

Non-analogue states in Gulf of Alaska salmon production

This notebook documents all the code needed to replicate results reported in Figures 2-6 of the submitted version of the paper.

First, load the required packages

```
library(gtools)
library(ncdf4)
library(MuMIn)
library(zoo)
library(scales)
library(nlme)
library(gplots)
library(dplyr)
library(maps)
library(mapdata)
library(chron)
library(fields)
library(pracma)
library(FactoMineR)
library(lmtest)
library(MuMIn)
library(broom)
library(reshape2)
```

Data accession details.

ERRST v4 data were obtained from <https://www.esrl.noaa.gov/psd/data/gridded/data.noaa.ersst.v4.html> on 2 February 2017.

NOAA-CIRES 20th Century Reanalysis v2c data were obtained from https://www.esrl.noaa.gov/psd/data/gridded/data.20thC_ReanV2c.html on 2 February 2017.

The PDO index was obtained from <http://research.jisao.washington.edu/pdo/PDO.latest.txt> on 3 March 2018.

The NPGO index was obtained from <http://www.o3d.org/npgo/npgo.php> on 14 February 2018.

Figs. 2 and S1

Nonstationary SLP variance.

```
# re-load SLP
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/climate data/prmsl.mon.mean.7.28.15.nc")

# get dates (hours since 1/1/1800)
raw <- ncvar_get(nc, "time")
h <- raw/24
```

```

d <- dates(h, origin = c(1,1,1800))

# Pick start and end dates (Jan 1950-Dec 2012):
d <- d[949:1704]

# Extract North Pacific SLP, 20-66 deg. N, 132-250 deg. E:
x <- ncvar_get(nc, "lon", start=67, count=60)
y <- ncvar_get(nc, "lat", start=13, count=24)

SLP <- ncvar_get(nc, "prmsl", start=c(67,13,949), count=c(60,24,length(d)), verbose = F)

# manipulate as needed
SLP <- aperm(SLP, 3:1) # First, reverse order of dimensions ("transpose" array)
SLP <- SLP[,24:1,] # Reverse order of latitudes to be increasing for convenience (in later plotting)
y <- rev(y)

SLP <- matrix(SLP, nrow=dim(SLP)[1], ncol=prod(dim(SLP)[2:3])) # Change to matrix

# Keep track of corresponding latitudes and longitudes of each column:
lat <- rep(y, length(x)) # Vector of latitudes
lon <- rep(x, each = length(y)) # Vector of longitudes
dimnames(SLP) <- list(as.character(d), paste("N", lat, "E", lon, sep=""))

X1 <- as.data.frame(SLP) # using data over land, too!

# remove seasonal signal
m <- months(d)

f <- function(x) tapply(x, m, mean)
mu <- apply(X1, 2, f) # Compute monthly means for each time series (location)
mu <- mu[rep(1:12, round(length(d)/12)),]
X1.anom <- X1 - mu # Compute matrix of anomalies!

# smooth with 11-yr rolling mean
sm.SLP <- rollmean(X1.anom, 11, fill=NA)
dimnames(sm.SLP) <- dimnames(X1.anom)

# Now calculate spatial patterns of change in SLP SD.
sd.diff <- NA # object to save the output

for(j in 1:ncol(sm.SLP)){
  sd.diff[j] <- sd(sm.SLP[469:756, j], na.rm=T) - sd(sm.SLP[1:468, j], na.rm=T)
}

# now the time series
# reload the SLP data
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/climate data/prmsl.mon.mean.7.28.15.nc") #new version!

raw <- ncvar_get(nc, "time")
h <- raw/24
d <- dates(h, origin = c(1,1,1800))

# Pick start and end dates (Jan 1950-Dec 2012):

```

```

d <- d[949:1704]

# just the box in spatial change in SD plot! 48-56 deg. N, 192-206 deg. E:
sd.x <- nvar_get(nc, "lon", start=97, count=8)
sd.y <- nvar_get(nc, "lat", start=18, count=5)

SLP1 <- nvar_get(nc, "prmsl", start=c(97,18,949), count=c(8,5,length(d)))

SLP1 <- aperm(SLP1, 3:1) # First, reverse order of dimensions ("transpose" array)
SLP1 <- SLP1[,5:1,] # Reverse order of latitudes to be increasing for convenience (in later plotting)
sd.y <- rev(sd.y) # Also reverse corresponding vector of latitudes

SLP1 <- matrix(SLP1, nrow=dim(SLP1)[1], ncol=prod(dim(SLP1)[2:3])) # Change to matrix
# Keep track of corresponding latitudes and longitudes of each column:
lat <- rep(sd.y, length(sd.x)) # Vector of latitudes
lon <- rep(sd.x, each = length(sd.y)) # Vector of longitudes

dimnames(SLP1) <- list(as.character(d), paste("N", lat, "E", lon, sep=""))

m <- months(d)
f <- function(x) tapply(x, m, mean) # function to compute monthly means for a single time series

p.mu1 <- apply(SLP1, 2, f) # Compute monthly means for each time series (location)
p.mu1 <- p.mu1[rep(1:12, round(length(d)/12)),]

# Compute matrix of anomalies!
SLP1.anom <- (SLP1 - p.mu1)

# and smoothing with 11-mo rolling mean
SLP1.anom <- rollmean(SLP1.anom, 11, fill=NA)
mean.anom <- rowMeans(SLP1.anom)

# make a data frame
slp <- data.frame(slp.11 = mean.anom, year=rep(1950:2012, each=12), month=rep(1:12, length.out=63*12))
slp$dec.yr <- slp$year + (slp$month-0.5)/12

# now 11-year (133-mo) rolling sd!
slp$sm.sd <- NA

for(i in 67:690){

  win <- (i-66):(i+66)
  slp$sm.sd[i] <- sd(slp$slp.11[win])

}

```

And plot the time series as Fig. 2.

```

# set up color scheme
my.col <- tim.colors(64)
grays <- c("gray98", "gray97", "gray96", "gray95", "gray94", "gray93", "gray92", "gray91", "gray90", "gray89", "gray88", "gray87", "gray86", "gray85", "gray84", "gray83", "gray82", "gray81", "gray80", "gray79", "gray78", "gray77", "gray76", "gray75", "gray74", "gray73", "gray72", "gray71", "gray70", "gray69", "gray68", "gray67", "gray66", "gray65", "gray64", "gray63", "gray62", "gray61", "gray60", "gray59", "gray58", "gray57", "gray56", "gray55", "gray54", "gray53", "gray52", "gray51", "gray50", "gray49", "gray48", "gray47", "gray46", "gray45", "gray44", "gray43", "gray42", "gray41", "gray40", "gray39", "gray38", "gray37", "gray36", "gray35", "gray34", "gray33", "gray32", "gray31", "gray30", "gray29", "gray28", "gray27", "gray26", "gray25", "gray24", "gray23", "gray22", "gray21", "gray20", "gray19", "gray18", "gray17", "gray16", "gray15", "gray14", "gray13", "gray12", "gray11", "gray10", "gray9", "gray8", "gray7", "gray6", "gray5", "gray4", "gray3", "gray2", "gray1", "gray0")
my.col[22:43] <- c(grays[11:1], grays)

# setup the layout

```

```

mt.cex <- 1.1
l.mar <- 1
l.cex <- 0.8
l.l <- 2
tc.l <- -0.2

png("Fig 2.png", 0.6*11.4/2.54, 0.5*9.5/2.54, units="in", res=300)

par(mar=c(1.5,3,1,1.5), tcl=tc.l, mgp=c(1.5,0.3,0), las=1, mfrow=c(1,1), cex.axis=0.8, cex.lab=0.8, oma=0)

plot(slp$dec.yr, slp$sm.sd/100, type="l", xlab="", ylab="hPa", col="red") # converted to hPa
abline(v=(1988+11.5/12), lty=2)
mtext("SLPa standard deviation", adj=0.5, cex=0.8)

dev.off()

## pdf
## 2

# and plot the spatial pattern as Fig. S1

png("Fig S1.png", 8, 6, units="in", res=300)
linex <- c(191, 191, 207, 207, 191)
liney <- c(47, 57, 57, 47, 47)
z <- sd.diff/100 # converting to hPa
lim <- range(sd.diff/100)
z <- t(matrix(z, length(y))) # Convert vector to matrix and transpose for plotting
image.plot(x,y,z, col=my.col, zlim=c(lim[1], -lim[1]), xlab = "", ylab = "", yaxt="n", xaxt="n", legend=FALSE,
  axis.args = list(tcl=tc.l, mgp=c(3,0.25,0)))

contour(x,y,z, add=T, drawlabels=T, col="white", vfont=c("sans serif", "bold"), labcex=1)
map('world2Hires', fill=F, xlim=c(130,250), ylim=c(20,70), add=T, lwd=1)
lines(linex, liney, lwd=2, col="magenta")
dev.off()

## pdf
## 2

```

Fig. 3

Non-analogue environmental dynamics in the GOA.

```

# load the time series for the six environmental variables
salm.cov <- read.csv("GOA_salmon_covariates.csv")

# And re-calculate SD for the AL area.
# re-load slp
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/climate data/prmsl.mon.mean.7.28.15.nc") #new version!

raw <- ncvar_get(nc, "time")
h <- raw/24
d <- dates(h, origin = c(1,1,1800))

# Pick start and end dates (Jan 1949-Dec 2012):

```

```

d <- d[937:1704]  # for later use!

# Extract "AL area" 48-56N, 192-206W
p.x3 <- ncvar_get(nc, "lon", start=97, count=8)
p.y3 <- ncvar_get(nc, "lat", start=18, count=5)

# AL area
SLP3 <- ncvar_get(nc, "prmsl", start=c(97,18,937), count=c(8,5,length(d)), verbose = F)

# manipulate as needed
SLP3 <- aperm(SLP3, 3:1) # First, reverse order of dimensions ("transpose" array)
SLP3 <- SLP3[,5:1,] # Reverse order of latitudes to be increasing for convenience (in later plotting)
p.y3 <- rev(p.y3)
SLP3 <- matrix(SLP3, nrow=dim(SLP3)[1], ncol=prod(dim(SLP3)[2:3])) # Change to matrix

# Keep track of corresponding latitudes and longitudes of each column:
p.lat3 <- rep(p.y3, length(p.x3)) # Vector of latitudes
p.lon3 <- rep(p.x3, each = length(p.y3)) # Vector of longitudes
dimnames(SLP3) <- list(as.character(d), paste("N", p.lat3, "E", p.lon3, sep=""))

# now change to monthly anomalies, i.e., remove seasonal signal
m <- months(d)
yr <- years(d)
f <- function(x) tapply(x, m, mean) # function to compute monthly means for a single time series

p.mu3 <- apply(SLP3, 2, f) # Compute monthly means for each time series (location)
p.mu3 <- p.mu3[rep(1:12, round(length(d)/12)),]
p.X3.anom <- SLP3 - p.mu3 # Compute matrix of anomalies!

# now smooth SLP w/ 11-mo rolling mean following Johnstone and Mantua 2014
sm.SLP3.anom <- rollmean(p.X3.anom, 11, fill=NA)
rownames(sm.SLP3.anom) <- rownames(p.X3.anom)

win <- c("Nov", "Dec", "Jan", "Feb", "Mar")
win.yr <- as.numeric(as.character(yr))
win.yr[m %in% c("Nov", "Dec")] <- win.yr[m %in% c("Nov", "Dec")] + 1

win.yr <- win.yr[m %in% win]
win.slp <- rowMeans(sm.SLP3.anom[m %in% win,], na.rm=T) # so these are raw SLP values, not scaled!
win.slp <- tapply(win.slp, win.yr, mean)

salm.cov$AL.SLP.win.mu <- win.slp[2:64]

```

Now the correlations between winter AL SLP and the various time series! First, for pre-1988/89.

```

# this code runs the Modified Chelton method for adjusting df in the presence of autocorrelation
X <- zoo(salm.cov[salm.cov$year < 1989, 8]) # these are the slp anomalies, which we're relating to every
YY <- 2:7 # these are the column names for the other params!

out <- matrix(nrow=length(YY), ncol=4) # matrix to catch the output
dimnames(out) <- list(colnames(salm.cov)[2:7], c("r early", "p early", "r late", "p late"))

for(i in 1:length(YY)){
  # i <- 1

```

```

Y <- zoo(salm.cov[salm.cov$year < 1989, YY[i]])

# length of TS
missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
X[missY] <- NA # drop the corresponding observations in X

N <- sum(!is.na(X)) # sample size of pairwise observations!

# number of lags to use
J <- round(N/5)

# means of each TS
Xbar <- mean(X, na.rm=T)
Ybar <- mean(Y, na.rm=T)

# calculate denominators for eq. 6
ssX <- ssY <- NA

for (ii in 1:length(Y)){
  ssX[ii] <- (X[ii]-Xbar)^2
  ssY[ii] <- (Y[ii]-Ybar)^2
}
denomX <- sum(ssX, na.rm=T)
denomY <- sum(ssY, na.rm=T)

rXX <- rYY <- NA # vectors of autocorrelation parameters

for (j in 1:J){
  ssX <- ssY <- NA
  Xl <- lag(X, j)
  Yl <- lag(Y, j)
  for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
    # k <- 1
    ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
    ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
  }

  numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  numerY <- sum(ssY, na.rm=T)
  adj <- N/(N-j) # adjustor from eq. 7
  rXX[j] <- adj*numerX/denomX
  rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)
cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!

```

```

r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

out[i,1] <- r.ac
out[i,2] <- p
}

# now for 1989-2012
X <- zoo(salm.cov[salm.cov$year > 1988, 8]) # these are the slp anomalies, which we're relating to ever
YY <- 2:7 # these are the column names for the other params!

for(i in 1:length(YY)){
  # i <- 1

  Y <- zoo(salm.cov[salm.cov$year > 1988, YY[i]])

  # length of TS
  missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
  X[missY] <- NA # drop the corresponding observations in X

  N <- sum(!is.na(X)) # sample size of pairwise observations!

  # number of lags to use
  J <- round(N/5)

  # means of each TS
  Xbar <- mean(X, na.rm=T)
  Ybar <- mean(Y, na.rm=T)

  # calculate denominators for eq. 6
  ssX <- ssY <- NA

  for (ii in 1:length(Y)){
    ssX[ii] <- (X[ii]-Xbar)^2
    ssY[ii] <- (Y[ii]-Ybar)^2
  }
  denomX <- sum(ssX, na.rm=T)
  denomY <- sum(ssY, na.rm=T)

  rXX <- rYY <- NA # vectors of autocorrelation parameters

  for (j in 1:J){
    ssX <- ssY <- NA
    Xl <- lag(X, j)
    Yl <- lag(Y, j)
    for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
      # k <- 1
      ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
      ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
    }

    numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  }
}

```

```

numerY <- sum(ssY, na.rm=T)
adj <- N/(N-j) # adjustor from eq. 7
rXX[j] <- adj*numerX/denomX
rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)
cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!
r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

out[i,3] <- r.ac
out[i,4] <- p
}

# sort by the early period correlation coefficients
# and save as an object for later printing
out <- out[order(-out[,1]),]
plot.cors <- out[,c(1,3)]
# and print the correlation coefficients and p values for the relationships with AL SLP values by era.
out

```

```

##           r early      p early      r late      p late
## sal20m      0.8545976 4.889366e-05  0.64447477 0.002101769
## downwell    0.4364873 9.484348e-02  0.03273523 0.882273827
## SLPgrad     -0.6515493 1.177871e-05  0.03021481 0.883621201
## FWdisch     -0.7515318 8.516883e-06 -0.38012871 0.075053991
## wind.stress -0.7742871 7.794308e-04 -0.04971404 0.825054348
## advection   -0.8382502 1.079457e-04 -0.37935785 0.077525032

```

Now, calculate separate PCA for pre/post 88/89.

```

include <- colnames(salm.cov)[2:7] # params to include! (excluding downwelling)

# first the early era

# scale the data
clim1 <- scale(salm.cov[salm.cov$year <= 1988,colnames(salm.cov) %in% include])
# and run svd
pc.clim1 <- svd(cov(clim1, use="pair"))

# and late...

#scale
clim2 <- scale(salm.cov[salm.cov$year > 1988,colnames(salm.cov) %in% include])
# and run
pc.clim2 <- svd(cov(clim2, use="pair"))

```



```

# get pc1 for each era!
salm.cov$pc1.clim1 <- salm.cov$pc1.clim2 <- NA

pc1.early <- clim1 %*% pc.clim1$u[,1]
pc1.late <- clim2 %*% pc.clim2$u[,1]

salm.cov$pc1.clim1[salm.cov$year <= 1988] <- pc1.early
salm.cov$pc1.clim2[salm.cov$year > 1988] <- pc1.late

# And calculate the variance explained in each era.

var1 <- pc.clim1$d^2/sum(pc.clim1$d^2)
var2 <- pc.clim2$d^2/sum(pc.clim2$d^2)
var.plot <- 100*cbind(var1,var2)

```

And plot the loadings for each era.

```

load.plot <- cbind(pc.clim1$u[,1], pc.clim2$u[,1])
rownames(load.plot) <- include
load.plot <- load.plot[order(load.plot[,1]),]

```

Now I'll also calculate time-dependent correlations with SST and PDO.

```

# re-load and process SST data
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/NSF-GOA/sst.mnmean.v4.nc")

# extract dates
d <- dates(ncvar_get(nc, "time"), origin=c(1,15,1800))

# Pick start and end dates (Jan 1900-Dec 2015):
d <- d[553:1944]

# extract study area
# 54-62 deg. N, 200-226 deg. E
x <- ncvar_get(nc, "lon", start=101, count=14)
y <- ncvar_get(nc, "lat", start=14, count=5)

SST <- ncvar_get(nc, "sst", start=c(101,14,553), count=c(14,5,length(d)), verbose = F)

# Change data from a 3-D array to a matrix of monthly data by grid point:
# First, reverse order of dimensions ("transpose" array)
SST <- aperm(SST, 3:1)

# Reverse order of latitudes to be increasing for convenience (in later plotting)
SST <- SST[,5:1,]
# Also reverse corresponding vector of latitudes
y <- rev(y)
# Change to matrix with column for each grid point, rows for monthly means
SST <- matrix(SST, nrow=dim(SST)[1], ncol=prod(dim(SST)[2:3]))

# Keep track of corresponding latitudes and longitudes of each column:
lat <- rep(y, length(x))
lon <- rep(x, each = length(y))

```

```

dimnames(SST) <- list(as.character(d), paste("N", lat, "E", lon, sep=""))

# plot to check

# need to drop Bristol Bay cells
BB <- c("N58E200", "N58E202", "N56E200")
SST[,BB] <- NA

# now get area-weighted average of sst (not anomalies) in each month

# now get monthly means weighted by area, using an arithmetic mean
weight <- sqrt(cos(lat*pi/180))

SST.mu <- apply(SST, 1, function(x) weighted.mean(x,weight, na.rm=T))

# now separate out winter
m <- months(d)
yr <- as.numeric(as.character(years(d)))
win <- c("Nov", "Dec", "Jan", "Feb", "Mar")

mean <- data.frame(year=yr, month=m, mean=SST.mu)
mean$win.yr <- mean$year
mean$win.yr[mean$month %in% c("Nov", "Dec")] <- mean$win.yr[mean$month %in% c("Nov", "Dec")] + 1

win.mean <- mean[mean$month %in% win,]

SST.win <- tapply(win.mean$mean, win.mean$win.yr, mean)

salm.cov$SST.win <- SST.win[match(salm.cov$year, (names(SST.win)))]

# Now get time-specific data on PDO and SST correlations.

pdo <- read.csv("pdo.csv")

salm.cov$win.pdo <- pdo$NDJFM[pdo$YEAR %in% 1950:2012]

# here is the code for the modified Chelton Method
X <- zoo(salm.cov$win.pdo[salm.cov$year <= 1988])
Y <- zoo(salm.cov$pc1.clim1[salm.cov$year <= 1988])

# length of TS
missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
X[missY] <- NA # drop the corresponding observations in X

N <- sum(!is.na(X)) # sample size of pairwise observations!

# number of lags to use
J <- round(N/5)

# means of each TS
Xbar <- mean(X, na.rm=T)
Ybar <- mean(Y, na.rm=T)

```

```

# calculate denominators for eq. 6
ssX <- ssY <- NA

for (ii in 1:length(Y)){
  ssX[ii] <- (X[ii]-Xbar)^2
  ssY[ii] <- (Y[ii]-Ybar)^2
}
denomX <- sum(ssX, na.rm=T)
denomY <- sum(ssY, na.rm=T)

rXX <- rYY <- NA # vectors of autocorrelation parameters

for (j in 1:J){
  ssX <- ssY <- NA
  Xl <- lag(X, j)
  Yl <- lag(Y, j)
  for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
    # k <- 1
    ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
    ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
  }

  numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  numerY <- sum(ssY, na.rm=T)
  adj <- N/(N-j) # adjustor from eq. 7
  rXX[j] <- adj*numerX/denomX
  rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)
cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!
r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

# set up matrices to plot!

sst.pdo.cor <- sst.pdo.p <- matrix(NA, 2, 2)
dimnames(sst.pdo.cor) <- dimnames(sst.pdo.p) <- list(c("early", "late"), c("PDO", "SST"))

sst.pdo.cor[1,1] <- r.ac
sst.pdo.p[1,1] <- p

```

Now print results for pre-88/89 PDO-clim PC1 results.

```

r.ac

## [1] -0.8097396

p

## [1] 0.0003907819

Calculate for the late era.

X <- zoo(salm.cov$win.pdo[salm.cov$year > 1988])
Y <- zoo(salm.cov$pc1.clim2[salm.cov$year > 1988])

# length of TS
missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
X[missY] <- NA # drop the corresponding observations in X

N <- sum(!is.na(X)) # sample size of pairwise observations!

# number of lags to use
J <- round(N/5)

# means of each TS
Xbar <- mean(X, na.rm=T)
Ybar <- mean(Y, na.rm=T)

# calculate denominators for eq. 6
ssX <- ssY <- NA

for (ii in 1:length(Y)){
  ssX[ii] <- (X[ii]-Xbar)^2
  ssY[ii] <- (Y[ii]-Ybar)^2
}
denomX <- sum(ssX, na.rm=T)
denomY <- sum(ssY, na.rm=T)

rXX <- rYY <- NA # vectors of autocorrelation parameters

for (j in 1:J){
  ssX <- ssY <- NA
  Xl <- lag(X, j)
  Yl <- lag(Y, j)
  for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
    # k <- 1
    ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
    ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
  }

  numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  numerY <- sum(ssY, na.rm=T)
  adj <- N/(N-j) # adjustor from eq. 7
  rXX[j] <- adj*numerX/denomX
  rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)

```

```

cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!
r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

sst.pdo.cor[2,1] <- r.ac
sst.pdo.p[2,1] <- p

```

This is the correlations for PDO vs PC1 clim in the late period.

```
r.ac
```

```
## [1] -0.3149453
```

```
p
```

```
## [1] 0.1513076
```

Now the correlation between SST and clim PC1 in the early era.

```
####
# now correlations with sst
X <- zoo(SST.win[names(SST.win) %in% 1950:1988])
Y <- zoo(salm.cov$pc1.clim1[salm.cov$year <= 1988])

# length of TS
missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
X[missY] <- NA # drop the corresponding observations in X

N <- sum(!is.na(X)) # sample size of pairwise observations!

# number of lags to use
J <- round(N/5)

# means of each TS
Xbar <- mean(X, na.rm=T)
Ybar <- mean(Y, na.rm=T)

# calculate denominators for eq. 6
ssX <- ssY <- NA

for (ii in 1:length(Y)){
  ssX[ii] <- (X[ii]-Xbar)^2
  ssY[ii] <- (Y[ii]-Ybar)^2
}
denomX <- sum(ssX, na.rm=T)
denomY <- sum(ssY, na.rm=T)

rXX <- rYY <- NA # vectors of autocorrelation parameters

```

```

for (j in 1:J){
  ssX <- ssY <- NA
  Xl <- lag(X, j)
  Yl <- lag(Y, j)
  for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
    # k <- 1
    ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
    ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
  }

  numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  numerY <- sum(ssY, na.rm=T)
  adj <- N/(N-j) # adjustor from eq. 7
  rXX[j] <- adj*numerX/denomX
  rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)

cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!
r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

sst.pdo.cor[1,2] <- r.ac
sst.pdo.p[1,2] <- p

```

SST-clim PC1 correlations early period.

```
r.ac
```

```
## [1] -0.7533412
```

```
p
```

```
## [1] 0.001532159
```

Now SST-clim PC correlations late.

```

X <- zoo(SST.win[names(SST.win) %in% 1989:2012])
Y <- zoo(salm.cov$pc1.clim2[salm.cov$year > 1988])

```

```
# length of TS
```

```

missY <- is.na(Y) # identify NAs in Y! this doesn't have to be done for X as there are no NAs there!
X[missY] <- NA # drop the corresponding observations in X

```

```
N <- sum(!is.na(X)) # sample size of pairwise observations!
```

```
# number of lags to use
```

```
J <- round(N/5)
```

```

# means of each TS
Xbar <- mean(X, na.rm=T)
Ybar <- mean(Y, na.rm=T)

# calculate denominators for eq. 6
ssX <- ssY <- NA

for (ii in 1:length(Y)){
  ssX[ii] <- (X[ii]-Xbar)^2
  ssY[ii] <- (Y[ii]-Ybar)^2
}
denomX <- sum(ssX, na.rm=T)
denomY <- sum(ssY, na.rm=T)

rXX <- rYY <- NA # vectors of autocorrelation parameters

for (j in 1:J){
  ssX <- ssY <- NA
  Xl <- lag(X, j)
  Yl <- lag(Y, j)
  for(k in 1:(length(Y)-J)){ # loop through the numerator for eq. 6
    # k <- 1
    ssX[k] <- (X[k]-Xbar)*(Xl[k]-Xbar)
    ssY[k] <- (Y[k]-Ybar)*(Yl[k]-Ybar)
  }

  numerX <- sum(ssX, na.rm=T) # numerator from eq. 6
  numerY <- sum(ssY, na.rm=T)
  adj <- N/(N-j) # adjustor from eq. 7
  rXX[j] <- adj*numerX/denomX
  rYY[j] <- adj*numerY/denomY
}

# now calculate effective sample size (Nef)

cross.prod <- NA

for (j in 1:J){
  cross.prod[j] <- ((N-j)/N)*rXX[j]*rYY[j]
}

Nef <- 1/((1/N)+(2/N)*sum(cross.prod))
if(Nef > N) Nef <- N # constrain to be no greater than N!
r.ac <- cor(X,Y, use="pair") # "r actual"
t <- sqrt((Nef*r.ac^2)/(1-r.ac^2))
p <- 2*pt(-abs(t), Nef-2)

sst.pdo.cor[2,2] <- r.ac
sst.pdo.p[2,2] <- p

```

SST-PC1 clim correlation 1989-2012:

```
r.ac
```

```
## [1] -0.3217566
```

```
p
```

```
## [1] 0.1359032
```

Now combine into Fig. 3.

```
png("Fig 3.png", 11.4/2.54, 11.4/2.54, units="in", res=300)
```

```
# setup the layout
```

```
mt.cex <- 1.1
```

```
l.mar <- 3
```

```
l.cex <- 0.8
```

```
l.l <- 0.2
```

```
tc.l <- -0.2
```

```
par(tcl=tc.l, mgp=c(1.5,0.3,0), mfrow=c(2,2), cex.axis=0.8, cex.lab=0.8, oma=c(0,0,0.2,0))
```

```
par(las=2, cex=0.8, mar=c(5,3,1,0.5), tcl=0.2)
```

```
barplot(t(plot.cors), beside=T, ylim=c(-1,1), col=c("#0072B2", "#E69F00"), ylab = "r", xaxt="n")
```

```
abline(h=0)
```

```
text(seq(2,19, by=3), -1.05,
```

```
      srt = 60, adj= 1, xpd = TRUE,
```

```
      labels = rownames(plot.cors), cex=0.8)
```

```
#legend("topright", c("Before 1988/89", "After 1988/89"), text.col=c("#0072B2", "#E69F00"), bty="n", cex=0.8, box())
```

```
par(las=1)
```

```
mtext("a", adj=0, cex=1.1)
```

```
mtext("Correlation with SLPa", adj=0.5, cex=0.7)
```

```
par(cex=0.8, mar=c(4,3,1,1.5), tcl=0.2)
```

```
barplot(t(var.plot), beside=T, col=c("#0072B2", "#E69F00"), ylab = "% variance", xlab="Axis", ylim=c(0,10),  
abline(h=0)
```

```
# legend("topright", c("Before 1988/89", "After 1988/89"), text.col=c("#0072B2", "#E69F00"), bty="n", cex=0.8, box())
```

```
mtext("b", adj=0, cex=1.1)
```

```
mtext("PCA variance explained", adj=0.5, cex=0.7)
```

```
par(mar=c(3,4.5,0.5,0.5))
```

```
barplot(t(load.plot), beside=T, col=c("#0072B2", "#E69F00"), xlab="Loading", names.arg = rownames(load.plot),  
abline(v=0)
```

```
# legend("topright", c("Before 1988/89", "After 1988/89"), text.col=c("#0072B2", "#E69F00"), bty="n", cex=0.8, box())
```

```
mtext("c", adj=0, cex=1.1)
```

```
mtext("PC1 loadings", adj=0.5, cex=0.7)
```

```
par(las=1, cex=0.8, mar=c(6,3,0.5,1.5), tcl=0.2)
```

```
barplot(sst.pdo.cor, beside=T, col=c("#0072B2", "#E69F00"), xlab="", ylab="r", ylim=c(-0.9,0))  
box()
```

```
par(las=1)
```

```
mtext("d", adj=0, cex=1.1)
```

```
mtext("Correlation with PC1", adj=0.5, cex=0.7)
```



```

mtext(c("1950-1988", "1989-2012"), col=c("#0072B2", "#E69F00"), side=1, line=c(2, 4), cex=1.5, at=-0.1,
dev.off()

```

```

## pdf
## 2

```

Fig. 4.

Comparison of analogue and non-analogue regression models for climate-salmon catch relationships.

```

# will re-load SST data
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/NSF-GOA/sst.mnmean.v4.nc")

d1 <- dates(ncvar_get(nc, "time"), origin=c(1,15,1800))

# Pick start and end dates (Jan 1964-Dec 2012):
d1 <- d1[1321:1908]

# now just study area
# 54-62 deg. N, 200-226 deg. E
t.x2 <- ncvar_get(nc, "lon", start=101, count=14)
t.y2 <- ncvar_get(nc, "lat", start=14, count=5)

# study area
SST2 <- ncvar_get(nc, "sst", start=c(101,14,1321), count=c(14,5,length(d1)), verbose = F)

SST2 <- aperm(SST2, 3:1) # First, reverse order of dimensions ("transpose" array)
SST2 <- SST2[,5:1,] # Reverse order of latitudes to be increasing for convenience (in later plotting)
t.y2 <- rev(t.y2) # Also reverse corresponding vector of latitudes
SST2 <- matrix(SST2, nrow=dim(SST2)[1], ncol=prod(dim(SST2)[2:3])) # Change to matrix

# Keep track of corresponding latitudes and longitudes of each column:
t.lat2 <- rep(t.y2, length(t.x2)) # Vector of latitudes
t.lon2 <- rep(t.x2, each = length(t.y2)) # Vector of longitudes
dimnames(SST2) <- list(as.character(d1), paste("N", t.lat2, "E", t.lon2, sep=""))

# need to drop Bristol Bay cells from study area!
BB <- c("N58E200", "N58E202", "N56E200")
SST2[,BB] <- NA

m <- months(d1)
yr <- as.numeric(as.character(years(d1)))

# now get area-weighted mean anomaly on smoothed
f <- function(x) tapply(x, m, mean, na.rm=T) # function to compute monthly means for a single time ser

mu2 <- apply(SST2, 2, f) # Compute monthly means for each time series (location)
mu2 <- mu2[rep(1:12, round(length(d1)/12)),]
SST2.anom <- SST2 - mu2 # Compute matrix of anomalies!

```

```

# now get monthly means weighted by area
weight <- sqrt(cos(t.lat2*pi/180))
SST2.mu <- apply(SST2.anom, 1, function(x) weighted.mean(x,weight, na.rm=T))

SST2.mu <- data.frame(month=m, year=yr, win.yr=yr, SST=SST2.mu)
add <- c("Nov", "Dec")
SST2.mu$win.yr[SST2.mu$month %in% add] <- SST2.mu$win.yr[SST2.mu$month %in% add]+1

# and get winter means
win <- c("Nov", "Dec", "Jan", "Feb", "Mar")
win.SST <- filter(SST2.mu, month %in% win)
win.SST <- tapply(win.SST$SST, win.SST$win.yr, mean)

# now pdo and npgo
# (this is a version with NDJFM means already calculated)
indices <- read.csv("winter_npgo_pdo.csv")

# and the salmon catch data!
catch <- read.csv("total_goa_catch.csv", row.names=1)

# limit to 1965:2012
pc.dat <- data.frame(sock=scale(catch$Sockeye[rownames(catch) %in% 1965:2012]),
                     pink=scale(catch$Pink[rownames(catch) %in% 1965:2012]),
                     chum=scale(catch$Chum[rownames(catch) %in% 1965:2012]),
                     coho=scale(catch$Coho[rownames(catch) %in% 1965:2012]))

pca <- prcomp(pc.dat)

data <- data_frame(year=1965:2012, catch=pca$x[,1], SST=win.SST[names(win.SST) %in% 1965:2012], PDO=indices$PDO)
data$era <- "late"
data$era[data$year <= 1988] <- "early"

# list of candidate models
mods <- c("SST", "PDO", "NPGO", "PDO + NPGO", "SST*era", "PDO*era", "NPGO*era", "PDO*era + NPGO*era")
type <- c(rep("stationary", 4), rep("nonstationary", 4))

models <- data.frame(expl=mods, type=type)
models$AICc <- NA
models$res.ar <- NA

for(i in 1:length(mods)){
  # i <- 3
  form <- as.formula(paste("catch ~ ", models$expl[i], sep=""))
  mod <- lm(formula=form, data=data)
  models$AICc[i] <- AICc(mod)

  # now test for autocorrelation in residuals
  res <- residuals(mod)
  test <- ar(res, aic=F, order.max=1)
  models$res.ar[i] <- test$ar
}

models <- arrange(models, AICc)

```

```

models$dAICc <- models$AICc-min(models$AICc)

pdf("Fig 4.pdf", 11.4/2.54, 7/2.54)
mm <- matrix(1:2, 1,2)
layout(mm, height=c(1,1),width=c(1,1))
par(las=2, mar=c(5.5,2,0,0.5), cex.axis= 0.8, tcl = 0.2, cex.lab=0.8, mgp=c(1.2,0.3,0), oma=c(0,0,2,0))
cols <- c(rep("#56B4E9",4), rep("#D55E00", 4))
models <- arrange(models, AICc)

barplot(models$dAICc, beside=T, col=cols, ylab=expression(paste(Delta, "-AICc", sep="")), cex.names = 0.7)
par(las=1)
mtext("a", adj=0.11, line=-1, cex=1.2)
text(seq(0.5,9.9,by=1.25), -3,
      srt = 60, adj= 1, xpd = TRUE,
      labels = models$expl, cex=0.65)

par(mgp=c(1.3,0.3,0), las=2)
barplot(models$res.ar, beside=T, col=cols,ylab="Residual AR(1)", cex.names=0.7)

text(seq(0.5,9.9,by=1.25), -0.04,
      srt = 60, adj= 1, xpd=T,
      labels = models$expl, cex=0.65)

par(las=1)
mtext("b", adj=0.11, line=-1, cex=1.2)
mtext("Non-analogue models", col="#56B4E9", adj=0.18, line=1, outer=T)
mtext("Analogue models", col="#D55E00", adj=0.165, line=0.2, outer=T)
dev.off()

## pdf
## 2

```

Fig. 5

Nonstationary SST-salmon relationships.

Begin with the timing of changing relationships between catches and SST.

```

# load and process SST data to get winter GOA means
nc <- nc_open("/Users/MikeLitzow 1/Documents/R/NSF-GOA/sst.mnmean.v4.nc")

# extract dates
d <- dates(ncvar_get(nc, "time"), origin=c(1,15,1800))

# Pick start and end dates (Jan 1900-Dec 2015):
d <- d[553:1944]

# extract study area
# 54-62 deg. N, 200-226 deg. E
x <- ncvar_get(nc, "lon", start=101, count=14)
y <- ncvar_get(nc, "lat", start=14, count=5)

SST <- ncvar_get(nc, "sst", start=c(101,14,553), count=c(14,5,length(d)))

```

```

# Change data from a 3-D array to a matrix of monthly data by grid point:
SST <- aperm(SST, 3:1)

# Reverse order of latitudes to be increasing for convenience (in later plotting)
SST <- SST[,5:1,]
# Also reverse corresponding vector of latitudes
y <- rev(y)
# Change to matrix with column for each grid point, rows for monthly means
SST <- matrix(SST, nrow=dim(SST)[1], ncol=prod(dim(SST)[2:3]))

# Keep track of corresponding latitudes and longitudes of each column:
lat <- rep(y, length(x))
lon <- rep(x, each = length(y))
dimnames(SST) <- list(as.character(d), paste("N", lat, "E", lon, sep=""))

# drop Bristol Bay cells
BB <- c("N58E200", "N58E202", "N56E200")
SST[,BB] <- NA

# now get monthly means weighted by area, using an arithmetic mean
weight <- sqrt(cos(lat*pi/180))

SST.mu <- apply(SST, 1, function(x) weighted.mean(x,weight, na.rm=T))

# now separate out winter
m <- months(d)
yr <- as.numeric(as.character(years(d)))
win <- c("Nov", "Dec", "Jan", "Feb", "Mar")

mean <- data.frame(year=yr, month=m, mean=SST.mu)
mean$win.yr <- mean$year

# assigns Nov and Dec to "winter year" corresponding to Jan
mean$win.yr[mean$month %in% c("Nov", "Dec")] <- mean$win.yr[mean$month %in% c("Nov", "Dec")] + 1

win.mean <- mean[mean$month %in% win,] # separate out NDJFM

SST.win <- tapply(win.mean$mean, win.mean$win.yr, mean)

```

Now define the best timing of the change. First step is to calculate PCA of catch.

```

data <- read.csv("total_goa_catch.csv", row.names=1)
data$year <- 1900:2014
pc.dat <- data.frame(sock=scale(data$Sockeye[data$year %in% 1965:2012]),
                     pink=scale(data$Pink[data$year %in% 1965:2012]),
                     chum=scale(data$Chum[data$year %in% 1965:2012]),
                     coho=scale(data$Coho[data$year %in% 1965:2012]))

pca <- prcomp(pc.dat)

```

These are the loadings and proportion of variance for catch PCA.

```
pca$rotation
```

##	PC1	PC2	PC3	PC4
----	-----	-----	-----	-----

```
## sock 0.4853216 0.7182108 -0.4885716485 0.09966928
## pink 0.5155319 -0.2333578 0.0008661709 -0.82448184
## chum 0.4942625 -0.6335569 -0.3408122029 0.48801354
## coho 0.5043753 0.1682945 0.8032085884 0.26858614
```

```
summary(pca)
```

```
## Importance of components:
##              PC1      PC2      PC3      PC4
## Standard deviation    1.8454 0.54549 0.4257 0.34030
## Proportion of Variance 0.8514 0.07439 0.0453 0.02895
## Cumulative Proportion 0.8514 0.92575 0.9710 1.00000
```

Now the model selection process comparing different threshold years. The explanatory variable is GOA winter SST, smoothed with a 3-yr rolling mean.

```
temp <- SST.win[names(SST.win) %in% 1900:2014]
sst.3 <- rollapply(temp, 3, mean, na.rm=T, fill=NA)
names(sst.3) <- 1900:2014

dat.thr <- data.frame(pc1=pca$x[,1], sst.3=sst.3[names(sst.3) %in% 1965:2012], year=1965:2012)

thr <- 1974:2002 # these candidate thresholds maintain at least 20% of the data in each era
pc.out <- NA # object to hold results (AICc scores)

for(i in 1:length(thr)){ # loop each threshold
  # i <- 1
  dat.thr$era <- "early"
  dat.thr$era[dat.thr$year > thr[i]] <- "late"
  mod <- gls(pc1 ~ sst.3*era, data=dat.thr, correlation = corAR1(form = ~1), method="ML")
  pc.out[i] <- AICc(mod)
}
```

Now compare the sst-catch regression pre/post 88/89

```
dat <- data.frame(pc1=pca$x[,1], sst=sst.3[names(sst.3) %in% 1965:2012],
  year=1965:2012)

dat$era <- "early"
dat$era[dat$year > 1988] <- "late"

mod <- gls(pc1 ~ sst*era, data=dat, correlation=corAR1()) # GLS model allowing correlated residuals
```

Get the t statistic and p-value for the sst x era interaction.

```
summary(mod)$tTable[4,3]
```

```
## [1] -6.037955
```

```
2*pt(-summary(mod)$tTable[4,3], 48, lower=FALSE)
```

```
## [1] 2.194019e-07
```

And now the era-specific p-values for catch PC1 on SST. Again, these use GLS models allowing for autocorrelated residuals.

```
dat1 <- dat[dat$era=="early",]
dat2 <- dat[dat$era=="late",]
```

1965:1988 results:

```
mod1 <- gls(pc1 ~ sst, data=dat1, correlation=corAR1())
summary(mod1)$tTable
```

```
##                Value Std.Error   t-value    p-value
## (Intercept) -22.587018 2.4487193 -9.224013 5.139428e-09
## sst          3.785603 0.4355044  8.692457 1.448351e-08
```

1989:2012 results:

```
mod2 <- gls(pc1 ~ sst, data=dat2, correlation=corAR1())
summary(mod2)$tTable
```

```
##                Value Std.Error   t-value    p-value
## (Intercept) -0.2417713 2.1559844 -0.1121397 0.9117296
## sst          0.2850030 0.3788861  0.7522129 0.4598960
```

And now prepare the pieces for a plot.

```
nd1 <- data.frame(sst = seq(min(dat$sst[dat$year<1989]), max(dat$sst[dat$year<1989])
                      , length.out=100))

nd2 <- data.frame(sst = seq(min(dat$sst[dat$year>1988]), max(dat$sst[dat$year>1988])
                      , length.out=100))

pc.p1 <- predict(gls(pc1 ~ sst, data=dat[dat$year<1989,], correlation=corAR1()), type="response",
                 se.fit=T, newdata=nd1)

pc.p2 <- predict(gls(pc1 ~ sst, data=dat[dat$year>1988,], correlation=corAR1()), type="response",
                 se.fit=T, newdata=nd1)
```

And now the hatchery/wild comparison.

```
# load the area-specific catch data
data <- read.csv("long_area_salmon_catch.csv", row.names=1)

# remove chinook
ch <- c(1, 6, 11, 16, 21, 26)
data <- data[,-ch]

# separate the wild and hatchery areas
wild <- c("se.pi", "pws.co", "ci.so", "ci.cm", "ci.co", "kod.so", "kod.co", "ch.so", "ch.co", "ch.pi",
wild.dat <- data[rownames(data) %in% 1965:2012, colnames(data)%in%wild]
not.dat <- data[rownames(data) %in% 1965:2012, !colnames(data)%in%wild]

coeffs <- ses <- NA # objects to save mixed-effect model output

# need to stack data (wrote this before I discovered melt!)
st.wild <- stack(wild.dat)
colnames(st.wild) <- c("catch", "group")
st.wild$sst.3 <- rep(sst.3[names(sst.3) %in% 1965:2012])
st.wild$year <- rep(1965:2012)

st.wild$era[st.wild$year<=1988] <- "early"
st.wild$era[st.wild$year>1988] <- "late"

st.hatch <- stack(not.dat)
```

```

colnames(st.hatch) <- c("catch", "group")
st.hatch$sst.3 <- rep(sst.3[names(sst.3) %in% 1965:2012])
st.hatch$year <- rep(1965:2012)

st.hatch$era[st.hatch$year<=1988] <- "early"
st.hatch$era[st.hatch$year>1988] <- "late"

# fit wild area model
wild <- lme(catch ~ sst.3*era, random = ~ sst.3 | group, corAR1(form = ~year|group), data=st.wild, 1

# and hatchery areas model
hatch <- lme(catch ~ sst.3*era, random = ~ sst.3 | group, corAR1(form = ~year|group), data=st.hatch

```

Print the t statistic and p-value for the wild fishery areas.

```
summary(wild)$tTable
```

##	Value	Std.Error	DF	t-value	p-value
## (Intercept)	2.2849129	0.5287849	702	4.321063	1.777004e-05
## sst.3	0.5954536	0.0912338	702	6.526678	1.288555e-10
## eralate	3.7194422	0.6834385	702	5.442249	7.276362e-08
## sst.3:eralate	-0.6128772	0.1202553	702	-5.096469	4.453958e-07

And the p-value for the hatchery-subsidized areas.

```
summary(hatch)$tTable
```

##	Value	Std.Error	DF	t-value	p-value
## (Intercept)	3.6857299	0.43408110	420	8.490879	3.575968e-16
## sst.3	0.4414097	0.09128964	420	4.835266	1.868910e-06
## eralate	2.6212105	0.60008258	420	4.368083	1.580944e-05
## sst.3:eralate	-0.3980938	0.10586456	420	-3.760406	1.936910e-04

And now the spawner-recruit analysis. Begin by running the best model for each species, as earlier established by exhaustive model selection and residual examination.

```

# load the pink salmon data!

data <- read.csv("pink_SR_data.csv")
# these are the data that I provided to Bethany/Lorenzo/Patri

# restrict to GOA, and use=1 to make sure we only include good data
pink <- filter(data, GOA==1, use==1)

# drop PWS
drop <- grep("PWS", pink$full.stock)
pink <- pink[-drop,]
# set "era" factor
pink$era <- "early"
pink$era[pink$entry.yr > 1988] <- "late"

# and log recruits
pink$ln.recr <- log(pink$recruits)

# fit the best model
pink.best <- lme(ln.recr ~ loc.sst*era + full.stock:spawners + offset(log(spawners)), random = ~ loc.sst

```

```

data <- read.csv("sockeye_SR_data.csv")

# these are the data that I provided to Bethany/Lorenzo/Patri

sock <- filter(data, GOA==1, use==1, brood.yr >=1960)

sock$ln.recr <- log(sock$recruits)
sock$era <- "early"
sock$era[sock$entry.yr > 1988] <- "late"

sock.best <- lme(ln.recr ~ loc.sst*era + stock:spawners + offset(log(spawners)), random = ~ loc.sst | s

data <- read.csv("chum_SR_data.csv")
chum <- filter(data, GOA==1, use==1)

chum$ln.recr <- log(chum$recruits)
chum$era <- "early"
chum$era[chum$entry.yr>1988] <- "late"

chum.best <- lme(ln.recr ~ loc.sst*era + stock:spawners + offset(log(spawners)), random = ~ loc.sst | s

```

Those are the best models. Run the likelihood ratio test for each spp. First, the ANOVA results and exact p-value for pinks.

```

# likelihood ratio test: ANOVA on the models with and without the interaction terms
pink.reduced <- lme(ln.recr ~ loc.sst + era + full.stock:spawners + offset(log(spawners)), random = ~ loc.sst | s

pink.test <- anova(pink.best, pink.reduced)
pink.test

```

```

##           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## pink.best      1 29 1204.885 1325.804 -573.4427
## pink.reduced   2 28 1231.345 1348.094 -587.6727 1 vs 2 28.45996 <.0001

```

```
pink.test$`p-value`
```

```
## [1]          NA 9.565622e-08
```

Sockeye results.

```

sock.reduced <- lme(ln.recr ~ loc.sst + era + stock:spawners + offset(log(spawners)), random = ~ loc.sst | s

sock.test <- anova(sock.best, sock.reduced)
sock.test

```

```

##           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## sock.best      1 18 791.7438 862.9022 -377.8719
## sock.reduced   2 17 802.7142 869.9193 -384.3571 1 vs 2 12.97035 3e-04

```

```
sock.test$`p-value`
```

```
## [1]          NA 0.0003164634
```

And chum.

```

chum.reduced <- lme(ln.recr ~ loc.sst + era + stock:spawners + offset(log(spawners)), random = ~ loc.sst | s

chum.test <- anova(chum.best, chum.reduced)

```



```
chum.test
```

```
##           Model df      AIC      BIC    logLik    Test  L.Ratio p-value
## chum.best      1 19 915.8871 992.4244 -438.9435
## chum.reduced    2 18 915.5854 988.0944 -439.7927 1 vs 2 1.698258 0.1925
```

```
chum.test$p-value`
```

```
## [1]      NA 0.192516
```

And the chi-sq test for overall P-value.

```
pp <- c(pink.test$p-value`[2], sock.test$p-value`[2], chum.test$p-value`[2])
test <- -2*sum(log(pp))
```

```
print(paste("Chi-squared w/ 6 df = ", (test), sep=""))
```

```
## [1] "Chi-squared w/ 6 df = 51.7367679654206"
```

```
print(paste("Overall p-value: ", round(pchisq(test, 6, lower.tail=F),13), sep=""))
```

```
## [1] "Overall p-value: 2.1065e-09"
```

Now prepare these results to plot.

```
# collect the fixed effects and std errors to plot
```

```
plot.out <- plot.CI <- matrix(nrow=3, ncol=2)
```

```
dimnames(plot.out) <- dimnames(plot.CI) <- list(c("Chum", "Sockeye", "Pink"), c("Pre-1988/89", "Post-1988/89"))
```

```
plot.out[,2] <- c(summary(chum.best)$tTable[2,1], summary(sock.best)$tTable[2,1], summary(pink.best)$tTable[2,1])
```

```
plot.out[,1] <- c(summary(chum.best)$tTable[2,1]+summary(chum.best)$tTable[4,1], summary(sock.best)$tTable[2,1]+summary(sock.best)$tTable[4,1], summary(pink.best)$tTable[2,1]+summary(pink.best)$tTable[4,1])
```

```
plot.CI[,2] <- c(summary(chum.best)$tTable[2,2], summary(sock.best)$tTable[2,2], summary(pink.best)$tTable[2,2])
```

```
plot.CI[,1] <- c(summary(chum.best)$tTable[4,2], summary(sock.best)$tTable[4,2], summary(pink.best)$tTable[4,2])
```

The last step is to run the same model-selection approach to identify best-supported change in response to SST by pink SR. First, run the AICc code.

```
thr.pi <- 1978:1994 # 15-yr minimum periods on either side of candidate thresholds
```

```
plot.thr.pi <- thr.pi + 0.5 # plot with an offset of a half year, i.e., between threshold years
```

```
AICc.pi <- NA # object to save output
```

```
for(i in 1:length(thr.pi)){
```

```
  # i <- 1
```

```
  pink$era <- "early"
```

```
  pink$era[pink$entry.yr > thr.pi[i]] <- "late"
```

```
  mod <- lme(ln.recr ~ loc.sst*era + full.stock:spawners + offset(log(spawners)), random = ~ loc.sst | site)
```

```
  AICc.pi[i] <- AICc(mod)
```

```
}
```

```
# get d-AICc for odd and even threshold years!
```

```
odd <- odd(thr.pi)
```

```
min.odd <- min(AICc.pi[odd])
```

```
even <- even(thr.pi)
```

```
min.even <- min(AICc.pi[even])
```

```
min.AIC <- rep(NA, length(AICc.pi))
```

```

min.AIC[odd] <- min.odd
min.AIC[even] <- min.even

dAICc.pi <- AICc.pi-min.AIC

```

And plot all the panels for the submitted version of Fig. 5.

```

pdf("Fig 5.pdf", 1.3*11.4/2.54, 1.3*(2/3)*10.5/2.54)

par(las=1, mfrow=c(2,2), mar=c(2,3,0.5,1), tcl=0.2, mgp=c(1.9,0.2,0))
mt.cex=0.7
cols <- c("#E69F00", "#56B4E9", "#D55E00", "#009E73")

par(mgp=c(1,0.2,0))
plot.thr <- thr+0.5
plot.aic <- pc.out-min(pc.out)
plot(plot.thr, plot.aic, type="l", xlab="Threshold year", ylab=expression(paste(Delta, "-AICc", sep="")))
abline(v=1988.5, lty=2)
#mtext("B) Timing of change", adj=0, cex=mt.cex)
mtext(" a", line=-1.6, cex=1.4, adj=0)

plot(dat$sst[dat$year<1989], dat$pc1[dat$year<1989], ylim=range(dat$pc1),
      xlim=range(dat$sst), pch=21, bg="#0072B2", xlab="Winter SST (°C)", ylab="PC1", cex.lab=0.9)
points(dat$sst[dat$year>1988], dat$pc1[dat$year>1988], pch=21, bg="#E69F00")
lines(nd1$sst, pc.p1$fit, col="#0072B2")
lines(nd2$sst, pc.p2$fit, col="#E69F00")

# set polygon coordinates
xpoly <- c(nd1$sst[1], nd1$sst, nd1$sst[100:1])
ypoly <- c(pc.p1$fit[1]+1.96*pc.p1$se.fit[1], pc.p1$fit-1.96*pc.p1$se.fit,
           pc.p1$fit[100:1]+1.96*pc.p1$se.fit[100:1])

polygon(xpoly,ypoly, border=NA, col=alpha("#0072B2", 0.25))

xpoly <- c(nd2$sst[1], nd2$sst, nd2$sst[100:1])
ypoly <- c(pc.p2$fit[1]+1.96*pc.p2$se.fit[1], pc.p2$fit-1.96*pc.p2$se.fit,
           pc.p2$fit[100:1]+1.96*pc.p2$se.fit[100:1])

polygon(xpoly,ypoly, border=NA, col=alpha("#E69F00", 0.25))

legend(5.35, -1.9, c("Before 1988/89", "After 1988/89"),
      text.col=c("#0072B2", "#E69F00"), cex=1.1, bty="n", horiz=F)

#mtext("C) Nonstationary SST-catch", adj=0, cex=mt.cex)
mtext(" b", line=-1.6, cex=1.4, adj=0)

# SR SST effects by era
cols <- c("#009E73", "#E69F00", "#D55E00")
barplot2(plot.out, beside=T, horiz=T, xlim=c(-0.4, 0.85),
          plot.ci=T, ci.l=plot.out-1.96*plot.CI, ci.u=plot.out+1.96*plot.CI, space=c(0.2,0.2,0.2,1,0.2,0.2),
          xlab="SST coefficient", col=cols, cex.lab=0.9, axisnames=F)
text(-0.38, 6.22, "Pink", col=cols[3], adj=0, cex=1.05)
text(-0.38, 5.22, "Sockeye", col=cols[2], adj=0, cex=1.05)
text(-0.38, 4.42, "Chum", col=cols[1], adj=0, cex=1.05)

```

```

box()
abline(v=0, lwd=0.7)
abline(h=4, col="black", lty=2, lwd=0.7)
text(0.265, 4.6, "Before 1988/89", adj=0, cex=1)
text(0.31, 3.4, "After 1988/89", adj=0, cex=1)
#mtext("E) Nonstationary SST-productivity", adj=0, cex=mt.cex)
mtext(" c", line=-1.6, cex=1.4, adj=0)
plot.thr <- thr.pi+0.5
plot(plot.thr, dAICc.pi, type="l", col="red", ylab=expression(paste(Delta, "-AICc", sep="")), xlab="Thr")
abline(v=1988.5, lty=2)
#mtext("F) Timing of change", adj=0, cex=mt.cex)
mtext(" d", line=-1.6, cex=1.4, adj=0)
dev.off()

## pdf
## 2

```

Fig. 6.

Non-stationary relationships between individual environmental variables and recruits per spawner.

```

# load the time series for the six environmental variables
salm.cov <- read.csv("GOA_salmon_covariates.csv")

# will re-load and process the SR data for each species
# load the pink salmon data
data <- read.csv("pink_SR_data.csv")

# restrict to GOA, and use=1 to make sure we only include good data
pink <- filter(data, GOA==1, use==1)

# drop PWS
drop <- grep("PWS", pink$full.stock)
pink <- pink[-drop,]

# and log recruits
pink$ln.recr <- log(pink$recruits)

# set up objects to save results
p.coef <- p.se <- p.p <- matrix(nrow=6, ncol=2)
dimnames(p.coef) <- dimnames(p.se) <- dimnames(p.p) <- list(colnames(salm.cov)[2:7], c("65.88", "89.10"))

# now we'll loop through the salmon covariates object
for(i in 1:6){

  pink$covar <- salm.cov[match(pink$entry.yr, salm.cov$year),i]

  # pre-88/89
  mod <- lme(ln.recr ~ covar + full.stock:spawners + offset(log(spawners)), data=pink[pink$entry.yr <= 1988,],
             method="REML", correlation=corAR1(form= ~ 1 | full.stock), na.action=na.exclude)

  p.p[i,1] <- summary(mod)$tTable[2,5]
  p.coef[i,1] <- summary(mod)$tTable[2,1]
}

```

```

p.se[i,1] <- summary(mod)$tTable[2,2]

# post-88/89
mod <- lme(ln.recr ~ covar + full.stock:spawners + offset(log(spawners)), data=pink[pink$entry.yr >= 1988,],
          method="REML", correlation=corAR1(form= ~ 1 | full.stock), na.action=na.exclude)

p.p[i,2] <- summary(mod)$tTable[2,5]
p.coef[i,2] <- summary(mod)$tTable[2,1]
p.se[i,2] <- summary(mod)$tTable[2,2]
}

# sockeye

data <- read.csv("sockeye_SR_data.csv")

sock <- filter(data, GOA==1, use==1, brood.yr >=1960) # limit to 1960 and later

sock$ln.recr <- log(sock$recruits)

s.coef <- s.se <- s.p <- matrix(nrow=6, ncol=2)
dimnames(s.coef) <- dimnames(s.se) <- dimnames(s.p) <- list(colnames(salm.cov)[2:7], c("65.88", "89.10"))

# now loop through the salmon covariates object
for(i in 1:6){
  # i <- 3
  sock$covar <- salm.cov[match(sock$entry.yr, salm.cov$year),i]

  # pre-88/89
  mod <- lme(ln.recr ~ covar + stock:spawners + offset(log(spawners)), data=sock[sock$entry.yr <= 1988,],
            method="REML", correlation=corAR1(form= ~ 1 | stock), na.action=na.exclude)

  s.p[i,1] <- summary(mod)$tTable[2,5]
  s.coef[i,1] <- summary(mod)$tTable[2,1]
  s.se[i,1] <- summary(mod)$tTable[2,2]

  # post-88/89
  mod <- lme(ln.recr ~ covar + stock:spawners + offset(log(spawners)), data=sock[sock$entry.yr >= 1989,],
            method="REML", correlation=corAR1(form= ~ 1 | stock), na.action=na.exclude)

  s.p[i,2] <- summary(mod)$tTable[2,5]
  s.coef[i,2] <- summary(mod)$tTable[2,1]
  s.se[i,2] <- summary(mod)$tTable[2,2]
}

#####
# chum

data <- read.csv("chum_SR_data.csv")

chum <- filter(data, GOA==1, use==1)

chum$ln.recr <- log(chum$recruits)

```

```

c.coef <- c.se <- c.p <- matrix(nrow=6, ncol=2)
dimnames(c.coef) <- dimnames(c.se) <- dimnames(c.p) <- list(colnames(salm.cov)[2:7], c("65.88", "89.10"))

# now we'll loop through the salmon covariates object

for(i in 1:6){

  chum$covar <- salm.cov[match(chum$entry.yr, salm.cov$year),i]

  # pre-88/89
  mod <- lme(ln.recr ~ covar + stock:spawners + offset(log(spawners)), data=chum[chum$entry.yr <= 1988,],
    method="REML", correlation=corAR1(form= ~ 1 | stock), na.action=na.exclude)

  c.p[i,1] <- summary(mod)$tTable[2,5]
  c.coef[i,1] <- summary(mod)$tTable[2,1]
  c.se[i,1] <- summary(mod)$tTable[2,2]

  # post-88/89
  mod <- lme(ln.recr ~ covar + stock:spawners + offset(log(spawners)), data=chum[chum$entry.yr >= 1989,],
    method="REML", correlation=corAR1(form= ~ 1 | stock), na.action=na.exclude)

  c.p[i,2] <- summary(mod)$tTable[2,5]
  c.coef[i,2] <- summary(mod)$tTable[2,1]
  c.se[i,2] <- summary(mod)$tTable[2,2]
}

# put the results together!
# combined p values object
p.vals <- data.frame(covar=rep(rownames(p.p), 3), species=c(rep("pink", nrow(p.p)), rep("sockeye", nrow(p.p))),
  p.p, s.p, c.p)

p.vals[,3:4] <- rbind(p.p, s.p, c.p)
names(p.vals)[3:4] <- c("p.65.88", "p.89.10")

# now! get adjusted alpha for the entire time series!
# following the approach of Verhoeven et al. 2005 (Dikos)
all.p <- c(p.vals[,3], p.vals[,4])

i <- 1:length(all.p)

FDR.a <- (0.05/length(all.p))*i
FDR.r <- all.p
FDR.r <- FDR.r[order(-FDR.r)]
FDR.a <- FDR.a[order(-FDR.a)]
combine <- cbind(FDR.r, FDR.a)
alpha <- max(combine[which(combine[,1]<=combine[,2]),1])

means <- tapply(p.vals$p.65.88, p.vals$covar, mean)
means <- means[order(means)]

this <- names(means)

p.vals$order <- rep(c(3,2,1,6,4,5),3)

```

```

p.vals <- arrange(p.vals, order)

early <- tapply(p.vals$p.65.88, list(p.vals$covar, p.vals$species), mean)

# reorder to plot
early <- early[c(4,1,2,3,6,5), c(2,3,1)]

late <- tapply(p.vals$p.89.10, list(p.vals$covar, p.vals$species), mean)

# reorder
late <- late[c(4,1,2,3,6,5), c(2,3,1)]

# and plot
pdf("Fig 6.pdf", 13/2.54, 9/2.54)
labs <- c("Salinity", "Freshwater", "SLP grad.", "Advection", "Wind stress", "Downwelling")
par(las=2, mar=c(5,2.5,2,0), cex.axis= 0.9, tcl = 0.2, cex.lab=0.9, mgp=c(1.7,0.3,0), oma=c(0,0.5,1,0),
barplot(t(early), beside=T, log="y", ylim=c(1e-11,1), col=c("#D55E00", "#E69F00", "#009E73"), axisnames=FALSE,
abline(h=alpha, lty=2)
# legend(x=-2, y=250, c("Pink", "Sockeye", "Chum"), bty="n", horiz=T, text.col = c("#D55E00", "#E69F00", "#009E73"),
axis(2, labels=c(expression(paste("10"^-2)), expression(paste("10"^-5)), expression(paste("10"^-8))),
par(las=1)
mtext("Pink", outer=T, adj=0.2, col="#D55E00", cex=1.3, line=-0.4)
mtext("Sockeye", outer=T, adj=0.5, col="#E69F00", cex=1.3, line=-0.4)
mtext("Chum", outer=T, adj=0.85, col="#009E73", cex=1.3, line=-0.4)
mtext("a", adj=0, line=0.2, cex=1.5)
mtext("Before 1988/89", adj=0.3, cex=1, line=0.2)
text(seq(2.5,22.5,by=4), 10^-11.5,
srt = 60, adj= 1, xpd = TRUE,
labels = labs, cex=1)

par(mar=c(5,1,2,1.5), las=2)
barplot(t(late), beside=T, log="y", ylim=c(1e-11,1), col=c("#D55E00", "#E69F00", "#009E73"), yaxt="n",
abline(h=alpha, lty=2)
axis(2, labels=c("", "", "", ""), at=c(1e-2, 1e-5, 1e-8, 1e-11))

text(seq(2.5,22.5,by=4), 10^-11.5,
srt = 60, adj= 1, xpd = TRUE,
labels = labs, cex=1)

par(las=3)
mtext("P-value", outer=T, side=2, line=-0.5)
par(las=1)
mtext("b", adj=0, line=0.2, cex=1.5)
mtext("After 1988/89", adj=0.3, cex=1, line=0.2)

dev.off()

## pdf
## 2

```