Supplement A: Example R Code for Conducting Random Forest Analysis to Identify Invasive Carp from the Upper Mississippi River, Iowa

```
install.packages("randomForest")

#A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3),
#18--22. R package version 4.6-14. https://CRAN.R-project.org/package=randomForest.


library(randomForest)

Directory <- "~"

setwd(Directory)

### Import data

Data  <-
read.csv(text=getURL("https://raw.githubusercontent.com/ccamacho0526/InvasiveCarpRandom
Forest/master/SupplementalDataSet.csv"), header=T)

#Data can be found at Github located in the above code or through Iowa State University's
#Datashare repository (https://doi.org/10.25380/iastate.9879716).

### View first few rows of data

head(Data)
```

Month	OrdinalDate	WaterTemperature	Conductivity	EmbryoMidlineLength	MembraneAverage	...
6	172	26.3	473	4.012	4.066	...
6	164	22.1	316	3.360	3.361	...
5	147	19.9	332	3.384	2.843	...
6	172	25.1	340	4.093	3.427	...
6	172	25.1	340	4.081	4.560	...
6	172	26.3	473	4.479	3.802	...
5	147	19.9	328	3.464	3.566	...

```
##### Records without genotypes to predict ##########

NOgeno <-Data[is.na(Data$Common.name) | Data$Common.name =="",]

### Records with genotypes to train ##########

geno <-Data[ !(is.na(Data$Common.name) | Data$Common.name ==""),]
```

```
########   Random Forest Analyses #########


########## Random Forest - Family ############

setwd(Directory)

dir.create("rf.Family")

setwd("rf.Family")

### get column name

names(c(geno))

#### remove extra columns leave only variables and classification

Train.Family <- geno[,-c(18,20:25)]

names(c(Train.Family))

#### set classification column as factor

Train.Family$Family <- factor(Train.Family$Family)

#### Use data that has all variables

rf.Train.Family <- Train.Family[which(complete.cases(Train.Family)),]

rf.Train.Family$Family <-factor(rf.Train.Family$Family)

####Random Forest

rf.Family <- randomForest(rf.Train.Family$Family ~.,data=rf.Train.Family,
importance=T,ntree=1000)

rf.Family

#### Variable Diagnostics

rf.Family.Accuracy = importance(rf.Family,type = 1)

rf.Family.Gini = importance(rf.Family,type = 2)

#### Write output files

capture.output(rf.Family,file = "rf_Family_Output.txt")

write.csv(rf.Family$confusion,'rf_Family_ConfusionMatrix.csv')

write.csv(rf.Family.Accuracy ,'rf_Family_Accuracy.csv')

write.csv(rf.Family.Gini ,'rf_Family_Gini.csv')
```

```
#### Remove data files from environment
remove(Train.Family)
remove(rf.Train.Family)
remove(rf.Family.Accuracy)
remove(rf.Family.Gini)
remove(rf.Family)
#### Reset to parent directory
setwd(Directory)

########## Random Forest - Genus ############
setwd(Directory)
dir.create("rf.Genus")
setwd("rf.Genus")
### get column name
names(c(geno))
#### remove extra columns leave only variables and classification
Train.Genus <- geno[,-c(18:19,21:25)]
names(c(Train.Genus))
#### set classification column as factor
Train.Genus$Genus <- factor(Train.Genus$Genus)
#### Use data that has all variables
rf.Train.Genus <- Train.Genus[which(complete.cases(Train.Genus)),]
rf.Train.Genus$Genus <-factor(rf.Train.Genus$Genus)
####Random Forest
rf.Genus <- randomForest(rf.Train.Genus$Genus ~.,data=rf.Train.Genus,
importance=T,ntree=1000)
rf.Genus
#### Variable Diagnostics
rf.Genus.Accuracy = importance(rf.Genus,type = 1)
```

```
rf.Genus.Gini = importance(rf.Genus,type = 2)
#### Write output files
capture.output(rf.Genus,file = "rf_Genus_Output.txt")
write.csv(rf.Genus$confusion,'rf_Genus_ConfusionMatrix.csv')
write.csv(rf.Genus.Accuracy ,'rf_Genus_Accuracy.csv')
write.csv(rf.Genus.Gini ,'rf_Genus_Gini.csv')
#### Remove data files from environment
remove(Train.Genus)
remove(rf.Train.Genus)
remove(rf.Genus.Accuracy)
remove(rf.Genus.Gini)
remove(rf.Genus)
#### Reset to parent directory
setwd(Directory)

########## Random Forest - Species ############
setwd(Directory)
dir.create("rf.Species")
setwd("rf.Species")
### get column name
names(c(geno))
#### remove extra columns leave only variables and classification
Train.Species <- geno[,-c(18:20,22:25)]
names(c(Train.Species))
#### set classification column as factor
Train.Species$Species <- factor(Train.Species$Species)
#### Use data that has all variables
rf.Train.Species <- Train.Species[which(complete.cases(Train.Species)),]
```

```
rf.Train.Species$Species <-factor(rf.Train.Species$Species)

####Random Forest

rf.Species <- randomForest(rf.Train.Species$Species ~.,data=rf.Train.Species,
importance=T,ntree=1000)

rf.Species

#### Variable Diagnostics

rf.Species.Accuracy = importance(rf.Species,type = 1)

rf.Species.Gini = importance(rf.Species,type = 2)

#### Write output files

capture.output(rf.Species,file = "rf_Species_Output.txt")

write.csv(rf.Species$confusion,'rf_Species_ConfusionMatrix.csv')

write.csv(rf.Species.Accuracy ,'rf_Species_Accuracy.csv')

write.csv(rf.Species.Gini ,'rf_Species_Gini.csv')

#### Remove data files from environment

remove(Train.Species)

remove(rf.Train.Species)

remove(rf.Species.Accuracy)

remove(rf.Species.Gini)

remove(rf.Species)

#### Reset to parent directory

setwd(Directory)


########## Random Forest - Family with aggregate invasive carp class (IC) ############

setwd(Directory)

dir.create("rf.Family.IC")

setwd("rf.Family.IC")

### get column name

names(c(geno))

#### remove extra columns leave only variables and classification
```

```
Train.Family.IC <- geno[,-c(18:22,24:25)]
names(c(Train.Family.IC))
#### set classification column as factor
Train.Family.IC$Family.IC <- factor(Train.Family.IC$Family.IC)
#### Use data that has all variables
rf.Train.Family.IC <- Train.Family.IC[which(complete.cases(Train.Family.IC)),]
rf.Train.Family.IC$Family.IC <-factor(rf.Train.Family.IC$Family.IC)
####Random Forest
rf.Family.IC <- randomForest(rf.Train.Family.IC$Family.IC ~.,data=rf.Train.Family.IC,
importance=T,ntree=1000)
rf.Family.IC
#### Variable Diagnostics
rf.Family.IC.Accuracy = importance(rf.Family.IC,type = 1)
rf.Family.IC.Gini = importance(rf.Family.IC,type = 2)
#### Write output files
capture.output(rf.Family.IC,file = "rf_Family.IC_Output.txt")
write.csv(rf.Family.IC$confusion,'rf_Family.IC_ConfusionMatrix.csv')
write.csv(rf.Family.IC.Accuracy ,'rf_Family.IC_Accuracy.csv')
write.csv(rf.Family.IC.Gini ,'rf_Family.IC_Gini.csv')
#### Remove data files from environment
remove(Train.Family.IC)
remove(rf.Train.Family.IC)
remove(rf.Family.IC.Accuracy)
remove(rf.Family.IC.Gini)
remove(rf.Family.IC)
#### Reset to parent directory
setwd(Directory)

########## Random Forest - Genus with aggregate invasive carp class (IC) ############
```

```
setwd(Directory)
dir.create("rf.Genus.IC")
setwd("rf.Genus.IC")
### get column name
names(c(geno))
#### remove extra columns leave only variables and classification
Train.Genus.IC <- geno[,-c(18:23,25)]
names(c(Train.Genus.IC))
#### set classification column as factor
Train.Genus.IC$Genus.IC <- factor(Train.Genus.IC$Genus.IC)
#### Use data that has all variables
rf.Train.Genus.IC <- Train.Genus.IC[which(complete.cases(Train.Genus.IC)),]
rf.Train.Genus.IC$Genus.IC <-factor(rf.Train.Genus.IC$Genus.IC)
####Random Forest
rf.Genus.IC <- randomForest(rf.Train.Genus.IC$Genus.IC ~.,data=rf.Train.Genus.IC,
importance=T,ntree=1000)
rf.Genus.IC
#### Variable Diagnostics
rf.Genus.IC.Accuracy = importance(rf.Genus.IC,type = 1)
rf.Genus.IC.Gini = importance(rf.Genus.IC,type = 2)
#### Write output files
capture.output(rf.Genus.IC,file = "rf_Genus.IC_Output.txt")
write.csv(rf.Genus.IC$confusion,'rf_Genus.IC_ConfusionMatrix.csv')
write.csv(rf.Genus.IC.Accuracy ,'rf_Genus.IC_Accuracy.csv')
write.csv(rf.Genus.IC.Gini ,'rf_Genus.IC_Gini.csv')
#### Remove data files from environment
remove(Train.Genus.IC)
remove(rf.Train.Genus.IC)
remove(rf.Genus.IC.Accuracy)
```

```
remove(rf.Genus.IC.Gini)
remove(rf.Genus.IC)
#### Reset to parent directory
setwd(Directory)

########## Random Forest - Species with aggregate invasive carp class (IC) ############
setwd(Directory)
dir.create("rf.Species.IC")
setwd("rf.Species.IC")
### get column name
names(c(geno))
#### remove extra columns leave only variables and classification
Train.Species.IC <- geno[,-c(18:24)]
names(c(Train.Species.IC))
#### set classification column as factor
Train.Species.IC$Species.IC <- factor(Train.Species.IC$Species.IC)
#### Use data that has all variables
rf.Train.Species.IC <- Train.Species.IC[which(complete.cases(Train.Species.IC)),]
rf.Train.Species.IC$Species.IC <-factor(rf.Train.Species.IC$Species.IC)
####Random Forest
rf.Species.IC <- randomForest(rf.Train.Species.IC$Species.IC ~.,data=rf.Train.Species.IC,
importance=T,ntree=1000)
rf.Species.IC
#### Variable Diagnostics
rf.Species.IC.Accuracy = importance(rf.Species.IC,type = 1)
rf.Species.IC.Gini = importance(rf.Species.IC,type = 2)
#### Write output files
capture.output(rf.Species.IC,file = "rf_Species.IC_Output.txt")
write.csv(rf.Species.IC$confusion,'rf_Species.IC_ConfusionMatrix.csv')
```

```
write.csv(rf.Species.IC.Accuracy ,'rf_Species.IC_Accuracy.csv')
write.csv(rf.Species.IC.Gini ,'rf_Species.IC_Gini.csv')
#### Partial Plots for variables
imp <- importance(rf.Species.IC)
impvar <- rownames(imp)[order(imp[, 1], decreasing=TRUE)]
op <- par(mfrow=c(2, 3))
for (i in seq_along(impvar)) {
  partialPlot(rf.Species.IC, rf.Train.Species.IC, impvar[i], xlab=impvar[i],
            main=paste("Partial Dependence on", impvar[i]))}
par(op)
#### Remove data files from environment
remove(Train.Species.IC)
remove(rf.Train.Species.IC)
remove(rf.Species.IC.Accuracy)
remove(rf.Species.IC.Gini)
remove(rf.Species.IC)
#### Reset to parent directory
setwd(Directory)


########## Random Forest - Species with aggregate invasive carp class (IC) reduced variables
############
setwd(Directory)
dir.create("rf.Species.IC.Reduced")
setwd("rf.Species.IC.Reduced")
### get column name
names(c(geno))
#### remove extra columns leave only variables and classification
Train.Species.IC <- geno[,-c(1,5,12,13,14,16,18:24)]
names(c(Train.Species.IC))
```

```
#### set classification column as factor

Train.Species.IC$Species.IC <- factor(Train.Species.IC$Species.IC)

#### Use data that has all variables

rf.Train.Species.IC <- Train.Species.IC[which(complete.cases(Train.Species.IC)),]

rf.Train.Species.IC$Species.IC <-factor(rf.Train.Species.IC$Species.IC)

####Random Forest

rf.Species.IC <- randomForest(rf.Train.Species.IC$Species.IC ~.,data=rf.Train.Species.IC,
importance=T,ntree=1000, type = "prob")

rf.Species.IC

#### Variable Diagnostics

rf.Species.IC.Accuracy = importance(rf.Species.IC,type = 1)

rf.Species.IC.Gini = importance(rf.Species.IC,type = 2)

varImpPlot(rf.Species.IC)

#### Write output files

capture.output(rf.Species.IC,file = "rf_Species.IC_Output.txt")

write.csv(rf.Species.IC$confusion,'rf_Species.IC_ConfusionMatrix.csv')

write.csv(rf.Species.IC.Accuracy ,'rf_Species.IC_Accuracy.csv')

write.csv(rf.Species.IC.Gini ,'rf_Species.IC_Gini.csv')

#### Partial Plots for variables

imp <- importance(rf.Species.IC)

impvar <- rownames(imp)[order(imp[, 1], decreasing=TRUE)]

op <- par(mfrow=c(3, 3))

for (i in seq_along(impvar)) {

  partialPlot(rf.Species.IC, rf.Train.Species.IC, impvar[i], which.class = "Notropis atherinoides",
xlab=impvar[i],

        main=paste("Partial Dependence on", impvar[i]))}

par(op)

impvar <- c("MembraneAverage","EmbryoAverage")

op <- par(mfrow=c(1, 2))
```

```
for (i in seq_along(impvar)) {
  partialPlot(rf.Species.IC, rf.Train.Species.IC, impvar[i], xlab=impvar[i])}
par(op)
predictNOgeno <- predict(rf.Species.IC,NOgeno,type = "prob")
#### Remove data files from environment
remove(Train.Species.IC)
remove(rf.Train.Species.IC)
remove(rf.Species.IC.Accuracy)
remove(rf.Species.IC.Gini)
remove(rf.Species.IC)
#### Reset to parent directory
setwd(Directory)
```