Temperature to Nitrate Lookup Tables for the Southern California Bight

Please cite the following publication:

Snyder, J. N., Bell, T. W., Siegel, D. A., Nidzieko, N. J., & Cavanaugh, K. C. (2020). Sea Surface Temperature Imagery Elucidates Spatiotemporal Nutrient Patterns for Offshore Kelp Aquaculture Siting in the Southern California Bight, 7(22), 1–14. https://doi.org/10.3389/fmars.2020.00022

Temperature and nitrate measurements were compiled from three campaigns in the Southern California Bight: UCSB Plumes and Blooms cruises, Santa Barbara Coastal Long Term Ecological Research cruises, and CalCOFI cruises[1,2,3]. Each dataset was binned by month and depth in MatLab, and filtered to include only surface measurements between 0 to 3 meters (we assume mixing of surface waters above 3 meters). Temperature and nitrate data from all cruises were compiled and binned into two groups according to a seasonal, climatological pattern in the Southern California Bight: cool and wet winter months (December – May), and warm and dry summer months (June – November)[4], using various functions in MatLab. These seasons exemplify the difference between upwelling events that typically occur during winter/spring months, versus the relaxed upwelling period typical of summer/fall months. Temperature and nitrate data were binned by regions: coastal vs. offshore (at least 10 km from the nearest coast), using 'knnsearch' and appropriate filters in MatLab. Data were also organized sub-regionally, (northern: >34.15°N and<−120.5°E, central: from 33.75°N to 34.41°N and −120.4°E to −119.3°E, and southern: < 34.03°N and >−119.3°E) inside the Southern California Bight by visually inspecting the cruise locations on a plot and generating arrays to include all data points collected within the coordinates for each region. All data was filtered for NaNs, and '0' values were converted to '0.0001' to meet the assumption of the log transform.

A GAM fit was implemented in R for each of the seasonal and regional categories. The 'mgcv' package was obtained and the 'gam' function was implemented for vector arrays of seasonal and regional temperature and nitrate data. GAM parameters were defined as 'k = 10', 'family = Tweedie', and 'p = 1.3'. A summary table of parameters was generated, the gam fit was plotted for visual inspection, and checked for statistical robustness. Vectors representing the minimum and maximum temperature range of each season and region, with steps of 0.1°, was generated in MatLab and uploaded to R. Look-Up-Tables were created using 'gam.predict' with the 'gam' output, the min/max temperature range vector, and 'se.fit' set to TRUE.

Citations

[1]CalCOFI.org [2]http://sbc.lternet.edu/ [3]http://www.oceancolor.ucsb.edu/plumes_and_blooms/
[4]Otero, M. P., and Siegel, D. A. (2004). Spatial and temporal characteristics of sediment plumes and phytoplankton blooms in the Santa Barbara Channel. Deep Sea Res. 51, 1129–1149. doi: 10.1016/j.dsr2.2004.04.004

All code written and implemented for this LUT is attached below for instructional purposes only, or by written consent from the authors.

MATLAB scripts

```
%% T2N relationship in the Santa Barbara Channel
%for Frontiers publication to advise kelp aquaculture siting using remotely sensed and publicly available nitrate data
%written by Jordan Snyder at the University of California Santa Barbara, Earth Research Department. Last updated on August 2019.


%% load cleaned and filtered Plumes and Blooms Dataset


PnB = load('/Users/jnsnyder/Documents/Data/PnB/PnB.mat');

for k = 1:3453;
    date(k) = PnB.PnB.data(k,1);
    month(k) = PnB.PnB.data(k,2);
    station(k) = PnB.PnB.data(k,3);
    depth(k) = PnB.PnB.data(k,4);
    temp(k) = PnB.PnB.data(k,5);
    no3(k) = PnB.PnB.data(k,6);
    po4(k) = PnB.PnB.data(k,7);
end

%get rid of NaNs
Date = isnan(date);
date(Date)= [];

Month = isnan(month);
month(Month)= [];

Station = isnan(station);
station(Station)= [];

Depth = isnan(depth);
depth(Depth)= [];

Temp = isnan(temp);
temp(Temp)= [];

No3 = isnan(no3);
no3(No3)= [];

Po4 = isnan(po4);
po4(Po4)= [];

clear Date Month Depth No3 Po4 Station Temp;

%
```

```
%figure(1);
%scatter(temp,no3);
%xlabel('Temperature');
%ylabel('Nitrate');


%% Separate data by season and station
% months with '1' are Jan, '2' are Feb, etc. etc.
JanDateIndex = find(month == 1);
JanTemps = temp(JanDateIndex);
JanNo3s = no3(JanDateIndex);
JanPo4s = po4(JanDateIndex);

FebDateIndex = find(month == 2);
FebTemps = temp(FebDateIndex);
FebNo3s = no3(FebDateIndex);
FebPo4s = po4(FebDateIndex);

MarDateIndex = find(month == 3);
MarTemps = temp(MarDateIndex);
MarNo3s = no3(MarDateIndex);
MarPo4s = po4(MarDateIndex);

AprDateIndex = find(month == 4);
AprTemps = temp(AprDateIndex);
AprNo3s = no3(AprDateIndex);
AprPo4s = po4(AprDateIndex);

MayDateIndex = find(month == 5);
MayTemps = temp(MayDateIndex);
MayNo3s = no3(MayDateIndex);
MayPo4s = po4(MayDateIndex);

JunDateIndex = find(month == 6);
JunTemps = temp(JunDateIndex);
JunNo3s = no3(JunDateIndex);
JunPo4s = po4(JunDateIndex);

JulDateIndex = find(month == 7);
JulTemps = temp(JulDateIndex);
JulNo3s = no3(JulDateIndex);
JulPo4s = po4(JulDateIndex);

AugDateIndex = find(month == 8);
AugTemps = temp(AugDateIndex);
AugNo3s = no3(AugDateIndex);
AugPo4s = po4(AugDateIndex);

SeptDateIndex = find(month == 9);
SeptTemps = temp(SeptDateIndex);
SeptNo3s = no3(SeptDateIndex);
SeptPo4s = po4(SeptDateIndex);

OctDateIndex = find(month == 10);
```

```
OctTemps = temp(OctDateIndex);
OctNo3s = no3(OctDateIndex);
OctPo4s = po4(OctDateIndex);

NovDateIndex = find(month == 11);
NovTemps = temp(NovDateIndex);
NovNo3s = no3(NovDateIndex);
NovPo4s = po4(NovDateIndex);

DecDateIndex = find(month == 12);
DecTemps = temp(DecDateIndex);
DecNo3s = no3(DecDateIndex);
DecPo4s = po4(DecDateIndex);

%make a single array with the months in order so I can plot them all on one
%regression

AllTemps = [JanTemps FebTemps MarTemps AprTemps MayTemps JunTemps JulTemps AugTemps
SeptTemps OctTemps NovTemps DecTemps];
AllNo3s = [JanNo3s FebNo3s MarNo3s AprNo3s MayNo3s JunNo3s JulNo3s AugNo3s SeptNo3s
OctNo3s NovNo3s DecNo3s];
AllPo4s = [JanPo4s FebPo4s MarPo4s AprPo4s MayPo4s JunPo4s JulPo4s AugPo4s SeptPo4s
OctPo4s NovPo4s DecPo4s];

%% load LTER data

LTERdata = load('/Users/jnsnyder/Documents/Data/LTER/LTER_bottle.mat');
load('/Users/jnsnyder/Documents/Data/LTER/LTERdatamonths.mat');

%delete data below 1m depth
toodeepIndex = find(LTERdata.data(:,1) >1);
LTERdata.data(toodeepIndex,:)= NaN;

%convert lter dates from vectors to month/day
%LTERdatamonths = LTERdata.date;
%LTERdates = LTERdata.date;
%DD = datestr(LTERdates);
%DDD = datetime(DD);
%LTERdatamonths = month(DDD);

%DDDD = month(DDD,monthType,'shortname');
%DDD = cellstr(DD);
%DDDD = extractBetween(DDD,4,6);
%LTERdatamonths = DDDD;
%DDDD = char(newStr);

LTERallTemps = LTERdata.data(:,2);
LTERallNo3s = LTERdata.data(:,5);
LTERallPo4s = LTERdata.data(:,4);

%delete nut data where no temp data
indexNoTemp = find(isnan(LTERallTemps));
LTERallNo3s(indexNoTemp)= NaN;
LTERallPo4s(indexNoTemp)= NaN;
```

```
%{
figure(4);
scatter(LTERallTemps,LTERallNo3s,LTERallTemps,LTERallPo4s);

%% Separate data by month
JanDateIndexLTER = find(LTERdatamonths == 1);
JanTempsLTER = LTERallTemps(JanDateIndexLTER);
JanNo3sLTER = LTERallNo3s(JanDateIndexLTER);
JanPo4sLTER = LTERallPo4s(JanDateIndexLTER);

FebDateIndexLTER = find(LTERdatamonths == 2);
FebTempsLTER = LTERallTemps(FebDateIndexLTER);
FebNo3sLTER = LTERallNo3s(FebDateIndexLTER);
FebPo4sLTER = LTERallPo4s(FebDateIndexLTER);

MarDateIndexLTER = find(LTERdatamonths == 3);
MarTempsLTER = LTERallTemps(MarDateIndexLTER);
MarNo3sLTER = LTERallNo3s(MarDateIndexLTER);
MarPo4sLTER = LTERallPo4s(MarDateIndexLTER);

AprDateIndexLTER = find(LTERdatamonths == 4);
AprTempsLTER = LTERallTemps(AprDateIndexLTER);
AprNo3sLTER = LTERallNo3s(AprDateIndexLTER);
AprPo4sLTER = LTERallPo4s(AprDateIndexLTER);

MayDateIndexLTER = find(LTERdatamonths == 5);
MayTempsLTER = LTERallTemps(MayDateIndexLTER);
MayNo3sLTER = LTERallNo3s(MayDateIndexLTER);
MayPo4sLTER = LTERallPo4s(MayDateIndexLTER);

JunDateIndexLTER = find(LTERdatamonths == 6);
JunTempsLTER = LTERallTemps(JunDateIndexLTER);
JunNo3sLTER = LTERallNo3s(JunDateIndexLTER);
JunPo4sLTER = LTERallPo4s(JunDateIndexLTER);

JulDateIndexLTER = find(LTERdatamonths == 7);
JulTempsLTER = LTERallTemps(JulDateIndexLTER);
JulNo3sLTER = LTERallNo3s(JulDateIndexLTER);
JulPo4sLTER = LTERallPo4s(JulDateIndexLTER);

AugDateIndexLTER =find(LTERdatamonths == 8);
AugTempsLTER = LTERallTemps(AugDateIndexLTER);
AugNo3sLTER = LTERallNo3s(AugDateIndexLTER);
AugPo4sLTER = LTERallPo4s(AugDateIndexLTER);

SepDateIndexLTER = find(LTERdatamonths == 9);
SepTempsLTER = LTERallTemps(SepDateIndexLTER);
SepNo3sLTER = LTERallNo3s(SepDateIndexLTER);
SepPo4sLTER = LTERallPo4s(SepDateIndexLTER);

OctDateIndexLTER = find(LTERdatamonths == 10);
OctTempsLTER = LTERallTemps(OctDateIndexLTER);
OctNo3sLTER = LTERallNo3s(OctDateIndexLTER);
OctPo4sLTER = LTERallPo4s(OctDateIndexLTER);
```

```
NovDateIndexLTER = find(LTERdatamonths == 11);
NovTempsLTER = LTERallTemps(NovDateIndexLTER);
NovNo3sLTER = LTERallNo3s(NovDateIndexLTER);
NovPo4sLTER = LTERallPo4s(NovDateIndexLTER);

DecDateIndexLTER = find(LTERdatamonths == 12);
DecTempsLTER = LTERallTemps(DecDateIndexLTER);
DecNo3sLTER = LTERallNo3s(DecDateIndexLTER);
DecPo4sLTER = LTERallPo4s(DecDateIndexLTER);

%%
%{
figure(5);
hold all;
h1 = plot(JanTempsLTER,JanNo3sLTER,'go','MarkerSize',10);%jan
h2 = plot(FebTempsLTER,FebNo3sLTER,'go','MarkerSize',10);%feb
h3 = plot(MarTempsLTER,MarNo3sLTER ,'go','MarkerSize',10);%mar
h4 = plot(AprTempsLTER,AprNo3sLTER,'go','MarkerSize',10);%apr
h5 = plot(MayTempsLTER,MayNo3sLTER,'yo','MarkerSize',10);%may
h6 = plot(JunTempsLTER,JunNo3sLTER,'yo','MarkerSize',10);%jun
h7 = plot(JulTempsLTER,JulNo3sLTER,'yo','MarkerSize',10);%jul
h8 = plot(AugTempsLTER,AugNo3sLTER,'ro','MarkerSize',10);%aug
h9 = plot(SepTempsLTER,SepNo3sLTER,'ro','MarkerSize',10);%sept
h10 = plot(OctTempsLTER,OctNo3sLTER ,'ro','MarkerSize',10);%oct
h11 = plot(NovTempsLTER,NovNo3sLTER,'ro','MarkerSize',10);%nov
h12 = plot(DecTempsLTER,DecNo3sLTER,'yo','MarkerSize',10);%dec
%plot(xL,yL,'k');
xlabel('Temperature [C]');
ylabel('Nitrate [uM]');
title('LTER T2N relationships');
legend('Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec');
f = 'Times New Roman';
set(gca,'Fontname',f,'Fontsize',12);
%(only 98 usable data points from LTER)

%set(gcf,'PaperUnits','inches','PaperPosition',[0 0 8 6]);
%print(gcf, '-dpng', '-r800', '/Users/jordansnyder/Desktop/LTER_t2n_allmonths.png')
%}


%% load CalCOFI data
calcofidata = load('/Users/jnsnyder/Documents/Data/CalCofi/CalCOFI_bottle.mat');
load('/Users/jnsnyder/Documents/Data/CalCofi/CCdatamonths.mat');
%index and delete data at lat and lon coords outside region of southern calif bight
mysubset = find(calcofidata.coords(:,1)<=35.7059 & calcofidata.coords(:,1)>=33.4957 &
calcofidata.coords(:,2) >= -121.186 & calcofidata.coords(:,2) <= -118.5940); % Santa Barbara Channel
calcofidataSubSet = calcofidata.data(mysubset,:);

%index values below 3 m depth
surfaceSubset = find(calcofidataSubSet(:,1) <= 3); %for subset within LS8
calcofidataSubSet2 = calcofidataSubSet(surfaceSubset,:);
%surfaceSubset = find(calcofidata.data(:,1) <= 3); %for entire westcoast
%calcofidataSubSet2 = calcofidata.data(surfaceSubset,:);
```

```
%make date array same size as data array
%date1 = calcofidata.date(mysubset);
%date2 = date1(surfaceSubset);
date2 = calcofidata.date(surfaceSubset);

%remove lines with no data
%remove temperature lines with no data
indSSTzero = find(calcofidataSubSet2(:,2) == -999);
calcofidataSubSet2(indSSTzero,:) = NaN;
%remove NO3 lines with no data
indNO3zero = find(calcofidataSubSet2(:,8) == -999);
calcofidataSubSet2(indNO3zero,:) = NaN;
%remove NO2 lines with no data
indNO2zero = find(calcofidataSubSet2(:,7) == -999);
calcofidataSubSet2(indNO2zero,7) = NaN;
%remove PO4 lines with no data
indPO4zero = find(calcofidataSubSet2(:,6) == -999);
calcofidataSubSet2(indPO4zero,6) = NaN;

%make separate arrays of temp and nut data outside of structure
CCallTemps = calcofidataSubSet2(:,2);
CCallDepths = calcofidataSubSet2(:,1);
CCallNO3 = calcofidataSubSet2(:,8);
CCallNO2 = calcofidataSubSet2(:,7);
CCallPO4 = calcofidataSubSet2(:,6);

CCallNits = CCallNO3;

%find % of no3 in no3+no2 for reference
%no2 has higher precision than no3, delete no2s smaller than 0.1
ccno2small = find(CCallNO2 <=0.1);
CCallNO2(ccno2small) = 0;
CCC = CCallNO3 + CCallNO2;
percentNO3 = CCallNO3./(CCC);
avePerNO3 = nanmean(percentNO3); % is 83%
stdPerNO3 = nanstd(percentNO3);

figure
scatter(1:4034,percentNO3);


Atemp = percentNO3;
Atemp(Atemp == 0) = NaN;
nanmean(Atemp)
nanstd(Atemp)

%% separate data into months
%convert calcofi dates from vectors to month/day

[~,monnum,~,~,~,~]=datevec(date2);
% CCDD = datestr(date2);
% CCDDD = cellstr(CCDD);
% CCCDDD = datetime(CCDDD)';
```

```
CCdatamonths = monnum;

JanDateIndexCC = find(CCdatamonths == 1);
JanTempsCC = CCallTemps(JanDateIndexCC);
JanNitsCC = CCallNits(JanDateIndexCC);
JanPO4sCC = CCallPO4(JanDateIndexCC);

FebDateIndexCC = find(CCdatamonths == 2);
FebTempsCC = CCallTemps(FebDateIndexCC);
FebNitsCC = CCallNits(FebDateIndexCC);
FebPO4sCC = CCallPO4(FebDateIndexCC);

MarDateIndexCC = find(CCdatamonths == 3);
MarTempsCC = CCallTemps(MarDateIndexCC);
MarNitsCC = CCallNits(MarDateIndexCC);
MarPO4sCC = CCallPO4(MarDateIndexCC);

AprDateIndexCC = find(CCdatamonths == 4);
AprTempsCC = CCallTemps(AprDateIndexCC);
AprNitsCC = CCallNits(AprDateIndexCC);
AprPO4sCC = CCallPO4(AprDateIndexCC);

MayDateIndexCC = find(CCdatamonths == 5);
MayTempsCC = CCallTemps(MayDateIndexCC);
MayNitsCC = CCallNits(MayDateIndexCC);
MayPO4sCC = CCallPO4(MayDateIndexCC);

JunDateIndexCC = find(CCdatamonths == 6);
JunTempsCC = CCallTemps(JunDateIndexCC);
JunNitsCC = CCallNits(JunDateIndexCC);
JunPO4sCC = CCallPO4(JunDateIndexCC);

JulDateIndexCC = find(CCdatamonths == 7);
JulTempsCC = CCallTemps(JulDateIndexCC);
JulNitsCC = CCallNits(JulDateIndexCC);
JulPO4sCC = CCallPO4(JulDateIndexCC);

AugDateIndexCC = find(CCdatamonths == 8);
AugTempsCC = CCallTemps(AugDateIndexCC);
AugNitsCC = CCallNits(AugDateIndexCC);
AugPO4sCC = CCallPO4(AugDateIndexCC);

SepDateIndexCC = find(CCdatamonths == 9);
SepTempsCC = CCallTemps(SepDateIndexCC);
SepNitsCC = CCallNits(SepDateIndexCC);
SepPO4sCC = CCallPO4(SepDateIndexCC);

OctDateIndexCC = find(CCdatamonths == 10);
OctTempsCC = CCallTemps(OctDateIndexCC);
OctNitsCC = CCallNits(OctDateIndexCC);
OctPO4sCC = CCallPO4(OctDateIndexCC);

NovDateIndexCC = find(CCdatamonths == 11);
NovTempsCC = CCallTemps(NovDateIndexCC);
```

```
NovNitsCC = CCallNits(NovDateIndexCC);
NovPO4sCC = CCallPO4(NovDateIndexCC);

DecDateIndexCC = find(CCdatamonths == 12);
DecTempsCC = CCallTemps(DecDateIndexCC);
DecNitsCC = CCallNits(DecDateIndexCC);
DecPO4sCC = CCallPO4(DecDateIndexCC);

%make a single array with the months in order so I can plot them all on one
%regression

AllTempsCC = vertcat(JanTempsCC, FebTempsCC, MarTempsCC, AprTempsCC, MayTempsCC,
JunTempsCC, JulTempsCC, AugTempsCC, SepTempsCC, OctTempsCC, NovTempsCC,
DecTempsCC);
AllNitsCC = vertcat(JanNitsCC, FebNitsCC, MarNitsCC, AprNitsCC, MayNitsCC, JunNitsCC, JulNitsCC,
AugNitsCC, SepNitsCC, OctNitsCC, NovNitsCC, DecNitsCC);
AllPO4sCC = vertcat(JanPO4sCC, FebPO4sCC, MarPO4sCC, AprPO4sCC, MayPO4sCC, JunPO4sCC,
JulPO4sCC, AugPO4sCC, SepPO4sCC, OctPO4sCC, NovPO4sCC, DecPO4sCC);



%you may need to adjust these
JANTempsCC = isnan(JanTempsCC);
JanTempsCC(JANTempsCC)= [];
JANNitsCC = isnan(JanNitsCC);
JanNitsCC(JANNitsCC)= [];
JANPO4sCC = isnan(JanPO4sCC);
JanPO4sCC(JANPO4sCC)= [];
%JanTempsCC= JanTempsCC(isfinite(JanPO4sCC));%make temp and po4s same size as nits
%JanNitsCC= JanNitsCC(isfinite(JanPO4sCC));%make temp and po4s same size as nits

FEBTempsCC = isnan(FebTempsCC);
FebTempsCC(FEBTempsCC)= [];
FEBNitsCC = isnan(FebNitsCC);
FebNitsCC(FEBNitsCC)= [];
FEBPO4sCC = isnan(FebPO4sCC);
FebPO4sCC(FEBPO4sCC)= [];
%FebTempsCC= FebTempsCC(isfinite(FebNitsCC));%already same size
%FebPO4sCC= FebPO4sCC(isfinite(FebNitsCC));%already same size

MARTempsCC = isnan(MarTempsCC);
MarTempsCC(MARTempsCC)= [];
MARNitsCC = isnan(MarNitsCC);
MarNitsCC(MARNitsCC)= [];
%MarTempsCC= MarTempsCC(isfinite(MarNitsCC));%make temp same size as nits
MARPO4sCC = isnan(MarPO4sCC);
MarPO4sCC(MARPO4sCC)= [];

APRTempsCC = isnan(AprTempsCC);
AprTempsCC(APRTempsCC)= [];
APRNitsCC = isnan(AprNitsCC);
AprNitsCC(APRNitsCC)= [];
APRPO4sCC = isnan(AprPO4sCC);
AprPO4sCC(APRPO4sCC)= [];
%AprTempsCC= AprTempsCC(isfinite(AprPO4sCC));%make temp same size as nits
```

```
%AprNitsCC= AprNitsCC(isfinite(AprPO4sCC));%make temp same size as nits


MAYTempsCC = isnan(MayTempsCC);
MayTempsCC(MAYTempsCC)= [];
MAYNitsCC = isnan(MayNitsCC);
MayNitsCC(MAYNitsCC)= [];
MAYPO4sCC = isnan(MayPO4sCC);
MayPO4sCC(MAYPO4sCC)= [];
%MayTempsCC= MayTempsCC(isfinite(MayPO4sCC));%make temp same size as nits
%MayNitsCC= MayNitsCC(isfinite(MayPO4sCC));%make temp same size as nits

JUNTempsCC = isnan(JunTempsCC);
JunTempsCC(JUNTempsCC)= [];
JUNNitsCC = isnan(JunNitsCC);
JunNitsCC(JUNNitsCC)= [];
%JunTempsCC= JunTempsCC(isfinite(JunNitsCC));%make temp same size as nits
JUNPO4sCC = isnan(JunPO4sCC);
JunPO4sCC(JUNPO4sCC)= [];

JULTempsCC = isnan(JulTempsCC);
JulTempsCC(JULTempsCC)= [];
JULNitsCC = isnan(JulNitsCC);
JulNitsCC(JULNitsCC)= [];
JULPO4sCC = isnan(JulPO4sCC);
JulPO4sCC(JULPO4sCC)= [];
%JulTempsCC= JulTempsCC(isfinite(JulPO4sCC));%make temp same size as nits
%JulNitsCC= JulNitsCC(isfinite(JulPO4sCC));%make temp same size as nits

AUGTempsCC = isnan(AugTempsCC);
AugTempsCC(AUGTempsCC)= [];
AUGNitsCC = isnan(AugNitsCC);
AugNitsCC(AUGNitsCC)= [];
%AugTempsCC= AugTempsCC(isfinite(AugNitsCC));%make temp same size as nits
AUGPO4sCC = isnan(AugPO4sCC);
AugPO4sCC(AUGPO4sCC)= [];

SEPTempsCC = isnan(SepTempsCC);
SepTempsCC(SEPTempsCC)= [];
SEPNitsCC = isnan(SepNitsCC);
SepNitsCC(SEPNitsCC)= [];
%SepTempsCC= SepTempsCC(isfinite(SepNitsCC));%make temp same size as nits
SEPPO4sCC = isnan(SepPO4sCC);
SepPO4sCC(SEPPO4sCC)= [];

OCTTempsCC = isnan(OctTempsCC);
OctTempsCC(OCTTempsCC)= [];
OCTNitsCC = isnan(OctNitsCC);
OctNitsCC(OCTNitsCC)= [];
OCTPO4sCC = isnan(OctPO4sCC);
OctPO4sCC(OCTPO4sCC)= [];
%OctTempsCC= OctTempsCC(isfinite(OctPO4sCC));%make temp same size as nits
%OctNitsCC= OctNitsCC(isfinite(OctPO4sCC));%make temp same size as nits
```

```
NOVTempsCC = isnan(NovTempsCC);
NovTempsCC(NOVTempsCC)= [];
NOVNitsCC = isnan(NovNitsCC);
NovNitsCC(NOVNitsCC)= [];
%NovTempsCC= NovTempsCC(isfinite(NovNitsCC));%make temp same size as nits
NOVPO4sCC = isnan(NovPO4sCC);
NovPO4sCC(NOVPO4sCC)= [];

DECTempsCC = isnan(DecTempsCC);
DecTempsCC(DECTempsCC)= [];
DECNitsCC = isnan(DecNitsCC);
DecNitsCC(DECNitsCC)= [];
%DecTempsCC= DecTempsCC(isfinite(DecNitsCC));%make temp same size as nits
DECPO4sCC = isnan(DecPO4sCC);
DecPO4sCC(DECPO4sCC)= [];


%% plot all coordinates on spatial map for use
%load coastline files
[coastlat, coastlon] = load_ca_coast_2mill;

%scatter(LTERdata.coords(:,2),LTERdata.coords(:,1))
LTERdata = load('/Users/jnsnyder/Documents/Data/LTER/LTER_bottle.mat');

%plot only stations used for SBC landsat image region
sbcStnslats = calcofidata.coords(mysubset,1);
sbcStnslons = calcofidata.coords(mysubset,2);

pnblats = [34.39017, 34.3435, 34.29683 34.25017 34.2035 34.15683 34.08333];
pnblons = [-119.8407 -119.8627 -119.8845 -119.9063 -119.9283 -119.9502 -120.0333];


figure(7);
hold on;
c1 = scatter(calcofidata.coords(:,2),calcofidata.coords(:,1),'filled');%plot all calcofi stations
hold on
%c2=scatter(sbcStnslons,sbcStnslats);
hold on
%enter lat lon of lter sites
c3=scatter(LTERdata.coords(:,2),LTERdata.coords(:,1),'filled');
hold on
%enter lat lon of pnb sites
c4=scatter(pnblons,pnblats,'filled');
box on;
xlabel('Longitude ^oE');
ylabel('Latitude ^oN');
legend([c1  c3 c4],{'All CalCOFI Stations','LTER Stations','Plumes and Blooms Stations'});
f = 'Times New Roman';
set(gca,'Fontsize',12);%'Fontname',f);
hold on;
cplot = plot(coastlon,coastlat,'k','LineWidth',0.1);%plot coastline
xlim([-121.5, -117]);
ylim([32, 36.5]);
```

```
%set(gcf,'PaperUnits','inches','PaperPosition',[0 0 5.2 6]);
%print(gcf, '-dpng', '-r800', '/Users/jnsnyder/Desktop/All_stations_map3.png')



%% Combine all three data sets for T2N relationships

JanTempsALL = vertcat(JanTemps', JanTempsLTER, JanTempsCC);
FebTempsALL = vertcat(FebTemps', FebTempsLTER, FebTempsCC);
MarTempsALL = vertcat(MarTemps', MarTempsLTER, MarTempsCC);
AprTempsALL = vertcat(AprTemps', AprTempsLTER, AprTempsCC);
MayTempsALL = vertcat(MayTemps', MayTempsLTER, MayTempsCC);
JunTempsALL = vertcat(JunTemps', JunTempsLTER, JunTempsCC);
JulTempsALL = vertcat(JulTemps', JulTempsLTER, JulTempsCC);
AugTempsALL = vertcat(AugTemps', AugTempsLTER, AugTempsCC);
SepTempsALL = vertcat(SeptTemps', SepTempsLTER, SepTempsCC);
OctTempsALL = vertcat(OctTemps', OctTempsLTER, OctTempsCC);
NovTempsALL = vertcat(NovTemps', NovTempsLTER, NovTempsCC);
DecTempsALL = vertcat(DecTemps', DecTempsLTER, DecTempsCC);

WinterTempsALL = vertcat(JanTempsALL, FebTempsALL, MarTempsALL, AprTempsALL,
MayTempsALL, DecTempsALL);
SummerTempsALL = vertcat(JunTempsALL, JulTempsALL, AugTempsALL, SepTempsALL,
OctTempsALL,NovTempsALL);

JanNutsALL = vertcat(JanNo3s', JanNo3sLTER, JanNitsCC);
FebNutsALL = vertcat(FebNo3s', FebNo3sLTER, FebNitsCC);
MarNutsALL = vertcat(MarNo3s', MarNo3sLTER, MarNitsCC);
AprNutsALL = vertcat(AprNo3s', AprNo3sLTER, AprNitsCC);
MayNutsALL = vertcat(MayNo3s', MayNo3sLTER, MayNitsCC);
JunNutsALL = vertcat(JunNo3s', JunNo3sLTER, JunNitsCC);
JulNutsALL = vertcat(JulNo3s', JulNo3sLTER, JulNitsCC);
AugNutsALL = vertcat(AugNo3s', AugNo3sLTER, AugNitsCC);
SepNutsALL = vertcat(SeptNo3s', SepNo3sLTER, SepNitsCC);
OctNutsALL = vertcat(OctNo3s', OctNo3sLTER, OctNitsCC);
NovNutsALL = vertcat(NovNo3s', NovNo3sLTER, NovNitsCC);
DecNutsALL = vertcat(DecNo3s', DecNo3sLTER, DecNitsCC);

WinterNutsALL = vertcat(JanNutsALL, FebNutsALL, MarNutsALL, AprNutsALL, MayNutsALL,
DecNutsALL);
SummerNutsALL = vertcat(JunNutsALL, JulNutsALL, AugNutsALL, SepNutsALL, OctNutsALL,
NovNutsALL);

JanPo4sALL = vertcat(JanPo4s', JanPo4sLTER, JanPO4sCC);
FebPo4sALL = vertcat(FebPo4s', FebPo4sLTER, FebPO4sCC);
MarPo4sALL = vertcat(MarPo4s', MarPo4sLTER, MarPO4sCC);
AprPo4sALL = vertcat(AprPo4s', AprPo4sLTER, AprPO4sCC);
MayPo4sALL = vertcat(MayPo4s', MayPo4sLTER, MayPO4sCC);
JunPo4sALL = vertcat(JunPo4s', JunPo4sLTER, JunPO4sCC);
JulPo4sALL = vertcat(JulPo4s', JulPo4sLTER, JulPO4sCC);
AugPo4sALL = vertcat(AugPo4s', AugPo4sLTER, AugPO4sCC);
SepPo4sALL = vertcat(SeptPo4s', SepPo4sLTER, SepPO4sCC);
OctPo4sALL = vertcat(OctPo4s', OctPo4sLTER, OctPO4sCC);
NovPo4sALL = vertcat(NovPo4s', NovPo4sLTER, NovPO4sCC);
```

```
DecPo4sALL = vertcat(DecPo4s', DecPo4sLTER, DecPO4sCC);

WinterPo4sALL = vertcat(JanPo4sALL, FebPo4sALL, MarPo4sALL, AprPo4sALL, MayPo4sALL,
DecPo4sALL);
SummerPo4sALL = vertcat(JunPo4sALL, JulPo4sALL, AugPo4sALL, SepPo4sALL, OctPo4sALL,
NovPo4sALL);


%% get rid of any excess nans and zeros

WINTERTEMPSALL = isnan(WinterTempsALL);
WinterTempsALL(WINTERTEMPSALL)= [];

SUMMERTEMPSALL = isnan(SummerTempsALL);
SummerTempsALL(SUMMERTEMPSALL)= [];

WINTERNUTSALL = isnan(WinterNutsALL);
WinterNutsALL(WINTERNUTSALL)= [];

SUMMERNUTSALL = isnan(SummerNutsALL);
SummerNutsALL(SUMMERNUTSALL)= [];

WINTERPO4SALL = isnan(WinterPo4sALL);
WinterPo4sALL(WINTERPO4SALL)= [];

SUMMERPO4SALL = isnan(SummerPo4sALL);
SummerPo4sALL(SUMMERPO4SALL)= [];

%turn NO3 0 values into 0.0001
indNUTSzero = find(WinterNutsALL(:,1) <= 0);
WinterNutsALL(indNUTSzero,:) = 0.0001;

indSNUTSzero = find(SummerNutsALL(:,1) <= 0);
SummerNutsALL(indSNUTSzero,:) = 0.0001;

%% plot relationship between no3 and po4

totalNO3 = vertcat(WinterNutsALL, SummerNutsALL);
totalPO4 = vertcat(WinterPo4sALL, SummerPo4sALL);

figure
scatter(totalNO3, totalPO4);
ylim([0 3]);
xlim([0 30]);
xlabel('NO3');
ylabel('PO4');
set(gcf,'PaperUnits','inches','PaperPosition',[0 0 8 6]);
print(gcf, '-dpng', '-r800', '/Users/jnsnyder/Desktop/no3po4.png')

%% Plot T2N for all data in seasons

figure(8);
hold all
```

```
h1 = scatter(WinterTempsALL,WinterNutsALL,'filled','SizeData',10);
h3 = scatter(SummerTempsALL,SummerNutsALL,'filled','SizeData',10);
box on;
ylim([0 30]);
xlabel('Temperature (^oC)');
ylabel('Nitrate conc. (\mumol L^-^1');
legend('Winter','Summer');
%title('T2N relationships');
%f = 'Times New Roman';
set(gca,'Fontsize',18);%,'Fontname',f);
%set(gcf,'PaperUnits','inches','PaperPosition',[0 0 8 6]);
%print(gcf, '-dpng', '-r800', '/Users/jnsnyder/Desktop/t2nALLCruises.png')


%% Export .mat files to csv for GAM fit in R

NutsTempsWinter2 = [WinterNutsALL(:,1) WinterTempsALL(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsWinters2.csv',NutsTempsWinter2,{'Nuts','Te
mp'});

NutsTempsSummer2 = [SummerNutsALL(:,1) SummerTempsALL(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsSummers2.csv',NutsTempsSummer2,{'Nuts',
'Temp'});


%%
%SST_vec1Win = 9.1:0.1:19.7;%winter
SST_vec1Win = 9.1:0.1:19.6;%winter whole coast
SST_vec1Win = SST_vec1Win';
csvwrite_with_headers('/Users/jnsnyder/Desktop/SST_vecWin.csv',SST_vec1Win,{'Temp'});

SST_vec1Sum = 9.1:0.1:22.9;%summer
%SST_vec1Sum = 9.1:0.1:24.1;%summer 2 - whole coast
SST_vec1Sum = SST_vec1Sum';
csvwrite_with_headers('/Users/jnsnyder/Desktop/SST_vecSum.csv',SST_vec1Sum,{'Temp'});



%% knn search for offshore vs inshore t2n gam WITHIN SBC LANDSAT SCENE
Uni = unique([sbcStnslats sbcStnslons], 'rows');%only use this if plotting
sbcStnslats2 = sbcStnslats(surfaceSubset);
sbcStnslons2 = sbcStnslons(surfaceSubset);

%[IDX,D] = knnsearch([coastlat coastlon], Uni);
figure
plot(coastlon, coastlat, 'k')
hold on
scatter(Uni(:,2), Uni(:,1),100,D,'.');

[IDX2,D2] = knnsearch([coastlat coastlon], [sbcStnslats2 sbcStnslons2]);

D2 = deg2km(D2);
```

```
%find where distance from shore is greater than 10 km
idx10km = find(D2 >= 10);
%make new calcofi data structure with offshore >10km
calcofidataSubSet10km = calcofidataSubSet2(idx10km,:);
CCNO310km = calcofidataSubSet10km(:,8);
CCTemps10km =calcofidataSubSet10km(:,2);
CCPO410km = calcofidataSubSet10km(:,6);

%find where distance from shore is less than 10km
idx9km = find(D2 <10);
calcofidataSubSet9km = calcofidataSubSet2(idx9km,:);
CCNO39km = calcofidataSubSet9km(:,8);
CCTemps9km =calcofidataSubSet9km(:,2);
CCPO49km = calcofidataSubSet9km(:,6);


%get rid of nans and 0s for gam
ccno39km = isnan(CCNO39km);
CCNO39km(ccno39km)= [];
ind9kmzero = find(CCNO39km(:,1) <= 0);%turn NO3 0 values into 0.0001
CCNO39km(ind9kmzero,:) = 0.0001;

ccno310km = isnan(CCNO310km);
CCNO310km(ccno310km)= [];
ind10kmzero = find(CCNO310km(:,1) <= 0);%turn NO3 0 values into 0.0001
CCNO310km(ind10kmzero,:) = 0.0001;

cctemps9km = isnan(CCTemps9km);
CCTemps9km(cctemps9km)= [];

cctemps10km = isnan(CCTemps10km);
CCTemps10km(cctemps10km)= [];

NutsTempsInshore = [CCNO39km(:,1) CCTemps9km(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsInshore.csv',NutsTempsInshore,{'Nuts','Temp'});

NutsTempsOffshore = [CCNO310km(:,1) CCTemps10km(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsOffshore.csv',NutsTempsOffshore,{'Nuts','Temp'});

SST_vec1In = 9.4:0.1:21.3;%inshore
SST_vec1In = SST_vec1In';

SST_vec1Off = 9.9:0.1:22.9;%offshore
SST_vec1Off = SST_vec1Off';
%csvwrite_with_headers('/Users/jnsnyder/Desktop/SST_vec.csv',SST_vec1In,{'Temp'});
%csvwrite_with_headers('/Users/jnsnyder/Desktop/SST_vec.csv',SST_vec1Off,{'Temp'});

SST_pred_outsIn = [];
SST_pred_outsOff = [];
SST_pred_outsIn = exp(SST_pred_outsIn);
SST_pred_outsOff = exp(SST_pred_outsOff);
```

```
sefitIn = [];
sefitOff = [];
sefitIn = exp(sefitIn);
sefitOff = exp(sefitOff);



figure(4)
subplot(1,2,1);
scatter(CCTemps9km, CCNO39km,'.');
hold on
scatter(CCTemps10km, CCNO310km,'.');
legend('Inshore','Offshore');
box on;
%xlabel('Temperature (^oC)');
ylabel('Nitrate conc. (\mumol L^-^1)');
set(gca,'Fontsize',18);

subplot(1,2,2);
hold all;
se1=shadedErrorBar(SST_vec1In, SST_pred_outsIn,sefitIn);
se2=shadedErrorBar(SST_vec1Off, SST_pred_outsOff,sefitOff);
box on;
hline=refline(0,0);
set(hline,'Color',[0 0 0],'Linewidth',1,'LineStyle','--');
ylabel('Estimated Nitrate conc. (\mumol L^-^1)');
hY=text(27,-2,'Temperature (^oC)','rotation',0,'fontsize',11, ...
        'horizontalalignment','center','verticalalignment','top','FontSize',18);
legend('Inshore','Offshore');
%title('T2N relationships');
set(gca,'Fontsize',18);
%set(gcf,'PaperUnits','inches','PaperPosition',[0 0 15 6]);
%print(gcf, '-dpng', '-r800', '/Users/jnsnyder/Desktop/t2nInvOff.png')



%% region 1 2 3 for t2n game WITHIN LANDSAT SCENE

region1a = ([Uni(7,1) Uni(12,1) Uni(14,1) Uni(15,1) Uni(16,1) Uni(17,1)])';
region1b = ([Uni(7,2) Uni(12,2) Uni(14,2) Uni(15,2) Uni(16,2) Uni(17,2)])';
region1 = [region1a  region1b];

region2a = ([Uni(13,1) Uni(10,1) Uni(11,1) Uni(9,1) Uni(8,1) Uni(5,1) Uni(3,1) Uni(1,1)]');
region2b= ([Uni(13,2) Uni(10,2) Uni(11,2) Uni(9,2) Uni(8,2) Uni(5,2) Uni(3,2) Uni(1,2)]');
region2= [region2a region2b];

region3a = ([Uni(6,1) Uni(4,1) Uni(2,1)]');
region3b = ([Uni(6,2) Uni(4,2) Uni(2,2)]');
region3 = [region3a  region3b];

sbcStns2 = [sbcStnslats2 sbcStnslons2];

AD1 = ismember(sbcStns2, region1, 'rows');
```

```
AD2 = ismember(sbcStns2, region2, 'rows');
AD3 = ismember(sbcStns2, region3, 'rows');

calcofidataSubSetR1 = calcofidataSubSet2(AD1,:);
calcofidataSubSetR2 = calcofidataSubSet2(AD2,:);
calcofidataSubSetR3 = calcofidataSubSet2(AD3,:);

%region1
CCNO3R1 = calcofidataSubSetR1(:,8);
CCTempsR1 =calcofidataSubSetR1(:,2);

ccno3R1 = isnan(CCNO3R1);
CCNO3R1(ccno3R1)= [];
indR1zero = find(CCNO3R1(:,1) <= 0);%turn NO3 0 values into 0.0001
CCNO3R1(indR1zero,:) = 0.0001;
cctempsR1 = isnan(CCTempsR1);
CCTempsR1(cctempsR1)= [];

%region2
CCNO3R2 = calcofidataSubSetR2(:,8);
CCTempsR2 =calcofidataSubSetR2(:,2);

ccno3R2 = isnan(CCNO3R2);
CCNO3R2(ccno3R2)= [];
indR2zero = find(CCNO3R2(:,1) <= 0);%turn NO3 0 values into 0.0001
CCNO3R2(indR2zero,:) = 0.0001;
cctempsR2 = isnan(CCTempsR2);
CCTempsR2(cctempsR2)= [];

%region3
CCNO3R3 = calcofidataSubSetR3(:,8);
CCTempsR3 =calcofidataSubSetR3(:,2);

ccno3R3 = isnan(CCNO3R3);
CCNO3R3(ccno3R3)= [];
indR3zero = find(CCNO3R3(:,1) <= 0);%turn NO3 0 values into 0.0001
CCNO3R3(indR3zero,:) = 0.0001;
cctempsR3 = isnan(CCTempsR3);
CCTempsR3(cctempsR3)= [];

NutsTempsR1 = [CCNO3R1(:,1) CCTempsR1(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsR1.csv',NutsTempsR1,{'Nuts','Temp'});

NutsTempsR2 = [CCNO3R2(:,1) CCTempsR2(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsR2.csv',NutsTempsR2,{'Nuts','Temp'});

NutsTempsR3 = [CCNO3R3(:,1) CCTempsR3(:,1)];
csvwrite_with_headers('/Users/jnsnyder/Desktop/NutsTempsR3.csv',NutsTempsR3,{'Nuts','Temp'});

%**********need to change this to have lowest and highest values in observed data for
%each set vvvvvvvv and then redo all
SST_vec1R1 = 9.3:0.1:20.1;%R1
SST_vec1R2 = 10.5:0.1:22.3;%R2
SST_vec1R3 = 12.1:0.1:22.9;%R3
```

```
SST_vec1R1 = SST_vec1R1';
SST_vec1R2 = SST_vec1R2';
SST_vec1R3 = SST_vec1R3';

csvwrite_with_headers('/Users/jnsnyder/Desktop/SST_vec.csv',SST_vec1,{'Temp'});

SST_pred_outsR1 = [];
SST_pred_outsR2 = [];
SST_pred_outsR3 = [];

SST_pred_outsR1 = exp(SST_pred_outsR1);
SST_pred_outsR2 = exp(SST_pred_outsR2);
SST_pred_outsR3 = exp(SST_pred_outsR3);

sefitR1 = [];
sefitR2 = [];
sefitR3 = [];

sefitR1 = exp(sefitR1);
sefitR2 = exp(sefitR2);
sefitR3 = exp(sefitR3);

figure(5);
subplot(1,2,1);
scatter(CCTempsR1, CCNO3R1,'.');
hold all
scatter(CCTempsR2, CCNO3R2,'.');
scatter(CCTempsR3, CCNO3R3,'.');
legend('Region 1','Region 2','Region 3');
box on;
ylabel('Nitrate conc. (\mumol L^-^1)');
set(gca,'Fontsize',18);

subplot(1,2,2);
hold all;
se1=shadedErrorBar(SST_vec1R1, SST_pred_outsR1,sefitR1);
se2=shadedErrorBar(SST_vec1R2, SST_pred_outsR2,sefitR2);
se3=shadedErrorBar(SST_vec1R3, SST_pred_outsR3,sefitR3);
box on;
hline=refline(0,0);
set(hline,'Color',[0 0 0],'Linewidth',1,'LineStyle','--');
ylabel('Estimated Nitrate conc. (\mumol L^-^1)');
hY=text(27,-2,'Temperature (^oC)','rotation',0,'fontsize',11, ...
        'horizontalalignment','center','verticalalignment','top','FontSize',18);
legend('Region 1','Region 2','Region 3');
%title('T2N relationships');
set(gca,'Fontsize',18);

%set(gcf,'PaperUnits','inches','PaperPosition',[0 0 15 6]);
%print(gcf, '-dpng', '-r800', '/Users/jnsnyder/Desktop/t2nr123.png')
```

Code From GAM_t2r.R
Written by Dr. Tom Bell, 2019

```
require(mgcv)

NutsTempsWinters2 = read.csv("~/Desktop/NutsTempsWinters2.csv")
#NutsTempsSummers2 = read.csv("~/Desktop/NutsTempsSummers2.csv")

#NutsTempsInshore = read.csv("~/Desktop/NutsTempsInshore.csv")
#NutsTempsOffshore = read.csv("~/Desktop/NutsTempsOffshore.csv")

#NutsTempsR1 = read.csv("~/Desktop/NutsTempsR1.csv")
#NutsTempsR2 = read.csv("~/Desktop/NutsTempsR2.csv")
#NutsTempsR3 = read.csv("~/Desktop/NutsTempsR3.csv")


b <- gam(Nuts ~ s(Temp, k = 10), family=Tweedie(p = 1.3), data = NutsTempsWinters2)
#b <- gam(Nuts ~ s(Temp, k = 10), family=Tweedie(p = 1.3), data = NutsTempsSummers2)
#b <- gam(Nuts ~ s(Temp, k = 10), family=Tweedie(p = 1.3), data = NutsTempsOffshore)

summary(b)
plot(b)
gam.check(b)

#R> library(R.matlab)
#R> data <- readMat('~/Desktop/sst_02218.mat')

SST_pred <- read.csv("~/Desktop/SST_vecWin.csv")
outs = predict.gam(b,SST_pred, se.fit=TRUE)
write.csv(outs, "~/Desktop/SST_vec_outsWin.csv")
```