EXAMPLES USING THE IMOS USER CODE LIBRARY (MATLAB VERSION)

Version 1.0

Prepared by *Laurent Besnard*

IMOS - eMII

15/05/2013

Integrated **Marine**
Observing System

Table of Contents

1.Introduction......1
1.1Installation of the IMOS User Code Library (MATLAB)...... 1
1.2Finding an IMOS NetCDF File...... 1
2.General Features of the IMOS user code library...... 2
2.1Output structure...... 2
2.2Discover Metadata...... 2
2.3Discover Variables...... 3
2.4Discover Dimensions...... 3
3.Dataset examples – Using the NetCDF Parser for Plotting......5
3.1AATAMS – Animal Tagging and Monitoring - non QC'd data...... 5
3.2ABOS – Deep Water Mooring...... 9
3.2.1Southern Ocean Time-series - non QC'd data...... 9
3.3ACORN – Ocean Radar - non QC'd data...... 10
3.4ANFOG – Ocean Gliders - QC'd good data...... 12
3.5ANMN – National Mooring Network - QC'd good data...... 14
3.6AUV – Autonomous Underwater Vehicle - non QC'd data...... 16
3.7Argo – Argo Floats Program - non QC'd data...... 18
3.7.1Example 1...... 18
3.7.2Example 2...... 21
3.8FAIMMS – Wireless Sensor Networks - QC'd good data...... 23
3.9SOOP – Ship Of Opportunities...... 25
3.9.1XBT - expandable bathythermographs - QC'd data...... 25
3.10SRS – Satellite Remote Sensing 29
3.10.1Bio-Optical database – Pigment data 29
3.10.2Bio-Optical database – Absorption data 31
3.10.3GHRSST – L3P mosaic 34
3.10.4GHRSST – L3C – multi swath, single sensor 36
3.10.5GHRSST – L3S – multi swath, multi sensor, one day 38

Revision History

Name	**Date**	**Reason For Changes**	**Version**

1. Introduction

This document intends to present how to load IMOS NetCDF data into a MATLAB environment, and offers some suggestions about how to use the data once loaded. The starting point is running the NetCDF parser from the IMOS user code library.

The examples provided in this document only represent a tiny bit of the content of most of the NetCDF files. There are usually many more variables available in a NetCDF file, and therefore many other ways to display data.

1.1 Installation of the IMOS User Code Library (MATLAB)

The IMOS User Code Library for MATLAB can be downloaded from:
https://github.com/aodn/imos_user_code_library/tree/master/MATLAB_R2011

It can be checked out using a Git client, or be downloaded as a zip file :
https://github.com/aodn/imos_user_code_library/archive/master.zip

A tertiary toolbox, nctooblox, needs to be installed. Please follow the description available at http://code.google.com/p/nctoolbox/.

1.2 Finding an IMOS NetCDF File

In order to find a dataset you are interested in, please refer to the portal help:
http://portalhelp.aodn.org.au/Portal2_help/
A HOWTO has been written to help any user in his way to find an IMOS NetCDF file.

For users who are already familiar with IMOS facilities and datasets, IMOS NetCDF files are also directly accessible via an OPeNDAP catalog at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/catalog.html

Once a NetCDF file has been chosen to work with (See http://portalhelp.aodn.org.au/Portal2_help/?q=node/112), the user needs to go to the ***'OPeNDAP Dataset Access Form'*** page. The **'Data URL'** located just above the **'Global Attributes'** field is the URL which needs to be replaced in the examples which we'll present in the following sections.

The NetCDF file can also be downloaded to the user’s local machine using the HTTP Server option on the THREDDS options page. In this case, the user has to replace the 'Data URL' variable with the local address of the NetCDF file when using ncParse in the following examples.

2. General Features of the IMOS user code library

The first step consists in parsing a NetCDF file, whether this file is available locally or remotely on an OPeNDAP server.

Type in your MATLAB command window:

```
file_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SOOP/SOOP-TMV/VLST_Spirit-of-Tasmania-1/transect/2011/07/IMOS_SOOP-TMV_TSB_20110709T092744Z_VLST_FV02_transect-D2M_END-20110709T203700Z.nc' ;
dataset = ncParse(file_URL) ;
```

2.1 Output structure

ncParse provides a Matlab structure with 2 fields variables and metadata. Metadata contains the NetCDF global attributes while variables contains the variable attributes and their values...

```
>> dataset
```

```
metadata: [1x1 struct]
variables: [1x1 struct]
dimensions: [1x1 struct]
```

2.2 Discover Metadata

In order to see all the global attributes available, type in your command window:

```
>> dataset.metadata
```

```
            project: 'Integrated Marine Observing System (IMOS)'
        conventions: 'IMOS-1.2'
              title: [1x159 char]
        institution: 'Environment Protection Authority Victoria (EPA Vic)'
       date_created: '2011-08-02T10:51:05Z'
           abstract: [1x330 char]
             source: 'ship observation'
           keywords: [1x192 char]
      platform_code: 'VLST'
     netcdf_version: '3.6'
   naming_authority: 'IMOS'
quality_control_set: '1'
 geospatial_lat_min: -41.177570000000003
 geospatial_lat_max: -37.850310000000000
 geospatial_lon_min: 1.446208400000000e+02
 geospatial_lon_max: 1.463750700000000e+02
geospatial_vertical_min: 0
geospatial_vertical_max: 0
time_coverage_start: '2011-07-09T09:27:44Z'
```

```
        time_coverage_end: '2011-07-09T20:37:00Z'
          local_time_zone: 10
        data_centre_email: 'info@emii.org.au'
              data_centre: 'eMarine Information Infrastructure (eMII)'
   principal_investigator: 'Lee, Randall EPA'
   institution_references: 'http://imos.org.au/emii.html'
                 citation: [1x126 char]
           acknowledgment: [1x385 char]
   distribution_statement: [1x118 char]
          netcdf_filename: 'IMOS_SOOP-TMV_TSB_20110709T092744Z_VLST_FV02_transect-D2M_END-20110709T203700Z.nc'
```

You can attach to the Matlab variable 'title' the corresponding NetCDF global attribute:

```
>> title = dataset.metadata.title
```

```
title =
Temperature, salinity, fluorescence and turbidity Data collected by the Spirit of Tasmania ferry 1 - Transect between Devonport and Melbourne on the09-Jul-2011
```

2.3 Discover Variables

In order to list all the variables available in each NetCDF file, type:

```
>> dataset.variables
```

```
TEMP: [1x1 struct]
TEMP_2: [1x1 struct]
COND: [1x1 struct]
PSAL: [1x1 struct]
CPHL: [1x1 struct]
TURB: [1x1 struct]
LATITUDE: [1x1 struct]
LONGITUDE: [1x1 struct]
```

To access the Temperature variable :

- data values

```
>> dataset.variables.TEMP.data
```

- names of corresponding dimension variables

```
>> dataset.variables.TEMP.dimensions
```

```
   'TIME'
```

- quality control flags

```
>> dataset.variables.TEMP.flags
```

- variable attributes

2.4 Discover Dimensions

The dimensions values attached to each variable, visible in dataset.variables.VARIABLE.dimensions, can be found in:

```
>> dataset.dimensions.TIME.data % 1st possibility
```

```
>> dataset.dimensions.(char(dataset.variables.TEMP.dimensions{1})).data % 2nd possibility to access to the data of the first dimension.
```

3. Dataset examples – Using the NetCDF Parser for Plotting

3.1 AATAMS – Animal Tagging and Monitoring - non QC'd data

The Australian Animal Tagging And Monitoring System (AATAMS) is a coordinated marine animal tagging project. CTD Satellite Relay Data Loggers are used to explore how marine mammal behaviour relates to their oceanic environment.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/AATAMS/marine_mammal_ctd-tag/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot all the animal's dives as a single profile time-series of temperature, measured by a CTD tag.

```
%% AATAMS - Animal Tagging and Monitoring
aatams_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/AATAMS/marine_mammal_ctd-tag/2009_2011_ct64_Casey_Macquarie/ct64-M746-09/IMOS_AATAMS-SATTAG_TSP_20100205T043000Z_ct64-M746-09_END-20101029T071000Z_FV00.nc';
aatams_DATA = ncParse(aatams_URL) ;

nProfiles = length (aatams_DATA.dimensions.profiles.data);

% creation of a 2 dimension array for temperature, pressure and salinity
for profileNumber = 1 : nProfiles
    indexVar = (aatams_DATA.variables.parentIndex.data == profileNumber); % a logical array of index

    tempVec = aatams_DATA.variables.TEMP.data(indexVar);
    tempData(profileNumber,1:length(tempVec)) = tempVec; clear tempVec

    presVec = aatams_DATA.variables.PRES.data(indexVar);
    presData(profileNumber,1:length(presVec)) = presVec; clear tempVec

    psalVec = aatams_DATA.variables.PSAL.data(indexVar);
    psalData(profileNumber,1:length(psalVec)) = psalVec; clear tempVec
end

% we replace the 0 values automatically created by Matlab with NaN
psalData(psalData == 0) = NaN;
presData(presData == 0) = NaN;
tempData(tempData == 0) = NaN;


timeData = aatams_DATA.variables.TIME.data;
latProfile = aatams_DATA.variables.LATITUDE.data;
lonProfile = aatams_DATA.variables.LONGITUDE.data;

%longitude in the original dataset goes from -180 to +180
```

```
%For a nicer plot, we change the values to the  [0  360] range
lonProfile(lonProfile < 0 ) = lonProfile(lonProfile < 0 ) +360 ;

% creation of the Time array
[nline, ncol] = size(tempData);
sizer = ones(1, ncol);
TIME_CYCLE_NUMBER2D = timeData * sizer;

%plot all the profiles as a timeseries
figure1 = figure;
set(figure1, 'Renderer', 'painters') %to get rid of renderer bug with dateticks
set(figure1, 'Position',  [1 1000 1100 900 ], 'Color',[1 1 1]);
subplot(2,2,1:2),
pcolor(TIME_CYCLE_NUMBER2D, -presData, tempData);
datetick('x',20)
shading interp
cmap = colorbar('location','EastOutside');
set(get(cmap,'ylabel'),'string',strrep([aatams_DATA.variables.TEMP.long_name ' in '
aatams_DATA.variables.TEMP.units ],'_',' '),'Fontsize',10)


title({[aatams_DATA.metadata.species_name  ' - released in ' aatams_DATA.metadata.release_site ' /
animal reference number : ' aatams_DATA.metadata.unique_reference_code],...
    })
zlabel(strrep([aatams_DATA.variables.TEMP.long_name ' in ' aatams_DATA.variables.TEMP.units],'_', '
'))
xlabel('Time in DD/MM/YY')
ylabel(strrep([aatams_DATA.variables.PRES.long_name ' in negative '
aatams_DATA.variables.PRES.units],'_', ' '))


%plot the  LAT timeseries
subplot(2,2,3),plot(TIME_CYCLE_NUMBER2D,latProfile)
title(strrep([aatams_DATA.variables.LATITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([aatams_DATA.variables.LATITUDE.long_name ' in '
aatams_DATA.variables.LATITUDE.units],'_', ' '))
datetick('x',20)

%plot the  LON timeseries
subplot(2,2,4),plot(TIME_CYCLE_NUMBER2D,lonProfile)
title(strrep([aatams_DATA.variables.LONGITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([aatams_DATA.variables.LONGITUDE.long_name ' in '
aatams_DATA.variables.LONGITUDE.units],'_', ' '))
datetick('x',20)

% plot of a single profile
profileToPlot = 1 ; % this is arbitrary. We can plot all profiles from 1 to nProfiles, modify
profileToPlot as desired

figure2 = figure;
set(figure2, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);
plot (tempData(profileToPlot,:),presData(profileToPlot,:))
title({aatams_DATA.metadata.title,...
```

```
    [ 'location',num2str(latProfile(profileToPlot),'%2.3f'),'/',num2str(lonProfile(profileToPlot),'
%3.2f') ],...
    [ datestr(timeData(profileToPlot)) 'UTC']})
xlabel([strrep(aatams_DATA.variables.TEMP.long_name,'_', ' ') ' in '
aatams_DATA.variables.TEMP.units])
ylabel([strrep(aatams_DATA.variables.PRES.long_name,'_', ' ') ' in '
aatams_DATA.variables.PRES.units])
```

Variables to modify :

- aatams_URL : the opendap url of the chosen file
- ProfileToPlot : the profile number to plot.

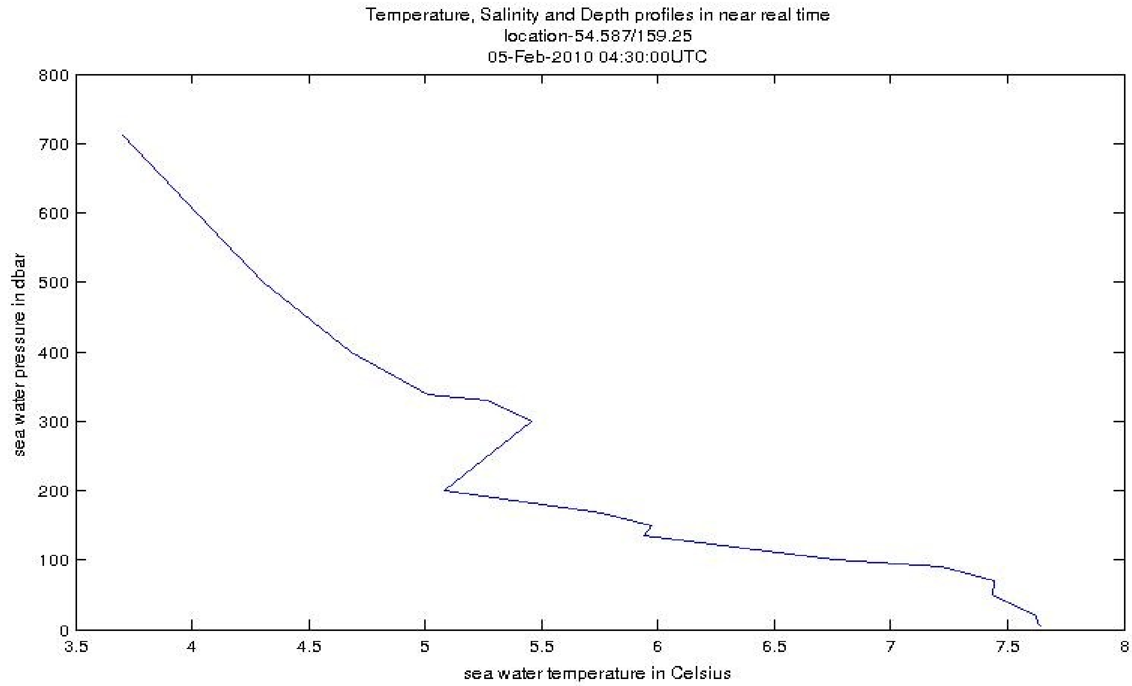

Illustration 1: Example of a Temperature profile from AATAMS data

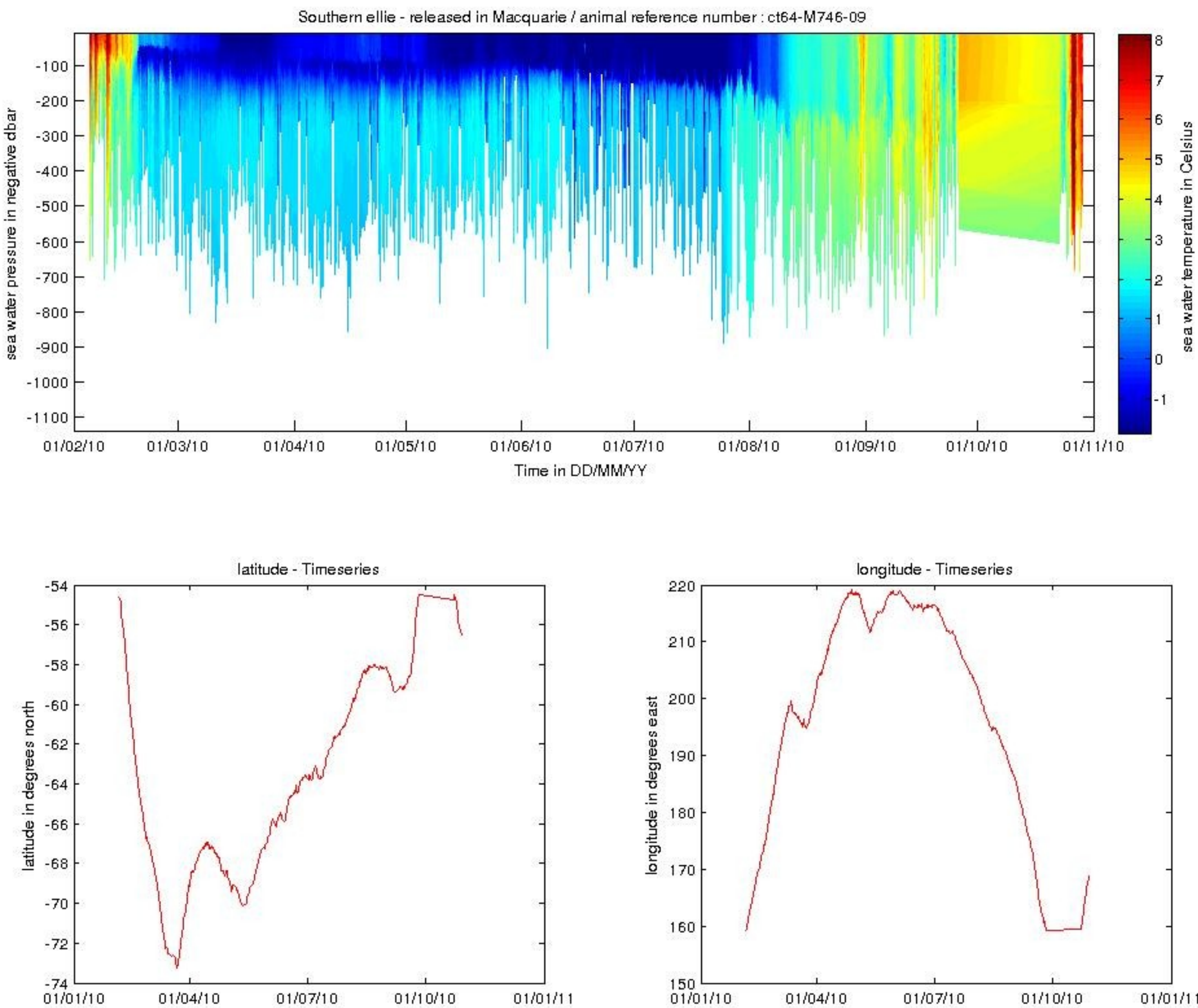

Illustration 2: Example of a Temperature Profile Time-series from AATAMS data

3.2 ABOS – Deep Water Mooring

3.2.1 Southern Ocean Time-series - non QC'd data

The Southern Ocean Time Series (SOTS) sub-facility provides high temporal resolution observations in sub-Antarctic waters. Observations are broad and include measurements of physical, chemical and biogeochemical parameters from multiple deep-water moorings.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/ABOS/SOTS/catalog.html

In the example below, the ncParse function is used to extract temperature data from a Pulse mooring instrument and then produce a temperature time series plot .

```
abos_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/ABOS/SOTS/Pulse/IMOS_ABOS-SOTS_20110803T000000Z_PULSE_FV01_PULSE-8-2011_END-20120719T000000Z_C-20121009T214808Z.nc' ;
abos_DATA = ncParse(abos_URL) ;

tempDataStructure = abos_DATA.variables.TEMP_85_1;
tempData = tempDataStructure.data;
timeData = abos_DATA.dimensions.(char(tempDataStructure.dimensions)).data;

abstract = abos_DATA.metadata.abstract;

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot (timeData,tempData)
title([abos_DATA.metadata.title ' at ' num2str(tempDataStructure.sensor_depth) ' m depth' ])
xlabel([strrep(abos_DATA.dimensions.(char(tempDataStructure.dimensions)).long_name,'_', ' ')])
ylabel([strrep( tempDataStructure.standard_name,'_', ' ') ' in '  tempDataStructure.units])
datetick('x',12)
```

It is possible to see the abstract by typing in the MATLAB command window

```
>> abstract
```

"The Pulse 6 mooring was deployed from September 2009 to March 2010 at Lat -46.3224, Lon 140.6776. Moored instruments are deployed by the IMOS Australian Bluewater Observing System (ABOS) Southern Ocean Time Series sub-facility for time-series observations of physical, biological, and chemical properties, in the Sub-Antarctic Zone southwest of Tasmania, with yearly servicing. The Southern Ocean Time Series (SOTS) Sub-Facility is responsible for the deployment of Pulse moorings. These time-series observations are crucial to resolving ecosystem processes that affect carbon cycling, ocean productivity and marine responses to climate variability and change, ocean acidification and other stresses. "

Variables to modify :

- abos_URL : the opendap url of the chosen file
- tempDataStructure : we arbitrarily chose the variable called TEMP_85_1, but many more are available.

3.3 ACORN – Ocean Radar - non QC'd data

The Australian Coastal Ocean Radar Network (ACORN) facility comprises a coordinated network of HF radars delivering real-time, non-quality controlled and delayed-mode, quality controlled surface current data into a national archive.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/ACORN/catalog.html

Monthly aggregated files are also available in the following folders:

- monthly_gridded_1h-avg-current-map_QC
- monthly_gridded_1h-avg-current-map_non-QC

In the example below, we demonstrate how to use the ncParse function to plot velocity data for one time value only in a latitude / longitude grid.

```
acorn_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/ACORN/monthly_gridded_1h-avg-current-map_non-QC/TURQ/2012/IMOS_ACORN_V_20121001T000000Z_TURQ_FV00_monthly-1-hour-avg_END-20121029T180000Z_C-20121030T160000Z.nc.gz' ;
acorn_DATA = ncParse(acorn_URL) ;

% we load the data. Casting data to double to be used afterwards with surface function
speedData = double(acorn_DATA.variables.SPEED.data);
latData = double(acorn_DATA.variables.LATITUDE.data);
lonData = double(acorn_DATA.variables.LONGITUDE.data);
timeData = acorn_DATA.dimensions.TIME.data;

% sea water U and V components
uData = (acorn_DATA.variables.UCUR.data);
vData = (acorn_DATA.variables.VCUR.data);

% Only one time value is being plotted. modify timeIndex if
% desired (value between 1 and length(timeData)
timeIndex = 5;

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

quiver(lonData,latData,squeeze(uData(timeIndex,:,:)),squeeze(vData(timeIndex,:,:)),1.5,'LineWidth',
1,'Color','k')
hold all

% to place a quiver plot on top of a surface plot
z = lonData .* exp(-lonData.^2 - latData.^2);
h = surface(lonData ,latData , squeeze(speedData(timeIndex,:,:)));
set(h,'ZData',-1+0*z) % Move the surface plot to Z = -1 in order to plot quivers over surface

shading interp
cmap = colorbar ;
caxis([min(min(min(speedData(timeIndex,:,:)))) max(max(max(speedData(timeIndex,:,:))))])
```

```
set(get(cmap,'ylabel'),'string',[acorn_DATA.variables.SPEED.long_name ' in '
acorn_DATA.variables.SPEED.units ],'Fontsize',10)

title({acorn_DATA.metadata.title ,...
    datestr(timeData(timeIndex),31) })
xlabel(acorn_DATA.variables.LONGITUDE.long_name)
ylabel(acorn_DATA.variables.LATITUDE.long_name)
```

Variables to modify :

- acorn_URL : the opendap url of the chosen file
- timeIndex : the time index number to plot

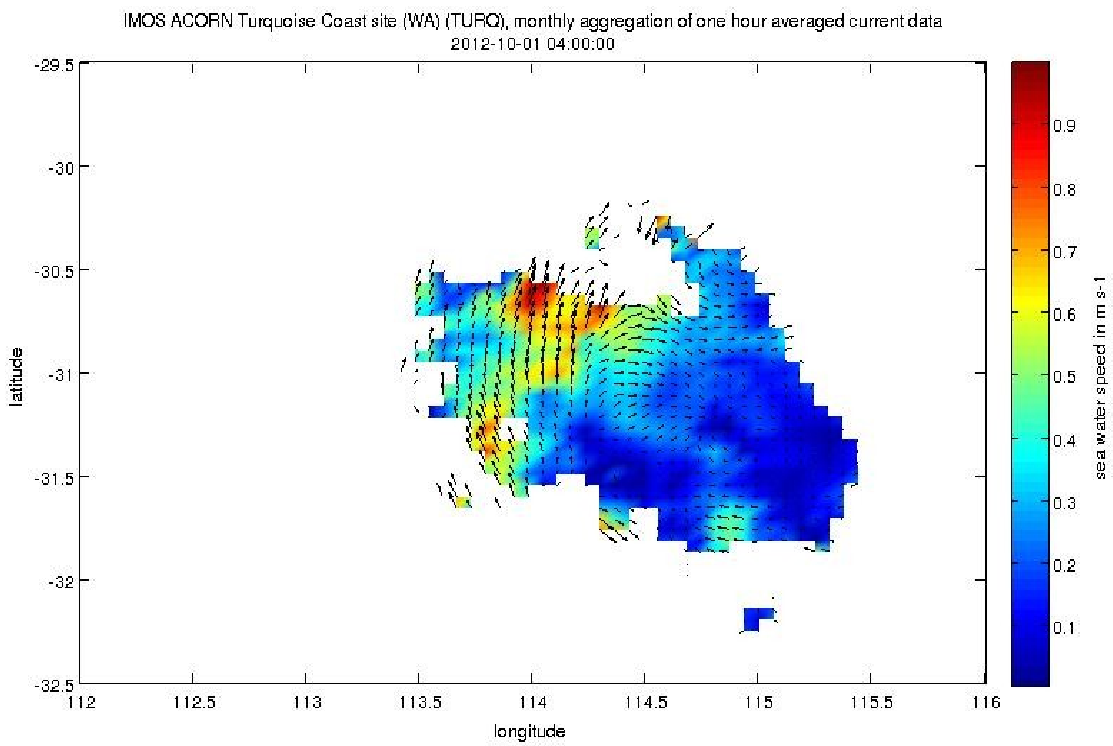

Illustration 3: Example of a Sea Water Speed gridded data with a Velocity Field from ACORN data

3.4 ANFOG – Ocean Gliders - QC'd good data

The Australian National Facility for Ocean Gliders (ANFOG), with IMOS/NCRIS funding, deploys a fleet of eight gliders around Australia.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/ANFOG/seaglider/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot salinity data as well as depth data in a same graph. Only the data points with a Quality Control flag greater than 1 (which means 'good data' , please refers to IMOS NetCDF User Manual for a description of the Quality Control, available at http://imos.org.au/facility_manuals.html)

```
anfog_URL =
'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/ANFOG/seaglider/SOTS20110420/IMOS_ANFOG_B
CEOSTUV_20110420T111022Z_SG517_FV01_timeseries_END-20110420T140511Z.nc' ;
anfog_DATA = ncParse(anfog_URL) ;

qcLevel = 1 ; % we use the quality control flags to only select the good_data
psalData = anfog_DATA.variables.PSAL.data (anfog_DATA.variables.PSAL.flag == qcLevel) ;
timeData = anfog_DATA.dimensions.TIME.data (anfog_DATA.variables.PSAL.flag == qcLevel) ;
depthData = anfog_DATA.variables.DEPTH.data (anfog_DATA.variables.PSAL.flag == qcLevel) ;

% get the flag meaning values to add it later in the figure title
flag_meanings = textscan(anfog_DATA.variables.PSAL.flag_meanings,'%s','delimiter',' ');

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

[AX,H1,H2] = plotyy(timeData,psalData,timeData,depthData,'plot');% plot 2 functions in same fig

set(get(AX(1),'Ylabel'),'String',[strrep( anfog_DATA.variables.PSAL.standard_name,'_', ' ') ' in '
anfog_DATA.variables.PSAL.units])
set(get(AX(2),'Ylabel'),'String',[strrep( anfog_DATA.variables.DEPTH.standard_name,'_', ' ') ' in '
anfog_DATA.variables.DEPTH.units '-positive =' anfog_DATA.variables.DEPTH.positive])

datetick(AX(1),'x',0,'keeplimits','keepticks')
set(AX(2),'XTick',[])


xlabel(anfog_DATA.dimensions.TIME.standard_name)
title({anfog_DATA.metadata.title,['plot of ' strrep(flag_meanings{1}{qcLevel+1},'_',' ') '
only'] })

set(H1,'LineStyle','--')
set(H2,'LineStyle',':')
```

Variables to modify :

- anfog_URL : the opendap url of the chosen file
- qcLevel : quality control value (varies from 0 to 9)

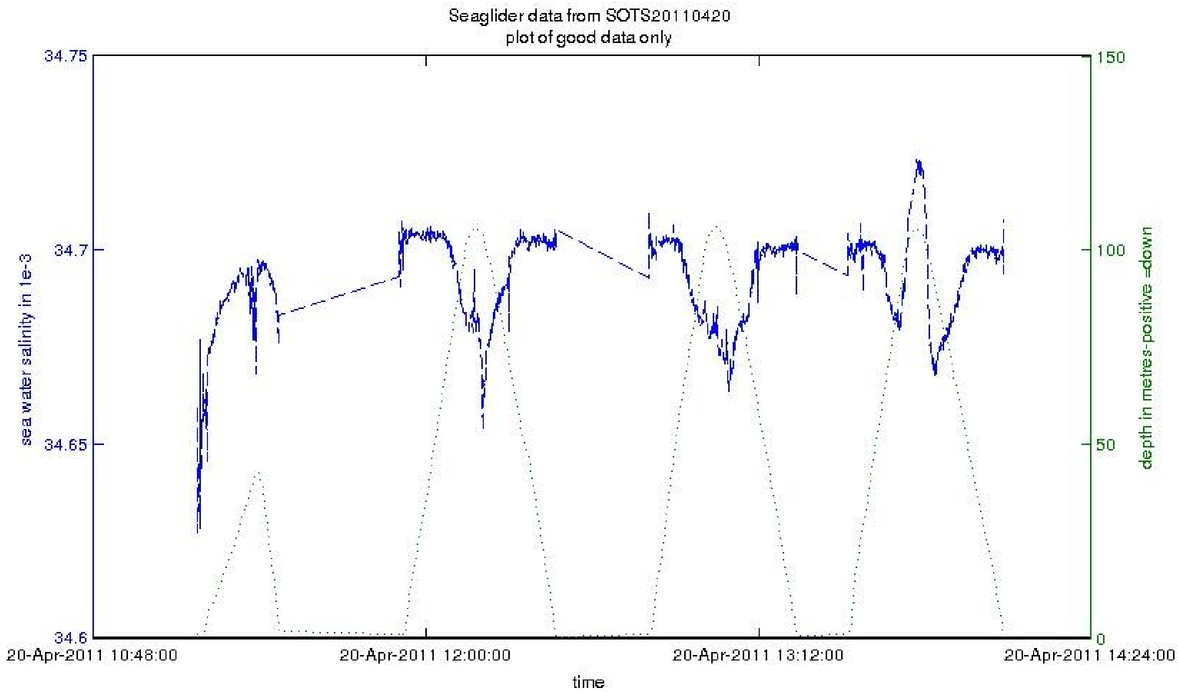

Illustration 4: Example of Sea Water Time-series taken during a SeaGlider Dive. Filtered to plot good data only

3.5 ANMN – National Mooring Network - QC'd good data

The Australian National Mooring Network Facility is a series of national reference stations and regional moorings designed to monitor particular oceanographic phenomena in Australian coastal ocean waters.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/ANMN/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot the U current variable measured with an ADCP instrument (in Western Australia).

```
anmn_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/ANMN/WA/WATR50/Velocity/IMOS_ANMN-WA_VATPE_20120516T040000Z_WATR50_FV01_WATR50-1205-Workhorse-ADCP-498_END-20121204T021500Z_C-20121207T023956Z.nc' ;
anmn_DATA = ncParse(anmn_URL) ;

%it is a lot more relevant for ADCP data to plot the good and probably good data only (flags 1 and 2).
qcLevel = [1 2];
qcIndex = anmn_DATA.variables.UCUR.flag == qcLevel(1) | anmn_DATA.variables.UCUR.flag == qcLevel(2) ;
uCurrentData =  anmn_DATA.variables.UCUR.data;
uCurrentData (~qcIndex) = NaN;

timeData = anmn_DATA.dimensions.TIME.data;
depthData = anmn_DATA.dimensions.HEIGHT_ABOVE_SENSOR.data;

% we create a matrix of similar size to be used afterwards with pcolor
[depthData_mesh,timeData_mesh] = meshgrid(depthData,timeData);

% get the flag meaning values to add it later in the figure title
flag_meanings = textscan(anmn_DATA.variables.UCUR.flag_meanings,'%s','delimiter',' ');

%% creation of a blue and red colormap centered in white
% initialise limits with RGB values
bluecolor = [0,0,1];% blue
redcolor =  [1,0,0];%white
whitecolor = [1,1,1];% red

% create each vector individually
maplength = 64; % number of color 'steps'
part1 = linspace(bluecolor(1),whitecolor(1),maplength/2);
part2 = linspace( whitecolor(1), redcolor(1),maplength/2);

part3 = linspace(bluecolor(2),whitecolor(2),maplength/2);
part4 = linspace( whitecolor(2), redcolor(2),maplength/2);

part5 = linspace(bluecolor(3),whitecolor(3),maplength/2);
part6 = linspace( whitecolor(3), redcolor(3),maplength/2);

% compose colormap
```

```
cmap_r_b = [horzcat(part1, part2)',horzcat(part3, part4)',horzcat(part5, part6)'];

%creation of the figure
figure1 = figure;
set(figure1, 'Renderer', 'painters') %to get rid of renderer bug with dateticks
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);
pcolor(timeData_mesh , depthData_mesh , uCurrentData)


shading flat
caxis([-max(max(abs(uCurrentData))) max(max(abs(uCurrentData)))]) % colorbar centered . we take the abs value
colormap(cmap_r_b)
cmap = colorbar;
set(get(cmap,'ylabel'),'string',strrep([anmn_DATA.variables.UCUR.long_name ' in '
anmn_DATA.variables.UCUR.units ],'_',' '),'Fontsize',10)
title({strrep([anmn_DATA.metadata.title ],'_',' ') , ['plot of ' strrep(flag_meanings{1}{qcLevel(1)+1},'_',' ')  ' and '
strrep(flag_meanings{1}{qcLevel(2)+1},'_',' ') ' only'] })
xlabel(anmn_DATA.dimensions.TIME.standard_name)
ylabel(strrep([anmn_DATA.dimensions.HEIGHT_ABOVE_SENSOR.long_name ' in '
anmn_DATA.dimensions.HEIGHT_ABOVE_SENSOR.units],'_',' '))

datetick('x',12)
```

Variables to modify :

- anmn_URL : the opendap url of the chosen file. (The example URL used here may not work if the file has been replaced by a newer version. A currently available file can be selected as described in section 1.2 above.)
- qcLevel : quality control value (varies from 0 to 9)

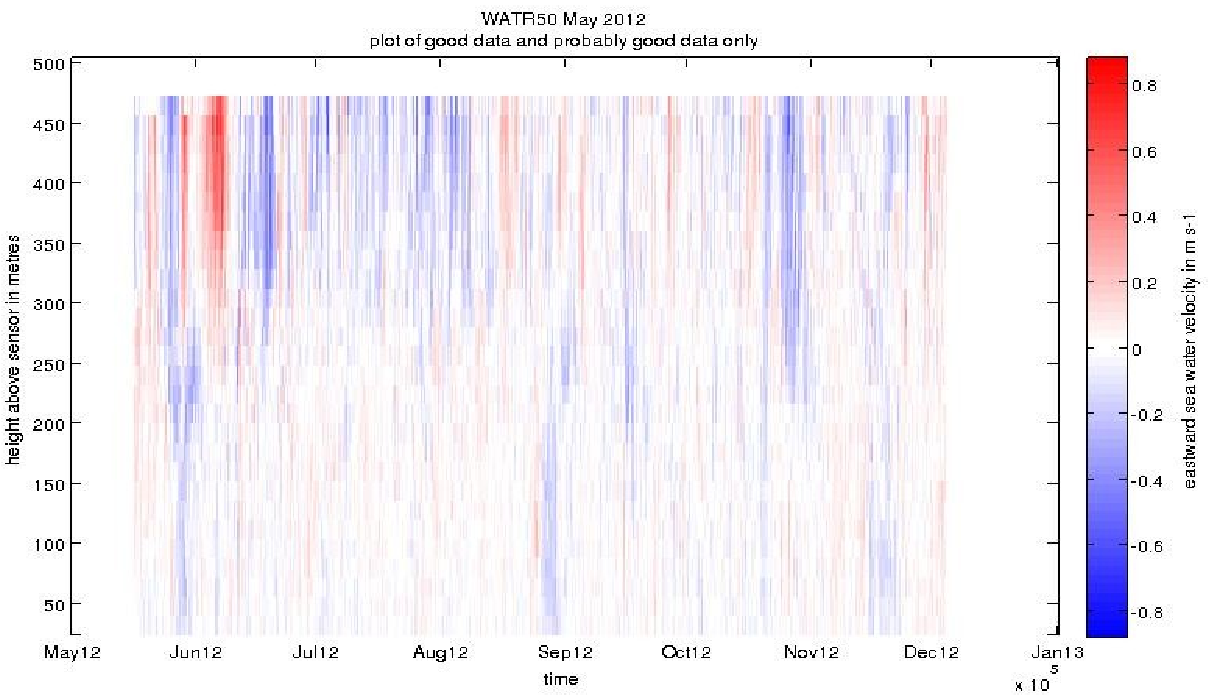

Illustration 5: Example of a Sea Water Velocity plot from ADCP data

3.6 AUV – Autonomous Underwater Vehicle - non QC'd data

The IMOS Autonomous Underwater Vehicle (AUV) Facility operates an ocean going AUV called Sirius capable of undertaking high resolution, geo-referenced survey work.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/AUV/catalog.html

In the example below, the ncParse function is used to extract depth, temperature, and time data and then produce a multiple time-series plot showing the variation of water temperature with depth and time during the robot's dive.

```
auv_URL =
'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/AUV/GBR201102/r20110301_012810_station119
5_09_transect/hydro_netcdf/IMOS_AUV_ST_20110301T012815Z_SIRIUS_FV00.nc' ;
auv_DATA = ncParse(auv_URL) ;

tempData = auv_DATA.variables.TEMP.data;
timeData = auv_DATA.dimensions.TIME.data;
depthData = auv_DATA.variables.DEPTH.data;
averageLat = mean(auv_DATA.variables.LATITUDE.data);
averageLon = mean(auv_DATA.variables.LONGITUDE.data);


figure1 = figure;
set(figure1, 'Position',  [1 500 1400 500 ], 'Color',[1 1 1]);

xlabel([strrep(auv_DATA.dimensions.(char(auv_DATA.variables.TEMP.dimensions)).long_name,'_', ' ')])
ylabel([strrep( auv_DATA.variables.TEMP.standard_name,'_', ' ') ' in '
auv_DATA.variables.TEMP.units])
datetick('x',15)

[AX,H1,H2] = plotyy (timeData,tempData,timeData,depthData);

set(get(AX(1),'Ylabel'),'String',[strrep( auv_DATA.variables.TEMP.standard_name,'_', ' ') ' in '
auv_DATA.variables.TEMP.units])
set(get(AX(2),'Ylabel'),'String',[strrep( auv_DATA.variables.DEPTH.standard_name,'_', ' ') ' in '
auv_DATA.variables.DEPTH.units '-positive =' auv_DATA.variables.DEPTH.positive])

datetick(AX(1),'x',31,'keeplimits','keepticks')
set(AX(2),'XTick',[])

xlabel(auv_DATA.dimensions.TIME.standard_name)
title({['campaign ' auv_DATA.metadata.title ],...
     ['location:lat=' num2str(averageLat) '; lon=' num2str(averageLon) ]})

set(H1,'LineStyle','--')
set(H2,'LineStyle',':')
```

Variables to modify :

- auv_URL : the opendap url of the chosen file

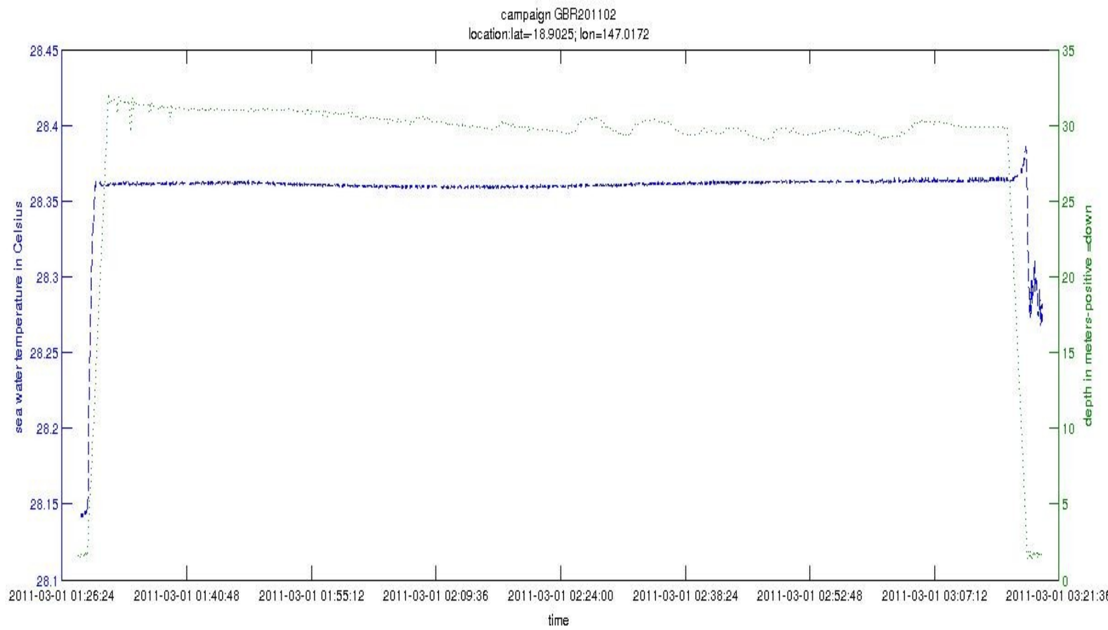

Illustration 6: Example of a Temperature Time-series plot during an AUV dive

3.7 Argo – Argo Floats Program - non QC'd data

Argo floats have revolutionised our understanding of the broad scale structure of the oceans to 2000 m depth. In the past 10 years more high resolution hydrographic profiles have been provided by Argo floats then from the rest of the observing system put together. Each Argo float is identified by a unique identification number called a WMO ID.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/Argo/aggregated_datasets/catalog.html

In the examples below, we demonstrate how to use the ncParse function to plot Argo data from an aggregated file (One file per year per basin : Atlantic, Indian, Pacific North, Pacific South). All the surface locations of all the Argo floats are plotted, as well as one temperature profile only.

3.7.1 Example 1

```
argo_URL =
'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/Argo/aggregated_datasets/south_pacific/IMOS_Argo_TPS-20020101T000000_FV01_yearly-aggregation-South_Pacific_C-20121102T220000Z.nc';
argo_DATA = ncParse(argo_URL) ;

nProfData = argo_DATA.dimensions.N_PROF.data; %Number of profiles contained in the file.
nLevelData = argo_DATA.dimensions.N_LEVELS.data;%Maximum number of pressure levels contained in a
profile.

% we choose a random profile number
profileNumber = 7;
% Casting data to double to be used afterwards with surface function
tempData = double(argo_DATA.variables.TEMP_ADJUSTED.data(profileNumber,:));
psalData = double(argo_DATA.variables.PSAL_ADJUSTED.data(profileNumber,:));
presData = double(argo_DATA.variables.PRES_ADJUSTED.data(profileNumber,:));
latProfile = argo_DATA.variables.LATITUDE.data(profileNumber);
lonProfile = argo_DATA.variables.LONGITUDE.data(profileNumber);
timeProfile = argo_DATA.variables.JULD.data(profileNumber);

latArgo = argo_DATA.variables.LATITUDE.data;
lonArgo = argo_DATA.variables.LONGITUDE.data;

% temperature profile
figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot (tempData,presData)
title({argo_DATA.metadata.description ,...
     datestr(timeProfile) ,...
     ['location:lat=' num2str(latProfile) '; lon=' num2str(lonProfile)],...
     ['Argo Float Number :' num2str(argo_DATA.variables.PLATFORM_NUMBER.data(profileNumber)) ]})
xlabel(strrep([argo_DATA.variables.TEMP_ADJUSTED.long_name ' in '
argo_DATA.variables.TEMP_ADJUSTED.units],'_', ' '))
```

```
ylabel(strrep([argo_DATA.variables.PRES_ADJUSTED.long_name ' in '
argo_DATA.variables.PRES_ADJUSTED.units],'_', ' '))

% salinity profile
figure2 = figure;
set(figure2, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot (psalData,presData)
title({argo_DATA.metadata.description ,...
     datestr(timeProfile) ,...
     ['location:lat=' num2str(latProfile) '; lon=' num2str(lonProfile)],...
     ['Argo Float Number :' num2str(argo_DATA.variables.PLATFORM_NUMBER.data(profileNumber)) ]})
xlabel(strrep([argo_DATA.variables.PSAL_ADJUSTED.long_name ' in '
argo_DATA.variables.PSAL_ADJUSTED.units],'_', ' '))
ylabel(strrep([argo_DATA.variables.PRES_ADJUSTED.long_name ' in '
argo_DATA.variables.PRES_ADJUSTED.units],'_', ' '))

% argo float trajectory
figure3 = figure;
set(figure3, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot(lonArgo,latArgo,'+')
xlabel(argo_DATA.variables.LONGITUDE.long_name)
ylabel(argo_DATA.variables.LATITUDE.long_name)
title('Argo Floats stations')
```

Variables to modify :

- argo_URL : the opendap url of the chosen file
- ProfileNumber: the number index of the profile to plot. This value can be changed by the user by restricting to a specific time/location.

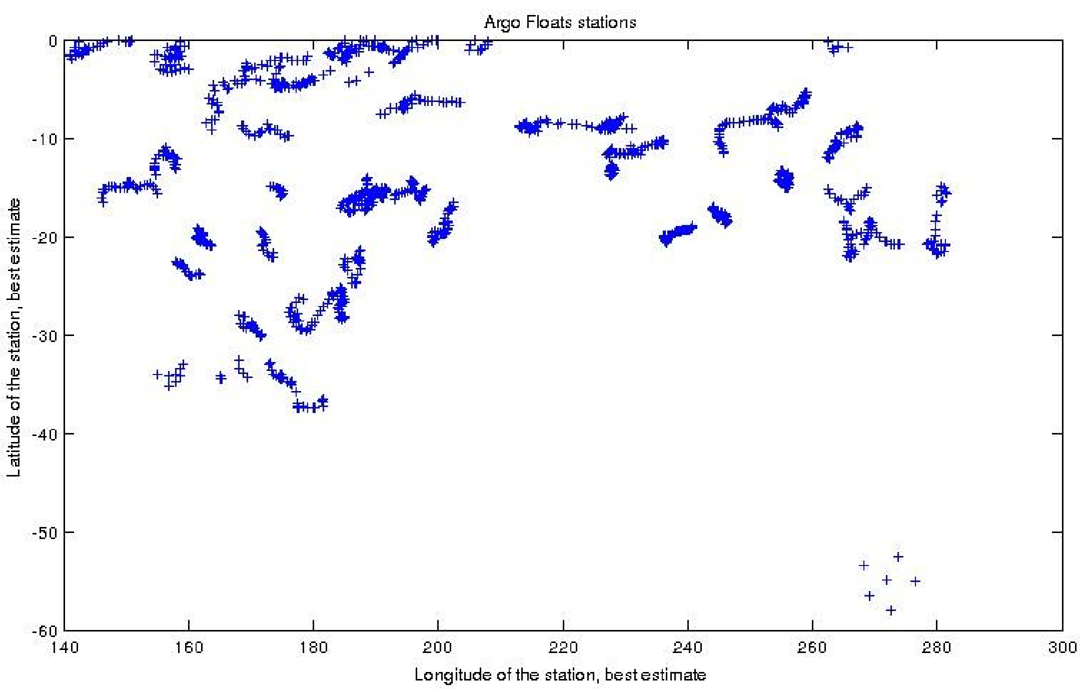

Illustration 7: Example of all Profile Locations of Argo Floats

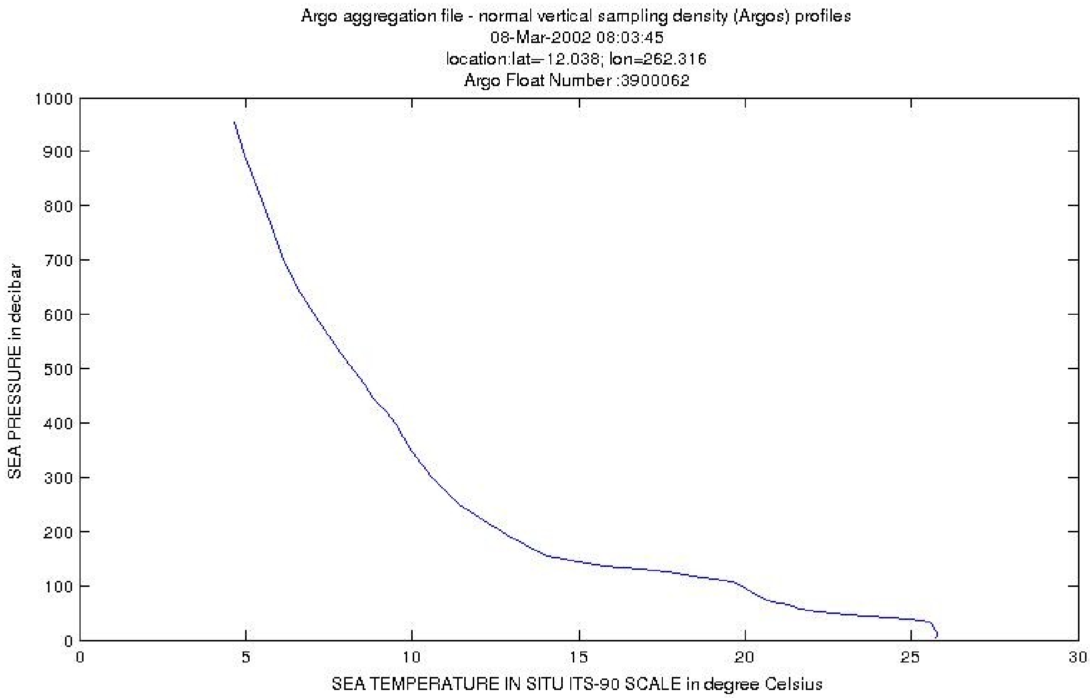

Illustration 8: Example of a Sea Water Temperature profile from an Argo float

3.7.2 Example 2

```
argo_URL =
'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/Argo/aggregated_datasets/south_pacific/IMOS_Argo_TPS-20020101T000000_FV01_yearly-aggregation-South_Pacific_C-20121102T220000Z.nc' ;
argo_DATA = ncParse(argo_URL) ;

nProfData = argo_DATA.dimensions.N_PROF.data; %Number of profiles contained in the file.
nLevelData = argo_DATA.dimensions.N_LEVELS.data;%Maximum number of pressure levels contained in a profile.

% we list all the Argo floats number in the variable 'argoFloatNumber' and
% chose one value
argoFloatNumber = unique(argo_DATA.variables.PLATFORM_NUMBER.data);
argoFloatNumberChosen = 5900106 ;% we randomly chose one float number;

% we load the data for this float. Casting data to double to be used afterwards with surface function
argoFloatProfilesIndexes = argo_DATA.variables.PLATFORM_NUMBER.data == argoFloatNumberChosen ;
tempData = double(argo_DATA.variables.TEMP_ADJUSTED.data(argoFloatProfilesIndexes,:));
psalData = double(argo_DATA.variables.PSAL_ADJUSTED.data(argoFloatProfilesIndexes,:));
presData = double(argo_DATA.variables.PRES_ADJUSTED.data(argoFloatProfilesIndexes,:));
latProfile = argo_DATA.variables.LATITUDE.data(argoFloatProfilesIndexes,:);
lonProfile = argo_DATA.variables.LONGITUDE.data(argoFloatProfilesIndexes,:);
timeProfile = argo_DATA.variables.JULD.data(argoFloatProfilesIndexes,:);

% creation of a time array which will be used by pcolor
[nline, ncol] = size(tempData);
sizer = ones(1, ncol);
CYCLE_NUMBER2D = double(argo_DATA.variables.CYCLE_NUMBER.data(argoFloatProfilesIndexes)) * sizer;
TIME_CYCLE_NUMBER2D = timeProfile * sizer;

figure1 = figure;
set(figure1, 'Position',  [1 1000 1100 900 ], 'Color',[1 1 1]);

%plot the argofloat TEMP timeseries
subplot(2,2,1:2),
pcolor(TIME_CYCLE_NUMBER2D, -presData, tempData);
datetick('x',20)
shading interp;
cmap = colorbar('location','EastOutside');
set(get(cmap,'ylabel'),'string',strrep([argo_DATA.variables.TEMP_ADJUSTED.long_name ' in '
argo_DATA.variables.TEMP_ADJUSTED.units ],'_',' '),'Fontsize',10)

title({argo_DATA.metadata.description ,...
     ['Argo Float Number :' num2str(argoFloatNumberChosen) ]})
xlabel('Time in DD/MM/YY')
ylabel(strrep([argo_DATA.variables.PRES_ADJUSTED.long_name ' in '
argo_DATA.variables.PRES_ADJUSTED.units],'_', ' '))

%plot the argofloat LAT timeseries
```

```
subplot(2,2,3),plot(TIME_CYCLE_NUMBER2D,latProfile)
title(strrep([argo_DATA.variables.LATITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([argo_DATA.variables.LATITUDE.long_name ' in '
argo_DATA.variables.LATITUDE.units],'_', ' '))
datetick('x',20)

%plot the argofloat LON timeseries
subplot(2,2,4),plot(TIME_CYCLE_NUMBER2D,lonProfile)
title(strrep([argo_DATA.variables.LONGITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([argo_DATA.variables.LONGITUDE.long_name ' in '
argo_DATA.variables.LONGITUDE.units],'_', ' '))
datetick('x',20)
set(figure1, 'Renderer', 'painters') %to get rid of renderer bug with dateticks
```

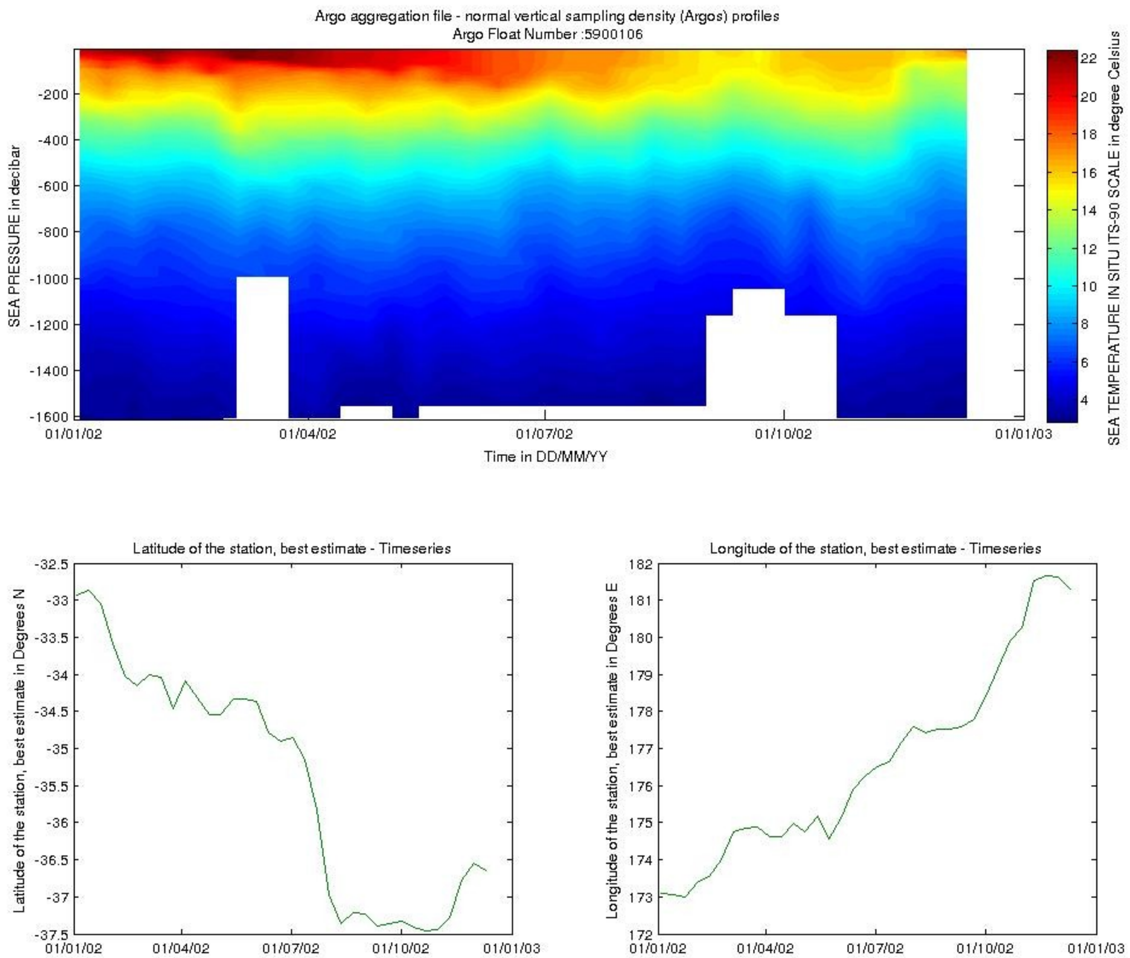

Illustration 9: Example of a Sea Water Temperature Time-series profile from an Argo float with its location over time

3.8 FAIMMS – Wireless Sensor Networks - QC'd good data

The IMOS Facility for Intelligent Monitoring of Marine Systems is a sensor network established in the Great Barrier Reef off the coast of Queensland, Australia. A 'sensor network' is an array of small, wirelessly interconnected sensors that collectively stream sensor data to a central data aggregation point. Sensor networks can be used to provide spatially dense bio-physical measurements in real-time.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/FAIMMS/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot a temperature time-series. Only data points which have a flag value equal to 1 are used (which means 'good data', please refers to IMOS NetCDF User Manual for a description of the Quality Control, available at http://imos.org.au/facility_manuals.html).

```
FAIMMS_URL = 
'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/FAIMMS/Myrmidon_Reef/Sensor_Float_1/water_temperature/sea_water_temperature@5.0m_channel_114/2012/QAQC/IMOS_FAIMMS_T_20121201T000000Z_FV01_END-20130101T000000Z_C-20130426T102459Z.nc' ;
faimms_DATA = ncParse(FAIMMS_URL) ;

qcLevel = 1; % only the Good data are being used
tempData = faimms_DATA.variables.TEMP.data (faimms_DATA.variables.TEMP.flag == qcLevel);
timeData = faimms_DATA.dimensions.TIME.data(faimms_DATA.variables.TEMP.flag == qcLevel);

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot (timeData,tempData)
title({faimms_DATA.metadata.title ,...
    [num2str(faimms_DATA.variables.TEMP.sensor_depth) ' m depth'] ,...
    ['location:lat=' num2str(faimms_DATA.dimensions.LATITUDE.data) '; lon=' 
num2str(faimms_DATA.dimensions.LONGITUDE.data) ]})
xlabel([strrep(faimms_DATA.dimensions.TIME.long_name,'_', ' ')])
ylabel([strrep( faimms_DATA.variables.TEMP.standard_name,'_', ' ') ' in ' 
faimms_DATA.variables.TEMP.units])
datetick('x',20)
```

Variables to modify :

- faimms_URL : the opendap url of the chosen file
- qcLevel : quality control value (varies from 0 to 9)

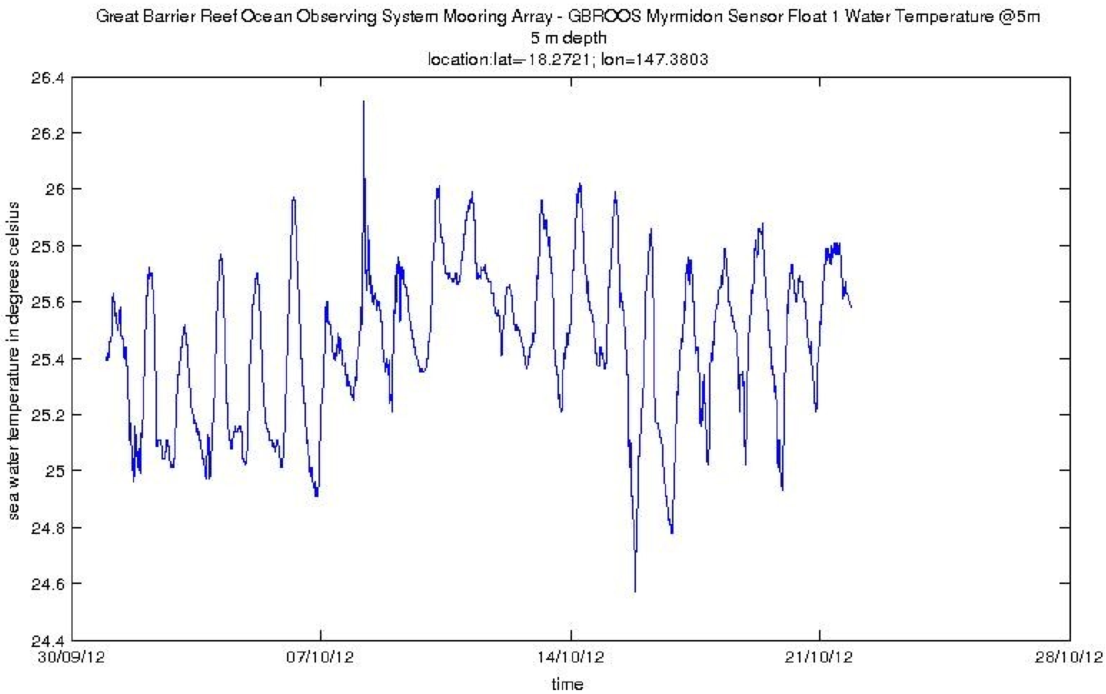

Illustration 10: Example of a Sea Water Temperature at 5m depth on the Great Barrier Reaf from FAIMMS data

3.9 SOOP – Ship Of Opportunities

3.9.1 XBT - expandable bathythermographs - QC'd data

IMOS Ship of Opportunity Underway Expandable Bathythermographs (XBT) group is a research and data collection project working within the IMOS Ship of Opportunity Multi-Disciplinary Underway Network sub-facility.

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/SOOP/SOOP-XBT/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot a XBT temperature profile.

```
xbt_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SOOP/SOOP-
XBT/aggregated_datasets/line_and_year/IX1/IMOS_SOOP-XBT_T_20040131T195300Z_IX1_FV01_END-
20041221T214400Z.nc';
xbt_DATA = ncParse(xbt_URL) ;

qcFlag = 4; % flag value to eliminate (bad data)

maxSample = length(xbt_DATA.dimensions.MAXZ.data); %
'maximum_number_of_samples_in_vertical_profile'
nProfiles = length(xbt_DATA.dimensions.INSTANCE.data); % number of profiles


%% we look for all the profiles of a similar cruise
cruiseData = xbt_DATA.variables.cruise_ID.data;
cruiseID = [];
for iiCruise = 1:length(cruiseData)
    cruiseID{iiCruise} = strrep(cruiseData(iiCruise,:),' ','');
end
uniqueCruiseIds = unique(cruiseID) ;
cruiseToPlot = uniqueCruiseIds{6}; %  'tb408504' , this is arbitrary. This value can be modified to
plot the cruise of choice
indexCruiseToPlot = strcmp(cruiseID , cruiseToPlot); % logical array

TEMP = xbt_DATA.variables.TEMP;
DEPTH = xbt_DATA.variables.DEPTH;
TIME = xbt_DATA.variables.TIME;

% we load the data for each cruise
timeCruise =  TIME.data(indexCruiseToPlot);
latCruise =  xbt_DATA.variables.LATITUDE.data(indexCruiseToPlot);
lonCruise =  xbt_DATA.variables.LONGITUDE.data(indexCruiseToPlot);

% we load only the data which does not have a quality control value equal to qcFlag (see above)
indexGoodData = xbt_DATA.variables.TEMP.flag(:,indexCruiseToPlot) ~= qcFlag;
tempCruise =  double(TEMP.data(:,indexCruiseToPlot));
depthCruise = double(DEPTH.data(:,indexCruiseToPlot));
```

```
% we modify the values which we don't want to plot to replace them with NaN
tempCruise(~indexGoodData) = NaN;
depthCruise(~indexGoodData) = NaN;

% creation of a profile array to use it with pcolor. same dimension of temp and depth
[nline, ncol] = size(tempCruise);
sizer = ones(nline,1) ;
profileIndex = 1:ncol;
prof_2D =  sizer * profileIndex ;

figure1 = figure;
set(figure1, 'Position',  [1 1000 1100 900 ], 'Color',[1 1 1]);

%plot the xbt TEMP timeseries
subplot(2,3,1:3),
pcolor(prof_2D, -depthCruise, tempCruise);
% datetick('x',20)
shading interp;
cmap = colorbar('location','EastOutside');
set(get(cmap,'ylabel'),'string',strrep([xbt_DATA.variables.TEMP.long_name ' in '
xbt_DATA.variables.TEMP.units ],'_',' '),'Fontsize',10)

title({xbt_DATA.metadata.title ,...
     ['Cruise :' char(cruiseToPlot) '-' xbt_DATA.metadata.XBT_line_description]})
xlabel('Profile Index')
ylabel(strrep([xbt_DATA.variables.DEPTH.long_name ' in ' xbt_DATA.variables.DEPTH.units],'_', ' '))

%plot the xbt LAT timeseries
subplot(2,3,4),plot(prof_2D,latCruise)
title(strrep([xbt_DATA.variables.LATITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([xbt_DATA.variables.LATITUDE.long_name ' in ' xbt_DATA.variables.LATITUDE.units],'_',
' '))
xlabel('Profile Index')

%plot the xbt LON timeseries
subplot(2,3,5),plot(prof_2D,lonCruise)
title(strrep([xbt_DATA.variables.LONGITUDE.long_name ' - Timeseries'],'_', ' '))
ylabel(strrep([xbt_DATA.variables.LONGITUDE.long_name ' in '
xbt_DATA.variables.LONGITUDE.units],'_', ' '))
xlabel('Profile Index')

%plot the xbt LON timeseries
subplot(2,3,6),plot(timeCruise,prof_2D)
title(strrep([xbt_DATA.variables.LONGITUDE.long_name ' - Timeseries'],'_', ' '))
xlabel([TIME.long_name   ' in dd/mm/yy'] )
ylabel('Profile Index')
datetick('x',20)
set(figure1, 'Renderer', 'painters') %to get rid of renderer bug with dateticks

% plot of a single profile
profileToPlot = 1 ; % this is arbitrary. We can plot all profiles from 1 to ncol, modify
```

```
profileToPlot if desired
figure2 = figure;
set(figure2, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);
plot (tempCruise(:,profileToPlot),-depthCruise(:,profileToPlot))
title({xbt_DATA.metadata.title ,...
    ['Cruise ' char(cruiseToPlot)] ,...
     xbt_DATA.metadata.XBT_line_description,...
     ['location:lat=' num2str(latCruise(profileToPlot)) '; lon='
num2str(lonCruise(profileToPlot))],...
     [datestr(timeCruise(profileToPlot)) ]})
xlabel(strrep([xbt_DATA.variables.TEMP.long_name ' in ' xbt_DATA.variables.TEMP.units],'_', ' '))
ylabel(strrep([xbt_DATA.variables.DEPTH.long_name ' in negative '
xbt_DATA.variables.DEPTH.units],'_', ' '))
```

Variables to modify :

- xbt_URL : the opendap url of the chosen file
- ProfileToPlot : the profile number attached to the cruise id to plot
- cruiseToPlot : the cruise id number or line to plot

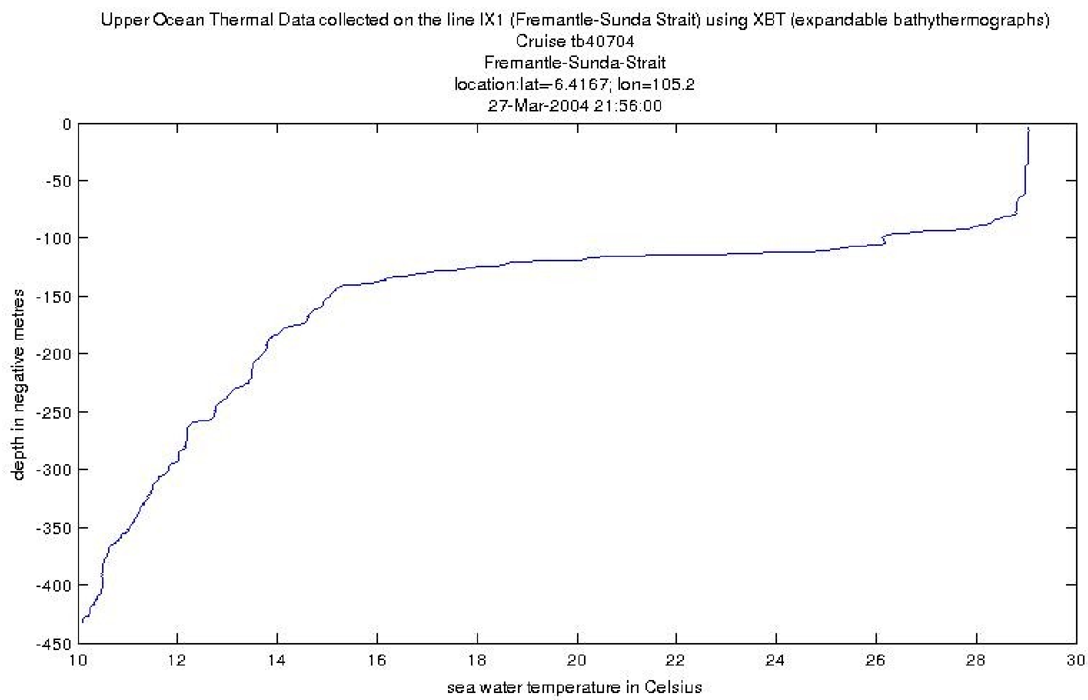

Illustration 11: Example of Sea Water Temperature Profile from XBT data

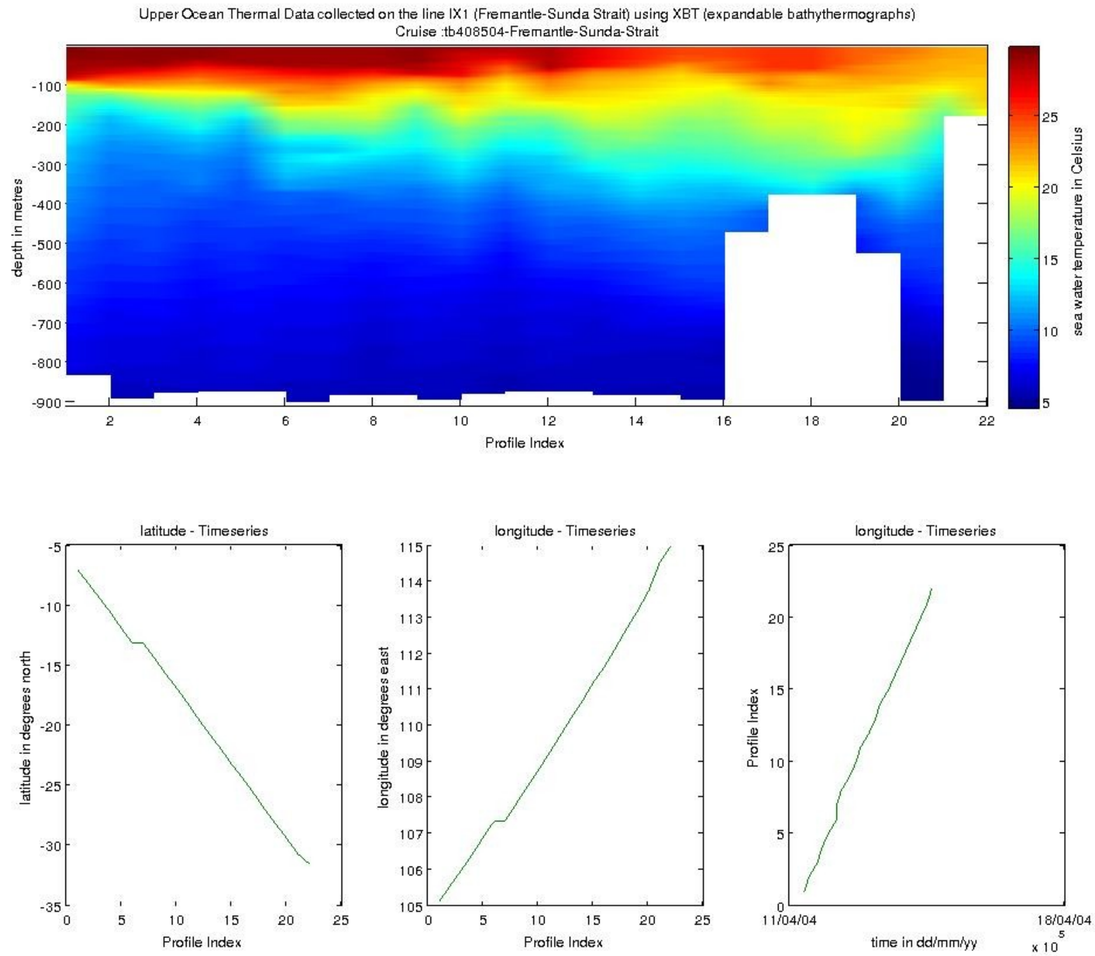

Illustration 12: Example of Sea Water Temperature Time-series Profile from XBT data with the profiles' location

3.10 SRS – Satellite Remote Sensing

3.10.1 Bio-Optical database – Pigment data

The bio-optical data base underpins the assessment of ocean colour products in the Australian region (e.g. chlorophyll a concentrations, phytoplankton species composition and primary production).

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/SRS/BioOptical/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot a Chlorophyll-a profile (High Performance Liquid Chromatography of pigments in discrete sea-water samples)

```
srs_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SRS/BioOptical/1997_cruise-
FR1097/pigment/IMOS_SRS-OC-BODBAW_X_19971201T052600Z_FR1097-pigment_END-19971207T220700Z_C-
20121129T120000Z.nc' ;
srs_DATA = ncParse(srs_URL) ;

nProfiles = length (srs_DATA.dimensions.profile.data);% number of profiles

% we choose the first profile
ProfileToPlot = 10; % this is arbitrary. We can plot all profiles from 1 to nProfiles
nObsProfile = srs_DATA.variables.rowSize.data(ProfileToPlot);  %number of observations for
ProfileToPlot
timeProfile = srs_DATA.variables.TIME.data(ProfileToPlot);
latProfile = srs_DATA.variables.LATITUDE.data(ProfileToPlot);
lonProfile = srs_DATA.variables.LONGITUDE.data(ProfileToPlot);

% we look for the observations indexes related to the chosen profile
indexObservationStart = sum( srs_DATA.variables.rowSize.data(1:ProfileToPlot)) -
srs_DATA.variables.rowSize.data(ProfileToPlot) +1;
indexObservationEnd = sum( srs_DATA.variables.rowSize.data(1:ProfileToPlot));
indexObservation =  indexObservationStart:indexObservationEnd ;

% we chose arbitrary to plot CPHL_a but there are many more variables
% available
cphl_aData = srs_DATA.variables.CPHL_a.data(indexObservation);  %for ProfileToPlot
depthData = srs_DATA.variables.DEPTH.data(indexObservation);

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

plot (cphl_aData,depthData)
title({srs_DATA.metadata.source ,...
    datestr(timeProfile),...
    ['location:lat=' num2str(latProfile) '; lon=' num2str(lonProfile) ]})
xlabel([strrep(srs_DATA.variables.CPHL_a.long_name,'_', ' ') ' in '
srs_DATA.variables.CPHL_a.units])
ylabel([strrep(srs_DATA.variables.DEPTH.long_name,'_', ' ') ' in ' srs_DATA.variables.DEPTH.units
```

```
';positive ' srs_DATA.variables.DEPTH.positive ])
```

Variables to modify :

- srs_URL : the opendap url of the chosen file
- ProfileToPlot : the profile number to plot

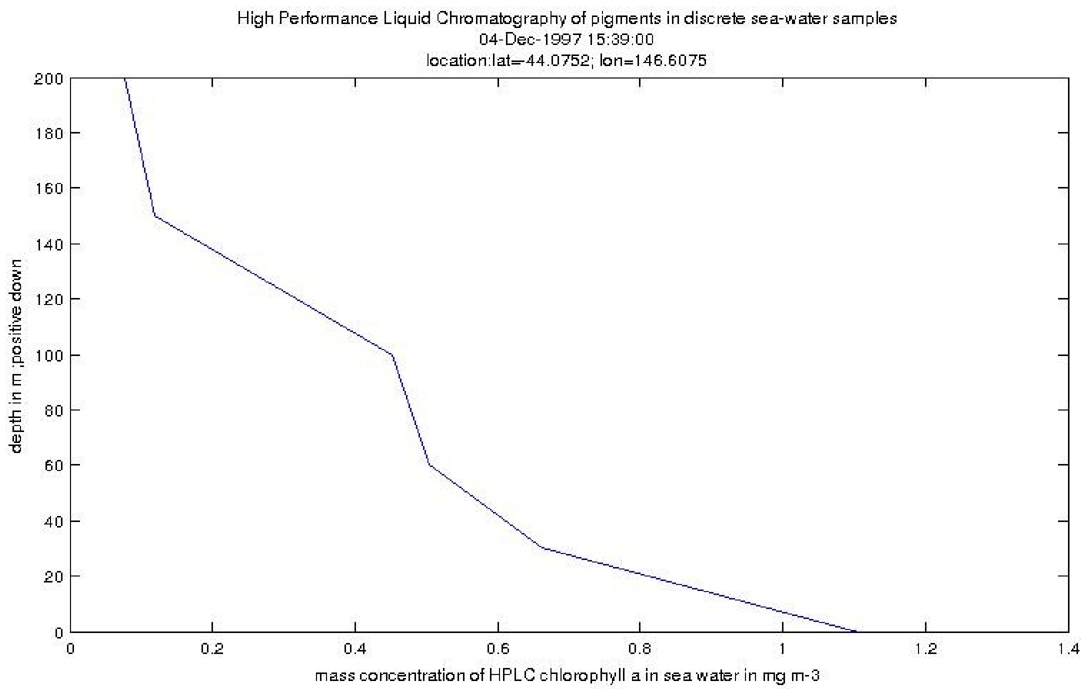

Illustration 13: Example of Pigment Data Profile from the BioOptical database dataset

3.10.2 Bio-Optical database – Absorption data

The bio-optical data base underpins the assessment of ocean colour products in the Australian region (e.g. chlorophyll a concentrations, phytoplankton species composition and primary production).

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/catalog/IMOS/SRS/BioOptical/catalog.html

In the example below, we demonstrate how to use the ncParse function to plot (1) the variation of Absorption coefficients of CDOM (gilvin) in discrete sea-water samples at different wavelengths and (2) the variation of absorption coefficients of CDOM at different wavelengths and different depths.

```
srs_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SRS/BioOptical/1997_cruise-FR1097/absorption/IMOS_SRS-OC-BODBAW_X_19971201T052600Z_FR1097-absorption-CDOM_END-19971207T180500Z_C-20121129T130000Z.nc' ;
srs_DATA = ncParse(srs_URL) ;

nProfiles = length (srs_DATA.dimensions.profile.data);% number of profiles

% we choose the first profile
ProfileToPlot = 10; % this is arbitrary. We can plot all profiles from 1 to nProfiles
nObsProfile = srs_DATA.variables.rowSize.data(ProfileToPlot);  %number of observations for ProfileToPlot
timeProfile = srs_DATA.variables.TIME.data(ProfileToPlot);
latProfile = srs_DATA.variables.LATITUDE.data(ProfileToPlot);
lonProfile = srs_DATA.variables.LONGITUDE.data(ProfileToPlot);

% we look for the observations indexes related to the chosen profile
indexObservationStart = sum( srs_DATA.variables.rowSize.data(1:ProfileToPlot)) - srs_DATA.variables.rowSize.data(ProfileToPlot) +1;
indexObservationEnd = sum( srs_DATA.variables.rowSize.data(1:ProfileToPlot));
indexObservation =  indexObservationStart:indexObservationEnd ;

agData = double(srs_DATA.variables.ag.data(indexObservation,:));
wavelengthData = double(srs_DATA.dimensions.wavelength.data);
depthData = double(srs_DATA.variables.DEPTH.data(indexObservation));

 % we create a matrix of similar size to be used afterwards with pcolor
[wavelengthData_mesh,depthData_mesh] = meshgrid(wavelengthData,depthData);

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);
pcolor(wavelengthData_mesh , depthData_mesh , agData)

shading flat
caxis([min(min(agData)) max(max(agData))])
cmap = colorbar;
set(get(cmap,'ylabel'),'string',strrep([srs_DATA.variables.ag.long_name ' in ' srs_DATA.variables.ag.units ],'_',' '),'Fontsize',10)
title(strrep([srs_DATA.metadata.source ],'_',' '))
```

```
xlabel( strrep([srs_DATA.dimensions.wavelength.long_name ' in: ',
srs_DATA.dimensions.wavelength.units],'_', ' '))
ylabel(strrep([srs_DATA.variables.DEPTH.long_name ' in ' srs_DATA.variables.DEPTH.units '; positive
' srs_DATA.variables.DEPTH.positive ],'_',' '))

%%%%%%%%%%%%%5
nDepth = length(depthData);
figure2 = figure;
set(figure2, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);
plot(wavelengthData,agData,'x')
unitsMainVar=char(srs_DATA.variables.ag.units);
ylabel( strrep([srs_DATA.variables.ag.long_name ' in: ', srs_DATA.variables.ag.units],'_', ' '))
xlabel( strrep([srs_DATA.dimensions.wavelength.long_name ' in: ',
srs_DATA.dimensions.wavelength.units],'_', ' '))

title({strrep(srs_DATA.variables.ag.long_name,'_',' '),...
    strcat('in units:',srs_DATA.variables.ag.units),...
    strcat('station :',char(srs_DATA.variables.station_name.data(ProfileToPlot,:)),...
    '- location',num2str(latProfile,'%2.3f'),'/',num2str(lonProfile,'%3.2f') ),...
    strcat('time :',datestr(timeProfile))
    })

for iiDepth=1:nDepth
    legendDepthString{iiDepth}=strcat('Depth:',num2str(depthData(iiDepth)),'m');
end
legend(legendDepthString)
```

Variables to modify :

- srs_URL : the opendap url of the chosen file
- ProfileToPlot : the profile number to plot

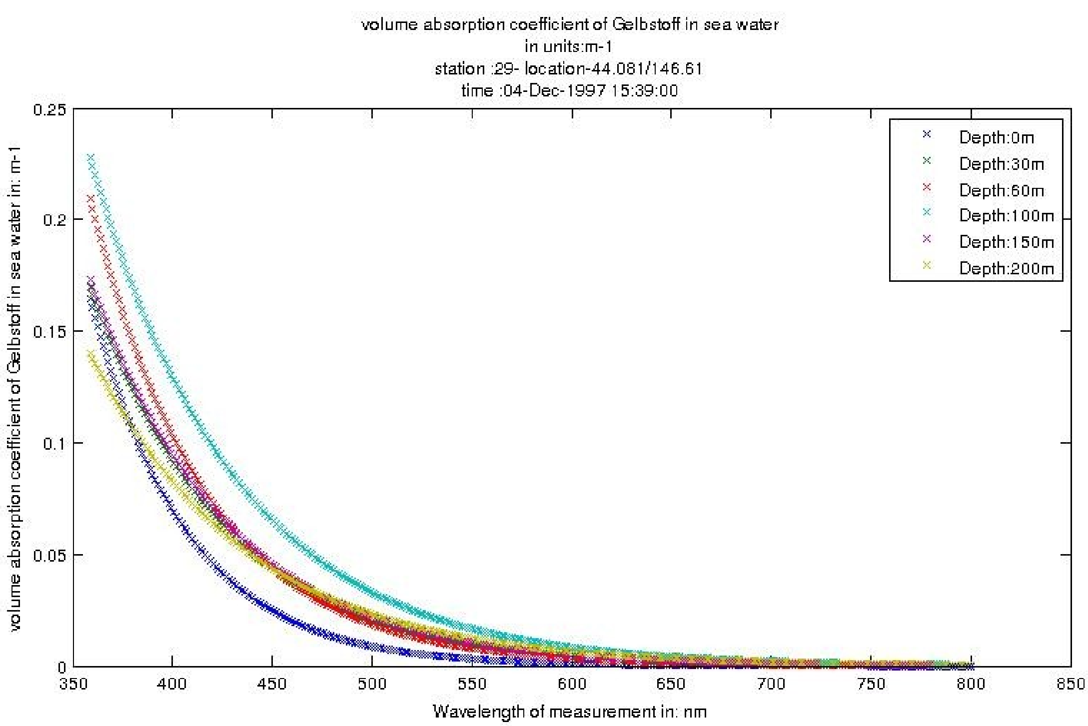

Illustration 14: Example of Absorption Data at different depth from the BioOptical database dataset

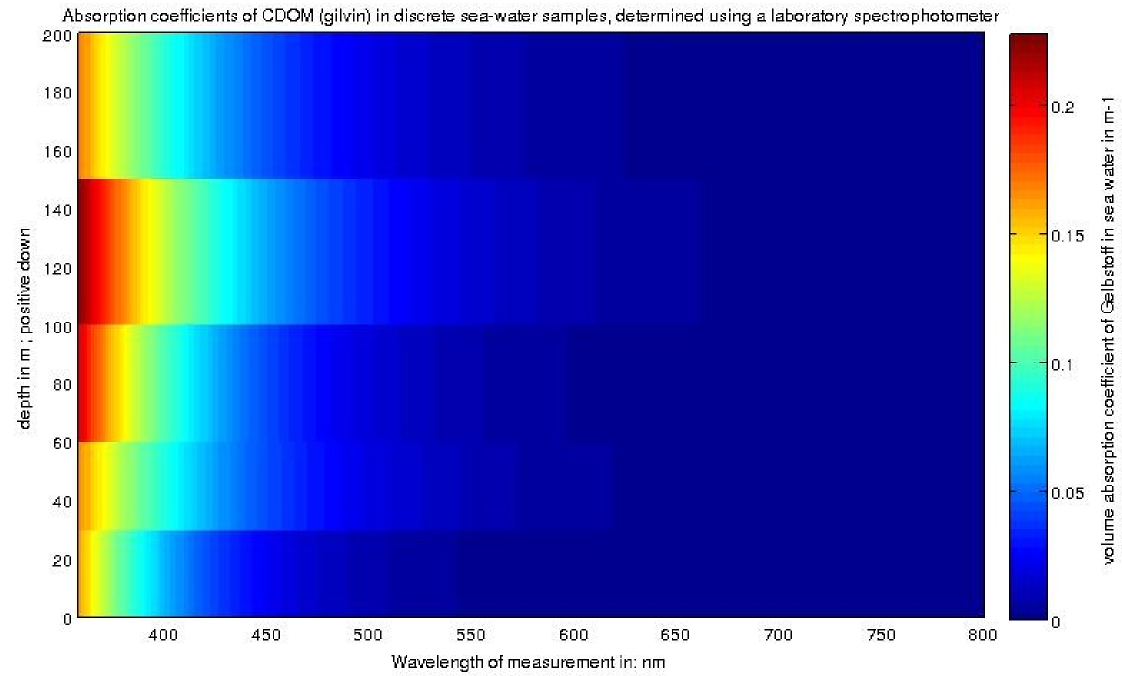

Illustration 15: Example of Absorption Data plot from the BioOptical database dataset

3.10.3 GHRSST – L3P mosaic

Please refer to the SRS product Help page : http://portalhelp.aodn.org.au/Portal2_help/?q=node/149

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/dodsC/IMOS/SRS/GHRSST-SSTsubskin/

In the example below, we demonstrate how to use the ncParse function to plot the Sea Surface Temperature from a gridded data product.

```
srs_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SRS/SRS-SST/L3P/2013/20130315-
ABOM-L3P_GHRSST-SSTsubskin-AVHRR_MOSAIC_01km-AO_DAAC-v01-fv01_0.nc' ;
srsL3P_DATA = ncParse(srs_URL) ;

step = 20; % we take one point out of 'step'. Only to make it faster to plot on Matlab
% squeeze the data to get rid of the time dimension in the variable shape
sst = squeeze(srsL3P_DATA.variables.sea_surface_temperature.data(1,1:step:end,1:step:end));
lat = squeeze(srsL3P_DATA.dimensions.lat.data(1:step:end));
lon = squeeze(srsL3P_DATA.dimensions.lon.data(1:step:end));

[lon_mesh,lat_mesh] = meshgrid(lon,lat);% we create a matrix of similar size to be used afterwards
with pcolor

figure1 = figure;
surface(lon_mesh ,lat_mesh , sst)
shading flat
caxis([min(min(sst)) max(max(sst))])
cmap = colorbar;
set(get(cmap,'ylabel'),'string',[srsL3P_DATA.variables.sea_surface_temperature.long_name ' in '
srsL3P_DATA.variables.sea_surface_temperature.units ],'Fontsize',10)
title([srsL3P_DATA.metadata.title '-' srsL3P_DATA.metadata.start_date ])
xlabel(strrep(([srsL3P_DATA.dimensions.lon.long_name ' in '
srsL3P_DATA.dimensions.lon.units]),'_',' '))
ylabel(strrep(([srsL3P_DATA.dimensions.lat.long_name ' in '
srsL3P_DATA.dimensions.lat.units]),'_',' '))
```

Variables to modify :

- srs_URL : the opendap url of the chosen file
- step : a number to lower the resolution. This helps to reduce memory issues.

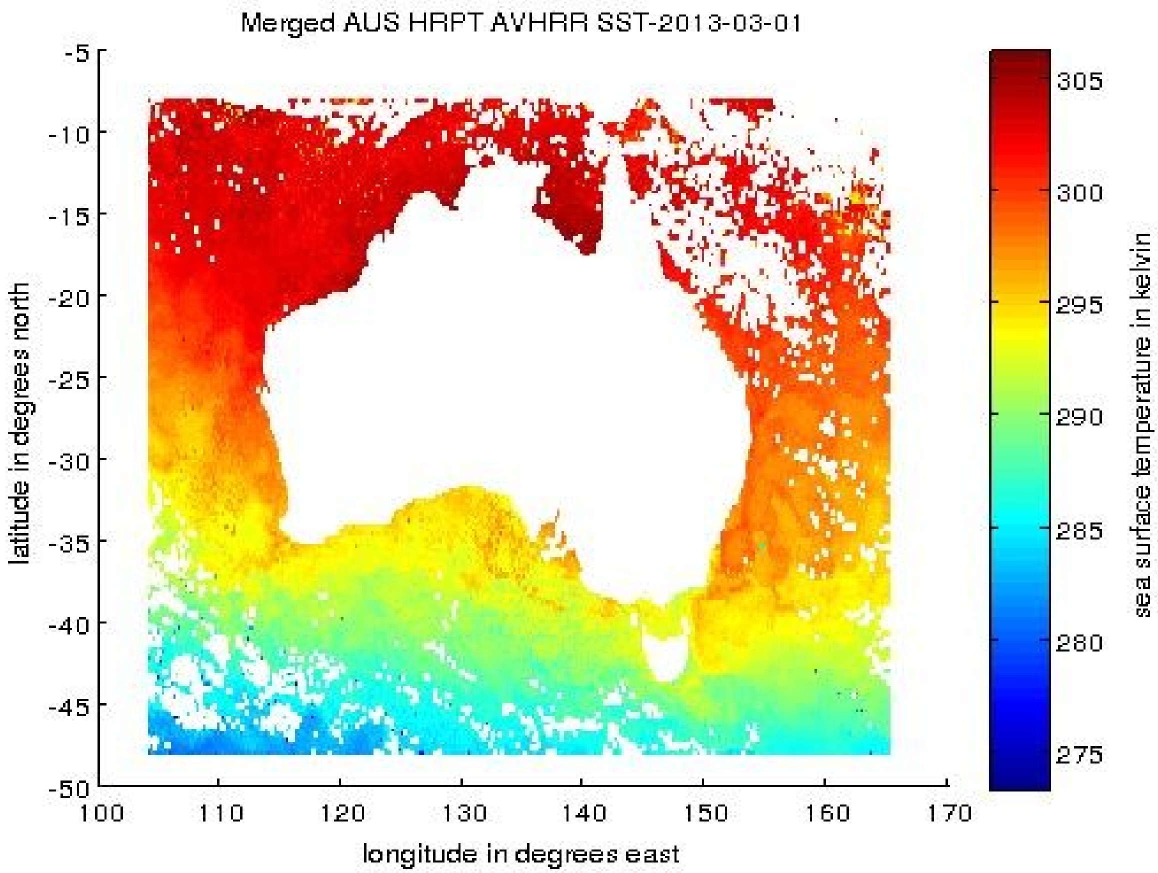

Illustration 16: Example of Sea Surface Temperature plot from a L3P product

3.10.4 GHRSST – L3C – multi swath, single sensor

Please refer to the SRS product Help page : http://portalhelp.aodn.org.au/Portal2_help/?q=node/149

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/dodsC/IMOS/SRS/SRS-SST/L3C-01day/

In the example below, we demonstrate how to use the ncParse function to plot the Sea Surface Temperature from a gridded data product.

```
srs_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SRS/SRS-SST/L3C-01day/L3C_GHRSST-SSTskin-AVHRR19_D-1d_night/2013/20130401152000-ABOM-L3C_GHRSST-SSTskin-AVHRR19_D-1d_night-v02.0-fv01.0.nc.gz' ;
srsL3C_DATA = ncParse(srs_URL) ;

step = 10; % we take one point out of 'step'. Only to make it faster to plot on Matlab
% squeeze the data to get rid of the time dimension in the variable shape
sst = squeeze(srsL3C_DATA.variables.sea_surface_temperature.data(1,1:step:end,1:step:end));
lat = squeeze(srsL3C_DATA.dimensions.lat.data(1:step:end));
% modify the longitude values which across the 180th meridian
lon = squeeze(srsL3C_DATA.dimensions.lon.data(1:step:end));
if sum(lon<0) > 0
    lon(lon<0) =  lon(lon<0)+360;
end

[lon_mesh,lat_mesh] = meshgrid(lon,lat);% we create a matrix of similar size to be used afterwards with pcolor

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

surface(lon_mesh , lat_mesh , sst)
shading flat
caxis([min(min(sst)) max(max(sst))])
cmap = colorbar;
set(get(cmap,'ylabel'),'string',[srsL3C_DATA.variables.sea_surface_temperature.long_name ' in ' srsL3C_DATA.variables.sea_surface_temperature.units ],'Fontsize',10)
title({srsL3C_DATA.metadata.title ,...
    srsL3C_DATA.metadata.start_time })
xlabel(strrep(([srsL3C_DATA.dimensions.lon.long_name ' in ' srsL3C_DATA.dimensions.lon.units]),'_',' '))
ylabel(strrep(([srsL3C_DATA.dimensions.lat.long_name ' in ' srsL3C_DATA.dimensions.lat.units]),'_',' '))
```

Variables to modify :

- srs_URL : the opendap url of the chosen file
- step : a number to lower the resolution. This helps to reduce memory issues.

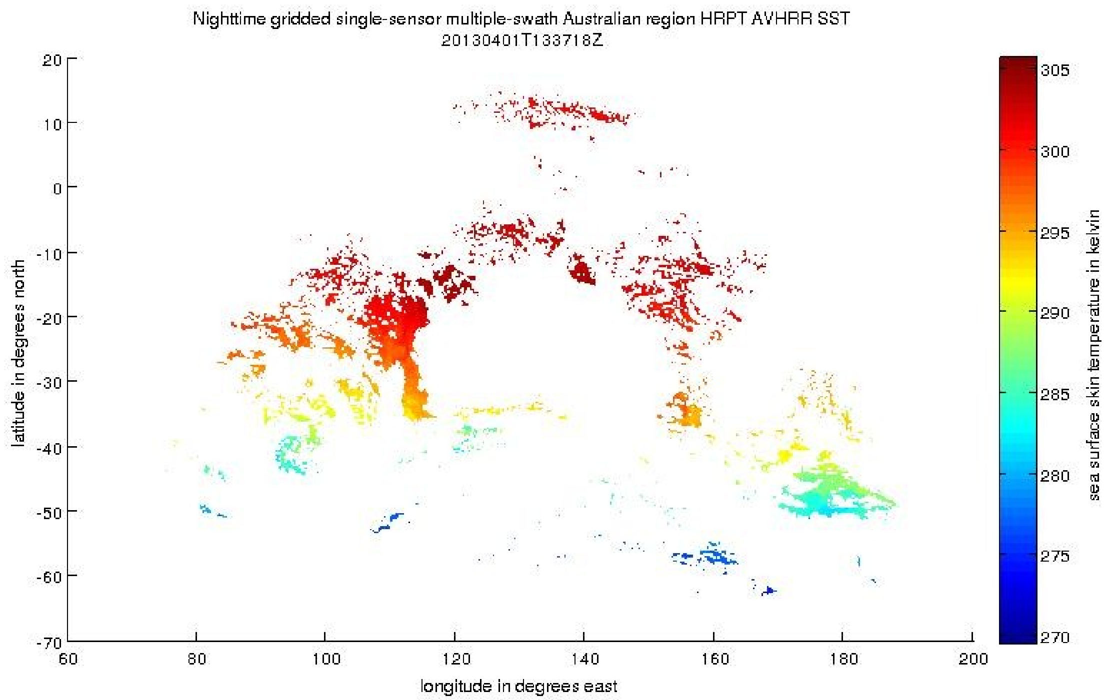

Illustration 17: Example of Sea Surface Temperature plot from a L3Cproduct at night

3.10.5 GHRSST – L3S – multi swath, multi sensor, one day

Please refer to the SRS product Help page : http://portalhelp.aodn.org.au/Portal2_help/?q=node/149

NetCDF files can be found at :
http://thredds.aodn.org.au/thredds/dodsC/IMOS/SRS/SRS-SST/L3S-01day/

In the example below, we demonstrate how to use the ncParse function to plot the Sea Surface Temperature from a gridded data product.

```
srs_URL = 'http://thredds.aodn.org.au/thredds/dodsC/IMOS/eMII/demos/SRS/SRS-SST/L3S-01day/L3S_1d_night/2013/20130401152000-ABOM-L3S_GHRSST-SSTskin-AVHRR_D-1d_night-v02.0-fv01.0.nc.gz';
srsL3S_DATA = ncParse(srs_URL) ;

step = 10; % we take one point out of 'step'. Only to make it faster to plot on Matlab
% squeeze the data to get rid of the time dimension in the variable shape
sst = squeeze(srsL3S_DATA.variables.sea_surface_temperature.data(1,1:step:end,1:step:end));
lat = squeeze(srsL3S_DATA.dimensions.lat.data(1:step:end));
% modify the longitude values which across the 180th meridian
lon = squeeze(srsL3S_DATA.dimensions.lon.data(1:step:end));
if sum(lon<0) > 0
    lon(lon<0) =  lon(lon<0)+360;
end

[lon_mesh,lat_mesh] = meshgrid(lon,lat);% we create a matrix of similar size to be used afterwards with pcolor

figure1 = figure;
set(figure1, 'Position',  [1 500 900 500 ], 'Color',[1 1 1]);

surface(lon_mesh , lat_mesh , sst)
shading flat
caxis([min(min(sst)) max(max(sst))])
cmap = colorbar;
set(get(cmap,'ylabel'),'string',[srsL3S_DATA.variables.sea_surface_temperature.long_name ' in ' srsL3S_DATA.variables.sea_surface_temperature.units ],'Fontsize',10)
title({srsL3S_DATA.metadata.title ,...
    srsL3S_DATA.metadata.start_time })
xlabel(strrep(([srsL3S_DATA.dimensions.lon.long_name ' in ' srsL3S_DATA.dimensions.lon.units]),'_',' '))
ylabel(strrep(([srsL3S_DATA.dimensions.lat.long_name ' in ' srsL3S_DATA.dimensions.lat.units]),'_',' '))
```

Variables to modify :

- srs_URL : the opendap url of the chosen file
- step : a number to lower the resolution. This helps to reduce memory issues.

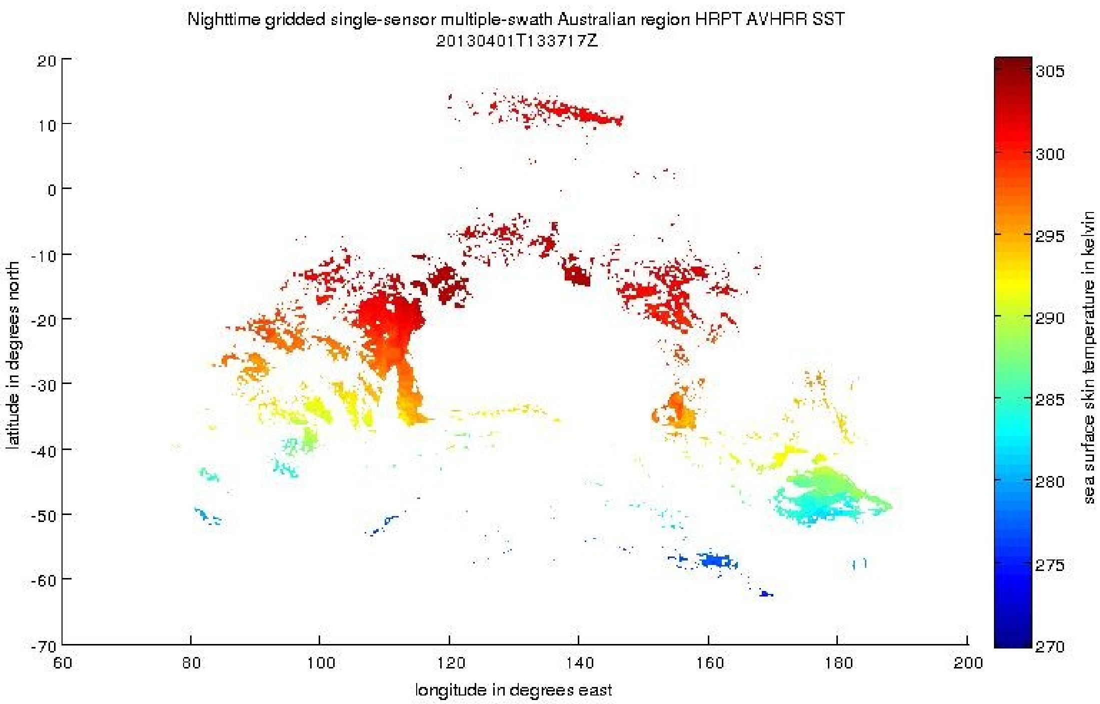

Illustration 18: Example of Sea Surface Temperature plot from a L3S product at night