Poxvirus Host Prediction

Katie Tseng, Dan Becker, Colin Carlson, etc.

Introduction

The following code reproduces the analysis from:

> etc.

Data Preparation

Load required packages and set system

```
##(1) libraries for preparing data for analysis
library(ape)
library(dplyr)
library(nlme)
library(tidyverse)
library(vroom)
## treespace dependencies include XQuartz v2.7.11 (https://www.xquartz.org/releases/XQuartz-2.7.11.html
library(rgl) # >install.packages("rgl"); >options(rgl.useNULL=TRUE)
library(treespace)

## libraries for phylogenetic analysis
library(ape)
library(caper)
library(data.table)
library(BiocManager)  ## BiocManager::install(c("Biostrings","ggtree"))
library(phylofactor)  ## devtools::install_github('reptalex/phylofactor'); more info at: https://reptal
library(treeio)       ## BiocManager::install("treeio")

## libraries for prediction model
library(gbm);
library(ROCR);
library(vegan);
library(plyr);library(dplyr);
library(mvtnorm)
library(xtable)
library(parallel)

#clean environment
rm(list=ls())
graphics.off()
```

```
#set working directory
setwd("~/Library/CloudStorage/OneDrive-WashingtonStateUniversity(email.wsu.edu)/Fernandez Lab/Projects
```

Load raw data

```
#(1) load data
load("Tseng2022.RData")

#(2) poxdata: host-OPV interactions detected via PCR/isolation from Virion database
##virion <- vroom('https://github.com/viralemergence/virion/blob/main/Virion/Virion.csv.gz')
poxdata <- virion %>% filter(VirusGenus == "orthopoxvirus" & (DetectionMethod %in% c("PCR/Sequencing","

#(3) taxa: mammal species taxonomy from vertlife
##vertlife <- read.csv(url('https://data.vertlife.org/mammaltree/taxonomy_mamPhy_5911species.csv'))
taxa <- vertlife

#(4) hostTraits: mammal traits from the COMBINE database <https://doi.org/10.1002/ecy.3344>
##path: ecy3344-sup-0001-datas1.zip > COMBINE_archives > trait_data_imputed.csv)
hostTraits <- combine

#(5) hostTree: mammal phylogeny tree from Dryad, <https://doi.org/10.5061/dryad.tb03d03>
##path: Data_S8_finalFigureFiles > _DATA > MamPhy_fullPosterior_BDvr_Completed_5911sp_topoCons_NDexp_MC
hostTree <- dryad

#(6) viralTraits: OPV accessory genes from ... (Steph to provide refined datatable)
viralTraits <- opvgenes

#(7) clean environment
rm(virion, vertlife, dryad, combine, opvgenes)
```

Aggregate poxdata to genus-level

```
#(1) exclude if host genus or virus is NA; exclude variola (smallpox) virus
poxdata <- poxdata[!is.na(poxdata$HostGenus),]
poxdata <- poxdata[!is.na(poxdata$Virus),]
poxdata <- poxdata[!(poxdata$Virus=="variola virus"),]

#(2) to dis-aggregate West African from Congo Basin MPXV clades, export MPXV interactions
mpxvdata <- poxdata %>% filter(Virus=="monkeypox virus" & (DetectionMethod %in% c("PCR/Sequencing","Isol
#write.csv(mpxvdata, "~/mpxvdata.csv")

#(3) merge clade-specific data
#TBD: Steph to share clade-specific data

#(4) extract PCR-positive data
pcr <- subset(poxdata[which(poxdata$DetectionMethod=="PCR/Sequencing"),], select=c("Host","HostGenus","V
pcr$Host <- ifelse(is.na(pcr$Host),"sp.",pcr$Host)
pcr$pcr <- 1
pcr <- aggregate(.~Host+HostGenus+Virus, data=pcr, sum)
```

```
#(5) extract isolation-positive data
competence <- subset(poxdata[which(poxdata$DetectionMethod=="Isolation/Observation"),], select=c("Host"
competence$Host <- ifelse(is.na(competence$Host),"sp.",competence$Host)
competence$competence <- 1
competence <- aggregate(.~Host+HostGenus+Virus, data=competence, sum)

#(6) merge PCR/isolation-positive data; create binary vars
poxdata <- merge(pcr, competence, by=c("Host","HostGenus","Virus"), all=TRUE)

#(7) create studies variable
poxdata$studies <- ifelse(is.na(poxdata$pcr),0,poxdata$pcr) + ifelse(is.na(poxdata$competence),0,poxdata

#(8) create binary variables for detection via pcr/competence
poxdata$pcr=ifelse(is.na(poxdata$pcr),0,1)
poxdata$competence=ifelse(is.na(poxdata$competence),0,1)

#(9) aggregate at genus level
agg_pcr <- aggregate(pcr~HostGenus+Virus, data=poxdata, max)
agg_competence <- aggregate(competence~HostGenus+Virus, data=poxdata, max)
agg_studies <- aggregate(studies~HostGenus+Virus, data=poxdata, sum)

#(10) merge pcr, competence and studies variables
gdata <- merge(agg_pcr,agg_competence)
gdata <- merge(gdata,agg_studies)

#(11) rename variables
gdata <- rename(gdata,c('HostGenus'='gen','Virus'='virus'))
gdata$gen <- str_to_title(gdata$gen)

#(12) clean environment
rm(poxdata, mpxvdata, pcr,competence,agg_pcr, agg_competence, agg_studies)
```

Merge poxdata with broader mammal taxa to create pseudoabsences

```
#(1) drop duplicate genera in taxa
gtaxa <- taxa[!duplicated(taxa$gen),]
gtaxa <- gtaxa[c('gen','fam','ord')]

#(2) check for mismatched names, then merge gdata with taxa
gdata$gen[!gdata$gen %in% taxa$gen]
data=merge(gtaxa,gdata,by='gen',all.x=TRUE)

#(3) keep only genera from orders in which positive associations exist
keep=subset(data, pcr==1 | competence==1)
data$keep=ifelse(data$ord %in% keep$ord,TRUE,FALSE)
data=subset(data,keep==TRUE)
data$keep=NULL

#(4) create dataframe of all possible host-OPV combinations (for mammal genera that exist in orders w/
uniq_gen <- unique(data$gen[!is.na(data$gen)])
uniq_virus <- unique(data$virus[!is.na(data$virus)])
```

```
combinations <- expand.grid(uniq_gen,uniq_virus)
combinations <- rename(combinations,c('Var1'='gen','Var2'='virus'))

#(5) merge host-OPV interaction data with all possible combinations
data <- merge(combinations,data,all.x=TRUE)

#(6) create binary variable for sampled host-OPV pairs
data$sampled=ifelse(is.na(data$pcr) & is.na(data$competence),0,1)

#(7) reclassify NAs as pseudo-absences for viral detection
data$pcr=ifelse(is.na(data$pcr),0,data$pcr)
data$competence=ifelse(is.na(data$competence),0,data$competence)
data$studies=ifelse(is.na(data$studies),0,data$studies)

#(8) replace NA taxonomic values based on host genera
data=merge(data,gtaxa,by='gen',all.x=TRUE)
data <- rename(data,c('fam.y'='fam','ord.y'='ord'))
data$fam.x=NULL
data$ord.x=NULL

#(9) clean environment
rm(gdata,taxa,gtaxa,keep,uniq_gen,uniq_virus,combinations)
```

Aggregate hostTraits to genus-level

```
#(1) observe variable names
colnames(hostTraits)

#(2) to aggregate continuous/integer variables, use the median as the summary measure
hostTraits_continuous=aggregate(cbind(adult_mass_g,brain_mass_g,adult_body_length_mm,adult_forearm_leng
                                      max_longevity_d,maturity_d,female_maturity_d,male_maturity_d,
                                      age_first_reproduction_d,gestation_length_d,teat_number_n,
                                      litter_size_n,litters_per_year_n,interbirth_interval_d,
                                      neonate_mass_g,weaning_age_d,weaning_mass_g,generation_length_d,
                                      dispersal_km,density_n_km2,home_range_km2,social_group_n,
                                      dphy_invertebrate,dphy_vertebrate,dphy_plant,
                                      det_inv,det_vend,det_vect,det_vfish,det_vunk,det_scav,det_fruit,det_
                                      upper_elevation_m,lower_elevation_m,altitude_breadth_m,habitat_bread
                                ~ order+family+genus, data=hostTraits, FUN=median, na.action=na.pass, na.r
##'na.action=na.pass, na.rm=TRUE' is specified such that if species w/in a genus has a combination of r

#(3) to aggregate binary variables, use the mean as the summary measure
hostTraits$fossoriality[hostTraits$fossoriality==2]<-0  #recode 0/1
hostTraits_binary=aggregate(cbind(hibernation_torpor,fossoriality,freshwater,marine,terrestrial_non.vola
                                  island_dwelling,disected_by_mountains,glaciation) ~ order+family+genus, 

#(4) to aggregate categorical variables transform into binary
hostTraits_cat <- hostTraits
hostTraits_cat$trophic_herbivores <- ifelse(hostTraits_cat$trophic_level==1,1,0)
hostTraits_cat$trophic_omnivores <- ifelse(hostTraits_cat$trophic_level==2,1,0)
hostTraits_cat$trophic_carnivores <- ifelse(hostTraits_cat$trophic_level==3,1,0)
```

```
hostTraits_cat$activity_nocturnal <- ifelse(hostTraits_cat$activity_cycle==1,1,0)
hostTraits_cat$activity_crepuscular <- ifelse(hostTraits_cat$activity_cycle==2,1,0) #nocturnal/crepuscu
hostTraits_cat$activity_diurnal <- ifelse(hostTraits_cat$activity_cycle==3,1,0)
hostTraits_cat$forager_marine <- ifelse(hostTraits_cat$foraging_stratum=="M",1,0)
hostTraits_cat$forager_ground <- ifelse(hostTraits_cat$foraging_stratum=="G",1,0)
hostTraits_cat$forager_scansorial <- ifelse(hostTraits_cat$foraging_stratum=="S",1,0)
hostTraits_cat$forager_arboreal <- ifelse(hostTraits_cat$foraging_stratum=="Ar",1,0)
hostTraits_cat$forager_aerial <- ifelse(hostTraits_cat$foraging_stratum=="A",1,0)
hostTraits_cat$island_end_marine <- ifelse(hostTraits_cat$island_endemicity=="Exclusively marine",1,0)
hostTraits_cat$island_end_mainland <- ifelse(hostTraits_cat$island_endemicity=="Occurs on mainland",1,0
hostTraits_cat$island_end_lgbridge <- ifelse(hostTraits_cat$island_endemicity=="Occurs on large land br
##hostTraits_cat$island_end_smbridge <- ifelse(hostTraits_cat$island_endemicity=="Occurs on small land
hostTraits_cat$island_end_isolated <- ifelse(hostTraits_cat$island_endemicity=="Occurs only on isolated
hostTraits_cat$biogeo_afrotropical <- ifelse(grepl("Afrotropical",hostTraits_cat$biogeographical_realm)
hostTraits_cat$biogeo_antarctic <- ifelse(grepl("Antarctic",hostTraits_cat$biogeographical_realm),1,0)
hostTraits_cat$biogeo_australasian <- ifelse(grepl("Australasian",hostTraits_cat$biogeographical_realm)
hostTraits_cat$biogeo_indomalayan <- ifelse(grepl("Indomalayan",hostTraits_cat$biogeographical_realm),1
hostTraits_cat$biogeo_nearctic <- ifelse(grepl("Nearctic",hostTraits_cat$biogeographical_realm),1,0)
hostTraits_cat$biogeo_neotropical <- ifelse(grepl("Neotropical",hostTraits_cat$biogeographical_realm),1
hostTraits_cat$biogeo_oceanian <- ifelse(grepl("Oceanian",hostTraits_cat$biogeographical_realm),1,0)
hostTraits_cat$biogeo_palearctic <- ifelse(grepl("Palearctic",hostTraits_cat$biogeographical_realm),1,0

#(5) to aggregate transformed categorical-to-binary variables, use the mean as the summary measure
hostTraits_cat=aggregate(cbind(trophic_herbivores,trophic_omnivores,trophic_carnivores,
                               activity_nocturnal,activity_crepuscular,activity_diurnal,
                               forager_marine,forager_ground,forager_scansorial,forager_arboreal,forager_a
                               island_end_marine,island_end_mainland,island_end_lgbridge,island_end_isolat
                               biogeo_afrotropical,biogeo_antarctic,biogeo_australasian,biogeo_indomalayan
                         ~ order+family+genus, data=hostTraits_cat, FUN=mean, na.action=na.pass, na.rm=TRU

#(6) merge continuous variables with binary variables and clean environment
hostTraits <- full_join(hostTraits_continuous, hostTraits_binary, by = c("order","family","genus"),keep=
hostTraits <- rename(hostTraits,c('order.x'='order','family.x'='family','genus.x'='genus'))
hostTraits=subset(hostTraits, select=-c(order.y,family.y,genus.y))

#(7) merge transformed categorical variables and clean environment
hostTraits <- full_join(hostTraits, hostTraits_cat, by = c("order","family","genus"),keep=TRUE)
hostTraits <- rename(hostTraits,c('order.x'='order','family.x'='family','genus.x'='genus'))
hostTraits=subset(hostTraits, select=-c(order.y,family.y,genus.y))

#(8) clean environment
rm(hostTraits_binary,hostTraits_cat,hostTraits_continuous)
```

Collapse hostTree to genus-level

```
#(1) reformat
hostTree$tip.label[hostTree$tip.label=="_Anolis_carolinensis"] <- "Anolis_carolinensis"

#(2) create dataframe linking tip labels with their corresponding categories (genus and species)
tdata <- data.frame(matrix(NA,nrow=length(hostTree$tip.label),ncol=0))
tdata$genus=sapply(strsplit(hostTree$tip.label,'_'),function(x) paste(x[1],sep='_'))
```

```
tdata$species=hostTree$tip.label

#(3) collapse tree to genus level
hostTree=makeCollapsedTree(tree=hostTree,df=tdata[c('genus','species')])

#(4) clean environment
rm(tdata)
```

Prepare poxdata for merging with hostTraits and trimming hostTree

```
#(1) are all poxdata genera in hostTree?
data$gtip <- data$gen
hostTree$gtip <- hostTree$tip.label
data$intree <- ifelse(data$gtip%in%setdiff(data$gtip,hostTree$gtip),'missing','upham')

#(2) are all poxdata genera in hostTraits?
hostTraits$gtip <- hostTraits$genus
data$intraits <- ifelse(data$gtip%in%setdiff(data$gtip,hostTraits$gtip),'missing','traits')

#(3) create dataframe of just observations with mismatched names
fix <- data[c('gtip','intree','intraits')]
fix <- fix[fix$intree=='missing'|fix$intraits=='missing',]
fix <- unique(fix)

#(4) identify homotypic synonyms or proxy species via IUCN (https://www.iucnredlist.org/) and NCBI (htt
fix$treename <- NA
fix$traitname <- NA
fix$proxy <- NA
fix$proxy <- ifelse(fix$gtip=="Calassomys","Delomys",fix$proxy)
  ##source: https://academic.oup.com/jmammal/article/95/2/201/860032
fix$traitname <- ifelse(fix$gtip=="Liomys","Heteromys",fix$traitname)
  ##source: https://www.iucnredlist.org/species/40768/22345036
fix$traitname <- ifelse(fix$gtip=="Oreonax","Lagothrix",fix$traitname)
  ##source: https://www.iucnredlist.org/species/39924/192307818
fix$traitname <- ifelse(fix$gtip=="Paralomys","Phyllotis",fix$traitname)
  ##source: https://www.iucnredlist.org/species/17226/22333354
fix$traitname <- ifelse(fix$gtip=="Pearsonomys","Geoxus",fix$traitname)
  ##source: https://www.iucnredlist.org/species/40768/22345036
fix$traitname <- ifelse(fix$gtip=="Pipanacoctomys","Tympanoctomys",fix$traitname)
  ##source: https://www.iucnredlist.org/species/136557/78324400#taxonomy
fix$traitname <- ifelse(fix$gtip=="Pseudalopex","Lycalopex",fix$traitname)
  ##source: https://www.iucnredlist.org/species/6926/87695615
## hostTraits$genus[which(grepl('Tympanoctomys',hostTraits$genus))]

#(5) merge revised names with poxdata
fix <- subset(fix, select=-c(intree,intraits))
data <- merge(data,fix,by='gtip',all.x=T)

#(5) treename will be used for merging data & hostTree
data$treename <- ifelse(data$treename=='',NA,as.character(data$treename))
data$treename <- ifelse(is.na(data$treename),as.character(data$gtip),as.character(data$treename))
```

```
#(6) traitname will be used for merging data & hostTraits
data$traitname <- ifelse(data$traitname=='',NA,as.character(data$traitname))
data$traitname <- ifelse(data$intraits=='missing' & is.na(data$traitname),as.character(data$proxy),
                      ifelse(data$intraits=='missing' & !is.na(data$traitname),as.character(data$traitna
                             as.character(data$gtip)))

#(7) simplify and clean environment
data <- subset(data, select=-c(intree,intraits,proxy))
rm(fix)
```

Merge poxdata with hostTraits and trim hostTree to mirror poxdata

```
#(1) check poxdata for NAs
which(is.na(data))

#(2) merge traits with poxdata
hostTraits$traitname <- hostTraits$gtip
data <- merge(data,hostTraits,by=c('traitname'),all.x=T)

#(3) trim data
data <- rename(data,c('gtip.x'='gtip'))
data <- subset(data,select=-c(order, family, genus,gtip.y))

#(4) trim hostTree
hostTree <- keep.tip(hostTree,hostTree$tip.label[hostTree$tip.label%in%data$treename])
hostTree$gtip <- NULL
hostTree=makeLabel(hostTree)

#(6) clean environment
rm(hostTraits)
```

Merge poxdata with viral accessory genes

```
#(1) simplify data
viralTraits <- head(viralTraits, -2)

#(2) rename column names
viralTraits <- viralTraits[,-2]
colnames(viralTraits) <- paste("ag" ,colnames(viralTraits),sep="_")
names(viralTraits)[1] <- c("virus")

#(3) to assess variation in viralTraits, create mode function
mode.prop <- function(x) {
  ux <- unique(x[is.na(x)==FALSE])        # creates array of unique values
  tab <- tabulate(match(na.omit(x), ux))  # creates array of the frequency a unique value appears in a
  max(tab)/length(x[is.na(x)==FALSE])     # max-frequency / number of elements in each column that are
}

#(4) assess variation across columns (2 indicates columns)
```

```
vars=data.frame(apply(viralTraits,2,function(x) mode.prop(x)),
                apply(viralTraits,2,function(x) length(unique(x))))    # number of unique elements in e
vars$variables=rownames(vars)
colnames(vars) <- c("var","uniq","column")

## trim
#vars <- vars[-c(1,2), ]

#(5) drop variables with no variation
vars <- subset(vars,vars$var<1)

# ## visualize distribution of NA
# png("/Users/katietseng/Downloads/virus_ag_variation.png", width=4,height=4,units="in",res=600)
# ggplot(vars,
#        aes(var))+
#   geom_histogram(bins=50)+
#   geom_vline(xintercept=0.70,linetype=2,size=0.5)+
#   theme_bw()+
#   theme(panel.grid.major=element_blank(),panel.grid.minor=element_blank())+
#   theme(axis.title.x=element_text(margin=margin(t=10,r=0,b=0,l=0)))+
#   theme(axis.title.y=element_text(margin=margin(t=0,r=10,b=0,l=0)))+
#   labs(y="frequency",
#        x="trait coverage across viral species")+
#   scale_x_continuous(labels=scales::percent)
# dev.off()

# ## drop based on threshold
# vars$keep=ifelse(vars$var>=0.7,"keep","cut")
# keeps=vars[-which(vars$keep=="cut"),]$column
# keeps <- append("virus",keeps)
# viralTraits=viralTraits[keeps]

#(6) merge with poxdata
data <- merge(data,viralTraits,by=c('virus'),all.x=TRUE)

#(7) clean environment
rm(viralTraits,vars,keeps,original_cols,mode.prop)

# ## identify rows with duplicate values (i.e., hosts with identical presence/absence of accessory gene
# which(duplicated(viralTraits[,-c(1)])| duplicated(viralTraits[,-c(1)], fromLast = TRUE))
# viralTraits$dup <- duplicated(viralTraits[,-c(1)])
```

Combine PubMed citations and evolutionary distinctiveness measure

```
#(1) load library for PubMed citations
library(easyPubMed)

#(2) create function to count citations
counter=function(name){
  as.numeric(as.character(get_pubmed_ids(gsub('_','-',name))$Count))
}
```

```
citations=c()

#(3) extract unique genera from poxdata
treename <- unique(data$treename)

#(4) apply counter function while looping through treenames
for(i in 1:length(treename)) {
  citations[i]=counter(treename[i])
  print(i)
}

#(5) compile citation numbers
cites <- data.frame(treename=treename,cites=citations)

#(6) merge cites with poxdata
data <- merge(data,cites,by='treename')

#(7) load library for evolutionary distinctiveness (ed) measure
library(picante)  #before loading picante, make sure latest version of nlme package is loaded
ed <- evol.distinct(hostTree,type='equal.splits') #calculates ed measures for a suite of species by equ

#(8) rename variables in ed
ed <- rename(ed,c('Species'='treename','w'='ed_equal'))

#(9) merge ed with poxdata
data <- merge(data,ed,by='treename')

#(10) clean environment
rm(cites,ed,citations,i,treename,counter)

## consider adding viral genome length, viral richness (number of virus detected in each genera), and h
```

Phylogenetic analysis

```
# 02_phylofactor.R
```

Exploratory boosted regression tree model

```
# 03_brt.R
```

Principal components analysis of viral accessory genes

```
library(ape)
library(vegan)
```

```r
#(1) load data of viral accessory genes
load("Tseng2022.RData")
genes <- opvgenes

#(2) trim and reformat
genes <- head(genes, -2)
genes <- genes[,-2]
colnames(genes) <- paste("ag" ,colnames(genes),sep="_")
names(genes)[1] <- c("virus")

#(3) create distance matrix: returns a measure of the pairwise similarity between each virus based on w
genmat <- as.matrix(genes[,-1])
rownames(genmat) <- genes[,1] %>% pull()
class(genmat) <- "numeric"
gendist <- vegdist(genmat)

#(4) principal coordinates analysis
genpca <- pcoa(gendist) #{ape}
#genpca <- prcomp(gendist)

#(4) principal components analysis
#genpca <- princomp(genmat) #{stats}

#(5) let's explore
genpca
genpca$vectors

#(6) plot coordinate pairs and eigenvectors
biplot(genpca)
```