**Table of Contents**

[Read Argo raw files](#MW_H_B32EFA2A)  
[Interpolate Z-direction (narrow depth levels)](#MW_H_0C69D15E)  
[Argo Temperature and Salinity values](#MW_H_18F26A0D)  
[Argo profile Classification](#MW_H_70F67B01)  
 [Plot Study region locations](#MW_H_23D4D5D6)  
[Regional Argo Selection Bothian Sea](#MW_H_90497490)  
 [Plot Geolocation: ARGO floats within Baltic region](#MW_H_15151A14)  
 [1) ALL Argo floats](#MW_H_3DF998AC)  
 [2) Argo floats in Winter](#MW_H_C43930FD)  
  [2) Argo floats in Spring](#MW_H_5DAED1EE)  
 [3) Argo floats in Summer](#MW_H_557EF9FC)  
 [4) Argo floats in Autumn](#MW_H_A875DDC6)  
[Argo Averaged profiles (T and S) in each region](#MW_H_4330378F)  
 [[Argo] Plot average Seasonal (winter/summer) profiles in BOS region](#MW_H_28D8E40B)  
[[Mooring] Read mooring data (location: 50N 145W)](#MW_H_4D66024E)  
 [Satellite and Argo comparison againts PAPA mooring (BOXPLOTS)](#MW_H_B3039FB2)  
 [[Mooring+ARGO+SAT] Compare Salinity at 10 m depth from Mooring and ARGO Vs Satellite data](#MW_H_BEF31689)  
 [SSS-BEC (sss\_bec) at mooring location](#MW_H_3184DF24)

% Syntax: Arctic\_argo\_RAW\_plot.m

%

% Description

%

% Argo floats within Arctic region. Use Argo as seen in the original

% data base as given by IFREMER-CORIOLIS site

% The results from this script are to contrast the database

% provided by BEC, seen in folder name "SSS>Arctic>BEC>indata")

%

% This version uses the Argo matlab files creted for each basin.

%

% rcatany (2019/10/10)

%

% =========================================================================

clear; clc; close all

re\_start = 0; % [1] delete fn\_out and start from scratch, or [0] not.

w = warning ('off','all'); % mute warnings

rmpath('folderthatisnotonpath') % check warning are on/off

warning(w)

path\_root = '/Volumes/Rogue/';

% path\_root = '/Volumes/NOC/data/';

folder\_tmp = ['~/Documents/GitHub/tmp/'];

foldercheck\_raf(folder\_tmp);

folder\_out = folder\_tmp; % keep folder out as tmp until new order is given

% inputs script

iyear = 2012;

imonth = [1:12];

% iday = 1;

% number of days in the last month of the series

num\_days = calendar(iyear(end),imonth(end))';

num\_days(num\_days == 0) = [];

num\_days = length(num\_days);

itime\_start = datenum(iyear(1),imonth(1),1,0,0,0);

itime\_end = datenum(iyear(end),imonth(end),num\_days,0,0,0);

ibasin = 9; % Basin number: 9 (Baltic)

[xmin,xmax,ymin,ymax, basin\_str] = map\_lim\_raf(ibasin);

nyear = length(iyear);

% Plot TS Profiles Limits axes

Tmin = 0;

Tmax = 15;

if ibasin == 7

Smin = 15;

Smax = 40;

elseif ibasin == 9

Smin = 0;

Smax = 10;

end

# Read Argo raw files

re\_start = 1; % re-start reading Argo raw file from scratch [1], or not [0]

A = ar\_rd\_RAW(iyear,imonth,re\_start);

Nobs = A.Nobs;

PRES\_ADJUSTED\_ERROR = A.PRES\_ADJUSTED\_ERROR;

SALT\_ADJUSTED\_ERROR = A.SALT\_ADJUSTED\_ERROR;

TEMP\_ADJUSTED\_ERROR = A.TEMP\_ADJUSTED\_ERROR;

temp\_qc = A.temp\_qc;

salt\_qc = A.salt\_qc;

pres\_qc = A.pres\_qc;

lon = A.lon;

lat = A.lat;

time\_ar = A.time\_ar;

platform = A.platform;

temp = A.temp;

salt = A.salt;

pres = A.pres;

# Interpolate Z-direction (narrow depth levels)

Interpolation specs: Grid dimensions

pres1 = 0; % min pres bin

pres2 = 300; % max depth (m) (Baltic depth\_max = 300 m)

grid\_p = 1; % interpolant distance (m) between depth levels

int\_method = 'linear'; % interpolation method (see doc interp1)

pres\_p = [pres1:grid\_p:pres2]'; % interpolated presure

zlevels = length(pres\_p); % number of depth levels in the interpolated vars

[i,j] = size(temp);

t2 = nan(zlevels,j);

s2 = nan(zlevels,j);

p2 = ones(zlevels,j).\*pres\_p;

% interpolate profiles z-axis (depth)

for j = 1 : Nobs

t1 = temp(:,j);

s1 = salt(:,j);

p1 = pres(:,j); % presure levels are the same for all the profiles

ind = isnan(t1) | isnan(p1) | isnan(s1) | p1 > pres2;

p1(ind) = [];

t1(ind) = [];

s1(ind) = [];

[p1, index] = unique(p1);

dp = diff(p1);

if min(dp) >= 0.01

t2\_interp = interp1(p1,t1(index),pres\_p,int\_method);

t2(:,j) = t2\_interp;

s2\_interp = interp1(p1,s1(index),pres\_p,int\_method);

s2 (:,j) = s2\_interp;

end

% clear vars within loop

clear t1 s1 p1 ind

end

t\_intp = t2;

s\_intp = s2;

p\_intp = p2;

# Argo Temperature and Salinity values

T\_ar = t\_intp;

S\_ar = s\_intp;

P\_ar = p\_intp;

clear t2 s2 p2

% 1/ plot T and S Argo profiles, raw Vs interp

nprof = find(platform==3901940); % Location North Atlantic [20161101]

% nprof = find(platform == 4901821); % Location: East Pacific [20161101];

if ~isempty(nprof)

T = temp (:,nprof);

S = salt (:,nprof);

P = pres (:,nprof);

T\_intp = t\_intp(:,nprof);

S\_intp = s\_intp(:,nprof);

P\_intp = p\_intp(:,nprof);

plat = platform(nprof);

Tmin = 0;

Tmax = 15;

Smin = 15;

Smax = 40;

figure(2); clf

% [1] Temperature profile

subplot(1,2,1)

h1 = plot (T, -P,'ko','markersize',10);

hold on

h2 = plot (T\_intp, -P\_intp,'ko','MarkerFaceColor','r');

ylim([-pres2 -pres1])

xlim ([Tmin Tmax])

xlabel ('T (\circ)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'raw';'interp'},'location','SouthWest')

title(['plaform: ' num2str(plat(1))]);

% [2] Salinity profile

subplot(1,2,2)

h1 = plot (S, -P,'ko','markersize',10);

hold on

h2 = plot (S\_intp(:,1), -P\_intp,'ko','MarkerFaceColor','b');

ylim([-pres2 -pres1])

xlim ([Smin Smax])

xlabel ('S (psu)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'raw' ;'interp'},'location','SouthWest')

title(['plaform: ' num2str(plat(1))]);

end

# Argo profile Classification

This sections makes two types of Argo classification, which are: Regional classification (Sub) regional Classification

* Bothian Sea [\*] (BOS) [61.5?N,19?E]
* Gulf of Finland (GOF)
* Northern Baltic Proper (NBP)
* Arkona Basin [\*] (ArB) [55?N, 16?E]

[\*] data to be presented in this validation. These regions contain both Argo and mooring data (i.e. allow study of MLD Argo Vs Mooring). Seasonal Classification

* Winter (wnt): December, January, February
* Spring (spr): March, April, May
* Summer (sum): June, July, August
* Autumn (aut): September, October, November

% Select Argo Floats within a searching distance 'r' (km) from region

% centre

r = 100; % radius (km)

lon\_BOS = 19.5;

lat\_BOS = 61.5;

lon\_GOF = 21.5;

lat\_GOF = 59.5;

lon\_NBP = 19.5;

lat\_NBP = 58.50;

lon\_ArB = 18;

lat\_ArB = 56;

## Plot Study region locations

% Set the map projection specifications

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

% h0 = plotm(lat,lon,'kp','MarkerFaceColor','r','MarkerSize',5);

h1 = plotm(lat\_BOS,lon\_BOS,...

'kp','MarkerFaceColor','g','MarkerSize',10); hold on

[c1] = plotm\_study\_region(lon\_BOS,lat\_BOS,r);

set(c1,'color','k','linewidth',2)

h2 = plotm(lat\_GOF,lon\_GOF,...

'kp','MarkerFaceColor','g','MarkerSize',10); hold on

[c1] = plotm\_study\_region(lon\_GOF,lat\_GOF,r);

set(c1,'color','k','linewidth',2)

h2 = plotm(lat\_NBP,lon\_NBP,...

'kp','MarkerFaceColor','g','MarkerSize',10); hold on

[c1] = plotm\_study\_region(lon\_NBP,lat\_NBP,r);

set(c1,'color','k','linewidth',2)

h2 = plotm(lat\_ArB,lon\_ArB,...

'kp','MarkerFaceColor','g','MarkerSize',10); hold on

[c1] = plotm\_study\_region(lon\_ArB,lat\_ArB,r);

set(c1,'color','k','linewidth',2)

% legend([h1(1),h2(1)],{'BOS''ArB'},'location','SouthEast');

framem on

axis off

tightmap

geoshow('landareas.shp','facecolor',[1 1 1].\*0.5)

rivers = shaperead('worldrivers', 'UseGeoCoords', true);

geoshow(rivers, 'Color', 'blue')

gridm on

title({[basin\_str ' Study locations']})

# Regional Argo Selection Bothian Sea

DIST\_BOS = Distance(lon\_BOS,lat\_BOS,lon,lat);

DIST\_ArC = Distance(lon\_ArB,lat\_ArB,lon,lat);

ind\_ar\_BOS = find(DIST\_BOS <= r);

ind\_ar\_ArB = find(DIST\_ArC <= r);

lon\_ar\_BOS = lon(ind\_ar\_BOS);

lat\_ar\_BOS = lat(ind\_ar\_BOS);

plat\_BOS = platform(ind\_ar\_BOS);

S\_ar\_BOS = S\_ar(:,ind\_ar\_BOS);

T\_ar\_BOS = T\_ar(:,ind\_ar\_BOS);

P\_ar\_BOS = P\_ar(:,ind\_ar\_BOS);

lon\_ar\_ArB = lon(ind\_ar\_ArB);

lat\_ar\_ArB = lat(ind\_ar\_ArB);

plat\_ArB = platform(ind\_ar\_ArB);

S\_ar\_ArB = S\_ar(:,ind\_ar\_ArB);

T\_ar\_ArB = T\_ar(:,ind\_ar\_ArB);

P\_ar\_ArB = P\_ar(:,ind\_ar\_ArB);

% [+] ARGO seasonal classificaton index of year

time\_ar\_season = datevec(time\_ar);

ind\_ar\_wnt = ismember(time\_ar\_season(:,2), [12,1,2]); % winter months (wnt)

ind\_ar\_spr = ismember(time\_ar\_season(:,2), [3,4,5]); % spring months (spr)

ind\_ar\_sum = ismember(time\_ar\_season(:,2), [6,7,8]); % summer months (sum)

ind\_ar\_aut = ismember(time\_ar\_season(:,2), [9,10,11]);% autumn months (aut)

% [+] Argo number of profiles each season

Nobs\_ar\_BOS = length(ind\_ar\_BOS);

Nobs\_ar\_ArB = length(ind\_ar\_ArB);

Nobs\_ar\_wnt = sum(ind\_ar\_wnt);

Nobs\_ar\_spr = sum(ind\_ar\_spr);

Nobs\_ar\_sum = sum(ind\_ar\_sum);

Nobs\_ar\_aut = sum(ind\_ar\_aut);

lat\_wnt = lat(ind\_ar\_wnt);

lon\_wnt = lon(ind\_ar\_wnt);

lat\_spr = lat(ind\_ar\_spr);

lon\_spr = lon(ind\_ar\_spr);

lat\_sum = lat(ind\_ar\_sum);

lon\_sum = lon(ind\_ar\_sum);

lat\_aut = lat(ind\_ar\_aut);

lon\_aut = lon(ind\_ar\_aut);

## Plot Geolocation: ARGO floats within Baltic region

### 1) ALL Argo floats

figure (1); clf; hold on

% Set the map projection specifications

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

title({[basin\_str ' (' num2str(lat\_min) '-' ...

num2str(lat\_max) '\circ N) ' ]; ...

['N\_{RAW} = ' num2str(Nobs)];...

[datestr(itime\_start,'yyyymmdd') '-'...

datestr(itime\_end,'yyyymmdd')]});

h1 = plotm(lat,lon,'ko','MarkerFaceColor','r');

h2 = plotm(lat,lon,'ko','MarkerFaceColor','b');

% legend([h1(1),h2(1)],{['Nobs\_{NA}=' num2str(Nobs\_NA)],...

% ['Nobs\_{PA}=' num2str(Nobs\_PA)]},'location','southeastoutside')

### 2) Argo floats in Winter

if sum(ind\_ar\_wnt)~=0 || sum(ind\_ar\_wnt)~=0

figure; clf; hold on

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

title({[basin\_str ' (' num2str(lat\_min) '-' ...

num2str(lat\_max) '\circ N) ' ]; ...

['N\_{wnt} = ' num2str(Nobs\_ar\_wnt)];...

[datestr(itime\_start,'yyyymmdd') '-'...

datestr(itime\_end,'yyyymmdd')]});

hold on

h1 = plotm(lat\_wnt,lon\_wnt,...

'ko','MarkerFaceColor',[0.3020 0.7451 0.9333]);

h2 = plotm(lat\_wnt,lon\_wnt,...

'ks','MarkerFaceColor',[0.3020 0.7451 0.9333]);

% legend([h1(1),h2(1)],{['Nobs\_{wnt}=' num2str(Nobs\_ar\_wnt)],...

% ['Nobs\_{PA}=' num2str(Nobs\_ar\_PA\_wnt)]},'location','southeastoutside')

end

### 2) Argo floats in Spring

if sum(ind\_ar\_spr)~=0 || sum(ind\_ar\_spr)~=0

figure; clf; hold on

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

title({[basin\_str ' (' num2str(lat\_min) '-' ...

num2str(lat\_max) '\circ N) ' ]; ...

['N\_{spr} = ' num2str(Nobs\_ar\_spr)];...

[datestr(itime\_start,'yyyymmdd') '-'...

datestr(itime\_end,'yyyymmdd')]});

hold on

icolor = [1 1 0]; % set FaceColor (c = uisetcolor)

h1 = plotm(lat\_spr,lon\_spr,'ko','MarkerFaceColor',icolor);

h2 = plotm(lat\_spr,lon\_spr,'ks','MarkerFaceColor',icolor);

% legend([h1(1),h2(1)],{['Nobs\_{NA}=' num2str(Nobs\_ar\_NA\_spr)],...

% ['Nobs\_{PA}=' num2str(Nobs\_ar\_PA\_spr)]},'location','southeastoutside')

end

### 3) Argo floats in Summer

if sum(ind\_ar\_sum)~=0 || sum(ind\_ar\_sum)~=0

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

title({[basin\_str ' (' num2str(lat\_min) '-' ...

num2str(lat\_max) '\circ N) ' ]; ...

['N\_{sum} = ' num2str(Nobs\_ar\_sum)];...

[datestr(itime\_start,'yyyymmdd') '-'...

datestr(itime\_end,'yyyymmdd')]});

hold on

icolor = [1.0000 0.4118 0.1608]; % set FaceColor (c = uisetcolor)

h1 = plotm(lat\_sum,lon\_sum,'ko','MarkerFaceColor',icolor);

h2 = plotm(lat\_sum,lon\_sum,'ks','MarkerFaceColor',icolor);

% legend([h1(1),h2(1)],{['Nobs\_{NA}=' num2str(Nobs\_ar\_NA\_sum)],...

% ['Nobs\_{PA}=' num2str(Nobs\_ar\_PA\_sum)]},'location','southeastoutside')

end

### 4) Argo floats in Autumn

if sum(ind\_ar\_aut)~=0 || sum(ind\_ar\_aut)~=0

map\_projection = 'merc';

lon\_min = xmin;

lon\_max = xmax;

lat\_min = ymin;

lat\_max = ymax;

lon\_step = 10;

lat\_step = 5;

figure; hold on

fillmap\_super(map\_projection,...

lon\_min,lon\_max,lat\_min,lat\_max,lon\_step,lat\_step);

hold on

title({[basin\_str ' (' num2str(lat\_min) '-' ...

num2str(lat\_max) '\circ N) ' ]; ...

['N\_{aut} = ' num2str(Nobs\_ar\_aut)];...

[datestr(itime\_start,'yyyymmdd') '-'...

datestr(itime\_end,'yyyymmdd')]});

hold on

icolor = [0.9294 0.6941 0.1255]; % set FaceColor (c = uisetcolor)

h1 = plotm(lat\_aut,lon\_aut,'ko','MarkerFaceColor',icolor);

h2 = plotm(lat\_aut,lon\_aut,'ks','MarkerFaceColor',icolor);

% legend([h1(1),h2(1)],{['Nobs\_{NA}=' num2str(Nobs\_ar\_NA\_aut)],...

% ['Nobs\_{PA}=' num2str(Nobs\_ar\_PA\_aut)]},'location','southeastoutside')

end

# Argo Averaged profiles (T and S) in each region

dim = 2; % average across columns (profiles)

flag = 0; % normalized std [1], or not [0]

plat\_BOS = platform(ind\_ar\_BOS);

plat\_ArB = platform(ind\_ar\_ArB);

T\_BOS = t\_intp(:,ind\_ar\_BOS);

T\_mean\_BOS = nanmean(T\_BOS,dim);

T\_std\_BOS = nanstd(T\_BOS,flag,dim);

T\_SEM\_BOS = T\_std\_BOS/(Nobs\_ar\_BOS).^.5; % std of the mean

T\_ArB = t\_intp(:,ind\_ar\_ArB);

T\_mean\_ArB = nanmean(T\_ArB,dim);

T\_std\_ArB = nanstd(T\_ArB,flag,dim);

T\_SEM\_ArB = T\_std\_ArB/(Nobs\_ar\_ArB).^.5; % std of the mean

S\_BOS = s\_intp(:,ind\_ar\_BOS);

S\_mean\_BOS = nanmean(S\_BOS,dim);

S\_std\_BOS = nanstd(S\_BOS,flag,dim);

S\_SEM\_BOS = T\_std\_BOS/(Nobs\_ar\_BOS).^.5; % std of the mean

S\_ArB = s\_intp(:,ind\_ar\_ArB);

S\_mean\_ArB = nanmean(S\_ArB,dim);

S\_std\_ArB = nanstd(S\_ArB,flag,dim);

S\_SEM\_ArB = T\_std\_ArB/(Nobs\_ar\_ArB).^.5; % std of the mean

P\_BOS = p\_intp(:,ind\_ar\_BOS);

P\_ArB = p\_intp(:,ind\_ar\_ArB);

P\_mean = p\_intp(:,1);

% Argo Seasonal classification in each region

% %

% Computation seasonal statistics, including mean, std, standard error of the

% mean (SEM).

%

% The following loop using "eval" function gives the following output (type

% AR\_SEASON):

% \* S\_ar\_aut: [501×188 double]

% \* S\_ar\_mean\_aut: [501×1 double]

% \* S\_ar\_std\_aut: [501×1 double]

% \* S\_ar\_SEM\_aut: [501×1 double]

season\_str = {'wnt','spr','sum','aut'};

param = {'T','S'};

region ={'BOS','ArB'};

for yy = 1:length(season\_str)

for xx = 1: length(param)

for zz = 1:length(region)

iseason = season\_str{yy};

iparam = param{xx};

iregion = region{zz};

eval(['plat\_' '\_' iseason ' = platform(ind\_ar\_' iseason ');']);

eval([iparam '\_ar\_' iseason ' = ' iparam '\_ar (:,ind\_ar\_' iseason ');']);

eval([iparam '\_ar\_mean\_' iseason ' = nanmean(' iparam '\_ar\_' iseason ',dim);']);

eval([iparam '\_ar\_std\_' iseason '= nanstd(' iparam '\_ar\_' iseason ',flag,dim);']);

eval([iparam '\_ar\_SEM\_' iseason ' = ' iparam '\_ar\_std\_' iseason '/(Nobs\_ar\_' iseason ').^.5;']); % std of the mean

AR\_SEASON = struct;

eval(['AR\_SEASON.' iparam '\_ar\_' iseason ' = ' iparam '\_ar\_' iseason ';'])

eval(['AR\_SEASON.' iparam '\_ar\_mean\_' iseason ' = ' iparam '\_ar\_mean\_' iseason ';'])

eval(['AR\_SEASON.' iparam '\_ar\_std\_' iseason ' = ' iparam '\_ar\_std\_' iseason ';'])

eval(['AR\_SEASON.' iparam '\_ar\_SEM\_' iseason ' = ' iparam '\_ar\_SEM\_' iseason ';'])

end

end

end

% [Argo] Plot average profiles in NA and PA region

Plot TS Profiles Limits axes

Tmin = 0;

Tmax = 15;

Smin = 0;

Smax = 10;

figure

% [1] Temperature profile

X1 = T\_mean\_BOS;

Y1 = -P\_mean;

Z1 = T\_std\_BOS;

X2 = T\_mean\_ArB;

Y2 = -P\_mean;

Z2 = T\_std\_ArB;

subplot(1,2,1)

h1 = boundedline(X1,Y1,Z1,...

'-r','alpha','orientation','horiz'); hold on

h1 = plot (X1, Y1,'-','linewidth',2,'Color','r');

hold on

h2 = boundedline(X2,Y2,Z2,...

'-b','alpha','orientation','horiz'); hold on

h2 = plot (X2, Y2,'-','linewidth',2,'Color','b');

ylim([-pres2 -pres1])

xlim ([Tmin Tmax])

xlabel ('T (\circ)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'BOS';'ArB'},'location','SouthWest')

title('Argo Temperature');

% [2] Salinity profile

X1 = S\_mean\_BOS;

Y1 = -P\_mean;

Z1 = S\_std\_BOS;

X2 = S\_mean\_ArB;

Y2 = -P\_mean;

Z2 = S\_std\_ArB;

subplot(1,2,2)

h1 = boundedline(X1,Y1,Z1,...

'-r','alpha','orientation','horiz'); hold on

h1 = plot (X1, Y1,'-','linewidth',2,'Color','r');

hold on

h2 = boundedline(X2,Y2,Z2,...

'-b','alpha','orientation','horiz'); hold on

h2 = plot (X2, Y2,'-','linewidth',2,'Color','b');

ylim([-pres2 -pres1])

xlim ([Smin Smax])

xlabel ('S (psu)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'BOS';'ArB'},'location','SouthWest')

title('Argo Salinity');

## [Argo] Plot average Seasonal (winter/summer) profiles in BOS region

X1 = T\_ar\_mean\_wnt;

Y1 = -P\_mean;

Z1 = T\_ar\_std\_wnt;

X2 = T\_ar\_mean\_sum;

Y2 = -P\_mean;

Z2 = T\_ar\_std\_sum;

figure

% [1] Temperature profile

subplot(1,2,1)

h1 = boundedline(X1,Y1,Z1,...

'-b','alpha','orientation','horiz');

hold on

h1 = plot (X1, Y1,'-','linewidth',2,'Color','b');

hold on

h2 = boundedline(X2,Y2,Z2,...

'-r','alpha','orientation','horiz');

hold on

h2 = plot (X2, Y2,'-','linewidth',2,'Color','r');

ylim([-pres2 -pres1])

xlim ([Tmin Tmax])

xlabel ('T (\circ)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'WNT';'SUM'},'location','SouthWest')

title('Argo Temperature');

% [2] Salinity profile

X1 = S\_ar\_mean\_wnt;

Y1 = -P\_mean;

Z1 = S\_ar\_std\_wnt;

X2 = S\_ar\_mean\_sum;

Y2 = -P\_mean;

Z2 = S\_ar\_std\_sum;

subplot(1,2,2)

h1 = boundedline(X1,Y1,Z1,...

'-b','alpha','orientation','horiz');

hold on

h1 = plot (X1,Y1,'-','linewidth',2,'Color','b');

hold on

h2 = boundedline(X2,Y2,Z2,...

'-r','alpha','orientation','horiz');

hold on

h2 = plot (X2, Y2,'-','linewidth',2,'Color','r');

ylim([-pres2 -pres1])

xlim ([Smin Smax])

xlabel ('S (psu)'); ylabel('depth (m)')

grid on

legend ([h1(1);h2(1)],{'WNT';'SUM'},'location','SouthWest')

title('Argo Salinity');

# [Mooring] Read mooring data (location: 50N 145W)

* There is no mooring data at the Baltic (at least not included in the indata)
* Alternatively plot T and S analysis profiles.

## Satellite and Argo comparison againts PAPA mooring (BOXPLOTS)

[PAPA+ARGO] Boxplot T and S at PAPA location Show example of floats operating at searching destance from PAPA mooring location.

itime = itime\_start; % first day of the month (i.e. use SSS colored map)

% data\_type = 1; % Arctic+

% plot\_type = 3; % lambert projection

% ibasin = 7; % Arctic region

% region = 'PA'; % Arctic Pacific region ['PA'] or North Atlantic ['PA'])

% idepth\_max = 10;% Depth (m) to compare profiles (argo and PAPA)

%

%

% % Get T and S at 10 m (P <= 10 m)

%

% %[\*] Argo@10m winter/summer

% [row,col] = find(P\_ar\_PA\_irange >= idepth\_max,1,'first');

%

% T10\_PA\_wnt\_irange = T\_ar\_PA\_wnt\_irange(row,:);

% S10\_PA\_wnt\_irange = S\_ar\_PA\_wnt\_irange(row,:);

% P10\_PA\_wnt\_irange = P\_ar\_PA\_wnt\_irange(row,:);

%

% T10\_PA\_sum\_irange = T\_ar\_PA\_sum\_irange(row,:);

% S10\_PA\_sum\_irange = S\_ar\_PA\_sum\_irange(row,:);

% P10\_PA\_sum\_irange = P\_ar\_PA\_sum\_irange(row,:);

%

% %[\*] PAPA@10m winter/summer

% [row,col] = find(p\_mean >= 10,1,'first');

%

% t10\_papa\_wnt = t\_papa\_wnt(row,:);

% s10\_papa\_wnt = s\_papa\_wnt(row,:);

%

% t10\_papa\_sum = t\_papa\_sum(row,:);

% s10\_papa\_sum = s\_papa\_sum (row,:);

%

%

% TEMP = nan(2000,4);

% SALT = TEMP;

%

% param\_temp = {'T10\_PA\_wnt\_irange','t10\_papa\_wnt','T10\_PA\_sum\_irange','t10\_papa\_sum'};

% param\_salt = {'S10\_PA\_wnt\_irange','s10\_papa\_wnt','S10\_PA\_sum\_irange','s10\_papa\_sum'};

%

% for n = 1:4

%

% eval(['ind\_temp = length(' param\_temp{n} ');'])

% eval(['ind\_salt = length(' param\_salt{n} ');'])

%

% eval(['TEMP(1:ind\_temp,' num2str(n) ') = ' param\_temp{n} ';'])

% eval(['SALT(1:ind\_salt,' num2str(n) ') = ' param\_salt{n} ';'])

%

% end

%

% SALT(SALT<20) = NaN; % Argo showed result (1 float) with S10 below 10 psu

%

%

% % BOPLOT: Show Mooring Vs Argo at 10 m depth

% figure

%

% boxplot(TEMP,param\_temp)

% title({['Tepeture (Argo and moooring) at PAPA location (' num2str(idepth\_max) ' km)'];num2str(iyear)});

%

% boxplot(SALT,param\_salt)

% title({['Salintiy (Argo and moooring) at PAPA location (' num2str(idepth\_max) ' km)'];num2str(iyear)});

%

## [Mooring+ARGO+SAT] Compare Salinity at 10 m depth from Mooring and ARGO Vs Satellite data

**Not done because there are not mooring stations.**

In the previous sections, plots showed that MLD is as large as 100 and 80 m in winter and summer respectively. Thus it is possible to assume that salinity at 10 m depth should compare well with the satellite retrieved salinity at skin of the of skin of the ocean (i.e. top milimeters of the surface).

Make scatterplot of salinity data including:

* Mooring salinity at 10 m (only one location)
* Argo salinity at 10 m (variable location)
* SSS map (satellite) surface map

## SSS-BEC (sss\_bec) at mooring location

* Baltic+ product ranges: February 2011- December 2013
* Red SSS data at each study region (i.e. BOS, GOF,NBP and ArB).

itime\_min = datenum(2011,02,1,0,0,0);

itime\_max = datenum(2014,01,05,0,0,0);

time\_range = itime\_min:itime\_max;

% To choose BEC product: [1] Arctic+ (v3), [2] BEC Arctic (v2), [3] Global

data\_type = 4;

ibasin = 9;

if data\_type == 4 % Baltic+ product (v3.0)

data\_version\_str = 'Baltic\_NOM\_v1';

elseif data\_type == 5 % Arctic BEC product (v2.0)

data\_version\_str = 'Baltic\_NS\_v1';

elseif data\_type == 3 % Use the BEC Global product (v001)

data\_version\_str = 'Global\_v001';

else % Use the BEC Global product (default mode)

data\_version\_str = 'Global\_v001';

end

[iX,iY] = find(~isnan(time\_range) & time\_range > itime\_min & time\_range <= itime\_max);

lon\_BOS = 19.5;

lat\_BOS = 61.5;

lon\_GOF = 21.5;

lat\_GOF = 59.5;

lon\_NBP = 19.5;

lat\_NBP = 58.50;

lon\_ArB = 18;

lat\_ArB = 56;

regions\_str ={'BOS','GOF','NBP','ArB'};

lon\_regions = [lon\_BOS, lon\_GOF, lon\_NBP,lon\_ArB];

lat\_regions = [lat\_BOS, lat\_GOF, lat\_NBP, lat\_ArB];

REGION = regions\_str{1}

folder\_out = [folder\_tmp 'Baltic/Validation/Regions/' REGION '/SSS/'];

foldercheck\_raf(folder\_out)

for xx = 1:length(regions\_str)

lon\_this = lon\_regions(xx);

lat\_this = lat\_regions(xx);

end

fn\_out = [folder\_out 'SSS\_SMOS\_BEC\_L4\_' data\_version\_str '\_' REGION...

'\_r' num2str(r) '.mat'];

nTIME = length(time\_range);

lon\_bec = nan(nTIME,1);

lat\_bec = nan(nTIME,1);

sss\_bec = nan(nTIME,1);

if exist(fn\_out,'file') ~= 2 % < SET TO '2' when you want to run this section! >

for tt = 1: length(iX)

disp ([num2str(tt) '/' num2str(length(iX))])

ind = iX(tt);

TIME2 = time\_range(ind);

if TIME2 >= itime\_min && TIME2 <= itime\_max

% [TT1] = rd\_smos\_L4\_BEC(TIME2,ibasin); % need to download data

[TT1] = rd\_smos\_L4\_BEC\_v1r2(TIME2,ibasin,data\_type);

TIME\_SMOS = TT1.time\_out;

SSS\_SMOS = TT1.sss;

LON = TT1.lon;

LAT = TT1.lat;

DIST = Distance(lon\_this,lat\_this,LON,LAT);

irange = find(DIST <= r);

Ln = length(irange);

if Ln > 1

lon\_bec(ind,1) = nanmean(LON(irange));

lat\_bec(ind,1) = nanmean(LAT(irange));

sss\_bec(ind,1) = nanmean(SSS\_SMOS(irange));

else

lon\_bec(ind,1) = LON(irange);

lat\_bec(ind,1) = LAT(irange);

sss\_bec (ind,1) = SSS\_SMOS(irange);

end

end

end

save(fn\_out,'lon\_bec','lat\_bec','sss\_bec')

elseif exist(fn\_out,'file') == 2

load(fn\_out)

end

% apply a running average to data (i.e. window of 5 and 10 days averages)

wndw1 = 5; % window1 of ndays1

wndw2 = 10; % window2 of ndays2

[sss1\_bec] = movmean(sss\_bec,wndw1); % sss-bec 5d averages

[sss2\_bec] = movmean(sss\_bec,wndw2); % sss-bec 10d averages

[sss3\_bec] = movmean(sss\_bec,20); % sss-bec 20d averages

sss1\_bec(1:wndw1,1) = NaN;

sss2\_bec(1:wndw2,1) = NaN;

% Seasonal classification of SSS-BEC product

* This section uses the SSS-BEC data loaded in the previous section.
* Data is in format of the PAPA mooring station (i.e. the fiudecial reference)
* The seasonal classification uses the seasonal index created using time\_papa (see above sections)

sss\_bec\_wnt = sss\_bec(ind\_wnt,1);

sss\_bec\_spr = sss\_bec(ind\_spr,1);

sss\_bec\_sum = sss\_bec(ind\_sum,1);

sss\_bec\_aut = sss\_bec(ind\_aut,1);

% Boxplot of the Seasonal salnity Argo, mooring and SSS-BEC values at PAPA location

param\_salt = {'sss\_bec\_wnt', 'S10\_PA\_wnt\_irange','s10\_papa\_wnt','sss\_bec\_sum','S10\_PA\_sum\_irange','s10\_papa\_sum'};

nparam = length(param\_salt);

SALT = nan(2000,nparam);

for n = 1:nparam

eval(['ind\_salt = length(' param\_salt{n} ');'])

eval(['SALT(1:ind\_salt,' num2str(n) ') = ' param\_salt{n} ';'])

end

SALT(SALT<20) = NaN; % Argo showed result (1 float) with S10 below 10 psu

% BOPLOT: Show Mooring Vs Argo at 10 m depth

figure

boxplot(SALT,param\_salt)

title({['Salinity (SSS-BEC, Argo and moooring) at PAPA location (depth max=' num2str(idepth\_max) ' m)'];num2str(iyear)});

nfloat\_irange = length(time\_ar\_PA\_irange);

n = 1;

itime = time\_ar\_PA\_irange(n);

Arctic\_plot\_fun(ibasin,itime,data\_type,plot\_type,region)

% Plot PAPA location over SSS-BEC map [SCATTERPLOT]

* Mooring location with a searching radius 'r' (km)
* Background SSS-BEC product --> Show the greater spatial coverage of satellite product

figure(gcf); hold on

[c1] = plotm\_study\_region(lon\_papa,lat\_papa,r);

set(c1,'color','k','linewidth',2)

scatterm(lat\_papa,lon\_papa,...

20,s\_papa\_intp(11,1),'filled','MarkerEdgeColor','k')