

# Habitat Suitability Index Modeling

Katie Lankowicz

02 April 2022

This document will preserve the workflow and explain the code for building suitability indices (SIs) and an overall habitat suitability index (HSI) for the modeled space that will serve as the backdrop for our individual-based model of juvenile menhaden movement.

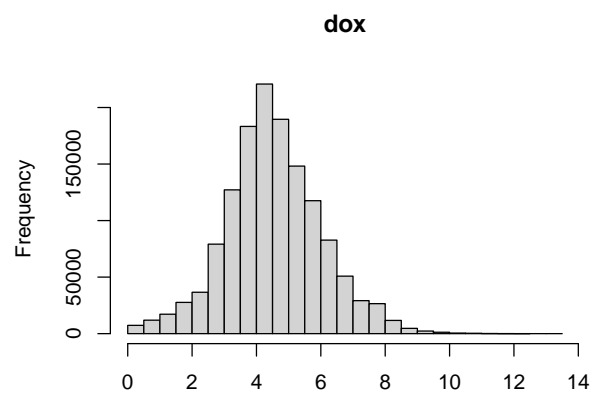
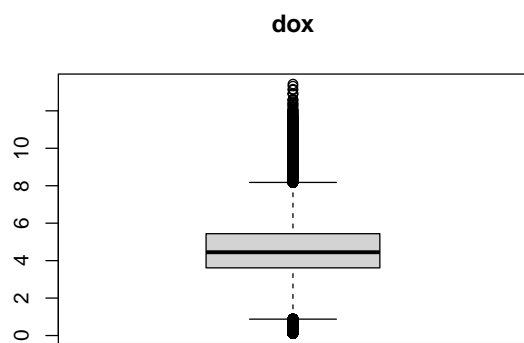
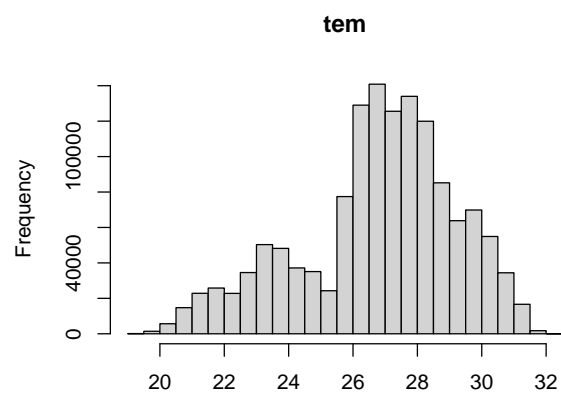
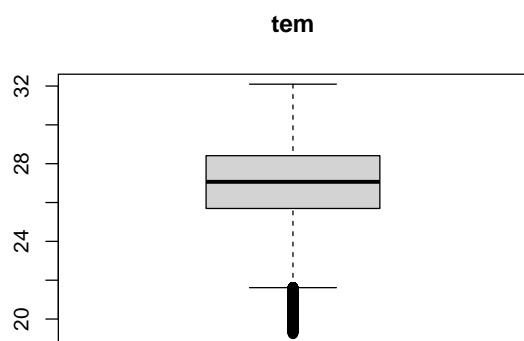
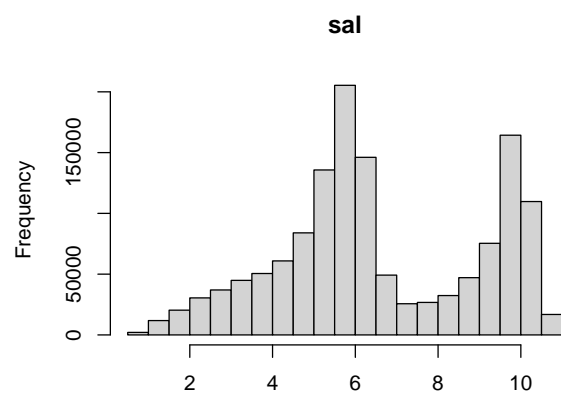
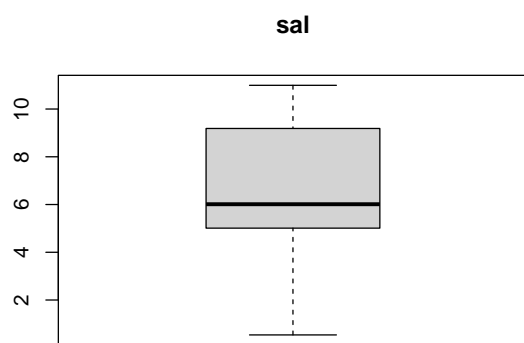
## Loading the environmental rasters

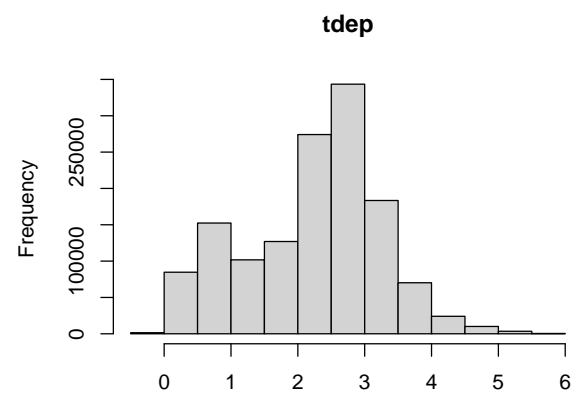
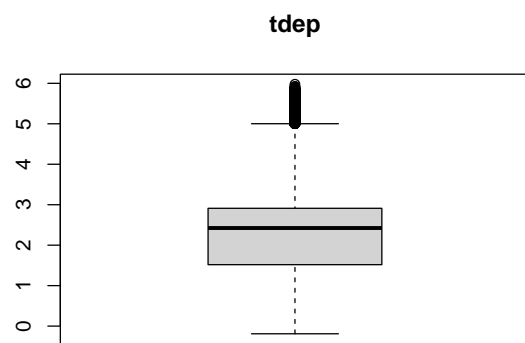
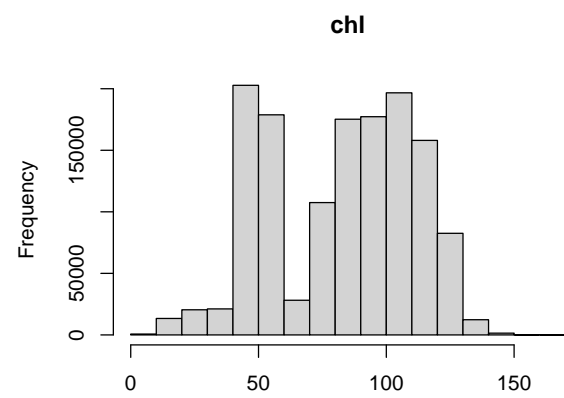
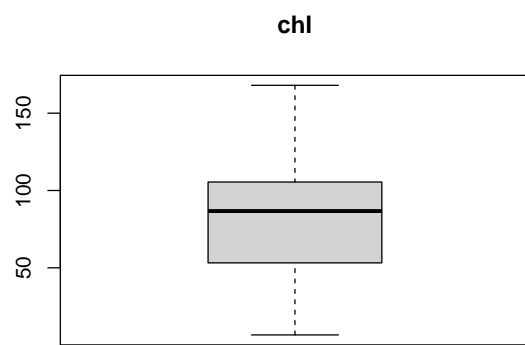
A previous document (“Preparing SCHISM-ICM netcdf files”) has explained the preparation of SCHISM-ICM data for inclusion in the model. Here, we will load the prepared environmental data and rasterized model space. We will also explore the range of these variables. Please keep in mind that these boxplots and histograms will show the range of the variables through the entire space and time of the model.

Variables available to us are salinity (ppt), temperature (°C), dissolved oxygen (mg/L), chlorophyll (ug/L), and depth (m). The ranges of these values are comparable to those the authors have recorded (albiet intermittently) in St. Leonard Creek, where our ARIS menhaden schooling information comes from.

```
# Load data
load(paste0('G:/My Drive/Documents_Backup/Modeling/Modeling_RCode/',
            'Raster_Data/ncdf_output.RData'))

# Set plotting arrangement
par(mfrow=c(1,2))
# Create boxplot and histogram for each variable
for(i in 1:length(env)){
  char <- names(env[i])
  val <- env[[i]]
  val$idx <- NULL
  val <- unlist(val, use.names = F)
  boxplot(val, main=paste(char))
  hist(val, main=paste(char), xlab=' ')
  rm(val, char)
}
```





## Relating habitat suitability to environmental variables

We now need to make some decisions about what makes the most suitable habitat for juvenile menhaden schools. We will need to generate suitability functions for each of our environmental variables. The shape of these functions is not likely to be linear. Values from the literature will provide priors from which we will build these functions.

### Dissolved oxygen

We will start with dissolved oxygen. It is obvious that deoxygenated waters ( $< 2\text{mg/L}$ ) are poor habitat and highly-oxygenated waters are more suitable. The shape of the suitability curve is loosely based on a 1982 report on habitat suitability indexes for estuarine gulf menhaden, which are closely-related clupeids. This report proposes that weekly DO concentrations of 3-5 mg/L have a linear increase in suitability, and increases in DO beyond 5 mg/L offer little additional benefit.

```
# Call dissolved oxygen
val <- env[['dox']]
# Remove indexing variable
val$idx <- NULL
# Convert from matrix to vector
val <- unlist(val, use.names = F)
# Save min/ max and rounded range
minval <- min(val)
maxval <- max(val)
rangeval <- ceiling(maxval) - floor(minval)

# Calculate number of points to have an estimate every 0.0001 x units
nest <- rangeval/0.0001

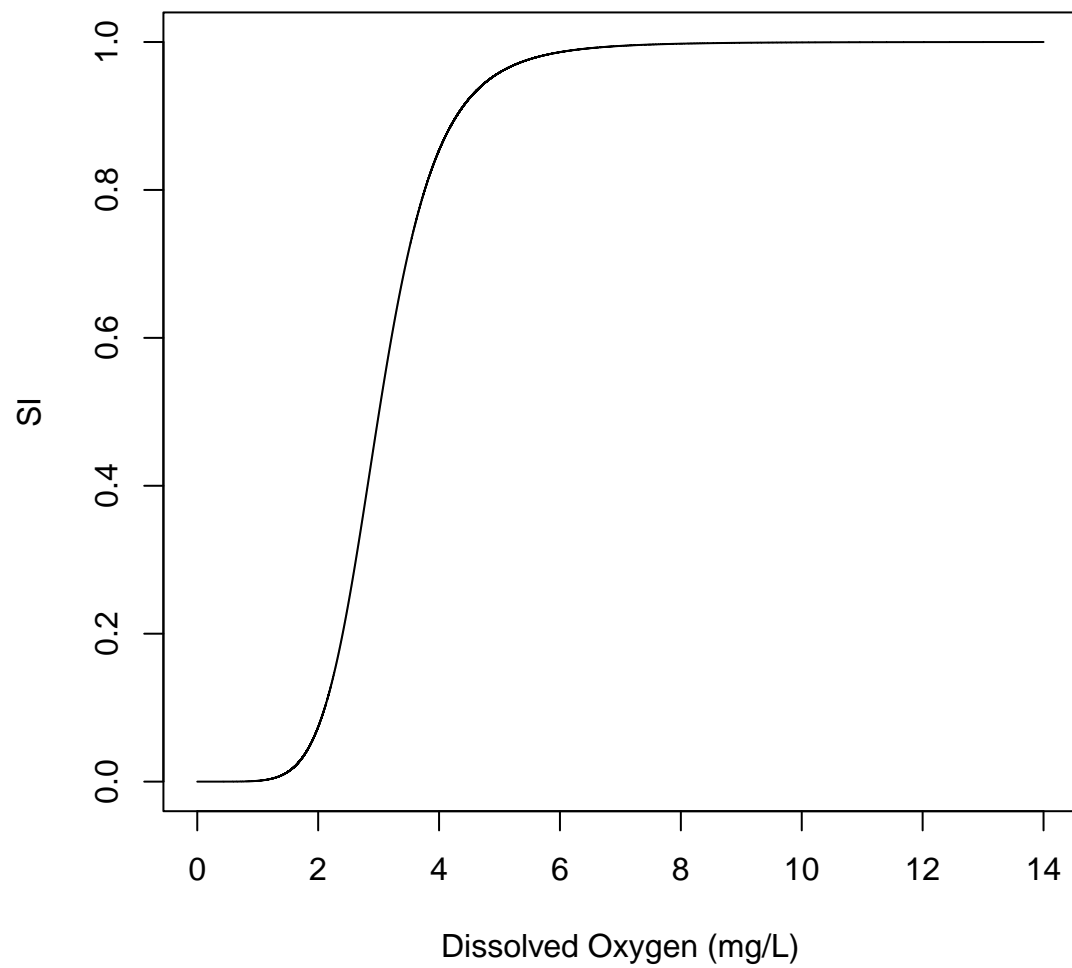
# Fit curve
dox_curve <- curve(0.9822161 + (-0.0309863 - 0.9822161)/(1 + (x/3.008742)^6.191692),
                    from=floor(minval),
                    to=ceiling(maxval),
                    xlab='Dissolved Oxygen (mg/L)',
                    ylab='SI',
                    n=nest
                    )

# Unlist from curve item to dataframe
dox_curve <- as.data.frame(t(do.call(rbind, dox_curve)))

# Rescale so there are no negative values
dox_curve$y <- range01(dox_curve$y)

#Plot
plot(dox_curve$x, dox_curve$y, type='l',
      xlab='Dissolved Oxygen (mg/L)', ylab='SI')

rm(minval, maxval, nest, rangeval, val)
```



## Depth

The tidal fluctuations in the SCHISM simulation result in “dry cells”, or times when near-coast areas are above the low water mark. Obviously, “dry cells” will have a suitability of 0. From empirical ARIS data, juvenile menhaden prefer shallow areas. In St. Leonard Creek, GAM output of density of menhaden decreases almost linearly from 1m to 5m, at which point density is less than the mean for the entire dataset. Areas deeper than 5m are still suitable habitat, so we will not scale the suitability of areas > 5m in depth to 0. We will represent this relationship with the following function.

```
# Call depth
val <- env[['tdep']]
# Remove indexing variable
val$idx <- NULL
# Convert from matrix to vector
val <- unlist(val, use.names = F)
# Save min/ max and rounded range
minval <- min(val)
maxval <- max(val)
rangeval <- ceiling(maxval) - floor(minval)

# Calculate number of points to have an estimate every 0.0001 x units
nest <- rangeval/0.0001

# Fit curve
tdep_curve1 <- as.data.frame(cbind(seq(-1,-0.0001,by=0.0001),
                                   rep(0,1000)))

# Section 2: 0 to 1
tdep_curve2 <- curve(-0.793*x^2 + 1.8279*x - 0.0313,
                    from=0,
                    to=1,
                    n=(1/0.0001))

# Section 3: 1+
tdep_curve3 <- curve(0.0019*x^3 - 0.0266*x^2 - 0.0142*x + 1.0333,
                    from=1,
                    to=ceiling(maxval),
                    xlab='Depth (m)',
                    ylab='SI',
                    n=((ceiling(maxval)-1) / 0.0001)
                    )

# Combine
tdep_curve <- tdep_curve3
tdep_curve$x <- c(tdep_curve1$V1, tdep_curve2$x, tdep_curve$x)
tdep_curve$y <- c(tdep_curve1$V2, tdep_curve2$y, tdep_curve$y)

# Convert to dataframe
tdep_curve <- as.data.frame(t(do.call(rbind, tdep_curve)))
row.names(tdep_curve) <- NULL

# Replace slightly over/ underestimated values
tdep_curve['y'][tdep_curve['y'] < 0] <- 0
```

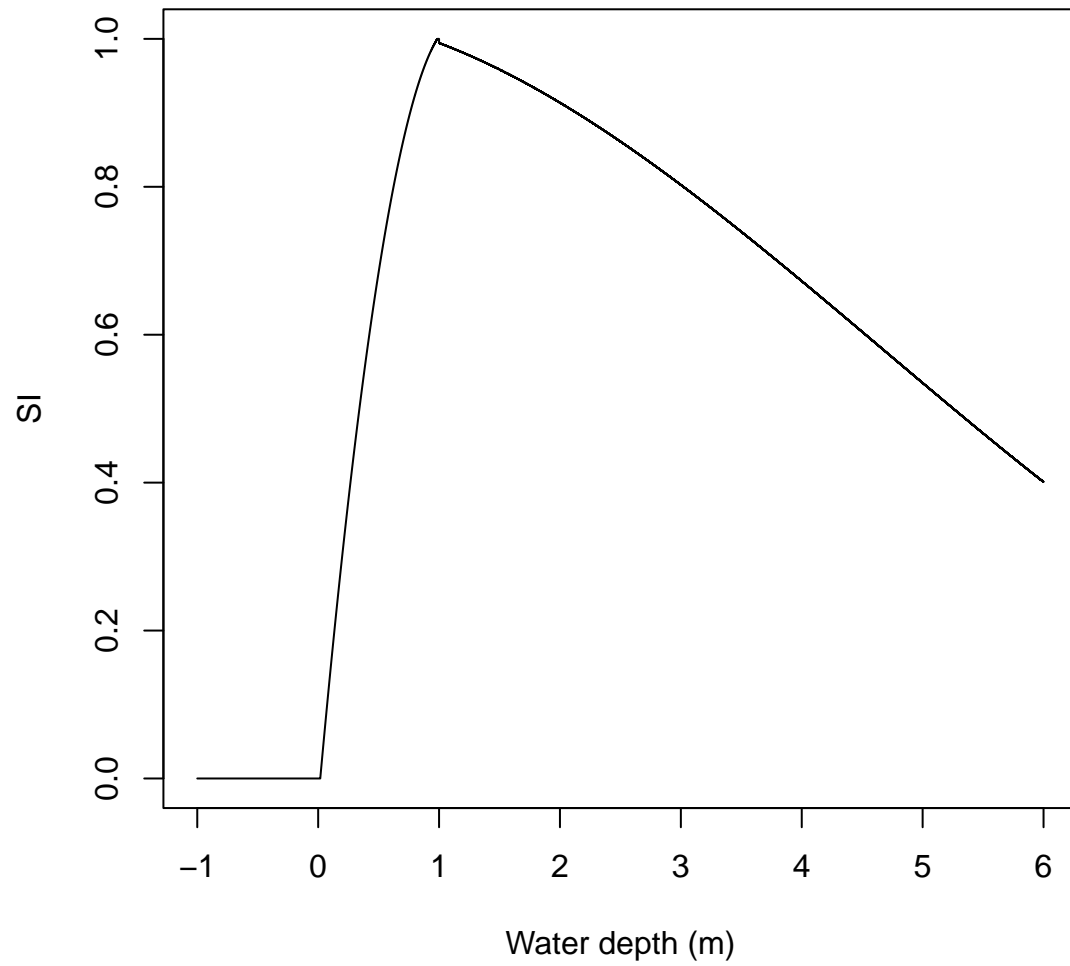
```
tdep_curve['y'][tdep_curve['y']>1] <- 1
```

```
#Plot
```

```
plot(tdep_curve$x, tdep_curve$y, type='l',
```

```
      xlab='Water depth (m)', ylab='SI')
```

```
rm(tdep_curve1, tdep_curve2, tdep_curve3, minval, maxval, nest, rangeval, val)
```



## Temperature

Priors for temperature come from SEDAR 69, which is ASMFC's January 2020 release on the stock assessment for menhaden. The relationship is for age-1+ menhaden in the coastal Atlantic, and so this is only a loose approximation. Further, estuarine environments in summer do not experience the full range of temperatures modeled by the ASMFC (0-35 °C). We will subset their temperature model to only include temperatures that we see in our model and scale the suitability index over our range of temperatures.

```
# Call temperature
val <- env[['tem']]
# Remove indexing variable
val$idx <- NULL
# Convert from matrix to vector
val <- unlist(val, use.names = F)
# Save min/ max and rounded range
minval <- min(val)
maxval <- max(val)
rangeval <- ceiling(maxval) - floor(minval)

# Calculate number of points to have an estimate every 0.0001 x units
nest <- rangeval/0.0001

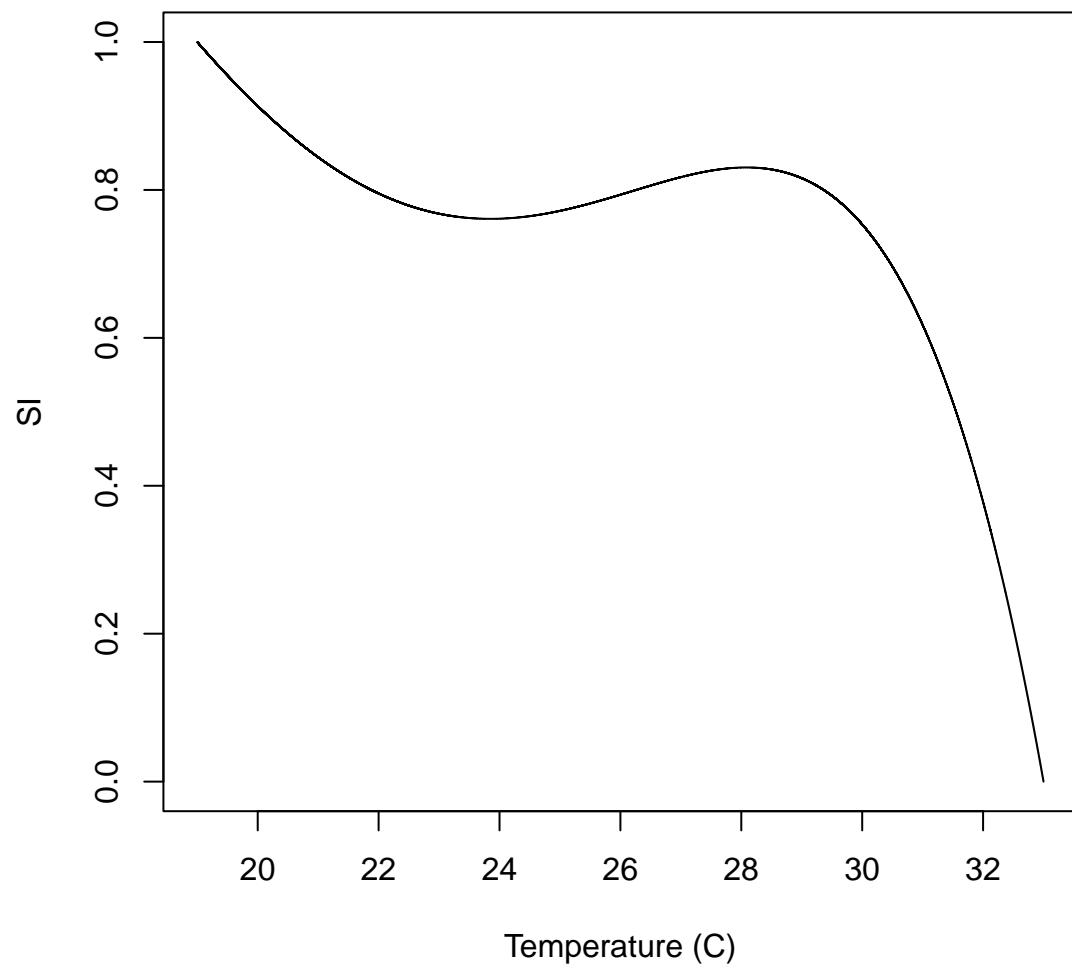
# Fit curve
tem_curve <- curve(0.2594068 - 0.04318046*x + 0.03738088*x^2 -
                   0.003929789*x^3 + 0.000146598*x^4 -
                   0.000001847609*x^5,
                   from=floor(minval),
                   to=ceiling(maxval),
                   xlab='Temperature (C)',
                   ylab='SI',
                   n=nest
                   )

# Convert to dataframe
tem_curve <- as.data.frame(t(do.call(rbind, tem_curve)))
row.names(tem_curve) <- NULL

# Rescale values
tem_curve$y <- range01(tem_curve$y)

#Plot
plot(tem_curve$x, tem_curve$y, type='l',
     xlab='Temperature (C)', ylab='SI')
rm(minval, maxval, nest, rangeval, val)
```





## Salinity

Similar to temperature, priors for salinity come from SEDAR 69. Also similar to temperature, our summer estuarine environment has a comparatively limited range of salinity. We will subset their salinity model and rescale the suitability index to cover our range of salinity values.

```
# Call salinity
val <- env[['sal']]
# Remove indexing variable
val$idx <- NULL
# Convert from matrix to vector
val <- unlist(val, use.names = F)
# Save min/ max and rounded range
minval <- min(val)
maxval <- max(val)
rangeval <- ceiling(maxval) - floor(minval)

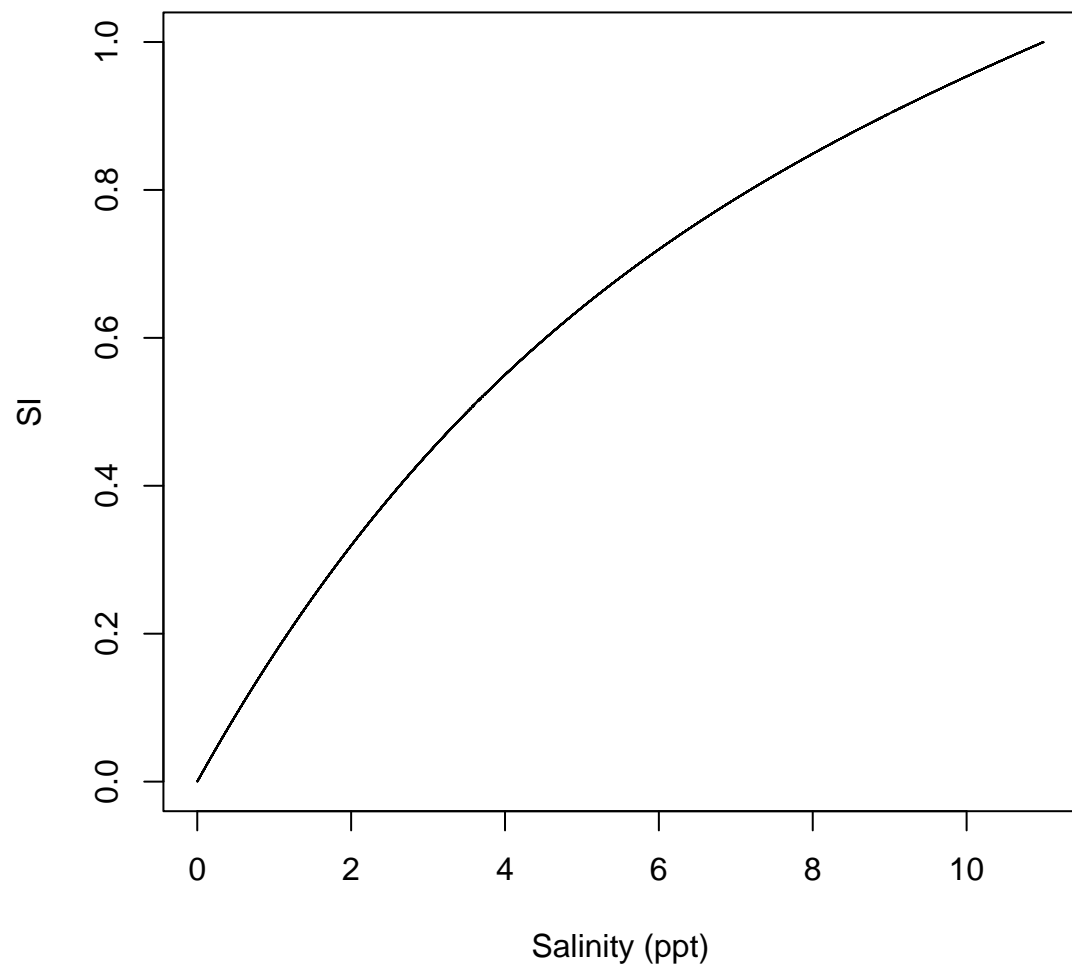
# Calculate number of points to have an estimate every 0.0001 x units
nest <- rangeval/0.0001

# Fit curve
sal_curve <- curve(8.217366 + 1.203565*x - 0.09766958*x^2 +
                   0.004988636*x^3 - 0.0001227273*x^4 +
                   0.000001071795*x^5,
                   from=floor(minval),
                   to=ceiling(maxval),
                   xlab='Salinity (ppt)',
                   ylab='SI',
                   n=nest
                   )

# Convert to dataframe
sal_curve <- as.data.frame(t(do.call(rbind, sal_curve)))
row.names(sal_curve) <- NULL

# Rescale values
sal_curve$y <- range01(sal_curve$y)

#Plot
plot(sal_curve$x, sal_curve$y, type='l',
     xlab='Salinity (ppt)', ylab='SI')
rm(minval, maxval, nest, rangeval, val)
```



## Chlorophyll

Finally, chlorophyll. There are several papers that discuss juvenile menhaden affinity for chlorophyll. Dalyander and Cerco (2010) suggest that the feeding threshold exists at 0.004 mg/L, or 4 ug/L. Lynch et al. (2010) suggest that grazing rates increase until around 200 ug/L, at which point it begins to level off. We will model the suitability of chlorophyll concentrations in the model as logarithmic relationship that increases past 4 ug/L and levels off at around 200 ug/L .

```
# Call chlorophyll
val <- env[['chl']]
# Remove indexing variable
val$idx <- NULL
# Convert from matrix to vector
val <- unlist(val, use.names = F)
# Save min/ max and rounded range
minval <- min(val)
maxval <- max(val)
rangeval <- ceiling(maxval) - floor(minval)

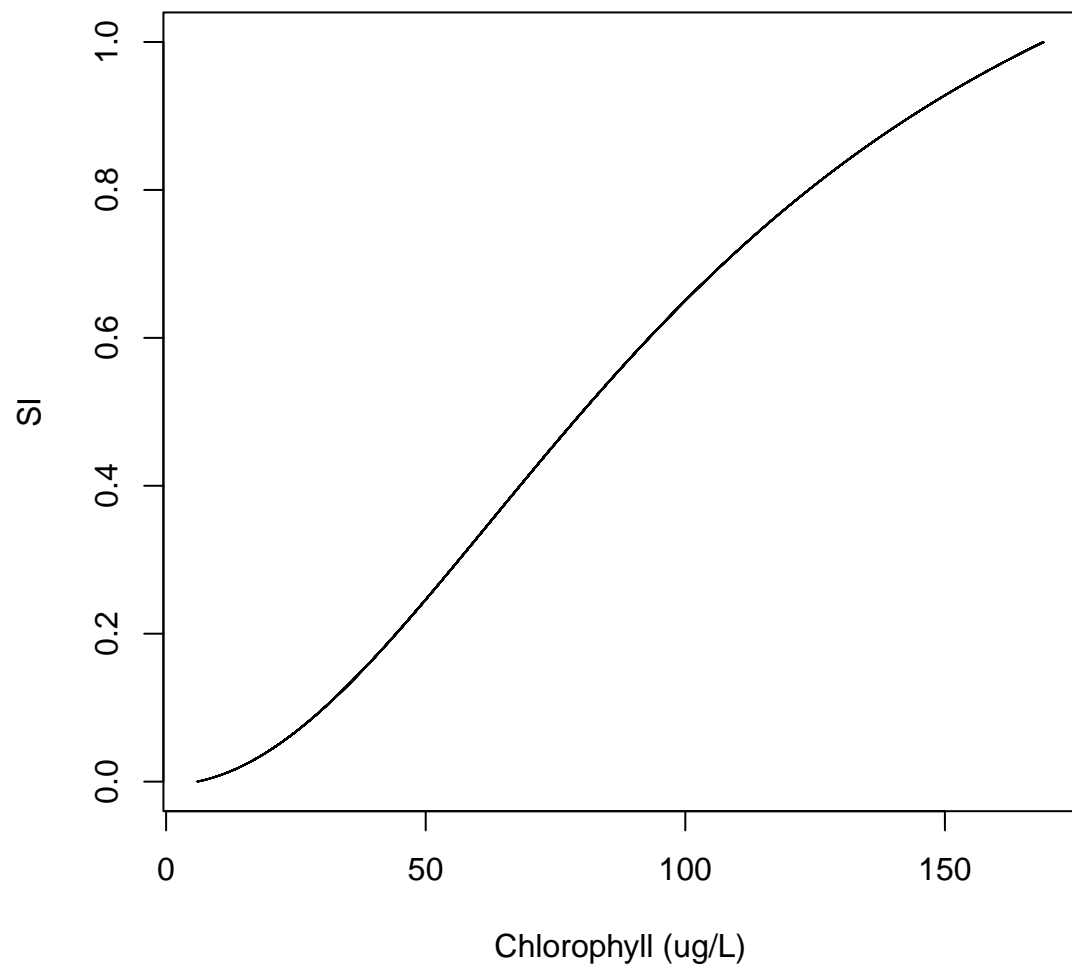
# Calculate number of points to have an estimate every 0.0001 x units
nest <- rangeval/0.0001

# Fit curve
chl_curve <- curve(1.285553 + (-0.01257163
                    - 1.285553)/(1 + (x/107.1234)^2.007354),
                  from=floor(minval),
                  to=ceiling(maxval),
                  xlab='Salinity (ppt)',
                  ylab='SI',
                  n=nest
                  )

# Convert to dataframe
chl_curve <- as.data.frame(t(do.call(rbind, chl_curve)))
row.names(chl_curve) <- NULL

# Rescale values
chl_curve$y <- range01(chl_curve$y)

#Plot
plot(chl_curve$x, chl_curve$y, type='l',
      xlab='Chlorophyll (ug/L)', ylab='SI')
rm(minval, maxval, nest, rangeval, val)
```



## Gather all curves into a list

We will collate all the curves into a single list for safekeeping.

```
# Gather into list and name
curv_list <- list(sal_curve, tem_curve, dox_curve, chl_curve,
                 tdep_curve)
names(curv_list) <- names(env)

# Remove intermediates
rm(sal_curve, tem_curve, dox_curve, chl_curve, tdep_curve)
```

## Loading environmental rasters

In the previous document, I went through the process of creating inverse distance weighted interpolated rasters of these environmental conditions through the model space at each time step. Rasters for the month of June will be loaded below.

```
# Load environmental raster lists
for(i in 1:length(env)){
  char <- names(env[i])
  char_rast <- readRDS(paste0("G:/My Drive/Documents_Backup/Modeling/",
                             "Modeling_RCode/Raster_Data/shortcut/",
                             char, "_rast_short.RDS"))
  char_rast <- assign(paste0(char, "_rast"), char_rast)
}

# Gather lists into one big list, rename
rast_list <- list(sal_rast, tem_rast, dox_rast, chl_rast, tdep_rast)
names(rast_list) <- names(env)

# Remove intermediates
rm(char_rast, char, sal_rast, tem_rast, dox_rast, chl_rast, tdep_rast)
```

## Substitution of suitability indices for environmental variables

We now have a collection of environmental rasters collated into a two-layer list. The outer layer of the list has five levels, one for each environmental variable of interest. Inside each of those levels of the outer layer are n rasters, where n represents the number of modeled time steps. Now that we have our SI curves, we need to replace the simulated values of the environmental conditions in each raster with the matching suitability index value for that environmental condition value. This code is intensive, and so will only be reproduced below without running.

```
# Set names of variables
env <- names(env)

# Set steps
steps <- length(rast_list[[1]])

for(q in 1:length(env)){
  # Call variable name
  char <- names(rast_list[q])

  # Call original raster list
  val_rast <- rast_list[[q]]

  # Call curve, round to 4 decimals
  val_curv <- curv_list[[q]]
  val_curv <- round(val_curv,4)

  # Loop and create SI for each raster
  for(i in 1:steps){
    # Call data, round to 4 decimals
    tempdata <- round(val_rast[[i]]@data@values,4)

    # Replace raster data values with matched SI values
    for(m in 1:length(tempdata)){
      newval <- tempdata[m]
      newval <- val_curv$y[val_curv$x==newval]
      # If (rarely) a single x-value is skipped, find average of neighboring values
      if(length(newval)==0){
        oneless <- tempdata[m]- 0.0001
        onemore <- tempdata[m]+ 0.0001
        valless <- val_curv$y[val_curv$x==oneless]
        valmore <- val_curv$y[val_curv$x==onemore]
        newval <- mean(valless, valmore)
        # Remove intermediates
        rm(oneless, onemore, valless, valmore)
      }
      tempdata[m] <- newval
    }

    # Replace values in raster with HSI values
    val_rast[[i]]@data@values <- tempdata

    # Rename as needed
```

```

names(val_rast[[i]]) <- char

# Remove intermediate
rm(tempdata, newval)

}
# Save output
setwd("G:/My Drive/Documents_Backup/Modeling/Modeling_RCode/Raster_Data/HSI")
saveRDS(val_rast, file=paste0(char, '_hsi_rast.rds'))
rm(char, val_rast, val_curv)
}

```



## Overall Habitat suitability index

Ideas from this section came from Fabrizio et al. 2021, where habitat suitability for several species was modeled in the entire Chesapeake Bay given data from an unTRIM three-dimensional hydrodynamic model. There are two methods to consider for calculating the overall habitat suitability index. The first is a simple arithmetic mean:

$$HSI_{am} = \frac{EV_1 + EV_2 + \dots + EV_p}{p}$$

where  $EV_1$  is the suitability for environmental variable 1, etc, and  $p$  is the total number of environmental variables considered.

| The second method is a geometric mean, which applies a limiting factor so that a single poor environmental variable can result in a low overall habitat suitability. This is given by:

$$HSI_{gm} = \sqrt[p]{EV_1 \times EV_2 \times \dots \times EV_p}$$

The geometric mean method may be more appropriate for us— extremely poor SI values for a single environmental variable, like water depth, would realistically “tank” the overall habitat suitability index in that raster cell.

We will now calculate the habitat suitability index for each raster cell using the geometric mean method. This will need to be done at each time step. Again, the code is simply reproduced here due to computational strain.

```
# Set names of variables
env <- names(env)

# Load SI rasters
for(i in 1:length(env)){
  char <- env[i]
  char_rast <- readRDS(paste0("G:/My Drive/Documents_Backup/Modeling/",
                              "Modeling_RCode/Raster_Data/HSI/short/",
                              char, "_hsi_rast_short.RDS"))
  char_rast <- assign(paste0(char, "_si"), char_rast)
}

# Gather SI rasters into list
SI_list <- list(sal_si, tem_si, dox_si, chl_si, tdep_si)
names(SI_list) <- env

# Remove intermediates
rm(sal_si, tem_si, dox_si, chl_si, tdep_si)

# Allocate blank list
HSI_rast <- vector(mode="list", length=length(SI_list[[1]]))

# Loop through SI rasters to create HSI
for(i in 1:length(SI_list[[1]])){
  # Overlay all rasters and take geometric mean HSI per cell
  HSI_rast[[i]] <- overlay(SI_list[['sal']][[i]],
                          SI_list[['tem']][[i]],
```

```
        SI_list[['dox']][[i]],
        SI_list[['chl']][[i]],
        SI_list[['tdep']][[i]],
        fun=psych::geometric.mean)

    print(i)
}

# Save results
setwd("G:/My Drive/Documents_Backup/Modeling/Modeling_RCode/Raster_Data/HSI/results")
saveRDS(HSI_rast, file='HSI_rast_short_gm.rds')
```

## Conclusion

These rasters are the final result of the pre-processing. We can now feed these results into the individual-based model of juvenile menhaden movement in which the menhaden sense and respond to their surrounding environment. A selection of HSI rasters will be animated in a GIF to visualize spatio-temporal suitability trends.