

Appendix A.—Narwhal Allocation Model R Code.

```
##### Narwhal Allocation Model #####
## August 2018, R version 3.3.3
## for questions, contact thomas.doniol-valcroze@dfo-mpo.gc.ca
#####

#####
## 0. Working directory and packages
## Set your working directory (must contain the input file "input_table.txt")
work.dir <- ("C:/Users/myusername/Documents")
setwd(work.dir)

## Check if packages are installed. If not, install them and load them
## Note that these packages are only needed to plot the uncertainty version of
## the Allocation matrix as a grid of histograms
packages=c('reshape2','ggplot2','RColorBrewer')
invisible(lapply(packages, function(x) {if (!require(x, character.only=T))
{install.packages(x);require(x)}}))
#####

#####
## 1. Read input table to determine proportional availability matrix P
## The input format can vary depending on the type of data (movements, genetics, etc.)
## In the case of satellite transmitters and expert opinion for narwhals,
## we load a single table for with information coded in the following manner:
## columns SS.x, JS.x, etc. are the number of tags (x) from each stock visiting each site
## columns SS.n, JS.n, etc. are the number of transmitting tags (n) for each site

## The values "9999" and "-1" are placeholders used to identify special cases
## that correspond to the cell categories outlined in Table 1:
## 0/9999 = Defined zero
## 0/-1 = Probable zero
## x/n = Partial hunt
## 1/-1 = Probable hunt
## 9999/9999 = Defined hunt

## Read input table
input <- read.delim("input_table.txt",header=T,row.names=1)

## Extract labels for stocks (names before the ".x" or ".n")
stocks <- unique(substr(colnames(input),1,regexpr(".",colnames(input),fixed=T)-1))
```

```

## Extract labels for hunting sites
sites <- rownames(input)

## Get the number of sites and stocks to set dimensions of P and A matrices
nsites <- length(sites)
nstocks <- length(stocks)
#####

#####

## 2. Format x and n matrices prior to computing P matrix

## Separate input table into a matrix of x values and a matrix of n values
xx.mat <- as.matrix(input[,endsWith(colnames(input),".x")])
nn.mat <- as.matrix(input[,endsWith(colnames(input),".n")])

## Select Defined and Probable zeros and ones based on placeholder values
Def1 <- which(xx.mat==9999 & nn.mat==9999)
Def0 <- which(xx.mat==0 & nn.mat==9999)
Prob1 <- which(xx.mat==1 & nn.mat==-1)
Prob0 <- which(xx.mat==0 & nn.mat==-1)

#####

## OPTIONAL: Visualize the x/n data in a single table:
xxnn.mat <- matrix(paste(as.character(xx.mat),"/",as.character(nn.mat),
sep=""),nrow=nsites,ncol=nstocks)
colnames(xxnn.mat) <- stocks
rownames(xxnn.mat) <- sites

## Replace 9999 placeholders with Defined and Probable 0 and 1 to yield Table 4:
xxnn.mat[Def1] <- 1
xxnn.mat[Def0] <- 0
xxnn.mat[Prob1] <- "1*"
xxnn.mat[Prob0] <- "0*"
print(xxnn.mat,quote=F) # Table 4

## Replace with Z for Table 5
xxnn.mat[Prob1] <- "Z/Z"
xxnn.mat[Prob0] <- "0/Z"
print(xxnn.mat,quote=F) # Table 5
#####

#####

## 3. Compute Proportional Availability matrix P without uncertainty

```

```

## Prior to calculation, for Probable 0s and 1s, must replace placeholder "-1" by 1
nn.mat[Prob1] <- 1
nn.mat[Prob0] <- 1

## The fixed version of the P matrix is:
prop.mat <- xx.mat/nn.mat

## Add proper column names and examine P (with 2 decimals)
colnames(prop.mat) <- stocks
print(round(prop.mat,2))
#####

#####
## 4. Compute Allocation matrix A without uncertainty

## Need a vector of abundance estimates and CVs (same order as "stocks")
## (could be read as a table)
## For Baffin Bay narwhal stocks, the Canadian and Greenland estimates are:
abund <- c(16360,12694,8368,3091,49768,35043,10489,17555)
abund.cv <- c(0.65,0.33,0.25,0.50,0.20,0.42,0.24,0.35)

## Calculate Allocation matrix A
alloc.mat <- t(apply(prop.mat,1,function(x) (abund*x)/sum(abund*x) ))

## Examine table and note that columns for each site should sum to 1
print(round(alloc.mat,2))
#####

#####
## 5. Apply Allocation Matrix to catches, to calculate removals from each stock

## input catches in same order as "sites" (or read a table of catches)
## Example: Catches of narwhal in Canada and Greenland in 2013 from Table 2):
catch <- c(0,87,5,0,4,82,101,66,1,33,23,43,167,4,30,82,58,11,9,143,3,1,18,0)

## This table shows the catches of each site as an extra column of the allocation matrix:
print(data.frame(round(alloc.mat,2),Catches=catch))

## these catch values are multiplied by allocation proportions in the same row
## to yield removals for each cell of the matrix
print(data.frame(round(alloc.mat*catch),Catches=catch))

## To obtain total removals from each stock, the allocation matrix is transposed

```

```

## and multiplied by the catch vector using matrix algebra
remov <- t(alloc.mat) %*% catch

## note: this is the equivalent of multiplying catches by each cell in the
## corresponding line (site) then summing each column (stock):
# remov <- apply(alloc.mat,2,function(x) sum(catch*x) )

## Adding this vector of removals to the bottom of the table
print(round(rbind(data.frame(round(alloc.mat*catch),Catches=catch),data.frame(t(data.frame(re
movals=remov)),Catches=sum(remov))))))

## Optional version of the same table with allocation proportions
print(round(rbind(data.frame(alloc.mat,Catches=catch),data.frame(t(data.frame(removals=remov
)),Catches=sum(remov))),2))

## optional: check that the sum of allocated removals equals the input catches
print(all.equal(sum(catch),sum(remov)))
#####

#####
##### UNCERTAINTY VERSION #####
#####

## These are alternate steps to compute the stochastic version of matrices P and A
## This process includes uncertainty in mixing proportions on hunting sites
## and uncertainty in abundance estimates

#####
## 3b. Compute Proportional Availability matrix P without uncertainty

## Choose value for sensitivity parameter Z for Probable zero's and Probable hunts
## A large value of Z represents higher certainty
## In the case of satellite transmitters, Z can be interpreted as a fictional
## number of transmitters deployed to estimate movement of a stock to a site;
## for instance, a "Probable 0" with a Z=100 means a degree of certainty equivalent
## to having 0 whale visiting the site out of 100 satellite deployments (i.e., quite certain)
## whereas a value of Z=1 is equivalent to having 0 whale visiting the site out of
## only 1 deployment (i.e., low certainty)

Z <- 1 #for sensitivity analysis, try values 10, 100, 1000, etc.

## Replace values of P for Probable 0s and 1s by Z
xx.mat[Prob1] <- Z
nn.mat[Prob1] <- Z

```

```

nn.mat[Prob0] <- Z

## In this version, the availability proportions follow these Beta distributions:
## 0/9999 -> beta(1,10000)   Defined zero
## 0/-1 -> beta(1,Z)        Probable zero
## x/n -> beta(x+1, n-x+1)   Partial hunt
## 1/-1 -> beta(Z,1)        Probable hunt
## 9999/9999 -> beta(10000,1) Defined hunt

## Plot example distributions to understand the impact of sample size (n) and Z
dev.new()
par(mfrow=c(3,2))
hist(rbeta(100000,6,6),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Partial hunt n/x = 5/10")
hist(rbeta(100000,51,51),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Partial hunt n/x = 50/100")
hist(rbeta(100000,1,10),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Probable zero Z = 10")
hist(rbeta(100000,1,100),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Probable zero Z = 100")
hist(rbeta(100000,10,1),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Probable hunt Z = 10")
hist(rbeta(100000,100,1),col="grey",xlim=c(0,1),breaks=seq(0,1,0.025),xlab="Proportion",
main="Probable hunt Z = 100")

## The distributions are drawn for each cell of the matrix using a number of
## draws and stored into an array:
draws <- 100000 # at least 10,000 are recommended but note that computing time increases
quickly past that
prop.array <- array(dim=c(nsites,nstocks,draws))
for(l in 1:nsites){
  for(c in 1:nstocks){
    prop.array[l,c] <- rbeta(draws,xx.mat[l,c]+1,nn.mat[l,c]-xx.mat[l,c]+1)
  }
}

## The means of all draws for each cell can be examined with:
print(round(apply(prop.array,c(1,2),mean),2))
## Note how Defined Hunts and Zeroes have not changed,
## Probable Hunts and Zeroes only change for low values of Z,
## and Partial Hunts proportions can change for low samples sizes (i.e., small n)
#####

#####
## 4b. Compute Allocation matrix with uncertainty on proportions (but not on abundance)

```

```

## If no uncertainty is included in abundance estimates, the Allocation matrix
## is calculated as before for each cell and draw in the array:
alloc.array.temp <- apply(prop.array,c(1,3),function(x) (abund*x)/sum(abund*x) )

```

```

## need to transpose the array because the previous line changed its orientation
alloc.array <- aperm(alloc.array.temp,c(2,1,3))

```

```

## 4c. Compute Allocation matrix with uncertainty in pop abundance (i.e., full uncertainty)

```

```

## To include uncertainty around population estimates, for each draw of
## the array, population abundance is drawn from a log-normal distribution:
## N(iter,log(abund^2/sqrt(abund^2*(1+abund.cv^2))),sqrt(log(1+(abund.cv^2))))
abund.mat <-
matrix(rlnorm(nstocks*draws,log(abund^2/sqrt(abund^2*(1+abund.cv^2))),sqrt(log(1+(abund.cv
^2))))),nrow=draws,ncol=nstocks,byrow=T)

```

```

## The Allocation is then calculated in the normal way for each cell:

```

```

alloc.array.full <- array(dim=c(nsites,nstocks,draws))
for(l in 1:nsites){
  for(c in 1:nstocks){
    for(i in 1:draws){
      alloc.array.full[l,c,i] <- abund.mat[i,c]*prop.array[l,c,i]/sum(abund.mat[i,]*prop.array[l,i])
    }
  }
}

```

```

#####

```

```

#####

```

```

## 5b. Apply allocation array to catches, to calculate removals from each stock

```

```

## full distribution:

```

```

remov.dist <- apply(alloc.array,c(2,3),function(x) sum(catch*x))

```

```

## summary stats:

```

```

mean.remov <- round(apply(remov.dist,1,mean),2)

```

```

cv.remov <- round(apply(remov.dist,1,function(x) sd(x)/mean(x)),2)

```

```

## 5c. apply alloc to catches (full uncertainty)

```

```

## full distribution:

```

```

remov.dist.full <- apply(alloc.array.full,c(2,3),function(x) sum(catch*x))

```

```

## summary stats:

```

```

mean.remov.full <- round(apply(remov.dist.full,1,mean),2)

```

```

cv.remov.full <- round(apply(remov.dist.full,1,function(x) sd(x)/mean(x)),2)

```

```

## Table of results:
print(round(rbind(remov,mean.remov,mean.remov.full)))

#####
#####
## PLOTTING THE ALLOCATION MATRICES (UNCERTAINTY VERSION)

## Because each cell of the P and A matrices now contain a full distribution
## of proportion values instead of a fixed value, it is harder to present the
## results in a standard table.

## One option is to take the mean of each cell for all random draws:
alloc.means <- apply(alloc.array.full,c(1,2),mean)
colnames(alloc.means) <- stocks ; row.names(alloc.means) <- sites
print(round(alloc.means,2))

## Another option is to plot the matrix as a grid of histograms:

## Expanded names for sites:
sites2<-c("Etah", "Qaanaaq", "GF Spring", "GF Summer", "GF
Fall", "Upernavik", "Ummannaq", "Disko Bay", "CCA Spring", "CCA Summer", "CCA Fall", "AB
Spring", "AB Summer", "AB Fall", "PI Spring", "PI Summer", "PI Fall", "BIC Spring", "BIC
Summer", "BIC Fall", "BIS Spring", "BIS Summer", "BIS Fall", "BIS Winter")

## If using the full uncertainty, run this line (otherwise, comment or skip it):
alloc.array <- alloc.array.full

## Name dimensions of the Allocation array:
dimnames(alloc.array)[[1]] <- sites2
dimnames(alloc.array)[[2]] <- stocks
dimnames(alloc.array)[[3]] <- paste("draws",1:draws)

## Create a matrix of average proportions (choose mean or median) and name dimensions:
averagealloc.mat <- round(apply(alloc.array,c(1,2),mean),2)
#average.mat<-round(apply(alloc.array,c(1,2),median),2) # uncomment line for median
dimnames(averagealloc.mat)[[1]] <- sites2
dimnames(averagealloc.mat)[[2]] <- stocks

## Melt allocation array to dataframe format and reorder variables:
alloc.df <- melt(alloc.array)
colnames(alloc.df) <- c("sites", "stocks", "draws", "proportion")

## The two following commands are to make sure the factors (stocks, sites) are
## in the right order (important for display); uncomment to choose a different order

```

```

#alloc.df$stocks <- factor(alloc.df$stocks,levels=c("SS", "JS", "IB", "MB", "SI", "AI", "ES", "EB"))
#alloc.df$sites <- factor(alloc.df$sites,levels=c("Etah", "Qaanaaq", "GF Spring", "GF
Summer", "GF Fall", "Upernavik", "Uummannaq", "Disko Bay", "CCA Spring", "CCA
Summer", "CCA Fall", "AB Spring", "AB Summer", "AB Fall", "PI Spring", "PI Summer", "PI Fall",
"BIC Spring", "BIC Summer", "BIC Fall", "BIS Spring", "BIS Summer", "BIS Fall", "BIS Winter"))

## Melt average matrix in dataframe format:
averagealloc.df <- melt(averagealloc.mat)
colnames(averagealloc.df) <- c("sites", "stocks", "prop")

## Choose color scheme:
col.brew <- brewer.pal(name="RdBu",n=10)

## Create ggplot object:
matrixA.plot <- ggplot(alloc.df, aes(x = proportion)) +
  geom_histogram(binwidth = 0.05, fill=col.brew[7], alpha = 1, position = "identity",
boundary=0) +
  facet_grid(sites ~ stocks, scales = "free_y") + xlab("Proportion") + ylab("Frequency") +
  scale_x_continuous(breaks=c(0.1,0.5,0.9), labels=c("0", "0.5", "1"),expand = c(0,0),limits =
c(0,1)) +
  scale_y_continuous(breaks=c(0,draws/8,draws/4,draws*6/8), labels=c("", "", "", "" ),expand =
c(0,0)) +
  coord_cartesian(xlim=c(0,1),ylim = c(0,draws/4)) +
  geom_text(data=averagealloc.df, aes(label=prop, group=NULL, y=draws/8,x=0.5,size=4))

## Open a new graphics window and plot the object with additional stylistic constraints:
dev.new()
print(matrixA.plot+theme_bw()+theme(legend.position='none')+theme(axis.ticks=element_blank
()) + theme(strip.text.x = element_text(angle=0)) + theme(strip.text.y = element_text(angle=0))+
theme(panel.grid.minor=element_blank(), panel.grid.major=element_blank()))

## For large matrices (especially in vertical format), screen size may be limiting
## In this case, you can save a png version for easier use:
ggsave("A_matrix.png", width = 9, height = 12)

```