



Oct 15, 2018

Working

Assessing the potential for competition between Pacific Halibut (*Hippoglossus stenolepis*) and Arrowtooth Flounder (*Atheresthes stomias*) in the Gulf of Alaska

Cheryl Barnes¹, Anne Beaudreau¹, Mary Hunsicker², Lorenzo Ciannelli³

¹College of Fisheries and Ocean Sciences, University of Alaska Fairbanks, ²Fish Ecology Division, Northwest Fisheries Science Center, National Marine Fisheries Service, National Oceanic and Atmospheric Administration, ³College of Earth, Ocean, and Atmospheric Sciences, Oregon State University

dx.doi.org/10.17504/protocols.io.uieeube



Cheryl Barnes

University of Alaska Fairbanks



ABSTRACT

Pacific Halibut (*Hippoglossus stenolepis*) support culturally and economically important fisheries in the Gulf of Alaska, though recent decreases in mean size-at-age have substantially reduced fishery yields, generating concerns among stakeholders and resource managers. Among the prevailing hypotheses for reduced size-at-age is intensified competition with Arrowtooth Flounder (*Atheresthes stomias*), a groundfish predator that exhibited nearly five-fold increases in biomass between the 1960s and mid-2010s. To assess the potential for competition between Pacific Halibut and Arrowtooth Flounder, we evaluated their degree of spatiotemporal and dietary overlap in the Gulf of Alaska using bottom trawl survey and food habits data provided by the Alaska Fisheries Science Center (NOAA; 1990 to 2017). We restricted analyses to fish measuring 30 to 69 cm fork length and used a delta modeling approach to quantify species-specific presence-absence and catch-per-unit-effort as a function of survey year, tow location, depth, and bottom temperature. We then calculated an index of spatial overlap across a uniform grid by multiplying standardized predictions of species' abundance. Dietary overlap was calculated across the same uniform grid using Schoener's similarity index. Finally, we assessed the relationship between spatial and dietary overlap as a measure of resource partitioning. We found increases in spatial overlap, moving from east to west in the Gulf of Alaska (eastern: 0.13 ± 0.20 ; central: 0.21 ± 0.11 ; western: 0.31 ± 0.13 SD). Dietary overlap was low throughout the study area (0.13 ± 0.20 SD). There was no correlation between spatial and dietary overlap, suggesting an absence of resource partitioning along the niche dimensions examined. This finding provides little indication that competition with Arrowtooth Flounder was responsible for changes in Pacific Halibut size-at-age in the Gulf of Alaska; however, it does not rule out competitive interactions that may have affected resource use prior to standardized data collection or at different spatiotemporal scales.

EXTERNAL LINK

<https://github.com/cheryl-barnes/ResourcePartitioning>

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Barnes, C.L., A.H. Beaudreau, M.E. Hunsicker, and L. Ciannelli (In Review). Assessing the potential for competition between Pacific Halibut (*Hippoglossus stenolepis*) and Arrowtooth Flounder (*Atheresthes stomias*) in the Gulf of Alaska. PLOS ONE.

PROTOCOL STATUS

Working

We use this protocol in our group and it is working.

MATERIALS TEXT

These files detail analyses used to estimate the degree of resource partitioning (as a proxy for the potential for competition) between Pacific Halibut and Arrowtooth Flounder (30 to 69 cm fork length) in the Gulf of Alaska.

Data Sources

We used standardized survey data procured from the Alaska Fisheries Science Center (National Marine Fisheries Service, National Oceanic and Atmospheric Association [NOAA]). Bottom trawl survey data (1990 to 2017) were collected by the Resource Assessment and Conservation Engineering (RACE) Division and are publicly accessible here:

https://www.afsc.noaa.gov/RACE/groundfish/survey_data/data.htm. Food habits data (1990 to 2013) were provided by the Resource Ecology and Ecosystem Modeling (REEM) Program and are publicly accessible here:

<https://access.afsc.noaa.gov/REEM/WebDietData/DietDataIntro.php>. All the data necessary to complete the following analyses can be found in the 'Data' folder. See von Szalay and Raring (2016) and Livingston et al. (2017) for data collection methods.

Spatial Overlap

The 'SpatialAnalyses' script file includes the code necessary to construct a delta model for estimating spatial overlap between Pacific Halibut and Arrowtooth Flounder in the Gulf of Alaska. Methods were modified from Hunsicker et al. (2013) and Shelton et al. (2017).

We used generalized additive models to quantify and predict the probability of occurrence and relative abundance of Pacific Halibut and Arrowtooth Flounder across a uniform grid spanning the study area. We multiplied probabilities of occurrence and relative abundances to estimate overall abundance in each survey year-grid cell combination. We then multiplied standardized abundances of each species to estimate spatial overlap in each combination of survey year and grid cell.

Dietary Overlap

The 'DietaryAnalyses' script file includes the code necessary to calculate species-specific proportions of prey by weight in each survey year and grid cell. We then used proportions of prey by weight to calculate Schoener's index of dietary overlap.

Resource Partitioning

The 'ResourcePartitioningAnalyses' script file combines spatial overlap and dietary overlap to quantify the correlation between the two measures and thus the degree of resource partitioning between Pacific Halibut and Arrowtooth Flounder in the Gulf of Alaska. Descriptions of resource partitioning can be found in Schoener (1974) and Ross (1986).

CoAuthors

Anne H. Beaudreau - College of Fisheries and Ocean Sciences | University of Alaska Fairbanks

Mary E. Hunsicker - Fish Ecology Division, Northwest Fisheries Science Center | National Marine Fisheries Service (NOAA)

Lorenzo Ciannelli - College of Earth, Ocean, and Atmospheric Sciences | Oregon State University

References

Chipps SR, Garvey JE. Assessment of diets and feeding patterns. In: Analysis and interpretation of freshwater fisheries data. Guy CS, Brown ML, editors. Bethesda: American Fisheries Society; 2007. pp. 473–514.

Hunsicker ME, Ciannelli L, Bailey KM, Zador S, Stige L. Climate and demography dictate the strength of predator-prey overlap in a subarctic marine ecosystem. PLOS ONE. 2013;8(6):e66025. doi:10.1371/journal.pone.006602

Livingston PA, Aydin K, Buckley TW, Lang GM, Yang MS, Miller BS. Quantifying food web interactions in the North Pacific – a data-based approach. Environ Biol Fishes. 2017;100(4):443–470.

Ross ST. Resource partitioning in fish assemblage: review of field studies. Copeia. 1986;1986(2):352–388.

Schoener TW. Resource partitioning in ecological communities. Science. 1974;185(4145):27–39.

Shelton AO, Hunsicker ME, Ward EJ, Feist BE, Blake R, Ward CL, et al. Spatio-temporal models reveal subtle changes to demersal communities following the Exxon Valdez oil spill. ICES J Mar Sci. 2017. doi: 10.1093/icesjms/fsx079

von Szalay PG, Raring NW. Data report: 2015 Gulf of Alaska bottom trawl survey. Seattle (WA): National Oceanic and Atmospheric Administration; 2016. Technical Memorandum: NMFS-AFSC-325. Sponsored by the US Department of Commerce.

Acknowledgments

Wayne Palsson and Kirstin Holsman queried and provided guidance regarding the use of AFSC databases. Analytical

recommendations from Franz Mueter and Jordan Watson were incorporated throughout.

Spatial Analyses - R Script File

```
1 # This script file includes code necessary to construct a delta GAM (generalized additive model) for estimating spatial overlap
  # between Pacific Halibut and Arrowtooth Flounder in the Gulf of Alaska. The methods used were modified from Hunsicker et al. (2013)
  # and Shelton et al. (2017).

  # We analyzed standardized survey data procured from the Alaska Fisheries Science Center (National Marine Fisheries Service,
  # National Oceanic and Atmospheric Association [NOAA]). Bottom trawl survey data (1990 to 2017) were collected by the Resource
  # Assessment and Conservation Engineering (RACE) Division and are publicly accessible here:
  # https://www.afsc.noaa.gov/RACE/groundfish/survey\_data/data.htm.

  # We used generalized additive models to quantify and predict the probability of occurrence and relative abundance of Pacific
  # Halibut and Arrowtooth Flounder across a uniform grid spanning the study area. We multiplied probabilities of occurrence and relative
  # abundances to estimate overall abundance in each survey year-grid cell combination. We then multiplied standardized abundances of
  # each species to estimate spatial overlap in each combination of survey year and grid cell.

  # Rcode developed by: Cheryl Barnes
  # cheryl.barnes@alaska.edu
  # Preliminary code (e.g., exploratory data analyses, sample size calculations, summary statistics, diagnostics) have been excluded.

  # References:
  # Hunsicker ME, Ciannelli L, Bailey KM, Zador S, Stige L. Climate and demography dictate the strength of predator-prey overlap in a
  # subarctic marine ecosystem. PLOS ONE. 2013;8(6):e66025. doi:10.1371/journal.pone.006602
  # Livingston PA, Aydin K, Buckley TW, Lang GM, Yang MS, Miller BS. Quantifying food web interactions in the North Pacific – a data-
  # based approach. Environ Biol Fishes. 2017;100(4):443–470.
  # Shelton AO, Hunsicker ME, Ward EJ, Feist BE, Blake R, Ward CL, et al. Spatio-temporal models reveal subtle changes to demersal
  # communities following the Exxon Valdez oil spill. ICES J Mar Sci. 2017. doi: 10.1093/icesjms/fsx079
  # von Szalay PG, Raring NW. Data report: 2015 Gulf of Alaska bottom trawl survey. Seattle (WA): National Oceanic and Atmospheric
  # Administration; 2016. Technical Memorandum: NMFS-AFSC-325. Sponsored by the US Department of Commerce.
  #####
  rm(list=ls())
  graphics.off()

  setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/")
  #####
  ### INITIAL DATA PREPARATION ###
  #####
  # Prepare and format AFSC bottom trawl survey data:
  # These data include all survey tows conducted between 1984 and 2017, including those with and without our species of interest.
  trawl = read.csv("Data/AFSC_TrawlData_1984_2017.csv")

  # Relabel species codes: 10110 = Arrowtooth Flounder (ATF), 10120 = Pacific Halibut (PH), 20510 = Sablefish (SBL), 21720 =
  # Pacific Cod (PC), 21740 = Walleye Pollock (WEP)
  trawl$SPECIES_CODE = as.factor(trawl$SPECIES_CODE)
  trawl$Species = trawl$SPECIES_CODE
  levels(trawl$Species) = list(ATF="10110", PH="10120", SBL="20510", PC="21720", WEP="21740")

  # Manually assign International North Pacific Fisheries Commission statistical areas (i.e., values) based on survey strata (i.e., index).
  # Note: The second number from the right corresponds with individual statistical areas (e.g., STRATUM 120 = StatArea 620, STRATUM
  # 251 = StatArea 650).
  index = c(unique(trawl$STRATUM))
  values = c("650", "650", "650", "650", "650", "640", "640", "640", "640", "640", "640", "640", "640", "620", "640", "620",
    "640", "610", "650", "610", "650", "610", "610", "610", "610", "610", "610", "610", "610", "620", "620",
    "620", "620", "630", "630", "630", "630", "630", "630", "630", "630", "640", "640", "630", "630", "630",
    "630", "630", "620", "620", "620", "620", "630", "630", "630", "650", "650")
  trawl$StatArea = values[match(trawl$STRATUM, index)]
  table(trawl$STRATUM, trawl$StatArea) # Check

  # Exclude data from 1984 and 1987 (survey methods were standardized in 1990):
  trawl = subset(trawl, YEAR!=1984)
```

```

trawl = subset(trawl, YEAR!=1987)

# Treat survey year as a factor for all models:
trawl$YEAR = as.factor(trawl$YEAR)

# Create unique haul identifier by concatenating VESSEL, CRUISE, and HAUL (allows for joining to proportional length data below):
trawl$Haul_Join = paste(trawl$VESSEL, trawl$CRUISE, trawl$HAUL, sep="")

# Convert data from long to wide format (allows for joining to proportional length data below):
require(reshape2)
trawl_wide = dcast(trawl, YEAR + HAULJOIN + VESSEL + CRUISE + Haul_Join + STRATUM + DISTANCE_FISHED + NET_WIDTH +
  STATIONID + START_LATITUDE + START_LONGITUDE + GEAR_DEPTH + GEAR_TEMPERATURE + StatArea ~ Species,
  value.var="NUMCPUE")

# Remove other species (i.e., PC, SBL, and WEP) from data frame:
trawl_wide = trawl_wide[1:16]
#####
### ADJUST HAUL-SPECIFIC CPUE TO MATCH SIZE CLASS OF INTEREST ###
#####
# Adjust haul-specific CPUE estimates (number per hectare) to reflect species-specific proportions of fish measuring within the size
class of interest (30 to 69 cm fork length). Note: 100 to 200 fish were subsampled for length measurements per haul. Length data
were provided by the RACE Division, Alaska Fisheries Science Center (NMFS, NOAA) upon request. We reduced the size of the
following CSV file to meet size limitations imposed by GitHub (i.e., unnecessary columns were removed prior to import).
lengths = read.csv("Data/race_length_by_haul_PH_ATFred.csv", header=T)

# Convert fork length units from mm to cm:
lengths$Length.cm. = lengths$Length.mm./10

# Exclude data from 1984 and 1987:
lengths = subset(lengths, Year!=1984)
lengths = subset(lengths, Year!=1987)

# Create unique haul identifier by concatenating VESSEL, CRUISE, and HAUL (allows for joining to trawl data prepared above):
lengths$Haul_Join = paste(lengths$Vessel.Number, lengths$Cruise.Number, lengths$Haul.Number, sep="")

# Calculate total number of fish observed for each fork length:
require(splitstackshape)
LFreq = expandRows(lengths, "Frequency")

# Limit data to size classes of interest (30 to 69 cm):
LFreq_red = subset(LFreq, Length.cm. > 29)
LFreq_red = subset(LFreq_red, Length.cm. < 70)

# Calculate species-specific number of fish per haul:
# Separate out Pacific Halibut (PH):
PH_all = subset(LFreq, SPECIES=="PH") # all lengths
PH_red = subset(LFreq_red, SPECIES=="PH") # 30-69 cm FL only
# Calculate total number (all size classes) of PH, by haul:
PH_1 = table(PH_all$Haul_Join, PH_all$SPECIES)
PH_1 = as.data.frame(PH_1)
PH_1 = subset(PH_1, Var2=="PH")
# Calculate haul-specific number of PH measuring 30-69 cm:
PH_2 = table(PH_red$Haul_Join, PH_red$SPECIES)
PH_2 = as.data.frame(PH_2)
PH_2 = subset(PH_2, Var2=="PH")
# Calculate proportion of haul measuring 30 to 69 cm:
PHprop = merge(PH_1, PH_2, by="Var1", all=TRUE)
PHprop = PHprop[,c(1:3,5)]
colnames(PHprop) = c("Haul_Join", "Species", "PH_All", "PH_30_69")
PHprop[is.na(PHprop)] = 0
PHprop$PHprop = PHprop$PH_30_69 / PHprop$PH_All

# Separate out Arrowtooth Flounder (ATF):
ATF_all = subset(LFreq, SPECIES=="ATF") # all lengths
ATF_red = subset(LFreq_red, SPECIES=="ATF") # 30-69 cm FL only

```

```

# Calculate total number (all size classes) of ATF, by haul:
ATF_1 = table(ATF_all$Haul_Join, ATF_all$SPECIES)
ATF_1 = as.data.frame(ATF_1)
ATF_1 = subset(ATF_1, Var2=="ATF")
# Calculate haul-specific number of ATF measuring 30-69 cm:
ATF_2 = table(ATF_red$Haul_Join, ATF_red$SPECIES)
ATF_2 = as.data.frame(ATF_2)
ATF_2 = subset(ATF_2, Var2=="ATF")
# Calculate proportion of haul measuring 30 to 69 cm:
ATFprop = merge(ATF_1, ATF_2, by="Haul_Join", all=TRUE)
ATFprop = ATFprop[,c(1:3,5)]
colnames(ATFprop) = c("Haul_Join", "Species", "ATF_All", "ATF_30_69")
ATFprop[is.na(ATFprop)] = 0
ATFprop$ATFprop = ATFprop$ATF_30_69 / ATFprop$ATF_All

# Join bottom trawl survey and proportional length data:
require(dplyr)
Lengthprop = merge(ATFprop, PHprop, by = "Haul_Join", all = TRUE)
trawl_wide_30_69 = trawl_wide %>% left_join(Lengthprop)

# Remove unnecessary columns:
trawl_wide_30_69 = trawl_wide_30_69[,c(1:16,18:20,22:24)]
# Replace NAs with zeros:
trawl_wide_30_69[,17:22][is.na(trawl_wide_30_69[,17:22])] = 0

# Calculate adjusted CPUE (number per ha) based on proportional catch of 30 to 69 cm size class:
trawl_wide_30_69$adjCPUE_ATF = trawl_wide_30_69$ATF * trawl_wide_30_69$ATFprop
trawl_wide_30_69$adjCPUE_PH = trawl_wide_30_69$PH * trawl_wide_30_69$PHprop
#####
### MODEL FITTING AND PLOTTING ###
#####
require(mgcv)
require(lme4)
require(MuMIn)
options(na.action = "na.fail")
require(sp)
require(maps)
require(mapdata)
require(visreg)
require(ggplot2)
require(PBSmapping)

# Set coordinate boundaries for plotting:
lonmin = -172
lonmax = -130
latmin = 52
latmax = 62

# Ensure that the same tows are included in each model by removing those with incomplete environmental data (i.e., rows with missing
depths or bottom temperatures):
trawl_comp = subset(trawl_wide_30_69, !is.na(GEAR_DEPTH))
trawl_comp = subset(trawl_comp, !is.na(GEAR_TEMPERATURE))

#####
### Model presence-absence (P/A) for Pacific Halibut ###

# Label each haul as being present or absent for P. Halibut:
trawl_comp$PHpa = as.numeric(trawl_comp$adjCPUE_PH > 0)

# Remove the extreme outlying station in considerably deep water:
trawl_compSub = subset(trawl_comp, Haul_Join!="148201101201")

# Run the full (global) model:
PH.pa.gam_full = gam(PHpa ~ YEAR + s(START_LONGITUDE, START_LATITUDE) + s(GEAR_DEPTH) + s(GEAR_TEMPERATURE, k=4),
data = trawl_compSub, family = binomial(link=logit), method="GCV.Cp")

```

```
summary(PH.pa.gam_full)
```

```
# Generate all possible alternative models and select the best-fit based on delta AIC:
```

```
PH.pa.gam_select = dredge(PH.pa.gam_full, beta=FALSE, evaluate=TRUE, rank="AIC", trace=FALSE)
```

```
print(PH.pa.gam_select, abbrev.names=FALSE, warnings=TRUE)
```

```
summary(PH.pa.gam_select)
```

```
# The full model = the best-fit model.
```

```
PH.pa.gam_best = PH.pa.gam_full
```

```
summary(PH.pa.gam_best)
```

```
### Plot P/A results, Pacific Halibut ###
```

```
# Plot the partial effect of latxlon on P/A of P. Halibut:
```

```
data(worldHiresMapEnv) # source world data for plot
```

```
vis.gam(PH.pa.gam_best, c("START_LONGITUDE", "START_LATITUDE"), plot.type = "contour", type="response", contour.col="black",  
color="heat", xlab="Longitude", ylab="Latitude", main="Pacific Halibut, Partial Effect on Presence or Absence", too.far=0.025,
```

```
n.grid=250, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))
```

```
maps::map('worldHires', fill=T, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax), add=T, col="lightgrey")
```

```
# Plot the partial effect of survey year on P/A of P. Halibut:
```

```
visreg(fit=PH.pa.gam_best, xvar="YEAR", band=TRUE, partial=FALSE, rug=FALSE, line=list(col="red"), fill=list(col="gray", alpha=0.2),  
points=list(col="lightgray", alpha=0.5, cex=0.25), trans=binomial()$linkinv, type="conditional", gg=TRUE, labels=c("1990", "", "96", "",  
"01", "", "05", "", "09", "", "13", "", "2017")) +
```

```
theme_bw() +
```

```
ggtitle("Pacific Halibut") +
```

```
theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
```

```
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
```

```
theme(axis.text.y = element_text(family="Arial", size=14)) +
```

```
theme(axis.text.x = element_text(family="Arial", size=14)) +
```

```
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
```

```
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
```

```
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
```

```
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
```

```
labs(x="GEAR_DEPTH (m)", y="Partial Effect on Presence (1) or Absence (0)") +
```

```
scale_y_continuous(limits=c(0.30, 1.0), breaks=c(0.30, 0.4, 0.50, 0.6, 0.7, 0.8, 0.9, 1.0)) +
```

```
theme(legend.background = element_rect(fill="transparent"))
```

```
# Plot the partial effect of depth on P/A of P. Halibut:
```

```
visreg(PH.pa.gam_best, xvar="GEAR_DEPTH", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),  
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=binomial()$linkinv, type="conditional", gg=TRUE) +
```

```
theme_bw() +
```

```
ggtitle("Pacific Halibut") +
```

```
theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
```

```
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
```

```
theme(axis.text.y = element_text(family="Arial", size=14)) +
```

```
theme(axis.text.x = element_text(family="Arial", size=14)) +
```

```
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
```

```
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
```

```
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
```

```
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
```

```
labs(x="GEAR_DEPTH (m)", y="Partial Effect on Presence (1) or Absence (0)") +
```

```
scale_x_continuous(expand=c(0,0), limits=c(0,1000), breaks=c(0,150,300,450,600,750,900)) +
```

```
theme(legend.background = element_rect(fill="transparent")) +
```

```
geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")
```

```
# Plot the partial effect of bottom temperature on P/A of P. Halibut:
```

```
visreg(PH.pa.gam_best, xvar="GEAR_TEMPERATURE", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),  
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=binomial()$linkinv, type="conditional", gg=TRUE) +
```

```
theme_bw() +
```

```
ggtitle("Pacific Halibut") +
```

```
theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
```

```
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
```

```
theme(axis.text.y = element_text(family="Arial", size=14)) +
```

```
theme(axis.text.x = element_text(family="Arial", size=14)) +
```

```

theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
labs(x=expression(paste("Bottom Temperature (",degree,"C)")), y="Partial Effect on Presence (1) or Absence (0)") +
theme(legend.background = element_rect(fill="transparent")) +
scale_x_continuous(expand=c(0,0), limits=c(0,13.5), breaks=c(0,3,6,9,12)) +
scale_y_continuous(limits=c(0,1), breaks=c(0,0.25, 0.50, 0.75, 1)) +
geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")
#####
### Model CPUE (where present) for Pacific Halibut ###

# Subset hauls to include only those that caught P. Halibut:
PH = subset(trawl_comp, adjCPUE_PH > 0)
# Remove the extreme outlying station in considerably deep water:
PH = subset(PH, Haul_Join!="148201101201")
PH$logPH = log(PH$adjCPUE_PH) # log-transform

# Run the full (global) model:
PH.cpue.gam_full = gam(logPH ~ YEAR + s(START_LONGITUDE, START_LATITUDE) + s(GEAR_DEPTH) + s(GEAR_TEMPERATURE,
k=4), data = PH, family = gaussian(link=identity), method="GCV.Cp")
summary(PH.cpue.gam_full)

# Generate all possible alternative models and select the best-fit model based on delta AIC:
PH.cpue.gam_select = dredge(PH.cpue.gam_full, beta=FALSE, evaluate=TRUE, rank="AIC", trace=FALSE)
print(PH.cpue.gam_select, abbrev.names=FALSE, warnings=TRUE)
summary(PH.cpue.gam_select)

# The full model = the best-fit model.
PH.cpue.gam_best = PH.cpue.gam_full
summary(PH.cpue.gam_best)

# Plot the partial effect of latlon on CPUE of P. Halibut:
vis.gam(PH.cpue.gam_best, c("START_LONGITUDE", "START_LATITUDE"), plot.type = "contour", type="response",
contour.col="black", color="heat", xlab="Longitude", ylab="Latitude", main="Pacific Halibut, Partial Effect on log-CPUE (number per
hectare)", too.far=0.025, n.grid=500, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))
maps::map('worldHires', fill=T, add=T, col="lightgrey")

# Plot the partial effect of survey year on CPUE of P. Halibut:
visreg(PH.cpue.gam_best, xvar="YEAR", line=list(col="red"), fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5,
cex=0.25), band=TRUE, partial=FALSE, rug=FALSE, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
theme_bw() +
ggtitle("Pacific Halibut") +
theme(plot.title = element_text(size=18, hjust = 0.5)) +
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
theme(axis.text.y = element_text(family="Arial", size=14)) +
theme(axis.text.x = element_text(family="Arial", size=14)) +
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
labs(x="", y="Partial Effect on log-CPUE (number per hectare)") +
scale_y_continuous(limits=c(4.5, 6.0), breaks=c(4.5, 5.0, 5.5, 6.0)) +
theme(legend.background = element_rect(fill="transparent"))

# Plot the partial effect of depth on CPUE of P. Halibut:
visreg(fit=PH.cpue.gam_best, xvar="GEAR_DEPTH", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
theme_bw() +
ggtitle("Pacific Halibut") +
theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
theme(axis.text.y = element_text(family="Arial", size=14)) +
theme(axis.text.x = element_text(family="Arial", size=14)) +
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +

```



```

theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
labs(x="GEAR_DEPTH (m)", y="Partial Effect on log-CPUE (number per hectare)") +
scale_x_continuous(expand=c(0,0), limits=c(0,1000), breaks=c(0,150,300,450,600,750,900)) +
scale_y_continuous(limits=c(2,8), breaks=c(2,4,6,8)) +
theme(legend.background = element_rect(fill="transparent")) +
geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")

# Plot the partial effect of bottom temperature on CPUE of P. Halibut:
visreg(PH.cpue.gam_best, xvar="GEAR_TEMPERATURE", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5,
cex=0.25), line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
theme_bw() +
ggtitle("Pacific Halibut") +
theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
theme(axis.text.y = element_text(family="Arial", size=14)) +
theme(axis.text.x = element_text(family="Arial", size=14)) +
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
scale_x_continuous(expand=c(0,0), limits=c(0,13.5), breaks=c(0,3,6,9,12)) +
scale_y_continuous(limits=c(4.5,7), breaks=c(5,6,7)) +
labs(x=expression(paste("Bottom Temperature (", degree, "C)")), y="Partial Effect on log-CPUE (number per hectare)") +
theme(legend.background = element_rect(fill="transparent")) +
geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")

#####
### Model presence-absence (P/A) for Arrowtooth Flounder ###

# Label each haul as being present or absent for Arrowtooth:
trawl_comp$ATFpa = as.numeric(trawl_comp$adjCPUE_ATF > 0)

# Run the full (global) model:
ATF.pa.gam_full = gam(ATFpa ~ YEAR + s(START_LONGITUDE, START_LATITUDE) + s(GEAR_DEPTH) + s(GEAR_TEMPERATURE, k=4),
data = trawl_comp, family = binomial(link=logit), method="GCV.Cp")
summary(ATF.pa.gam_full)

# Generate all possible alternative models and select the best-fit model based on delta AIC:
ATF.pa.gam_select = dredge(ATF.pa.gam_full, beta=FALSE, evaluate=TRUE, rank="AIC", trace=FALSE)
print(ATF.pa.gam_select, abbrev.names=FALSE, warnings=TRUE)
summary(ATF.pa.gam_select)

# The full model = the best-fit model.
ATF.pa.gam_best = ATF.pa.gam_full
summary(ATF.pa.gam_best)

# Plot the partial effect of latlon on P/A of Arrowtooth Flounder:
vis.gam(ATF.pa.gam_best, c("START_LONGITUDE", "START_LATITUDE"), plot.type = "contour", type="response",
contour.col="black", color="heat", xlab="Longitude", ylab="Latitude", main="Arrowtooth Flounder, Partial Effect on Presence or
Absence", too.far=0.025, n.grid=500, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))
maps::map('worldHires', fill=T, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax), add=T, col="lightgrey")

# Plot the partial effect of survey year on P/A of Arrowtooth Flounder:
visreg(fit=ATF.pa.gam_best, xvar="YEAR", band=TRUE, partial=FALSE, rug=FALSE, line=list(col="red"), fill=list(col="gray", alpha=0.2),
points=list(col="lightgray", alpha=0.5, cex=0.25), trans=binomial()$linkinv, type="conditional", gg=TRUE) +
theme_bw() +
ggtitle("Arrowtooth Flounder") +
theme(plot.title = element_text(size=18, hjust = 0.5)) +
theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
theme(axis.text.y = element_text(family="Arial", size=14)) +
theme(axis.text.x = element_text(family="Arial", size=14)) +
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +

```



```

theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
theme(axis.title.y = element_text(vjust=1.0, size=14)) +
labs(x="", y="Partial effect on Presence (1) or Absence (0)") +
scale_y_continuous(limits=c(0.30,1.0), breaks=c(0.30, 0.4, 0.50, 0.6, 0.7, 0.8, 0.9, 1.0)) +
theme(legend.background = element_rect(fill="transparent"))

```

Plot the partial effect of depth on P/A of Arrowtooth Flounder:

```

visreg(ATF.pa.gam_best, xvar="GEAR_DEPTH", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=binomial()$linkinv, type="conditional", gg=TRUE) +
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(axis.text.y = element_text(family="Arial", size=14)) +
  theme(axis.text.x = element_text(family="Arial", size=14)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.0, size=14)) +
  labs(x="GEAR_DEPTH (m)", y="Partial Effect on Presence (1) or Absence (0)") +
  scale_x_continuous(expand=c(0,0), limits=c(0,1000), breaks=c(0,150,300,450,600,750,900)) +
  theme(legend.background = element_rect(fill="transparent")) +
  geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")

```

Plot the partial effect of bottom temperature on P/A of Arrowtooth Flounder:

```

visreg(ATF.pa.gam_best, xvar="GEAR_TEMPERATURE", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=binomial()$linkinv, type="conditional", gg=TRUE) +
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(axis.text.y = element_text(family="Arial", size=14)) +
  theme(axis.text.x = element_text(family="Arial", size=14)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.0, size=14)) +
  labs(x=expression(paste("Bottom Temperature (", degree, "C)")), y="Partial Effect on Presence (1) or Absence (0)") +
  theme(legend.background = element_rect(fill="transparent")) +
  scale_x_continuous(expand=c(0,0), limits=c(0,13.5), breaks=c(0,3,6,9,12)) +
  scale_y_continuous(limits=c(0,1), breaks=c(0,0.25, 0.50, 0.75, 1)) +
  geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")

```

```

#####
### Model CPUE (where present) for Arrowtooth Flounder ###

```

Subset hauls to include only those that caught Arrowtooth:

```

ATF = subset(trawl_comp, adjCPUE_ATF > 0)
ATF$logATF = log(ATF$adjCPUE_ATF) # log-transform

```

Run the full (global) model:

```

ATF.cpue.gam_full = gam(logATF ~ YEAR + s(START_LONGITUDE, START_LATITUDE) + s(GEAR_DEPTH) + s(GEAR_TEMPERATURE,
k=4), data = ATF, family = gaussian(link=identity), method="GCV.Cp")
summary(ATF.cpue.gam_full)

```

Generate all possible alternative models and select best-fit model based on delta AIC:

```

ATF.cpue.gam_select = dredge(ATF.cpue.gam_full, beta=FALSE, evaluate=TRUE, rank="AIC", trace=FALSE)
print(ATF.cpue.gam_select, abbrev.names=FALSE, warnings=TRUE)
summary(ATF.cpue.gam_select)

```

The full model = the best-fit model.

```

ATF.cpue.gam_best = ATF.cpue.gam_full
summary(ATF.cpue.gam_best)

```

Plot the partial effect of latxlon on CPUE, Arrowtooth Flounder:

```
vis.gam(ATF.cpue.gam_best, c("START_LONGITUDE", "START_LATITUDE"), plot.type = "contour", type="response",
contour.col="black", color="heat", xlab="Longitude", ylab="Latitude", main="Arrowtooth Flounder, Partial Effect on log-CPUE (number
per hectare)", too.far=0.025, n.grid=500, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))
maps::map('worldHires', fill=T, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax), add=T, col="lightgrey")
```

Plot the partial effect of survey year on CPUE of Arrowtooth Flounder:

```
visreg(ATF.cpue.gam_best, xvar="YEAR", line=list(col="red"), fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5,
cex=0.25), band=TRUE, partial=FALSE, rug=FALSE, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(size=18, hjust = 0.5)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(axis.text.y = element_text(family="Arial", size=14)) +
  theme(axis.text.x = element_text(family="Arial", size=14)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.0, size=14)) +
  labs(x="", y="Partial Effect on log-CPUE (number per hectare)") +
  theme(legend.background = element_rect(fill="transparent"))
```

Plot the partial effect of depth on CPUE of Arrowtooth Flounder:

```
visreg(ATF.cpue.gam_best, xvar="GEAR_DEPTH", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(axis.text.y = element_text(family="Arial", size=14)) +
  theme(axis.text.x = element_text(family="Arial", size=14)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.0, size=14)) +
  labs(x="GEAR_DEPTH (m)", y="Partial Effect on log-CPUE (number per hectare)") +
  scale_x_continuous(expand=c(0,0), limits=c(0,1000), breaks=c(0,150,300,450,600,750,900)) +
  scale_y_continuous(limits=c(2,10), breaks=c(2,4,6,8,10)) +
  theme(legend.background = element_rect(fill="transparent")) +
  geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")
```

Plot the partial effect of bottom temperature on CPUE of Arrowtooth Flounder:

```
visreg(ATF.cpue.gam_best, "GEAR_TEMPERATURE", fill=list(col="gray", alpha=0.2), points=list(col="lightgray", alpha=0.5, cex=0.25),
line=list(col="red"), band=TRUE, partial=FALSE, rug=1, trans=gaussian()$linkinv, type="conditional", gg=TRUE) +
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5, family="Arial", size=18)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(axis.text.y = element_text(family="Arial", size=14)) +
  theme(axis.text.x = element_text(family="Arial", size=14)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.0, size=14)) +
  labs(x=expression(paste("Bottom Temperature (", degree, "C)")), y="Partial Effect on log-CPUE (number per hectare)") +
  theme(legend.background = element_rect(fill="transparent")) +
  scale_x_continuous(expand=c(0,0), limits=c(0,13.5), breaks=c(0,3,6,9,12)) +
  scale_y_continuous(limits=c(7,10.7), breaks=c(7,8,9,10)) +
  geom_rug(sides="b", alpha=0.2, size=0.5, position="jitter")
```

```
#####
```

```
### MODEL PREDICTIONS ###
```

```
#####
```

Create a uniform grid spanning the bottom trawl survey area to make predictions (to grid cell centers):

```
require(dplyr)
require(tidyr)
require(sp)
```

```

require(raster)
require(rgeos)
require(rgbif)
require(viridis)
require(gridExtra)
require(rasterVis)
require(purrr)
require(mapproj)
require(devtools)
require(stringr)
require(maptools)
require(rgdal)
require(PBSmapping)
require(mapdata)
require(ggplot2)
require(ggmap)

# Establish boundaries of the uniform grid:
# Read in and prepare INPFC Stat Area Shapefile (610 to 650). Note: Need to dissolve INPFC boundaries for creation of grid within.
setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/Data/")
INPFC = readOGR(".", "GOA_Shapes")
INPFC_Pr = spTransform(INPFC, CRS("+proj=utm +zone=5, +datum=WGS84 +units=km +no_defs"))

INPFCdata = data.frame()
INPFCdata = rbind(INPFCdata, INPFC_Pr@data)
INPFCdata$OBJECTID = as.character(INPFCdata$OBJECTID)
INPFCdata$REP_AREA = as.character(INPFCdata$REP_AREA)
INPFCdata$Region = NA
INPFCdata$Region = "GOA"

INPFC_Pr@data$OBJECTID = as.character(INPFC_Pr@data$OBJECTID)
INPFC_Pr@data = full_join(INPFC_Pr@data, INPFCdata, by = "REP_AREA")
row.names(INPFC_Pr) = row.names(INPFC_Pr@data)
INPFC_Pr = spChFIDs(INPFC_Pr, row.names(INPFC_Pr))
INPFC_Pr = gUnaryUnion(INPFC_Pr, id = INPFC_Pr@data$Region)
row.names(INPFC_Pr) = as.character(1:length(INPFC_Pr))

INPFCdata = unique(INPFCdata$Region)
INPFCdata = as.data.frame(INPFCdata)
colnames(INPFCdata) = "Region"
INPFC_Pr = SpatialPolygonsDataFrame(INPFC_Pr, INPFCdata)
# plot(INPFC_Pr)

# Set size (km) for (square) grid cells. Note: Large size is necessary for aggregating sparse diet data.
my.interval=100

# INPFC SAMPLING DESIGN:
# 5x5 km grids = individual stations
# tows ~ 1.5 km distance swept

# Select range of coordinates for grid boundaries (UTM to maintain constant grid cell area regardless of geographic location).
lonmin = -875
lonmax = 1975
latmin = 5075
latmax = 7975

# Compile series of points for grid:
mygrd = expand.grid(
  LON = seq(lonmin, lonmax, by=my.interval),
  LAT = seq(latmin, latmax, by=my.interval)) %>%
  mutate(my.z=1:n()) %>%
  data.frame

# Convert mygrd to spatial dataframe:
coordinates(mygrd) = ~ LON + LAT

```

```

# Convert SpatialPoints to SpatialPixelsDataFrame:
mygrd = (as(SpatialPixelsDataFrame(mygrd, mygrd@data, tolerance=.00086), "SpatialPolygonsDataFrame"))
# plot(mygrd)

# Project, clip, and reproject mygrid to INPFC stat areas:
proj4string(mygrd) = proj4string(INPFC_Pr)
INPFC_Pr = gBuffer(INPFC_Pr, byid=TRUE, width=0)
mygrd = gBuffer(mygrd, byid=TRUE, width=0)
clip_INPFC = gIntersection(INPFC_Pr, mygrd, byid = TRUE, drop_lower_td = TRUE)
proj4string(clip_INPFC) = proj4string(INPFC_Pr)
# plot(clip_INPFC, col="grey")

# Convert to data frame (in decimal degrees) for ggplot:
clip2_INPFC = clip_INPFC
clip2_INPFC = spTransform(clip2_INPFC, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"))
clip2_INPFC = fortify(clip2_INPFC)

setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/")
write.csv(clip2_INPFC, "Data/clip2_INPFC.csv")
# clip2_INPFC = read.csv("Data/clip2_INPFC.csv")
##### # Calculate mean depth and bottom
temperature for each grid cell and survey year for use in model predictions:
trawl_enviro = trawl

# Convert bottom trawl survey data to spatial data frame:
clipTrawl = clip_INPFC
clipTrawl = spTransform(clipTrawl, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs "))
coordinates(trawl_enviro) = c("START_LONGITUDE", "START_LATITUDE")
proj4string(trawl_enviro) = proj4string(clipTrawl)

# Find where data overlap with the clipped grid and convert back to data frame:
tempdat = data.frame(myrows=names(over(trawl_enviro, clipTrawl)), mygrid=over(trawl_enviro, clipTrawl))
trawl_enviro$id2 = over(trawl_enviro, clipTrawl)
trawl_enviro = as.data.frame(trawl_enviro)

# Create data frame with id values and grid cell coordinates:
mycenter_INPFC = as.data.frame(gCentroid(clip_INPFC, byid=TRUE)) %>%
  mutate(id2=1:n(),
    EEZgrid=row.names())
colnames(mycenter_INPFC)[3] = "id"
xy=mycenter_INPFC[,c(1,2)]
mycenter_INPFC_sp = SpatialPointsDataFrame(coords=xy, data=mycenter_INPFC, proj4string=CRS("+proj=utm +zone=5,
+datum=WGS84 +units=km +no_defs"))
mycenter_INPFC_Pr = spTransform(mycenter_INPFC_sp, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs "))
mycenter_INPFC_df = as.data.frame(mycenter_INPFC_Pr)
names(mycenter_INPFC_df) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "x.lon", "y.lat")

# Join trawl and grid cell data (and clean up):
mycenter_INPFC_S = mycenter_INPFC_df %>% left_join(trawl_enviro)
mycenter_INPFC_S = unique(mycenter_INPFC_S)

# Summarize data by grid cell ("id2")
trawl_enviroDepth = subset(mycenter_INPFC_S, GEAR_DEPTH>=0)
meanDepth = trawl_enviroDepth %>%
  group_by(YEAR, id2) %>%
  summarize(meanDepth = mean(GEAR_DEPTH))

trawl_enviroBT = subset(mycenter_INPFC_S, GEAR_TEMPERATURE>=0)
meanBT = trawl_enviroBT %>%
  group_by(YEAR, id2) %>%
  summarize(meanBT = mean(GEAR_TEMPERATURE))

HaulCentEnviroData = merge(trawl_enviroDepth, trawl_enviroBT)
coordinates(HaulCentEnviroData) = ~ x.lon + y.lat

```

```

proj4string(HaulCentEnviroData) = proj4string(mycenter_INPFC_sp)
HaulCentEnviroData = spTransform(HaulCentEnviroData, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs "))
HaulCentEnviroData = as.data.frame(HaulCentEnviroData)
HaulCentEnviroData = HaulCentEnviroData[,c(1:4,6,22:25,27:28)]
HaulCentEnviroData = unique(HaulCentEnviroData)

write.csv(HaulCentEnviroData, "Data/GridCellPredictions.csv")
# GridCellPredictions = read.csv("Data/GridCellPredictions.csv")
#####
# Use GAM results to predict distributions, abundances, and spatial overlap for Pacific Halibut and Arrowtooth Flounder:

### Predict Presence-Absence, PACIFIC HALIBUT ###
PHpa_predictGAM = predict.gam(PH.pa.gam_best, newdata=HaulCentEnviroData, type="response", se.fit=TRUE)
PHpa_predGAM = cbind(HaulCentEnviroData, PHpa_predictGAM)

# Calculate 95% confidence intervals:
PHpa_pred_Clgam = within(PHpa_predGAM, {
  lower = fit-1.96*se.fit
  upper = fit+1.96*se.fit
})
colnames(PHpa_pred_Clgam) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "YEAR", "START_LATITUDE", "START_LONGITUDE",
"GEAR_DEPTH", "GEAR_TEMPERATURE", "StatArea", "Haul_Join", "PHpa_fit", "PHpa_se.fit", "PHpa_lowerCI", "PHpa_upperCI")

### CPUE, PACIFIC HALIBUT ###
PHcpue_predictGAM = predict.gam(PH.cpue.gam_best, newdata=HaulCentEnviroData, type="response", se.fit=TRUE)
PHcpue_predGAM = cbind(HaulCentEnviroData, PHcpue_predictGAM)

# Calculate 95% confidence intervals:
PHcpue_pred_Clgam = within(PHcpue_predGAM, {
  lower = fit-1.96*se.fit
  upper = fit+1.96*se.fit
})
colnames(PHcpue_pred_Clgam) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "YEAR", "START_LATITUDE", "START_LONGITUDE",
"GEAR_DEPTH", "GEAR_TEMPERATURE", "StatArea", "Haul_Join", "PHcpue_fit", "PHcpue_se.fit", "PHcpue_lowerCI",
"PHcpue_upperCI")

PHpredictionsGAM = PHpa_pred_Clgam %>% left_join(PHcpue_pred_Clgam)

# Calculate relative abundance, accounting for presence-absence:
PHpredictionsGAM$PHpredAbun = PHpredictionsGAM$PHpa_fit * PHpredictionsGAM$PHcpue_fit

# Standardize by maximum predicted abundance (almost identical results as standardizing by mean or median, though on desired
scale):
PHpredictionsGAM$PHstdAbun = PHpredictionsGAM$PHpredAbun/max(PHpredictionsGAM$PHpredAbun)
#####
### Predict Presence-Absence, ARROWTOOTH FLOUNDER ###
ATFpa_predictGAM = predict.gam(ATF.pa.gam_best, newdata=HaulCentEnviroData, type="response", se.fit=TRUE)
ATFpa_predGAM = cbind(HaulCentEnviroData, ATFpa_predictGAM)

# Calculate 95% confidence intervals:
ATFpa_pred_Clgam = within(ATFpa_predGAM, {
  lower = fit-1.96*se.fit
  upper = fit+1.96*se.fit
})

colnames(ATFpa_pred_Clgam) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "YEAR", "START_LATITUDE", "START_LONGITUDE",
"GEAR_DEPTH", "GEAR_TEMPERATURE", "StatArea", "Haul_Join", "ATFpa_fit", "ATFpa_se.fit", "ATFpa_lowerCI", "ATFpa_upperCI")

### CPUE, ARROWTOOTH FLOUNDER ###
ATFcpue_predictGAM = predict.gam(ATF.cpue.gam_best, newdata=HaulCentEnviroData, type="response", se.fit=TRUE)
ATFcpue_predGAM = cbind(HaulCentEnviroData, ATFcpue_predictGAM)

# Calculate 95% confidence intervals:
ATFcpue_pred_Clgam = within(ATFcpue_predGAM, {
  lower = fit-1.96*se.fit

```

```

upper = fit+1.96*se.fit
})

colnames(ATFcpue_pred_Clgam) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "YEAR", "START_LATITUDE", "START_LONGITUDE",
"GEAR_DEPTH", "GEAR_TEMPERATURE", "StatArea", "Haul_Join", "ATFcpue_fit", "ATFcpue_se.fit", "ATFcpue_lowerCI",
"ATFcpue_upperCI")

ATFpredictionsGAM = ATFpa_pred_Clgam %>% left_join(ATFcpue_pred_Clgam)

# Calculate relative abundance, accounting for presence-absence:
ATFpredictionsGAM$ATFpredAbun = ATFpredictionsGAM$ATFpa_fit * ATFpredictionsGAM$ATFcpue_fit

# Standardize by maximum predicted abundance (almost identical results as standardizing by mean or median, though on desired
scale):
ATFpredictionsGAM$ATFstdAbun = ATFpredictionsGAM$ATFpredAbun/max(ATFpredictionsGAM$ATFpredAbun)
#####
# Merge PH and ATF predictions:
PH_ATF_spatialGAM = PHpredictionsGAM %>% left_join(ATFpredictionsGAM)

# Remove grid cells with std abundances < 0.25 for PH and ATF - suggesting low habitat suitability for large-bodied flatfishes:
PH_ATF_spatialGAMred = PH_ATF_spatialGAM[which(PH_ATF_spatialGAM$PHstdAbun >= 0.25 | PH_ATF_spatialGAM$ATFstdAbun >=
0.25),]
# Calculate spatial overlap for each survey year-grid cell combination:
PH_ATF_spatialGAMred$$ = PH_ATF_spatialGAMred$PHstdAbun * PH_ATF_spatialGAMred$ATFstdAbun
PH_ATF_spatialGAMred$$ = round(PH_ATF_spatialGAMred$$, digits=3)
#####
### PLOT RESULTS ###
#####
# Load map data and set new grid boundaries (DD):
data(nepacLLhigh)
lonmin = -172.5
lonmax = -129.5
latmin = 49
latmax = 62

# Clip maps to predetermined boundaries (this will take a few moments to complete):
world = fortify(nepacLLhigh)
world2 = clipPolys(world, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/Data/")
Canada = raster::getData("GADM", country = "CAN", level = 0)
Canada = fortify(Canada)
names(Canada) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
Canada$PID = as.numeric(Canada$PID)
Canada = clipPolys(Canada, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

INPFC_shape = readOGR(".", "GOA_Shapes")
INPFC_shape = spTransform(INPFC_shape, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"))
INPFC_plot = fortify(INPFC_shape)
names(INPFC_plot) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
INPFC_plot$PID = as.numeric(INPFC_plot$PID)
INPFC_plot = clipPolys(INPFC_plot, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

# Adjustment labels:
INPFC_cent = coordinates(INPFC_shape)
INPFC_cent = INPFC_cent[c(1, 3:6),]
INPFC_cent = as.data.frame(INPFC_cent)
names(INPFC_cent) = c("START_LONGITUDE", "START_LATITUDE")
rownames(INPFC_cent) = c()
INPFC_cent$StatArea = NA
INPFC_cent[1,3] = "SE"
INPFC_cent[2,3] = "Shumagin"
INPFC_cent[3,3] = "Chirikof"
INPFC_cent[4,3] = "Kodiak"
INPFC_cent[5,3] = "Yakutat"

```

```

INPFC_cent$StatArea = as.factor(INPFC_cent$StatArea)

# Shift labels to not overlap with grid cell objects:
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Shumagin"] =
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Shumagin"] - 0.4
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Chirikof"] =
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Chirikof"] - 1.25
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Kodiak"] = INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Kodiak"]
- 1.25
INPFC_cent$START_LONGITUDE[INPFC_cent$StatArea=="Kodiak"] =
INPFC_cent$START_LONGITUDE[INPFC_cent$StatArea=="Kodiak"] + 0.75
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Yakutat"] = INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="Yakutat"]
- 0.5
INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="SE"] = INPFC_cent$START_LATITUDE[INPFC_cent$StatArea=="SE"] - 1.25
INPFC_cent$START_LONGITUDE[INPFC_cent$StatArea=="SE"] = INPFC_cent$START_LONGITUDE[INPFC_cent$StatArea=="SE"] -
0.55

# Read in and prepare IPHC regulatory area layer:
IPHC_shape = readOGR(".", "GOA_Den")
IPHC_shape = spTransform(IPHC_shape, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"))
IPHC_2C = subset(IPHC_shape, REG_AREA=="2C")
IPHC_3A = subset(IPHC_shape, REG_AREA=="3A")
IPHC_3B = subset(IPHC_shape, REG_AREA=="3B")
IPHC_4A = subset(IPHC_shape, REG_AREA=="4A")

Area2C = gUnaryUnion(IPHC_2C, IPHC_2C@data$REG_Area)
Area2C_df = fortify(Area2C)
row.names(Area2C_df) = c()
Area2C_df = subset(Area2C_df, group=="1.1")
names(Area2C_df) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
Area2C_df$PID = as.numeric(Area2C_df$PID)
Area2C_df = clipPolys(Area2C_df, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

Area3A = gUnaryUnion(IPHC_3A, IPHC_3A@data$REG_Area)
Area3A_df = fortify(Area3A)
row.names(Area3A_df) = c()
Area3A_df = subset(Area3A_df, group=="1.1")
names(Area3A_df) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
Area3A_df$PID = as.numeric(Area3A_df$PID)
Area3A_df = clipPolys(Area3A_df, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

Area3B = gUnaryUnion(IPHC_3B, IPHC_3B@data$REG_Area)
Area3B_df = fortify(Area3B)
row.names(Area3B_df) = c()
Area3B_df = subset(Area3B_df, group=="1.1")
names(Area3B_df) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
Area3B_df$PID = as.numeric(Area3B_df$PID)
Area3B_df = clipPolys(Area3B_df, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

Area4A = gUnaryUnion(IPHC_4A, IPHC_4A@data$REG_Area)
Area4A_df = fortify(Area4A)
row.names(Area4A_df) = c()
names(Area4A_df) = c("X", "Y", "POS", "hole", "piece", "id", "PID")
Area4A_df$PID = as.numeric(Area4A_df$PID)
Area4A_df = clipPolys(Area4A_df, xlim=c(lonmin, lonmax), ylim=c(latmin, latmax))

setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/")
# Calculate mean estimates for each grid cell (all years combined):
stdPH = PH_ATF_spatialGAMred[c("id2", "EEZ grid", "PHstdAbun")]
stdAbunPH = stdPH %>%
  group_by(id2) %>%
  mutate(stdAbun = mean(PHstdAbun))
stdAbunPH = unique(stdAbunPH[c(1:2,4)])

stdATF = PH_ATF_spatialGAMred[c("id2", "EEZ grid", "ATFstdAbun")]

```



```

stdAbunATF = stdATF %>%
  group_by(id2) %>%
  mutate(stdAbun = mean(ATFstdAbun))
stdAbunATF = unique(stdAbunATF[c(1:2,4)])

spatial_Grid = PH_ATF_spatialGAMred[c("YEAR", "id2", "EEZ grid", "START_LONGITUDE", "START_LATITUDE", "S")]
S_Grid = spatial_Grid %>%
  group_by(YEAR, id2) %>%
  mutate(S = mean(S))
S_Grid = as.data.frame(unique(S_Grid))

# Write CSV for analyses of resource partitioning:
write.csv(S_Grid, "Data/PH_ATF_S.csv")
# S_overlap = read.csv("Data/PH_ATF_S.csv")

spatial = PH_ATF_spatialGAMred[c("id2", "EEZ grid", "S")]
S_overlap = spatial %>%
  group_by(id2) %>%
  mutate(meanS = mean(S))
S_overlap = as.data.frame(unique(S_overlap[c(1:2,4)]))

# Join summary information and spatial data:
goa.df = fortify(clip2_INPFC, region='id')
colnames(goa.df)[colnames(goa.df) == "id"] = "EEZ grid"
plot_PHstdAbun = goa.df %>% right_join(stdAbunPH)
plot_ATFstdAbun = goa.df %>% right_join(stdAbunATF)
plot_PH_ATF_S = goa.df %>% right_join(S_overlap)

# Plot mean values by grid cell:
IPHC_2C = data.frame(text = c("2C"))
IPHC_3A = data.frame(text = c("3A"))
IPHC_3B = data.frame(text = c("3B"))
IPHC_4A = data.frame(text = c("4A"))
textStdAbun = data.frame(text = c("Std. Abundance"))
textS = data.frame(text = c("Spatial Overlap"))
#####
### Standardized Abundance, PACIFIC HALIBUT ###
PHstdAbun = ggplot() +
  geom_polygon(data=plot_PHstdAbun, aes(x=long, y=lat, group=group, fill=stdAbun), col="black", lwd = 0.25) +
  geom_polygon(data=Area2C_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="cadetblue1",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area3A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="deepskyblue2",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area3B_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="blue",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area4A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="midnightblue",
  linetype="solid", lwd=1) +
  scale_fill_distiller(palette="Spectral", name="", limits=c(0,1), breaks=c(0.00,0.25,0.50,0.75,1.00)) +
  geom_polygon(data=INPFC_plot, aes(x=X, y=Y, group=factor(PID)), fill=NA, col="black", lwd=0.25) +
  geom_polygon(data=world2, aes(x=X, y=Y, group=factor(PID)), fill="lightgrey", col="black", lwd=0.25) +
  geom_polygon(data=Canada, aes(x=X, y=Y, group=factor(PID)), fill="gray91", col="black", lwd=0.25) +
  geom_text(data=textStdAbun, aes(label=text, x=-132.55, y=61.65, size=12), show.legend=FALSE) +
  geom_text(data=INPFC_cent, aes(group=StatArea, label=StatArea, x=START_LONGITUDE, y=START_LATITUDE, size=12),
  show.legend = FALSE) +
  geom_text(data=IPHC_2C, aes(label=text, x=-134.84, y=56.06, size=12), col="paleturquoise1", fontface="bold", show.legend =
  FALSE) +
  geom_text(data=IPHC_3A, aes(label=text, x=-145.40, y=59.9, size=12), col="deepskyblue3", fontface="bold", show.legend = FALSE)
+
  geom_text(data=IPHC_3B, aes(label=text, x=-157.39, y=55.65, size=12), col="blue", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_4A, aes(label=text, x=-167.75, y=53.05, size=12), col="midnightblue", fontface="bold", show.legend = FALSE)
+
  theme_bw() +
  ggtitle("Pacific Halibut") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +

```

```

theme(legend.position = c(0.957, 0.830)) +
theme(plot.title = element_text(hjust = 0.5, size=14)) +
theme(legend.title = element_blank()) +
theme(legend.text = element_text(family="Arial", size=12), legend.text.align = 1) +
theme(legend.background = element_rect(fill="transparent")) +
theme(legend.key = element_blank()) +
theme(legend.key.width = unit(6.0, "mm")) +
theme(legend.key.height = unit(8.0, "mm")) +
theme(legend.spacing.x = unit(2.0, "mm")) +
theme(axis.text.y = element_text(family="Arial", size=12)) +
theme(axis.text.x = element_text(family="Arial", size=12)) +
theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
theme(axis.title.x = element_text(hjust=0.53, size=12)) +
theme(axis.title.y = element_text(hjust=0.46, size=12)) +
labs(x="Longitude", y="Latitude") +
scale_x_continuous(limits = c(lonmin, lonmax), expand = c(0,0)) +
scale_y_continuous(limits = c(latmin, latmax), expand = c(0,0), labels = function(x) round(as.numeric(x), digits=0)) +
theme(legend.background = element_rect(fill="transparent"))

PHstdAbun
ggsave(filename="Plots/PHstdAbun.png", plot=PHstdAbun, dpi=500, width=12, height=8, units="in")
#####
### Standardized Abundance, ARROWTOOTH FLOUNDER ###
ATFstdAbun = ggplot() +
  geom_polygon(data=plot_ATFstdAbun, aes(x=long, y=lat, group=group, fill=stdAbun), col="black", lwd = 0.25) +
  geom_polygon(data=Area2C_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="cadetblue1",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area3A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="deepskyblue2",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area3B_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="blue",
  linetype="solid", lwd=1) +
  geom_polygon(data=Area4A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="midnightblue",
  linetype="solid", lwd=1) +
  scale_fill_distiller(palette="Spectral", name="", limits=c(0,1), breaks=c(0.00,0.25,0.50,0.75,1.00)) +
  geom_polygon(data=INPFC_plot, aes(x=X, y=Y, group=factor(PID)), fill=NA, col="black", lwd=0.25) +
  geom_polygon(data=world2, aes(x=X, y=Y, group=factor(PID)), fill="lightgrey", col="black", lwd=0.25) +
  geom_polygon(data=Canada, aes(x=X, y=Y, group=factor(PID)), fill="gray91", col="black", lwd=0.25) +
  geom_text(data=textStdAbun, aes(label=text, x=-132.55, y=61.65, size=12), show.legend=FALSE) +
  geom_text(data=INPFC_cent, aes(group=StatArea, label=StatArea, x=START_LONGITUDE, y=START_LATITUDE, size=12),
  show.legend = FALSE) +
  geom_text(data=IPHC_2C, aes(label=text, x=-134.84, y=56.06, size=12), col="cadetblue2", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_3A, aes(label=text, x=-145.40, y=59.9, size=12), col="deepskyblue3", fontface="bold", show.legend = FALSE)
+
  geom_text(data=IPHC_3B, aes(label=text, x=-157.39, y=55.65, size=12), col="blue", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_4A, aes(label=text, x=-167.75, y=53.05, size=12), col="midnightblue", fontface="bold", show.legend = FALSE)
+
  theme_bw() +
  ggtitle("Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(legend.position = c(0.957, 0.830)) +
  theme(plot.title = element_text(hjust = 0.5, size=14)) +
  theme(legend.title = element_blank()) +
  theme(legend.text = element_text(family="Arial", size=12), legend.text.align = 1) +
  theme(legend.background = element_rect(fill="transparent")) +
  theme(legend.key = element_blank()) +
  theme(legend.key.width = unit(6.0, "mm")) +
  theme(legend.key.height = unit(8.0, "mm")) +
  theme(legend.spacing.x = unit(2.0, "mm")) +
  theme(axis.text.y = element_text(family="Arial", size=12)) +
  theme(axis.text.x = element_text(family="Arial", size=12)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(hjust=0.53, size=12)) +

```

```

theme(axis.title.y = element_text(hjust=0.46, size=12)) +
labs(x="Longitude", y="Latitude") +
scale_x_continuous(limits = c(lonmin, lonmax), expand = c(0,0)) +
scale_y_continuous(limits = c(latmin, latmax), expand = c(0,0), labels = function(x) round(as.numeric(x), digits=0)) +
theme(legend.background = element_rect(fill="transparent"))

ATFstdAbun
ggsave(filename="Plots/ATFstdAbun.png", plot=ATFstdAbun, dpi=500, width=12, height=8, units="in")
#####
### Spatial Overlap, PACIFIC HALIBUT and ARROWTOOTH FLOUNDER ###
PH_ATF_Splot = ggplot() +
  geom_polygon(data=plot_PH_ATF_S, aes(x=long, y=lat, group=group, fill=meanS), col="black", lwd=0.25) +
  geom_polygon(data=Area2C_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="cadetblue1",
linetype="solid", lwd=1) +
  geom_polygon(data=Area3A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="deepskyblue2",
linetype="solid", lwd=1) +
  geom_polygon(data=Area3B_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="blue",
linetype="solid", lwd=1) +
  geom_polygon(data=Area4A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="midnightblue",
linetype="solid", lwd=1) +
  scale_fill_distiller(palette="Spectral", name="", limits=c(0,1), breaks=c(0.00,0.25,0.50,0.75,1.00)) +
  geom_polygon(data=INPFC_plot, aes(x=X, y=Y, group=factor(PID)), fill=NA, col="black", linetype="solid", lwd=0.25) +
  geom_polygon(data=world2, aes(x=X, y=Y, group=factor(PID)), fill="lightgrey", col="black", lwd=0.25) +
  geom_polygon(data=Canada, aes(x=X, y=Y, group=factor(PID)), fill="gray91", col="black", lwd=0.25) +
  geom_text(data=textS, aes(label=text, x=-132.48, y=61.65, size=12), show.legend=FALSE) +
  geom_text(data=INPFC_cent, aes(group=StatArea, label=StatArea, x=START_LONGITUDE, y=START_LATITUDE, size=12),
show.legend = FALSE) +
  geom_text(data=IPHC_2C, aes(label=text, x=-134.84, y=56.06, size=12), col="paleturquoise1", fontface="bold", show.legend =
FALSE) +
  geom_text(data=IPHC_3A, aes(label=text, x=-145.40, y=59.9, size=12), col="deepskyblue3", fontface="bold", show.legend = FALSE)
+
  geom_text(data=IPHC_3B, aes(label=text, x=-157.39, y=55.65, size=12), col="blue", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_4A, aes(label=text, x=-167.75, y=53.05, size=12), col="midnightblue", fontface="bold", show.legend = FALSE)
+
  theme_bw() +
  ggtitle("Spatial Overlap between Pacific Halibut and Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(legend.position = c(0.957, 0.830)) +
  theme(plot.title = element_text(hjust = 0.5, size=14)) +
  theme(legend.title = element_blank()) +
  theme(legend.text = element_text(family="Arial", size=12), legend.text.align = 1) +
  theme(legend.background = element_rect(fill="transparent")) +
  theme(legend.key = element_blank()) +
  theme(legend.key.width = unit(6.0, "mm")) +
  theme(legend.key.height = unit(8.0, "mm")) +
  theme(legend.spacing.x = unit(2.0, "mm")) +
  theme(axis.text.y = element_text(family="Arial", size=12)) +
  theme(axis.text.x = element_text(family="Arial", size=12)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(hjust=0.53, size=12)) +
  theme(axis.title.y = element_text(hjust=0.46, size=12)) +
  labs(x="Longitude", y="Latitude") +
  scale_x_continuous(limits = c(lonmin, lonmax), expand = c(0,0)) +
  scale_y_continuous(limits = c(latmin, latmax), expand = c(0,0), labels = function(x) round(as.numeric(x), digits=0)) +
  theme(legend.background = element_rect(fill="transparent"))

PH_ATF_Splot
ggsave(filename="Plots/S_PH_ATF.png", plot=PH_ATF_Splot, dpi=500, width=12, height=8, units="in")
#####
### TEST FOR SPATIOTEMPORAL CHANGES IN SPATIAL OVERLAP ###
#####
# Run ANCOVAs to test for relationships among spatial overlap, survey year, and INPFC statistical area or IPHC regulatory area:

```

```

# Prepare overlap data:
spatial_Grid = unique(PH_ATF_spatialGAMred[,c("YEAR", "id2", "EEZ grid", "START_LONGITUDE", "START_LATITUDE")])
S_Grid_ANCOVA = as.data.frame(unique(S_Grid[,c(1:2,6)]))
SpatOverData = merge(S_Grid_ANCOVA, spatial_Grid, by="id2")
coordinates(SpatOverData) = ~ START_LONGITUDE + START_LATITUDE

# Join overlap and spatial data to identify INPFC/IPHC areas:
setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/Data/")
INPFC_shape = readOGR(".", "GOA_Shapes")
INPFC_shape = spTransform(INPFC_shape, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"))
proj4string(SpatOverData) = proj4string(INPFC_shape)
SpatOverData$INPFC = over(SpatOverData, INPFC_shape)
SpatOverData$IPHC = over(SpatOverData, IPHC_shape)
SpatOverDf = as.data.frame(SpatOverData)
SpatOverDf$YEAR.x = as.factor(SpatOverDf$YEAR.x)
SpatOverDf = na.omit(SpatOverDf)
#####
# Spatial overlap by INPFC statistical areas:
SpatINPFC = SpatOverDf[,c(1:3,9)]
SpatINPFCdf = unique(SpatINPFC)
SpatINPFCdf = subset(SpatINPFCdf, INPFC.REP_AREA!="649")
SpatINPFCdf = subset(SpatINPFCdf, INPFC.REP_AREA!="659")

# With interaction:
INPFCmodel_A = aov(S ~ YEAR.x * INPFC.REP_AREA, data=SpatINPFCdf)
summary(INPFCmodel_A)
# Without (non-significant) interaction:
INPFCmodel_B = aov(S ~ YEAR.x + INPFC.REP_AREA, data=SpatINPFCdf)
summary(INPFCmodel_B)
TukeyHSD(INPFCmodel_B, "YEAR.x")
plot(SpatINPFCdf$S ~ SpatINPFCdf$YEAR.x)
TukeyHSD(INPFCmodel_B, "INPFC.REP_AREA")
plot(SpatINPFCdf$S ~ SpatINPFCdf$INPFC.REP_AREA)
#####
# Spatial overlap by IPHC regulatory areas:
SpatIPHC = SpatOverDf[,c(1:3,18)]
SpatIPHCdf = unique(SpatIPHC)
SpatIPHCdf = subset(SpatIPHCdf, IPHC.REG_AREA!="2B")

# With interaction:
IPHCmodel_A = aov(S ~ YEAR.x * IPHC.REG_AREA, data=SpatIPHCdf)
summary(IPHCmodel_A)
# Without (non-significant) interaction:
IPHCmodel_B = aov(S ~ YEAR.x + IPHC.REG_AREA, data=SpatIPHCdf)
summary(IPHCmodel_B)
TukeyHSD(IPHCmodel_B, "YEAR.x")
plot(SpatIPHCdf$S ~ SpatIPHCdf$YEAR.x)
TukeyHSD(IPHCmodel_B, "IPHC.REG_AREA")
plot(SpatIPHCdf$S ~ SpatIPHCdf$IPHC.REG_AREA)
#####

```

Dietary Analyses - R Script File

- 2 # This script file relies on results (e.g., spatial data frame manipulations, construction of uniform grid) from the 'SpatialAnalyses' file and includes code necessary to calculate species-specific proportions of prey by weight in each survey year and grid cell. We then used proportions of prey by weight (Chipps and Garvey 2007) to calculate Schoener's index of dietary overlap (Schoener 1968).

We analyzed standardized survey data procured from the Alaska Fisheries Science Center (National Marine Fisheries Service, National Oceanic and Atmospheric Association [NOAA]). Food habits data (1990 to 2013) were provided by the Resource Ecology and Ecosystem Modeling (REEM) Program and are publicly accessible here:

<https://access.afsc.noaa.gov/REEM/WebDietData/DietDataIntro.php>. All the data necessary to complete the following analyses can be found in the 'Data' folder. See von Szalay and Raring (2016) and Livingston et al. (2017) for data collection methods.

Rcode developed by: Cheryl Barnes
cheryl.barnes@alaska.edu

Preliminary code (e.g., exploratory data analyses, sample size calculations, summary statistics, diagnostics) have been excluded.

References:

Chipps SR, Garvey JE. Assessment of diets and feeding patterns. In: Analysis and interpretation of freshwater fisheries data. Guy CS, Brown ML, editors. Bethesda: American Fisheries Society; 2007. pp. 473–514.

Livingston PA, Aydin K, Buckley TW, Lang GM, Yang MS, Miller BS. Quantifying food web interactions in the North Pacific – a data-based approach. *Environ Biol Fishes*. 2017;100(4):443–470.

Schoener TW. The anolis lizards of Bimini: resource partitioning in a complex fauna. *Ecol*. 1968;49(4):704–726.

von Szalay PG, Raring NW. Data report: 2015 Gulf of Alaska bottom trawl survey. Seattle (WA): National Oceanic and Atmospheric Administration; 2016. Technical Memorandum: NMFS-AFSC-325. Sponsored by the US Department of Commerce.

```
setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/")
```

```
# source(SpatialAnalyses.R)
```

```
#####
```

```
### INITIAL DATA PREPARATION ###
```

```
#####
```

```
# Read in and format food habits data:
```

```
preyWT = get(load("Data/preyWTGOA_2013.Rdata"))
```

```
# Create unique haul identifier by concatenating VESSEL, CRUISE, and HAUL (allows for joining to trawl survey data below):
```

```
preyWT$Haul_Join = paste(prexWT$Vessel, preyWT$Cruise, preyWT$Haul, sep="")
```

```
# Manually assign International North Pacific Fisheries Commission statistical areas (i.e., values) based on survey strata (i.e., index).
```

Note: The second number from the right corresponds with individual statistical areas (e.g., STRATUM 120 = StatArea 620, STRATUM 251 = StatArea 650).

```
index = unique(prexWT$Strata)
```

```
values = c("610", "610", "610", "610", "630", "630", "630", "630", "630", "630", "630", "630", "640", "640", "640", "620", "620", "620",  
"610", "610", "610", "610", "610", "620", "620", "620", "620", "630", "640", "640", "640", "640", "640", "640", "640",  
"650", "650", "650", "650", "650", "650", "620", "630", "630", "630", "640", "630", "630", "630", "630", "620", "630",  
"620", "630", "650", "650", "610", "640", "620")
```

```
prexWT$StatArea = values[match(prexWT$Strata, index)]
```

```
table(prexWT$Strata, preyWT$StatArea) # Check
```

```
# Rename INPFC statistical areas:
```

```
prexWT$StatArea = as.factor(prexWT$StatArea)
```

```
levels(prexWT$StatArea) = c("Shumagin", "Chirikof", "Kodiak", "Yakutat", "Southeastern")
```

```
# Exclude data from 1981, 1984 and 1987 (survey methods were standardized in 1990):
```

```
prexWT = subset(prexWT, Yr!=1981)
```

```
prexWT = subset(prexWT, Yr!=1984)
```

```
prexWT = subset(prexWT, Yr!=1987)
```

```
# Select only predators of interest:
```

```
prexWT = subset(prexWT, Species_name==c("PACIFIC HALIBUT", "ARROWTOOTH FLOUNDER"))
```

```
# Limit data to size classes of interest (30 to 69 cm):
```

```
prexWT_red = subset(prexWT, PredL > 29)
```

```
prexWT_red = subset(prexWT_red, PredL < 70)
```

```
table(prexWT_red$Species_name)
```

```
# Set 10 cm fork length bins (30-39, 40-49, 50-59, 60-69):
```

```
prexWT_red$FLBin = cut(prexWT_red$PredL, breaks = c(0, 29, 39, 49, 59, 69, 171))
```

```
levels(prexWT_red$FLBin) = c("<30", "30-39", "40-49", "50-59", "60-69", ">=70")
```

```
table(prexWT_red$Species_name, preyWT_red$PredL, preyWT_red$FLBin) # Check
```

```
# Remove all empty stomachs:
```

```
prexWT_red$sumWT = rowSums(prexWT_red[,39:148])
```

```
prexWT_red = preyWT_red[, c(1:63, 65:152)]
```

```
prexWT_contents = subset(prexWT_red, preyWT_red$sumWT > 0)
```

```
# Compute sample sizes for proportion of prey by weight plot:
```

```
table(prexWT_contents$StatArea, preyWT_contents$FLBin, preyWT_contents$Species_name)
```

```
#####
```

```
# Join spatial and diet data:
```

```
require(dplyr)
```

```

require(stats)
require(scales)
require(ggplot2)
require(gtable)
require(grid)
options(max.print = 1000)

# Find where data overlap with the clipped grid and convert back to data frame:
coordinates(preWT contents) = c("RLONG", "RLAT")
proj4string(preWT contents) = proj4string(clipTrawl)

# Find where data overlap with the clipped grid and convert back to data frame:
tempdat = data.frame(myrows=names(over(preWT contents, clipTrawl)), mygrid=over(preWT contents, clipTrawl))
preWT contents$id2 = over(preWT contents, clipTrawl)
preWT contents = as.data.frame(preWT contents)

# Create data frame with id values and grid cell coordinates:
mycenter_INPFC = as.data.frame(gCentroid(clip_INPFC, byid=TRUE)) %>%
  mutate(id2=1:n(),
         EEZgrid=row.names())
colnames(mycenter_INPFC)[3] = "id"
xy=mycenter_INPFC[,c(1,2)]
mycenter_INPFC_sp = SpatialPointsDataFrame(coords=xy, data=mycenter_INPFC, proj4string=CRS("+proj=utm +zone=5,
+datum=WGS84 +units=km +no_defs"))
mycenter_INPFC_Pr = spTransform(mycenter_INPFC_sp, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"))
mycenter_INPFC_df = as.data.frame(mycenter_INPFC_Pr)
names(mycenter_INPFC_df) = c("x.UTM", "y.UTM", "id2", "EEZgrid", "x.lon", "y.lat")

# Join trawl and grid cell data (and clean up):
mycenter_INPFC_D = mycenter_INPFC_df %>% right_join(preWT contents)

# Relabel predator names:
levels(mycenter_INPFC_D$Species_name) = c("ATF", "PH")

# Reshape data, wide to long (selecting only columns of interest):
require(reshape2)
preWT_long = melt(mycenter_INPFC_D, id.vars = c("id2", "EEZgrid", "Yr", "StatArea", "Haul_Join", "RLONG", "RLAT", "Species_name",
"PredL", "FLBin"), measure.vars = 45:153, variable.name = "PreySpecies", value.name = "WT")
#####
### CALCULATE PREY DIVERSITY AND EVENNESS ###
#####
# Calculate prey diversity and evenness, by predator:
require(vegan)
Hsum = preWT_long %>%
  group_by(Species_name, PreySpecies) %>%
  summarize(sumWT = sum(WT))
Hcalc = spread(Hsum, key = PreySpecies, value = sumWT)
Hcalc[is.na(Hcalc)] = 0

# PACIFIC HALIBUT #
H_PH = subset(Hcalc, Species_name=="PH")
H_PH = H_PH[2:110]
Hprime_PH = diversity(H_PH)
Hprime_PH

H_PH$s = rowSums(H_PH > 0)
J_PH = Hprime_PH/log(ncol(H_PH))
J_PH

# ARROWTOOTH FLOUNDER #
H_ATF = subset(Hcalc, Species_name=="ATF")
H_ATF = H_ATF[2:110]
Hprime_ATF = diversity(H_ATF)
Hprime_ATF

```



```

H_ATF$S = rowSums(H_ATF > 0)
J_ATF = Hprime_ATF/log(ncol(H_ATF))
J_ATF
#####
### CALCULATE AND PLOT PROPORTIONS OF PREY BY WEIGHT ###
#####
# Subset diet by predator for calculating proportions of prey by weight:
PH_all = subset(preyWT_long, Species_name=="PH")
ATF_all = subset(preyWT_long, Species_name=="ATF")

PHprop_all = PH_all %>%
  group_by(PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(PHprop_all) = c("PreySpecies", "sum_PH", "prop_PH")
PHprop_all$prop_PH = round(PHprop_all$prop_PH, digits=5)

ATFprop_all = ATF_all %>%
  group_by(PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(ATFprop_all) = c("PreySpecies", "sum_ATF", "prop_ATF")
ATFprop_all$prop_ATF = round(ATFprop_all$prop_ATF, digits=5)

# Regroup prey taxa constituting less than 1% total diet by weight (for plotting only):
preyWTgrouped = preyWT_long
levels(preyWTgrouped$PreySpecies) = list(Other="Algae", Pleuronectiformes="AK.Plaice", Chondrichthyes="AK.Skate",
Chondrichthyes="Aleutian.Skate", Cnidaria="Anemones", Pleuronectiformes="Arrow.or.Kam", Atheresthes.stomias="Arrowtooth",
Pleurogrammus.monopterygius="Atka", Chionoecetes.sp="Bairdi", Argentiniformes="Bathylagidae", Crustacea="Benth.Amph",
Other="Benthic..Hydroid", Other="Benth.Urochordata", Chondrichthyes="Big.Skate", Other="Birds", Echinodermata="Brittle.Star",
Scorpaeniformes="Canary.Rock", Mallotus.villosus="Capelin", Other="Chaeteg.etc.", Mollusca="Clam", Crustacea="Copepod",
Pleuronectiformes="Dover.Sole", Crustacea="Dungeness", Scorpaeniformes="Dusky.Rock", Perciformes="Eelpout",
Osmeriformes="Eulachon", Euphausiacea="Euphausiid", Pleuronectiformes="FH.Sole", Teleostei="Fish.Eggs",
Pleuronectiformes="Gen.Rock.Sole", Scorpaeniformes="Gen.Thorny", Mollusca="Gen..Cephalopod", Clupeiformes="Gen..Clupeids",
Crustacea="Gen..Crab", Crustacea="Gen..Crustacea", Echinodermata="Gen..Echinoderm", Teleostei="Gen..Fish",
Pleuronectiformes="Gen..Flatfish", Gadiformes="Gen..Gadid", Scorpaeniformes="Gen..Hexagrammidae", Cnidaria="Gen..Hydrozoa",
Mollusca="Gen..Mollusc", Other="Gen..Particulate", Scorpaeniformes="Gen..Rockfish", Inorganic.Material="Gen..Rocks.et.al",
Scorpaeniformes="Gen..Sebastes", Osmeriformes="Gen..Smelt", Chondrichthyes="Gen..Shark.Skate", Gadiformes="Giant.Grenadier",
Other="Glopp", Pleuronectiformes="Gr..Turbot", Scorpaeniformes="Greenlings", Gadiformes="Hake", Paguroidea="Hermit.Crab",
Clupea.pallasii="Herring", Pleuronectiformes="Kamchat.fl", Crustacea="King.Crab", Scorpaeniformes="Lg.Sculpin",
Gadiformes="Macrouridae", Teleostei="Managed.Forage", Crustacea="Misc..Crab", Crustacea="Misc..Crustacean",
Pleuronectiformes="Misc..Flatfish", Other="Misc..Worm..Etc.", Myctophiformes="Myctophidae", Crustacea="Mysid",
Pleuronectiformes="N.Rock.Sole", Pandalidae="NP.Shrimp", Octopoda="Octopus", Offal="Offal", Other="Opilo",
Osmeriformes="Other.pel..Smelt", Gadus.macrocephalus="P..Cod", Pleuronectiformes="P..Halibut", Pandalidae="Pandalidae",
Crustacea="Pel.Amph", Cnidaria="Pel..Gel..Filter.Feeder", Other="Polychaete", Sebastes.alutus="POP", Other="Prickle.squish.deep",
Other="Prickle.squish.round", Other="Protozoan", Mollusca="Pteropod", Chondrichthyes="Ratfish", Pleuronectiformes="Rex.sole",
Pleuronectiformes="S.Rock.Sole", Scorpaeniformes="Sablefish", Salmoniformes="Salmon", Ammodytes.hexapterus="Sandlance",
Scorpaeniformes="Sculpin", Cnidaria="Scypho.Jellies", Cnidaria="Sea.Pens", Echinodermata="Sea.Star",
Scorpaeniformes="Sebastes", Scorpaeniformes="Sharpchin.Rock", Scorpaeniformes="Shortsp.Thorny", Mollusca="Snail",
Porifera="Sponge", Teuthida="Squid", Chondrichthyes="Unid.Bathyrja", Chionoecetes.sp="Unid.Chion",
Chondrichthyes="Unid.Rajidae", Echinodermata="Urchins.dollars.cucumbers", Gadus.chalcogrammus="W..Pollock",
Chondrichthyes="WhtBlotch.Skate", Pleuronectiformes="YF.Sole")

# Order prey items by phylogeny:
preyWTgrouped$PreySpecies = ordered(preyWTgrouped$PreySpecies, levels = c("Porifera", "Cnidaria", "Mollusca", "Teuthida",
"Octopoda", "Crustacea", "Euphausiacea", "Pandalidae", "Chionoecetes.sp", "Paguroidea", "Echinodermata", "Chondrichthyes",
"Teleostei", "Clupeiformes", "Clupea.pallasii", "Salmoniformes", "Osmeriformes", "Mallotus.villosus", "Argentiniformes",
"Myctophiformes", "Gadiformes", "Gadus.chalcogrammus", "Gadus.macrocephalus", "Perciformes", "Ammodytes.hexapterus",
"Scorpaeniformes", "Sebastes.alutus", "Pleurogrammus.monopterygius", "Pleuronectiformes", "Atheresthes.stomias", "Offal",
"Other", "Inorganic.Material"))

# Calculate proportions for each survey year-grid cell combination (using new prey taxa groupings - for plot only):
PH_grouped = subset(preyWTgrouped, Species_name=="PH")
ATF_grouped = subset(preyWTgrouped, Species_name=="ATF")

```



```

PHprop_grouped = PH_grouped %>%
  group_by(FLBin, StatArea, PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(PHprop_grouped) = c("FLBin", "StatArea", "PreySpecies", "n_PH", "prop_PH")
PHprop_grouped$prop_PH = round(PHprop_grouped$prop_PH, digits=5)

ATFprop_grouped = ATF_grouped %>%
  group_by(FLBin, StatArea, PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(ATFprop_grouped) = c("FLBin", "StatArea", "PreySpecies", "n_ATF", "prop_ATF")
ATFprop_grouped$prop_ATF = round(ATFprop_grouped$prop_ATF, digits=5)

# Merge PH and ATF proportional data:
WTprop_grouped = merge(PHprop_grouped, ATFprop_grouped, by = c("FLBin", "StatArea", "PreySpecies"), all = TRUE)

# Order and relabel predators:
preyWT_grouped$Species_name = ordered(preyWT_grouped$Species_name, levels = c("PH", "ATF"))
levels(preyWT_grouped$Species_name) = c("Pacific Halibut", "Arrowtooth Flounder")

# Plot proportions of prey by predator, size class, and stat area:
quartz()
preyWT_plot = ggplot(preyWT_grouped, aes(x = FLBin, y = WT, fill = PreySpecies, order = -as.numeric(PreySpecies))) +
  geom_bar(position = "fill", stat = "identity") +
  facet_grid(Species_name ~ StatArea) +
  theme_bw() +
  ggtitle("") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(colour="black", size=1, line="solid")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(colour="black", size=1, line="solid")) +
  theme(panel.spacing.x = unit(0.5, "lines")) +
  theme(panel.spacing.y = unit(0.8, "lines")) +
  theme(legend.title = element_text(size=11, face="bold", vjust=1.5)) +
  theme(legend.text = element_text(family="Arial", size=9.5, legend.text.align = 0)) +
  theme(axis.text = element_text(family="Arial", size=12)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(vjust=-0.13, size=14)) +
  theme(axis.title.y = element_text(vjust=1.1, size=14)) +
  theme(strip.background = element_rect(colour="white", fill="white")) +
  theme(strip.text = element_text(family="Arial", size=14)) +
  scale_fill_manual(values = c("Porifera"="lightpink3", "Cnidaria"="lavenderblush2", "Mollusca"="brown4", "Teuthida"="firebrick3",
    "Octopoda"="red", "Crustacea"="chocolate", "Euphausiacea"="chocolate1", "Pandalidae"="orange", "Chionoecetes.sp"="gold",
    "Paguroidea"="khaki1", "Echinodermata"="lemonchiffon", "Chondrichthyes"="darkolivegreen1", "Teleostei"="chartreuse2",
    "Clupeiformes"="forestgreen", "Clupea.pallasii"="darkgreen", "Salmoniformes"="lightcyan1", "Osmeriformes"="cadetblue1",
    "Mallotus.villosus"="turquoise3", "Argentiniformes"="deepskyblue3", "Myctophiformes"="deepskyblue4", "Gadiformes"="blue",
    "Gadus.chalcogrammus"="mediumblue", "Gadus.macrocephalus"="blue4", "Perciformes"="mediumpurple1",
    "Ammodytes.hexapterus"="darkviolet", "Scorpaeniformes"="purple4", "Sebastes.alutus"="black",
    "Pleuronectiformes.monopterygius"="hotpink1", "Pleuronectiformes"="mediumvioletred", "Atheresthes.stomias"="maroon4",
    "Offal"="gray91", "Other"="azure4", "Inorganic.Material"="black"), name="", labels = c("Porifera", "Cnidaria", "Mollusca", "Teuthida",
    "Octopoda", "Crustacea", "Euphausiacea", "Pandalidae", expression(paste(italic("Chionoecetes spp."))), "Paguroidea",
    "Echinodermata", "Chondrichthyes", "Teleostei", "Clupeiformes", expression(paste(italic("Clupea pallasii"))), "Salmoniformes",
    "Osmeriformes", expression(paste(italic("Mallotus villosus"))), "Argentiniformes", "Myctophiformes", "Gadiformes",
    expression(paste(italic("Gadus chalcogrammus"))), expression(paste(italic("Gadus macrocephalus"))), "Perciformes",
    expression(paste(italic("Ammodytes hexapterus"))), "Scorpaeniformes", expression(paste(italic("Sebastes alutus"))),
    expression(paste(italic("Pleuronectes monopterygius"))), "Pleuronectiformes", expression(paste(italic("Atheresthes stomias"))),
    "Offal", "Other", "Inorganic Material"), drop=FALSE) +
  labs(x="Fork Length (cm)", y="Proportion by Weight") +
  scale_y_continuous(breaks = c(0, 0.5, 1.0), expand = c(0, 0.01)) +
  scale_x_discrete(expand = c(0.03, 0.03)) +
  guides(fill=guide_legend(nrow=5)) +
  theme(legend.direction = "horizontal", legend.position = "bottom", legend.box.margin = margin(0,0,0,0), legend.background =
    element_rect(fill="transparent"))

```

preyWTplot

```
ggsave(filename="Plots/preyWTplot.png", plot=preyWTplot, dpi=500, width=11.5, height=7, units="in")
#####
### CALCULATE AND PLOT MEAN DIETARY OVERLAP ###
#####
# Eliminate survey year-grid cell combinations with less than three fish sampled for either species:
Nfish = mycenter_INPFC_D %>%
  group_by(Yr, id2, Species_name) %>%
  summarise(NoFish=length(unique(PPID)))
Nfish_wide = spread(Nfish, key = Species_name, value = NoFish)
Nfish_wide[is.na(Nfish_wide)] = 0
Nfish_sub = subset(Nfish_wide, ATF >= 3)
Nfish_sub = subset(Nfish_sub, PH >= 3)
preyWT_longGrid = Nfish_sub %>% left_join(preyWT_long)
#####
# Summarize diet data (ungrouped prey taxa for overlap calculations):

# PACIFIC HALIBUT #
PH = subset(preyWT_longGrid, Species_name == "PH")
PHpropGrid = PH %>%
  group_by(Yr, id2, EEZgrid, PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(PHpropGrid) = c("Yr", "id2", "EEZgrid", "PreySpecies", "n_PH", "prop_PH")
PHpropGrid$prop_PH = round(PHpropGrid$prop_PH, digits=5)

# ARROWTOOTH FLOUNDER #
ATF = subset(preyWT_longGrid, Species_name == "ATF")
ATFpropGrid = ATF %>%
  group_by(Yr, id2, EEZgrid, PreySpecies) %>%
  summarize(n = sum(WT)) %>%
  mutate(proportion = n / sum(n))
colnames(ATFpropGrid) = c("Yr", "id2", "EEZgrid", "PreySpecies", "n_ATF", "prop_ATF")
ATFpropGrid$prop_ATF = round(ATFpropGrid$prop_ATF, digits=5)

# Merge PH and ATF proportional data:
WTprop = merge(PHpropGrid, ATFpropGrid, by = c("id2", "EEZgrid", "Yr", "PreySpecies"), all = TRUE)
WTprop[is.na(WTprop)] = 0 # Set NA values to 0 (absent)

# Remove prey taxa not consumed by either predator:
WTprop$sumWT = rowSums(WTprop[,c("prop_PH", "prop_ATF")])
WTprop = subset(WTprop, sumWT > 0)
#####
# Calculate dietary overlap (Schoener 1968):
WTprop$overlap = abs(WTprop$prop_PH - WTprop$prop_ATF)

D_Grid = WTprop %>%
  group_by(Yr, id2, EEZgrid) %>%
  summarize(n = sum(overlap)) %>%
  mutate(D = 1-0.5*(n))
D_Grid$D = round(D_Grid$D, digits=3)
D_Grid = D_Grid[, c(1:3, 5)]

# Write CSV for analyses of resource partitioning:
write.csv(D_Grid, "Data/PH_ATF_D.csv")
# D_overlap = read.csv("Data/PH_ATF_D.csv")

# Summarize by grid cell (all years combined):
D_Grid_mean = D_Grid %>%
  group_by(id2) %>%
  mutate(meanD = mean(D))
D_Grid_mean = D_Grid_mean[, c(2:3, 5)]
D_overlap = unique(D_Grid_mean)
```

```

# Join summary information and spatial data:
goa.df = fortify(clip2_INPFC, region='id')
colnames(goa.df)[colnames(goa.df) == "id"] = "EEZ grid"
plot_PH_ATF_D = goa.df %>% right_join(D_overlap)
#####
# Plot mean dietary overlap (all years combined):

### Dietary Overlap ###
textD = data.frame(text = c("Dietary Overlap"))

PH_ATF_Dplot = ggplot() +
  geom_polygon(data=plot_PH_ATF_D, aes(x=long, y=lat, group=group, fill=meanD), col="black", lwd=0.25) +
  geom_polygon(data=Area2C_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="cadetblue1",
linetype="solid", lwd=1) +
  geom_polygon(data=Area3A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="deepskyblue2",
linetype="solid", lwd=1) +
  geom_polygon(data=Area3B_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="blue",
linetype="solid", lwd=1) +
  geom_polygon(data=Area4A_df, alpha=0.3, aes(x=X, y=Y, group=factor(PID)), fill=NA, show.legend=FALSE, col="midnightblue",
linetype="solid", lwd=1) +
  scale_fill_distiller(palette="Spectral", name="", limits=c(0,1), breaks=c(0.00,0.25,0.50,0.75,1.00)) +
  geom_polygon(data=INPFC_plot, aes(x=X, y=Y, group=factor(PID)), fill=NA, col="black", linetype="solid", lwd=0.25) +
  geom_polygon(data=world2, aes(x=X, y=Y, group=factor(PID)), fill="lightgrey", col="black", lwd=0.25) +
  geom_polygon(data=Canada, aes(x=X, y=Y, group=factor(PID)), fill="gray91", col="black", lwd=0.25) +
  geom_text(data=textD, aes(label=text, x=-132.48, y=61.65, size=12), show.legend=FALSE) +
  geom_text(data=INPFC_cent, aes(group=StatArea, label=StatArea, x=START_LONGITUDE, y=START_LATITUDE, size=12),
show.legend = FALSE) +
  geom_text(data=IPHC_2C, aes(label=text, x=-134.84, y=56.06, size=12), col="cadetblue1", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_3A, aes(label=text, x=-145.40, y=59.9, size=12), col="deepskyblue3", fontface="bold", show.legend = FALSE)
+
  geom_text(data=IPHC_3B, aes(label=text, x=-157.39, y=55.65, size=12), col="blue", fontface="bold", show.legend = FALSE) +
  geom_text(data=IPHC_4A, aes(label=text, x=-167.75, y=53.05, size=12), col="midnightblue", fontface="bold", show.legend = FALSE)
+
  theme_bw() +
  ggtitle("Dietary Overlap between Pacific Halibut and Arrowtooth Flounder") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(panel.background = element_rect(fill="white", colour="black", size=1, line="solid")) +
  theme(legend.position = c(0.957, 0.830)) +
  theme(plot.title = element_text(hjust = 0.5, size=14)) +
  theme(legend.title = element_blank()) +
  theme(legend.text = element_text(family="Arial", size=12), legend.text.align = 1) +
  theme(legend.background = element_rect(fill="transparent")) +
  theme(legend.key = element_blank()) +
  theme(legend.key.width = unit(6.0, "mm")) +
  theme(legend.key.height = unit(8.0, "mm")) +
  theme(legend.spacing.x = unit(2.0, "mm")) +
  theme(axis.text.y = element_text(family="Arial", size=12)) +
  theme(axis.text.x = element_text(family="Arial", size=12)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  theme(axis.title.x = element_text(hjust=0.53, size=12)) +
  theme(axis.title.y = element_text(hjust=0.46, size=12)) +
  labs(x="Longitude", y="Latitude") +
  scale_x_continuous(limits = c(lonmin, lonmax), expand = c(0,0)) +
  scale_y_continuous(limits = c(latmin, latmax), expand = c(0,0), labels = function(x) round(as.numeric(x), digits=0)) +
  theme(legend.background = element_rect(fill="transparent"))

PH_ATF_Dplot
ggsave(filename="Plots/D_PH_ATF.png", plot=PH_ATF_Dplot, dpi=500, width=12, height=8, units="in")
#####
### TEST FOR SPATIOTEMPORAL CHANGES IN DIETARY OVERLAP ###
#####
# Run ANCOVAs to test for relationships among dietary overlap, survey year, and INPFC statistical area or IPHC regulatory area:

# Prepare overlap data:

```

```

dietary_Grid = unique(PH_ATF_spatialGAMred[c("YEAR", "id2", "EEZ_grid", "START_LONGITUDE", "START_LATITUDE")])
colnames(D_Grid) = c("YEAR", "id2", "EEZ_grid", "D")
D_Grid$YEAR = as.factor(D_Grid$YEAR)
DietOverData = merge(D_Grid, dietary_Grid, all.x=TRUE)
DietOverData = na.omit(DietOverData)
coordinates(DietOverData) = ~ START_LONGITUDE + START_LATITUDE

# Join overlap and spatial data to identify INPFC/IPHC areas:
proj4string(DietOverData) = proj4string(INPFC_shape)
DietOverData$INPFC = over(DietOverData, INPFC_shape)
DietOverData$IPHC = over(DietOverData, IPHC_shape)
DietOverDf = as.data.frame(DietOverData)
DietOverDf$YEAR = as.factor(DietOverDf$YEAR)
DietOverDf = na.omit(DietOverDf)
#####
# INPFC statistical areas:
DietINPFC = DietOverDf[,c(1:2,4,8)]
DietINPFCdf = unique(DietINPFC)
DietINPFCdf = subset(DietINPFCdf, INPFC.REP_AREA!="649")

# With interaction:
INPFCmodel_A = aov(D ~ YEAR * INPFC.REP_AREA, data=DietINPFCdf)
summary(INPFCmodel_A)
# Without (non-significant) interaction:
INPFCmodel_B = aov(D ~ YEAR + INPFC.REP_AREA, data=DietINPFCdf)
summary(INPFCmodel_B)
TukeyHSD(INPFCmodel_B, "YEAR")
plot(DietINPFCdf$D ~ DietINPFCdf$YEAR)
#####
# IPHC regulatory areas:
DietIPHC = DietOverDf[,c(1:2,4,17)]
DietIPHCdf = unique(DietIPHC)
DietIPHCdf = subset(DietIPHCdf, IPHC.REG_AREA!="2B")

# With interaction:
IPHCmodel_A = aov(D ~ YEAR * IPHC.REG_AREA, data=DietIPHCdf)
summary(IPHCmodel_A)
# Without (non-significant) interaction:
IPHCmodel_B = aov(D ~ YEAR + IPHC.REG_AREA, data=DietIPHCdf)
summary(IPHCmodel_B)
TukeyHSD(IPHCmodel_B, "YEAR")
plot(DietIPHCdf$D ~ DietIPHCdf$Yr)
#####

```

Resource Partitioning - R Script File

- 3 # This script file relies on results (e.g., spatial data frame manipulations, construction of uniform grid, overlap measure calculations) from the 'SpatialAnalyses' and 'DietaryAnalyses' files. Here, we combine spatial overlap and dietary overlap to quantify the correlation between the two measures and thus the degree of resource partitioning between Pacific Halibut and Arrowtooth Flounder in the Gulf of Alaska. Descriptions of resource partitioning can be found in Schoener (1974) and Ross (1986).

```

# Rcode developed by: Cheryl Barnes
# cheryl.barnes@alaska.edu
# Preliminary code (e.g., exploratory data analyses, sample size calculations, summary statistics, diagnostics) were excluded.

```

```

# References:
# Ross ST. Resource partitioning in fish assemblage: review of field studies. Copeia. 1986;1986(2):352–388.
# Schoener TW. Resource partitioning in ecological communities. Science. 1974;185(4145):27–39.

```

```

setwd("~/Documents/UAF/Dissertation/GitHub/ResourcePartitioning/")
# source(SpatialAnalyses.R)
# source(DietaryAnalyses.R)
#####
### DATA PREPARATION AND ANALYSES ###
#####

```

```

# Read in and join overlap data from 'SpatialAnalyses' and 'DietaryAnalyses' script files:
spatial = S_Grid
colnames(spatial) = c("X", "id2", "YEAR", "S", "INPFC.REP_AREA")
dietary = D_Grid

require(dplyr)
overlap = S_Grid %>% right_join(D_Grid)
overlap = overlap[complete.cases(overlap), ]

# Join overlap and additional spatial data to assign INPFC statistical areas and IPHC regulatory areas:
require(rgdal)
require(sp)
coordinates(overlap) = ~ START_LONGITUDE + START_LATITUDE

proj4string(overlap) = proj4string(INPFC_shape)
overlap$INPFC = over(overlap, INPFC_shape)

overlap$IPHC = over(overlap, IPHC_shape)
overlap = as.data.frame(overlap)
overlap_red = unique(overlap[,c(1,2,6,7,9,18)])
#####
# Test for basin-wide correlation between spatial overlap and dietary overlap (i.e., degree of resource partitioning - proxy for
competition):
overlap_corr = unique(overlap_red[,c(1:4)])
cor.test(overlap_corr$D, overlap_corr$S, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
#####
# Test for region-specific correlations:
# INPFC #
overlap_INPFC = unique(overlap_red[,c(1:5)])
levels(overlap_INPFC$INPFC.REP_AREA) = c("Shumagin", "Chirikof", "Kodiak", "Yakutat", "NA", "Southeastern", "NA")
Shum = subset(overlap_INPFC, INPFC.REP_AREA == "Shumagin")
Chir = subset(overlap_INPFC, INPFC.REP_AREA == "Chirikof")
Kod = subset(overlap_INPFC, INPFC.REP_AREA == "Kodiak")
Yak = subset(overlap_INPFC, INPFC.REP_AREA == "Yakutat")
SE = subset(overlap_INPFC, INPFC.REP_AREA == "Southeastern")

cor.test(Shum$S, Shum$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(Chir$S, Chir$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(Kod$S, Kod$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(Yak$S, Yak$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(SE$S, SE$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)

# IPHC #
overlap_IPHC = unique(overlap_red[,c(1:4,6)])
IPHC4A = subset(overlap_IPHC, IPHC.REG_AREA == "4A")
IPHC3B = subset(overlap_IPHC, IPHC.REG_AREA == "3B")
IPHC3A = subset(overlap_IPHC, IPHC.REG_AREA == "3A")
IPHC2C = subset(overlap_IPHC, IPHC.REG_AREA == "2C")

cor.test(IPHC4A$S, IPHC4A$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(IPHC3B$S, IPHC3B$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(IPHC3A$S, IPHC3A$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
cor.test(IPHC2C$S, IPHC2C$D, alternative="two.sided", method="pearson", exact=TRUE, conf.level = 0.90)
#####
# Plot relationship between spatial overlap and dietary overlap:
overlap_long = melt(overlap, id.vars = c("YEAR", "id2", "EEZ grid", "START_LONGITUDE", "START_LATITUDE"), measure.vars = 6:7,
variable.name = "OverlapIndex", value.name = "measure")
library(ggplot2)
require(plyr)
overlap_summ = ddply(overlap_long, c("OverlapIndex", "YEAR"), summarise,
  N = length(measure),
  mean = mean(measure),
  sd = sd(measure),
  se = sd / sqrt(N))
overlap_summ

```

```

overlap_summ$YEAR = as.numeric(as.character(overlap_summ$YEAR))
pd = position_dodge(0.1)
overlap_summ$CI = 1.96 * overlap_summ$sd

NicheOverlap = ggplot(data=overlap_summ, aes(x=YEAR, y=mean, col=OverlapIndex)) +
  geom_line(aes(linetype=OverlapIndex)) +
  geom_point(aes(shape=OverlapIndex), size=2.5) +
  geom_errorbar(aes(ymin=mean-CI, ymax=mean+CI), width=0.5) +
  scale_color_manual(values=c("blue", "red")) +
  theme_bw() +
  ggtitle("") +
  theme(legend.position = c(0.05, 0.93)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(legend.title = element_blank()) +
  theme(legend.title.align = 0) +
  theme(legend.text = element_text(family="Arial", size=12), legend.text.align = 0) +
  theme(legend.background = element_rect(fill="transparent")) +
  theme(legend.key = element_blank()) +
  theme(legend.key.width = unit(1.9, "mm")) +
  theme(legend.key.height = unit(6.0, "mm")) +
  theme(legend.spacing.x = unit(2.0, "mm")) +
  theme(axis.text.y = element_text(family="Arial", size=12)) +
  theme(axis.text.x = element_text(family="Arial", size=11)) +
  theme(panel.grid.major.y = element_blank(), panel.grid.minor.y = element_blank()) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  labs(x="Survey Year", y="Overlap Measure") +
  theme(axis.title.x = element_text(vjust=-0.13, size=12)) +
  theme(axis.title.y = element_text(vjust=1.1, size=12)) +
  theme(strip.text.x = element_text(family="Arial", size=12)) +
  theme(strip.text = element_text(hjust=0.5)) +
  theme(strip.background = element_rect(fill="white", colour="white", size=1, line="solid")) +
  scale_x_continuous(breaks = c(1990, 1993, 1996, 1999, 2001, 2003, 2005, 2007, 2009, 2011, 2013)) +
  scale_y_continuous(limits=c(-0.5,1.0))

NicheOverlap
ggsave(filename="Plots/NicheOverlap.png", plot=NicheOverlap, dpi=500, width=6, height=5, units="in")

plot(overlap$D ~ overlap$S)

```



This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited