dbRDA

[redacted]

2024-04-10

1. Packages to install

```
library(codep)
library(adespatial)
library(adegraphics)
library(vegan)
library(ape)
library(car)
library(adegenet)
library(dplyr)
library(pegas)
library(caret)
library(poppr)
library(tseries)
library(fmsb)
library(marmap)
library(reshape2)
library(shape)
library(tidyverse)
library(ggmap)
library(ggplot2)
library(ade4)
library(glmnet)
library(foreach)
```

2. Function to select variables according to VIF

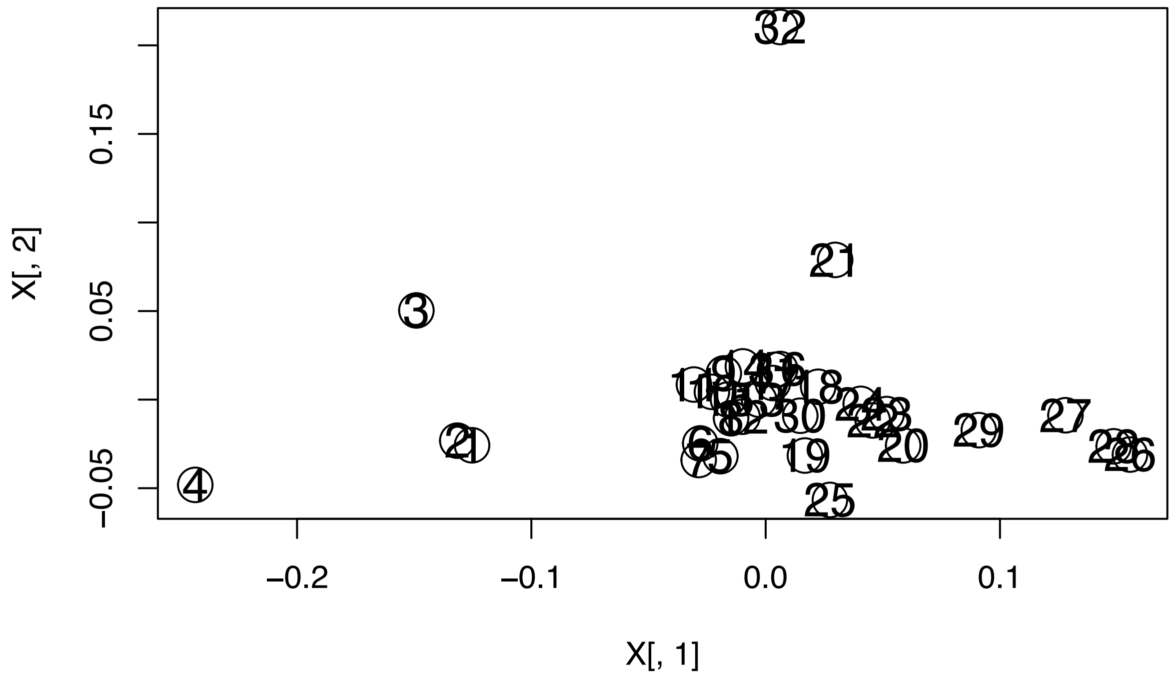
```
vif_func<-function(in_frame,thresh=100,trace=T,...){

  if(any(!'data.frame' %in% class(in_frame))) in_frame<-data.frame(in_frame)

  #get initial vif value for all comparisons of variables
  vif_init<-NULL
  var_names <- names(in_frame)
  for(val in var_names){
    regressors <- var_names[-which(var_names == val)]
    form <- paste(regressors, collapse = '+')
    form_in <- formula(paste(val, '~', form))
    vif_init<-rbind(vif_init, c(val, VIF(lm(form_in, data = in_frame, ...))))
  }
  vif_max<-max(as.numeric(vif_init[,2]), na.rm = TRUE)
```

```r
  if(vif_max < thresh){
    if(trace==T){ #print output of each iteration
      prmatrix(vif_init,collab=c('var','vif'),rowlab=rep('',nrow(vif_init)),quote=F)
      cat('\n')
      cat(paste('All variables have VIF < ', thresh,', max VIF ',round(vif_max,2), sep=''),'\n\n')
    }
    return(var_names)
  } else{

    in_dat<-in_frame

    #backwards selection of explanatory variables, stops when all VIF values are below 'thresh'
    while(vif_max >= thresh){

      vif_vals<-NULL
      var_names <- names(in_dat)

      for(val in var_names){
        regressors <- var_names[-which(var_names == val)]
        form <- paste(regressors, collapse = '+')
        form_in <- formula(paste(val, '~', form))
        vif_add<-VIF(lm(form_in, data = in_dat, ...))
        vif_vals<-rbind(vif_vals,c(val,vif_add))
      }
      max_row<-which(vif_vals[,2] == max(as.numeric(vif_vals[,2]), na.rm = TRUE))[1]

      vif_max<-as.numeric(vif_vals[max_row,2])

      if(vif_max<thresh) break

      if(trace==T){ #print output of each iteration
        prmatrix(vif_vals,collab=c('var','vif'),rowlab=rep('',nrow(vif_vals)),quote=F)
        cat('\n')
        cat('removed: ',vif_vals[max_row,1],vif_max,'\n\n')
        flush.console()
      }

      in_dat<-in_dat[,!names(in_dat) %in% vif_vals[max_row,1]]

    }

    return(names(in_dat))

  }
}
```

This function comes from (checked on the 10.04.2024): https://rdrr.io/github/walterxie/ComMA/src/R/vif_function.R We have chosen a threshold value of 10. The output indicates the VIF values for each variable after each stepwise comparison. The function calculates the VIF values for all explanatory variables, removes the variable with the highest value, and repeats until all VIF values are below the threshold. The final output is a list of variable names with VIF values that fall below the threshold.

3. Import Fst matrix - PcoA

```
Fst <- read.matrix(file = "32sites_pairwise_FST.txt",header = FALSE, sep ='\t')
nba_matrix <- data.matrix(Fst, rownames.force = TRUE)
FST_dist <- as.dist(as(nba_matrix, "matrix"))

Pcoa=ape::pcoa(FST_dist)
X=Pcoa$vectors
plot(X[,1], X[,2], pch=21, cex=2.4); text(X, col="black", cex=1.5) # populations
```



The coordinate of each population in the PcoA will correspond to the response variable

4. Explanatory data

Each row should be the data associated to one population, each column is one of the explanatory variable (SST, Sampling year, dbMEM1, dbMEM2. . . , AEM1, AEM2,. . .). Choose the colors corresponding to each population (here we use 8 colors, corresponding to the number of genetic clusters). For the partial db-RDA, one would want to partition the variance by taking into account the dbMEM only, the AEM only, and the other variables such as sampling year, temperatures, habitat discontinuity. When doing the global db-rda, we would incorporate every variables. We will do so in this code (the first 6 columns of the dataset are not included as they contain information regarding the sampling position, island/coastline, in which genetic cluster they are part of etc.

In our case, the number of variables is too important (40) compared to the number of populations (32). It is therefore necessary to use a method that will allow us to consider that some of the variables are correlated and therefore some variables can be discarded. This is done by doing an elastic net linear regression (see: https://www.r-bloggers.com/2017/09/variable-selection-with-elastic-net). We will take an alpha = 0.08.

5. Elastic net linear regression

```
cv1 <- cv.glmnet(Y, X[,1:2], family = "mgaussian", nfold = 10, type.measure = "deviance", paralle = TRU
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
md1 <- glmnet(Y, X[,1:2], family = "mgaussian", lambda = cv1$lambda.1se, alpha = 0.04)
```

Given the results of the elastic net linear regression, we can discard some of the variables.

```
selected_variables <- subset(variables, select = c(Year,Temp_10y_mean,Temp_10y_min,dbMEM1,dbMEM2,dbMEM3
                                                   AEM1,AEM2,AEM3,AEM4,AEM6,AEM7,AEM8,AEM9,AEM10,AEM12,
                                                   AEM17,AEM18,AEM19,AEM20,AEM21,AEM23,AEM24,AEM26,AEM2

#vif_func(in_frame=selected_variables,thresh=11,trace=T)
selected_variables <- subset(selected_variables, select = -c(Temp_10y_min))
selected_variables <- subset(selected_variables, select = -c(dbMEM6))
selected_variables <- subset(selected_variables, select = -c(Temp_10y_mean))
selected_variables <- subset(selected_variables, select = -c(dbMEM2))
# All variables have VIF < 10, max VIF 9.67
```

We will then only keep the variables with a VIF < 10, using the vif_func described above, and with a threshold of 10. We repeat this until all the variables have a VIF below the threshold. Once we have this, we can start with the dbRDA.

6. dbRDA

```
rda1<- rda(X ~., selected_variables)
#summary(rda1)
resOrdi2step <- ordiR2step(rda(X~1, data=selected_variables), scope= formula(rda1), direction= "both",
```

```
## Step: R2.adj= 0
## Call: X ~ 1
##
##                    R2.adjusted
## <All variables>  6.683784e-01
## + dbMEM1         3.706725e-01
## + AEM7           5.729645e-02
## + AEM6           5.396788e-02
## + dbMEM3         4.882699e-02
## + AEM26          3.738134e-02
## + AEM9           2.886717e-02
## + AEM21          2.700166e-02
## + AEM18          2.539037e-02
## + AEM1           2.531489e-02
## + dbMEM5         2.365086e-02
## + Year           1.591814e-02
## + AEM2           4.170276e-03
## + AEM15          5.083676e-05
## <none>           0.000000e+00
## + AEM23         -3.615036e-03
## + AEM10         -8.424814e-03
## + AEM3          -1.141846e-02
## + AEM19         -1.246613e-02
## + AEM4          -1.463390e-02
## + AEM27         -1.517364e-02
## + AEM8          -1.841275e-02
## + AEM20         -2.204787e-02
## + AEM24         -2.204969e-02
## + AEM28         -2.279402e-02
```

```
## + AEM12         -2.523186e-02
## + AEM17         -2.782910e-02
## + dbMEM4        -2.914412e-02
## + AEM14         -3.206960e-02
##
##          Df     AIC      F   Pr(>F)
## + dbMEM1  1 -154.39 19.259 0.000999 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.3706725
## Call: X ~ dbMEM1
##
##                 R2.adjusted
## <All variables>   0.6683784
## + dbMEM3          0.4339650
## + dbMEM5          0.4079208
## + AEM26           0.4056288
## + AEM3            0.3982619
## + AEM9            0.3957667
## + AEM1            0.3890311
## + AEM21           0.3884624
## + Year            0.3838315
## + AEM23           0.3826910
## + AEM7            0.3754047
## + AEM2            0.3746340
## <none>            0.3706725
## + AEM15           0.3699212
## + AEM8            0.3675209
## + AEM10           0.3652213
## + AEM18           0.3651297
## + AEM27           0.3638850
## + AEM6            0.3614909
## + AEM28           0.3610805
## + AEM20           0.3602628
## + AEM24           0.3597508
## + AEM19           0.3593395
## + AEM4            0.3578616
## + AEM12           0.3574242
## + AEM17           0.3554658
## + dbMEM4          0.3533053
## + AEM14           0.3508892
##
##          Df     AIC      F   Pr(>F)
## + dbMEM3  1 -156.87 4.3545 0.005994 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.433965
## Call: X ~ dbMEM1 + dbMEM3
##
##                 R2.adjusted
## <All variables>   0.6683784
## + dbMEM5          0.4748040
```

```
## + AEM1            0.4711493
## + AEM26           0.4697004
## + AEM3            0.4605973
## + AEM9            0.4565597
## + AEM23           0.4470166
## + AEM7            0.4445239
## + AEM2            0.4411369
## + AEM21           0.4357945
## <none>            0.4339650
## + Year            0.4319672
## + AEM8            0.4319577
## + AEM6            0.4317863
## + AEM15           0.4310380
## + AEM10           0.4300798
## + AEM27           0.4292750
## + AEM20           0.4256674
## + AEM18           0.4253202
## + AEM4            0.4248799
## + AEM24           0.4248697
## + AEM19           0.4246807
## + AEM28           0.4219365
## + AEM12           0.4211073
## + AEM17           0.4201673
## + dbMEM4          0.4182379
## + AEM14           0.4173134
##
##          Df     AIC     F   Pr(>F)
## + dbMEM5  1 -158.39 3.255 0.006993 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.474804
## Call: X ~ dbMEM1 + dbMEM3 + dbMEM5
##
##                 R2.adjusted
## <All variables>   0.6683784
## + AEM1            0.5160555
## + AEM3            0.5036591
## + AEM26           0.5023344
## + AEM8            0.4872761
## + AEM7            0.4871604
## + AEM9            0.4863605
## + AEM2            0.4842115
## + AEM23           0.4810772
## + AEM21           0.4778174
## + AEM18           0.4763261
## + Year            0.4750668
## <none>            0.4748040
## + AEM6            0.4725451
## + AEM15           0.4724125
## + AEM27           0.4712811
## + AEM4            0.4709569
## + AEM10           0.4685750
## + AEM19           0.4668025
```

```
## + AEM24           0.4656110
## + AEM20           0.4651024
## + AEM28           0.4644922
## + AEM12           0.4618833
## + AEM17           0.4615942
## + dbMEM4          0.4600070
## + AEM14           0.4591472
##
##        Df     AIC      F  Pr(>F)
## + AEM1  1 -160.17 3.3867 0.00999 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5160555
## Call: X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1
##
##                 R2.adjusted
## <All variables>   0.6683784
## + AEM3            0.5477838
## + AEM26           0.5450634
## + AEM2            0.5369942
## + AEM8            0.5304698
## + AEM23           0.5249536
## + AEM9            0.5244284
## + AEM7            0.5236540
## + AEM18           0.5188616
## + Year            0.5175301
## <none>            0.5160555
## + AEM21           0.5158493
## + AEM4            0.5138576
## + AEM27           0.5138275
## + AEM28           0.5132601
## + AEM15           0.5121416
## + AEM10           0.5114986
## + AEM19           0.5082244
## + AEM24           0.5081605
## + AEM20           0.5073800
## + AEM6            0.5066776
## + AEM12           0.5057841
## + AEM17           0.5045319
## + dbMEM4          0.5025958
## + AEM14           0.5017377
##
##        Df     AIC      F  Pr(>F)
## + AEM3  1 -161.55 2.8944 0.01199 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5477838
## Call: X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3
##
##                 R2.adjusted
## <All variables>   0.6683784
## + AEM26           0.5799963
```

```
## + AEM2            0.5718881
## + AEM8            0.5645862
## + AEM23           0.5594394
## + AEM7            0.5566647
## + AEM9            0.5564988
## + Year            0.5533721
## + AEM21           0.5488801
## + AEM28           0.5488259
## + AEM18           0.5487841
## <none>            0.5477838
## + AEM27           0.5461694
## + AEM4            0.5453430
## + AEM15           0.5448994
## + AEM10           0.5437173
## + AEM24           0.5413643
## + AEM19           0.5411682
## + AEM6            0.5407375
## + AEM20           0.5401203
## + dbMEM4          0.5387029
## + AEM12           0.5381849
## + AEM17           0.5364459
## + AEM14           0.5339605
##
##          Df     AIC      F  Pr(>F)
## + AEM26  1 -163.17 2.9941 0.05095 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
resOrdi2step$anova
```

```
##                   R2.adj Df     AIC       F   Pr(>F)
## + dbMEM1         0.37067  1 -154.39 19.2589 0.000999 ***
## + dbMEM3         0.43397  1 -156.87  4.3545 0.005994 **
## + dbMEM5         0.47480  1 -158.39  3.2550 0.006993 **
## + AEM1           0.51606  1 -160.17  3.3867 0.009990 **
## + AEM3           0.54778  1 -161.55  2.8944 0.011988 *
## <All variables> 0.66838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
selected_variables1 <- subset(selected_variables, select = c(dbMEM1,dbMEM3,dbMEM5,AEM1, AEM3, AEM26, AE
rda2<- rda(X ~., selected_variables1)
#summary(rda2)
RsquareAdj(rda2)
```

```
## $r.squared
## [1] 0.719777
##
## $adj.r.squared
## [1] 0.6223081
```

```
anova(rda2, perm=999)  # default is permutation=999
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
```

```
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3 + AEM26 + AEM2 + AEM8, data = selecte
##          Df  Variance      F Pr(>F)
## Model     8 0.0086438 7.3847  0.001 ***
## Residual 23 0.0033652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(rda2, by="axis", perm.max=999) # test axes for significance
```

```
## Permutation test for rda under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3 + AEM26 + AEM2 + AEM8, data = selecte
##          Df  Variance       F Pr(>F)
## RDA1      1 0.0060317 41.2246  0.001 ***
## RDA2      1 0.0014610  9.9857  0.003 **
## RDA3      1 0.0005065  3.4617  0.143
## RDA4      1 0.0004101  2.8030  0.284
## RDA5      1 0.0001222  0.8353  0.993
## RDA6      1 0.0000667  0.4561  1.000
## RDA7      1 0.0000415  0.2834  1.000
## RDA8      1 0.0000040  0.0277  1.000
## Residual 23 0.0033652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(rda2, by="terms", permu=999) # test for sign. environ. variables
```

```
## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3 + AEM26 + AEM2 + AEM8, data = selecte
##          Df  Variance       F Pr(>F)
## dbMEM1    1 0.0046952 32.0901  0.001 ***
## dbMEM3    1 0.0009548  6.5260  0.003 **
## dbMEM5    1 0.0006622  4.5262  0.009 **
## AEM1      1 0.0006349  4.3395  0.009 **
## AEM3      1 0.0005070  3.4655  0.016 *
## AEM26     1 0.0004872  3.3295  0.044 *
## AEM2      1 0.0003875  2.6486  0.039 *
## AEM8      1 0.0003149  2.1521  0.113
## Residual 23 0.0033652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
selected_variables2 <- subset(selected_variables, select = c(dbMEM1,dbMEM3,dbMEM5,AEM1,AEM3))
rda3 <- rda(X ~., selected_variables2)
# summary(rda3)
#                         RDA1     RDA2      RDA3      RDA4      RDA5
# Eigenvalue            0.005334 0.001351 0.0003752 0.000336 5.841e-05
# Proportion Explained  0.715565 0.181192 0.0503367 0.045071 7.836e-03
```

```r
# Cumulative Proportion 0.715565 0.896756 0.9470929 0.992164 1.000e+00

RsquareAdj(rda3)
```

```
## $r.squared
## [1] 0.6207219
##
## $adj.r.squared
## [1] 0.5477838
```

```r
anova(rda3, perm=999)  # default is permutation=999
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3, data = selected_variables2)
##          Df  Variance      F Pr(>F)
## Model     5 0.0074542 8.5103  0.001 ***
## Residual 26 0.0045547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(rda3, by="axis", perm.max=999) # test axes for significance
```

```
## Permutation test for rda under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3, data = selected_variables2)
##          Df  Variance       F Pr(>F)
## RDA1      1 0.0053340 30.4482  0.001 ***
## RDA2      1 0.0013506  7.7099  0.006 **
## RDA3      1 0.0003752  2.1419  0.273
## RDA4      1 0.0003360  1.9178  0.222
## RDA5      1 0.0000584  0.3334  0.935
## Residual 26 0.0045547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

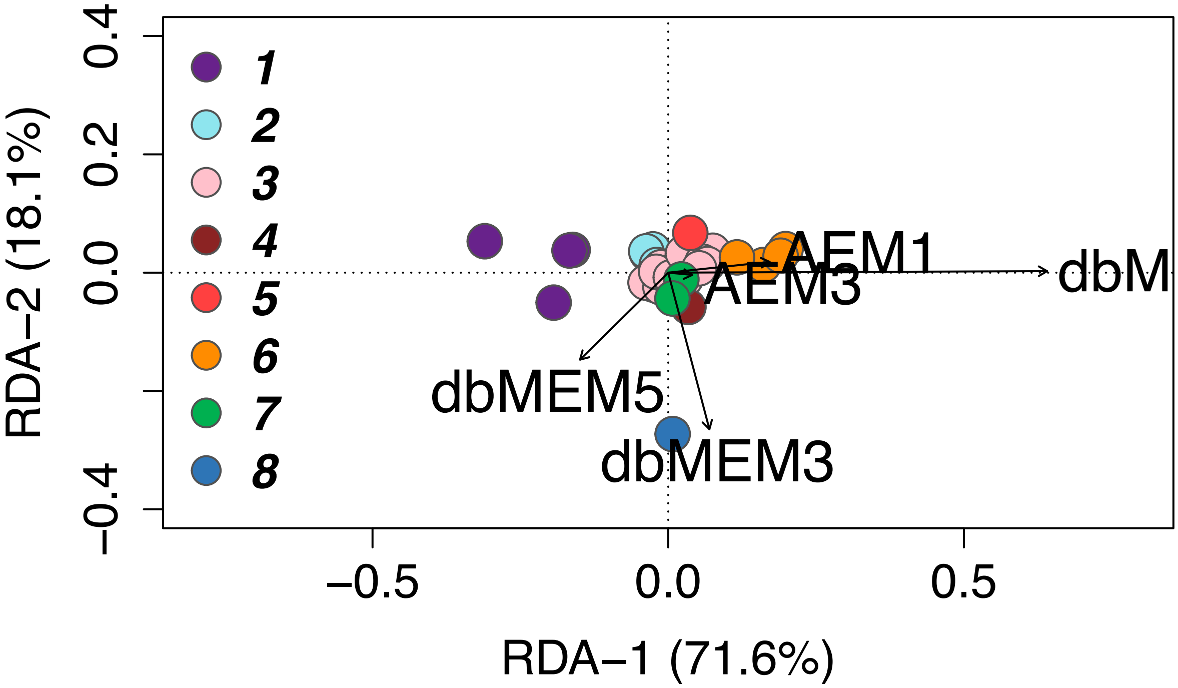
```r
anova(rda3, by="terms", permu=999) # test for sign. environ. variables
```

```
## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = X ~ dbMEM1 + dbMEM3 + dbMEM5 + AEM1 + AEM3, data = selected_variables2)
##          Df  Variance       F Pr(>F)
## dbMEM1    1 0.0046952 26.8017  0.001 ***
## dbMEM3    1 0.0009548  5.4505  0.006 **
## dbMEM5    1 0.0006622  3.7803  0.017 *
## AEM1      1 0.0006349  3.6243  0.022 *
## AEM3      1 0.0005070  2.8944  0.035 *
## Residual 26 0.0045547
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7. Plot it !

```
# db-RDA axes 1 & 2
plot(rda3, type="n", scaling=1, xlab=c("RDA-1 (71.6%)"), ylab=c("RDA-2 (18.1%)"), xlim=c(-0.4, 0.4), yl
```




```
#If one wants to put populations instead of the predictors: text(rda3, scaling=1, col="black", cex=2)

# If one wants to plot on 2nd and 3rd axis
# plot(rda3, type="n", choices = c(2, 3), xlab=c("RDA-2"), ylab=c("RDA-3"), xlim=c(-0.4, 0.4), ylim=c(-
```