



[marine.copernicus.eu](http://marine.copernicus.eu)



# CMEMS User & Training Workshop – IBI region

## In Situ TAC

Fernando Manzano, Charles Troupin

Puertos del Estado, SOCIB

Lisboa, 11 December 2015

1. Ocean Data View

2. Python

## 1. Ocean Data View

1.1– Objective 1: time series

1.2– Objective 2: CORA dataset

## 2. Python

2.1– ipython notebooks

2.2– Example 1: plotting

# Ocean Data View



> 40000 registered users

Data analysis + visualisation

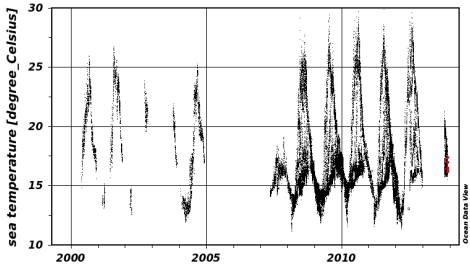
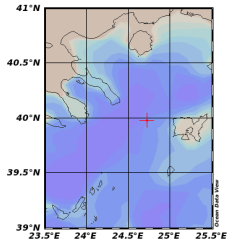
Almost every format supported

# Working with ODV on Time Series

# Objective: plotting time series



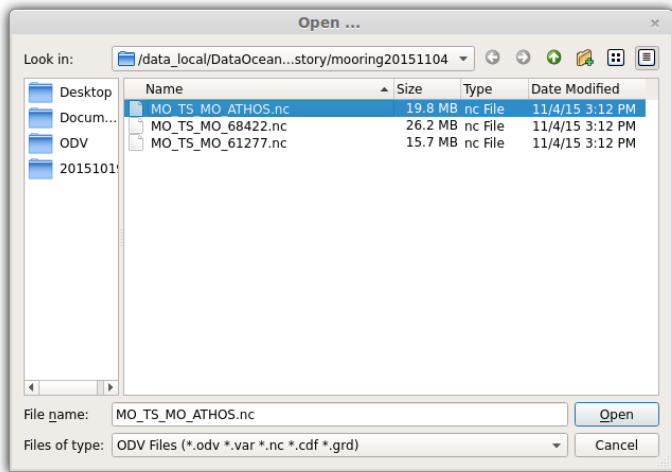
## Temperature at mooring Athos



# Opening the netCDF file



File → Open → netcdf



## Dimension and variables : Next

**NetCDF Setup Wizard**

Select Dimensions (Step 1 of 4)

---

**NetCDF dimensions**

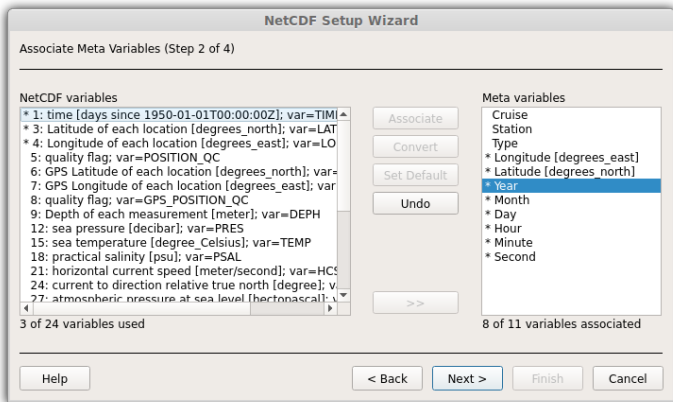
- TIME[36552]
- LATITUDE[36552]
- LONGITUDE[36552]
- POSITION[36552]
- DEPTH[6]

All 5 dimensions selected

**Corresponding netCDF variables**

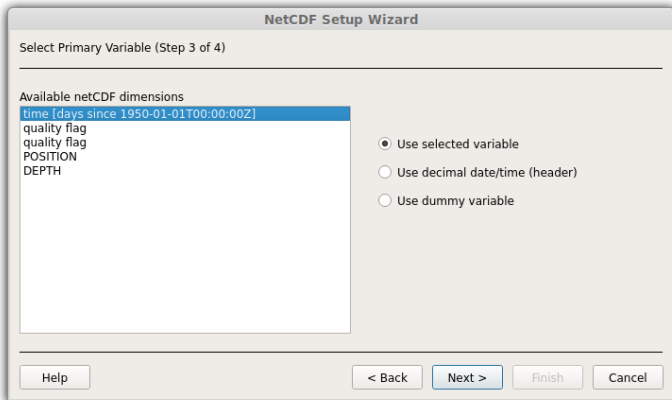
- time [days since 1950-01-01T00:00:00Z]; var=TIME
- Latitude of each location [degrees\_north]; var=LATITUDE
- Longitude of each location [degrees\_east]; var=LONGITUDE
- quality flag; var=POSITION\_QC
- GPS Latitude of each location [degrees\_north]; var=GPS\_LAT
- GPS Longitude of each location [degrees\_east]; var=GPS\_LON
- quality flag; var=GPS\_POSITION\_QC
- Depth of each measurement [meter]; var=DEPTH
- sea pressure [decibar]; var=PRES
- sea temperature [degree\_Celsius]; var=TEMP
- practical salinity [psu]; var=PSAL
- horizontal current speed [meter/second]; var=HCSP
- current to direction relative true north [degree]; var=HCDD
- atmospheric pressure at sea level [hectopascal]; var=ATMS
- air temperature in dry bulb [degree\_Celsius]; var=DRYT

## Variable association : Next

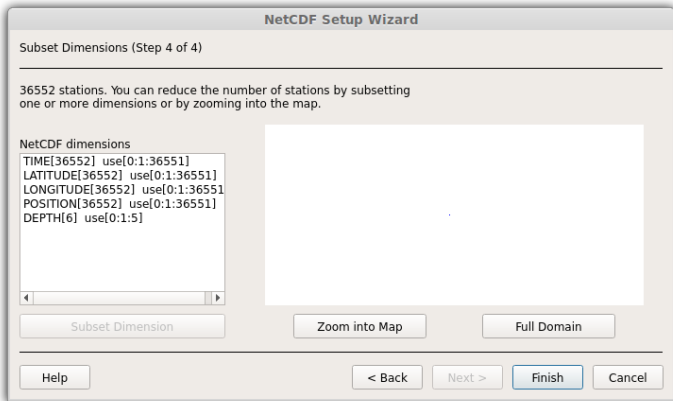


 Quality Control variables not visible at this stage

## Primary variables : Next



## Subset dimensions : Finish





# Plot the time series



View → Layout template → Scatter window

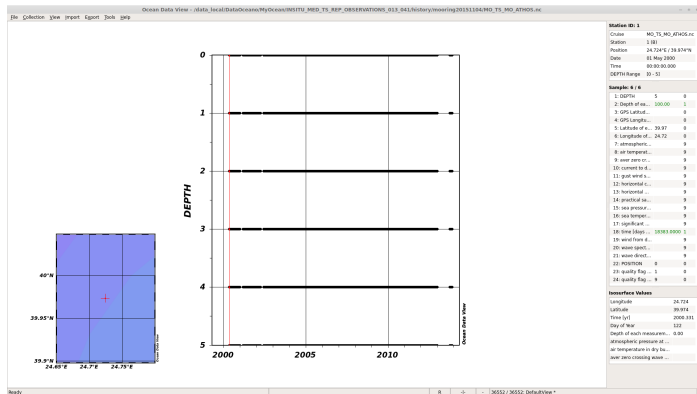
The screenshot shows the Ocean Data View software interface. The 'View' menu is open, showing options like 'Full Screen Map', '1 STATION Window', '2 STATION Windows', '3 STATION Windows', '4 STATION Windows', '5 STATION Windows', '6 STATION Windows', '7 STATION Windows', '8 STATION Windows', '9 STATION Windows', '10 STATION Windows', '11 STATION Windows', '12 STATION Windows', '13 STATION Windows', '14 STATION Windows', '15 STATION Windows', '16 STATION Windows', '17 STATION Windows', '18 STATION Windows', '19 STATION Windows', '20 STATION Windows', '21 STATION Windows', '22 STATION Windows', '23 STATION Windows', '24 STATION Windows', '25 STATION Windows', '26 STATION Windows', '27 STATION Windows', '28 STATION Windows', '29 STATION Windows', '30 STATION Windows', '31 STATION Windows', '32 STATION Windows', '33 STATION Windows', '34 STATION Windows', '35 STATION Windows', '36 STATION Windows', '37 STATION Windows', '38 STATION Windows', '39 STATION Windows', '40 STATION Windows', '41 STATION Windows', '42 STATION Windows', '43 STATION Windows', '44 STATION Windows', '45 STATION Windows', '46 STATION Windows', '47 STATION Windows', '48 STATION Windows', '49 STATION Windows', '50 STATION Windows', '51 STATION Windows', '52 STATION Windows', '53 STATION Windows', '54 STATION Windows', '55 STATION Windows', '56 STATION Windows', '57 STATION Windows', '58 STATION Windows', '59 STATION Windows', '60 STATION Windows', '61 STATION Windows', '62 STATION Windows', '63 STATION Windows', '64 STATION Windows', '65 STATION Windows', '66 STATION Windows', '67 STATION Windows', '68 STATION Windows', '69 STATION Windows', '70 STATION Windows', '71 STATION Windows', '72 STATION Windows', '73 STATION Windows', '74 STATION Windows', '75 STATION Windows', '76 STATION Windows', '77 STATION Windows', '78 STATION Windows', '79 STATION Windows', '80 STATION Windows', '81 STATION Windows', '82 STATION Windows', '83 STATION Windows', '84 STATION Windows', '85 STATION Windows', '86 STATION Windows', '87 STATION Windows', '88 STATION Windows', '89 STATION Windows', '90 STATION Windows', '91 STATION Windows', '92 STATION Windows', '93 STATION Windows', '94 STATION Windows', '95 STATION Windows', '96 STATION Windows', '97 STATION Windows', '98 STATION Windows', '99 STATION Windows', '100 STATION Windows'. The '1 SCATTER window' option is highlighted.

The main window displays a scatter plot of a predefined window layout. The plot shows a grid of latitude (39.0°N to 40°N) and longitude (24.65°E to 24.75°E) with a red crosshair. The plot area is labeled 'Window 1 STATION'. Other window templates (Window 2 to Window 6) are visible in the background, each with a placeholder text: 'Press ENTER to add the data of the current station to the plot.' The right sidebar shows 'Station ID: 1' with details like Cruise (MO\_TS\_MO\_ATHOS.nc), Station (188), Position (24.724°E / 39.874°N), Date (01 May 2000), Time (00:00:00.000), and DEPTH Range (30 - 51). Below this is a 'Sample # / n' table with columns for various parameters and their values. At the bottom, there is a 'Innosurface Values' section with parameters like Longitude (24.724), Latitude (39.874), Time (hr) (2000.311), Day of Year (122), Depth of each measurement (0.00), atmospheric pressure at sea level, air temperature in dry bulb, and water crossing mass.

# Plot the time series



Right-click on plot:  
Change X and Y variable (temperature vs. time)

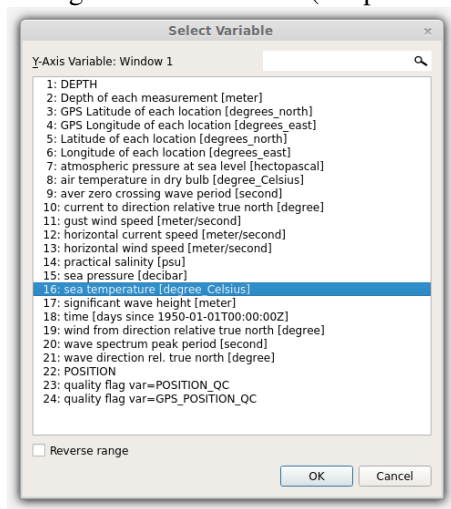


# Plot the time series



Right-click on plot:

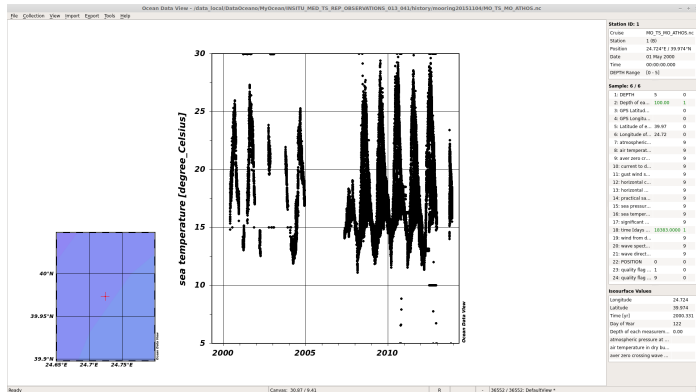
Change X and Y variable (temperature vs. time)



# Plot the time series



Right-click on plot:  
Change X and Y variable (temperature vs. time)

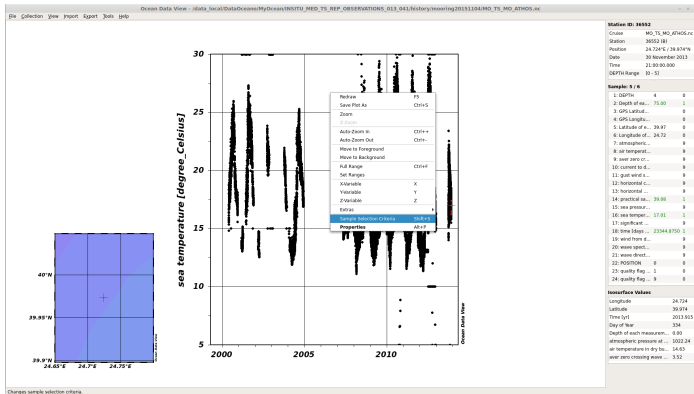


Bad values: will be treated later

# Plot the time series



Right-click on plot: Sample Select Criteria  
→ depth range



# Plot the time series



Right-click on plot: Sample Select Criteria  
→ depth range

Sample Selection Criteria

Range Quality

Variable  
DEPTH

Acceptable Range  
2 - 2

Relax this range filter Relax all range filters

0 of 24 variables range filtering  
0 of 24 variables quality filtering

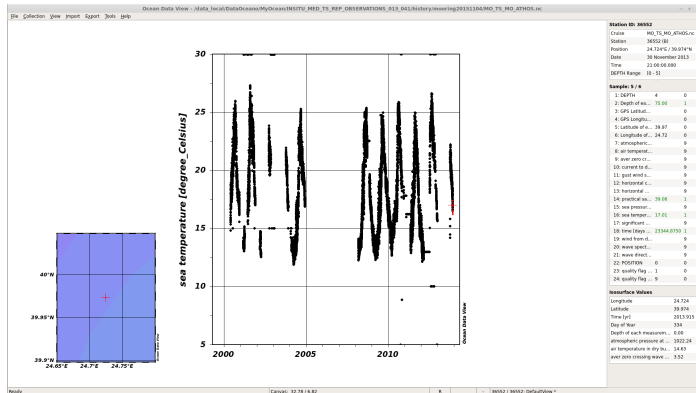
Apply these sample selection criteria globally

Help OK Cancel

# Plot the time series



Right-click on plot: Sample Select Criteria  
→ depth range

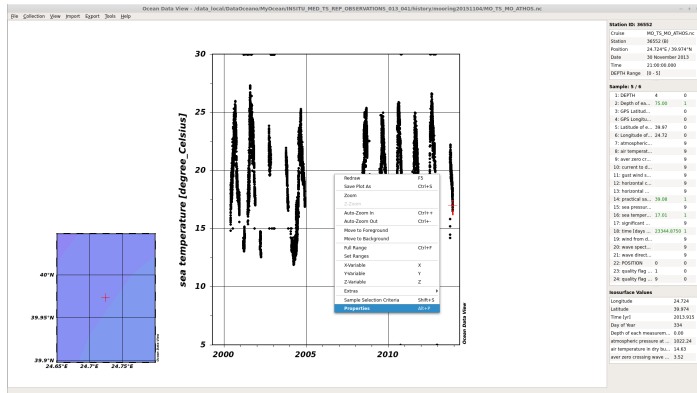


Now we have the series at 2 depth

# Improve the plot

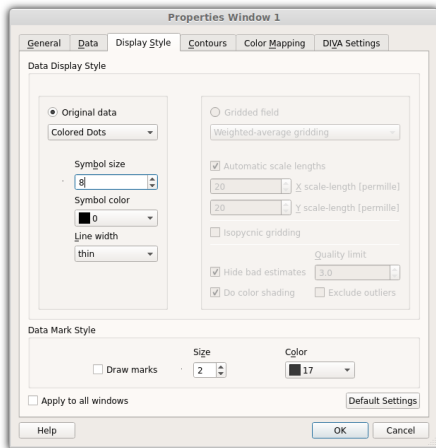


## Right-click on plot: Properties





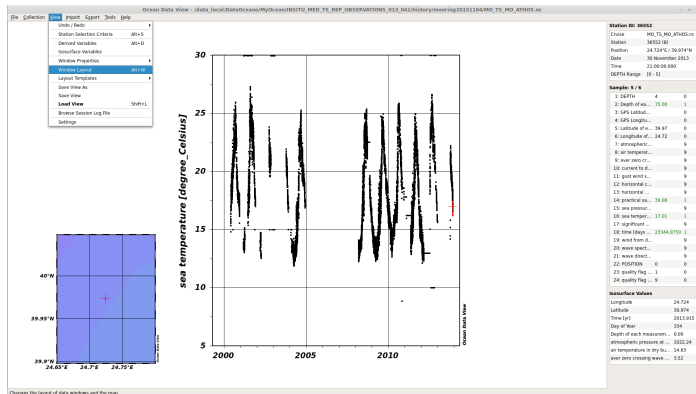
## Display Style: modify Symbols Size



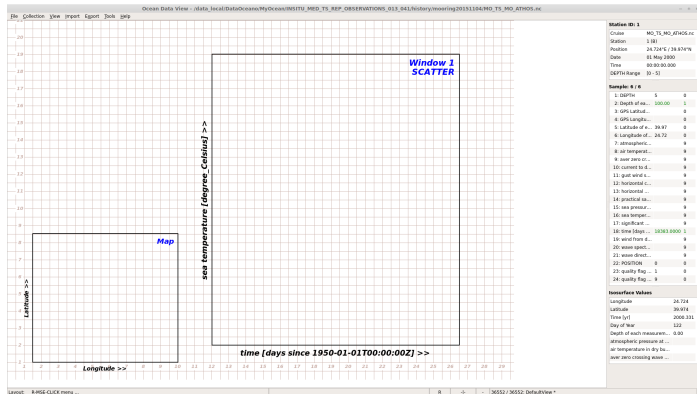
# Improve the plot



## View → Window Layout




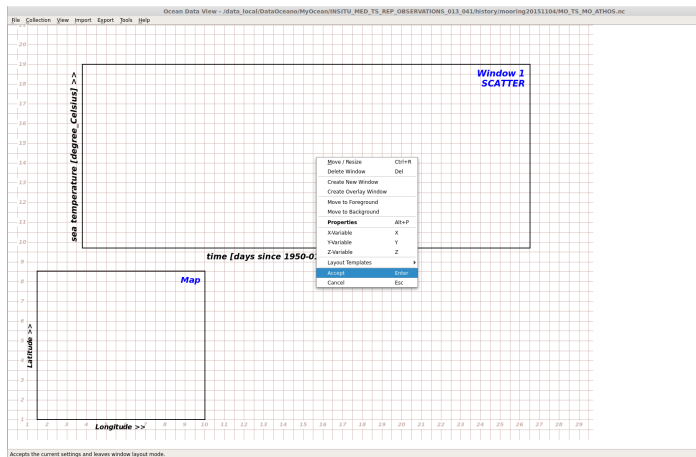
## Adapt size of the Scatter window



# Improve the plot



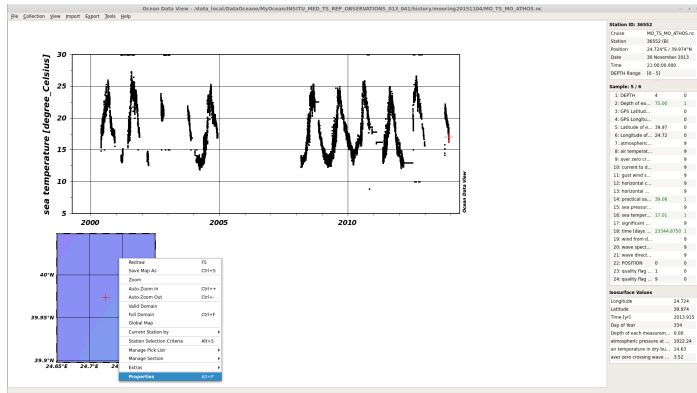
Accept the change (Enter )



# Improve the plot



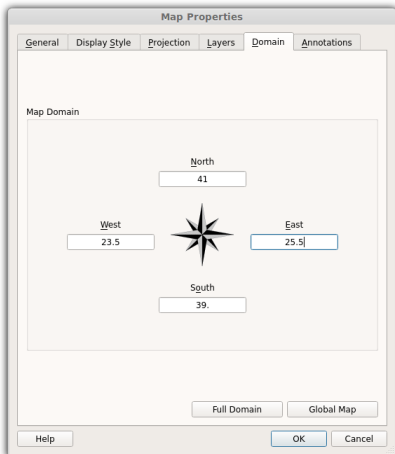
Right-click on plot: Properties → Domain



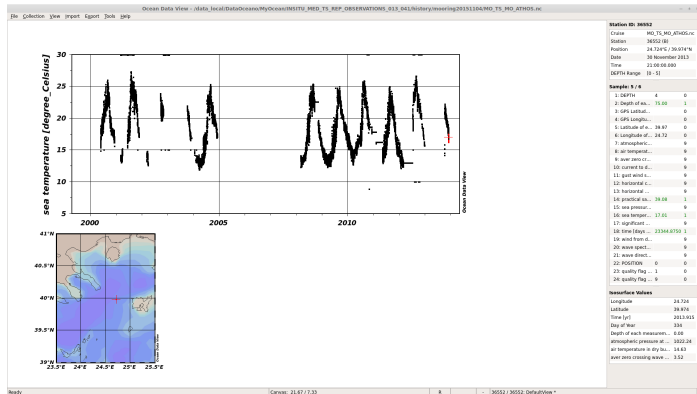
# Improve the plot



## Enlarge the map domain



## Enlarge the map domain



# Apply quality flags



Right-click on plot: Sample Select Criteria → depth range  
Select good data only

Sample Selection Criteria

Range Quality

Variable  
sea temperature [degree\_Celsius]

Acceptable Quality Flags

- 0: no QC was performed
- 1: good data**
- 2: probably good data
- 3: bad data that are potentially correctable
- 4: bad data
- 5: value changed
- 7: nominal value
- 8: interpolated value
- 9: missing value

Relax this quality filter Apply to all variables

1 of 24 variables range filtering  
0 of 24 variables quality filtering

Apply these sample selection criteria globally

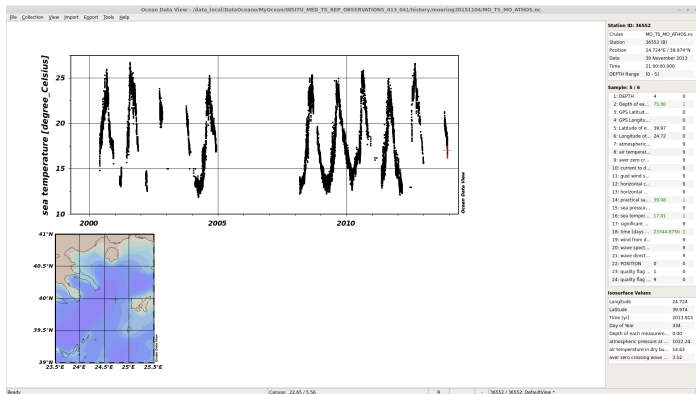
Help OK Cancel



# Apply quality flags



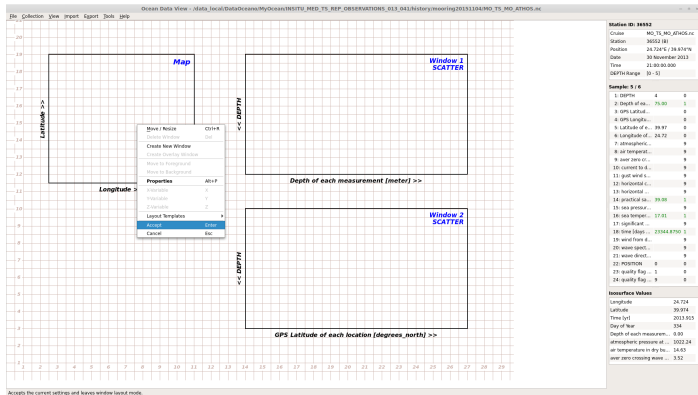
Right-click on plot: Sample Select Criteria → depth range  
Select good data only



# Apply quality flags



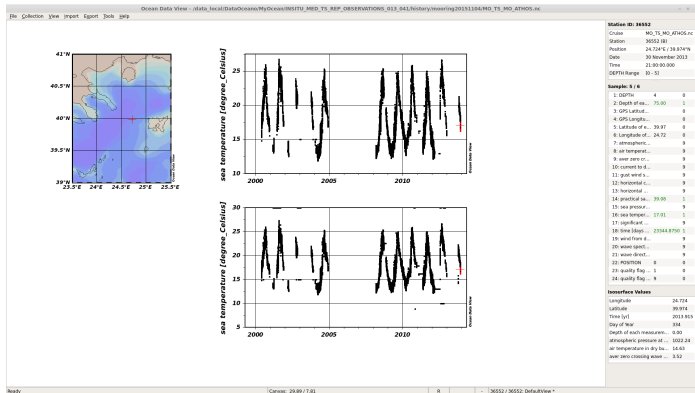
Configure Window Layout to have the 2 time series  
(with and without QC)



# Apply quality flags



Configure Window Layout to have the 2 time series  
(with and without QC)



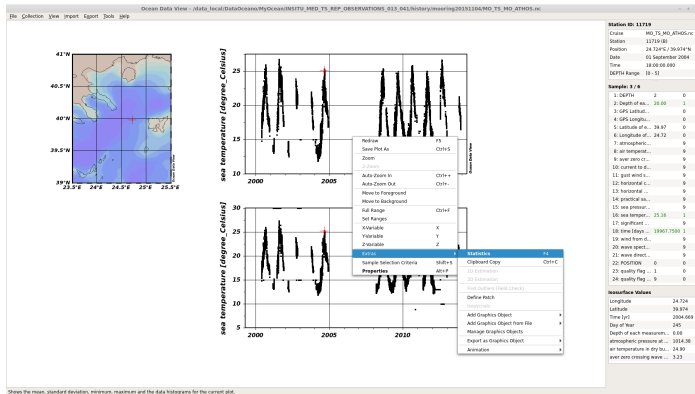
Extreme values are removed

# Apply quality flags



## Compare histograms:

📄 Right-click on plot → Extra → Statistics

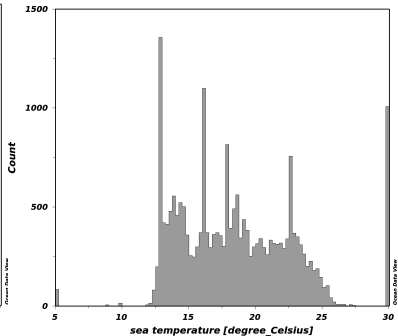
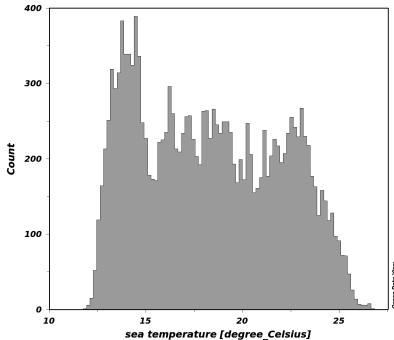


# Apply quality flags



Compare histograms:

📄 Right-click on plot → Extra → Statistics



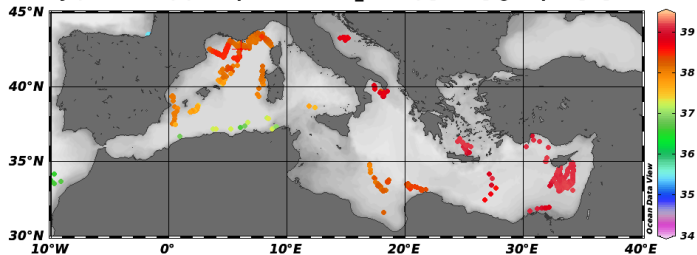
Distribution is improved

# Working with ODV on CORA data set

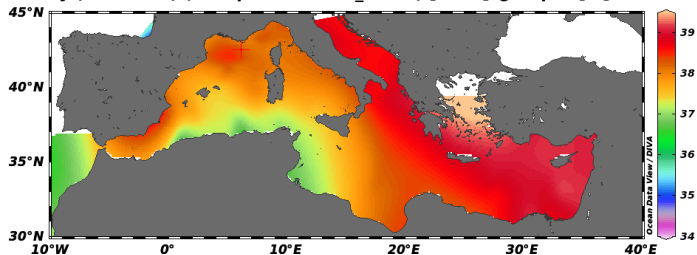
# Objective: process CORA dataset



**Salinity (S78 - PSS) (interpolated on Z\_levels) [none] @ depth [m]=0**



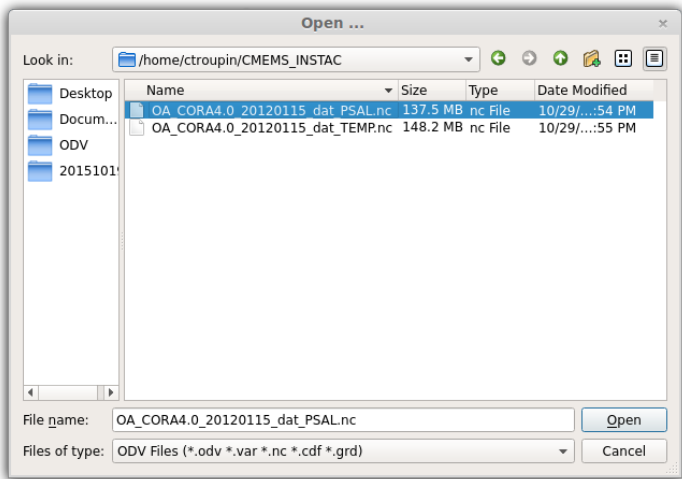
**Salinity (S78 - PSS) (interpolated on Z\_levels) [none] @ depth [m]=0**



# Opening the netCDF file



File → Open → Select the netCDF file





## Dimension and variables : Next

**NetCDF Setup Wizard**

Select Dimensions (Step 1 of 4)

---

NetCDF dimensions	Corresponding netCDF variables
<ul style="list-style-type: none"><li>N_PROF[36038]</li><li>N_LEVELS[152]</li></ul>	<ul style="list-style-type: none"><li>Cycle number; var=CYCLE_NUMBER</li><li>Julian day (UTC) relative to REFERENCE_DATE_TIME [days since</li><li>Latitude of the station, best estimate [degree_north]; var=LAT</li><li>Longitude of the station, best estimate [degree_east]; var=LO</li><li>depth [m]; var=DEPH</li><li>profile processing level; var=PSAL_PROC</li><li>Quality flag on interpolated variable; var=PSAL_QC</li><li>Salinity (S78 - PSS) (interpolated on Z_levels) [none]; var=PSA</li><li>Climatology mean for profile [none]; var=PSAL_CLMN</li><li>Climatology standard deviation for profile [none]; var=PSAL_C</li><li>Measurement error [none]; var=PSAL_ERME</li><li>Error from unresolved scales [none]; var=PSAL_ERUR</li><li>Residual [none]; var=PSAL_RESI</li><li>N_PROF; var=N_PROF</li><li>N_LEVELS; var=N_LEVELS</li></ul>

All 2 dimensions selected

---

## Variable association : Next

**NetCDF Setup Wizard**

Associate Meta Variables (Step 2 of 4)

NetCDF variables	Meta variables
3: Cycle number; var=CYCLE_NUMBER	Cruise
* 11: Julian day (UTC) relative to REFERENCE_DATE_TIM	Station
* 12: Latitude of the station, best estimate [degree_nor	Type
* 13: Longitude of the station, best estimate [degree_e	* Longitude [degrees_east]
14: depth [m]; var=DEPH	* Latitude [degrees_north]
15: profile processing level; var=PSAL_PROC	* Year
16: Quality flag on interpolated variable; var=PSAL_Q	* Month
17: Salinity (S78 - PSS) (interpolated on Z_levels) [nor	* Day
18: Climatology mean for profile [none]; var=PSAL_CL	* Hour
19: Climatology standard deviation for profile [none];	* Minute
20: Measurement error [none]; var=PSAL_ERME	* Second
21: Error from unresolved scales [none]; var=PSAL_EF	
22: Residual [none]; var=PSAL_RESI	
23: N_PROF; var=N_PROF	

3 of 15 variables used

8 of 11 variables associated

Buttons: Help, < Back, Next >, Finish, Cancel

NetCDF Setup Wizard controls: Associate, Convert, Set Default, Undo, >>

## Primary variables : Next

**NetCDF Setup Wizard**

Select Primary Variable (Step 3 of 4)

---

Available netCDF dimensions

- Cycle number
- Julian day (UTC) relative to REFERENCE\_DATE\_TIME [days]
- Latitude of the station, best estimate [degree\_north]
- Longitude of the station, best estimate [degree\_east]
- depth [m]
- profile processing level
- N\_PROF
- N\_LEVELS**

Use selected variable

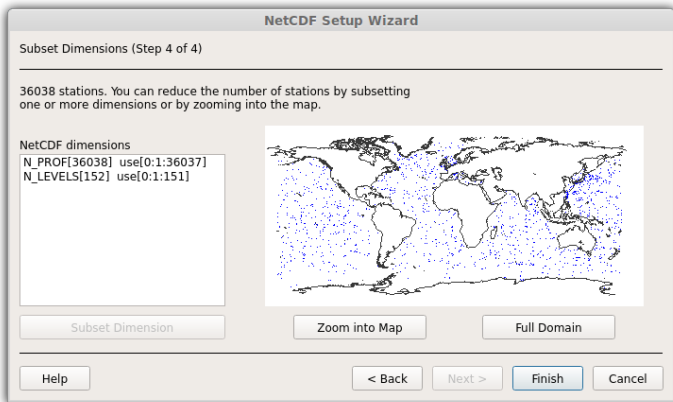
Use decimal date/time (header)

Use dummy variable

---

Help      < Back      Next >      Finish      Cancel

## Subset dimensions : Finish



# Opening the netCDF file



## We get this window

Ocean Data View - home/euroscipio/CHEM5\_INSTAC/OA\_CORR&0\_20120115\_dut\_P5AL.nc

File | Detection | View | Import | Export | Tools | Help

Window 1 STATION  
Press ENTER to add the data of the current station to the plot.  
Cycle number >>>

Window 2 STATION  
Press ENTER to add the data of the current station to the plot.

Window 3 STATION  
Press ENTER to add the data of the current station to the plot.

Window 4 STATION  
Press ENTER to add the data of the current station to the plot.

Window 5 STATION  
Press ENTER to add the data of the current station to the plot.

Window 6 STATION  
Press ENTER to add the data of the current station to the plot.

Climatology standard deviation for profile [msec]

Error from unresolved scales [msec] >>>

Climatology mean for profile [msec] >>>

latitude to REFERENCE\_DATE\_TIME [days since REF]

altitude of the station, best estimate [degrees\_pos]

88°W 80°W 60°W 0° 30°E 120°E

Station ID: 1

Crane	06_CORRA_0_20120115_dut_P5AL.nc
Station	1181
Position	64.489°W 55.352°S
Date	08 December 0991
Time	00:45:55.667
N_LEVELS Range	10 - 3512

Sample: 1 / 232

1: N_LEVELS	0	1
2: Cycle number	98	1
3: Climatology mean for profile [msec]	34.09	1
4: Climatology standard deviation [msec]	6.134	1
5: Error from unresolved scales [msec]	6.133	1
6: Julian Day (UTC) relative to REF [days since REF]	23023.04069	1
7: Latitude of the station, best est. [degrees_pos]	-55.1517	1
8: Longitude of the station, best est. [degrees_pos]	-64.4890	1
9: Measurement error [msec]	0.07	1
10: N_PROF	0	1
11: Quality flag on interpolated var. [1]	1	1
12: Residual [msec]	0.06	1
13: Satiny (519 - 952) Interpolated. [days since REF]	34.09	1
14: profile processing level [1]	1	1
15: depth [m]	0.0000	1

Surface Values

Longitude	64.489
Latitude	-55.152
Time [yr]	63.931
Day of Year	342
Climatology mean for profile [msec] @ N_LEVELS=first	34.09
Climatology standard deviation for profile [msec] @ N_	6.14
Error from unresolved scales [msec] @ N_LEVELS=first	6.13
Measurement error [msec] @ N_LEVELS=first	0.07

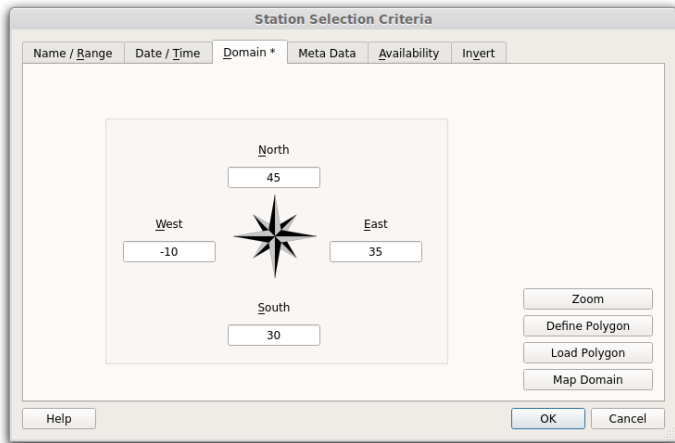
Ready

30038 / 30038: DefaultView\*

# Region selection and basic statistics



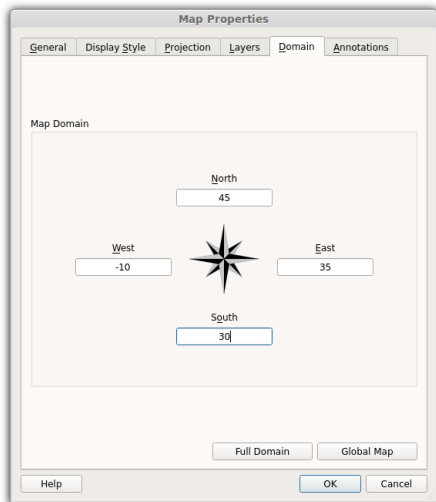
Right-click on image → Station Selection Criteria → Domain



# Region selection and basic statistics



Right-click on image → Properties → Domain



# Region selection and basic statistics



Right-click on image → Extra → Statistics

**Map Statistics**

Summary

-----Visible stations-----

	Mean	Stand. Dev.	# Points	Minimum	Maximum
Longitude	8.3463	+ 11.8378	1093	[ -9.992	34.2175 ]
Latitude	41.5833	+ 4.717	1093	[ 32.096	47.8219 ]

Distributions

X Histogram   Y Histogram   X/Y Distribution

Time Histogram   Season Histogram

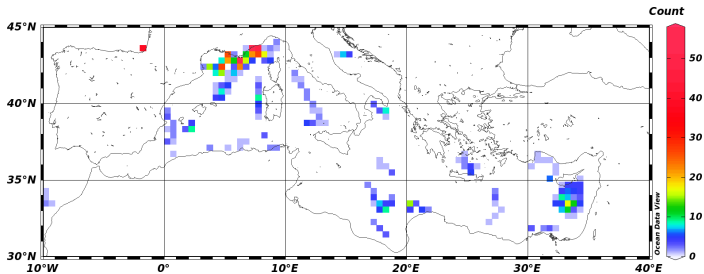
Help   Clipboard Copy   Close



# Region selection and basic statistics



Figure → X/Y Distribution



☒ Data scarcity and inhomogeneous distribution

# Map improvement



☞ Right-click on map → **Properties**

General: palette, colors etc

The image shows a 'Map Properties' dialog box with the following settings:

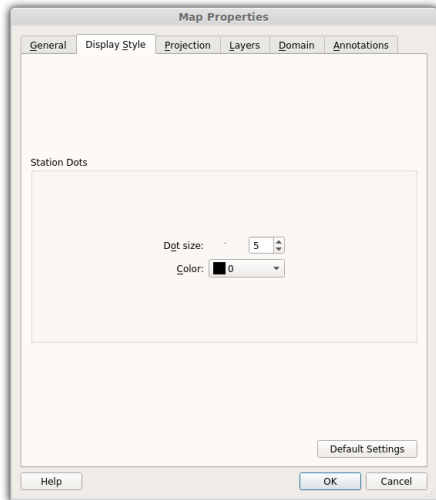
- General** (selected tab)
- General** section:
  - Palette: Odv
  - Background color: (none)
- Font** section:
  - Font base size [pt]: (automatic)
  - Font size factor: 100 %
- Axis Style** section:
  - Axis color: 0
  - Draw grid
- Buttons: Help, OK, Cancel, Default Settings

# Map improvement



☞ Right-click on map → **Properties**

Display style: increase dot size, change color

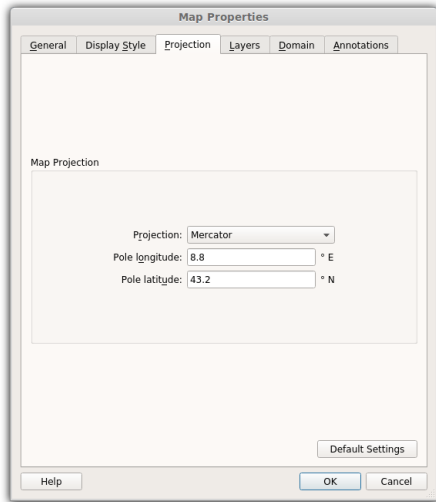


# Map improvement



☞ Right-click on map → **Properties**

Projection: modify according to preference

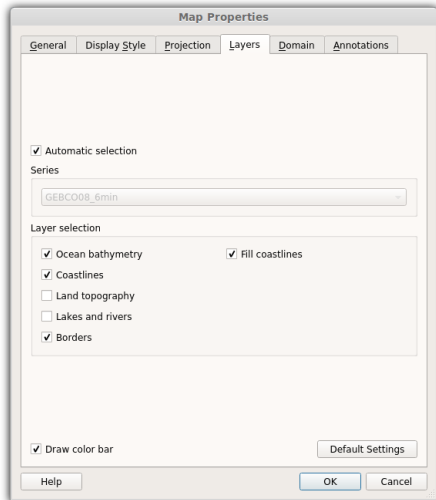


# Map improvement



☞ Right-click on map → **Properties**

Layers: bathymetry + coastlines

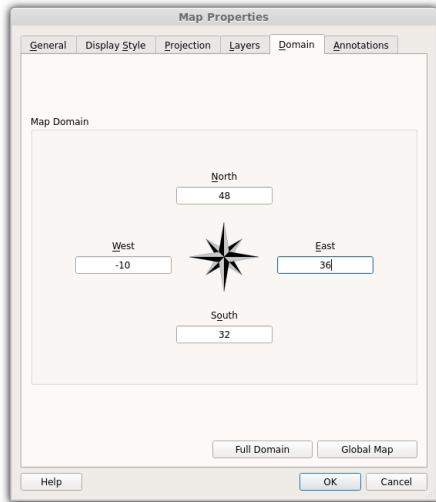


# Map improvement



☞ Right-click on map → **Properties**

Domain: adjust limits (already done)



# Map improvement



☞ Right-click on map → **Properties**

Annotations: not necessary

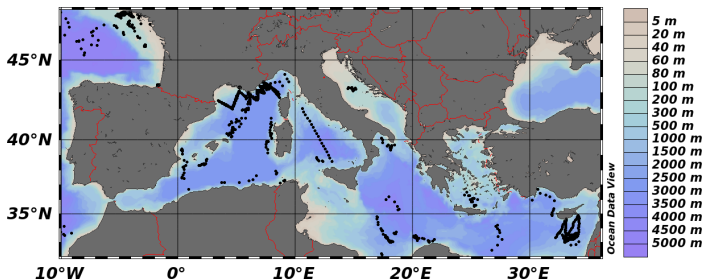


# Map improvement



Right-click on map → **Properties**

View → Layout template → Full Screen Map F8





# Station window



View → Layout template → Station window

The screenshot shows the Ocean Data View software interface. The 'View' menu is open, showing options like 'Full Screen Map', '1 Station window', '2 STATION Windows', etc. The 'Station window' layout template is selected, showing a grid of windows. The 'Station ID: 1' panel is visible on the right, displaying station information and sample data.

**Station ID: 1**

Crane OA\_C0944\_0\_20120115\_1\_1  
Station 1180  
Position 6.163°E 42.485°N  
Date 05 December 1981  
Time 17:18:53.000  
N\_LEV\_10 - 1511

**Sample 1 / 152**

1: N_LEVELS	0	1
2: Cycle number	0	2
3: Climatology	...	1
4: Climatology	...	1
5: Error from u...	...	1
6: Julian day (M...	22616.7159	3
7: Latitude of...	42.4851	3
8: Longitude of...	6.1633	3
9: Measurement...	...	1
10: N_PROF	396	1
11: Quality Flag	9	1
12: RepeatID (n...	...	2
13: Salinity CT...	...	1
14: profile proc...	1	3
15: depth [m]	0.0000	1

**Surface Values**

Longitude	6.163
Latitude	42.485
Time [yr]	41.826
Day of Year	330
Climatology mean for prof...	...
Climatology standard deviat...	...
Error from unresolved scale...	...
Measurement error [name] ...	...

View → Layout template → Station window

Ocean Data View - Home>trajploc\MEMO\_INSTRC\DA\_CDR&4.0\_20120115\_dlat\_PXAL.nc

File Collection View Import Export Tools Help

60°N  
40°N  
20°N  
0°  
30°W 0° 30°E 60°E

Window 1  
STATION

<< N\_LEVELS

Press ENTER to add the data of  
the current station to the plot.

Cycle number >>

**Station ID: 197**

Cruise: GC-COMAR-3\_20120115\_E...  
Station: 107 (E)  
Position: 45°N, 40.185°W  
Date: 01 December 1984  
Time: 13:59:49.000  
N\_LEVEL: 00 - 1511

**Sample 1 / 152**

1: N_LEVELS	0	1
2: Cycle number	96	1
3: Climatology	37.95	1
4: Climatology	0.23	1
5: Error from u...	0.21	1
6: Julian day (J...)	2861.5554	1
7: Latitude of t...	40.1850	1
8: Longitude of...	4.5700	1
9: Measurement...	0.02	1
10: N_PROF	994	1
11: Quality Flag	2	1
12: Height [m]	-0.41	1
13: Salinity STD	37.28	1
14: profile proc...	2	1
15: depth [m]	0.0000	1

**isocurface Values**

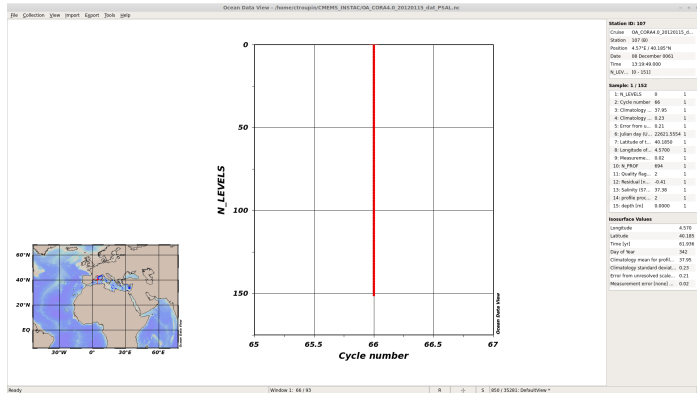
Longitude	4.570
Latitude	40.185
Time [yr]	61.998
Day of year	342
Climatology mean for profile	37.95
Climatology standard deviat...	0.23
Error from unrescaled scale	0.21
Measurement error (standard)	0.02

Ready Window 1: k + S | 850 / 35281: DefaultView \*

# Station window



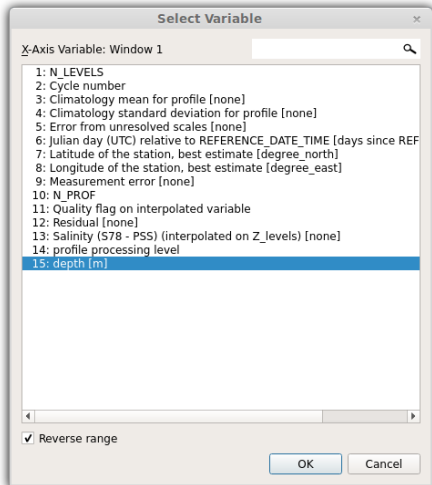
Enter 



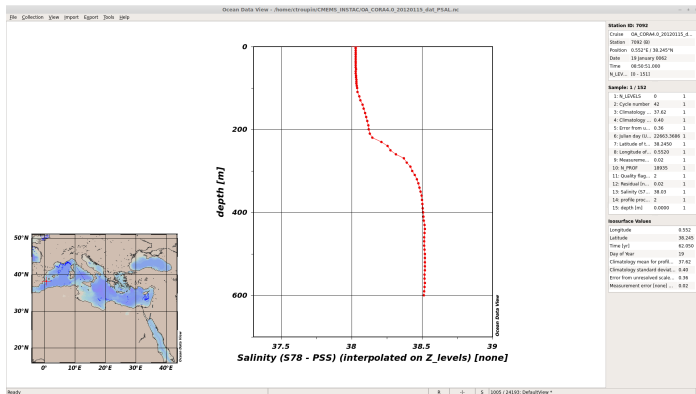
Right-click on map:

Change X and Y variable (salinity vs. depth)

Y variable → Check the reverse range box



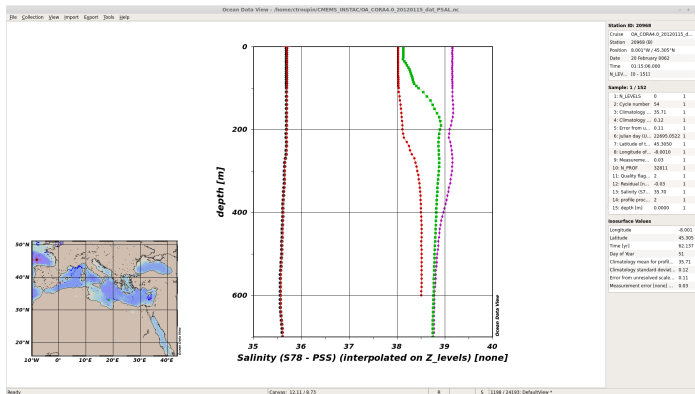
Double click on the map (left) to get profiles at different locations



☒ Very different properties according to the basin

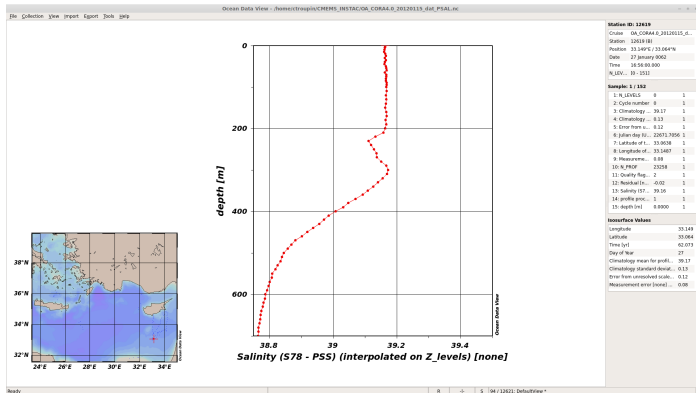
👉 Maybe needed to adjust range, otherwise not visible

## Compare profiles in different sub-regions

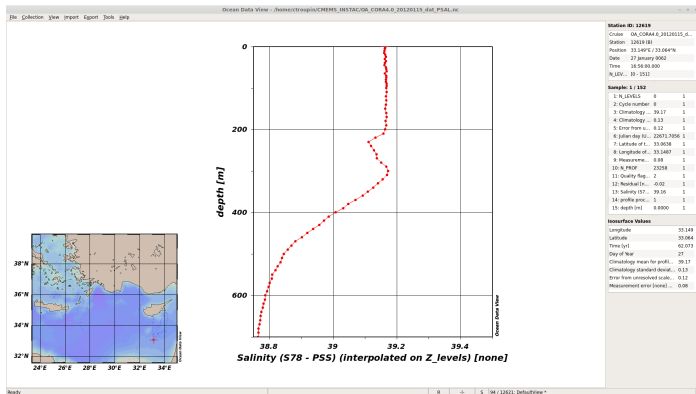


To remove stations: Manage Pick List → Remove all Stations

What happens with this profile south of Cyprus?



What happens with this profile south of Cyprus?



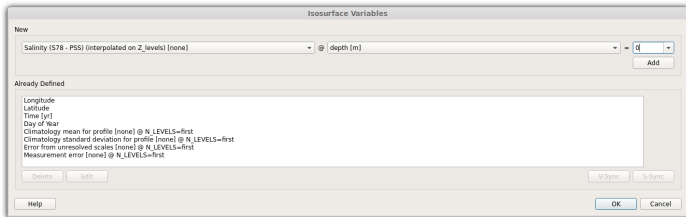
☒ Mixed-layer depth



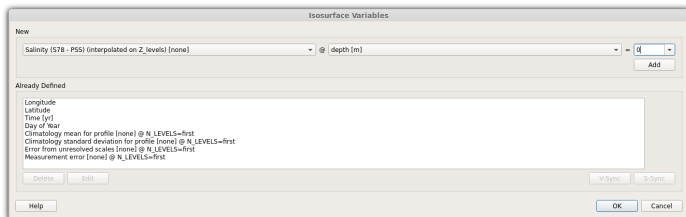
Define new isosurface variables:

View → Isosurface Variables → salinity at depth = 0

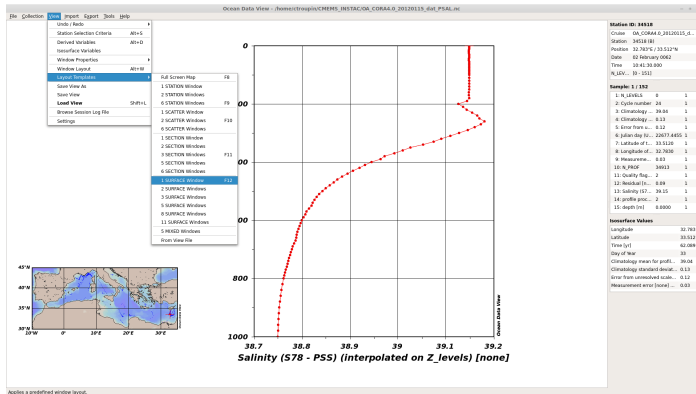
Click on "Add"



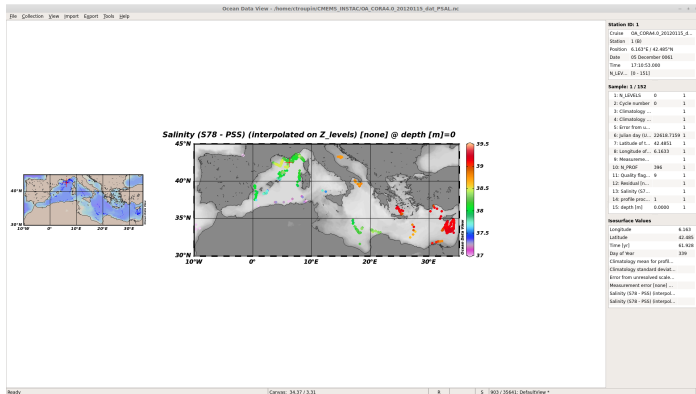
Define new isosurface variables:  
Same at depth = 200



View → Layout Template → SURFACE Window



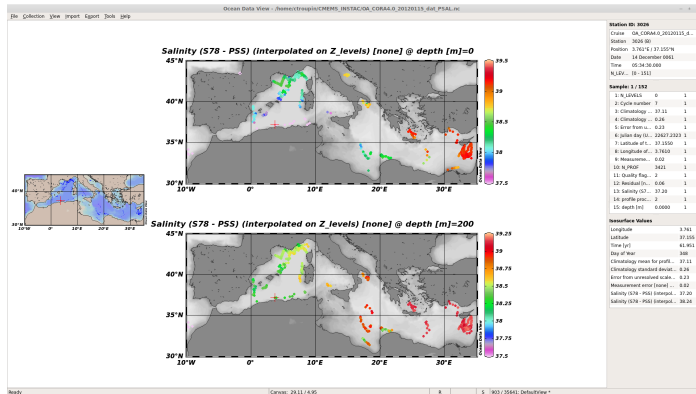
📁 Right-click Z-variable → select newly created variable



📧 higher salinity values in the Eastern Basin

👉 Adapt the range for the selected variable

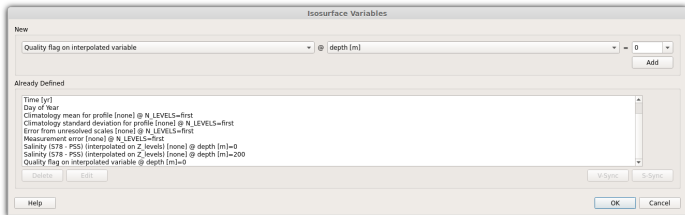
Also possible to have several Surface Windows



# Surface window: quality flag



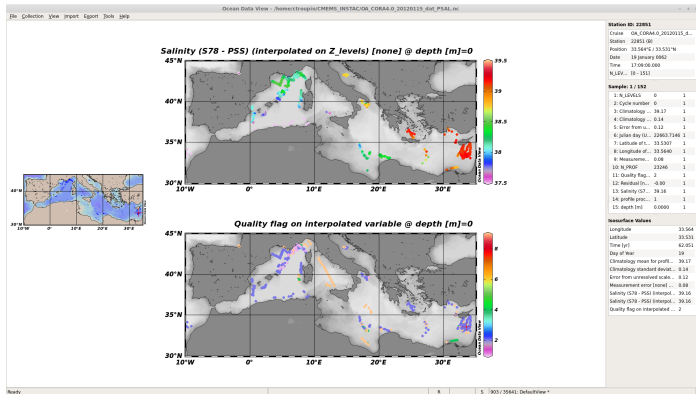
View → Isosurface Variables → Quality flag on interpolated variable at depth = 0



# Surface window: quality flag



Quality flag: integer value reflecting the confidence in the observations



## CORA Quality flags:

- 1 good
- 2 rather good
- 3 quite good
- 4 acceptable
- 5 bad quality interpolation
- 6, 7, 8 not used
- 9 not interpolated



ODV definitions for the flags are different!



# Surface window: quality flag



Right-click Sample Selection Criteria → Quality → Accepted  
quality flags = 1

**Sample Selection Criteria**

Range   Quality

Variable  
\* Salinity (S78 - PSS) (interpolated on Z\_levels) [none]

Acceptable Quality Flags  
0: good quality  
1: unknown quality  
4: questionable quality  
8: bad quality

Relax this quality filter   **Apply to all variables**

0 of 15 variables range filtering  
All 15 variables quality filtering

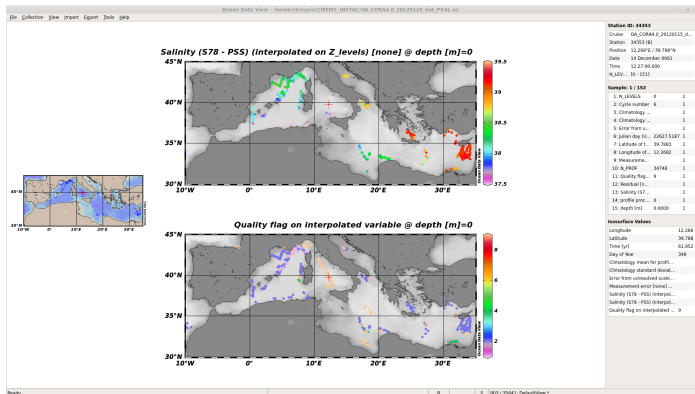
Apply these sample selection criteria globally

Help   OK   Cancel

# Surface window: quality flag



Right-click Sample Selection Criteria → Quality → Accepted  
quality flags = 1



higher salinity values in the Eastern Basin

# Surface window: gridding



View → Layout Template → SURFACE Window (× 2)

The screenshot shows the Ocean Data View software interface. The main window displays a map of the Atlantic Ocean with a grid overlay. The map is titled "Ocean Data View - AtmosTropicsCMMMS\_WS76C/OA\_CORA4.0\_20120115\_01d\_P5AL.nc". The map shows latitude from 30°N to 45°N and longitude from 10°W to 30°E. A menu is open over the map, showing options for "View", "Import", "Export", "Tools", and "Help". The "View" menu is expanded, showing options for "Grids / Reeds", "Station Selection Criteria", "Derived Variables", "Non-surface Variables", "Window Properties", "Window Layout", "Layout Templates", "Save View As", "Save View", "Load View", "Browse Screen Log File", and "Settings". The "Layout Templates" menu is further expanded, showing options for "6-8 Screen Map", "1 STATION Windows", "2 STATION Windows", "6 STATION Windows", "1 SCATTER Windows", "2 SCATTER Windows", "6 SCATTER Windows", "1 SECTION Windows", "2 SECTION Windows", "3 SECTION Windows", "5 SECTION Windows", "6 SECTION Windows", "1 SURFACE Windows", "2 SURFACE Windows", "3 SURFACE Windows", "5 SURFACE Windows", "8 SURFACE Windows", "11 SURFACE Windows", "5 MIXED Windows", and "From View File".

The right-hand panel displays station and non-surface values. The station information is as follows:

Station ID: 8533		
Cruid	OA_CORA4.0_20120115_01d_	
Station	34518 (8)	
Position	32.783°S / 33.512°W	
Date	02 February 0862	
Time	10:41:30.800	
N_LEV...	10 - 1011	

The non-surface values are as follows:

Sample 1 / 352		
1: N_LEVELS	8	1
2: Cycle number	24	1
3: Climatology	30.04	1
4: Climatology	0.13	1
5: Error from ...	0.12	1
6: Julian day No.	20073.4852	1
7: Latitude of ...	33.5126	1
8: Longitude of ...	32.7826	1
9: Measurement	0.03	1
10: N_PROF	36613	1
11: Quality flag	2	1
12: Benchmark [m]	0.89	1
13: Salinity [ST]	39.15	1
14: profile proc...	2	1
15: depth [m]	8.8900	1

The non-surface values are as follows:

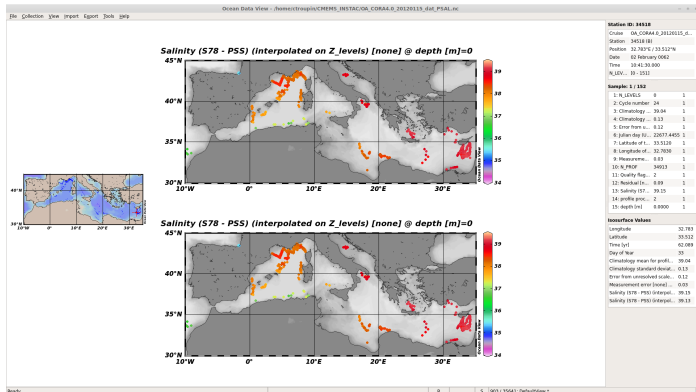
Non-surface Values		
Longitude	32.783	
Latitude	33.512	
Time [yr]	42.008	
Day of Year	33	
Climatology mean for prof...	30.04	
Climatology standard deviat...	0.13	
Error from unresolved scale...	0.12	
Measurement error [m]	0.03	

At the bottom left of the interface, there is a button labeled "Apply a predefined window layout."

# Surface window: gridding



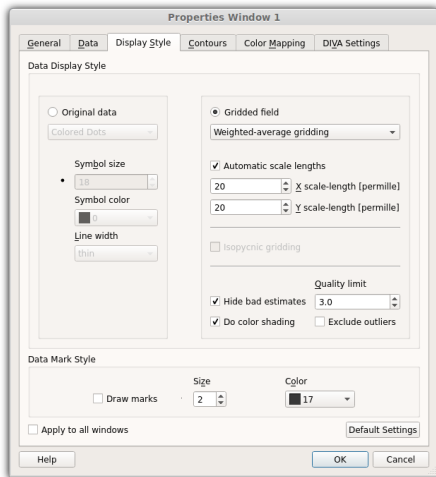
Set Z variable to be Salinity at 0 m



# Surface window: gridding



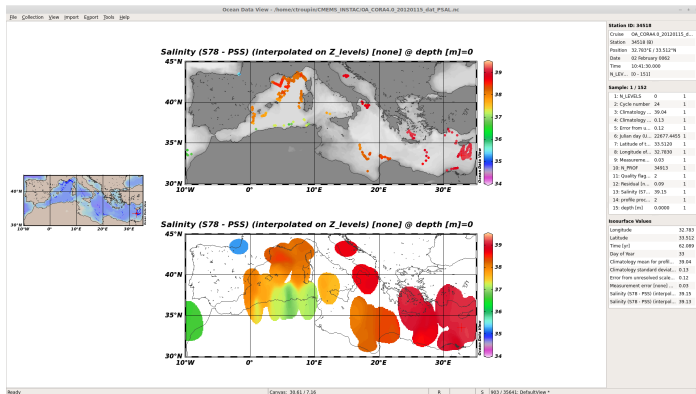
Right-click Properties → Display style → Gridded → Weighted-Average gridding (default parameters 20 X 20)



# Surface window: gridding



## Gridded field of salinity

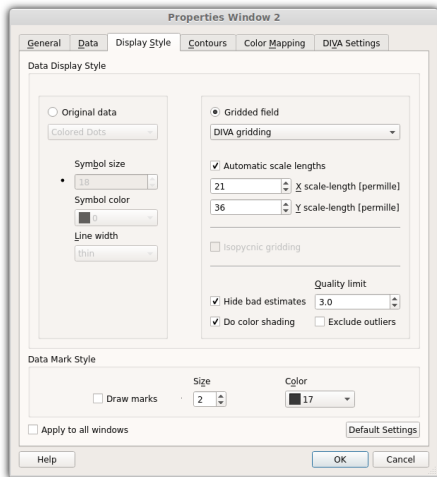


Normal interpolation does not consider boundaries!

# Surface window: gridding



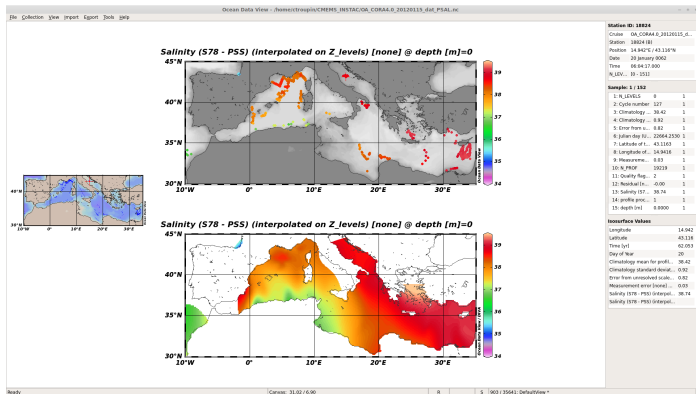
## Change Gridding method to DIVA Gridding



# Surface window: gridding



## DIVA gridded field of salinity



Field with error above threshold is masked



Interpolation technique is crucial with in situ data



View → Layout Template → SECTION Window

Ocean Data View - /home/traugott/CNERS\_INSTAC/0A\_CORA4.0\_20120115\_dlv\_PSAI.nc

File Collection View Import Export Tools Help

Units: None

Station Selection Criteria: All+S

Derived Variables: All+D

Interface Variables:

Window Layout: All+W

Window Properties:

Layout Templates:

Save View As:

Save View:

Load View: Shift+L

Browse Session Log File:

Settings:

Full Screen Map: F10

1 STATION Window:

2 STATION Windows: F9

3 STATION Windows: F10

4 STATION Windows: F11

5 STATION Windows: F12

6 SCATTER Window:

7 SCATTER Windows:

8 SCATTER Windows:

9 SCATTER Windows:

10 SCATTER Windows:

11 SCATTER Windows:

12 SECTION Windows:

13 SECTION Windows: F11

14 SECTION Windows: F12

1 SURFACE Window:

2 SURFACE Windows:

3 SURFACE Windows:

4 SURFACE Windows:

5 SURFACE Windows:

6 SURFACE Windows:

7 SURFACE Windows:

8 SURFACE Windows:

9 SURFACE Windows:

10 SURFACE Windows:

11 SURFACE Windows:

12 MIXED Windows:

From View File:

**Climatology mean for profile [none] @ N\_LEVELS=first**

**Standard deviation for profile [none] @ N\_LEVELS=first**

**Station ID: 18824**

Create: 0A\_CORA4.0\_20120115\_dlv\_PSAI.nc

Station: 18824 (8)

Position: 14.842°E / 43.116°N

Date: 20 January 2002

Time: 06:04:17.000

N\_LEVELS: 18 - 15(1)

**Sample: 3 / 392**

1: N_LEVELS	0	1
2: Cycle number	127	1
3: Climatology	-18.42	1
4: Climatology	-0.92	1
5: Error from...	0.82	1
6: Julian day (J...)	32964.2533	1
7: Latitude of...	43.1163	1
8: Longitude of...	14.8416	1
9: Measurement...	-0.02	1
10: N_PROF	18219	1
11: Quality Flag...	2	1
12: Residual (R...)	-0.80	1
13: Salinity (S...)	-38.74	1
14: profile proc...	1	1
15: depth (m)	0.0000	1

**seasurface Values**

Longitude	14.842
Latitude	43.116
Time (yr)	62.853
Day of Year	20
Climatology mean for profil...	-18.42
Climatology standard deviat...	0.82
Error from unresolved scale...	0.82
Measurement error (Datal...	0.82
Salinity (S78 - PSS) Interpol...	-38.74
Salinity (S78 - PSS) Interpol...	...

Applies a predefined window layout.

Right-click Manage Section → Define Section  
Draw line along section

Ocean Data View - RemoteGraphics\CMEMS\_InstC\OA\_CORAA\_0\_20120115\_d4t\_PSL.nc

File Collection View Import Export Tools Help

Window 1  
SECTION

<< N\_LEVELS

Define a section using  
Manage Section>Define Section  
to add data to the plot.

Newline	F5
Save Map As	Ctrl+S
Zoom	
Auto-Down In	Ctrl+I
Auto-Down Out	Ctrl+O
Hold Domain	
Full Domain	Ctrl+F
Global Map	
Current Station by	
Station Selection Criteria	Alt+S
Manage PCK List	
Define Section	Ctrl+D
Properties	Alt+P
Section Properties	

45°N  
40°N  
35°N  
30°N  
10°W 0° 10°E 20°E 30°E

Metadata ID: 18024  
Cruise: OA-CORAA-0\_20120115\_d...  
Station: 18024 (R)  
Position: 34.842° / 43.116°W  
Date: 30 January 2002  
Time: 06:04:17.000  
N\_LEV...: (0 - 151)

Sample 1: 1502

1: N_LEVELS	0	1
2: Cycle-number	127	1
3: Climatology	36.42	1
4: Climatology	0.92	1
5: Error flag	0.00	1
6: Julian Day No.	229642.2130	1
7: Latitude of ...	43.1164	1
8: Longitude of ...	34.8418	1
9: Measurement	0.00	1
10: N_PROF	190219	1
11: Quality flag	2	1
12: SeaLevel [m]	-0.00	1
13: Salinity [S]	36.74	1
14: profile proc.	1	1
15: depth [m]	0.0000	1

Isosurface Values

Longitude	14.942
Latitude	43.116
Time [yr]	62.003
Day of Year	20
Climatology mean for prof...	36.42
Climatology standard devi...	0.92
Error from unsmoothed scale...	0.02
Measurement error [msec]	0.03
Salinity [S] - PSS Interpol...	36.74
Salinity [S] - PSS Uninterp...	

Defines a new section along a user specified track. L.MSE click adds current point, R.MSE click removes choosed point. Press ENTER to accept or ESC to abort.

## Edit Section Properties

Ocean Data View - ResearchViz\ICEMEM5\INSTSC\DA\_CORA4.0\_20120115\_EUT\_P5AL.IN

File Collection View Import Export Tools Help

**Section Properties**

Section title:

Section Coordinate:

Distance based  Longitude  Latitude

Bathymetry

No bathymetry

Station bottom depth

File

Mean Width:

Bathymetry Color:

**Station ID: 18624**

Cruise:

Station:

Position:

Date:

Time:

H.LEV.:

**Sample: 1 / 182**

1: N_LEVELS	0	1
2: Cycle number	123	1
3: Climatology	-36.42	1
4: Climatology ...	0.92	1
5: Error from ...	0.62	1
6: Julian day (J...)	20464.2038	1
7: Latitude of ...	43.1163	1
8: Longitude of ...	14.9428	1
9: Measurement ...	0.03	1
10: H_PROF	19219	1
11: Quality Flag ...	2	1
12: Residual (m...)	-4.66	1
13: Salinity (ST ...)	36.74	1
14: profile proc...	1	1
15: depth (m)	0.0008	1

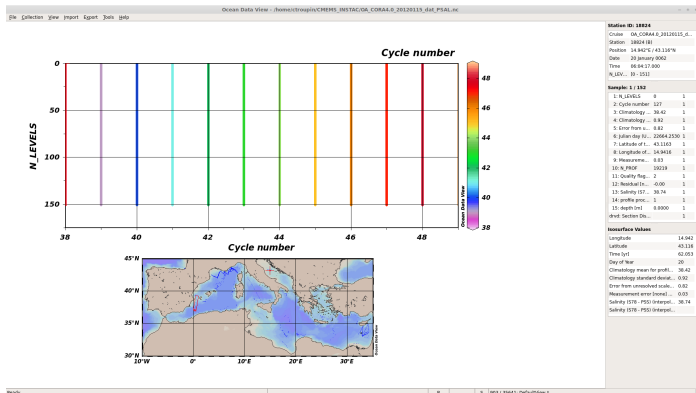
**Resurface Values**

Longitude	14.942
Latitude	43.116
Time (yr)	62.653
Day of Year	29
Climatology mean for profile...	36.42
Climatology standard deviat...	0.92
Error from unrescaled scale...	0.62
Measurement error (rescaled)...	0.03
Salinity (STH - PSS) Untepl...	36.74
Salinity (STH - PSS) Untepl...	

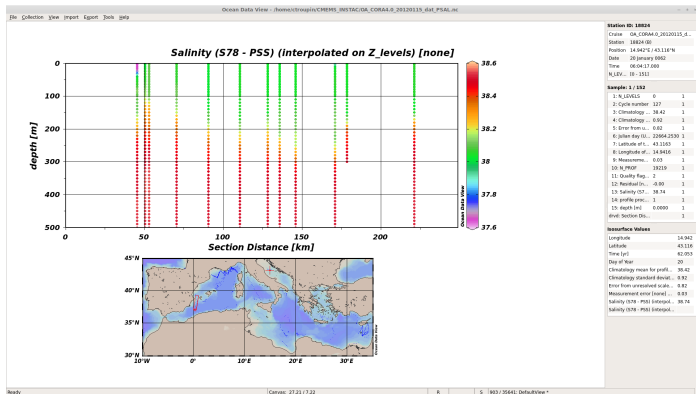
Get Point: L-RSG add point, R-RSG delete point, ONTCA accept, ESC abort

S | 843 / 3564: DefaultView \*

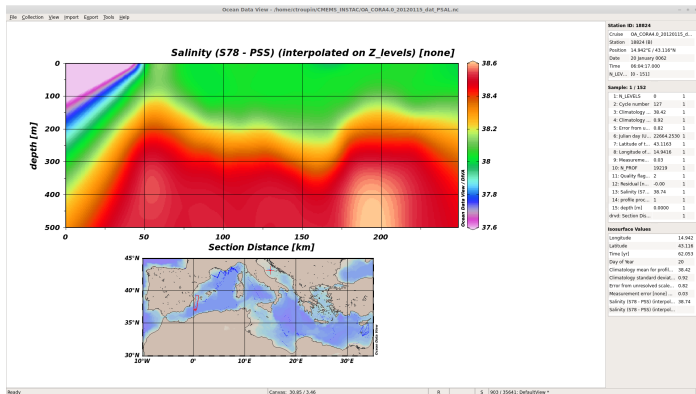
Change X, Y and Z variables  
 → Distance, Depth and Salinity



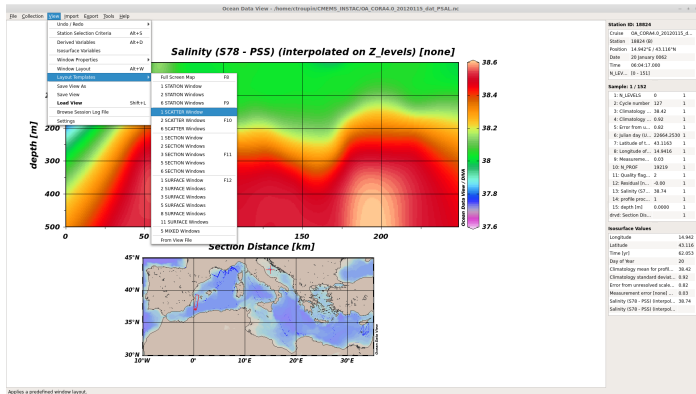
Set Z range between 0 and 500 m



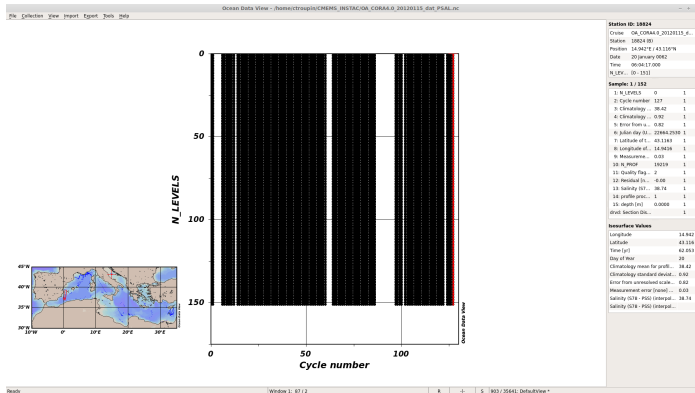
## Grid using DIVA interpolation



View → Layout Template → SECTION Window



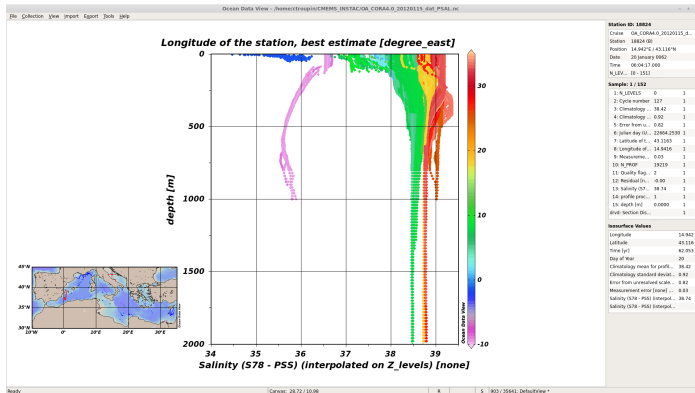
## Change X, Y and Z variables



☒ Usually: Salinity vs. Temperature (*T-S diagram*)



## Scatter plot: Salinity, Depth and Longitude



☒ Lower salinity near Atlantic

# Working on data using Python

# What is an ipython notebook?

---



**Python:** high-level programming language  
<https://www.python.org/>

# What is an ipython notebook?

---



**Python:** high-level programming language

<https://www.python.org/>

**IPython:** command shell for interactive computing

<http://ipython.org/>

# What is an ipython notebook?

---



**Python:** high-level programming language

<https://www.python.org/>

**IPython:** command shell for interactive computing

<http://ipython.org/>

**IPython notebook:** web-based interactive computational environment  
combining code, text, figures, ...

<http://ipython.org/notebook.html>

# How to get the code?

---



The code is made available through github:

[https://github.com/ctroupin/OceanData\\_NoteBooks](https://github.com/ctroupin/OceanData_NoteBooks)

# How to get the code?



The code is made available through github:  
[https://github.com/ctroupin/OceanData\\_NoteBooks](https://github.com/ctroupin/OceanData_NoteBooks)

The screenshot shows the GitHub repository page for `ctroupin / OceanData_NoteBooks`. At the top, there are navigation options: "Unwatch", "Star" (0), and "Fork" (1). Below this, the repository title is displayed along with statistics: 12 commits, 1 branch, 0 releases, and 2 contributors. The current branch is `master`. A table lists the repository's files and their commit history:

File	Commit	Time
<code>LICENSE</code>	Initial commit	3 months ago
<code>Plot_TimeSeries1.ipynb</code>	Various small changes	3 months ago
<code>README.md</code>	modified readme	3 months ago
<code>Read_CORA_dataset.ipynb</code>	Modified text	2 months ago
<code>Read_TimeSeries_1.ipynb</code>	First commit	2 months ago
<code>Read_TimeSeries_2.ipynb</code>	First commit	2 months ago
<code>Read_TimeSeries_3.ipynb</code>	First commit	2 months ago
<code>Read_drifter_data_1.ipynb</code>	Text corrections	2 months ago
<code>Read_drifter_data_2.ipynb</code>	First commit	2 months ago
<code>Read_drifter_data_3.ipynb</code>	First commit	2 months ago

Below the file list, the `README.md` content is visible:

## OceanData\_NoteBooks

Examples of data processing with python notebooks using netCDF files.

On the right side of the repository page, there is a sidebar with navigation options: Code, Issues (0), Pull requests (0), Wiki, Pulse, Graphs, and Settings. At the bottom of the sidebar, there is an SSH clone URL: `git@github.com:ctroupin/OceanData_NoteBooks`, and a "Download ZIP" button.

# How to get the code?

---



1. Download the zipped archive on your computer  
(in `~/CMEMS_INSTAC_Training`)



# How to get the code?

---



1. Download the zipped archive on your computer  
(in `~/CMEMS_INSTAC_Training`)
2. Extract the archive

---

```
unzip OceanData_NoteBooks-master.zip
```

---

# How to get the code?

---



1. Download the zipped archive on your computer  
(in `~/CMEMS_INSTAC_Training`)
2. Extract the archive

---

```
unzip OceanData_NoteBooks-master.zip
```

---

3. Go in the main directory

---

```
cd ~/CMEMS_INSTAC_Training/OceanData_NoteBooks-master/
```

---

# How to run a notebook?

---



1. Download the zipped archive on your computer  
(in `~/CMEMS_INSTAC_Training`)
2. Extract the archive

---

```
unzip OceanData_NoteBooks-master.zip
```

---

3. Go in the main directory

---

```
cd ~/CMEMS_INSTAC_Training/OceanData_NoteBooks-master/
```

---

4. In a terminal, type

---

```
ipython notebook Read_TimeSeries_1.ipynb
```

---

# How to run a notebook?



1. Download the zipped archive on your computer  
(in `~/CMEMS_INSTAC_Training`)

2. Extract the archive

```
unzip OceanData.NoteBooks-master.zip
```

3. Go in the main directory

```
cd ~/CMEMS_INSTAC_Training/OceanData.NoteBooks-master/
```

4. In a terminal, type

```
ipython notebook Read_TimeSeries_1.ipynb
```

You should obtain something like that:

```
ipython Read_TimeSeries_1.ipynb
```

The goal is to see how we can read the data contained in a netCDF file. However, possibilities will be explained.

### 1. Reading a local file

Leta's assume we have downloaded a file from [Copernicus](#). We define the directory and the file name. adapt them to be adapted according to your case.

```
In [1]: import os
import sys
from netCDF4 import Dataset

# Define the path to the file
file_path = "/home/your_username/Downloads/Read_TimeSeries_1.ipynb"

# Read the file
with open(file_path, 'r') as f:
    data = f.read()

print(data)
```

where the first argument of the file and 'r' indicates that it's open for reading ('r' would be used for writing); we continue with the information about the dataset:

- Read the file's metadata
- Open the file
- Read the data

Jupyter Read\_TimeSeries\_1 (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CMEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4 interface](#) for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date_update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

# Structure of a notebook



Jupyter Read\_TimeSeries\_1 (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CIEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4](#) interface for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date_update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

Run current cell

# Structure of a notebook



**Jupyter** Read\_TimeSeries\_1 (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CMEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4](#) interface for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date_update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

Run current cell  
Add a new cell

# Structure of a notebook



Jupyter Read\_TimeSeries\_1 (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CIEMES](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4 interface](#) for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
```

```
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date_update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

Run current cell  
Add a new cell  
Select type of cell



# Structure of a notebook



Jupyter Read\_TimeSeries\_1 (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CMEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4 interface](#) for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
```

```
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

Run current cell  
Add a new cell  
Select type of cell  
Code cell

# Structure of a notebook



Jupyter Read\_TimeSeries\_1 (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

## 1. Reading a local file

Let's assume we have downloaded a file from [CMEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_M0_61198.nc"
```

To read the file we need the [netCDF4 interface](#) for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

Where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing). `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

```
In [3]: ds
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3 CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date_update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo platform code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

Run current cell  
Add a new cell  
Select type of cell  
Code cell  
Text cell

# Structure of a repository

---



In the directory containing the notebooks, type:

---

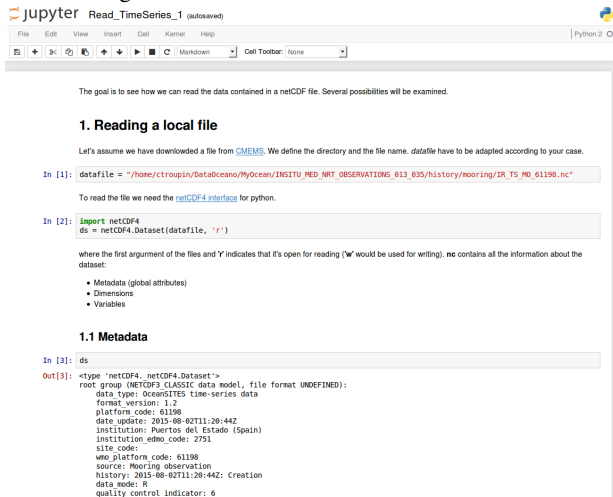
```
ipython notebook
```

---

## In the directory containing the notebooks, type:

```
ipython notebook
```

## You should get:



The screenshot shows a Jupyter Notebook window titled "Read\_TimeSeries\_1 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The notebook content is as follows:

The goal is to see how we can read the data contained in a netCDF file. Several possibilities will be examined.

### 1. Reading a local file

Let's assume we have downloaded a file from [CMEMS](#). We define the directory and the file name. `datafile` have to be adapted according to your case.

```
In [1]: datafile = "/home/ctroupin/DataOceano/MyOcean/INSITU_MED_NRT_OBSERVATIONS_013_035/history/mooring/IR_TS_MD_61198.nc"
```

To read the file we need the [netCDF4 interface](#) for python.

```
In [2]: import netCDF4
ds = netCDF4.Dataset(datafile, 'r')
```

where the first argument of the files and 'r' indicates that it's open for reading ('w' would be used for writing), `nc` contains all the information about the dataset:

- Metadata (global attributes)
- Dimensions
- Variables

### 1.1 Metadata

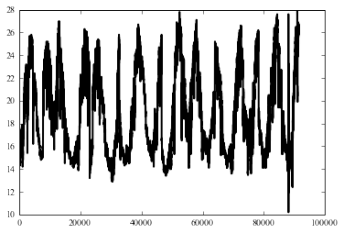
```
In [3]: ds
Out[3]: <type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3_CLASSIC data model, file format UNDEFINED):
  data type: OceanSITES time-series data
  format version: 1.2
  platform code: 61198
  date update: 2015-08-02T11:20:44Z
  institution: Puertos del Estado (Spain)
  institution_edmo_code: 2751
  site code:
  wmo_platform_code: 61198
  source: Mooring observation
  history: 2015-08-02T11:20:44Z: Creation
  data mode: R
  quality_control_indicator: 6
```

# What's inside the repository?

---



Read\_TimeSeries\_1.ipynb: reading a local `netCDF` file

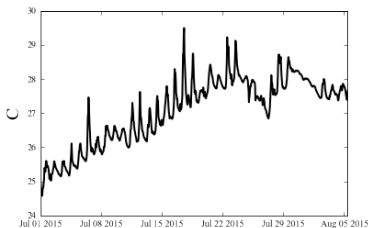


# What's inside the repository?



Read\_TimeSeries\_1.ipynb: reading a local **netCDF** file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol



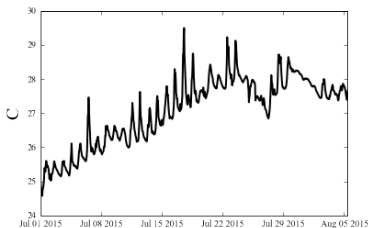
# What's inside the repository?



Read\_TimeSeries\_1.ipynb: reading a local **netCDF** file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol

Read\_TimeSeries\_3.ipynb: reading a netCDF using the **CF module**



# What's inside the repository?

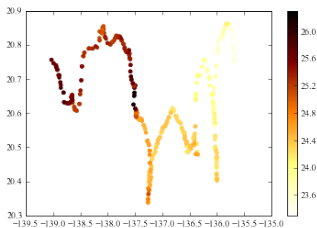


Read\_TimeSeries\_1.ipynb: reading a local netCDF file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using OPeNDAP protocol

Read\_TimeSeries\_3.ipynb: reading a netCDF using the CF module

Read\_drifter\_data\_1.ipynb: basic plot of a drifter trajectory





# What's inside the repository?



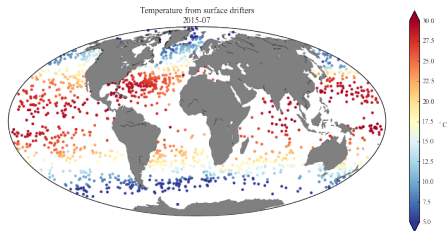
Read\_TimeSeries\_1.ipynb: reading a local netCDF file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using OPeNDAP protocol

Read\_TimeSeries\_3.ipynb: reading a netCDF using the CF module

Read\_drifter\_data\_1.ipynb: basic plot of a drifter trajectory

Read\_drifter\_data\_2.ipynb: plotting temperature observations from drifters



# What's inside the repository?



Read\_TimeSeries\_1.ipynb: reading a local **netCDF** file

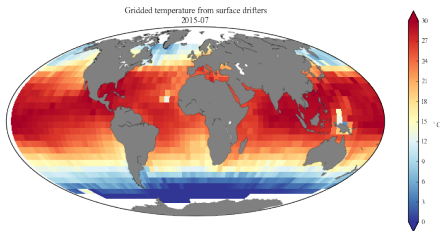
Read\_TimeSeries\_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol

Read\_TimeSeries\_3.ipynb: reading a netCDF using the **CF module**

Read\_drifter\_data\_1.ipynb: basic plot of a drifter trajectory

Read\_drifter\_data\_2.ipynb: plotting temperature observations from drifters

Read\_drifter\_data\_3.ipynb: gridding temperature observations from drifters



# What's inside the repository?



Read\_TimeSeries\_1.ipynb: reading a local **netCDF** file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol

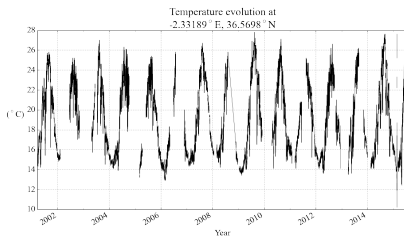
Read\_TimeSeries\_3.ipynb: reading a netCDF using the **CF module**

Read\_drifter\_data\_1.ipynb: basic plot of a drifter trajectory

Read\_drifter\_data\_2.ipynb: plotting temperature observations from drifters

Read\_drifter\_data\_3.ipynb: gridding temperature observations from drifters

Plot\_TimeSeries1.ipynb: plotting temperature from a mooring



# What's inside the repository?



Read\_TimeSeries\_1.ipynb: reading a local **netCDF** file

Read\_TimeSeries\_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol

Read\_TimeSeries\_3.ipynb: reading a netCDF using the **CF module**

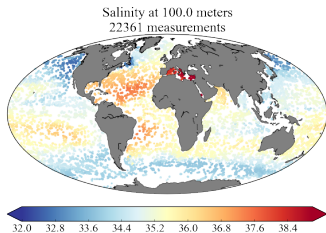
Read\_drifter\_data\_1.ipynb: basic plot of a drifter trajectory

Read\_drifter\_data\_2.ipynb: plotting temperature observations from drifters

Read\_drifter\_data\_3.ipynb: gridding temperature observations from drifters

Plot\_TimeSeries1.ipynb: plotting temperature from a mooring

Read\_CORA\_dataset.ipynb: reading and plotting data from **CORA** dataset



# Example: plotting a time series

---



**Notebook file:** `Plot_TimeSeries1.ipynb`

**Product:** Mediterranean Sea near real-time observations  
(INSITU\_MED\_NRT\_OBSERVATIONS\_013\_035)

**Data file:** `IR_TS_MO_61198.nc` Mooring managed by Puertos del Estado (Spain)

# Example: plotting a time series

---



**Notebook file:** Plot\_TimeSeries1.ipynb

**Product:** Mediterranean Sea near real-time observations  
(INSITU\_MED\_NRT\_OBSERVATIONS\_013\_035)

**Data file:** IR\_TS\_MO\_61198.nc Mooring managed by Puertos del Estado (Spain)

- Objectives:**
1. Read a netCDF file
  2. Apply the quality flags to the observations
  3. Generate high-quality plot

# Example: plotting a time series



Notebook file: `Plot_TimeSeries1.ipynb`

Product: Mediterranean Sea near real-time observations  
(`INSITU_MED_NRT_OBSERVATIONS_013_035`)

Data file: `IR_TS_MO_61198.nc` Mooring managed by Puertos del Estado (Spain)

- Objectives:
1. Read a netCDF file
  2. Apply the quality flags to the observations
  3. Generate high-quality plot

