PHYLOGEOGRAPHIC APPROACH FOR DDRAD-SEQ READS

Jeronymo Dalapicolla

For studies of population genomics and phylogeography we used the STACKS 1.45 pipeline (Catchen *et al.* 2013). We processed the database separately for each species, and STACKS 1.45 processed the raw sequences in 5 steps. In the first step, the *process_radtags* demultiplexed the reads according to the barcodes list (parameter -*b*), with value 2 for the distance allowed between barcodes (--*barcode_dist*). Reads without barcodes, with uncalled nucleotides, with deficient restriction enzyme cut sites, or with low quality scores for Illumina, Phed ≤ 32 (-*E*) were excluded. At the end of this step the high-quality reads had 140 bp (without 10 pb of barcodes) and they were separated into individual files by individuals. The following step, *ustacks*, the reads of each individual were aligned by the *de novo* approach, and arranged in stacks with identical reads by individual. We excluded stacks with lower than 6 reads (-*m*) and we merged the stacks in loci, allowing up to 3 stacks by loci (-*max_locus_stacks*) and 3 nucleotides of distance between stacks (-*M*) (Paris *et al.* 2017). At the end of this stage, we had a set of putative loci with polymorphisms and alleles inferred per those loci. *cstacks*, the third step, grouped the loci across all individuals and create a unique catalog with all loci for each one of the target species. When two loci were grouped, *cstacks* joined their SNPs in the catalog, with 3 fixed differences expected between individuals (-*n*) (Paris *et al.* 2017). In the fourth step, *sstacks,* the loci from each individual were compared to all catalog loci, recording the matches into a new file. Individual loci similar to more than one loci of the catalog were excluded because their information is ambiguous. The *populations* program was the last step and it processed the reads individually using the same catalog-matched data created on the previous step. In this step we splitted individuals into different populations (see Material and Methods for details). Only the loci present in at least two populations in each lineage (-*p*), and with minimum depth of coverage for each loci equal to 6 (-*m*)

were used to create the output in Variant Call Format (VCF). The VCF was edited, eliminating the very variable loci using a script (Appendix 1) in R plataform (R-Development CoreTeam 2018), and removing loci and individuals up to 20% of missing data in PLINK 1.9 (Purcell *et al.* 2007). *populations* program was performed again using the filtered data, and only one SNP per loci randomly chosen (--*write_random_snp*) was used to create the outputs for the subsequent analyzes.

REFERENCES

Catchen, J., Hohenlohe, P.A., Bassham, S., Amores, A. & Cresko, W.A. (2013) Stacks: an analysis tool set for population genomics. *Molecular Ecology* 22, 3124–3140.

Paris, J.R., Stevens, J.R. & Catchen, J.M. (2017) Lost in parameter space: a road map for stacks S. Johnston (Ed). *Methods in Ecology and Evolution* 8, 1360–1373.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D., Maller, J., Sklar, P., de Bakker, P.I.W., Daly, M.J. & Sham, P.C. (2007) PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. *The American Journal of Human Genetics* 81, 559–575.

R-Development CoreTeam (2018) R Development Core Team. *R: A Language and Environment for Statistical Computing* 55, 275–286.

Appendix 1: Script in R created by Andrea Thomaz for Stacks version 1.35 and modified by Sarp Kaya for Stacks version 1.45 or above, and used here to read *.vcf* file from stacks output for: (I) plot the frequency of variable sites per position along all loci; and (ii) calculate theta based on number of segregating sites and individuals to create blacklist to delete very variable loci.

```
require(plyr)
require(pegas)
options(scipen = 999)  # To close scientific numerics because it creates problem in whitelist, if you want to open use options(scipen = 0)

#READ VCF
setwd("") #choose a path
data <- read.table('name_file.vcf', header = FALSE, sep = "\t")
head(data[1:20,1:20])

#SEQUENCE LENGTH
seq_len <- 140 #MODIFY HERE according the length of the sequence (deletes positions at the end of seq, 10 less than original sequences)

#selecting loci ID and position from column $V

#this is for ID column problem in stacks v.1.46
loci_num<-as.numeric(sub('_.*', '', data$V3))
pos2 <- as.numeric(sub('.*_', '', data$V3))

#creates dataframe with loci ID, the variable positions and the number of individuals in each loci
new_data <- data.frame(loci_ID = loci_num,
                       pos_vcf1 = data[,2],
                       pos = pos2,
                       #pos_dea = (data[,2] - seq_len*(loci_num-1))-2,
                       ind = rowSums(data[,10:length(data)] != "./.:0:.,."))

head(new_data)
min(new_data$pos)#should always be position 5 (first positions are the adapters)
```

```
max(new_data$pos)#should always be position 139
length(unique(new_data$loci_ID))#how many loci do I have
length(new_data$loci_ID)

par(mar = rep(2, 4))
#saving graph with frequency of variable sites along the loci
pdf("./SNPdistr_pos140bp.pdf")
hist(new_data$pos, xlim = c(-1,seq_len), breaks = c(seq(-1, seq_len-1, by=1)), xlab = 'Position along the loci', main = 'The position of segregating sites');
abline(2100, 0, col = "red")#helps to find where starts to increase toward the end, last positions have strong increase
abline(v = 131, col = "red")#helps to figure out where to cut off before increase in bad calls
#move the lines around to visualize depending on the case
dev.off()

#BASE ON THE GRAPH, CHOOSE HOW MANY POSITION TO DELETE FROM THE END
to_del <- 11 #how many sites to delete in the end of the sequence
#11 is based on the 130 I chose for the abline above
seq_len_cut <- seq_len - to_del

#create a whitelist to exclude those 11 (to_del) positions
whitelist <- subset(new_data, pos < seq_len_cut)[,c(1,3,4)]
pdf("./SNPdistr_pos_cutto129bp.pdf")
hist(whitelist$pos, xlim = c(0,seq_len_cut), breaks = c(seq(-1, seq_len_cut -1 , by=1)), xlab = 'Position along the loci', main = 'The position of segregating sites');
dev.off()

#calculating theta for all loci
var.sites <- count(whitelist, "loci_ID")
length(var.sites$loci_ID)
max(var.sites$freq)
theta_calc <- merge(unique(whitelist[,-2]), var.sites, by = "loci_ID")
theta_calc$theta <- 0
head(theta_calc)
for (i in 1:length(theta_calc$theta)){
  theta_calc[i,4] <- theta.s(theta_calc$freq[i], theta_calc$ind[i])/seq_len_cut
}
```

```
#calculating the 95% quantile to exclude loci extremely variable
quant <- quantile(theta_calc$theta, probs=0.95) #set the value to be
quant
pdf("./theta129bp.pdf")
hist(theta_calc$theta)
abline(v = quant, col="red")
dev.off()

#what is the maximum number of mutations in a loci
max(theta_calc$freq) #max theta before
x <- subset(theta_calc, theta < quant)
max(x$freq) #max theta after, make sure is realistic for a 140 bp sequence
#think about what mutation rate the spp might have

#saving whitelist for re-run populations in stacks
blacklist <- subset(theta_calc, theta > quant)[,1]
#write.table(blacklist, file="blacklist.txt", sep = '\n', row.names = F, col.names = F)

#removes the blacklist from the whitelist and write off white list
whitelist$blacklist <- match(whitelist$loci_ID, blacklist, nomatch = 0)
whitelist_final <- subset(whitelist, blacklist == 0)
length(unique(whitelist_final$loci_ID)) #number of unique loci, this is the number I need to get out with "write random loci"
length(whitelist_final$loci_ID) #number of snps
write.table(whitelist_final[,1:2], file="whitelist_final.txt", sep = '\t', row.names = F, col.names = F)
```