course Notes by Jarrod Hadfield.

1. Estimating the heritability of CHCs

NB We ran a univariate MCMCglmm model for each cuticular hydrocarbon compound (32 models in total). The script below represents the pipeline used to estimate the heritability of a single compound (Peak 1: x-$C_{25:1}$).

Load packages

```
library(MCMCglmm)
library(nadiv)
library(ggplot2)
```

Packages (and dependencies) version is indicated below.

```
##    ggplot2      nadiv   MCMCglmm        ape       coda     Matrix
##    "3.0.0" "2.16.0.0"     "2.28"      "5.1"   "0.19-1"   "1.2-11"
```

Load datasets

```
PEDIGREE=read.table("Genotype Pedigree - Heritability.txt", header=T)# genotype pedigree
PEDIGREE.COLONY=read.table("Colony Pedigree - Heritability.txt",header=T)# colony info
CHC=read.table("CHC - Heritability.txt",header=T)# log-ratio transformed CHCs areas
```

Encode CHC dataset variables

- Create the "animal" variable - This variable represents the colony replicate.
  NB *The variable must have the same name of the variable contained in the pedigree file*

- Create the "ID" variable. This factor is the same as the "animal" variable.
  NB *The variable allows to disassociate individual records from the pedigree, modelling the non-additive contributions to fixed among-individual differences (permanent environment effects). To prevent upward bias in the additive genetic variance (V_a), this variable must be included in repeated measures models*

```
CHC$Block<-as.factor(CHC$Block)# convert block vector into factor
CHC$Colony<-as.factor(CHC$Colony)# convert colony vector into factor
CHC$animal<-interaction(CHC$Genotype,CHC$Colony,drop=T)# "animal" variable
CHC$ID<-CHC$animal# "ID" variable
```

Encode colony info dataset variables

```
PEDIGREE.COLONY$Block<-as.factor(PEDIGREE.COLONY$Block)# convert block into factor
PEDIGREE.COLONY$Colony<-as.factor(PEDIGREE.COLONY$Colony)# convert colony into factor
```

Encode the final pedigree dataset

```
NEW.PEDIGREE=merge(PEDIGREE.COLONY,PEDIGREE,by.x="Genotype",by.y="animal",all.X=TRUE)
NEW.PEDIGREE$animal=interaction(NEW.PEDIGREE$Genotype,NEW.PEDIGREE$Colony,drop = T)
NEW.PEDIGREE=NEW.PEDIGREE[c("animal","dam","sire","sex")]
PEDIGREE<-PEDIGREE[c("animal","dam","sire","sex")]
FINAL.PEDIGREE=rbind(NEW.PEDIGREE,PEDIGREE) # pedigree
rm(NEW.PEDIGREE,PEDIGREE,PEDIGREE.COLONY)
```

Create the inverse of the additive genetic relationship matrix

The standard pipeline for analysis in MCMCglmm expects diploid genetics. Because ants are haplodiploid, we used the function makeS() in the nadiv package [3] to construct a sex-chromosomal additive genetic relatedness matrix. This relatedness matrix reflects the covariance among relatives as a result of the different inheritance patterns of sex chromosomes as compared to the authosomes. This inheritance pattern is the same as what would be expected in haplodiploid organisms (see [4]). Here, the heterogametic sex is the male.

```
Mat_A <- makeS(prepPed(FINAL.PEDIGREE),heterogametic="M",returnS=T)
```

Formulate univariate priors for the MCMCglmm model

Often used univariate priors correspond to an inverse-Gamma distribution with shape and scale parameters equal to 0.01. However, this inverse-Gamma could become unwantedly 'informative' when variance components tend to 0 (see [5]), which was our case for certain CHC compounds. We ran models with different prior specifications (not showed), until we found priors that ensured convergence of the model, no autocorrelation and credible heritability estimates for all the 32 univariate models.

```
prior <- list(R = list(V=1, nu=1.002),# R represent the fixed effect term
              G = list(G1 = list(V=1, nu=1.002),# G represents the random effect part
              G2 = list(V=1, nu=1.002),
              G3 = list(V=1, nu=1.002)))
```

The MCMCglmm model

The distribution of the CHC response variable is set to "gaussian". The model includes "animal", "ID" and "Block" as random effect terms, which represent the variance components of the total phenotypic variance. "animal" factor represent the additive genetic variance; "ID" models the permanent environment effect deriving from the repeated measures structure of the model; "Block" is an extra-source of environmental variation due to the sampling of colonies at different time intervals. The model also includes the relationship matrix previously created. Each model runs for 1 million iterations (*nitt*) and has a burning step of 10000 iterations (*burnin*). The sampling is made every 500 data points (*thin*). These settings results in a maximum sample size of 1980 permutations.

```
MCMC <- MCMCglmm(Peak1~ 1, random=~ID+animal+Block,
                 ginverse=list(animal=Mat_A$Sinv),
                 nitt = 1e6, burnin = 1e4, thin = 5e2,
                 family = c(rep("gaussian",1)),
                 pl = T , pr = T, data = CHC,prior = prior)
```

Model diagnostic

1. We first checked that the autocorrelation across montecarlo chains for all variables was as close to zero as possible for all lag values greater than zero (values below 0.1 are considered acceptable).

```
autocorr(MCMC$VCV)
```

```
## , , ID
##
##                      ID       animal        Block       units
## Lag 0      1.0000000000 -0.169555245 -0.030905799 -0.05720418
## Lag 500   -0.0394814139 -0.005307975  0.013885904 -0.03563460
## Lag 2500  -0.0311332978 -0.010750014 -0.001739255  0.02253845
## Lag 5000   0.0009561665  0.027381157  0.002953666  0.01505842
## Lag 25000  0.0061168285  0.005835220  0.013638345  0.01273553
##
## , , animal
##
##                     ID       animal        Block        units
## Lag 0     -0.169555245  1.000000000 -0.026203963 -0.001422314
## Lag 500    0.011769781  0.014137884 -0.058845962  0.031625670
## Lag 2500   0.014287648 -0.015147288  0.006829835  0.029549432
## Lag 5000   0.029581183 -0.003089071  0.020284515  0.010954501
## Lag 25000 -0.008896085 -0.012391405  0.032318369 -0.007183878
##
## , , Block
##
##                    ID       animal       Block        units
## Lag 0     -0.03090580 -0.026203963  1.00000000 -0.025574982
## Lag 500   -0.05382054  0.018605298 -0.01475671 -0.012455652
## Lag 2500   0.03869399  0.003179041 -0.02771510  0.000170783
## Lag 5000   0.01207161  0.002298699  0.01294678  0.033366093
## Lag 25000  0.02959508  0.007924578  0.02655532 -0.006094292
##
## , , units
##
##                     ID        animal       Block        units
## Lag 0     -0.057204177 -0.0014223143 -0.02557498  1.000000000
## Lag 500    0.048618287 -0.0014070140  0.00538656 -0.017994076
## Lag 2500   0.017076699  0.0163947225 -0.01498444  0.035464874
## Lag 5000  -0.004131627  0.0003095034 -0.03978819 -0.006417577
## Lag 25000  0.007241886 -0.0263569913  0.01366125  0.026073509
```

2. We checked for model convergence.

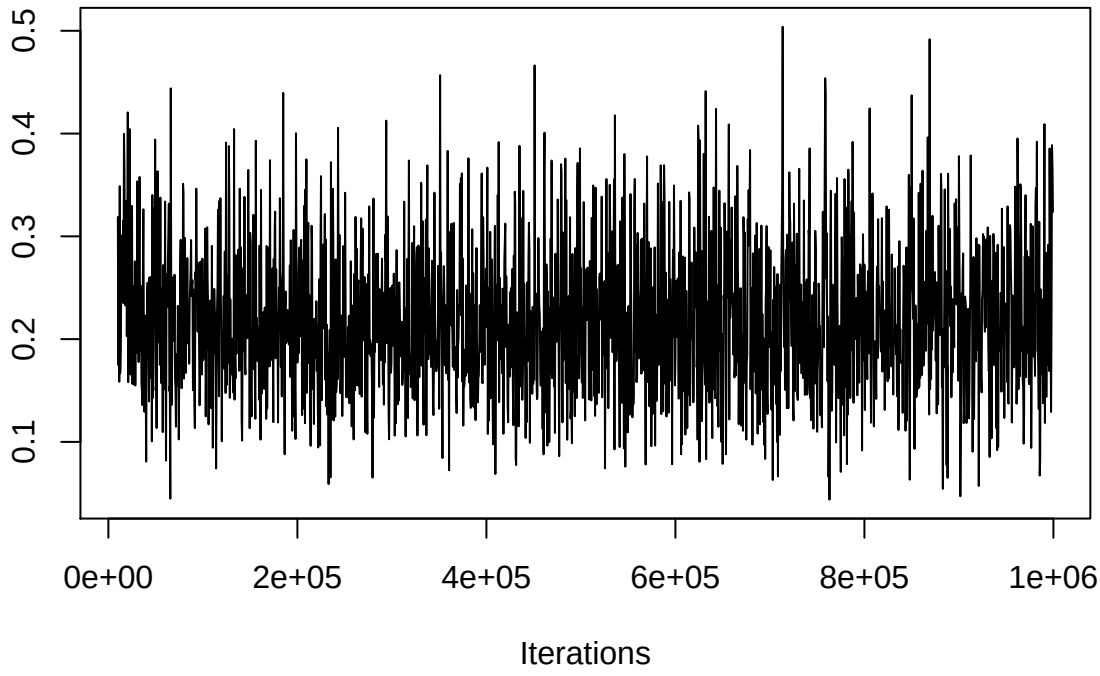
```
heidel.diag(MCMC$VCV)
```

```
##
##        Stationarity start     p-value
##        test         iteration
## ID     passed       1         0.129
## animal passed       1         0.370
## Block  passed       1         0.482
## units  passed       1         0.287
##
##        Halfwidth Mean   Halfwidth
##        test
## ID     passed    0.0734 0.000787
```

```
## animal passed    0.1094 0.001418
## Block  passed    0.2845 0.006850
## units  passed    0.0611 0.000310
```

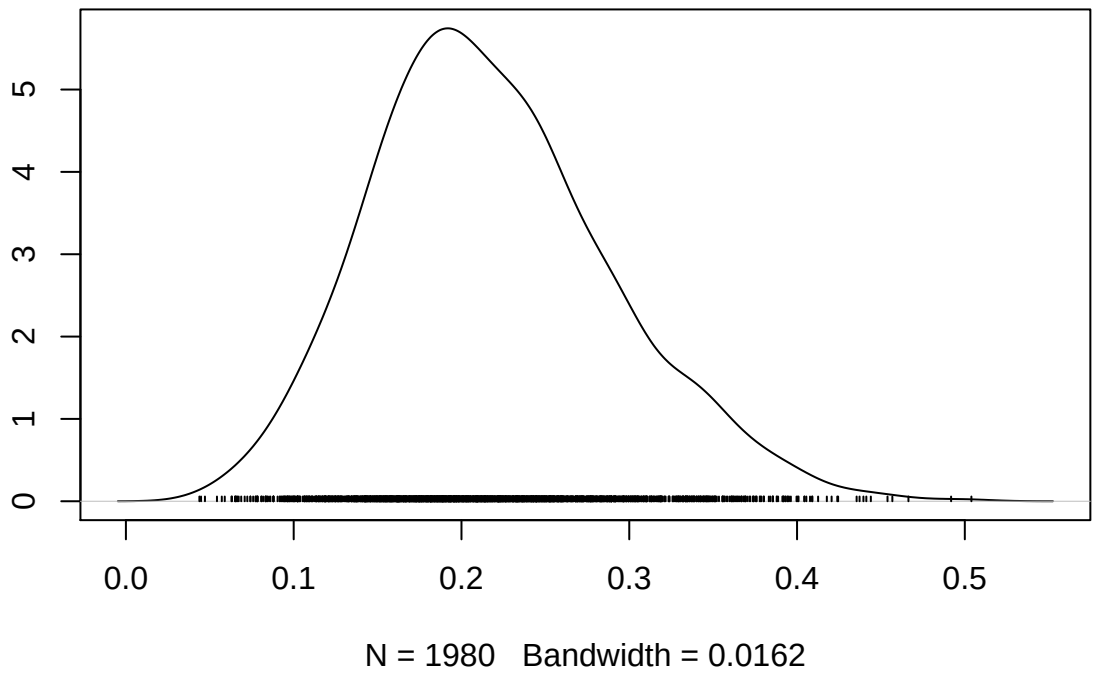
3. We visually inspected the well mixing of chains for the "animal" variable using the function traceplot(). If chains mixed well, the graph should look as a fuzzy caterpillar, without any visible trend.

```
traceplot((MCMC$VCV[, "animal"])/(MCMC$VCV[, "ID"] +
           MCMC$VCV[, "units"]+MCMC$VCV[, "animal"] +
           MCMC$VCV[, "Block"]))
```



4. We visually inspected the posterior density plot of the "animal" parameter using the function densplot(). The distribution, ideally, should look unimodal.

```
densplot((MCMC$VCV[, "animal"])/(MCMC$VCV[, "ID"] +
          MCMC$VCV[, "units"]+MCMC$VCV[, "animal"] +
          MCMC$VCV[, "Block"]))
```



5. We ensured that the model retained enough independent runs (at least 1500) by looking at the effective sample size for the "animal" parameter.

```
effectiveSize((MCMC$VCV[, "animal"])/(MCMC$VCV[, "ID"] +
               MCMC$VCV[, "units"]+MCMC$VCV[, "animal"] +
               MCMC$VCV[, "Block"]))
```

```
## var1
## 1980
```

Calculate posterior heritability and the associated 95% confidence intervals

Narrow sense heritability (h^2) is defined as the proportion of the phenotypic variation that is due to additive genetic values, and is calculated as follow: $h^2 = \frac{V_A}{V_P}$. h^2 and associated 95% confidence intervals can be easily calculated from the model dividing the additive genetic variance term by the total phenotypic variance.

Posterior heritability

NB *Note that as this is an MCMC analysis the output obtained will differ slightly between model runs and hence you are unlikely to get exactly the same values as here*

```
posterior.mode((MCMC$VCV[, "animal"])/(MCMC$VCV[, "ID"] +
                MCMC$VCV[, "units"]+MCMC$VCV[, "animal"] +
                MCMC$VCV[, "Block"]))
```

```
##      var1
## 0.1890774
```

Confidence intervals

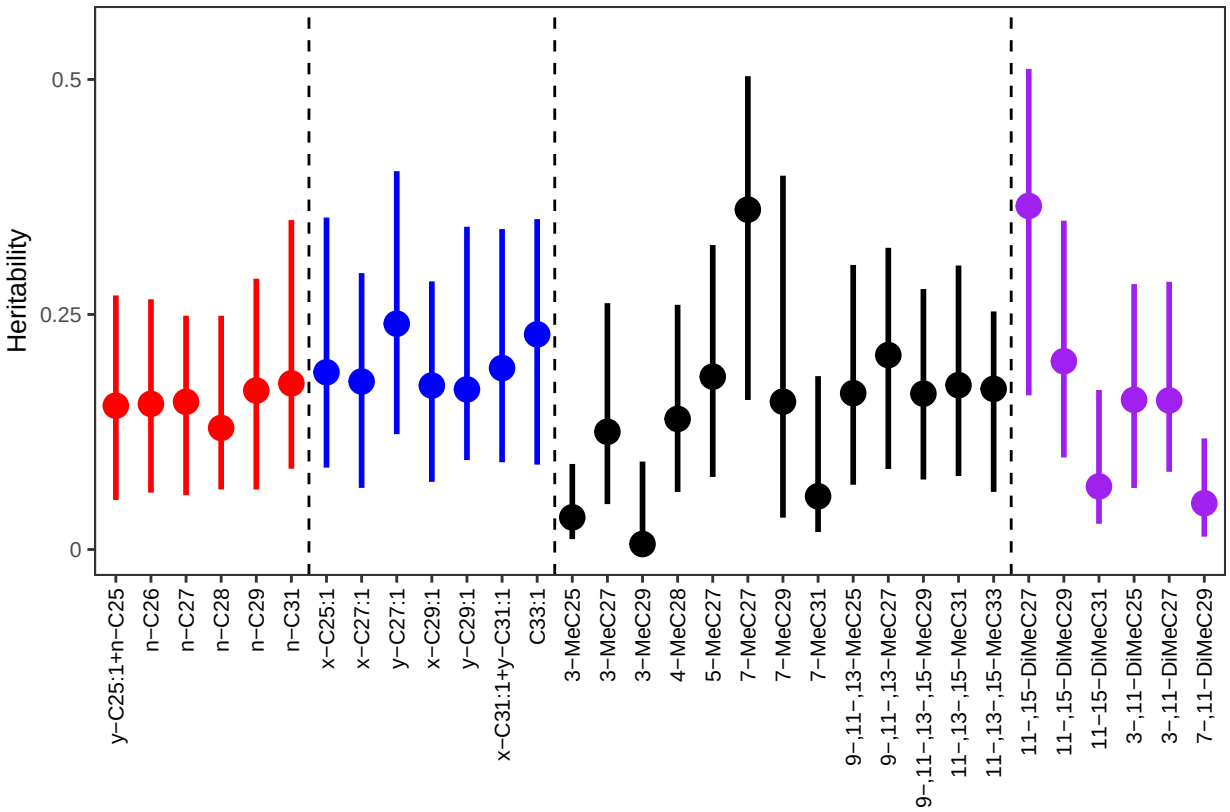
```
HPDinterval((MCMC$VCV[, "animal"])/(MCMC$VCV[, "ID"] +
             MCMC$VCV[, "units"]+MCMC$VCV[, "animal"] +
             MCMC$VCV[, "Block"]))
```

```
##           lower     upper
## var1 0.09198999 0.3692351
## attr(,"Probability")
## [1] 0.95
```

Posterior heritability and confidence interval values for each CHC are stored in the file **Results heritability.txt**. The file can be used to reproduce the plot presented in figure 1b in the paper (code below). **NB** *The first peak in the plot should be blue/red as it is a mix of two compounds belonging to different CHC classes (linear alkanes and alkenes)*

Heritability plot (Figure 1b)

```
DATA=read.table("Results heritability.txt", header=T)
DATA$Compound <- factor(DATA$Compound,levels =
                 c("y-C25:1+n-C25","n-C26","n-C27",
                   "n-C28","n-C29","n-C31","x-C25:1",
                   "x-C27:1", "y-C27:1","x-C29:1",
                   "y-C29:1","x-C31:1+y-C31:1","C33:1",
                   "3-MeC25","3-MeC27","3-MeC29","4-MeC28",
                   "5-MeC27","7-MeC27","7-MeC29","7-MeC31",
                   "9-,11-,13-MeC25","9-,11-,13-MeC27",
                   "9-,11-,13-,15-MeC29","11-,13-,15-MeC31",
                   "11-,13-,15-MeC33","11-,15-DiMeC27",
                   "11-,15-DiMeC29","11-15-DiMeC31","3-,11-DiMeC25",
                   "3-,11-DiMeC27", "7-,11-DiMeC29"))
ggplot(DATA, aes(x=Compound, y=estimate,color= Compound))+
  geom_point(size=4)+
  geom_errorbar(width=0, size=1, aes(ymin=low, ymax=up,
                                     colour= Compound))+
  scale_color_manual("Compound", breaks=c(1:32),
                     values=c(rep("red",6),rep("blue",7),
                              rep("black",13), rep("purple",6)))+
  xlab("")+
  ylab("Heritability")+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        axis.text.x = element_text(color="black",
                                   angle=90,hjust=1,vjust=0.5),
        panel.grid.minor = element_blank())+
  theme(text = element_text(size = 10))+
  scale_y_continuous(limits=c(0,0.55),
                     breaks=c(0,0.25,0.5), labels=c(0,0.25,0.5))+
  geom_vline(xintercept=c(6.5), linetype="dashed")+
  geom_vline(xintercept=c(13.5), linetype="dashed")+
  geom_vline(xintercept=c(26.5), linetype="dashed")
```



REFERENCES

1. Hadfield, J.D. (2010). MCMC methods for multi-response generalized linear mixed models: The mcmcglmm r package. 2010 *33*, 22. Available at: https://www.jstatsoft.org/v033/i02.

2. Wilson, A.J., Réale, D., Clements, M.N., Morrissey, M.M., Postma, E., Walling, C.A., Kruuk, L.E.B., and Nussey, D.H. (2010). An ecologist's guide to the animal model. Journal of Animal Ecology *79*, 13–26. Available at: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2656.2009.01639.x.

3. Wolak, M.E. (2012). Nadiv: An r package to create relatedness matrices for estimating non-additive genetic variances in animal models. Methods in Ecology and Evolution *3*, 792–796. Available at: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/j.2041-210X.2012.00213.x.

4. Sheehan Michael, J., Choo, J., and Tibbetts Elizabeth, A. Heritable variation in colour patterns mediating individual recognition. Royal Society Open Science *4*, 161008. Available at: https://doi.org/10.1098/rsos.161008.

5. Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). Bayesian Anal. *1*, 515–534. Available at: https://projecteuclid.org:443/euclid.ba/1340371048.

2) Estimating Genetic correlation across CHCs

We ran a bivariate MCMCglmm animal model for each pairwise combination of CHC variables in our dataset to estimate the genetic correlation (r_G) between compounds (total of 496 bivariate models). The script below represents the pipeline used to estimate the genetic correlation between peak 21 (n-C_{26}) and peak 29 (n-C_{31}). **NB** *the code is the same as the one specified in section 1 - Estimating the heritability of CHCs. The only change is represented by the priors structure, now reflecting the bivariate nature of the model.*

Load packages

```
library(MCMCglmm)
library(nadiv)
```

Load datasets

```
PEDIGREE=read.table("Genotype Pedigree - Heritability.txt", header=T)# genotype pedigree
PEDIGREE.COLONY=read.table("Colony Pedigree - Heritability.txt",header=T)# colony info
CHC=read.table("CHC - Heritability.txt",header=T)# log-ratio transformed CHCs areas
```

Encode CHC dataset variables

```
CHC$Block<-as.factor(CHC$Block)# convert block vector into factor
CHC$Colony<-as.factor(CHC$Colony)# convert colony vector into factor
CHC$animal<-interaction(CHC$Genotype,CHC$Colony,drop=T)# "animal" variable
CHC$ID<-CHC$animal# "ID" variable
```

Encode colony info dataset variables

```
PEDIGREE.COLONY$Block<-as.factor(PEDIGREE.COLONY$Block)# convert block into factor
PEDIGREE.COLONY$Colony<-as.factor(PEDIGREE.COLONY$Colony)# convert colony into factor
```

Create the final pedigree dataset

```
NEW.PEDIGREE=merge(PEDIGREE.COLONY,PEDIGREE,by.x="Genotype",by.y="animal",all.X=TRUE)
NEW.PEDIGREE$animal=interaction(NEW.PEDIGREE$Genotype,NEW.PEDIGREE$Colony,drop = T)
NEW.PEDIGREE=NEW.PEDIGREE[c("animal","dam","sire","sex")]
PEDIGREE<-PEDIGREE[c("animal","dam","sire","sex")]
FINAL.PEDIGREE=rbind(NEW.PEDIGREE,PEDIGREE) # pedigree
rm(NEW.PEDIGREE,PEDIGREE,PEDIGREE.COLONY)
```

Create the inverse of the additive genetic relationship matrix

```
Mat_A <- makeS(prepPed(FINAL.PEDIGREE),heterogametic="M",returnS=T)
```

Formulate bivariate priors for the model

```
prior = list(R=list(V=diag(2),nu=1.002),
             G=list(G1 =list(V=diag(2),nu=1.002),
             G2=list(V=diag(2),nu=1.002),
             G3=list(V=diag(2),nu=1.002)))
```

The MCMCglmm model

```
CHC.1<-as.numeric(CHC$Peak21)
CHC.2<-as.numeric(CHC$Peak29)
```

```
MCMC<-MCMCglmm(cbind(CHC.1,CHC.2)~trait-1,
               random=~us(trait):Block+us(trait):animal+
               us(trait):ID,
               rcov=~us(trait):units,
               ginverse=list(animal=Mat_A$Sinv),
               nitt = 1e6, burnin = 1e4, thin =5e2,
               family = c(rep("gaussian",1),rep("gaussian",1)),
               pl = T , pr = T, data = CHC,prior = prior)
```

Model diagnostic

We ran the same model diagnostic used in the univariate model. For the sake of simplicity, here we show only the results obtained for the test to check for model convergence.

1. convergence

```
heidel.diag(MCMC$VCV)
```

```
##
##                              Stationarity start     p-value
##                              test         iteration
## traitCHC.1:traitCHC.1.Block  passed       199       0.4253
## traitCHC.2:traitCHC.1.Block  passed         1       0.1934
## traitCHC.1:traitCHC.2.Block  passed         1       0.1934
## traitCHC.2:traitCHC.2.Block  passed         1       0.8589
## traitCHC.1:traitCHC.1.animal passed         1       0.6700
## traitCHC.2:traitCHC.1.animal passed         1       0.7789
## traitCHC.1:traitCHC.2.animal passed         1       0.7789
## traitCHC.2:traitCHC.2.animal passed         1       0.6962
## traitCHC.1:traitCHC.1.ID     passed         1       0.3707
## traitCHC.2:traitCHC.1.ID     passed         1       0.3245
## traitCHC.1:traitCHC.2.ID     passed         1       0.3245
## traitCHC.2:traitCHC.2.ID     passed         1       0.0844
## traitCHC.1:traitCHC.1.units  passed         1       0.1639
## traitCHC.2:traitCHC.1.units  passed         1       0.1795
## traitCHC.1:traitCHC.2.units  passed         1       0.1795
## traitCHC.2:traitCHC.2.units  passed         1       0.2568
##
##                              Halfwidth Mean   Halfwidth
##                              test
## traitCHC.1:traitCHC.1.Block  passed    0.3664 0.009677
## traitCHC.2:traitCHC.1.Block  passed    0.1587 0.006454
## traitCHC.1:traitCHC.2.Block  passed    0.1587 0.006454
## traitCHC.2:traitCHC.2.Block  passed    0.2588 0.006660
## traitCHC.1:traitCHC.1.animal passed    0.0928 0.001265
## traitCHC.2:traitCHC.1.animal passed    0.0482 0.001037
## traitCHC.1:traitCHC.2.animal passed    0.0482 0.001037
## traitCHC.2:traitCHC.2.animal passed    0.0887 0.001135
## traitCHC.1:traitCHC.1.ID     passed    0.0789 0.000863
## traitCHC.2:traitCHC.1.ID     passed    0.0459 0.000656
## traitCHC.1:traitCHC.2.ID     passed    0.0459 0.000656
## traitCHC.2:traitCHC.2.ID     passed    0.0677 0.000705
## traitCHC.1:traitCHC.1.units  passed    0.0856 0.000423
## traitCHC.2:traitCHC.1.units  passed    0.0559 0.000327
## traitCHC.1:traitCHC.2.units  passed    0.0559 0.000327
```

```
## traitCHC.2:traitCHC.2.units  passed    0.0628 0.000311
```

Calculate genetic correlation between CHCs and the associated 95% confidence intervals

The genetic correlation between two traits (r_G) is defined as the additive genetic covariance divided by the square-root of the product between the respective additive genetic variances.

$$r_G = \frac{Cov_{xy}}{\sqrt{(\sigma_x^2 \sigma_y^2)}}$$

Genetic correlation estimates and associated 95% confidence intervals can be extracted as follow.

```
genetic.correlation.CHC.1.CHC.2 =
  MCMC$VCV[,"traitCHC.1:traitCHC.2.animal"]/
  sqrt(MCMC$VCV[,"traitCHC.1:traitCHC.1.animal"]*
  MCMC$VCV[,"traitCHC.2:traitCHC.2.animal"])
```

Genetic correlation

```
posterior.mode(genetic.correlation.CHC.1.CHC.2)
```

```
##      var1
## 0.5644043
```

95% Confidence intervals

```
HPDinterval(genetic.correlation.CHC.1.CHC.2)
```

```
##          lower     upper
## var1 0.2188256 0.7519864
## attr(,"Probability")
## [1] 0.95
```

We deemed a genetic correlation to be statistically significant when the confidence intervals did not overlap with zero. Results are stored in the file **Results Genetic Correlation.txt**, which is used to build the correlation plot in Figure 2.

Genetic correlation plot (Figure 2)

Load packages

```
library(plyr)
library(dplyr)
library(reshape2)
library(corrplot)
library(RColorBrewer)
```

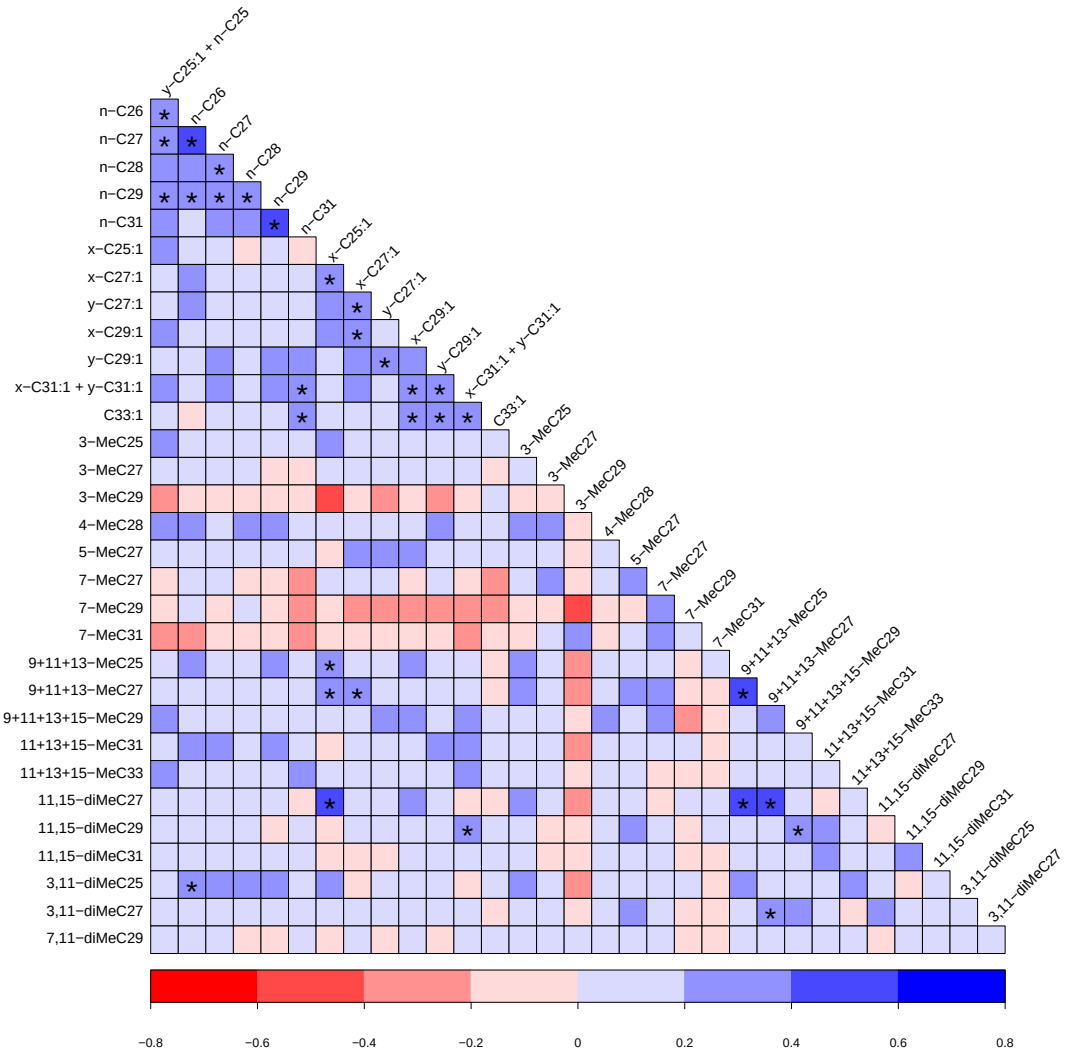
Packages (and dependencies) version is indicated below.

```
## RColorBrewer     corrplot     reshape2        dplyr         plyr
##      "1.1-2"       "0.84"      "1.4.3"      "0.7.6"      "1.8.4"
##        nadiv     MCMCglmm          ape         coda       Matrix
##   "2.16.0.0"       "2.28"        "5.1"     "0.19-1"     "1.2-11"
```

Load dataset

```
DATA=read.table("Results Genetic Correlation.txt",header=T,
                colClasses = c(rep("factor",4),
                rep("numeric",3),"factor"))
DATA$Significance=as.numeric(DATA$Significance)
DATA$Significance[DATA$Significance==2]=0.04
GEN.CORR=DATA[,c(1:2,5,8)]
GEN.CORR=droplevels(GEN.CORR)
GEN.CORR$Peak1 <- factor(GEN.CORR$Peak1, levels = c("2.3","6","10","16","21","29",
                                                    "1","8","9","19","20","27.28","33",
                                                    "5","15","26","18","13","12","23",
                                                    "31","4","11","22","30","34",
                                                    "14", "24", "32", "7","17","25"))
GEN.CORR$Peak2 <- factor(GEN.CORR$Peak2, levels = c("2.3","6","10","16","21","29",
                                                    "1","8","9","19","20","27.28","33",
                                                    "5","15","26","18","13","12","23",
                                                    "31","4","11","22","30","34",
                                                    "14", "24", "32", "7","17","25"))
GEN.CORR$Peak1=revalue(GEN.CORR$Peak1, c("1"="x-C25:1", "2.3"="y-C25:1 + n-C25",
                                         "4"="9+11+13-MeC25","5"="3-MeC25",
                                         "6"="n-C26", "7"="3,11-diMeC25","8"="x-C27:1",
                                         "9"= "y-C27:1","10"="n-C27",
                                         "11"= "9+11+13-MeC27","12"="7-MeC27",
                                         "13"= "5-MeC27","14"="11,15-diMeC27",
                                         "15"= "3-MeC27","16"="n-C28",
                                         "17"= "3,11-diMeC27","18"="4-MeC28",
                                         "19"= "x-C29:1","20"="y-C29:1",
                                         "21"= "n-C29","22"="9+11+13+15-MeC29",
                                         "23"= "7-MeC29","24"="11,15-diMeC29",
                                         "25"= "7,11-diMeC29","26" = "3-MeC29",
                                         "27.28"= "x-C31:1 + y-C31:1","29"= "n-C31",
                                         "30"="11+13+15-MeC31","31"= "7-MeC31",
                                         "32"="11,15-diMeC31","33"= "C33:1",
                                         "34"="11+13+15-MeC33"))
GEN.CORR$Peak2=revalue(GEN.CORR$Peak2, c("1"="x-C25:1", "2.3"="y-C25:1 + n-C25",
                                         "4"="9+11+13-MeC25","5"="3-MeC25",
                                         "6"="n-C26", "7"="3,11-diMeC25","8"="x-C27:1",
                                         "9"= "y-C27:1","10"="n-C27",
                                         "11"= "9+11+13-MeC27","12"="7-MeC27",
                                         "13"= "5-MeC27","14"="11,15-diMeC27",
                                         "15"= "3-MeC27","16"="n-C28",
                                         "17"= "3,11-diMeC27","18"="4-MeC28",
                                         "19"= "x-C29:1","20"="y-C29:1",
                                         "21"= "n-C29","22"="9+11+13+15-MeC29",
                                         "23"= "7-MeC29","24"="11,15-diMeC29",
                                         "25"= "7,11-diMeC29","26" = "3-MeC29",
                                         "27.28"= "x-C31:1 + y-C31:1","29"= "n-C31",
                                         "30"="11+13+15-MeC31","31"= "7-MeC31",
                                         "32"="11,15-diMeC31","33"= "C33:1",
                                         "34"="11+13+15-MeC33"))
GEN.CORR.ESTIMATE=GEN.CORR[,-4]
GEN.CORR.PVALUE=GEN.CORR[,-3]
GEN.CORR.ESTIMATE=t(acast(GEN.CORR.ESTIMATE, Peak1~Peak2, value.var="Estimate", drop=F))
diag(GEN.CORR.ESTIMATE)=0.8 #needed to set up the boundaries of the heatmap color scale
```

```
GEN.CORR.PVALUE=t(acast(GEN.CORR.PVALUE, Peak1~Peak2, value.var="Significance", drop=F))
par(xpd=TRUE)
corrplot(GEN.CORR.ESTIMATE, p.mat = GEN.CORR.PVALUE, method="color", type="lower",
         tl.srt = 45,cl.lim=c(-0.8,0.8), col=colorRampPalette(c("red",
                                                                "white","blue"))(8),
         insig="label_sig", pch.col="black",diag=F,pch.cex = 2, tl.col= "black",
         tl.cex=1, outline=T,mar=c(3,3,3,3))
```



3) Estimating linear and quadratic selection gradients

We calculated linear and quadratic selection gradients for the relationship between each cuticular hydrocarbon (phenotypic trait) and two measures of colony productivity: 1) sexual pupae production and 2) worker pupae production (fitness trait). We followed the approach outlined by Morrissey and Sakrejda in [1].
Briefly, we first estimated the fitness function relating colony productivity (sexual or worker production) to the abundance of a specific CHC with a generalized additive model (GAM), using the R package mgcv [2]. Then, linear and quadratic selection gradients are obtained from the fitted GAM model using the function gam.gradients() in the package gsg [3].
We ran a total of 64 univariate models (32 phenotypic traits $\times$ 2 fitness measures), and below we outline the code used to calculate linear and quadratic selection gradients for the relationship between Peak 1 (x-$C_{25:1}$) and each of the two measures of fitness.
NB *Because each colony replicate has one measure of colony productivity, but repeated CHC measures, we used the mean CHC value for each colony replicate as phenotypic measure*

Load packages

```
library(mgcv)
library(gsg)
library(AER)
library(car)
```

Packages (and dependencies) version is indicated below.

```
##       AER  survival  sandwich    lmtest       zoo       car   carData
##   "1.2-6"  "2.41-3"   "2.4-0"  "0.9-36"   "1.8-1"   "3.0-0"   "3.0-1"
##       gsg      mgcv      nlme
##     "2.0"  "1.8-20" "3.1-131"
```

Load datasets

```
Data=read.table("CHC - Selection.txt",header=T)
```

Standardize CHC peaks

Prior running the model, CHC variables are mean-centered and variance standardized.

```
Data$z1<-as.numeric(Data$Peak1) #Change for each peak
Data$z1<-(Data$z1-mean(Data$z1))/sd(Data$z1)
```

The general additive model (sexual pupae)

Because of the variation in sexual pupae production across colonies, with many colonies producing no sexual pupae, we used a "tweedie" distribution to model the fitness response. This distribution has nonnegative support and can have a discrete mass at zero, making it useful to model responses that are a mixture of zeros and positive values.

```
model.peak1.sexualpupae<-gam(SP~s(z1),family="tw",data=Data)
```

The general additive model (worker pupae)

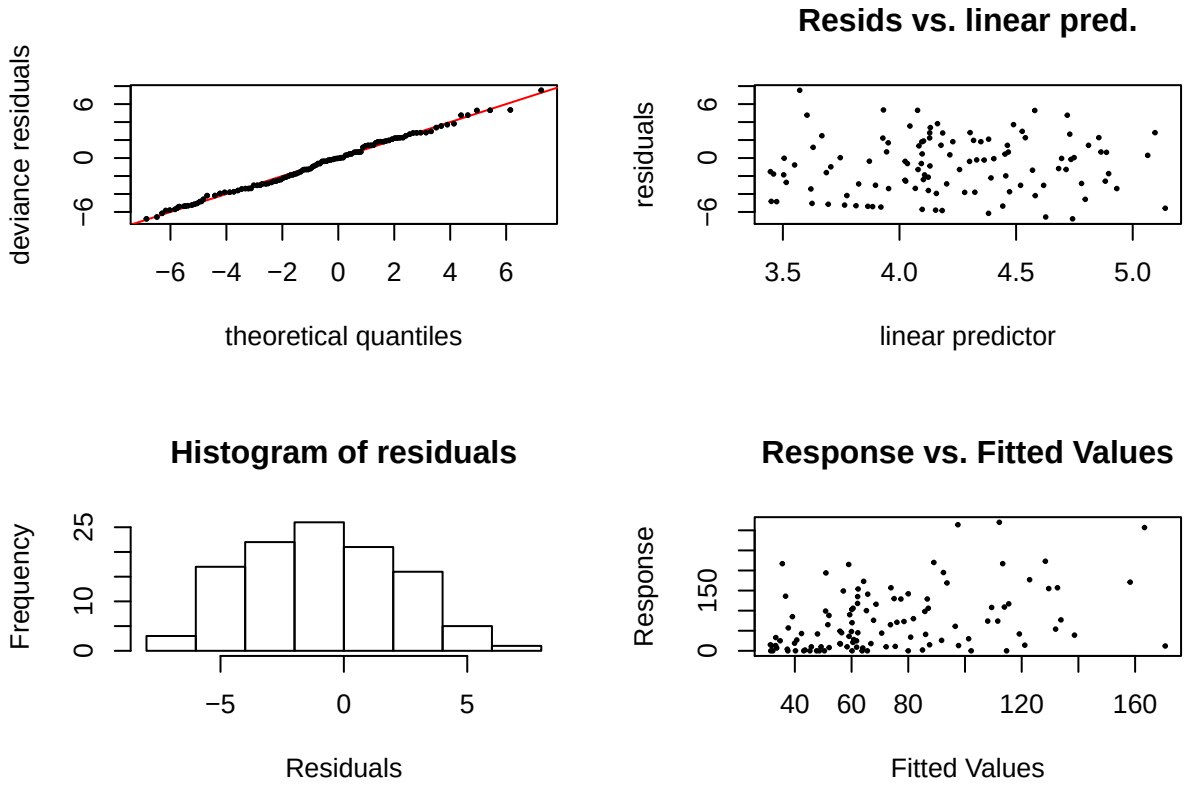
In the case of worker pupae production, we used a negative binomial distribution to account for the overdispersion in the data .

```
model.peak1.workerpupae<-gam(WP~s(z1),family="nb",data=Data)
```

Model diagnostic (sexual pupae)

We ran the function gam.check() to obtain standard diagnostic plots, smoothing parameter estimation convergence information and the results of tests which may indicate if the smoothing basis dimension for a term is too low.
Specifically, we checked that the k-index term does not go below 1 too much (which may indicate a missed pattern left in the residuals), that p-values are not low (which may indicate theat the basis dimension, k, has been set too low) and that the reported edf is not close to K'.

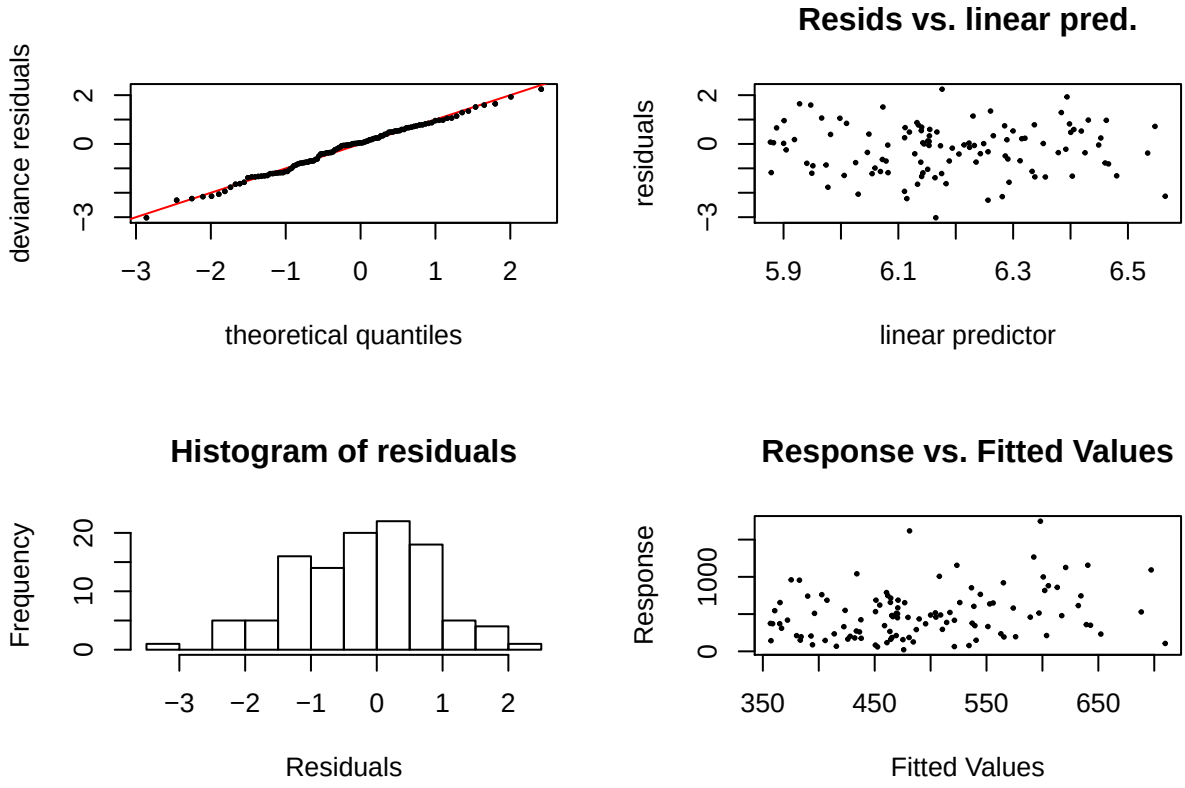
```
par(mfrow=c(2,2))
gam.check(model.peak1.sexualpupae, pch=19, cex=.3)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 8 iterations.
## Gradient range [-0.000234335,0.0001655369]
## (score 559.3386 & scale 8.619697).
## Hessian positive definite, eigenvalue range [0.0002343209,145.5348].
## Model rank =  10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##       k' edf k-index p-value
## s(z1)  9   1    0.92    0.44
```

Model diagnostic (worker pupae)

```
par(mfrow=c(2,2))
gam.check(model.peak1.workerpupae, pch=19, cex=.3)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-0.001702027,-0.0003616319]
## (score 785.0863 & scale 1).
## Hessian positive definite, eigenvalue range [0.0003696063,61.69522].
## Model rank =  10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##         k'  edf k-index p-value
## s(z1) 9.00 1.01    1.17    0.99
```

Estimate CHC selection gradients (sexual pupae)

We obtained linear and quadratic selection gradients, standard error and p-values through case bootstrapping.

```
fit.sexualpupae<-gam.gradients(mod=model.peak1.sexualpupae,phenotype="z1", standardized =F,
                   se.method = "boot.case",n.boot=10000,
                   refit.smooth = T)
```

Estimate CHC selection gradients (worker pupae)

```
fit.workerpupae<-gam.gradients(mod=model.peak1.workerpupae,phenotype="z1", standardized =F,
                   se.method = "boot.case",n.boot=10000,
                   refit.smooth = T)
```

Linear and Quadratic selection gradients estimates (sexual pupae)

B and G term represent the linear and quadratic selection gradients, respectively.

```
fit.sexualpupae$ests
```

```
##      estimates        SE P.value
## B-z1 0.4178793 0.1066749   0.002
## G-z1 0.1745629 0.2627137   0.391
```

Linear and Quadratic selection gradients estimates (worker pupae)

```
fit.workerpupae$ests
```

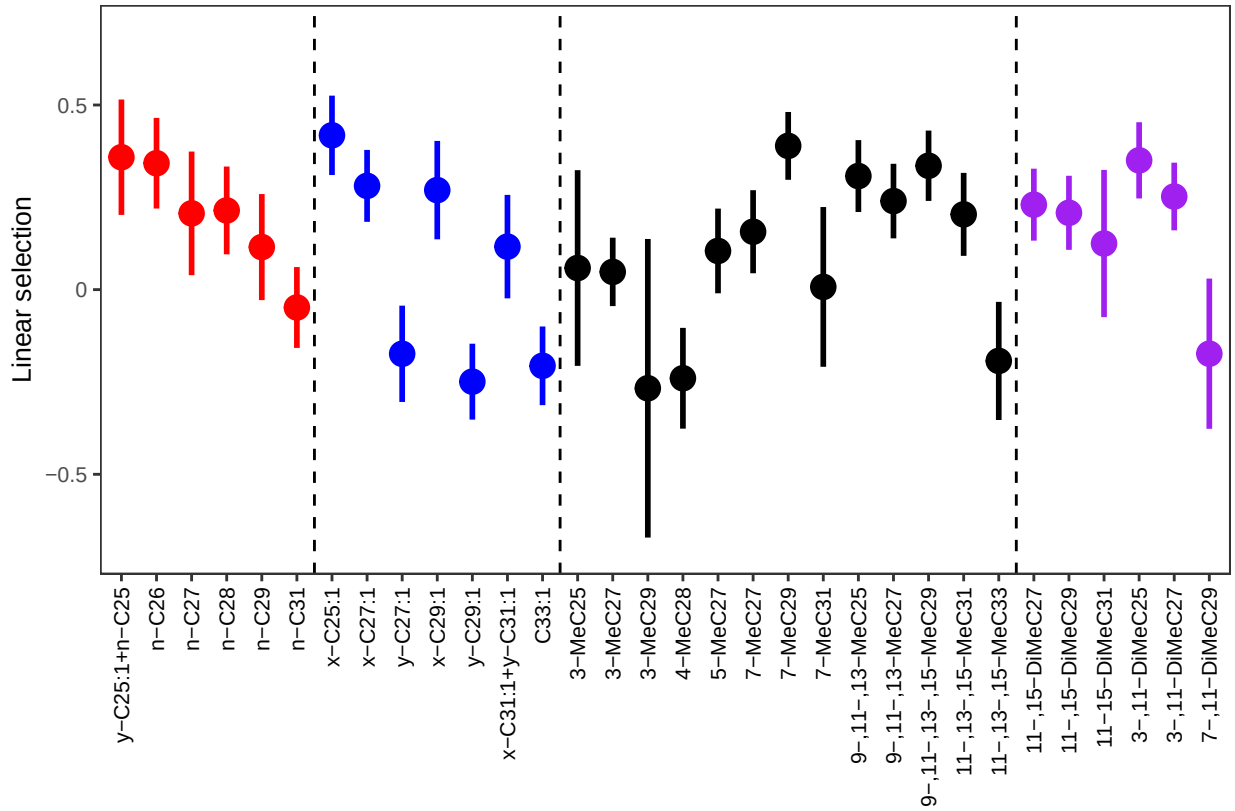
```
##       estimates         SE P.value
## B-z1 0.16985586 0.07617755  0.0498
## G-z1 0.02922135 0.20907201  0.3584
```

Linear and quadratic selection estimates, as well as associated standard errors and p-values, were stored in the file **Results Selection.txt**. We used the linear selection gradient estimates and SE for sexual pupae production for the plot presented in figure 3a of the manuscript.
NB *The first peak in the plot should be blue/red as it is a mix of two compounds belonging to different CHC classes (linear alkanes and alkenes)*

Plot (Figure 3a)

```
library(ggplot2)
DATA=read.table("Results Selection.txt", header=T)
DATA$Compound <- factor(DATA$Compound,levels =
                          c("y-C25:1+n-C25","n-C26","n-C27",
                            "n-C28","n-C29","n-C31","x-C25:1",
                            "x-C27:1", "y-C27:1","x-C29:1",
                            "y-C29:1","x-C31:1+y-C31:1","C33:1",
                            "3-MeC25","3-MeC27","3-MeC29","4-MeC28",
                            "5-MeC27","7-MeC27","7-MeC29","7-MeC31",
                            "9-,11-,13-MeC25","9-,11-,13-MeC27",
                            "9-,11-,13-,15-MeC29","11-,13-,15-MeC31",
                            "11-,13-,15-MeC33","11-,15-DiMeC27",
                            "11-,15-DiMeC29","11-15-DiMeC31","3-,11-DiMeC25",
                            "3-,11-DiMeC27", "7-,11-DiMeC29"))
ggplot(DATA[c(1:32),], aes(x=Compound, y=Linear,color= Compound))+
  geom_point(size=4)+
  geom_errorbar(width=0, size=1, aes(ymin=Linear-LinearSE, ymax=Linear+LinearSE,
                                     colour= Compound))+
  scale_color_manual("Compound", breaks=c(1:32),
                     values=c(rep("red",6),rep("blue",7),
                              rep("black",13), rep("purple",6)))+
  xlab("")+
  ylab("Linear selection")+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        axis.text.x = element_text(color="black",
```

```
                                angle=90,hjust=1,vjust=0.5),
      panel.grid.minor = element_blank())+
theme(text = element_text(size = 10))+
scale_y_continuous(limits=c(-0.7,0.7),
                   breaks=c(-0.5,0,0.5), labels=c(-0.5,0,0.5))+
geom_vline(xintercept=c(6.5), linetype="dashed")+
geom_vline(xintercept=c(13.5), linetype="dashed")+
geom_vline(xintercept=c(26.5), linetype="dashed")
```

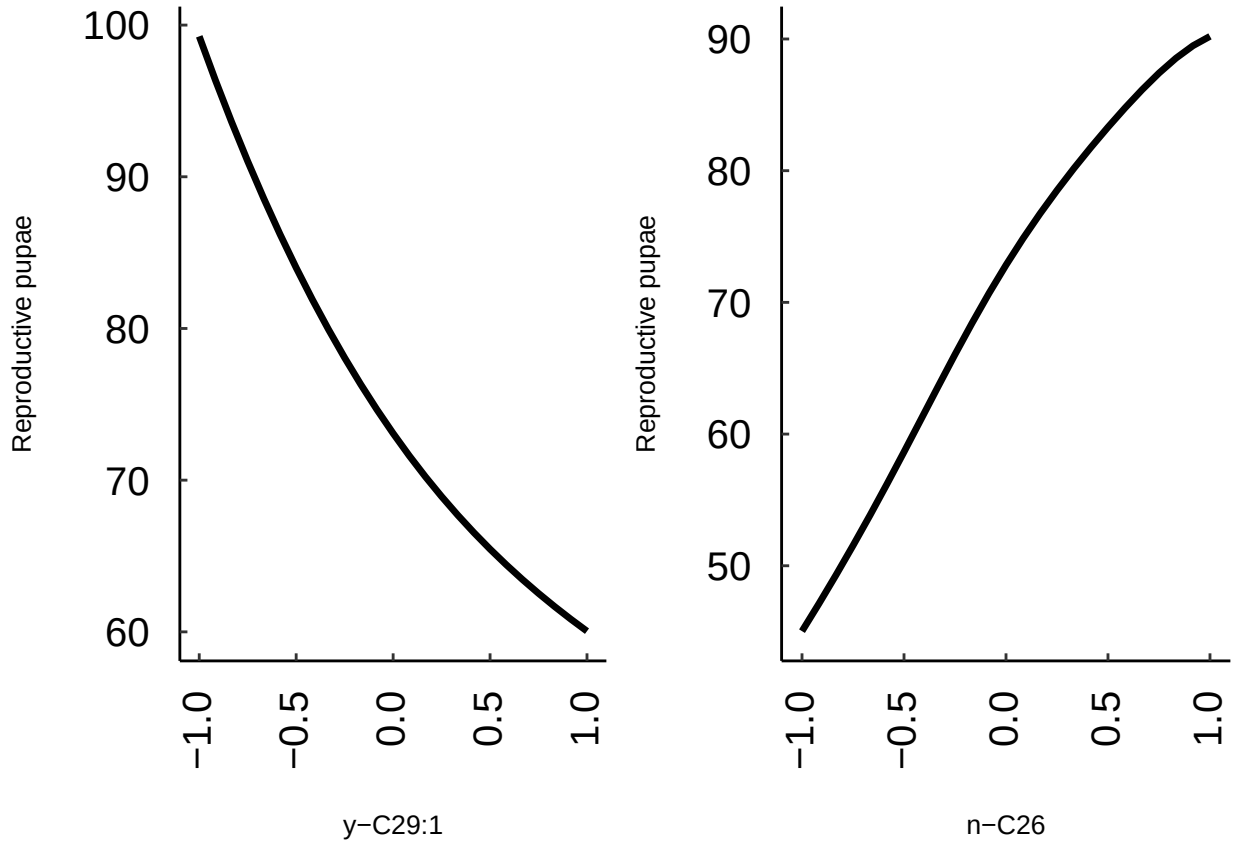


Plot (Figure 3b-c)

Using the fitness.landscape() function included in the gsg package, we provide a visual representation of the magnitude of selection for two CHCs: y-$C_{29:1}$ and n-C_{26}.

```
library(grid)
library(gridExtra)
Data=read.table("CHC - Selection.txt",header=T)
Data$z1<-as.numeric(Data$Peak6) #Change for each peak
Data$z1<-(Data$z1-mean(Data$z1))/sd(Data$z1)
model.peak6<-gam(SP~s(z1),family="tw",data=Data)
fl<-fitness.landscape(mod=model.peak6,phenotype=c("z1"),PI.method="n",
                      parallel = "no", ncpus = 1, refit.smooth = T)
x<-fl$points[,1]
y<-fl$Wbar
Data2<-data.frame(x,y)
C26=ggplot(data=Data2, aes(x=x, y=y)) +
```

```
  geom_line(size=1.2)+
  xlab("n-C26")+
  ylab("Reproductive pupae")+
  theme(text = element_text())+
  theme_bw()+
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        axis.ticks.length=unit(-0.10,"cm"),
        axis.text.x = element_text(size=15,color="black", angle=90,vjust=0.4,hjust=1,
                                   margin=unit(c(0.5,0.5,0.5,0.5), "cm")),
        axis.text.y = element_text(size=15,color="black",
                                   margin=unit(c(0.5,0.5,0.5,0.5), "cm")),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))+
  theme(text = element_text(size = 10))
Data$z2<-as.numeric(Data$Peak20) #Change for each peak
Data$z2<-(Data$z2-mean(Data$z2))/sd(Data$z2)
model.peak20<-gam(SP~s(z2),family="tw",data=Data)
fl2<-fitness.landscape(mod=model.peak20,phenotype=c("z2"),PI.method="n",
                       parallel = "no", ncpus = 1, refit.smooth = T)
m<-fl2$points[,1]
n<-fl2$Wbar
Data3<-data.frame(m,n)
yC29=ggplot(data=Data3, aes(x=m, y=n)) +
  geom_line(size=1.2)+
  xlab("y-C29:1")+
  ylab("Reproductive pupae")+
  theme(text = element_text())+
  theme_bw()+
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        axis.ticks.length=unit(-0.10,"cm"),
        axis.text.x = element_text(size=15,color="black", angle=90,vjust=0.4,hjust=1,
                                   margin=unit(c(0.5,0.5,0.5,0.5), "cm")),
        axis.text.y = element_text(size=15,color="black",
                                   margin=unit(c(0.5,0.5,0.5,0.5), "cm")),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))+
  theme(text = element_text(size = 10))
grid.arrange(yC29,C26, ncol=2)
```



REFERENCES

1. Morrissey, M.B., and Sakrejda, K. (2013). Unification of regression-based methods for the analysis of natural selection. Evolution *67*, 2094–2100. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/evo.12077.

2. Wood, S.N. (2017). Generalized additive models: An introduction with r (Chapman; Hall/CRC).

3. Morrissey, M.B., and Sakrejda, K. (2014). Gsg: Calculation of selection coefficients. R package version *2*.

4) Correlations between CHCs and collective behavior

We ran Spearman's rank-order correlations to evaluate the strenght and direction of association between CHCs and collective behavior. We ran a model between each CHC and each of the five collective behavior (160 models in total). P-values were adjusted using the false discovery rate (FDR) method.
NB *Because each colony replicate has one behavioral record, but repeated CHC measures, we used the mean CHC value for each colony replicate*

Load packages

```
library(plyr)
library(dplyr)
library(ggplot2)
```

The version used for each package is reported below.

```
## ggplot2   dplyr    plyr
## "3.0.0" "0.7.6" "1.8.4"
```

Load dataset

```
Data=read.table("CHC - Heritability.txt",header=T)
```

Average CHC peak value per colony replicate

```
DATA.AVER=aggregate(Data[,c(6:42)],by=list(Data$Colony), mean)
```

Spearman's rank order correlations

The following code lines create a table of results including Spearman correlation coefficients (estimate), test statistic (statistic), p-values and FDR-adjusted p-values (p.adjusted).

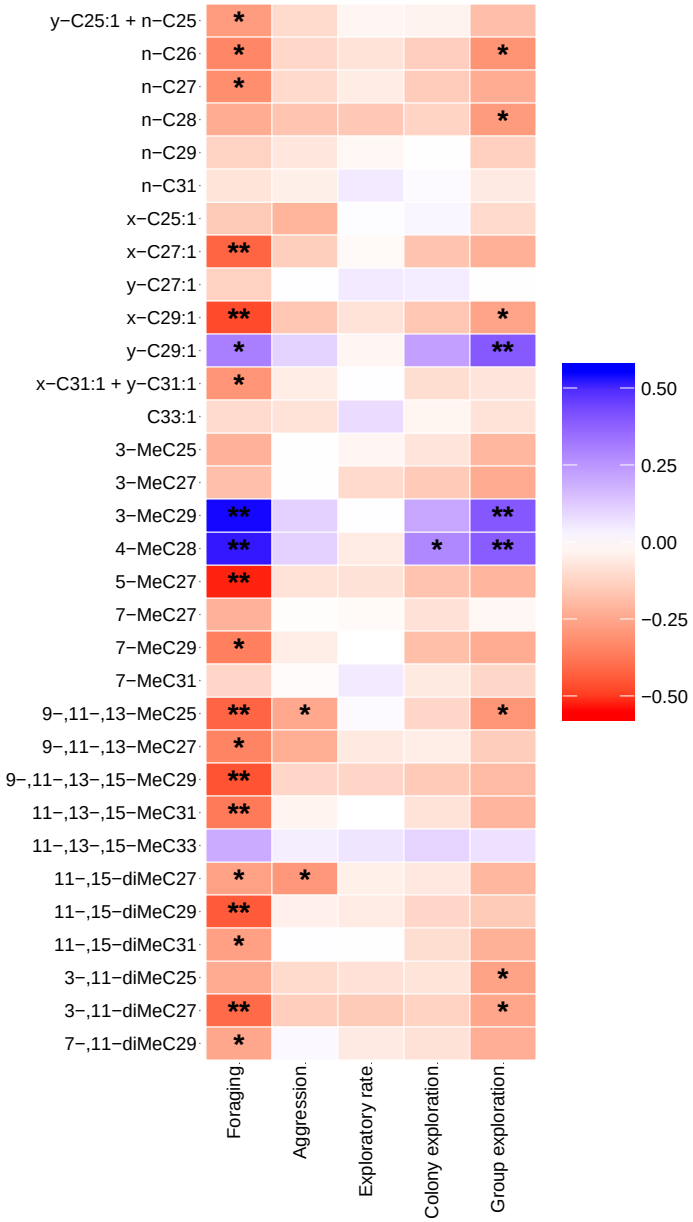
```
CHCs<-DATA.AVER[,2:33]
Behaviors=DATA.AVER[,34:38]
Results <- NULL
for(i in 1:ncol(CHCs)){
  data3<-NULL
  data2<- NULL
  for(j in 1:length(Behaviors)){
    Peak <- colnames(CHCs[i])
    Behavior <- colnames(Behaviors[j])
    Correlation <- cor.test(CHCs[[i]], Behaviors[[j]],
                            method="spearman")
    estimate<-Correlation$estimate[1]
    statistic<-Correlation$statistic[1]
    p.value <-Correlation$p.value[1]
    data2 <- data.frame(Peak, Behavior,estimate,
                        statistic,p.value)
    data3 <- rbind(data3, data2)
  }
  Results <- as.data.frame(rbind(Results, data3))
  rownames(Results) <- NULL
}
p.adjusted <-round(p.adjust(Results$p.value,
                            method = "fdr"),8)
Results <-data.frame(Results,p.adjusted)
```

```
rm(data2,data3)
Results$Peak=revalue(Results$Peak,
                     c("Peak1"="x-C25:1", "Peak2.3"="y-C25:1 + n-C25",
                       "Peak4"="9-,11-,13-MeC25",
                       "Peak5"="3-MeC25",
                       "Peak6"="n-C26", "Peak7"="3-,11-diMeC25",
                       "Peak8"="x-C27:1","Peak9"= "y-C27:1",
                       "Peak10"="n-C27","Peak11"= "9-,11-,13-MeC27",
                       "Peak12"="7-MeC27","Peak13"= "5-MeC27",
                       "Peak14"="11-,15-diMeC27","Peak15"= "3-MeC27",
                       "Peak16"="n-C28","Peak17"= "3-,11-diMeC27",
                       "Peak18"="4-MeC28","Peak19"= "x-C29:1",
                       "Peak20"="y-C29:1","Peak21"= "n-C29",
                       "Peak22"="9-,11-,13-,15-MeC29","Peak23"= "7-MeC29",
                       "Peak24"="11-,15-diMeC29","Peak25"= "7-,11-diMeC29",
                       "Peak26" = "3-MeC29", "Peak27.28"= "x-C31:1 + y-C31:1",
                       "Peak29"= "n-C31",
                       "Peak30"="11-,13-,15-MeC31","Peak31"= "7-MeC31",
                       "Peak32"="11-,15-diMeC31","Peak33"= "C33:1",
                       "Peak34"="11-,13-,15-MeC33"))
Results$Peak <- factor(Results$Peak,levels =
                         c("y-C25:1 + n-C25","n-C26","n-C27",
                           "n-C28","n-C29","n-C31","x-C25:1",
                           "x-C27:1", "y-C27:1","x-C29:1",
                           "y-C29:1","x-C31:1 + y-C31:1","C33:1",
                           "3-MeC25","3-MeC27","3-MeC29","4-MeC28",
                           "5-MeC27","7-MeC27","7-MeC29","7-MeC31",
                           "9-,11-,13-MeC25","9-,11-,13-MeC27",
                           "9-,11-,13-,15-MeC29","11-,13-,15-MeC31",
                           "11-,13-,15-MeC33","11-,15-diMeC27",
                           "11-,15-diMeC29","11-,15-diMeC31","3-,11-diMeC25",
                           "3-,11-diMeC27", "7-,11-diMeC29"))
Results$Peak = with(Results, factor(Peak, levels = rev(levels(Peak))))
```

Plot (Supplementary Figure 3)

```
ggplot(data = Results, aes(Behavior, Peak, fill = estimate))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
                       midpoint = 0, limit = c(-0.55,0.55), space = "Lab",
                       name="")+
  theme_bw()+
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(size=15,color="black",angle=90,hjust=1,vjust=0.1),
        axis.text.y = element_text(size=15,color="black"),
        axis.ticks.length = unit(0,"cm"))+
  coord_fixed(ratio=0.5)+
  xlab("")+
  ylab("")+
  theme(legend.key.size = unit(0.8, "in"))+
  theme(legend.text=element_text(size=15))+
```

```
  scale_x_discrete(labels=c("Aggression" ="Aggression",
                            "Exploratory.rate"="Exploratory rate",
                            "Group.exploration" = "Group exploration",
                            "Colony.exploration"="Colony exploration",
                            "Foraging"="Foraging"))+
  annotate("text", x = 1, y = 1, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 2, label = "**", size=8,fontface="bold")+
  annotate("text", x = 5, y = 2, label = "*", size=8,fontface="bold")+
  annotate("text", x = 5, y = 3, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 4, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 5, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 6, label = "*", size=8,fontface="bold")+
  annotate("text", x = 2, y = 6, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 8, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 9, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 10, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 11, label = "**", size=8,fontface="bold")+
  annotate("text", x = 2, y = 11, label = "*", size=8,fontface="bold")+
  annotate("text", x = 5, y = 11, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 13, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 15, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 16, label = "**", size=8,fontface="bold")+
  annotate("text", x = 4, y = 16, label = "*", size=8,fontface="bold")+
  annotate("text", x = 5, y = 16, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 17, label = "**", size=8,fontface="bold")+
  annotate("text", x = 5, y = 17, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 21, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 22, label = "*", size=8,fontface="bold")+
  annotate("text", x = 5, y = 22, label = "**", size=8,fontface="bold")+
  annotate("text", x = 1, y = 23, label = "**", size=8,fontface="bold")+
  annotate("text", x = 5, y = 23, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 25, label = "**", size=8,fontface="bold")+
  annotate("text", x = 5, y = 29, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 30, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 31, label = "*", size=8,fontface="bold")+
  annotate("text", x = 5, y = 31, label = "*", size=8,fontface="bold")+
  annotate("text", x = 1, y = 32, label = "*", size=8,fontface="bold")
```


y−C25:1 + n−C25
n−C26
n−C27
n−C28
n−C29
n−C31
x−C25:1
x−C27:1
y−C27:1
x−C29:1
y−C29:1
x−C31:1 + y−C31:1
C33:1
3−MeC25
3−MeC27
3−MeC29
4−MeC28
5−MeC27
7−MeC27
7−MeC29
7−MeC31
9−,11−,13−MeC25
9−,11−,13−MeC27
9−,11−,13−,15−MeC29
11−,13−,15−MeC31
11−,13−,15−MeC33
11−,15−diMeC27
11−,15−diMeC29
11−,15−diMeC31
3−,11−diMeC25
3−,11−diMeC27
7−,11−diMeC29
Foraging
Aggression
Exploratory rate
Colony exploration
Group exploration
0.50
0.25
0.00
−0.25
−0.50

5) Random forest classification analysis

We used a random forest classification analysis [1] to assess whether colony genotypes could be discriminated on the basis of their chemical profiles and, eventually, which compounds are responsible for the discrimination. Random Forest (RF) models are increasingly used in chemical ecology studies of social insects because they deal well with multivariate datasets often characterized by large number of predictors relative to the number of samples (e.g., hydrocarbon profiles; see [2–4]). In addition, RF models do not require transformation of constrained data and deal well with multicollinearity, non-normal distribution and non-linear relationship of predictors, which often characterize hydrocarbon datasets and can constitute limitations for the use of other multivariate analyses such as Principal Component Analysis and Discriminant Analysis [5,6].
We used the R package "randomForest" [7] for the RF genotype classification analysis. Log-ratio transformed peak areas were used as predictors in the model.
NB *RF does not take into account pedigree information. Hence, the following analysis is a simple attempt to identify CHCs that are highly variable across genotypes in our dataset, but that are not necessarily inherited*

Load packages

```
library(plyr)
library(dplyr)
library(randomForest)
library(ggplot2)
```

Packages (and dependencies) version is indicated below.

```
##      ggplot2 randomForest        dplyr         plyr
##      "3.0.0"     "4.6-14"      "0.7.6"      "1.8.4"
```

Load datasets

```
CHC=read.table("CHC - Heritability.txt",header=T)
CHC=CHC[,c(1,6:37)]#keep only variables needed
CHC=droplevels(CHC)
table(CHC$Genotype)#show number of samples per genotype
```

```
##
##   4013   4071   4085   5021     64   H301   H318   H342 H5.100 H5.106
##      9      8      7      8      3      7      6      6      5      3
## H5.122 H5.137 H5.150 H5.153 H5.154 H5.158 H5.159 H5.163 H5.164 H5.165
##      7      7      5      3      3      6      6      5      4      9
##   H505   H518   H519   H520   H521   H524   H532   H534   H542   H550
##      6      2      5      3      3      7      2     13      2      9
##   H553   H555   H563   H566   H571   H577   H578   H583   H584   H586
##      4      5      4      3      9      4      5      6      3      3
##   H588   H589   H590   H591   H593   H594   H597   H598
##      2      7      2      5      5      6      7      8
```

Make a vector for stratified sampling

As the number of CHC samples per genotype was unbalanced, we performed a stratified sampling by specifying the number of samples to be drawn from each genotype. This ensured that at least one sample from each genotype was drawn to build each tree in the RF. The number of samples to draw from each genotype was chosen in an attempt to retain the same relative proportions between classes, and to obtain a "training" dataset representing ~70% of the whole dataset. The undrawn samples (~30%) constitutes the out-of-bag (OOB) set of observations, equivalent to a "test" dataset, that are not used to build the RF tree.
NB *Each decision tree in the RF is built using a different set of drawn and undrawn samples. By running*

thousands of trees, we ensure that each sample in the dataset will be used as training or test sample multiple times.

```
vectorsize=c(6,6,5,5,2,5,4,4,4,2,5,5,4,2,2,4,4,
             4,3,6,4,1,4,2,2,5,1,9,1,6,3,4,3,2,
             6,3,4,4,2,2,1,5,1,4,4,4,5,6)
sum(vectorsize)#number of drawn samples
```

```
## [1] 180
```

```
180/257# total number of drawn samples is ~70% of the dataset
```

```
## [1] 0.7003891
```

Random forest model

RF analysis with replacement was ran using 100000 trees. Five randomly selected CHCs were used to build each node of the tree (denoted by *mtry*). This number is the recommended default value (the square root of the number of predictor variables, which is 5 in our case; see [8]) and provided the best classification accuracy.

```
CHC.rf=randomForest(Genotype ~ .,data=CHC,ntree=100000,
                    sampsize=vectorsize, replace=T,mtry=5,
                    proximity=T,importance=T)
```

Assess overall error rate

The unselected samples in a given bootstrap iteration are used to generate the OOB error for each genotype (i.e. the classification error obtained when the OOB samples are examined). Here we show the overall OOB rate of the model (The OOB error rate for each genotype can be seen in Supplementary table 2)

```
print(tail(CHC.rf$err.rate,1)[1]*100)
```

```
## [1] 17.12062
```

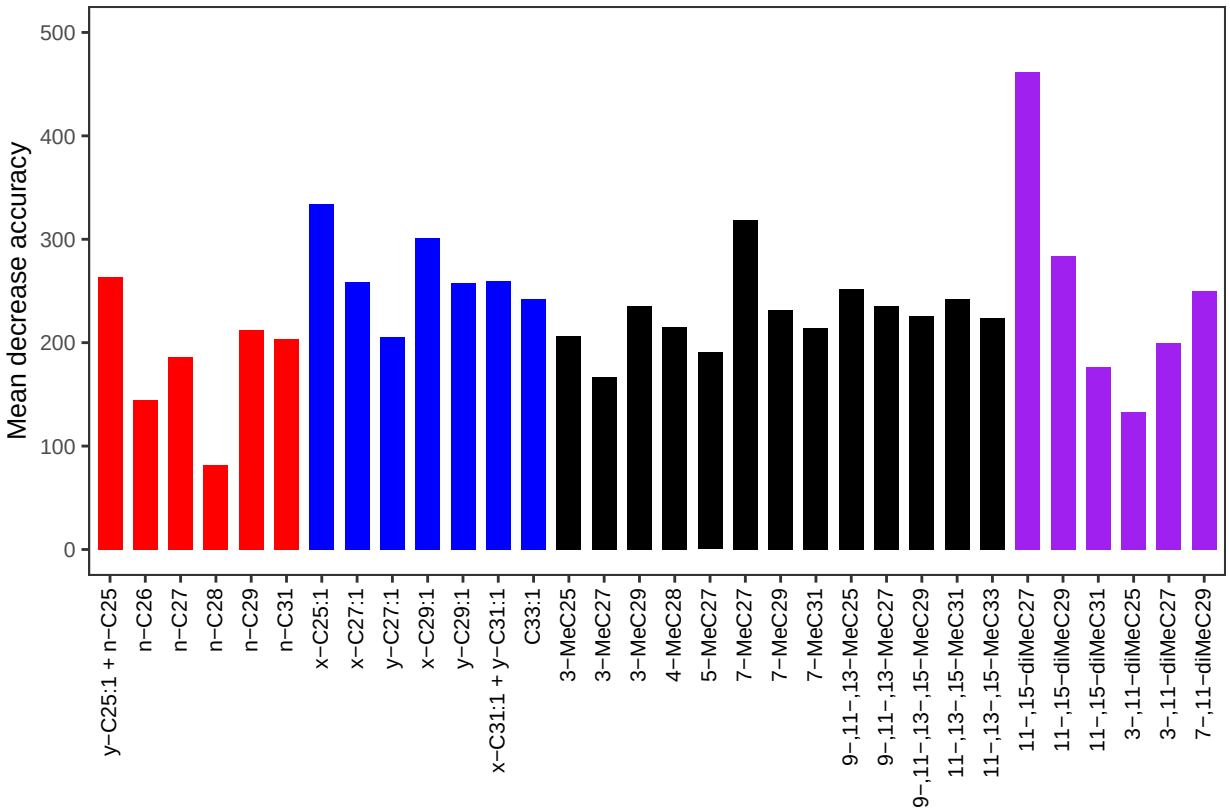
Produce the variable importance plot (Supplementary figure 4)

We used the mean decrease in accuracy (MDA), as suggested by Cutler et al. [8] to interpret hydrocarbons importance in classifying the genotypes. The MDA for a variable is the normalized difference of the classification accuracy for the out-of-bag data when the data for that variable is included as observed, and the classification accuracy for the out-of-bag data when the values of the variable in the out-of-bag data have been randomly permuted. **Higher values of mean decrease in accuracy indicate variables that are more important to the classification**
NB*The first peak in the plot should be blue/red as it is a mix of two compounds belonging to different CHC classes (linear alkanes and alkenes)*

```
imp <- varImpPlot(CHC.rf, n.var=32)
```

```
imp <- as.data.frame(imp)
imp$varnames <- rownames(imp) # row names to column
rownames(imp) <- NULL
imp$varnames=revalue(imp$varnames, c("Peak1"="x-C25:1", "Peak2.3"="y-C25:1 + n-C25",
                                     "Peak4"="9-,11-,13-MeC25",
                                     "Peak5"="3-MeC25",
                                     "Peak6"="n-C26", "Peak7"="3-,11-diMeC25",
                                     "Peak8"="x-C27:1","Peak9"= "y-C27:1",
                                     "Peak10"="n-C27","Peak11"= "9-,11-,13-MeC27",
                                     "Peak12"="7-MeC27","Peak13"= "5-MeC27",
                                     "Peak14"="11-,15-diMeC27","Peak15"= "3-MeC27",
```

```
                                    "Peak16"="n-C28","Peak17"= "3-,11-diMeC27",
                                    "Peak18"="4-MeC28","Peak19"= "x-C29:1",
                                    "Peak20"="y-C29:1","Peak21"= "n-C29",
                                    "Peak22"="9-,11-,13-,15-MeC29","Peak23"= "7-MeC29",
                                    "Peak24"="11-,15-diMeC29","Peak25"= "7-,11-diMeC29",
                                    "Peak26" = "3-MeC29", "Peak27.28"= "x-C31:1 + y-C31:1",
                                    "Peak29"= "n-C31",
                                    "Peak30"="11-,13-,15-MeC31","Peak31"= "7-MeC31",
                                    "Peak32"="11-,15-diMeC31","Peak33"= "C33:1",
                                    "Peak34"="11-,13-,15-MeC33"))
imp$varnames <- factor(imp$varnames,levels =
                         c("y-C25:1 + n-C25","n-C26","n-C27",
                           "n-C28","n-C29","n-C31","x-C25:1",
                           "x-C27:1", "y-C27:1","x-C29:1",
                           "y-C29:1","x-C31:1 + y-C31:1","C33:1",
                           "3-MeC25","3-MeC27","3-MeC29","4-MeC28",
                           "5-MeC27","7-MeC27","7-MeC29","7-MeC31",
                           "9-,11-,13-MeC25","9-,11-,13-MeC27",
                           "9-,11-,13-,15-MeC29","11-,13-,15-MeC31",
                           "11-,13-,15-MeC33","11-,15-diMeC27",
                           "11-,15-diMeC29","11-,15-diMeC31","3-,11-diMeC25",
                           "3-,11-diMeC27", "7-,11-diMeC29"))
ggplot(imp, aes(varnames, MeanDecreaseAccuracy,fill=varnames)) +
  geom_bar(stat="identity", width=0.7) +
  scale_y_continuous(name="Mean decrease accuracy", limits=c(0, 500))+
  scale_fill_manual("varnames", breaks=c(1:32),
                    values=c(rep("red",6),rep("blue",7),
                             rep("black",13), rep("purple",6)))+
  xlab("")+
  ylab("Heritability")+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        axis.text.x = element_text(color="black",
                                   angle=90,hjust=1,vjust=0.4),
        panel.grid.minor = element_blank())+
  theme(text = element_text(size = 10))
```



REFERENCES

1. Breiman, L. (2001). Random forests. Machine Learning *45*, 5–32. Available at: https://link.springer.com/content/pdf/10.1023%2FA%3A1010933404324.pdf.

2. Jansen, J., Pokorny, T., and Schmitt, T. (2016). Disentangling the effect of insemination and ovary development on the cuticular hydrocarbon profile in the bumblebee bombus terrestris (hymenoptera: Apidae). Apidologie *47*, 101–113. Available at: https://link.springer.com/content/pdf/10.1007%2Fs13592-015-0379-5.pdf.

3. Loope, K.J., Millar, J.G., and Wilson Rankin, E.E. (2018). Weak nestmate discrimination behavior in native and invasive populations of a yellowjacket wasp (vespula pensylvanica). Biological Invasions *20*, 3431–3444. Available at: https://link.springer.com/content/pdf/10.1007%2Fs10530-018-1783-3.pdf.

4. Monnin, T., Helft, F., Leroy, C., d'Ettorre, P., and Doums, C. (2018). Chemical characterization of young virgin queens and mated egg-laying queens in the ant cataglyphis cursor: Random forest classification analysis for multivariate datasets. Journal of Chemical Ecology *44*, 127–136. Available at: https://link.springer.com/content/pdf/10.1007%2Fs10886-018-0923-7.pdf.

5. Martin, S.J., and Drijfhout, F.P. (2009). How reliable is the analysis of complex cuticular hydrocarbon profiles by multivariate statistical methods? Journal of Chemical Ecology *35*, 375–382. Available at: https://link.springer.com/content/pdf/10.1007%2Fs10886-009-9610-z.pdf.

6. Ranganathan, Y., and Borges, R.M. (2011). To transform or not to transform. Plant Signaling & Behavior *6*, 113–116. Available at: https://www.tandfonline.com/doi/pdf/10.4161/psb.6.1.14191?needAccess=true.

7. Liaw, A., and Wiener, M. (2002). Classification and regression by randomForest. R news *2*, 18–22.

8. Cutler, D.R., Edwards Jr., T.C., Beard, K.H., Cutler, A., Hess, K.T., Gibson, J., and Lawler, J.J. (2007).

Random forests for classification in ecology. Ecology *88*, 2783–2792. Available at: https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.1890/07-0539.1.