GEA environment,selection and outliers - Melbourne

Bernd Gruber (based on code from Brenna Forrester)

2019-06-10

Contents

Foreword

This tutorial is based on code and comments from Brenna Forrester, Colorado State University.

The Wolf data set is from Schweizer et al. 2016. Genetic subdivision and candidate genes under selection in North American grey wolves Molecular Ecology 25, 380-402. Dryad doi:10.5061/dryad.c9b25.

Multivariate GEA: Redundancy Analysis

RDA is a multivariate ordination technique that analyzes matricies of loci and environmental predictors simultaneously. RDA determines how groups of loci covary in response to the multivariate environment.

RDA is a two-step analysis in which genetic and environmental data are analyzed using multivariate linear regression, producing a matrix of fitted values. Then PCA of the fitted values is used to produce canonical axes, which are linear combinations of the predictors.

More information on RDA & other multivariate GEAs (such as Random Forest) can be found in our paper: Forester BR, Lasky JR, Wagner HH, Urban DL (2018) Comparing methods for detecting multilocus adaptation with multivariate genotype-environment associations Molecular Ecology 27, 2215-2233.

RDA can be used on both individual and population-based sampling designs. The distinction between the two may not be straightforward in all cases. A simple guideline would be to use an individual-based framework when you have individual coordinates for most of your samples, and the resolution of your environmental data would allow for a sampling of environmental conditions across the site/study area. More on RDA with population level data in the notes below.

To run RDA on population level data, calculate allele frequencies (columns) for populations (rows). Applying a Hellinger transformation/standardization to the data will ensure it behaves well in the RDA, e.g. if our input of population allele frequencies was called “pop.data”:

```
pop.data.hellinger <- decostand(pop.data, method = "hellinger")  # square root of (allele frequency / popula
```

RDA does not require corrections for multiple tests because it analyzes the genomic and environmental data simultaneously!

I highly recommend the following book for more information on interpreting RDA output from vegan: Borcard et al. (2011) Numerical Ecology with R.

Load necessary libraries

```
library(psych)
library(GGally)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
library(robust)
```

```
## Loading required package: fit.models
```

```
library(vegan)
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-4
```

```
library(qvalue)  #bioconductor
```

Before we can start with our redundancy analysis we need to make sure, we select "suitable" environmental predictors. Meaning the predictors should have a "good" chance to have an effect on our response (the genetic structure) and should not be correlated between each other.

Please see Dormann et al. (2013) Ecography for a discussion of these issues. Generally, the |r| > 0.7 "rule of thumb" is a good guideline for removing correlated predictors. We will also check for multicollinearity below using Variance Inflation Factors.

We will use the Wolf data set mentioned above to run an examplary redundancy analysis. To convert a genlight object into the correct format simply use:

```
gen <- as.matrix(gl)
```

Load the genetic data set from a csv file:

```
## [1]    94 10000
```

```
##      chr9.22100598 chr2.19649684
## [1,]             0             0
## [2,]             1             0
## [3,]             0             2
## [4,]             0             0
## [5,]             1             1
##      chr38.14463749 chr13.20658362
## [1,]              0              0
## [2,]              0              2
## [3,]              0              1
## [4,]              0              2
## [5,]              1              0
```

```
##      chr23.16849044
## [1,]              0
## [2,]              1
## [3,]              0
## [4,]              1
## [5,]              0
```

```
# Read from github

fp <- "https://raw.githubusercontent.com/green-striped-gecko/dartRworkshop/master/data"

gen <- read.csv(file.path(fp, "Wolf_Gen_sample_6pops_94indiv.csv"))
dim(gen)  # 94 individuals in rows; 10,000 SNPs in columns
gen[1:5, 1:5]  # coded as 0/1/2
```

Load the environmental data set

```
##  [1] "individual"
##  [2] "ecotype"
##  [3] "long"
##  [4] "lat"
##  [5] "ann_mean_temp"
##  [6] "mean_diurnal_range"
##  [7] "temp_seasonality"
##  [8] "max_temp_warmest_month"
##  [9] "min_temp_coldest_month"
## [10] "ann_precip"
## [11] "precip_seasonality"
## [12] "precip_coldest_quarter"
## [13] "land_cover"
## [14] "ndvi"
## [15] "elev"
## [16] "percent_tree_cover"
```

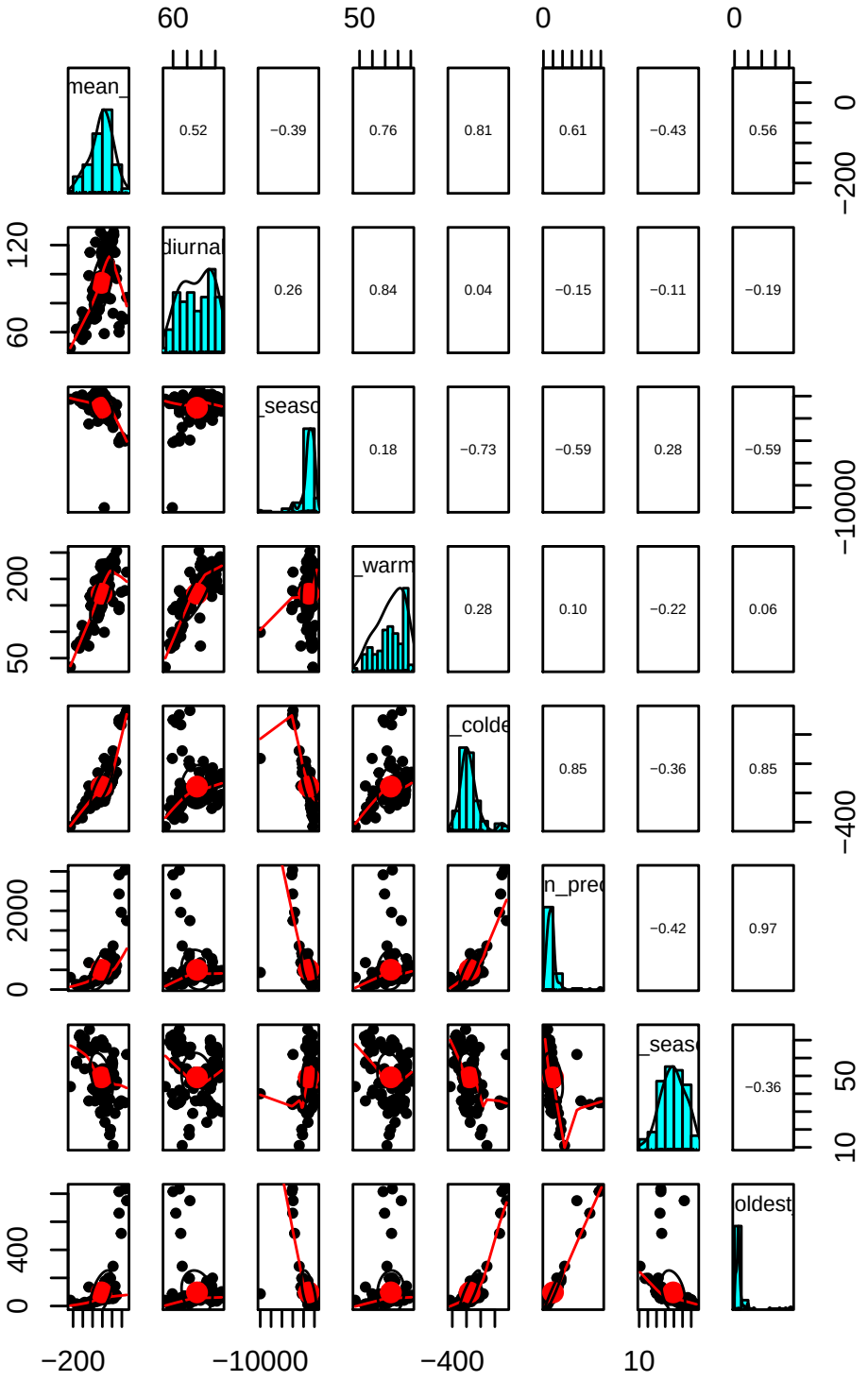
```
# Read from github

fp <- "https://raw.githubusercontent.com/green-striped-gecko/dartRworkshop/master/data"

env <- read.csv(file.path(fp, "Wolf_Env_6pops_94indiv.csv"))
names(env)
```

Or in case you are using a genlight object you should be able to use the @other$ind.metrics table from a genlight data set.

```
env <- @other$ind.metrics
```

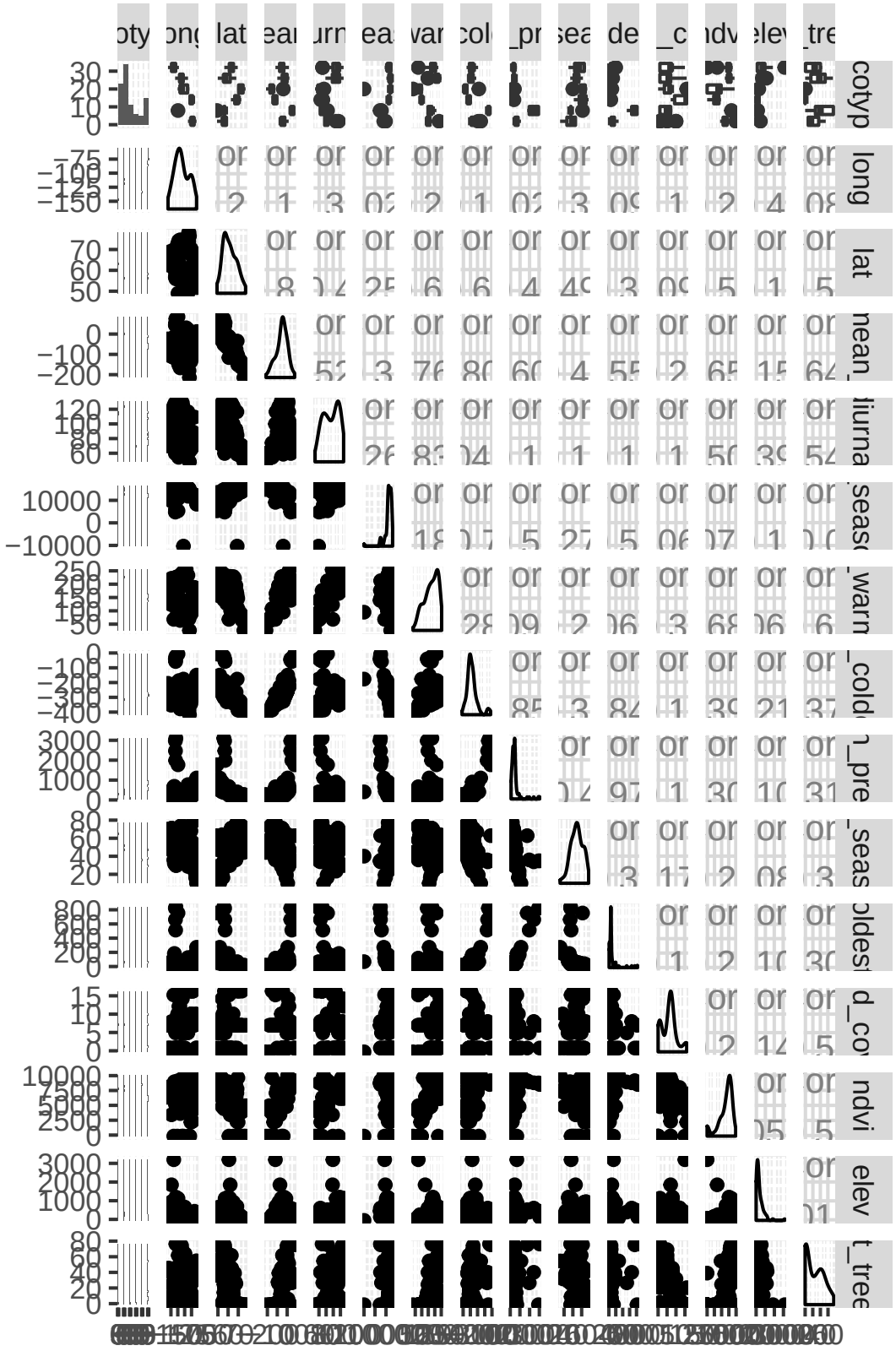
Check for multicollinearity between predictors (|r|>0.7)

```
pairs.panels(env[, 5:12])  #good for contineous predictors only
```


mean_
diurna
_seasc
_warm
_colde
n_prec
_seas
oldest
0.52
−0.39
0.76
0.81
0.61
−0.43
0.56
0.26
0.84
0.04
−0.15
−0.11
−0.19
0.18
−0.73
−0.59
0.28
−0.59
0.28
0.10
−0.22
0.06
0.85
−0.36
0.85
−0.42
0.97
−0.36
60
50
0
0
120
60
200
50
2000
0
400
0
−200
0
−10000
−400
50
10
−200
−10000
−400
10

```
ggpairs(env[, -1])  #allows also factors [except individual]
```

```
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick
## better value with `binwidth`.
```


cotyp
long
lat
mean_
diurna
seasc
warm
_cold
n_pre
_seas
oldest
d_co
ndvi
elev
t_tree

Next we look at landscape cover (a factor)

```
table(env$land_cover)  # let's look at land cover, which is a factor
```

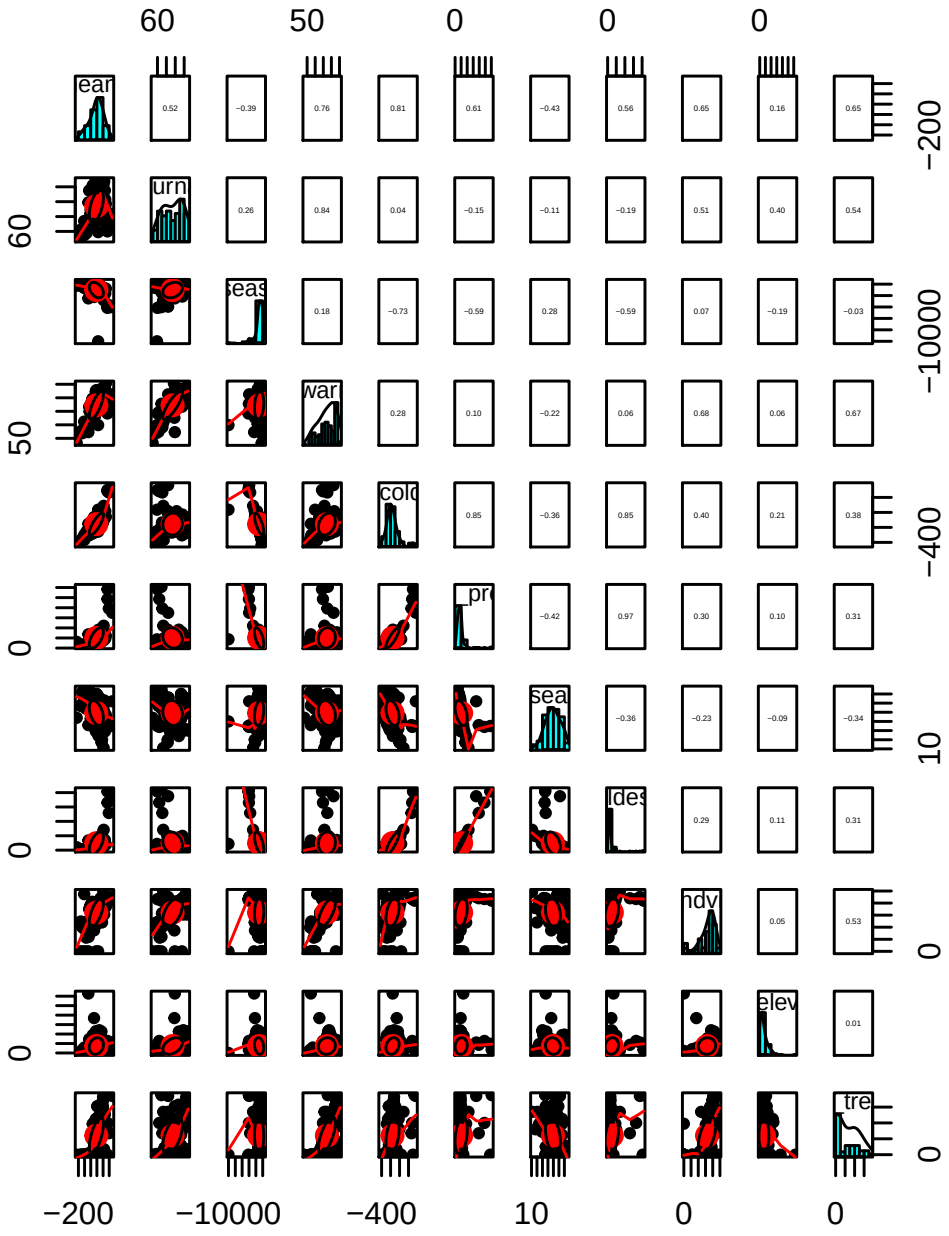
```
##
##  0  1  5  7  8 10 12 14 15 16
##  3 23  7 40  7  4  2  1  1  6
```

The distribution of the factor levels is very uneven and dominated by classes 1,7 & 8, so either we would need to combine classes or delete the predictor.

```
env <- env[, -13]  #remove land cover
```

```
pairs.panels(env[, 5:15], scale = T)
```

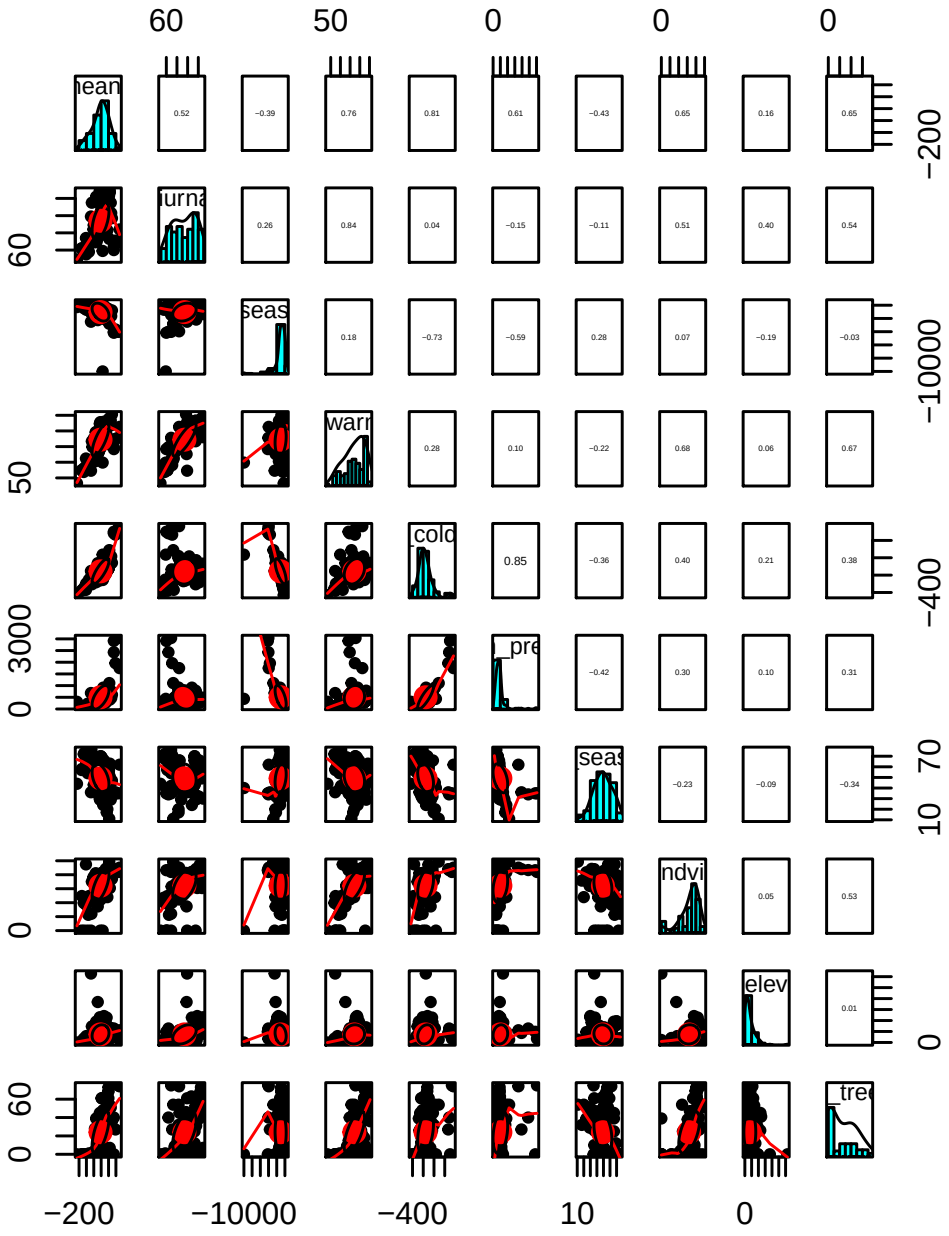

UNIVERSITY OF CANBERRA dartR DArTSeq™ Data



And we also remove factors with high collinearity

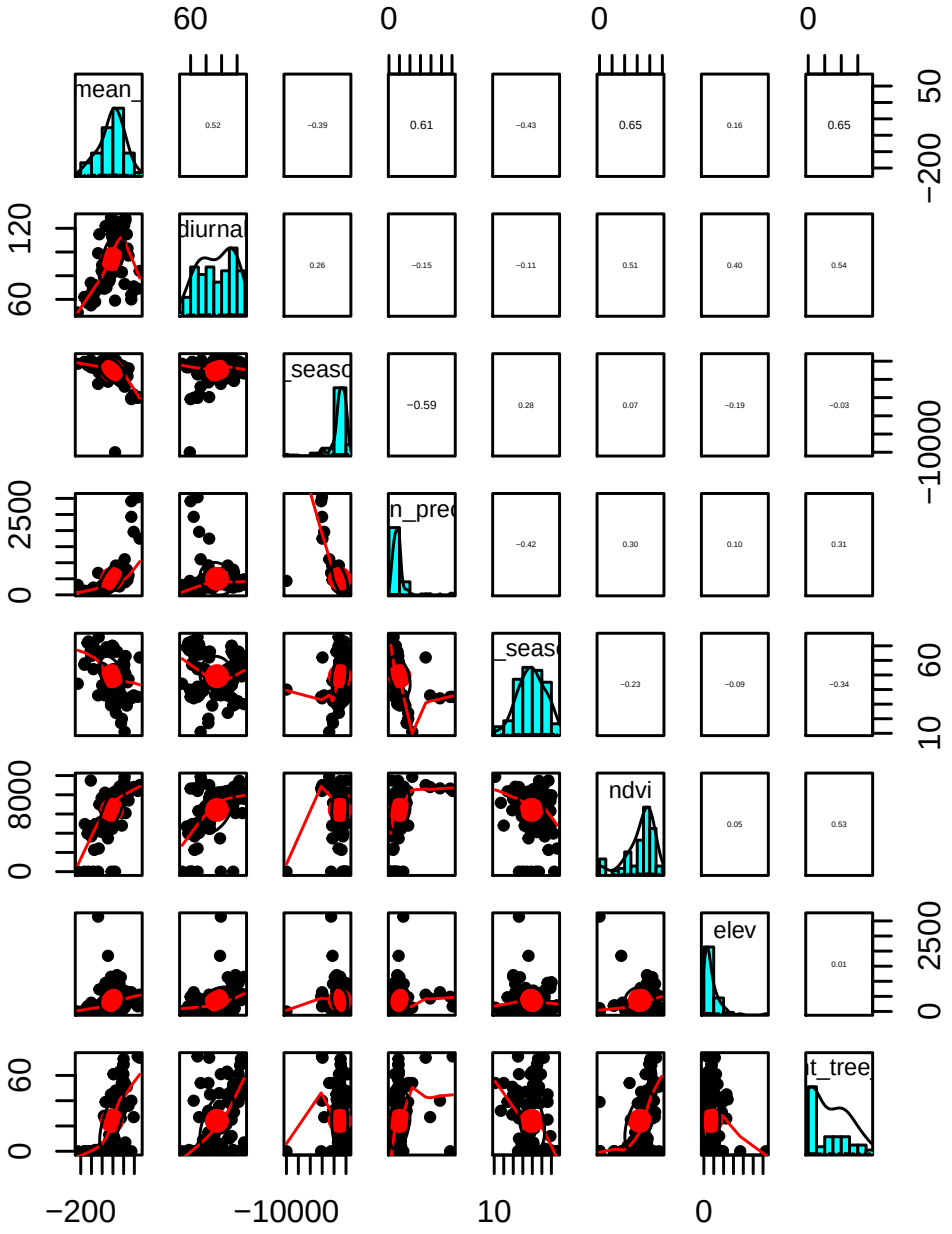
```
env <- env[, -12]  # remove precip coldest quarter: 0.97 corr with annual precip
pairs.panels(env[, 5:14], scale = T)
```


UNIVERSITY OF CANBERRA
dartR
DArTSeq™ Data




```
env <- env[, -c(8, 9)]  # remove maxT_warm, minT_cold; covers remaining strong correlations in a small # of p
pairs.panels(env[, 5:12], scale = T)
```


UNIVERSITY OF CANBERRA
dartR
DArTSeq™ Data




```
# Try this reduced set of predictors:
pred <- env[, 5:12]
colnames(pred) <- c("AMT", "MDR", "sdT", "AP",
    "cvP", "NDVI", "Elev", "Tree")
```


UNIVERSITY OF CANBERRA
dartR
DArTSeq™ Data

Finally we need to scale those predictors (as they are in different units)

```
pred <- scale(pred, scale = TRUE, center = TRUE)  # default is center and scale = T; this subtracts the mean
```

Running the actual RDA is fairly simple:

```
wolf.rda <- rda(gen ~ ., data = as.data.frame(pred),
    scale = T)  # vegan wants data frames, not matrices

wolf.rda
```

```
## Call: rda(formula = gen ~ AMT + MDR +
## sdT + AP + cvP + NDVI + Elev + Tree,
## data = as.data.frame(pred), scale = T)
##
##                  Inertia Proportion Rank
## Total          1.000e+04  1.000e+00
## Constrained    1.357e+03  1.357e-01    8
## Unconstrained 8.643e+03  8.643e-01   85
## Inertia is correlations
##
## Eigenvalues for constrained axes:
##  RDA1  RDA2  RDA3  RDA4  RDA5  RDA6  RDA7
## 344.8 267.2 226.8 125.2 106.4 102.5  94.3
##  RDA8
##  89.4
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7
## 341.7 279.6 215.1 160.2 147.1 141.6 130.2
##   PC8
## 128.6
## (Showing 8 of 85 unconstrained eigenvalues)
```

We have as many constrained ("RDA") axes as we have predictors. Residual variance is then modeled by PCA (the unconstrained "PC" axes). Variance explained by the environmental predictors is under "Proportion" and "Constrained" = equivalent to R2 in multiple regression, so it is biased & should be adjusted based on the # of predictors:

```
RsquareAdj(wolf.rda)
```

```
## $r.squared
## [1] 0.1356641
##
## $adj.r.squared
## [1] 0.05431482
```

Our constrained ordination explains about 5% of the variation [Not to surprising given our large # of SNPs and other processes at work]

How many of our 8 constrained axes should we retain for detecting outliers?

We can do a formal significance test, but it takes quite a while to run…(& is probably not necessary!)

```
signif.axis <- anova.cca(wolf.rda, by = "axis")
# takes a few HOURS for the full 41K SNP data
# set [See ?anova.cca for more details and
# options]
```

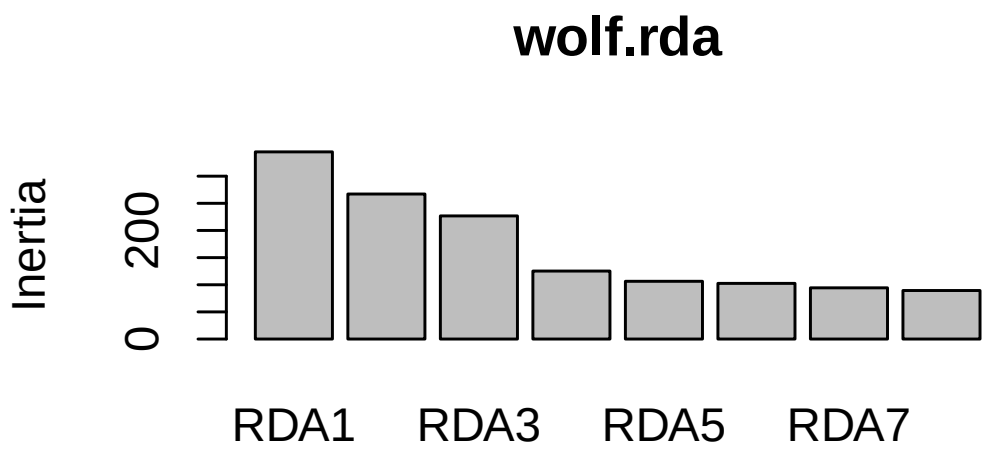
Instead, let's look at the eigenvalues for the constrained axes & plot them

```
summary(eigenvals(wolf.rda, model = "constrained"))
```

```
## Importance of components:
##                            RDA1     RDA2
## Eigenvalue            344.7729 267.1741
## Proportion Explained    0.2541   0.1969
## Cumulative Proportion   0.2541   0.4511
##                            RDA3      RDA4
## Eigenvalue            226.8258 125.22824
## Proportion Explained    0.1672   0.09231
## Cumulative Proportion   0.6183   0.71058
##                             RDA5      RDA6
## Eigenvalue            106.38043 102.54389
## Proportion Explained    0.07841   0.07559
## Cumulative Proportion   0.78899   0.86458
##                            RDA7     RDA8
## Eigenvalue            94.34423 89.37134
## Proportion Explained   0.06954  0.06588
## Cumulative Proportion  0.93412  1.00000
```

```
screeplot(wolf.rda)  # variance explained drops off after 3 axes
```


UNIVERSITY OF CANBERRA dartR DArTSeq™ Data



Finally let's check VIFs for our predictors (they should be ok)

```
vif.cca(wolf.rda)
```

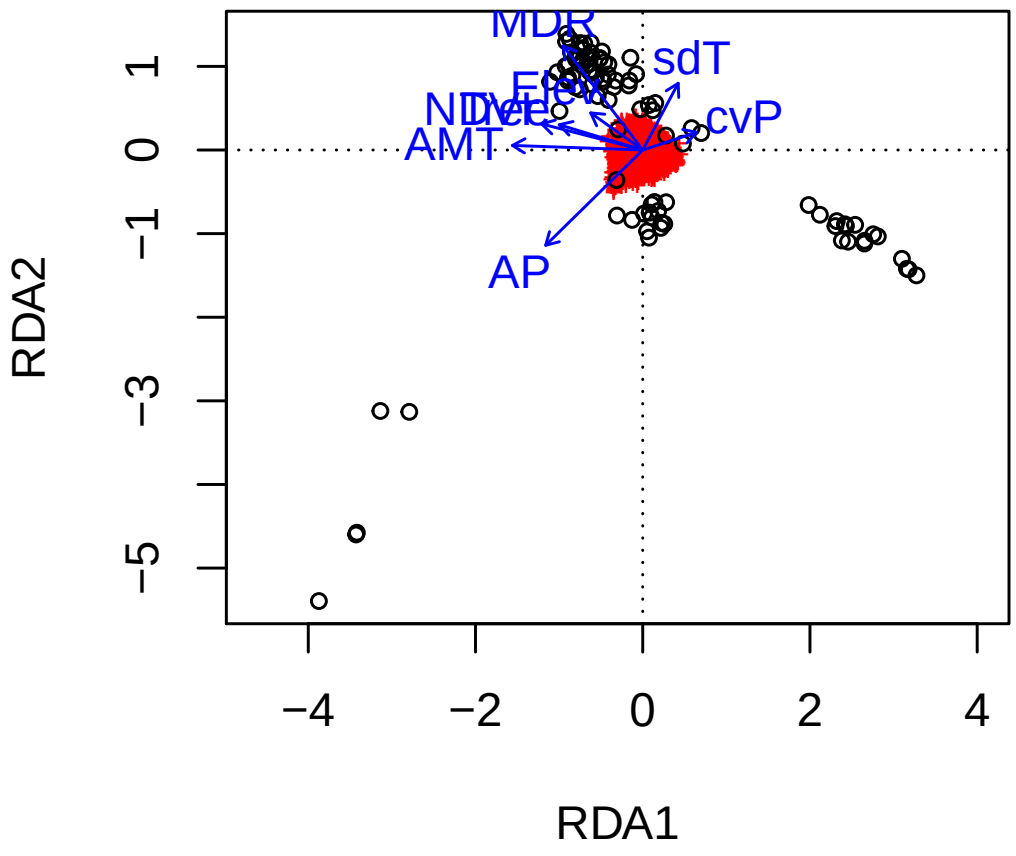
```
##       AMT      MDR      sdT       AP      cvP
## 7.854243 6.495892 2.775059 4.051610 1.318631
##     NDVI     Elev     Tree
## 2.285632 2.028377 2.260139
```

All are below 10 and most are below 5 = multicollinearity shouldn't be a problem.

Plot the RDA

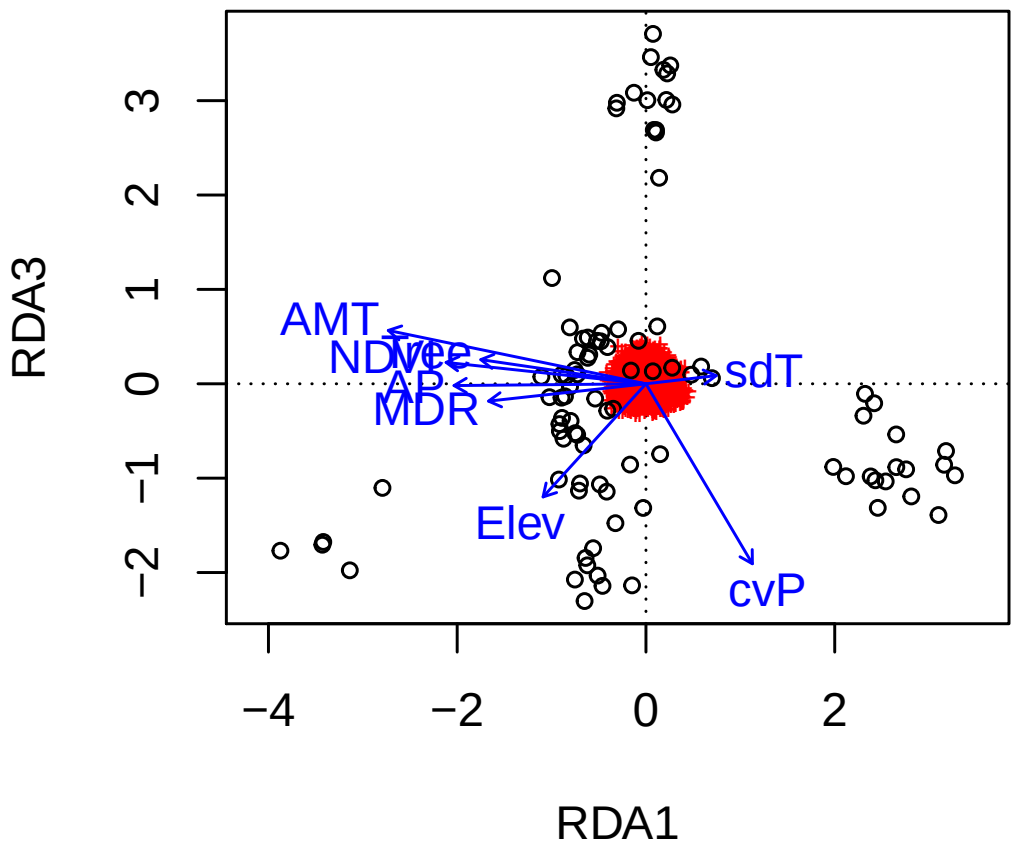
Simple plots

```
plot(wolf.rda, scaling = 3)
```



scaling=3 means symmetrical scaling (SNP & indiv scores are scaled by the square root of the eigenvalues) [See Borcard et al. 2011 or vegan help for more information on scaling RDA plots]

```
plot(wolf.rda, choices = c(1, 3), scaling = 3)  # axes 1 and 3
```



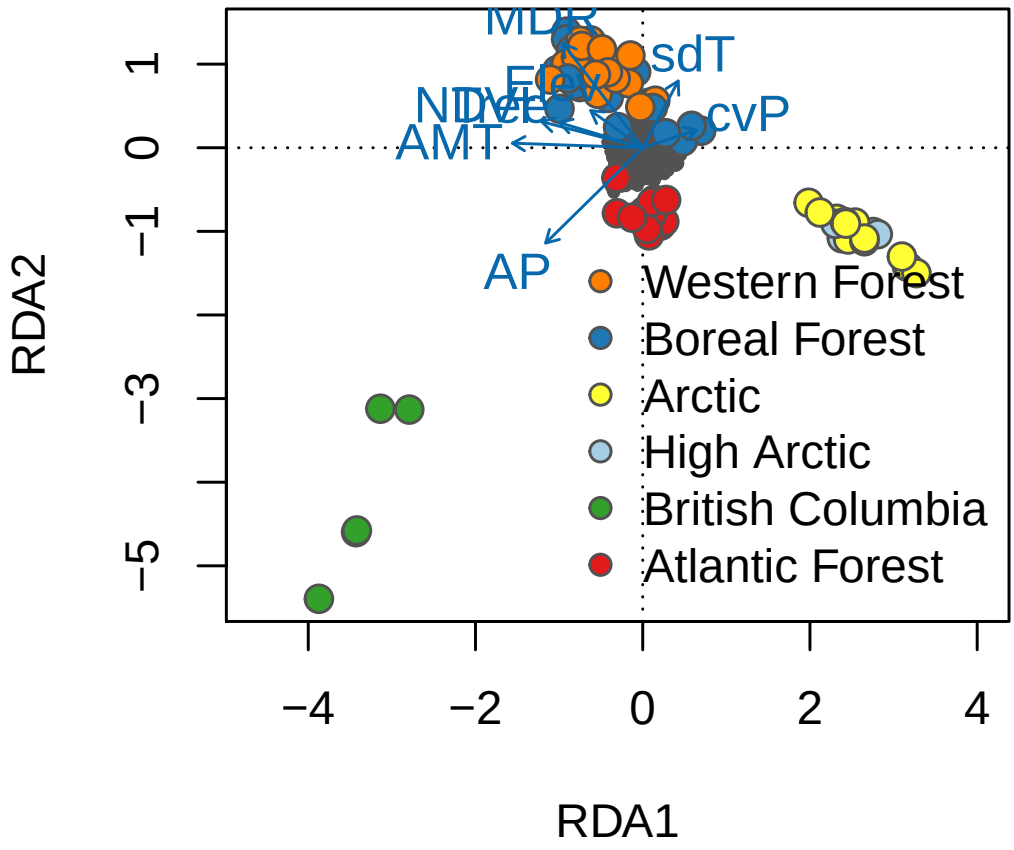
SNPs are in red; individuals are black circles; blue vectors are predictors.

The relative arrangement of these items in the ordination space reflects their relationship with the ordination axes, which are linear combinations of the predictor variables

Let's make some nicer plots:

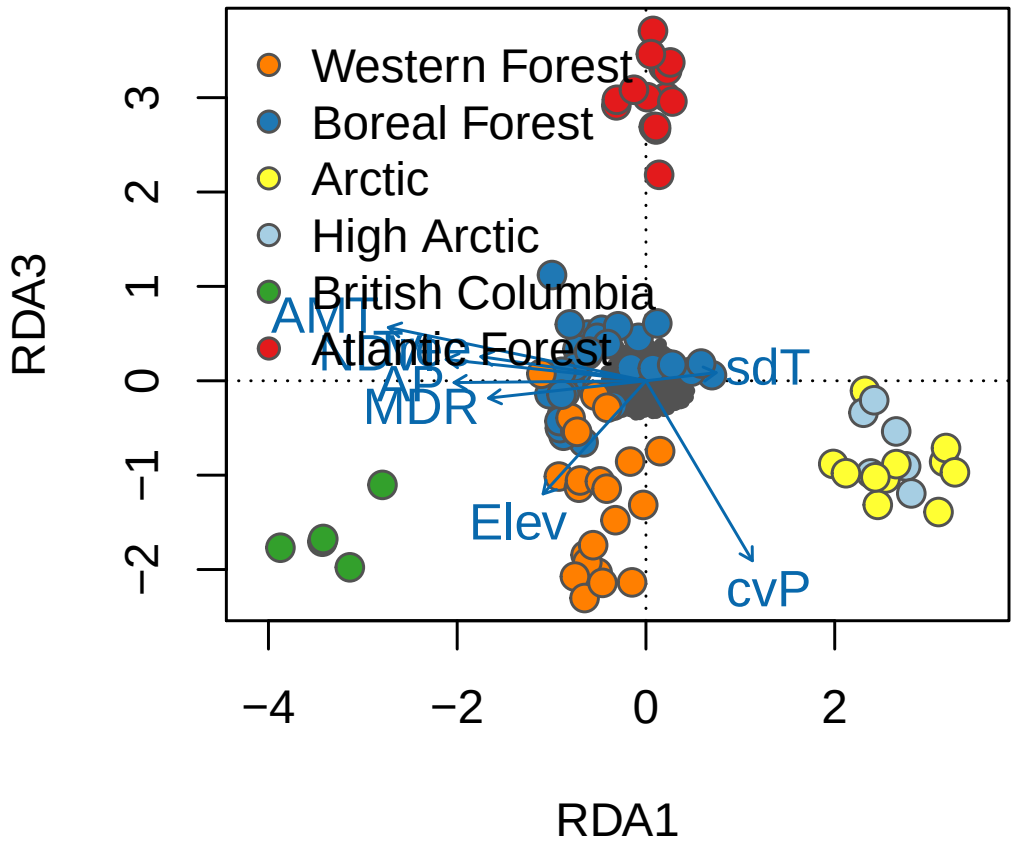
```
levels(env$ecotype) <- c("Western Forest", "Boreal Forest",
    "Arctic", "High Arctic", "British Columbia",
    "Atlantic Forest")
colorby <- env$ecotype
bg <- c("#ff7f00", "#1f78b4", "#ffff33", "#a6cee3",
    "#33a02c", "#e31a1c")

# axes 1 & 2
```

```
plot(wolf.rda, type = "n", scaling = 3)
points(wolf.rda, display = "species", pch = 20,
    cex = 0.7, col = "gray32", scaling = 3)  # the snps
points(wolf.rda, display = "sites", pch = 21,
    cex = 1.3, col = "gray32", scaling = 3, bg = bg[colorby])  # the wolves
text(wolf.rda, scaling = 3, display = "bp", col = "#0868ac",
    cex = 1.1)  # the predictors
legend("bottomright", legend = levels(colorby),
    bty = "n", col = "gray32", pch = 21, cex = 1,
    pt.bg = bg)
```



UNIVERSITY OF CANBERRA
dartR
DArTSeq™ Data

```
# axes 1 & 3
plot(wolf.rda, type = "n", scaling = 3, choices = c(1,
    3))
points(wolf.rda, display = "species", pch = 20,
    cex = 0.7, col = "gray32", scaling = 3, choices = c(1,
        3))
points(wolf.rda, display = "sites", pch = 21,
    cex = 1.3, col = "gray32", scaling = 3, bg = bg[colorby],
    choices = c(1, 3))
text(wolf.rda, scaling = 3, display = "bp", col = "#0868ac",
    cex = 1.1, choices = c(1, 3))
legend("topleft", legend = levels(colorby), bty = "n",
    col = "gray32", pch = 21, cex = 1, pt.bg = bg)
```



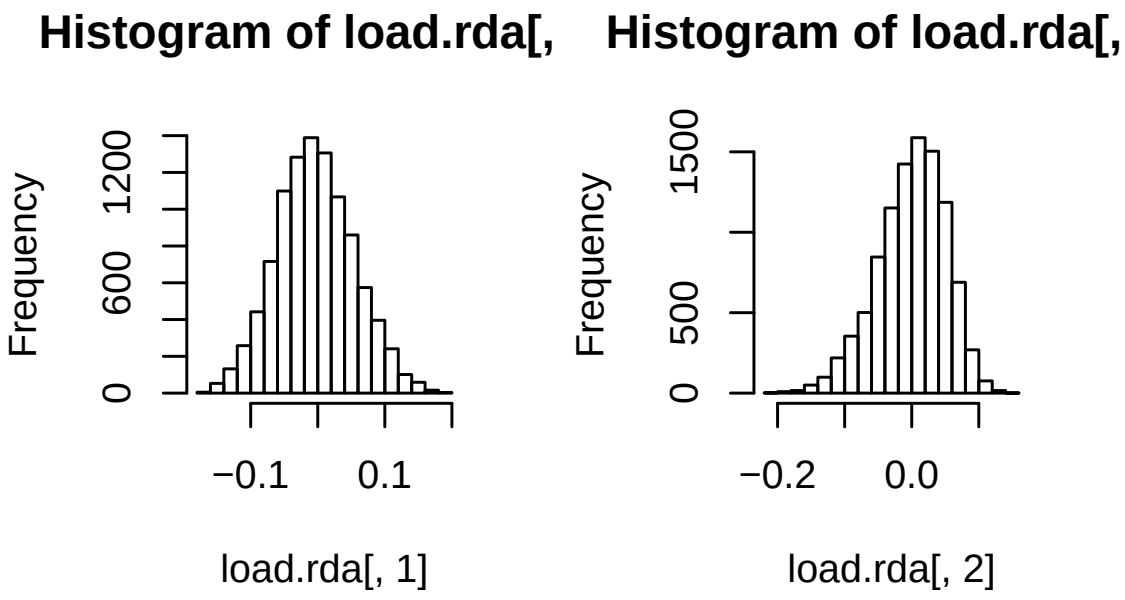
The impact of continental scale climate is reflected in these plots: BC is wet with low temp seasonality; Arctic areas have small mean diurnal range, low annual temps, andlow tree cover and veg. greenness (NDVI); Atlatic and Western Forsts show weak & strong precip seasonality, respectively.

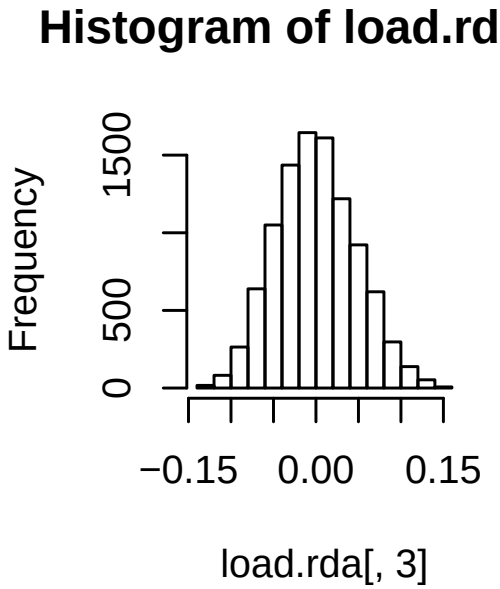
Identify candidates

In their publication (Forester et al. 2018, ME), they pulled SNPs with extreme loadings from relevant constrained axes:

```
load.rda <- summary(wolf.rda)$species[, 1:3]  # RDA loadings ('species') for axes 1:3


par(mfrow = c(2, 2))
```

```
hist(load.rda[, 1])
hist(load.rda[, 2])
hist(load.rda[, 3])  # loadings relatively normally distributed
par(mfrow = c(1, 1))
```





This approach works well when selection gradients are well understood (high TP, low FP).

For analyses with many predictors, or where the analysis is more exploratory, a multidimensional outlier identification approach will help reduce FP rates (though this also brings down TP rates, of course).

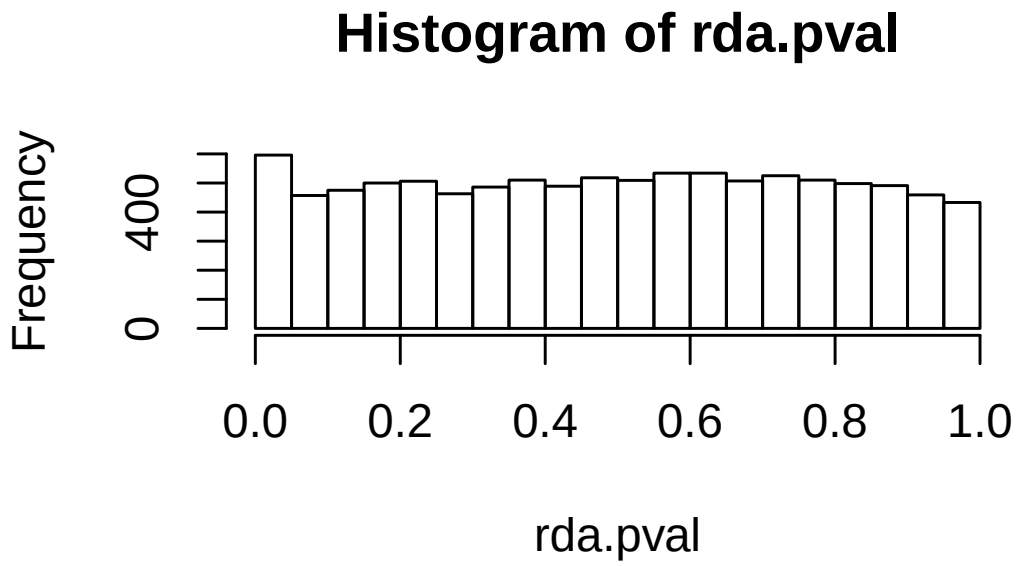
Mahalanobis post-processing of RDA outlier detection results:

Capblancq T, Luu K, Blum MGB, Bazin E (2018) Mol Ecol Resources

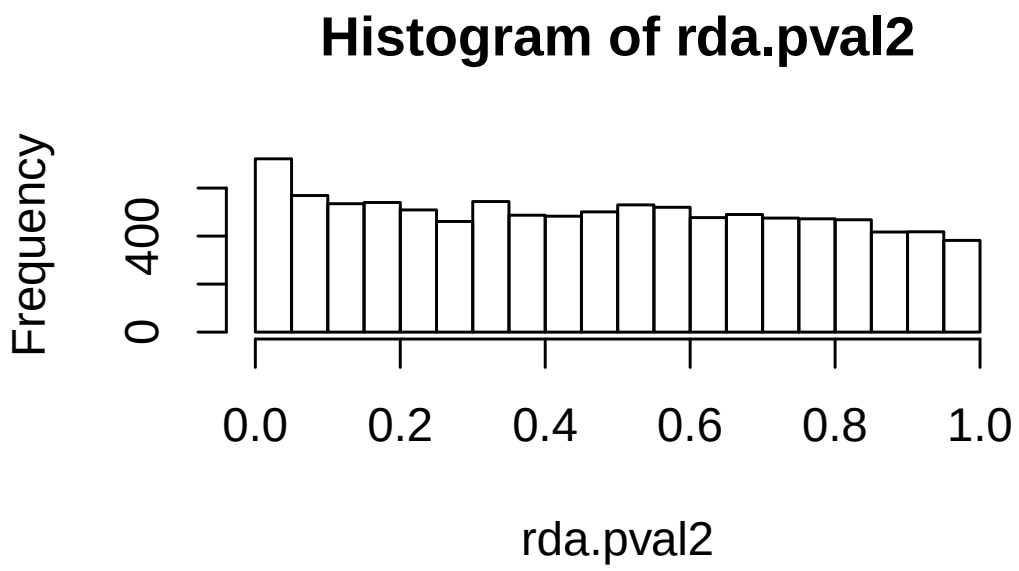
For code to pull outliers on each axis, see our tutorial/vignette: `https://popgen.nescent.org/2018-03-27_RDA_GEA.html`

We will just use the Mahalanobis distance approach suggested by Capblancq et al.

```
K <- 3  # the number of RDA axes you're looking at

zscale <- apply(load.rda, 2, scale)  # I'm not sure scaling is the best approach here...
mscale <- covRob(zscale, distance = TRUE, na.action = na.omit,
    estim = "pairwiseGK")$dist
gif <- median(mscale)/qchisq(0.5, df = K)

rda.pval <- pchisq(mscale/gif, df = K, lower.tail = FALSE)  # Remember: you can always change the GIF!!
hist(rda.pval)  #looks a little conservative
```



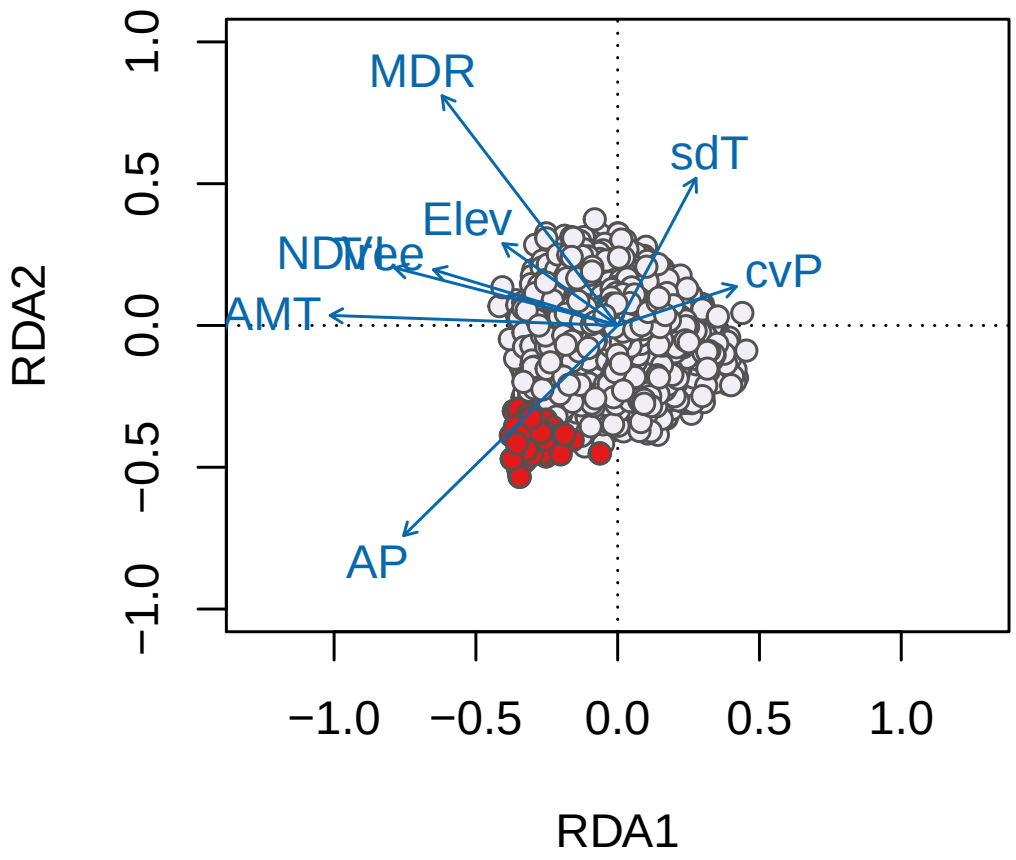

```
rda.pval2 <- pchisq(mscale/1.15, df = K, lower.tail = FALSE)  # Less conservative GIF (smaller)
hist(rda.pval2)
```



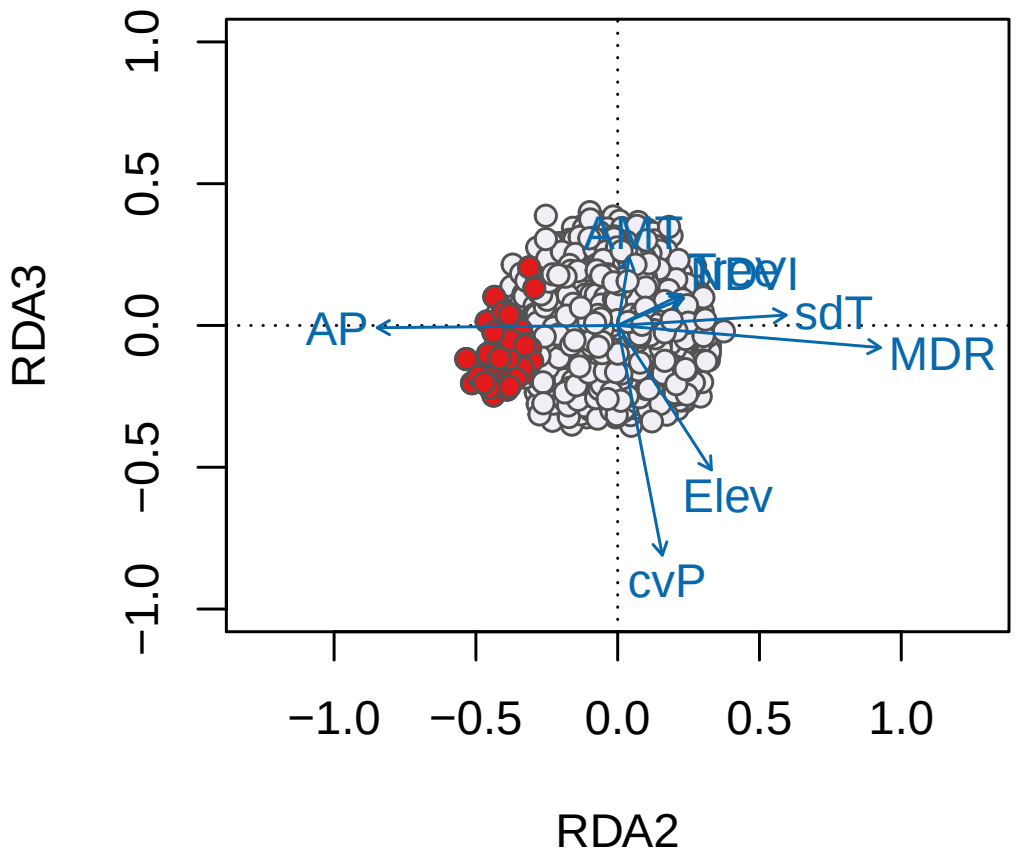

```
rda.qval <- qvalue(rda.pval2)$qvalues
rda.FDR.1 <- colnames(gen)[which(rda.qval < 0.1)]
```

Let's look at where these candidates are in the ordination space:

```
# color by predictor:
snp.color <- as.data.frame(colnames(gen), stringsAsFactors = F)
snp.color[, 2] <- apply(snp.color, 1, function(x) if (x %in%
    rda.FDR.1) "gray32" else "#00000000")
snp.color[, 3] <- apply(snp.color, 1, function(x) if (x %in%
    rda.FDR.1) "#e31a1c" else "#00000000")

# axes 1 & 2
plot(wolf.rda, type = "n", scaling = 3, xlim = c(-1,
    1), ylim = c(-1, 1))
points(wolf.rda, display = "species", pch = 21,
    cex = 1, col = "gray32", bg = "#f1eef6", scaling = 3)
points(wolf.rda, display = "species", pch = 21,
    cex = 1, col = snp.color[, 2], bg = snp.color[,
        3], scaling = 3)
text(wolf.rda, scaling = 3, display = "bp", col = "#0868ac",
    cex = 1)
```




```
# axes 2 & 3
plot(wolf.rda, type = "n", scaling = 3, xlim = c(-1,
    1), ylim = c(-1, 1), choices = c(2, 3))
points(wolf.rda, display = "species", pch = 21,
    cex = 1, col = "gray32", bg = "#f1eef6", scaling = 3,
    choices = c(2, 3))
points(wolf.rda, display = "species", pch = 21,
    cex = 1, col = snp.color[, 2], bg = snp.color[,
        3], scaling = 3, choices = c(2, 3))
text(wolf.rda, scaling = 3, display = "bp", col = "#0868ac",
    cex = 1, choices = c(2, 3))
```



Well done!!

That is the end. Well done you finished. Feel free to have another go or have a well deserved beverage!!!