Starling-May18
Projects/Katarina

PDF Version generated by

Katarina Stuart (z5188231@ad.unsw.edu.au)

on

Feb 12, 2022 @03:41 PM AEDT

Table of Contents

2021.06.18.GradientForest_allgeno

Katarina Stuart (z5188231@ad.unsw.edu.au) - Jul 04, 2021, 5:17 PM AEST

Gradient Forest - all 24 sites

Refilter data so it has all sample sites

```
module load samtools/1.10
module load java/8u121
module load gatk/4.1.0.0
module load picard/2.18.26
module load vcftools/0.1.16
```

```
cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv6_Morphology/analysis/ml_mapping/gradientforest_allgeno
```

```
vcftools --vcf /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv6_Morphology/data/vcf/populations_sorted_reordered.vcf --max-missing 0.5 --maf 0.05 --min-meanDP 2 --max-meanDP 50 --min-alleles 2 --max-alleles 2 --minGQ 15 --recode --out populations_sorted_reordered_maf005_miss50
```

```
bcftools +prune -l 0.6 -w 1000 populations_sorted_reordered_maf005_miss50.recode.vcf -Ov -o populations_sorted_reordered_maf005_miss50_r2.vcf
```

working out individual missingness:

```
vcftools --vcf populations_sorted_reordered_maf005_miss50_r2.vcf --missing-indv --out populations_sorted_reordered_maf005_miss50_r2.vcf
```

manually check for missingness levels at my filter50 thresholds. Filter for those individuals;

```
vcftools --vcf populations_sorted_reordered_maf005_miss50_r2.vcf --keep populations_sorted_reordered_maf005_miss50_r2_indmiss50.txt --recode --out populations_sorted_reordered_maf005_miss50_r2_missind50
```

```
module load vcftools/0.1.16
VCF=populations_sorted_reordered_maf005_miss50_r2_missind50.recode.vcf
vcftools --vcf ${VCF} --012 --out populations_sorted_reordered_maf005_miss50_r2_missind50
```

```
cut -f2- populations_sorted_reordered_maf005_miss50_r2_missind50.012 | sed 's/-1/NA/g' >populations_sorted_reordered_maf005_miss50_r2_missind50.temp
tr -d '\t' <populations_sorted_reordered_maf005_miss50_r2_missind50.012.pos | tr '\n' '\t' | sed 's/[[:space:]]*$//' >header
paste <(echo "ID" | cat - populations_sorted_reordered_maf005_miss50_r2_missind50.012.indv) <(echo "" | cat header - populations_sorted_reordered_maf005_miss50_r2_missind50.temp) > populations_sorted_reordered_maf005_miss50_r2_missind50.forR
rm header populations_sorted_reordered_maf005_miss50_r2_missind50.temp
```

Move into R

```
module load python/3.8.3
module load perl/5.28.0
module load gdal/3.2.1
module load R/3.5.3
R
library(gradientForest)
```

```
setwd("/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv6_Morphology/analysis/ml_mapping/gradientforest_allgeno")
starling.snp <- read.table("populations_sorted_reordered_maf005_miss50_r2_missind50.forR", header = T, row.names = 1)
```

```
library(raster)
sample.coord <-read.table("samp321_lat_long.txt", header=T, stringsAsFactors=F)
sample.coord
points_samp <- SpatialPoints(sample.coord, proj4string=climdata@crs)
#climdata
climdata <- getData('worldclim',download=TRUE,var='bio',res=5)
values_clim <- extract(climdata,points_samp)
#Elevation/alt
altdata <- getData('alt',country='AUS', mask=TRUE)
values_alt <- extract(altdata,points_samp)
#combine
clim.points <- cbind.data.frame(sample.coord, values_clim, values_alt)
clim.points <- cbind.data.frame(sample.coord, values_clim[,c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12", "bio15",  "bio18",  "bio19")], values_alt)
```

Gradient forest (GF) analysis

model the associations of spatial and climate variables with allele frequencies (genotypes) of individuals. For the spatial variables, one could use latitude and longitude, but a more sophisticated approach might be to use PCNMs or MEMs (principal coordinates of neighbor matrices or Moran's eigenvector maps). These approaches generate a set of uncorrelated spatial variables. Code to generate the PCNM spatial variables:

```
library(vegan)
```

```
coord <- clim.points[,c("Longitude","Latitude")]
pcnm <- pcnm(dist(coord))  #this generates the PCNMs, you could stop here if you want all of them
keep <- round(length(which(pcnm$value > 0))/2)
pcnm.keep <- scores(pcnm)[,1:keep]  #keep half of positive ones as suggested by some authors
pcnm.keep
```

create a file that contains only the climate and PCNM spatial variables (no lat/lon). In GF, a maximum number of splits can be defined following the developers suggestion

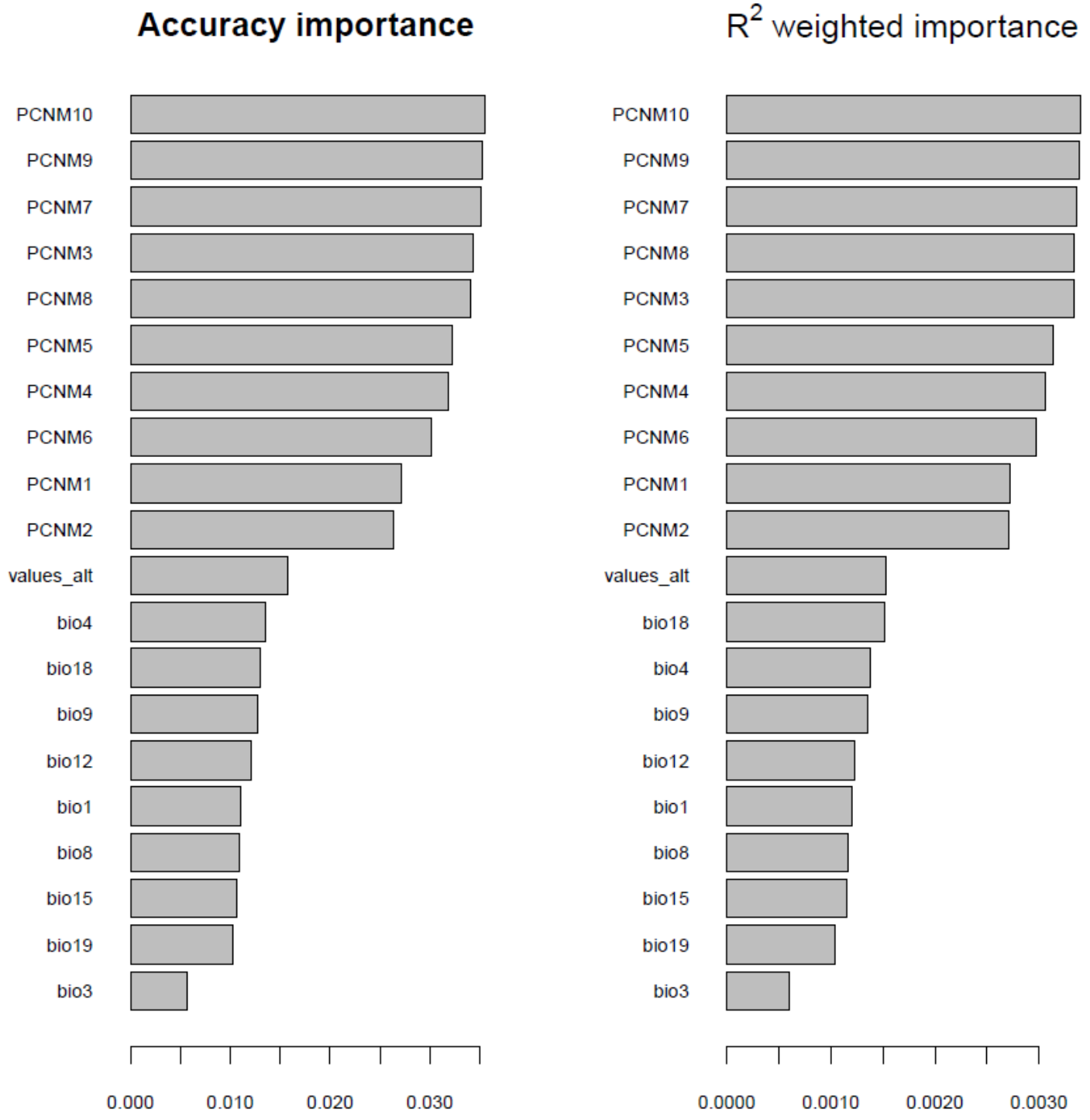
```
library(gradientForest)
env.gf <- cbind(clim.points[,3:12], pcnm.keep)
maxLevel <- log2(0.368*nrow(env.gf)/2)
```

Run the GF [took 12-18 hrs to run on high mem node in R]

```
gf <- gradientForest(cbind(env.gf, starling.snp), predictor.vars=colnames(env.gf),
response.vars=colnames(starling.snp), ntree=500, maxLevel=maxLevel, trace=T, corr.threshold=0.50)
```

When it finishes, there will be warnings about having less than five values for response variables, which is because we have only three: 0, 1, or 2. You can ignore them.

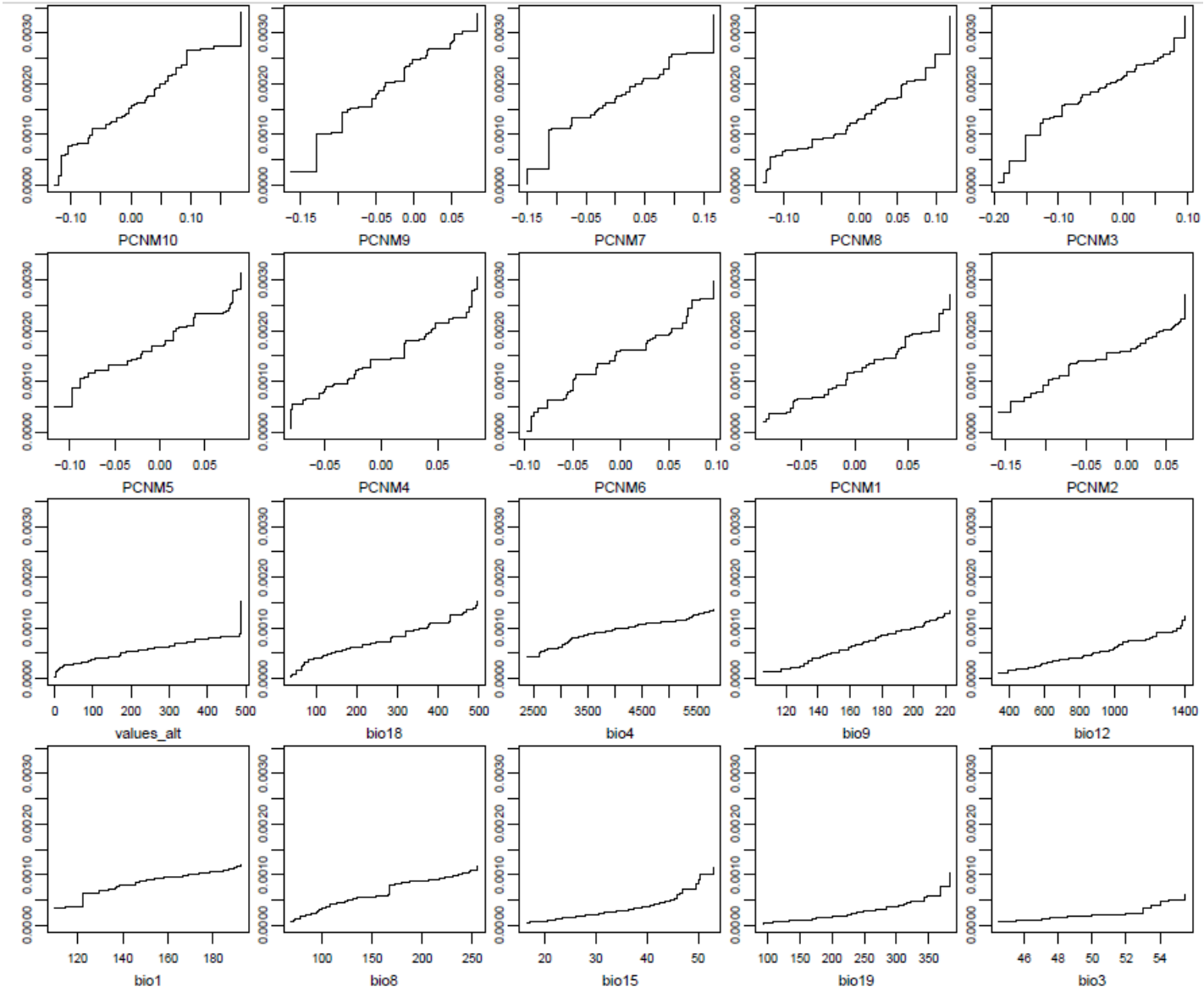
```
pdf("Sv6_gradientforest_model2_VariableImportance.pdf")
plot(gf, plot.type = "O")
dev.off()
```



We can also plot the "turnover functions" showing how allelic composition changes along the spatial or environmental gradients. The shapes are nonlinear and large jumps show steep genetic changes along certain portions of the environmental gradient. The height that the function acheives on the right side of the plot is the total importance and should match the barplot. First, organize the variables by importance and then plot:

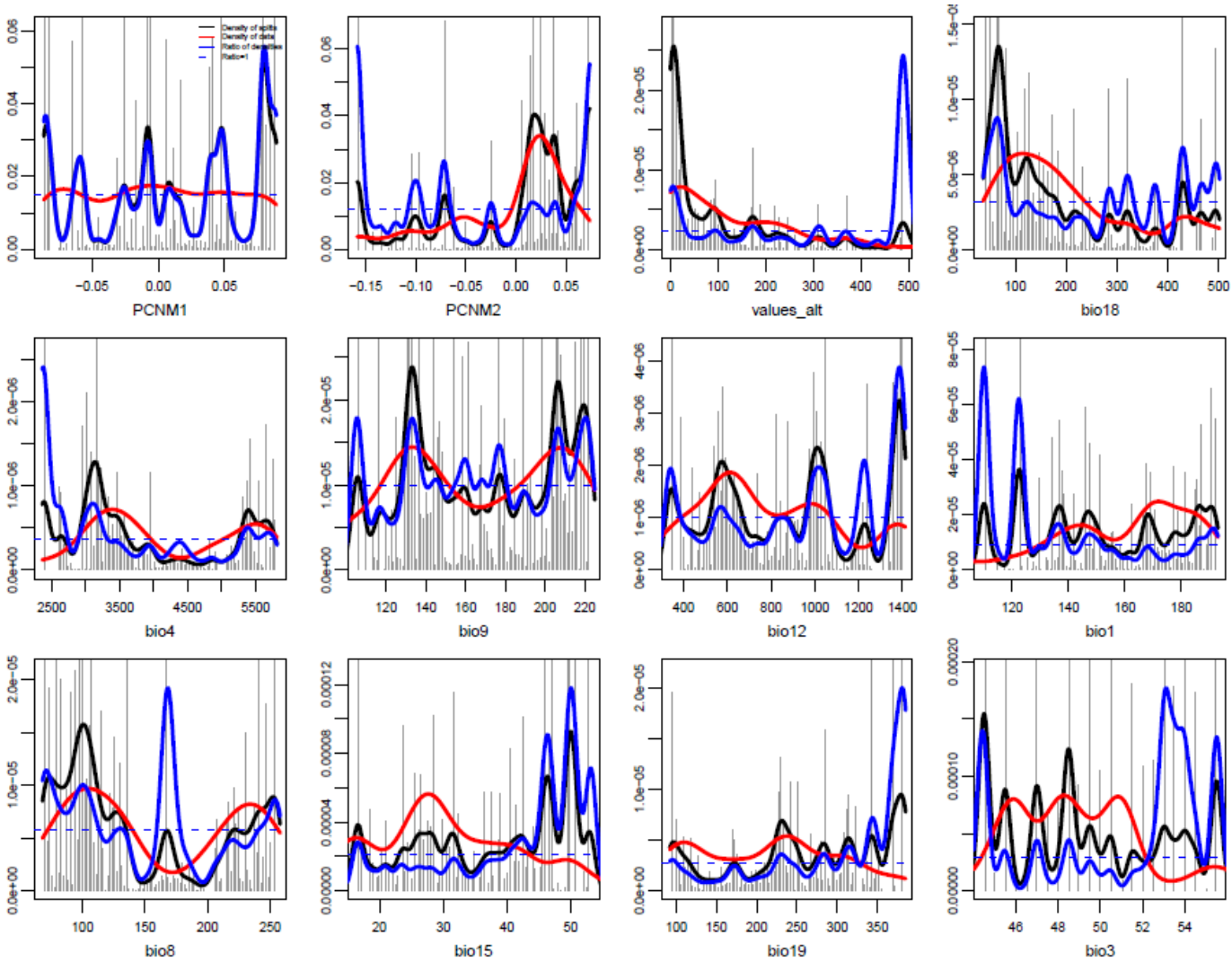
```
most_important <- names(importance(gf))[1:25]
par(mgp = c(2, 0.75, 0))
pdf("Sv6_gradientforest_model2_speccum.pdf")
plot(gf, plot.type = "C", imp.vars = most_important, show.species = F, common.scale = T, cex.axis = 0.6, cex.lab =
```

```
0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5, 0.5, 0), mar = c(2.5, 1, 0.1, 0.5), omi = c(0, 0.3, 0, 0)))
dev.off()
```

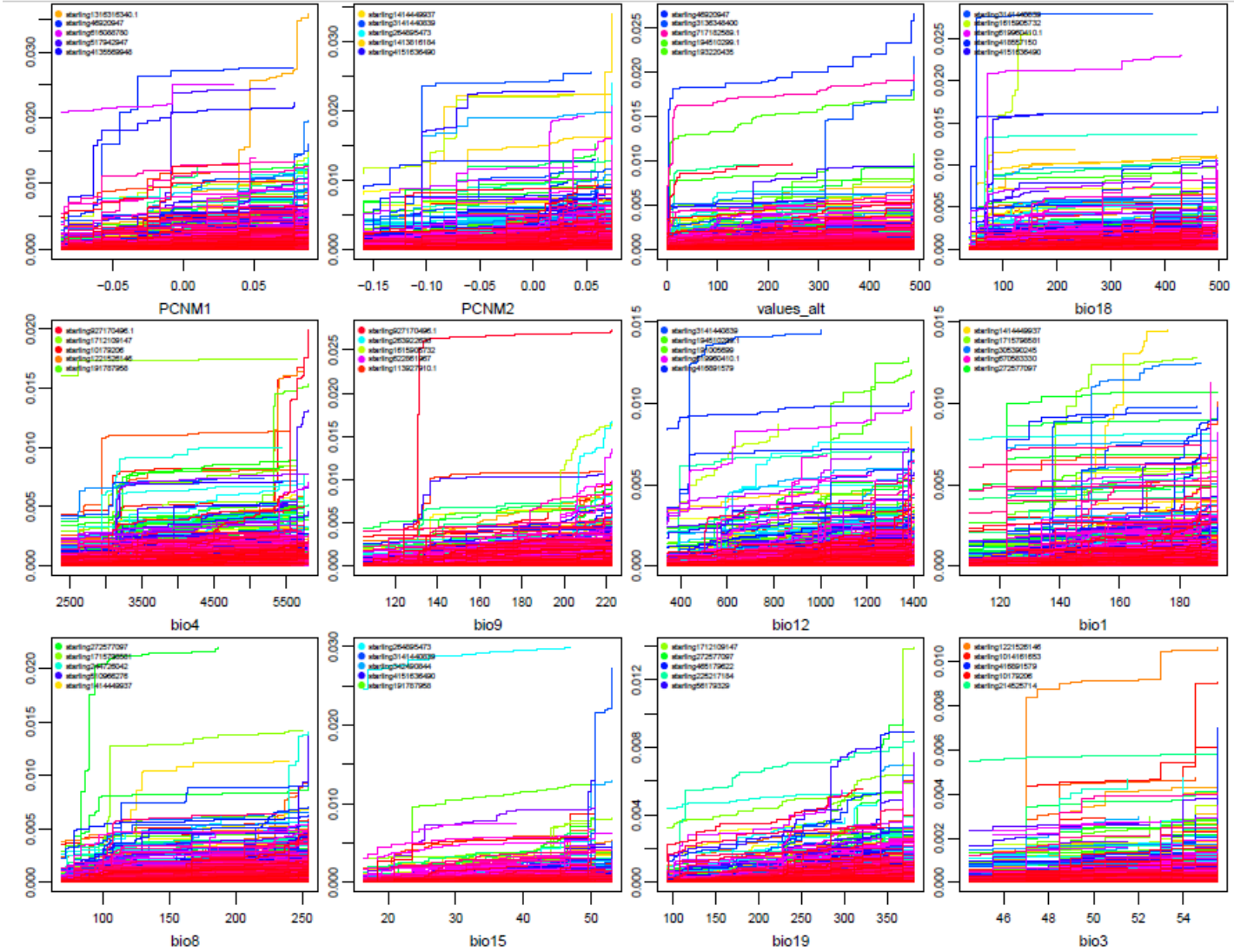



```
most_important <- names(importance(gf))[9:24]
pdf("Sv6_gradientforest_model2_impdens.pdf")
plot(gf, plot.type = "S", imp.vars = most_important, leg.posn = "topright", cex.legend = 0.4, cex.axis = 0.6, cex.lab =
0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5,0.5, 0), mar = c(3.1, 1.5, 0.1, 1)))
dev.off()
```

Error in integrate(approxfun(d, rule = 2), lower = min(d$x), upper = max(d$x)) : roundoff error was detected applicable for the 4th in list of importance??




```
pdf("Sv6_gradientforest_model2_cumimp.pdf")
 plot(gf, plot.type = "C", imp.vars = most_important, show.overall = F, legend = T, leg.posn = "topleft", leg.nspecies = 5, cex.lab = 0.7, cex.legend = 0.4, cex.axis = 0.6, line.ylab = 0.9, par.args = list(mgp = c(1.5, 0.5, 0), mar = c(2.5, 1, 0.1, 0.5), omi = c(0, 0.3, 0, 0)))
dev.off()
```

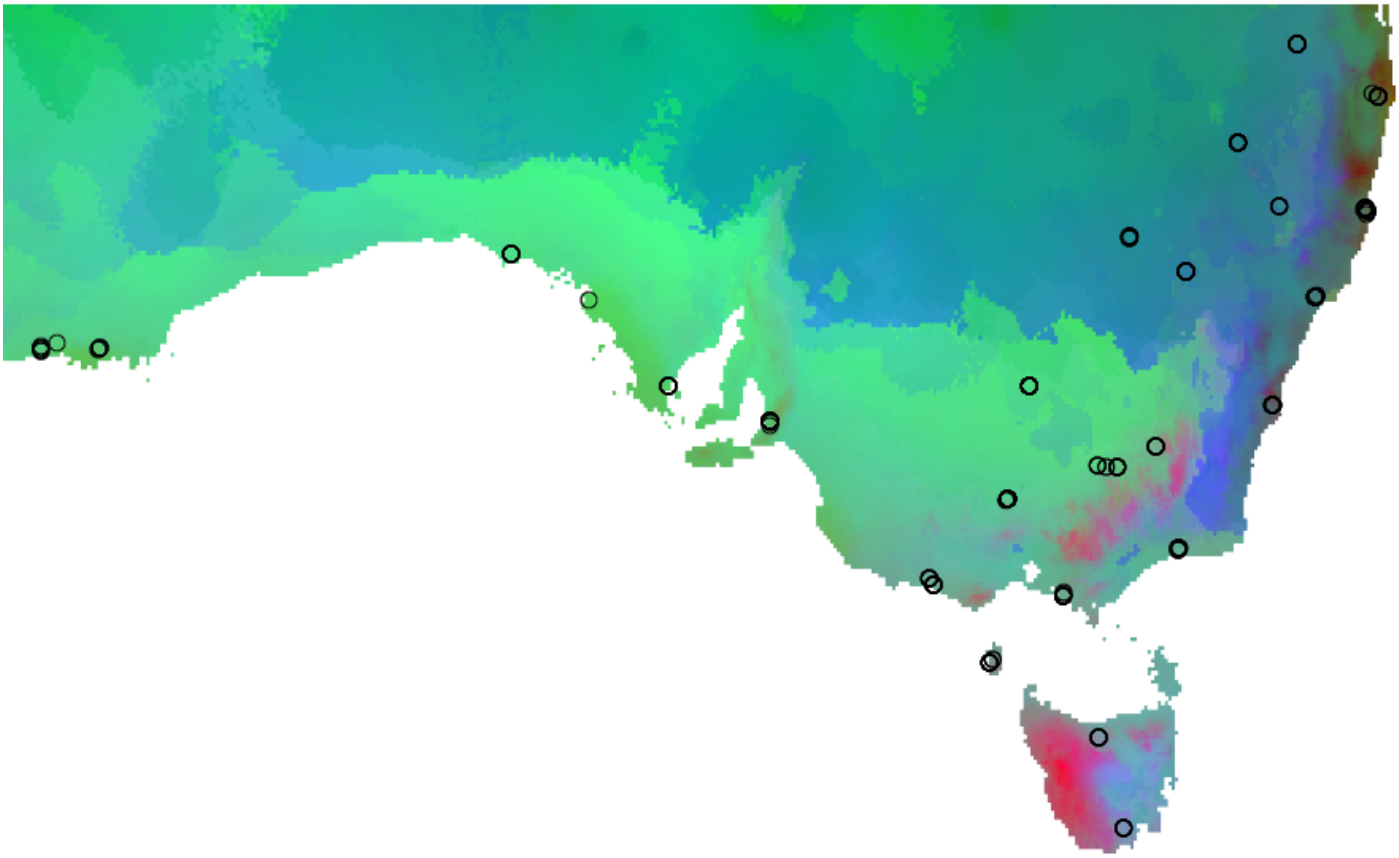



```
extent <- c(120, 155, -44, -27)

values_alt  <- getData('worldclim',download=TRUE,var='alt',res=5)

climdata.subset<- subset(climdata, c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12",  "bio15",  "bio18",  "bio19"))

merged.data <- addLayer(climdata.subset, values_alt)

names(merged.data)[10]<- "values_alt"

clim.layer.crop <- crop(merged.data, extent)

clim.land <- extract(clim.layer.crop, 1:ncell(clim.layer.crop), df = TRUE)
clim.land <- na.omit(clim.land)
pred <- predict(gf, clim.land[,-1])
PCs <- prcomp(pred, center=T, scale.=F)
r <- PCs$x[, 1]
g <- PCs$x[, 2]
b <- PCs$x[, 3]
```

```
r <- (r - min(r))/(max(r) - min(r)) * 255
g <- (g - min(g))/(max(g) - min(g)) * 255
b <- (b - min(b))/(max(b) - min(b)) * 255
mask<-clim.layer.crop$bio4
mask[]<-as.numeric(mask[]>0)
rastR <- rastG <- rastB <- mask
rastR[clim.land$ID] <- r
rastG[clim.land$ID] <- g
rastB[clim.land$ID] <- b
rgb.rast <- stack(rastR, rastG, rastB)
```

```
pdf("Sv6_gradientforest_model2_Map2_predlistwithalt.pdf")
plotRGB(rgb.rast, bgalpha=0)
points(sample.coord$Longitude, sample.coord$Latitude)
dev.off()
```

The colors represent genetic variation as predicted based on the modeled relationships with environmental and spatial variables. Similar colors are more similar genetically.



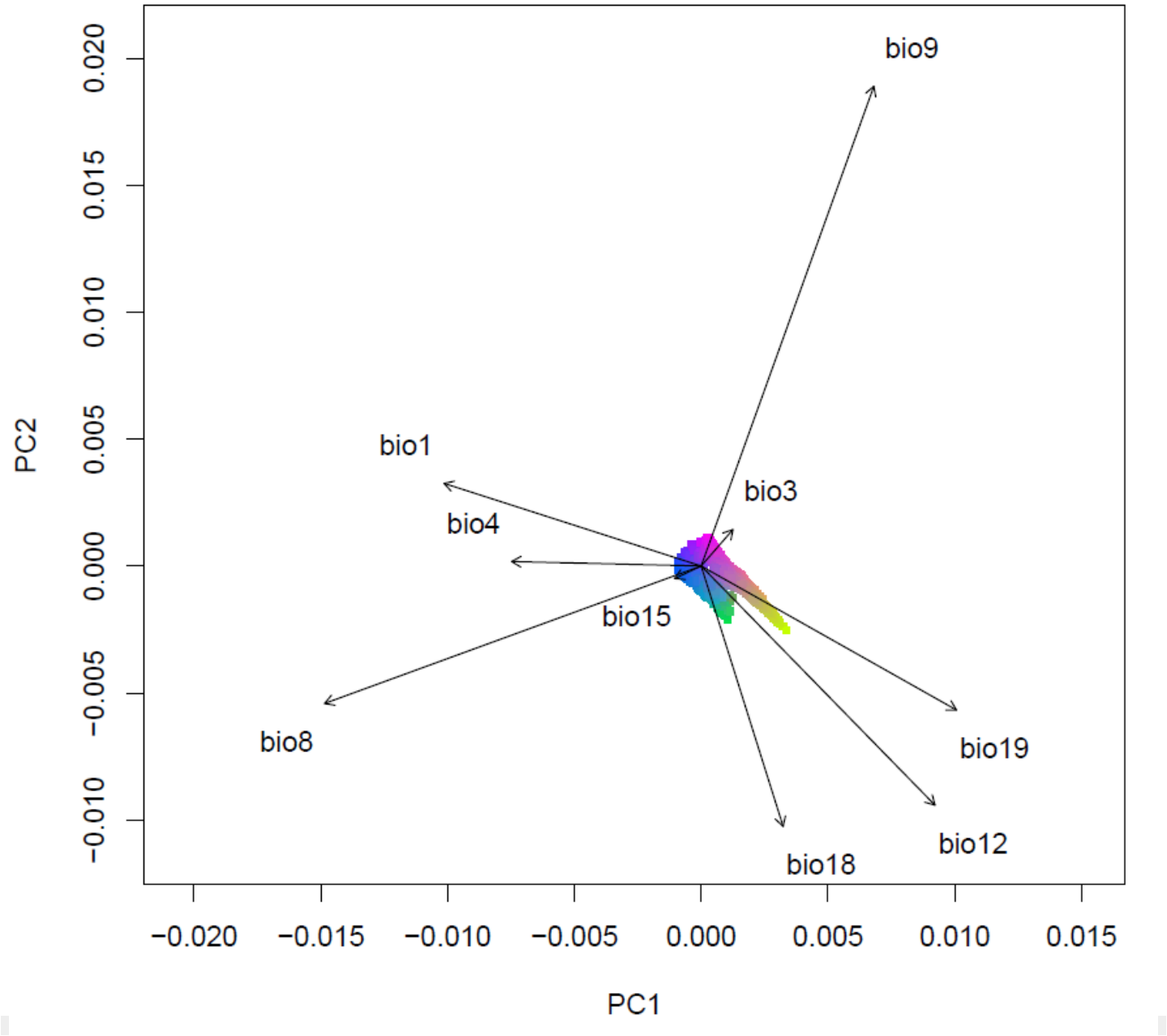
Biplot of the biological space

```
#imp.vars <- names(importance(gf))
#Trns_grid <- cbind(Phys_grid[, c("EAST", "NORTH")], + predict(gf, Phys_grid[, imp.vars]))
#PCs <- prcomp(Trns_grid[, imp.vars])
```

```
a1 <- PCs$x[, 1]
a2 <- PCs$x[, 2]
a3 <- PCs$x[, 3]
r <- a1 + a2
g <- -a2
b <- a3 + a2 - a1
r <- (r - min(r))/(max(r) - min(r)) * 255
g <- (g - min(g))/(max(g) - min(g)) * 255
b <- (b - min(b))/(max(b) - min(b)) * 255
```

```
nvs <- dim(PCs$rotation)[1]
vec <- c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12",  "bio15",  "bio18",  "bio19","values_alt") #picked top from VariableImportance
lv <- length(vec)
vind <- rownames(PCs$rotation) %in% vec
scal <- 40
xrng <- range(PCs$x[, 1], PCs$rotation[, 1]/scal) * 1.1
yrng <- range(PCs$x[, 2], PCs$rotation[, 2]/scal) * 1.1
```

```
pdf("Sv6_gradientforest_model2_biplot_predlist1withalt.pdf")
plot((PCs$x[, 1:2]), xlim = xrng, ylim = yrng, pch = ".", cex = 4, col = rgb(r, g, b, max = 255), asp = 1)
points(PCs$rotation[!vind, 1:2]/scal, pch = "+")
arrows(rep(0, lv), rep(0, lv), PCs$rotation[vec, 1]/scal, PCs$rotation[vec, 2]/scal, length = 0.0625)
jit <- 0.0015
text(PCs$rotation[vec, 1]/scal + jit * sign(PCs$rotation[vec, 1]), PCs$rotation[vec, 2]/scal + jit *
sign(PCs$rotation[vec, 2]), labels = vec)
dev.off()
```



MODEL 3:

```
module load python/3.8.3
module load perl/5.28.0
module load gdal/3.2.1
module load R/3.5.3
R
library(gradientForest)
```

```
setwd("/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv6_Morphology/analysis/ml_mapping/gradientforest_allgeno")
starling.snp <- read.table("populations_sorted_reordered_maf005_miss50_r2_missind50.forR", header = T, row.names = 1)
```

```
library(raster)
sample.coord <-read.table("samp321_lat_long.txt", header=T, stringsAsFactors=F)
sample.coord
points_samp <- SpatialPoints(sample.coord, proj4string=climdata@crs)
#climdata
climdata <- getData('worldclim',download=TRUE,var='bio',res=5)
values_clim <- extract(climdata,points_samp)
#Elevation/alt
altdata <- getData('alt',country='AUS', mask=TRUE)
altitude<- extract(altdata,points_samp)
#combine
clim.points <- cbind.data.frame(sample.coord, values_clim, altitude)
clim.points <- cbind.data.frame(sample.coord, values_clim[,c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12",
 "bio14",  "bio15",  "bio18",  "bio19")], altitude)
```

Gradient forest (GF) analysis

model the associations of spatial and climate variables with allele frequencies (genotypes) of individuals. For the spatial variables, one could use latitude and longitude, but a more sophisticated approach might be to use PCNMs or MEMs (principal coordinates of neighbor matrices or Moran's eigenvector maps). These approaches generate a set of uncorrelated spatial variables. Code to generate the PCNM spatial variables:

```
library(vegan)
```

```
coord <- clim.points[,c("Longitude","Latitude")]
pcnm <- pcnm(dist(coord))  #this generates the PCNMs, you could stop here if you want all of them
keep <- round(length(which(pcnm$value > 0))/2)
pcnm.keep <- scores(pcnm)[,1:keep]  #keep half of positive ones as suggested by some authors
pcnm.keep
```

create a file that contains only the climate and PCNM spatial variables (no lat/lon). In GF, a maximum number of splits can be defined following the developers suggestion

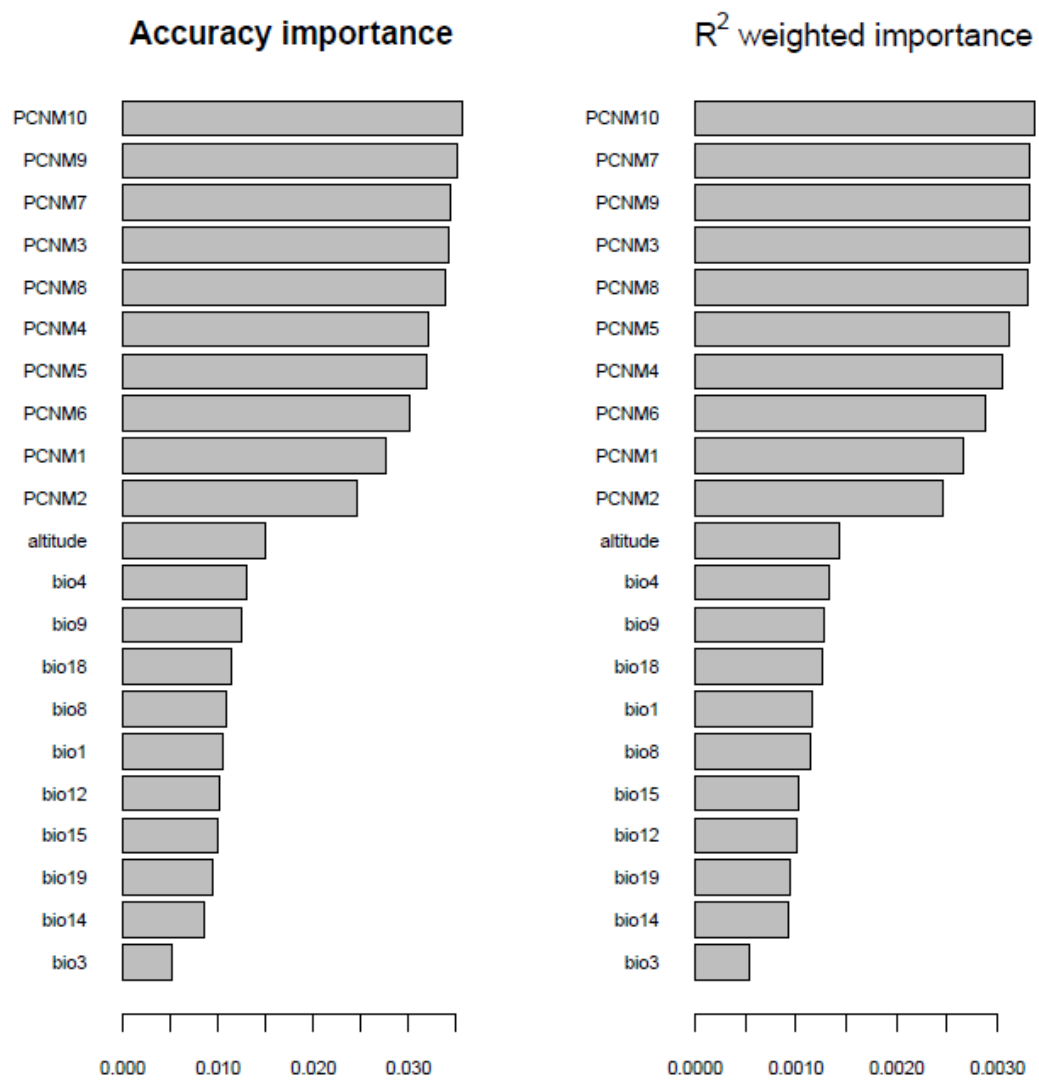
```
library(gradientForest)
env.gf <- cbind(clim.points[,3:13], pcnm.keep)
maxLevel <- log2(0.368*nrow(env.gf)/2)
```

Run the GF [took 12-18 hrs to run on high mem node in R]

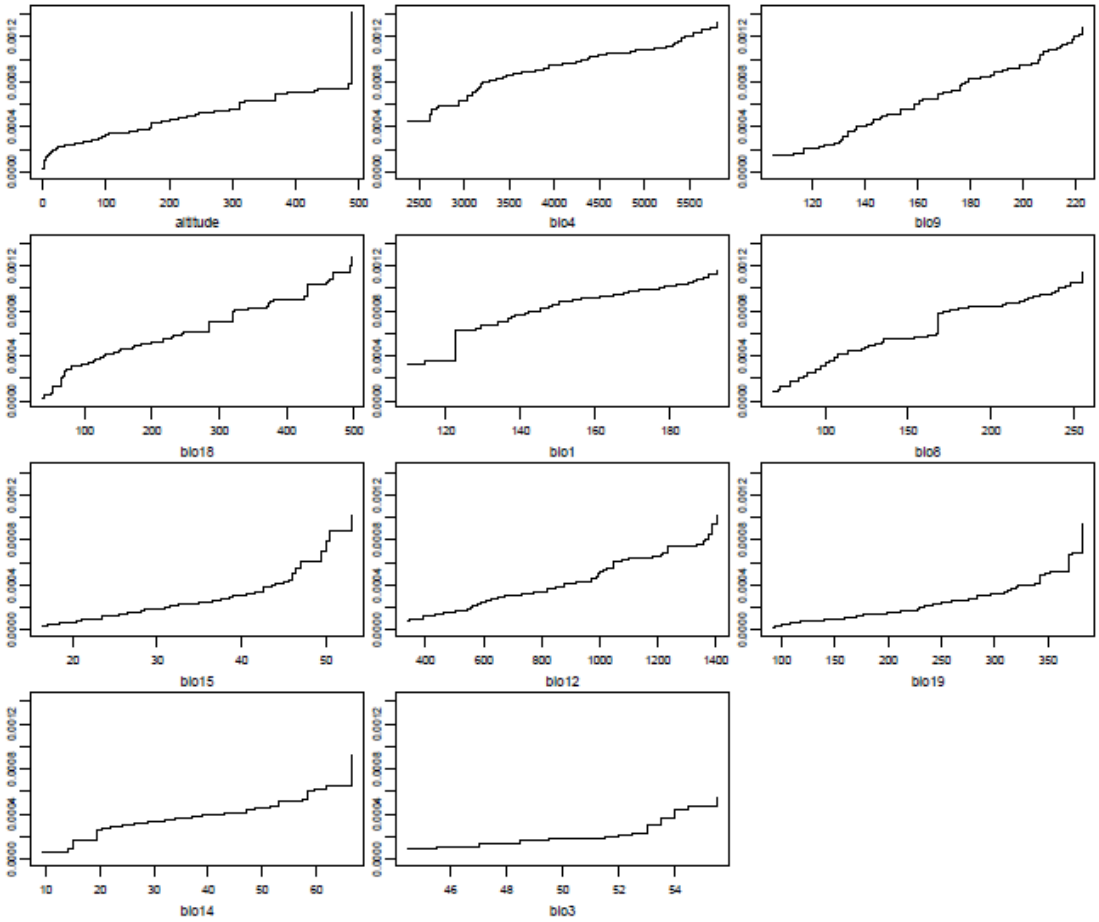
```
gf3 <- gradientForest(cbind(env.gf, starling.snp), predictor.vars=colnames(env.gf),
response.vars=colnames(starling.snp), ntree=500, maxLevel=maxLevel, trace=T, corr.threshold=0.50)
```

```
pdf("Sv6_gradientforest_model3_VariableImportance.pdf")
plot(gf3, plot.type = "O")
```

```
dev.off()
```




```
most_important <- names(importance(gf3))[11:25]
par(mgp = c(2, 0.75, 0))
pdf("Sv6_gradientforest_model3_speccum.pdf")
plot(gf3, plot.type = "C", imp.vars = most_important, show.species = F, common.scale = T, cex.axis = 0.6, cex.lab = 0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5, 0.5, 0), mar = c(2.5, 1, 0.1, 0.5), omi = c(0, 0.3, 0, 0)))
dev.off()
```

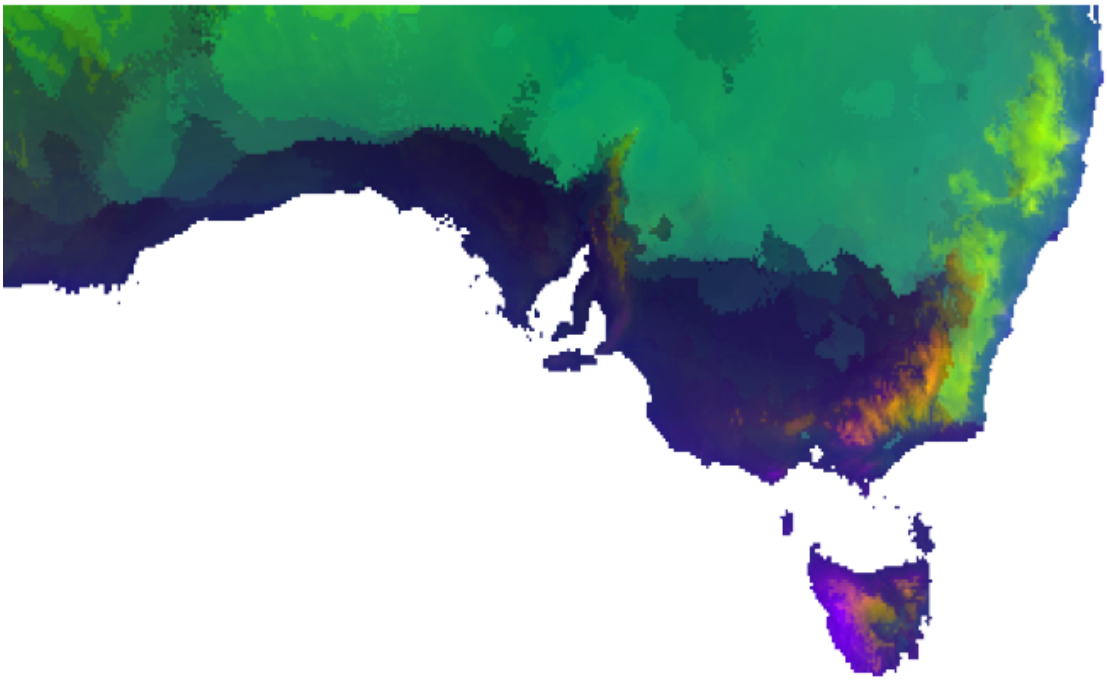



```
most_important <- names(importance(gf3))[9:21]
pdf("Sv6_gradientforest_model3_impdens.pdf")
plot(gf3, plot.type = "S", imp.vars = most_important, leg.posn = "topright", cex.legend = 0.4, cex.axis = 0.6, cex.lab =
0.7, line.ylab = 0.9, par.args = list(mgp = c(1.5,0.5, 0), mar = c(3.1, 1.5, 0.1, 1)))
dev.off()
```

```
extent <- c(120, 155, -44, -27)

values_alt  <- getData('worldclim',download=TRUE,var='alt',res=5)

climdata.subset<- subset(climdata, c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12", "bio14", "bio15",  "bio18",
"bio19"))

merged.data <- addLayer(climdata.subset, values_alt)

names(merged.data)[11]<- "altitude"

clim.layer.crop <- crop(merged.data, extent)

clim.land <- extract(clim.layer.crop, 1:ncell(clim.layer.crop), df = TRUE)
clim.land <- na.omit(clim.land)
```

```
pred <- predict(gf3, clim.land[,-1])
PCs <- prcomp(pred, center=T, scale.=F)
r <- PCs$x[, 1]
g <- PCs$x[, 2]
b <- PCs$x[, 3]
r <- (r - min(r))/(max(r) - min(r)) * 255
g <- (g - min(g))/(max(g) - min(g)) * 255
b <- (b - min(b))/(max(b) - min(b)) * 255
mask<-clim.layer.crop$bio4
mask[]<-as.numeric(mask[]>0)
rastR <- rastG <- rastB <- mask
rastR[clim.land$ID] <- r
rastG[clim.land$ID] <- g
rastB[clim.land$ID] <- b
rgb.rast <- stack(rastR, rastG, rastB)
```

```
pdf("Sv6_gradientforest_model3_Map.pdf")
plotRGB(rgb.rast, bgalpha=0)
#points(sample.coord$Longitude, sample.coord$Latitude)
dev.off()
```



Biplot of the biological space

```
a1 <- PCs$x[, 1]
a2 <- PCs$x[, 2]
a3 <- PCs$x[, 3]
r <- a1 + a2
g <- -a2
b <- a3 + a2 - a1
r <- (r - min(r))/(max(r) - min(r)) * 255
```

```
g <- (g - min(g))/(max(g) - min(g)) * 255
b <- (b - min(b))/(max(b) - min(b)) * 255
```

```
nvs <- dim(PCs$rotation)[1]
vec <- c("bio1", "bio3",  "bio4",  "bio8",  "bio9",  "bio12","bio14",  "bio15",  "bio18",  "bio19","altitude") #picked top from VariableImportance
lv <- length(vec)
vind <- rownames(PCs$rotation) %in% vec
scal <- 40
xrng <- range(PCs$x[, 1], PCs$rotation[, 1]/scal) * 1.1
yrng <- range(PCs$x[, 2], PCs$rotation[, 2]/scal) * 1.1
```

```
pdf("Sv6_gradientforest_model3_biplot.pdf")
plot((PCs$x[, 1:2]), xlim = xrng, ylim = yrng, pch = ".", cex = 4, col = rgb(r, g, b, max = 255), asp = 1,cex.main=1, cex.lab=1, cex.axis=1)
points(PCs$rotation[!vind, 1:2]/scal, pch = "+")
arrows(rep(0, lv), rep(0, lv), PCs$rotation[vec, 1]/scal, PCs$rotation[vec, 2]/scal, length = 0.0625)
jit <- 0.0015
text(PCs$rotation[vec, 1]/scal + jit * sign(PCs$rotation[vec, 1]), PCs$rotation[vec, 2]/scal + jit *
sign(PCs$rotation[vec, 2]), labels = vec, cex = 1.5)
dev.off()


,
```