**Supplementary data file 11.** *SD11\_bionut.evolCorMat.R*. Script for evolutionary correlations

rm(list=ls())

library(phytools)

library(geiger)

#FUNCTION TO REMOVE SPECIES WITH MISSING DATA

removeNA<-function(dat){

missing.list<-sapply(1:ncol(dat), function(x) as.numeric(which(is.na(dat[,x])==TRUE)))

missing<-missing.list[[1]]

for(i in 2:length(missing.list)) {missing<-c(missing, missing.list[[i]][which(match(missing.list[[i]], missing, nomatch=0)==0)])}

dat<-dat[-missing,]

return(dat)

}

#GET DATA AND TREE

fullTree<-read.newick("Rabosky\_etal2014.timetree.nex")

fullTree$tip.label[grep("Acanthopagrus\_schlegelii",fullTree$tip.label)]<-"Acanthopagrus\_schlegelii"

fullTree$tip.label[grep("Auxis\_thazard",fullTree$tip.label)]<-"Auxis\_thazard"

fullTree$tip.label[grep("Auxis\_rochei",fullTree$tip.label)]<-"Auxis\_rochei"

fullTree$tip.label[grep("Carassius\_auratus",fullTree$tip.label)]<-"Carassius\_auratus"

fullTree$tip.label[grep("Diagramma\_picta",fullTree$tip.label)]<-"Diagramma\_pictum"

fullTree$tip.label[grep("Helicolenus\_dactylopterus",fullTree$tip.label)]<-"Helicolenus\_dactylopterus"

fullTree$tip.label[grep("Oncorhynchus\_masou",fullTree$tip.label)]<-"Oncorhynchus\_masou"

fullTree$tip.label[grep("Osmerus\_mordax\_mordax",fullTree$tip.label)]<-"Osmerus\_mordax"

fullTree$tip.label[grep("Salvelinus\_alpinus",fullTree$tip.label)]<-"Salvelinus\_alpinus"

fullTree$tip.label[grep("Salvelinus\_leucomaenis\_leucomaenis",fullTree$tip.label)]<-"Salvelinus\_leucomaenis"

fullTree$tip.label[grep("Sebastes\_pachycephalus\_pachycephalus",fullTree$tip.label)]<-"Sebastes\_pachycephalus"

fullTree$tip.label[grep("Thymallus\_arcticus",fullTree$tip.label)]<-"Thymallus\_arcticus"

fullDat<-read.csv("bionut\_working.csv")

rownames(fullDat)<-fullDat[,1]

fullDat<-fullDat[,-1]

datSp<-rownames(fullDat)

species<-datSp[which(match(datSp, fullTree$tip.label, nomatch=0)!=0)]

exclude<-fullTree$tip.label[which(match(fullTree$tip.label, datSp, nomatch=0)==0)]

#NUTRIENT DATA

#nutDat<-fullDat[species,c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

nutDat<-fullDat[species,c("protein", "fat", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

logNut<-log((nutDat + 1))

nut<-removeNA(logNut)

tree1<-drop.tip(fullTree, which(match(fullTree$tip.label, rownames(nut), nomatch=0)==0))

#LIFE HISTORY DATA

lifeHistDat<-fullDat[species, c("depmin", "depmax", "maxlen", "a\_lw", "b\_lw", "troph")]

logLife<-log(lifeHistDat + 1)

life<-removeNA(logLife)

tree2<-drop.tip(fullTree, which(match(fullTree$tip.label, rownames(life), nomatch=0)==0))

#NUTRIENT AND LIFE HISTORY DATA COMBINED

logCombined<-data.frame(logLife, logNut)

combined<-removeNA(logCombined)

tree3<-drop.tip(fullTree, which(match(fullTree$tip.label, rownames(combined), nomatch=0)==0))

#EVOLUTIONARY CORRELATIONS AMONG NUTRIENT VARIABLES

lambda1<-phyl.pca(tree1, nut, method="lambda", mode="corr")$lambda

lambdaTree1<-rescale(tree1, model="lambda", lambda=lambda1)

eVCV1<-ratematrix(lambdaTree1, nut)

eCorMat1<-matrix(,nrow(eVCV1),ncol(eVCV1), dimnames=dimnames(eVCV1))

for (i in 1:nrow(eVCV1)){ for (j in 1:ncol(eVCV1)){eCorMat1[i,j]<-eVCV1[i,j] / sqrt(eVCV1[i,i] \* eVCV1[j,j])}}

eCorMat1<-round(eCorMat1, 3)

write.csv(eCorMat1, file="bionut.evolCorrel.csv")

p1<-matrix(,nrow(eCorMat1), ncol(eCorMat1), dimnames=dimnames(eCorMat1))

for (i in 1:nrow(eCorMat1)){

for (j in 1:ncol(eCorMat1)){

t<-eCorMat1[i,j] / sqrt((1-eCorMat1[i,j]^2) / (nrow(nut)-2))

p1[i,j]<-dt(t, nrow(nut)-2)

}

}

p1<-round(p1, 4)

write.csv(p1, file="bionut.evolCorrel\_pvalues.csv")

#EVOLUTIONARY CORRELATIONS AMONG LIFE HISTORY VARIABLES

lambda2<-phyl.pca(tree2, life, method="lambda", mode="corr")$lambda

lambdaTree2<-rescale(tree2, model="lambda", lambda=lambda2)

eVCV2<-ratematrix(lambdaTree2, life)

eCorMat2<-matrix(,nrow(eVCV2),ncol(eVCV2), dimnames=dimnames(eVCV2))

for (i in 1:nrow(eVCV2)){for (j in 1:ncol(eVCV2)){eCorMat2[i,j]<-eVCV2[i,j] / sqrt(eVCV2[i,i] \* eVCV2[j,j])}}

eCorMat2<-round(eCorMat2, 3)

write.csv(eCorMat2, file="lifeHist.evolCorrel.csv")

p2<-matrix(,nrow(eCorMat2), ncol(eCorMat2), dimnames=dimnames(eCorMat2))

for (i in 1:nrow(eCorMat2)){

for (j in 1:ncol(eCorMat2)){

t<-eCorMat2[i,j] / sqrt((1-eCorMat2[i,j]^2) / (nrow(life)-2))

p2[i,j]<-dt(t, nrow(nut)-2)

}

}

p2<-round(p2, 4)

write.csv(p2, file="lifeHist.evolCorrel\_pvalues.csv")

#EVOLUTIONARY CORRELATIONS BETWEEN LIFE HISTORY AND NUTRIENT VARIABLES

lambda3<-phyl.pca(tree3, combined, method="lambda", mode="corr")$lambda

lambdaTree3<-rescale(tree3, model="lambda", lambda=lambda3)

eVCV3<-ratematrix(lambdaTree3, combined)

eCorMat3<-matrix(,nrow(eVCV3),ncol(eVCV3), dimnames=dimnames(eVCV3))

for (i in 1:nrow(eVCV3)){for (j in 1:ncol(eVCV3)){eCorMat3[i,j]<-eVCV3[i,j] / sqrt(eVCV3[i,i] \* eVCV3[j,j])}}

eCorMat3<-round(eCorMat3, 3)

write.csv(eCorMat3, file="bionut\_lifeHist.evolCorrel.csv")

p3<-matrix(,nrow(eCorMat3), ncol(eCorMat3), dimnames=dimnames(eCorMat3))

for (i in 1:nrow(eCorMat3)){

for (j in 1:ncol(eCorMat3)){

t<-eCorMat3[i,j] / sqrt((1-eCorMat3[i,j]^2) / (nrow(combined)-2))

p3[i,j]<-dt(t, nrow(combined)-2)

}

}

p3<-round(p3, 4)

write.csv(p3, file="bionut\_lifeHist.evolCorrel\_pvalues.csv")

**Supplementary data file 12.** *SD12\_bionut.PGLS.R.* Script for phylogenetic least squares regression.

rm(list=ls())

library(phytools)

library(geiger)

library(phylolm)

#FUNCTION TO REMOVE SPECIES WITH MISSING DATA

removeNA<-function(dat){

missing.list<-sapply(1:ncol(dat), function(x) as.numeric(which(is.na(dat[,x])==TRUE)))

missing<-missing.list[[1]]

for(i in 2:length(missing.list)) {missing<-c(missing, missing.list[[i]][which(match(missing.list[[i]], missing, nomatch=0)==0)])}

dat<-dat[-missing,]

return(dat)

}

#GET DATA AND TREE

fullTree<-read.newick("Rabosky\_etal2014.timetree.nex")

fullTree$tip.label[grep("Acanthopagrus\_schlegelii",fullTree$tip.label)]<-"Acanthopagrus\_schlegelii"

fullTree$tip.label[grep("Auxis\_thazard",fullTree$tip.label)]<-"Auxis\_thazard"

fullTree$tip.label[grep("Auxis\_rochei",fullTree$tip.label)]<-"Auxis\_rochei"

fullTree$tip.label[grep("Carassius\_auratus",fullTree$tip.label)]<-"Carassius\_auratus"

fullTree$tip.label[grep("Diagramma\_picta",fullTree$tip.label)]<-"Diagramma\_pictum"

fullTree$tip.label[grep("Helicolenus\_dactylopterus",fullTree$tip.label)]<-"Helicolenus\_dactylopterus"

fullTree$tip.label[grep("Oncorhynchus\_masou",fullTree$tip.label)]<-"Oncorhynchus\_masou"

fullTree$tip.label[grep("Osmerus\_mordax\_mordax",fullTree$tip.label)]<-"Osmerus\_mordax"

fullTree$tip.label[grep("Salvelinus\_alpinus",fullTree$tip.label)]<-"Salvelinus\_alpinus"

fullTree$tip.label[grep("Salvelinus\_leucomaenis\_leucomaenis",fullTree$tip.label)]<-"Salvelinus\_leucomaenis"

fullTree$tip.label[grep("Sebastes\_pachycephalus\_pachycephalus",fullTree$tip.label)]<-"Sebastes\_pachycephalus"

fullTree$tip.label[grep("Thymallus\_arcticus",fullTree$tip.label)]<-"Thymallus\_arcticus"

fullDat<-read.csv("bionut\_working.csv")

rownames(fullDat)<-fullDat[,1]

fullDat<-fullDat[,-1]

datSp<-rownames(fullDat)

species<-datSp[which(match(datSp, fullTree$tip.label, nomatch=0)!=0)]

exclude<-fullTree$tip.label[which(match(fullTree$tip.label, datSp, nomatch=0)==0)]

tree<-drop.tip(fullTree, exclude)

nutDat<-fullDat[species,c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

nutDat<-data.frame(nutDat[,"protein"], sapply(c("fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

colnames(nutDat)[1]<-"protein"

lifeHistDat<-fullDat[species, c("maxlen", "troph", "depmax")]

nutList<-lapply(1:ncol(nutDat), function(x) data.frame(lifeHistDat[which(is.na(nutDat[,x])==FALSE),], nutDat[which(is.na(nutDat[,x])==FALSE),x]))

names(nutList)<-colnames(nutDat)

for (i in 1:length(nutList)) {colnames(nutList[[i]])[ncol(nutList[[i]])]<-names(nutList)[i]}

for (i in 1:length(nutList)) {

for (j in 1:ncol(nutList[[i]])){

nutList[[i]]<-nutList[[i]][which(is.na(nutList[[i]][,j])==FALSE),]

}

}

treeList<-lapply(1:length(nutList), function(x) drop.tip(tree, tree$tip.label[which(match(tree$tip.label, rownames(nutList[[x]]), nomatch=0)==0)]))

#ESTIMATING REGRESSION COEFFICIENTS USING PGLS [BAPU HAS ALREADY DONE THIS]

pgls.table<-matrix(, length(nutList), 18, dimnames=list(names(nutList), c("n", "lambda", "b0", "b1\_maxlen", "b1\_troph", "b1\_depmax", "se\_b0", "se\_b1\_maxlen", "se\_b1\_troph", "se\_b1\_depmax", "tstat\_b0", "tstat\_maxlen", "tstat\_troph", "tstat\_depmax", "P\_b0", "P\_maxlen", "P\_troph", "P\_depmax")))

fit.table<-matrix(, length(nutList), 9, dimnames=list(names(nutList), c("lnL\_lmLambda", "lnL\_lambda", "lnL\_white", "AICc\_lmLambda", "AICc\_lambda", "AICc\_white", "dAICc\_lmLambda", "dAICc\_lambda", "dAICc\_white")))

lrt<-matrix(, length(nutList), 4, dimnames=list(names(nutList), c("lnL\_lmLambda", "lnL\_white", "-2lnLR", "P\_Chi-sq\_4df")))

lrt2<-matrix(, length(nutList), 4, dimnames=list(names(nutList), c("lnL\_lmLambda", "lnL\_lambda", "-2lnLR", "P\_Chi-sq\_3df")))

#coeffList<-list()

for (i in 1:length(nutList)){

maxlen<-nutList[[i]]$maxlen

names(maxlen)<-rownames(nutList[[i]])

troph<-nutList[[i]]$troph

names(troph)<-rownames(nutList[[i]])

depmax<-nutList[[i]]$depmax

names(depmax)<-rownames(nutList[[i]])

nut<-nutList[[i]][,4]

names(nut)<-rownames(nutList[[i]])

phylo<-treeList[[i]]

lmLambda<-phylolm(nut ~ maxlen + troph + depmax, phy=phylo, model="lambda")

lmBrown<-phylolm(nut ~ maxlen + troph + depmax, phy=phylo, model="BM")

lambda<-fitContinuous(phy=phylo, dat=nut, model="lambda")$opt

brown<-fitContinuous(phy=phylo, dat=nut, model="BM")$opt

white<-fitContinuous(phy=phylo, dat=nut, model="white")$opt

fit.table[i,1:3]<-c(lmLambda$logLik, lambda$lnL, white$lnL)

k<-c(6, 3, 2)

names(k)<-c("lmLambda", "lambda", "white")

aic\_lmLambda<-(2 \* k["lmLambda"]) - (2 \* lmLambda$logLik) + (((2\* k["lmLambda"]) \* (k["lmLambda"] + 1)) / (length(nut) - k["lmLambda"] - 1))

aic\_lambda<-(2 \* k["lambda"]) - (2 \* lambda$lnL) + (((2\* k["lambda"]) \* (k["lambda"] + 1)) / (length(nut) - k["lambda"] - 1))

aic\_white<-(2 \* k["white"]) - (2 \* white$lnL) + (((2\* k["white"]) \* (k["white"] + 1)) / (length(nut) - k["white"] - 1))

fit.table[i,4:6]<-c(aic\_lmLambda, aic\_lambda, aic\_white)

fit.table[i, 7:9]<-sapply(c(4:6), function(x) fit.table[i,x] - min(fit.table[i,4:6]))

lrt[i,1:2]<-c(lmLambda$logLik, white$lnL)

lrt[i,3]<-(-2) \* (lrt[i,2] - lrt[i, 1])

lrt[i,4]<-dchisq(lrt[i,3], df=4)

lrt2[i,1:2]<-c(lmLambda$logLik, lambda$lnL)

lrt2[i,3]<-(-2) \* (lrt2[i,2] - lrt2[i,1])

lrt2[i,4]<-dchisq(lrt2[i,3], df=3)

pgls.table[i,1]<-nrow(nutList[[i]])

pgls.table[i,2]<-round(lmLambda[[3]], 3)

pgls.table[i,3:6]<-round(summary(lmLambda)[[2]][,"Estimate"], 5)

pgls.table[i,7:10]<-round(summary(lmLambda)[[2]][,"StdErr"], 5)

pgls.table[i,11:14]<-round(summary(lmLambda)[[2]][,"t.value"], 2)

pgls.table[i,15:18]<-round(summary(lmLambda)[[2]][,"p.value"], 5)

# coeffList[[i]]<-c(lmLambda[[3]], summary(lmLambda)[[2]][,"Estimate"])

write.csv(pgls.table, file="nutPGLS\_table.csv")

write.csv(fit.table, file="nutPGLS\_fitTable.csv")

write.csv(lrt, file="nutPGLS\_LRT.csv")

write.csv(lrt, file="nutPGLS\_LRT2.csv")

}

**Supplementary data file 13.** *SD13\_phylosigByVar.R.* Script for estimating phylogenetic signal of nutrient variables.

rm(list=ls())

library(phytools)

library(geiger)

library(phylolm)

#FUNCTION TO REMOVE SPECIES WITH MISSING DATA

removeNA<-function(dat){

missing.list<-sapply(1:ncol(dat), function(x) as.numeric(which(is.na(dat[,x])==TRUE)))

missing<-missing.list[[1]]

for(i in 2:length(missing.list)) {missing<-c(missing, missing.list[[i]][which(match(missing.list[[i]], missing, nomatch=0)==0)])}

dat<-dat[-missing,]

return(dat)

}

#GET DATA AND TREE

fullTree<-read.newick("Rabosky\_etal2014.timetree.nex")

fullTree$tip.label[grep("Acanthopagrus\_schlegelii",fullTree$tip.label)]<-"Acanthopagrus\_schlegelii"

fullTree$tip.label[grep("Auxis\_thazard",fullTree$tip.label)]<-"Auxis\_thazard"

fullTree$tip.label[grep("Auxis\_rochei",fullTree$tip.label)]<-"Auxis\_rochei"

fullTree$tip.label[grep("Carassius\_auratus",fullTree$tip.label)]<-"Carassius\_auratus"

fullTree$tip.label[grep("Diagramma\_picta",fullTree$tip.label)]<-"Diagramma\_pictum"

fullTree$tip.label[grep("Helicolenus\_dactylopterus",fullTree$tip.label)]<-"Helicolenus\_dactylopterus"

fullTree$tip.label[grep("Oncorhynchus\_masou",fullTree$tip.label)]<-"Oncorhynchus\_masou"

fullTree$tip.label[grep("Osmerus\_mordax\_mordax",fullTree$tip.label)]<-"Osmerus\_mordax"

fullTree$tip.label[grep("Salvelinus\_alpinus",fullTree$tip.label)]<-"Salvelinus\_alpinus"

fullTree$tip.label[grep("Salvelinus\_leucomaenis\_leucomaenis",fullTree$tip.label)]<-"Salvelinus\_leucomaenis"

fullTree$tip.label[grep("Sebastes\_pachycephalus\_pachycephalus",fullTree$tip.label)]<-"Sebastes\_pachycephalus"

fullTree$tip.label[grep("Thymallus\_arcticus",fullTree$tip.label)]<-"Thymallus\_arcticus"

fullDat<-read.csv("bionut\_working.csv")

rownames(fullDat)<-fullDat[,1]

fullDat<-fullDat[,-1]

datSp<-rownames(fullDat)

species<-datSp[which(match(datSp, fullTree$tip.label, nomatch=0)!=0)]

exclude<-fullTree$tip.label[which(match(fullTree$tip.label, datSp, nomatch=0)==0)]

tree<-drop.tip(fullTree, exclude)

nutDat<-fullDat[species,c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

nut<-data.frame(nutDat[,"protein"], sapply(c("fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

nut<-data.frame(sapply(c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

rownames(nut)<-rownames(nutDat)

colnames(nut)[1]<-"protein"

nutList<-lapply(1:ncol(nut), function(x) nut[which(is.na(nut[,x])==FALSE),x])

names(nutList)<-colnames(nut)

for(i in 1:length(nutList)){names(nutList[[i]])<-rownames(nut)[which(is.na(nut[,i])==FALSE)]}

nutList.n<-sapply(1:length(nutList), function(x) length(nutList[[x]]))

treeList<-lapply(1:length(nutList), function(x) drop.tip(tree, which(match(tree$tip.label, names(nutList[[x]]), nomatch=0)==0)))

phylosig2<-function(tree, X){

lambdaFit<-fitContinuous(tree, X, model="lambda")$opt

lambda<-round(lambdaFit$lambda, 5)

sigsq<-round(lambdaFit$sigsq, 5)

logL<-round(lambdaFit$lnL, 3)

starTree<-rescale(tree, model="lambda", lambda=0)

lambda0Fit<-fitContinuous(starTree, X, model="BM")$opt

logL0<-round(lambda0Fit$lnL, 3)

pLRT<-dchisq((-2\*(logL0 - logL)), 1)

pLRT<-round(pLRT, 4)

result<-list(lambda, sigsq, logL, logL0, pLRT)

names(result)<-c("lambda", "sigsq", "logL", "logL0", "P\_LRT")

return(result)

}

phylosig.result<-lapply(1:length(nutList), function(x) phylosig2(treeList[[x]], nutList[[x]]))

#phylosig.result<-lapply(1:length(nutList), function(x) phylosig(treeList[[x]], nutList[[x]], method="lambda", test=TRUE))

names(phylosig.result)<-names(nutList)

#modelcomp<-function(k, n, loglik){

# aic<-sapply(1:length(loglik), function(x) (2\*k[x]) - (2\*loglik[x]))

# aicc<-sapply(1:length(aic), function(x) aic[x] + (((2\*k[x]) \* (k[x] +1)) / (n - k[x] -1)))

# aicc<-round(aicc, 3)

# daicc<-sapply(1:length(aicc), function(x) aicc[x] - min(aicc))

# daicc<-round(daicc, 3)

# support<-sapply(1:length(daicc), function(x) exp(-0.5 \* daicc[x]))

# aicw<-sapply(1:length(support), function(x) support[x] / sum(support))

# aicw<-round(aicw, 3)

# result<-list(aicc, daicc, aicw)

# names(result)<-c("AICc", "dAICc", "AICw")

# return(result)

#}

#lambda.modelcomp<-lapply(1:length(phylosig.result), function(x) modelcomp(c(2,1), length(nutList[[x]]), c(phylosig.result[[x]]$logL, phylosig.result[[x]]$logL0)))

# names(lambda.modelcomp)<-names(nutList)

#lambdaConfInt.PBoot<-function(tree, data, nsim){

#FIT LAMBDA MODEL TO DATA AND TREE

# fitResult<-fitContinuous(tree, data, model="lambda")

# lambda<-fitResult$opt$lambda

#SIMULATE SPECIES DATA UNDER EMPIRICALLY DERIVED PARAMETER VALUES

# lambdaTree<-rescale(tree, model="lambda", lambda=lambda)

# pseudoDat<-sim.char(lambdaTree, par=fitResult$opt$sigsq , nsim=nsim, model="BM")

#FIT MODEL TO PSEUDOREPLICATES OF SPECIES DATA

# lambdaDist<-sapply(1:dim(pseudoDat)[3], function(x) fitContinuous(tree, pseudoDat[,,x], model="lambda")$opt$lambda)

# lambdaDist<-round(as.numeric(lambdaDist), 4)

# lambdaCI<-c(lambdaDist[order(lambdaDist)][round(0.025\*nsim, 0)], lambdaDist[order(lambdaDist)][round(0.975\*nsim, 0)])

# names(lambdaCI)<-c("LCL\_lambda", "UCL\_lambda")

# result<-list(lambda, lambdaCI)

# names(result)<-c("lambda", "CI\_lambda")

# return(result)

#}

#lambdaCIsByVar<-lapply(1:length(nutList), function(x) lambdaConfInt.PBoot(treeList[[x]], nutList[[x]], nsim=100))

# names(lambdaCIsByVar)<-names(nutList)

#phylosig.table<-matrix(,length(phylosig.result), 4+length(phylosig.result[[1]]), dimnames=list(c(1:length(phylosig.result)), c("Variable", "N", "lambda", "LCL\_lambda", "UCL\_lambda", "logL(lambda)", "logL(BM)", "P\_value(LRT)")))

# phylosig.table[,1]<-names(phylosig.result)

# phylosig.table[,2]<-nutList.n

# for (i in 1:length(phylosig.result)) {

# phylosig.table[i,3]<-round(as.numeric(phylosig.result[[i]]$lambda),4)

# phylosig.table[i,4:5]<-round(as.numeric(lambdaCIsByVar[[i]]$CI\_lambda),4)

# phylosig.table[i,6:ncol(phylosig.table)]<-round(as.numeric(phylosig.result[[i]][c("logL", "logL0", "P")]),4)

# }

# phylosig.table<-data.frame(phylosig.table)

phylosig.table<-matrix(,length(phylosig.result), 1+length(phylosig.result[[1]]), dimnames=list(names(phylosig.result), c("N", "lambda", "sigmasq", "logL(lambda)", "logL(lambda=0)", "P(lambda=0)")))

phylosig.table[,1]<-nutList.n

for (i in 1:length(phylosig.result)) {phylosig.table[i,2:ncol(phylosig.table)]<-round(as.numeric(phylosig.result[[i]]),4)}

write.csv(phylosig.table, file="phylosigByVar.table")

**Supplementary data file 14.** *SD14\_predValidation.lambda-only.R.* Script for carrying out validation for predictions under the lambda model.

#THIS SCRIPT CARRIES OUT VALIDATION FOR PREDICTIONS UNDER THE LAMBDA MODEL

rm(list=ls())

library(phytools)

library(geiger)

source("phyloPred.jknife.R")

source("starPhyloPred.jknife.R")

#GET DATA AND TREE

fullTree<-read.newick("Rabosky\_etal2014.timetree.nex")

fullTree$tip.label[grep("Acanthopagrus\_schlegelii",fullTree$tip.label)]<-"Acanthopagrus\_schlegelii"

fullTree$tip.label[grep("Auxis\_thazard",fullTree$tip.label)]<-"Auxis\_thazard"

fullTree$tip.label[grep("Auxis\_rochei",fullTree$tip.label)]<-"Auxis\_rochei"

fullTree$tip.label[grep("Carassius\_auratus",fullTree$tip.label)]<-"Carassius\_auratus"

fullTree$tip.label[grep("Diagramma\_picta",fullTree$tip.label)]<-"Diagramma\_pictum"

fullTree$tip.label[grep("Helicolenus\_dactylopterus",fullTree$tip.label)]<-"Helicolenus\_dactylopterus"

fullTree$tip.label[grep("Oncorhynchus\_masou",fullTree$tip.label)]<-"Oncorhynchus\_masou"

fullTree$tip.label[grep("Osmerus\_mordax\_mordax",fullTree$tip.label)]<-"Osmerus\_mordax"

fullTree$tip.label[grep("Salvelinus\_alpinus",fullTree$tip.label)]<-"Salvelinus\_alpinus"

fullTree$tip.label[grep("Salvelinus\_leucomaenis\_leucomaenis",fullTree$tip.label)]<-"Salvelinus\_leucomaenis"

fullTree$tip.label[grep("Sebastes\_pachycephalus\_pachycephalus",fullTree$tip.label)]<-"Sebastes\_pachycephalus"

fullTree$tip.label[grep("Thymallus\_arcticus",fullTree$tip.label)]<-"Thymallus\_arcticus"

fullDat<-read.csv("bionut\_working.csv")

rownames(fullDat)<-fullDat[,1]

fullDat<-fullDat[,-1]

datSp<-rownames(fullDat)

species<-datSp[which(match(datSp, fullTree$tip.label, nomatch=0)!=0)]

exclude<-fullTree$tip.label[which(match(fullTree$tip.label, datSp, nomatch=0)==0)]

tree<-drop.tip(fullTree, exclude)

nutDat<-fullDat[species,c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

nut<-data.frame(nutDat[,"protein"], sapply(c("fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

#nut<-data.frame(sapply(c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

rownames(nut)<-rownames(nutDat)

colnames(nut)[1]<-"protein"

nutList<-lapply(1:ncol(nut), function(x) nut[which(is.na(nut[,x])==FALSE),x])

names(nutList)<-colnames(nut)

for(i in 1:length(nutList)){names(nutList[[i]])<-rownames(nut)[which(is.na(nut[,i])==FALSE)]}

nutList.n<-sapply(1:length(nutList), function(x) length(nutList[[x]]))

treeList<-lapply(1:length(nutList), function(x) drop.tip(tree, which(match(tree$tip.label, names(nutList[[x]]), nomatch=0)==0)))

errorMat<-matrix(,length(nutList),3, dimnames=list(names(nutList), c("median\_diff\_percMean", "median\_diff\_nSD", "errorRate")))

for (i in 1:length(nutList)){

nut<-nutList[[i]]

names(nut)<-names(nutList[[i]])

pred<-matrix(, length(nutList[[i]]), 6, dimnames=list(names(nutList[[i]]), c("measuredVal", "predVal", "lcl\_predInt", "ucl\_predInt", "diff\_percMean", "diff\_nSampleSD")))

withinCIMat<-matrix(, length(nutList[[i]]), 1, dimnames=list(names(nutList[[i]]), "withinCI"))

for (j in 1:length(nutList[[i]])){

result<-phyloPred.jknife (treeList[[i]], nut, names(nut)[j])

predVal<-as.numeric(result$PredVal)

measVal<-as.numeric(nut[j])

diff<-abs(measVal - predVal)

predSE<-as.numeric(result$PredSE)

diff\_nPredSE<-diff / predSE

if (diff\_nPredSE < 2) withinCIMat[j]<-TRUE else withinCIMat[j]<-FALSE

lclPredInt<-predVal - (1.96 \* predSE)

uclPredInt<-predVal + (1.96 \* predSE)

if (i > 1) predVal<-exp(predVal) - 1

if (i > 1) measVal<-exp(nut[j]) - 1

if (i > 1) lclPredInt<-exp(lclPredInt) - 1

if (i > 1) uclPredInt<-exp(uclPredInt) - 1

if (i > 1) nutSD<-sd(exp(nut) -1) else nutSD<-sd(nut)

if (measVal == 0) diff1<-NA else diff1<-as.numeric((abs(measVal - predVal) / measVal) \* 100)

if (measVal == 0) diff2<-NA else diff2<-as.numeric((abs(measVal - predVal) / nutSD))

pred[j,]<-round(c(measVal, predVal, lclPredInt, uclPredInt, diff1, diff2), 3)

}

pred<-data.frame(pred, withinCIMat)

if (i == 1) write.csv(pred, file="phyloPredValid.protein.csv")

if (i == 2) write.csv(pred, file="phyloPredValid.lipid.csv")

if (i == 3) write.csv(pred, file="phyloPredValid.omega3.csv")

if (i == 4) write.csv(pred, file="phyloPredValid.omega6.csv")

if (i == 5) write.csv(pred, file="phyloPredValid.iron.csv")

if (i == 6) write.csv(pred, file="phyloPredValid.zinc.csv")

if (i == 7) write.csv(pred, file="phyloPredValid.vitA.csv")

if (i == 8) write.csv(pred, file="phyloPredValid.vitB12.csv")

if (i == 9) write.csv(pred, file="phyloPredValid.vitD.csv")

errorMat[i,1]<-median(pred[,"diff\_percMean"], na.rm=TRUE)

errorMat[i,2]<-median(pred[,"diff\_nSampleSD"], na.rm=TRUE)

errorMat[i,3]<-length(which(withinCIMat == FALSE))/nrow(pred)

write.csv(errorMat, file="phyloPredValid.error.csv")

}

**Supplementary data file 15.** *SD15\_predValidation.pgls-lambda.R.* Script for carrying out validation for predictions under the lambda plus phylogenetic regression model.

#THIS SCRIPT CARRIES OUT VALIDATION FOR PREDICTIONS UNDER THE PHYLOGENETIC REGRESSION + LAMBDA MODEL

rm(list=ls())

library(phytools)

library(geiger)

library(phylolm)

source("phyloPred.jknife.R")

source("starPhyloPred.jknife.R")

#GET DATA AND TREE

fullTree<-read.newick("Rabosky\_etal2014.timetree.nex")

fullTree$tip.label[grep("Acanthopagrus\_schlegelii",fullTree$tip.label)]<-"Acanthopagrus\_schlegelii"

fullTree$tip.label[grep("Auxis\_thazard",fullTree$tip.label)]<-"Auxis\_thazard"

fullTree$tip.label[grep("Auxis\_rochei",fullTree$tip.label)]<-"Auxis\_rochei"

fullTree$tip.label[grep("Carassius\_auratus",fullTree$tip.label)]<-"Carassius\_auratus"

fullTree$tip.label[grep("Diagramma\_picta",fullTree$tip.label)]<-"Diagramma\_pictum"

fullTree$tip.label[grep("Helicolenus\_dactylopterus",fullTree$tip.label)]<-"Helicolenus\_dactylopterus"

fullTree$tip.label[grep("Oncorhynchus\_masou",fullTree$tip.label)]<-"Oncorhynchus\_masou"

fullTree$tip.label[grep("Osmerus\_mordax\_mordax",fullTree$tip.label)]<-"Osmerus\_mordax"

fullTree$tip.label[grep("Salvelinus\_alpinus",fullTree$tip.label)]<-"Salvelinus\_alpinus"

fullTree$tip.label[grep("Salvelinus\_leucomaenis\_leucomaenis",fullTree$tip.label)]<-"Salvelinus\_leucomaenis"

fullTree$tip.label[grep("Sebastes\_pachycephalus\_pachycephalus",fullTree$tip.label)]<-"Sebastes\_pachycephalus"

fullTree$tip.label[grep("Thymallus\_arcticus",fullTree$tip.label)]<-"Thymallus\_arcticus"

fullDat<-read.csv("bionut\_working.csv")

rownames(fullDat)<-fullDat[,1]

fullDat<-fullDat[,-1]

datSp<-rownames(fullDat)

species<-datSp[which(match(datSp, fullTree$tip.label, nomatch=0)!=0)]

exclude<-fullTree$tip.label[which(match(fullTree$tip.label, datSp, nomatch=0)==0)]

tree<-drop.tip(fullTree, exclude)

nutDat<-fullDat[species,c("protein", "fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd")]

nutDat<-data.frame(nutDat[,"protein"], sapply(c("fat", "fac3\_pol", "fac6\_pol", "iron", "zinc", "ret\_eq", "vitb12", "vitd"), function(x) log(nutDat[,x]+1)))

colnames(nutDat)[1]<-"protein"

lifeHistDat<-fullDat[species, c("maxlen", "troph", "depmax")]

nutList<-lapply(1:ncol(nutDat), function(x) data.frame(lifeHistDat[which(is.na(nutDat[,x])==FALSE),], nutDat[which(is.na(nutDat[,x])==FALSE),x]))

names(nutList)<-colnames(nutDat)

for (i in 1:length(nutList)) {colnames(nutList[[i]])[ncol(nutList[[i]])]<-names(nutList)[i]}

for (i in 1:length(nutList)) {

for (j in 1:ncol(nutList[[i]])){

nutList[[i]]<-nutList[[i]][which(is.na(nutList[[i]][,j])==FALSE),]

}

}

treeList<-lapply(1:length(nutList), function(x) drop.tip(tree, tree$tip.label[which(match(tree$tip.label, rownames(nutList[[x]]), nomatch=0)==0)]))

#pgls.table<-matrix(, length(nutList), 18, dimnames=list(names(nutList), c("n", "lambda", "b0", "b1\_maxlen", "b1\_troph", "b1\_depmax", "se\_b0", "se\_b1\_maxlen", "se\_b1\_troph", "se\_b1\_depmax", "tstat\_b0", "tstat\_maxlen", "tstat\_troph", "tstat\_depmax", "P\_b0", "P\_maxlen", "P\_troph", "P\_depmax")))

errorMat<-matrix(,length(nutList),3, dimnames=list(names(nutList), c("median\_diff\_percMean", "median\_diff\_nSD", "errorRate")))

for (i in 1:length(nutList)){

pred<-matrix(, nrow(nutList[[i]]), 6, dimnames=list(rownames(nutList[[i]]), c("measuredVal", "predVal", "lcl\_predInt", "ucl\_predInt", "diff\_percMean", "diff\_nSampleSD")))

withinCIMat<-matrix(, nrow(nutList[[i]]), 1, dimnames=list(rownames(nutList[[i]]), "withinCI"))

maxlen<-nutList[[i]][,"maxlen"]

names(maxlen)<-rownames(nutList[[i]])

troph<-nutList[[i]][,"troph"]

names(troph)<-rownames(nutList[[i]])

depmax<-nutList[[i]][,"depmax"]

names(depmax)<-rownames(nutList[[i]])

nut<-nutList[[i]][,4]

names(nut)<-rownames(nutList[[i]])

for (j in 1:nrow(nutList[[i]])){

maxlen.rm<-maxlen[-j]

troph.rm<-troph[-j]

depmax.rm<-depmax[-j]

nut.rm<-nut[-j]

tree.rm<-drop.tip(treeList[[i]], rownames(nutList[[i]])[j])

result<-phylolm(nut.rm ~ maxlen.rm + troph.rm + depmax.rm, phy=tree.rm, model="lambda")

coeff<-c(result[[3]], result$coeff)

names(coeff)<-c("lambda", "b0", "b1\_maxlen", "b1\_troph", "b1\_depmax")

Yhat<-(coeff["b1\_maxlen"] \* maxlen) + (coeff["b1\_troph"] \* troph) + (coeff["b1\_depmax"] \* depmax) + coeff["b0"]

res<- nut - Yhat

pred.res<-phyloPred.jknife(treeList[[i]], res, rownames(nutList[[i]])[j])

predVal<-Yhat[j] + as.numeric(pred.res$PredVal)

measVal<-as.numeric(nut[j])

diff<-abs(measVal - predVal)

predSE<-as.numeric(pred.res$PredSE)

diff\_nPredSE<-diff / predSE

if (diff\_nPredSE < 2) withinCIMat[j]<-TRUE else withinCIMat[j]<-FALSE

lclPredInt<-predVal - (1.96 \* predSE)

uclPredInt<-predVal + (1.96 \* predSE)

if (i > 1) predVal<-exp(predVal) - 1

if (i > 1) measVal<-exp(nut[j]) - 1

if (i > 1) lclPredInt<-exp(lclPredInt) - 1

if (i > 1) uclPredInt<-exp(uclPredInt) - 1

if (i > 1) nutSD<-sd(exp(nut) -1) else nutSD<-sd(nut)

if (measVal == 0) diff1<-NA else diff1<-as.numeric((abs(measVal - predVal) / measVal) \* 100)

if (measVal == 0) diff2<-NA else diff2<-as.numeric((abs(measVal - predVal) / nutSD))

pred[j,]<-round(c(measVal, predVal, lclPredInt, uclPredInt, diff1, diff2), 3)

}

if (i == 1) write.csv(pred, file="phyloPredValid.PGLS.protein.csv")

if (i == 2) write.csv(pred, file="phyloPredValid.PGLS.lipid.csv")

if (i == 3) write.csv(pred, file="phyloPredValid.PGLS.omega3.csv")

if (i == 4) write.csv(pred, file="phyloPredValid.PGLS.omega6.csv")

if (i == 5) write.csv(pred, file="phyloPredValid.PGLS.iron.csv")

if (i == 6) write.csv(pred, file="phyloPredValid.PGLS.zinc.csv")

if (i == 7) write.csv(pred, file="phyloPredValid.PGLS.vitA.csv")

if (i == 8) write.csv(pred, file="phyloPredValid.PGLS.vitB12.csv")

if (i == 9) write.csv(pred, file="phyloPredValid.PGLS.vitD.csv")

errorMat[i,1]<-median(pred[,"diff\_percMean"], na.rm=TRUE)

errorMat[i,2]<-median(pred[,"diff\_nSampleSD"], na.rm=TRUE)

errorMat[i,3]<-length(which(withinCIMat == FALSE))/nrow(pred)

write.csv(errorMat, file="phyloPredValidPGLS.error.csv")

}