



marine.copernicus.eu



1st CMEMS MED User & Training Workshop

In Situ TAC

Charles Troupin, Antonis Chalkiopoulos

SOCIB, HCMR

La Spezia, 4 December 2015

1. Introduction
2. How to get the data?
3. How to work with the data?
4. Ocean Data View
5. Python

1. Introduction

1.1– Data quality

1.2– About the material

2. How to get the data?

3. How to work with the data?

3.1– Inspection

3.2– Visualisation

3.3– Processing

4. Ocean Data View

4.1– Objective 1: time series

4.2– Objective 2: CORA dataset

5. Python

5.1– ipython notebooks

5.2– Example 1: plotting

*”Without sufficient observations,
useful prediction will likely never be
possible.”*

*”Models will evolve and improve,
but, without data, will be untestable,
and observations not taken today are
lost forever.”*

C. Wunsch et al. (2010) PNAS

Why in situ data?



1. Model initialisation
2. Model validation
3. Data assimilation
4. Satellite cannot see below the surface

models are idealisation of the reality

Why in situ data?



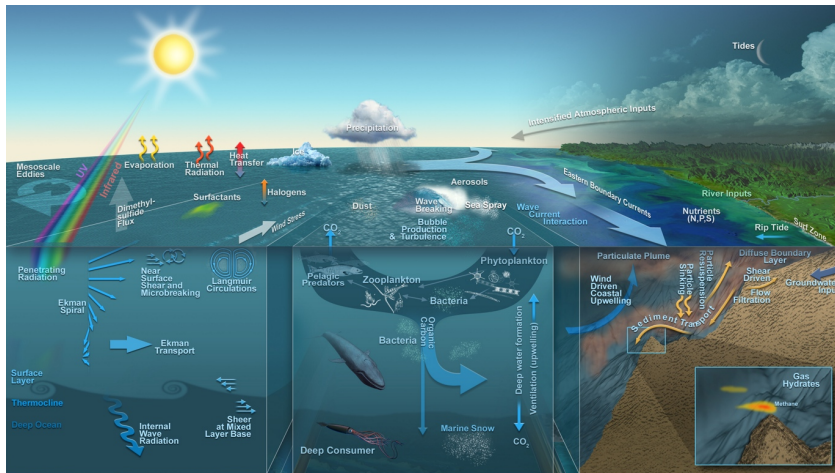
1. Model initialisation
2. Model validation
3. Data assimilation
4. Satellite cannot see below the surface

models are idealisation of the reality

”Without data assimilation, any attempt to produce reliable forecasts is almost certain to end in failure.”

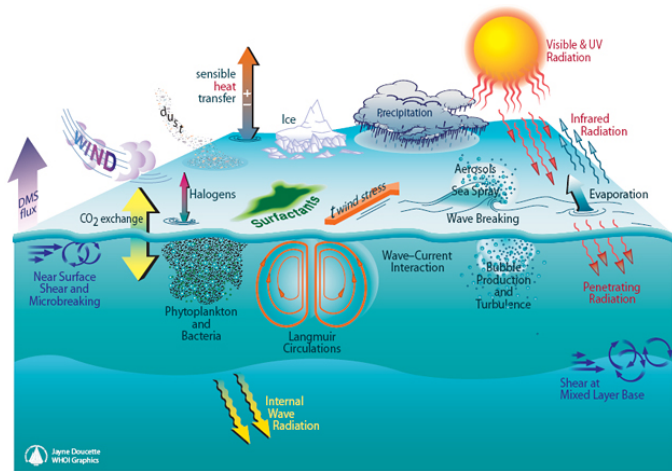
<http://www.metoffice.gov.uk/learning/science/first-steps>

The ocean is complex



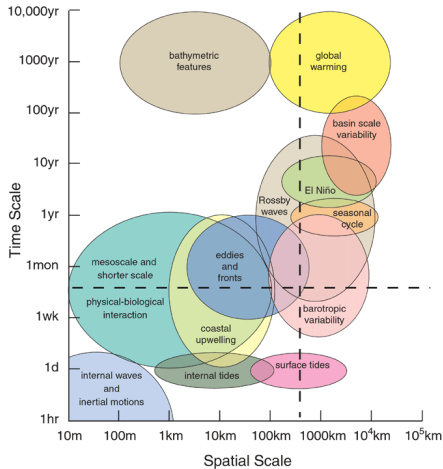
Many processes

The ocean is complex



Many processes

The ocean is complex



Many processes and many scales

A multi-platform approach is essential



”We must be able to document conditions and measure fluxes within the volume of the ocean, simultaneously and in real time, over many scales of time and space, regardless of the depth, energy, mobility, or complexity of the processes involved.”

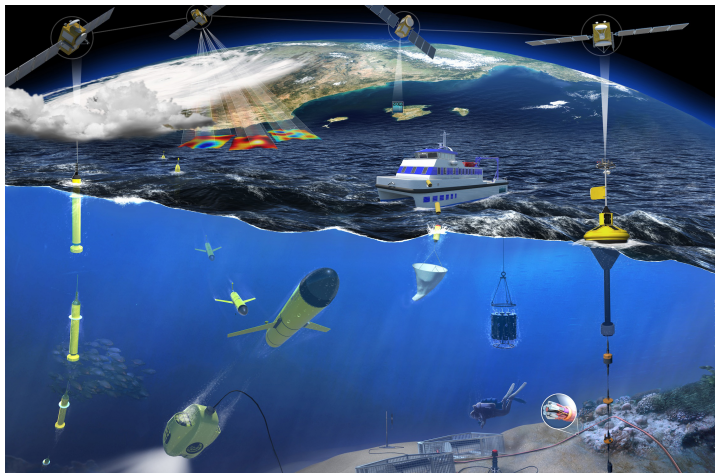
Delaney and Bargas (2009)

A multi-platform approach is essential



Credit: Global Ocean Observing System Office (IOC-GOOS)

A multi-platform approach is essential



Balearic Islands Coastal Ocean Observing and Forecasting System

www.socib.es

A multi-platform approach is essential



Coastal Observing System for Northern and Arctic Seas

<http://codm.hzg.de/codm/>

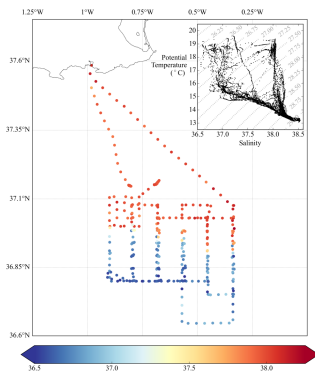
Types of in situ data



Research Vessel

temperature, salinity, currents, oxygen, ...

Feature type: trajectory of profiles for CTD
trajectory for thermosalinograph



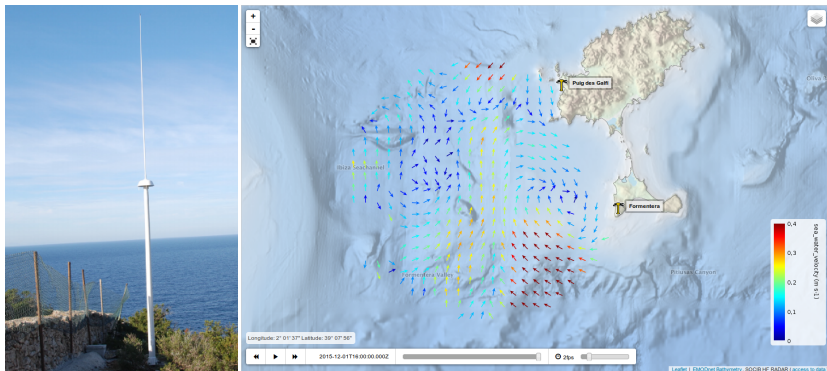
Types of in situ data



Coastal HF Radar

Current speed and direction

Feature type: grid



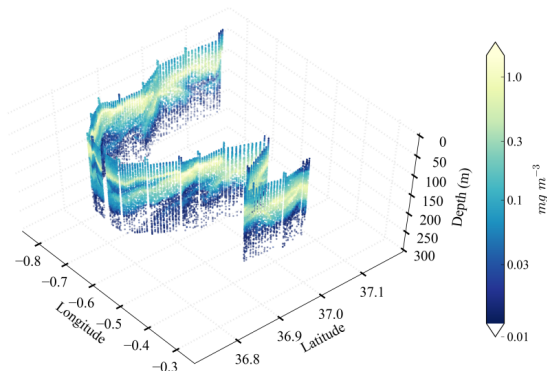
Types of in situ data



Glider

Temperature, salinity, currents, chlorophyll, ...

Feature type: trajectory



Types of in situ data



Drifting buoys and profilers

Temperature, salinity, currents, ...

Feature type: trajectory and trajectory of profiles



Drifters, November 2015



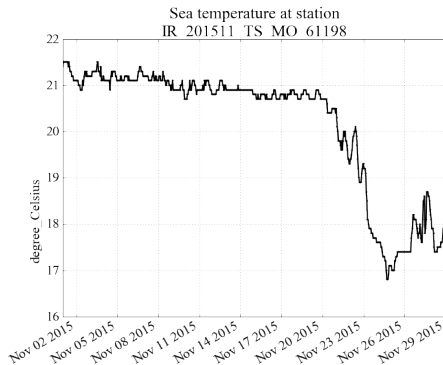
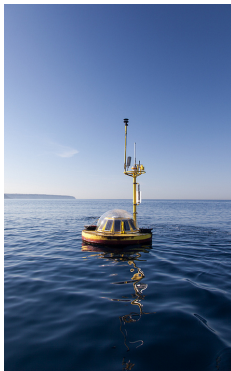
Types of in situ data



Fixed stations

Sea-level, weather/water column variables

Feature type: time series



Data quality

Why data are not always good?



Why data are not always good?



Why data are not always good?



1. variety of instruments
→ different precision, accuracy and methods
2. a given variable should undergo common QC
with testing depending on the instrument/platform
3. needs for standards indicating reliability
4. needs for easily found documentation of the test procedures
5. original values must be preserved
6. problems found by users → reported back to the provider

Quality flags are stored in the netCDF files



Example: temperature from a profiler:

```
...
float TEMP(TIME, DEPTH) ;
  TEMP:long_name = "Sea temperature" ;
  TEMP:standard_name = "sea_water_temperature" ;
  TEMP:units = "degree_Celsius" ;
  TEMP:_FillValue = 9.96921e+36f ;
byte TEMP_QC(TIME, DEPTH) ;
  TEMP_QC:long_name = "quality flag" ;
  TEMP_QC:conventions = "OceanSites reference table 2" ;
  TEMP_QC:_FillValue = -128b ;
  TEMP_QC:valid_min = 0b ;
  TEMP_QC:valid_max = 9b ;
  TEMP_QC:flag_values = 0b, 1b, 2b, 3b, 4b, 5b, 6b, 7b, 8b, 9b ;
  TEMP_QC:flag_meanings = "no-qc-performed good_data probably-good_data
  bad_data_that_are_potentially_correctable bad_data value_changed
  not_used nominal_value interpolated_value missing_value" ;
...
```

Quality flag meaning



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

Quality flag meaning



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

In most situations: only use data with flag=1

Real-time QC cannot detect all the anomalies

- ▶ Real-time QC automatic tests thresholds are a compromise between:
 1. letting bad data going through and
 2. stopping good data
- ▶ Delayed mode QC implies visual inspection by an operator

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test
- ▶ Density Inversion

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test
- ▶ Density Inversion
- ▶ Grey List

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test
- ▶ Density Inversion
- ▶ Grey List
- ▶ Gross salinity or temperature sensor drift

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test
- ▶ Density Inversion
- ▶ Grey List
- ▶ Gross salinity or temperature sensor drift
- ▶ Frozen profile

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

Tests applied on Argo vertical profiles



- ▶ Deepest Pressure Test
- ▶ Platform Identification
- ▶ Impossible Date Test
- ▶ Impossible Location Test
- ▶ Position on Land Test
- ▶ Impossible Speed Test
- ▶ Global Range Test
- ▶ Regional Range Test
- ▶ Pressure Increasing Test
- ▶ Spike Test
- ▶ Gradient Test
- ▶ Digit Rollover Test
- ▶ Stuck Value Test
- ▶ Density Inversion
- ▶ Grey List
- ▶ Gross salinity or temperature sensor drift
- ▶ Frozen profile
- ▶ Visual QC

More details: doi:[10.13155/33951](https://doi.org/10.13155/33951)

1. Various types of platforms available

1. Various types of platforms available
2. Quality flags assigned to the measurements

1. Various types of platforms available
2. Quality flags assigned to the measurements
3. In situ data are essential for numerical model

1. Various types of platforms available
2. Quality flags assigned to the measurements
3. In situ data are essential for numerical model
4. In situ observations are scarce

Training material

ipython notebooks distributed in github

https://github.com/ctroupin/OceanData_NoteBooks

Examples of data processing in Python using netCDF files. — Edit

11 commits 1 branch 0 releases 2 contributors

Branch: master OceanData_NoteBooks / +

Text corrections		
ctroupin authored 18 days ago		latest commit acf4d358c8
LICENSE	Initial commit	a month ago
Plot_TimeSeries1.ipynb	Various small changes	27 days ago
README.md	modified readme	26 days ago
Read_CORA_dataset.ipynb	Various small changes	27 days ago
Read_TimeSeries_1.ipynb	First commit	18 days ago
Read_TimeSeries_2.ipynb	First commit	18 days ago
Read_TimeSeries_3.ipynb	First commit	18 days ago

Code

- Issues
- Pull requests
- Wiki
- Pulse
- Graphs
- Settings

HTTPS clone URL
<https://github.com>

You can clone with HTTPS, SSH,

Why ipython notebooks?



IP[y]: IPython
Interactive Computing

- ▶ User-friendly
- ▶ Free, easy to write, easy to read
- ▶ Code and results visible online via <http://nbviewer.ipython.org>

Why github?



- ▶ Public access, easy to download
- ▶ Collaborative development
- ▶ Bug tracking, feature request, wikis, ...

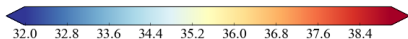
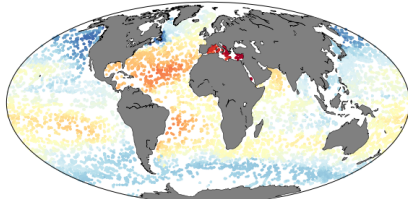
How does it look like?



Finally, the colorbar will be placed below the map.

```
In [157]: fig = plt.figure(figsize=(10,8))
m.scatter(lon p, lat p, s=10, c=salinity_atdepth masked, edgecolor='None', cmap=cmap, norm=norm)
plt.colorbar(scatter, extend='both', orientation='horizontal', pad=0.05)
m.fillcontinents(color='gray', lake_color='white')
m.drawcoastlines(linewidth=0.5)
plt.title('Salinity at ' + str(mydepth) + ' meters \n' + str(goodmeasurements) + ' measurements')
plt.show()
```

Salinity at 100.0 meters
22361 measurements



Even with this type of scatter plot, we can see interesting characteristics of the salinity field.

How to get the
data?

Getting CORA dataset



<http://marine.copernicus.eu>: click on ONLINE CATALOGUE

COPERNICUS MARINE ENVIRONMENT MONITORING SERVICE
Providing PRODUCTS and SERVICES for all marine applications

Search terms OK

ABOUT US | BENEFITS | NEWS | SCIENCE & LEARNING | TRAINING | SERVICES PORTFOLIO

ACCESS TO PRODUCTS
Search and download your datasets!

FIRST VISIT ?

Select your:

AREA

- ▶ GLOBAL OCEAN
- ▶ ARCTIC OCEAN
- ▶ BALTIC SEA
- ▶ EUROPEAN NORTH WEST SHELF SEAS
- ▶ IBERIA-BISCAY-IRELAND REGIONAL SEAS
- ▶ MEDITERRANEAN SEA
- ▶ BLACK SEA

2015
15
OCT

SHORT-CUT TO SERVICES

- REGISTER NOW
- VALIDATION STATISTICS
- ONLINE TUTORIALS
- COLLABORATIVE FORUM

LATEST NEWS FLASH

CMEMS-3211
Issue with upstream data [SST OSTIA | Degraded]
Resolved

ALL NEWS FLASH

PDF CATALOGUE | OBSERVATIONS OVERVIEW | **ONLINE CATALOGUE** | MODELS OVERVIEW

28
MAY

EVENTS AGENDA

PARTNERS AND STAKEHOLDERS

FOCUS ON

TRAINING AGENDA

COLLOQUIUM - 23/27 MAY 2016 - THE 48TH INTERNATIONAL LIÈGE COLLOQUIUM ON OCEAN DYNAMICS

Submesoscale Processes: Mechanisms, Implications and new Frontiers

This colloquium aims to advance our collective understanding of submesoscale processes, their mechanistic functioning, relevance, and implications across a range of oceanic disciplines. Discussions will include observational, modeling and theoretical approaches for studying submesoscale processes.

READ MORE

Scale Processes: Mechanisms, Implications and new Frontiers
International Liège Colloquium on Ocean Dynamics
Liège, Belgium
23rd - 27th May 2016



Select "Global Ocean" and type "CORA" in search box

ONLINE CATALOGUE

CATALOGUE PDF FIRST VISIT ? MY CART 0

CORA SEARCH

GLOBAL OCEAN PHYSICS REANALYSIS GLORYS2V3 (1993-2013)

Numerical-model, Sea-ice, Currents, Sea-level, Salinity, Temperature, Multi-year, Global-ocean

GLOBAL_REANALYSIS_PHYS_001_009



MORE INFO ADD TO CART

You can find here the new Mercator Ocean (Toulouse, FR) GLORYS2V3 (1993-2013) global ocean reanalysis (i.e. one of the four global ocean reanalysis GLOBAL_REANALYSIS_PHYS_001_009, 010, 011 and 017) for the Global Ocean and Sea Ice Physics : monthly means of Temperature, Salinity, Currents, Sea Surface Height and Sea Ice Parameters, at 1/4 degree horizontal resolution, with 75 vertical levels, forced by ERA-Interim atmospheric variables and covering the 1993-2013 time period, with SEEK/IAU Data Assimilation of Temperature and Salinity profiles as well as Sea Level Anomalies, Sea Ice Concentration and Sea Surface Temperature. It also provides with daily means of surface or near surface fields (Sea Surface Temperature, Sea Surface Salinity, Sea Surface Height, currents at depth 0m and 15m, sea ice variables) and 2D diagnostics of mixed layer depth (computed using 3 different criteria) over the 1993-2013 time period.

GLOBAL OCEAN- CORA- IN-SITU OBSERVATIONS YEARLY DELIVERY IN DELAYED MODE (1950-2013)

In-situ-observation, Salinity, Temperature, Multi-year, Global-ocean

INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b



MORE INFO ADD TO CART

For the Global Ocean- In-situ observation yearly delivery in delayed mode. The In Situ delayed mode product designed for reanalysis purposes integrates the best available version of in situ data for temperature and salinity measurements. These data are collected from main global networks (Argo, GOSUD, OceanSITES, World Ocean Database) completed by European data provided by EURO-GOOS regional systems and national system by the regional INS TAC components. It is updated on a yearly basis. The time coverage has been extended in the past by integration of EN4 data for the period 1950-1990.

NEW SEARCH

AREA

- All areas
- Global Ocean (3)
- Arctic Ocean (0)
- Baltic Sea (0)
- European North-West Shelf Seas (0)
- Iberia-Biscay-Ireland Regional Seas (0)
- Mediterranean Sea (0)
- Black Sea (0)

PARAMETER

- All parameters
- Ocean Temperature (3)
- Ocean Salinity (3)
- Ocean Currents (1)
- Sea Ice (1)
- Sea Level (1)
- Winds (0)
- Ocean Optics (0)
- Ocean Chemistry (0)
- Ocean Biology (0)
- Ocean Chlorophyll (0)

TIME COVERAGE

Download product

GLOBAL OCEAN- CORA- IN-SITU OBSERVATIONS YEARLY DELIVERY IN DELAYED MODE (1950-2013)



Metadata provided by CMEMS
Credits: Copernicus Marine Service

[BACK TO SEARCH](#)

[ADD TO
CART](#)



[VIEW
PRODUCT](#)



[DOWNLOAD
PRODUCT](#)



INFORMATION



DOCUMENTATION

SERVICES

PRODUCT IDENTIFIER

INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b

OVERVIEW

For the Global Ocean- In-situ observation yearly delivery in delayed mode. The In Situ delayed mode product designed for reanalysis purposes integrates the best available version of in situ data for temperature and salinity measurements. These data are collected from main global networks (Argo, GOSUD, OceanSITES, World Ocean Database) completed by European data provided by EUROGOOS regional systems and national system by the regional INS TAC components. It is updated on a yearly basis. The time coverage has been extended in the past by integration of EN4 data for the period 1950-1990.

[FULL OVERVIEW](#)

VARIABLES

sea_water_salinity
sea_water_temperature

**GEOGRAPHICAL
COVERAGE**

-180.0



-90.0

Areas:
global-ocean

180.0

Use your username & password

DATA ACCESS

[BACK TO SEARCH](#)

MY CART



INSITU_GLO_TS_REP_O
BSERVATIONS_013_001
_b

DATA ACCESS

Fill your login/password and click on LOGIN to download data.

USERNAME

PASSWORD

LOGIN

If you are not registered yet click on REGISTER

REGISTER


Thank you for using Copernicus Marine
Service products

If you have trouble logging in, make sure your browser is set to
accept cookies.


For security reasons, please Exit your web browser when you quit
services requiring authentication!

FTP access (username & password)

DATA ACCESS BACK TO SEARCH


MY CART 

INSITU_GLO_TS_REP_O
BSERVATIONS_013_001
_b

 DOWNLOAD « BACK TO DATASET SELECTION

FTP

Filtering is not applicable for "FTP Access" (no criteria taken into account).
You can connect to the FTP server with your Copernicus Marine Service credentials
to select dataset files.

 FTP ACCESS

Getting CORA dataset



OA directory

Index of ftp://ftp1.ifremer.fr
/Core/INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b/CORIOLIS-GLOBAL-
CORA04.0-OBS/

↑ Up to higher level directory

Name	Size	Last Modified
 OA		12/30/2013 12:00:00 AM
 RAW		03/20/2014 12:00:00 AM
 gzip		01/31/2014 12:00:00 AM
 readme.txt	2 KB	01/23/2014 12:00:00 AM

Getting CORA dataset



data directory

Index of ftp://ftp1.ifremer.fr
/Core/INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b/CORIOLIS-GLOBAL-
CORA04.0-OBS/OA/

[↑ Up to higher level directory](#)

Name	Size	Last Modified
 data		01/21/2014 12:00:00 AM
 field		01/21/2014 12:00:00 AM

Select year of interest

Index of ftp://ftp1.ifremer.fr
/Core/INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b/CORIOLIS-GLOBAL-
CORA04.0-OBS/OA/data/

























[↑ Up to higher level directory](#)

Name	Size	Last Modified
1990		01/21/2014 12:00:00 AM
1991		01/22/2014 12:00:00 AM
1992		01/21/2014 12:00:00 AM
1993		01/21/2014 12:00:00 AM
1994		01/21/2014 12:00:00 AM
1995		01/21/2014 12:00:00 AM
1996		01/21/2014 12:00:00 AM
1997		01/21/2014 12:00:00 AM
1998		01/21/2014 12:00:00 AM
1999		01/21/2014 12:00:00 AM
2000		01/21/2014 12:00:00 AM
2001		01/21/2014 12:00:00 AM
2002		01/21/2014 12:00:00 AM
2003		01/21/2014 12:00:00 AM
2004		01/21/2014 12:00:00 AM
2005		01/21/2014 12:00:00 AM
2006		01/21/2014 12:00:00 AM
2007		01/21/2014 12:00:00 AM
2008		01/21/2014 12:00:00 AM
2009		01/21/2014 12:00:00 AM
2010		01/21/2014 12:00:00 AM
2011		01/21/2014 12:00:00 AM
2012		01/21/2014 12:00:00 AM

Select month and variable

Index of ftp://ftp1.ifremer.fr
/Core/INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b/CORIOLIS-GLOBAL-
CORA04.0-OBS/OA/data/2012/

⬆ Up to higher level directory

Name	Size	Last Modified
 OA_CORA4.0_20120115_dat_PSAL.nc	140850 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120115_dat_TEMP.nc	151762 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120215_dat_PSAL.nc	137051 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120215_dat_TEMP.nc	152575 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120315_dat_PSAL.nc	140596 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120315_dat_TEMP.nc	157491 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120415_dat_PSAL.nc	145290 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120415_dat_TEMP.nc	158519 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120515_dat_PSAL.nc	146845 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120515_dat_TEMP.nc	159215 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120615_dat_PSAL.nc	148416 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120615_dat_TEMP.nc	162869 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120715_dat_PSAL.nc	153579 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120715_dat_TEMP.nc	166300 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120815_dat_PSAL.nc	163225 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120815_dat_TEMP.nc	177919 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120915_dat_PSAL.nc	167766 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20120915_dat_TEMP.nc	181738 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121015_dat_PSAL.nc	166023 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121015_dat_TEMP.nc	179268 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121115_dat_PSAL.nc	157182 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121115_dat_TEMP.nc	172260 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121215_dat_PSAL.nc	107337 KB	01/22/2014 12:00:00 AM
 OA_CORA4.0_20121215_dat_TEMP.nc	116967 KB	01/22/2014 12:00:00 AM

How to work
with the data?

Quick inspection: ncdump



Home page: <https://www.unidata.ucar.edu/software/netcdf/docs/netcdf/ncdump.html>

What it does: text representation of a netCDF dataset (header information, variables, ...)

ncdump applied on a file

```
ncdump -h 20140628_d-OC.CNR-L3-CHL-MedOC3_A.1KM-MED-DT-v02.nc
```

```
netcdf \20140628_d-OC.CNR-L3-CHL-MedOC3_A.1KM-MED-DT-v02 {
dimensions:
    time = 1 ;
    lat = 1580 ;
    lon = 3308 ;
variables:
    int time(time) ;
    time:long_name = "reference time" ;
    time:standard_name = "time" ;
    time:axis = "T" ;
    time:calendar = "Gregorian" ;
    time:units = "seconds since 1981-01-01 00:00:00" ;
...
    "SUBSAMP=1\n" ,
    "OUTMODE=0\n" ,
    "" ;
}
```



Home page: <http://www.ferret.noaa.gov/Ferret/>

What it does: visualization and analysis environment

Ferret to get basic info on file

```
ctroupin@SCBD046 ~/Desktop $ ferret_c
NOAA/PMEL TMAP
FERRET v6.62
Linux (gfortran) 2.6.9 - 89.0.20.ELsmp - 07/06/13
25-Nov-15 12:23

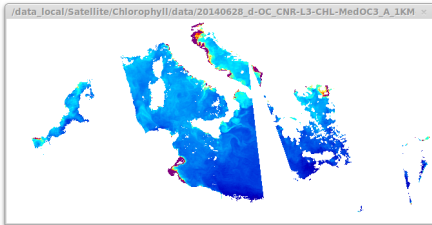
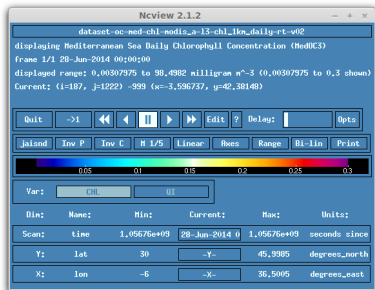
yes? SET DATA 20140628_d-OC.CNR-L3-CHL-MedOC3_A.1KM-MED-DT-v02.nc
yes? SHOW DATA
    currently SET data sets:
1> 20140628_d-OC.CNR-L3-CHL-MedOC3_A.1KM-MED-DT-v02.nc (default)
name      title                                     I           J           K           L
CHL       Mediterranean Sea Daily Chlorop        1:3308      1:1580      ...         1:1
QI        Quality Index of Mediterranean         1:3308      1:1580      ...         1:1

yes?
```



Home page: http://meteora.ucsd.edu/~pierce/ncview_home_page.html

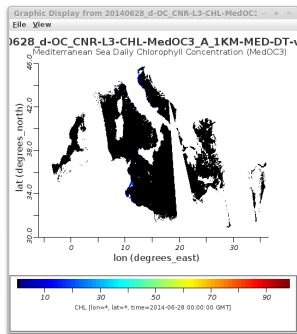
What it does: quick visualisation of 3-4D fields





Home page: <http://www.epic.noaa.gov/java/ncBrowse/>

What is does: interactive graphical display



CHL from 20140628_d-OC_CNRL3-CHL-MedOC3_A_1KM-MED-DT-v02.nc (Domain - + -)

```
float CHL(time=1, lat=1580, lon=3308);
long_name = "Mediterranean Sea Daily Chlorophyll Concentration (MedOC3)";
standard_name = "mass_concentration_of_chlorophyll_a_in_sea_water";
source = "MODIS AQUA - 11A";
type = "surface";
units = "milligram m^-3";
missing_value = -999.0 // float
```

Axes

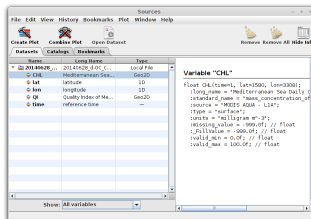
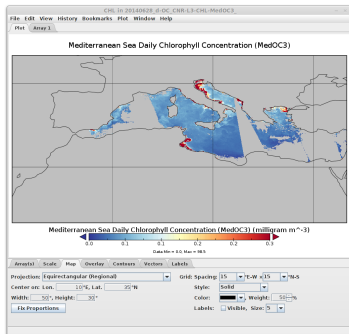
name	units	Dependent Variable		Reverse	Start	End
		x	y			
time	seconds since 1981-01-01 00:00:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2014-06-28 00:00:00	2014-06-28 00:00:00
lat	degrees_north	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	30.0	45.998547
lon	degrees_east	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-5.0	36.50046

Graph Variable Close



Home page: <http://www.giss.nasa.gov/tools/panoply/>

What it does: plot, slice, combine, overlay, ...





Home page: <https://code.zmaw.de/projects/cdo>

What is does: manipulate (merging, averaging) netCDF files (+other formats)

Examples: ▶ Basic info (min, max, avg, size, ...):

```
cdo info input.nc
```

▶ Compute standard deviation:

```
cdo fldstd input.nc output.nc
```



Home page: <http://nco.sourceforge.net/>

What is does: command line operations on netCDF files

Examples: ▶ Average variable over domain:

```
ncwa -O -a lon , lat  input.nc  output.nc
```

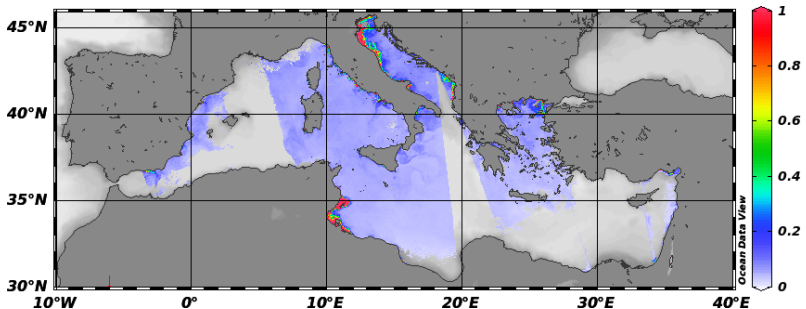
▶ Extract subregion:

```
ncks -d lon ,13. ,18.0 -d lat ,33.0 ,36.0  
input.nc  output.nc
```



Home page: <http://odv.awi.de/en/home/>

What it does: interactive exploration, analysis and visualization of oceanographic data



Wanna know more? Click [here](#)



High-level functions to read/write data from/to a netCDF file:

<http://octave.sourceforge.net/netcdf/overview.html>

<http://es.mathworks.com/help/matlab/network-common-data-form.html>

Example with Octave

```
nc = netcdf('input.nc','r');           % open netcdf file in read-only  
  
CHL = nc{'CHL'}(:);                    % retrieve variable  
CHL_units = nc{'CHL'}.units;          % retrieve the attribute units  
CHL_valid_range = nc{'CHL'}.valid_range; % retrieve the attribute valid_range  
global_history = nc.history;           % retrieve the global attribute history
```



Python interface to the netCDF C library:

<http://unidata.github.io/netcdf4-python/>

Example with ipython

```
In [1]: import netCDF4
In [2]: nc = netCDF4.Dataset('20140628_d-OC_CNR-L3-CHL-MedOC3_A_1KM-MED-DT-v02.nc')
In [3]: print nc
<type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3_CLASSIC data model, file format UNDEFINED):
  Conventions: CF-1.4
  title: dataset-oc-med-chl-modis_a-l3-chl_1km_daily-rt-v02
  references: R. Santoleri, G. Volpe, S. Marullo and B. Buongiorno Nardelli (2008),
  ...
In [4]: CHL = nc.variables['CHL'][:]
In [5]: nc.close()
```

Wanna know more? Click [here](#)

Ocean Data View

> 40000 registered users

Data analysis + visualisation

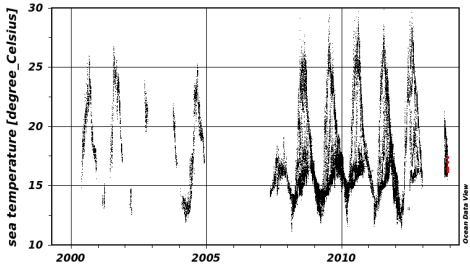
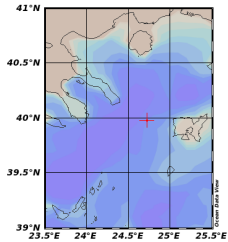
Almost every format supported

Working with ODV on Time Series

Objective: plotting time series



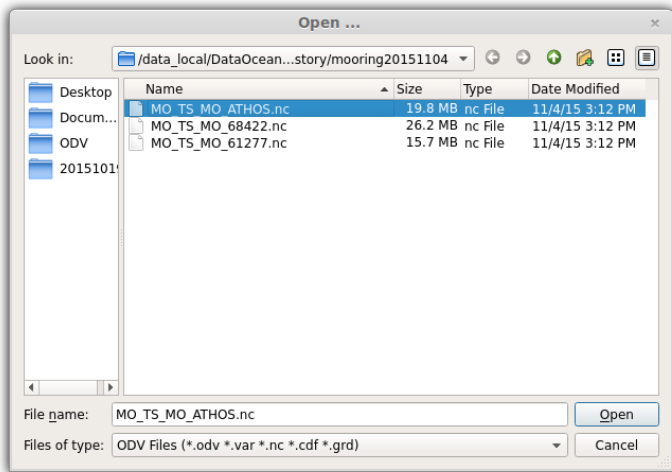
Temperature at mooring Athos



Opening the netCDF file



File → Open → netcdf



Dimension and variables : Next

NetCDF Setup Wizard

Select Dimensions (Step 1 of 4)

NetCDF dimensions

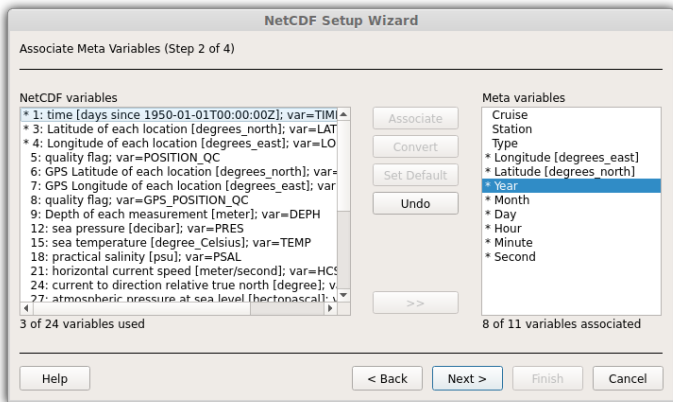
- TIME[36552]
- LATITUDE[36552]
- LONGITUDE[36552]
- POSITION[36552]
- DEPTH[6]

All 5 dimensions selected

Corresponding netCDF variables

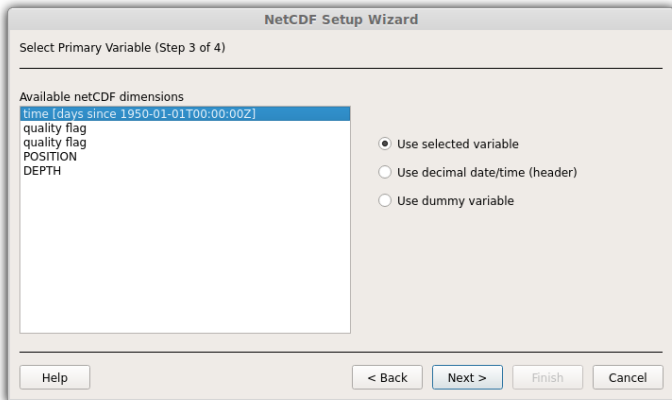
- time [days since 1950-01-01T00:00:00Z]; var=TIME
- Latitude of each location [degrees_north]; var=LATITUDE
- Longitude of each location [degrees_east]; var=LONGITUDE
- quality flag; var=POSITION_QC
- GPS Latitude of each location [degrees_north]; var=GPS_LAT
- GPS Longitude of each location [degrees_east]; var=GPS_LC
- quality flag; var=GPS_POSITION_QC
- Depth of each measurement [meter]; var=DEPH
- 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=HCDDT
- atmospheric pressure at sea level [hectopascal]; var=ATMS
- air temperature in dry bulb [degree_Celsius]; var=DRYT

Variable association : Next

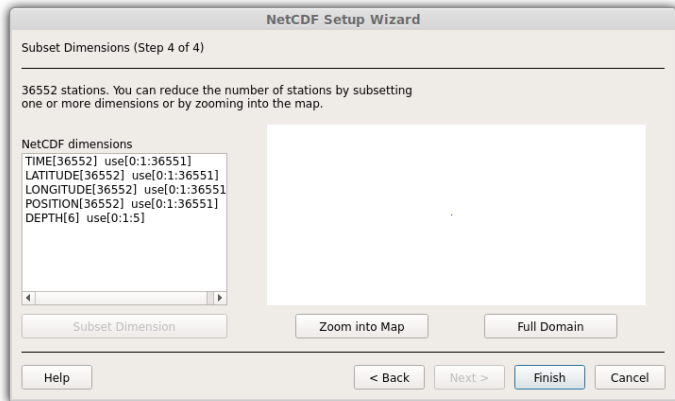


 Quality Control variables not visible at this stage

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, and 'Scatter window' is selected. The main window displays a map of the ocean surface with a grid and a red crosshair. The map shows latitude from 39.0°N to 40°N and longitude from 24.65°E to 24.75°E. The map is titled 'Applies a predefined window layout.' Below the map, there are several empty scatter plot windows, each with a title 'WINDOW n STATION' and a prompt 'Press ENTER to add the data of the current station to the plot.' The right-hand side of the interface shows a 'Station ID: 1' panel with various metadata and a 'timesurface Values' panel with a list of variables and their values.

Station ID: 1

Cruise	MO_TS_MO_ATHOS.rc
Station	1 (R)
Position	34.724°E / 24.874°N
Date	05 May 2000
Time	00:00:00.000
DEPTH Range	30 - 51

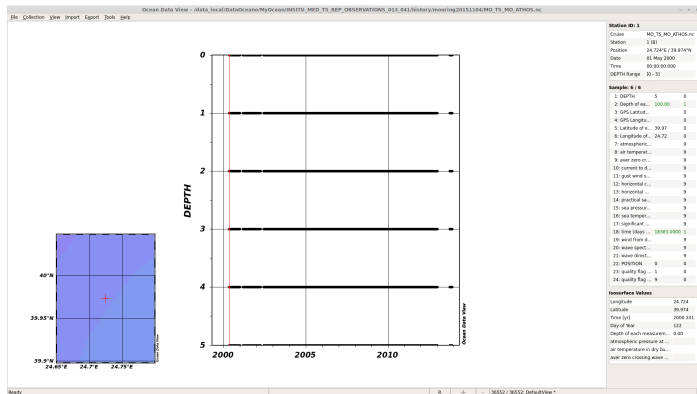
timesurface Values

Longitude	24.724
Latitude	39.974
Time (yr)	2000.331
Day of Year	122
Depth of each measure...	0.00
atmospheric pressure at...	
air temperature in dry be...	
aver sea crossing wave ...	

Plot the time series



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

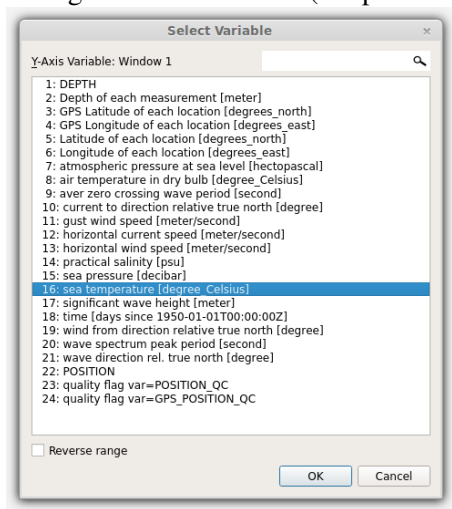


Plot the time series



Right-click on plot:

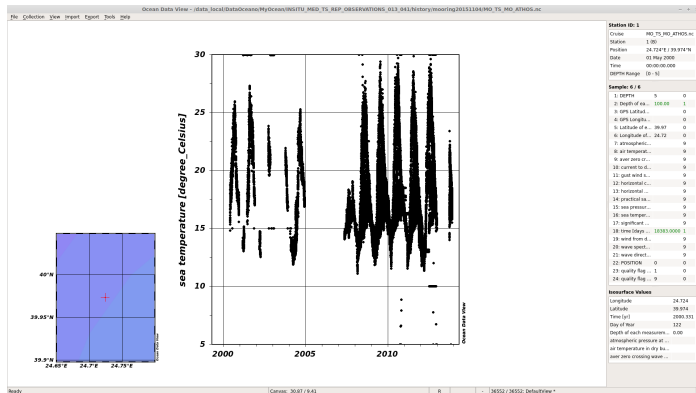
Change X and Y variable (temperature vs. time)



Plot the time series



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

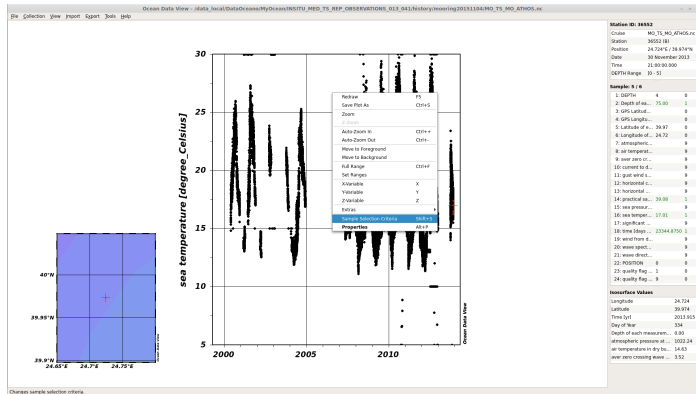


Bad values: will be treated later

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

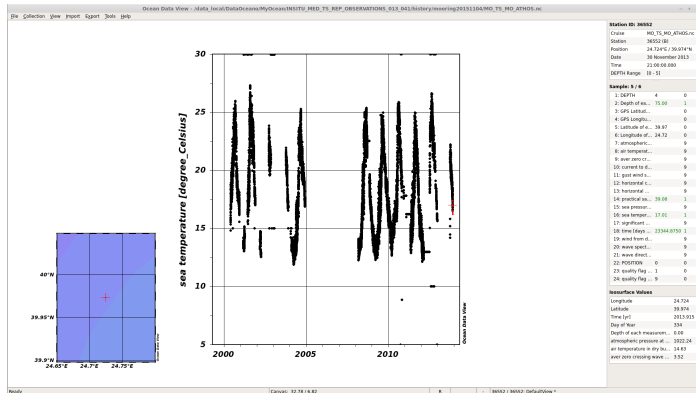
Apply these sample selection criteria globally

Help OK Cancel

Plot the time series



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

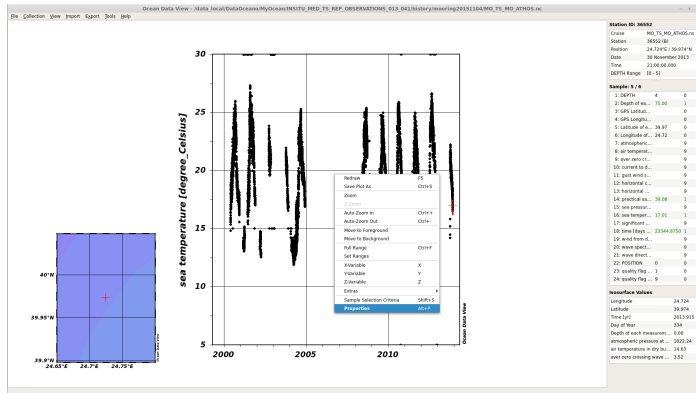


Now we have the series at 2 depth

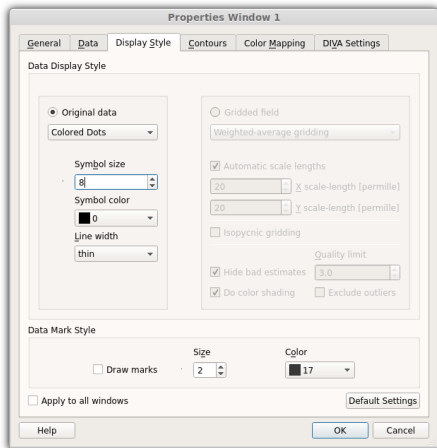
Improve the plot



Right-click on plot: Properties



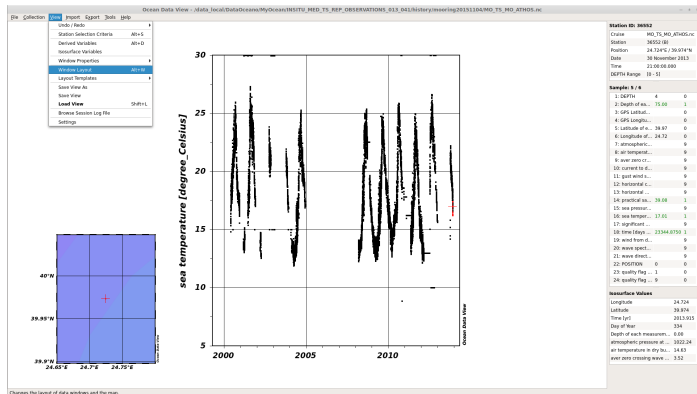
Display Style: modify Symbols Size



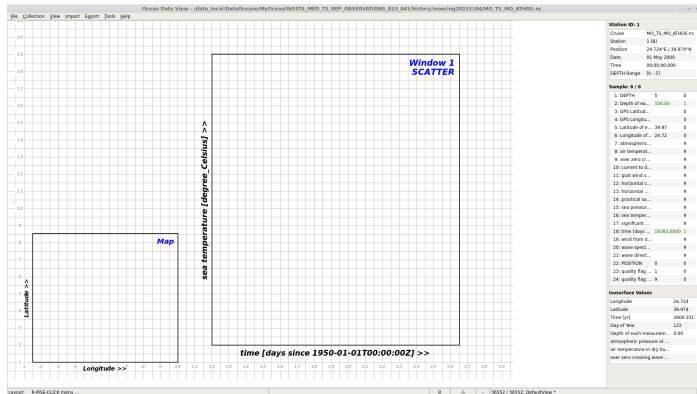
Improve the plot



View → Window Layout




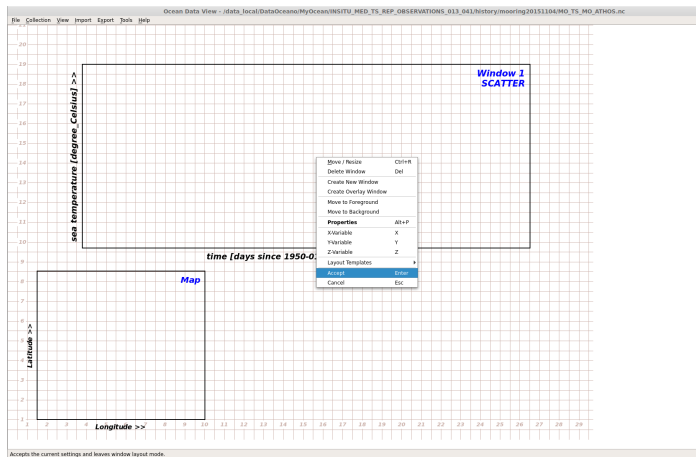
Adapt size of the Scatter window



Improve the plot



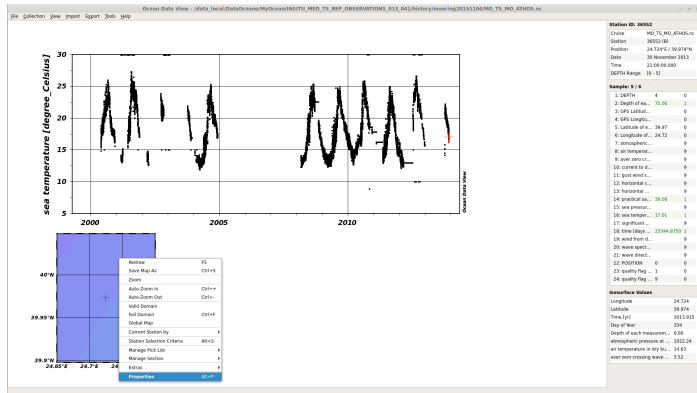
Accept the change (Enter )



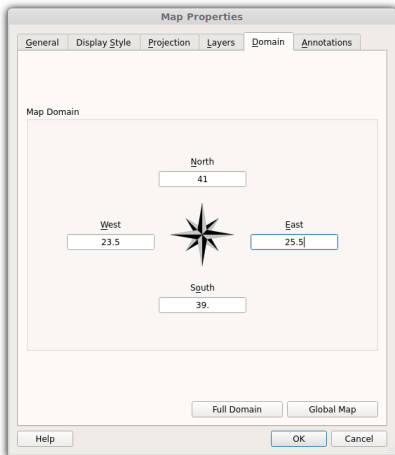
Improve the plot



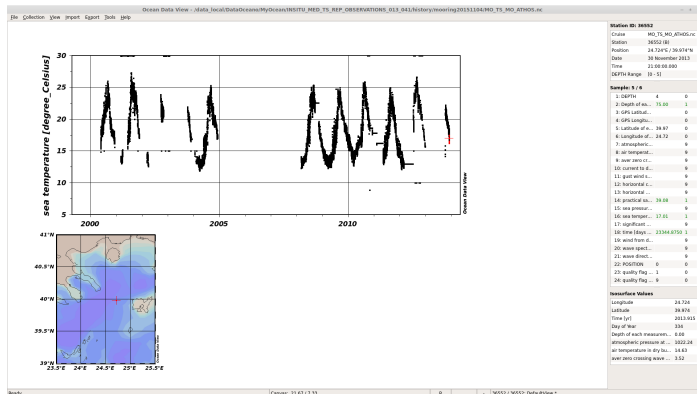
Right-click on plot: Properties → Domain



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

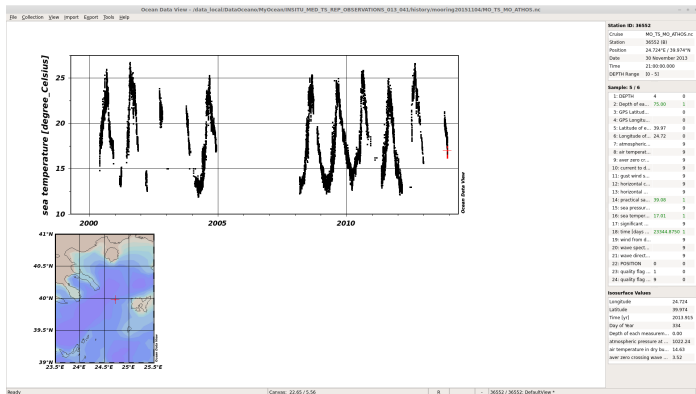
Apply these sample selection criteria globally

Help OK Cancel

Apply quality flags



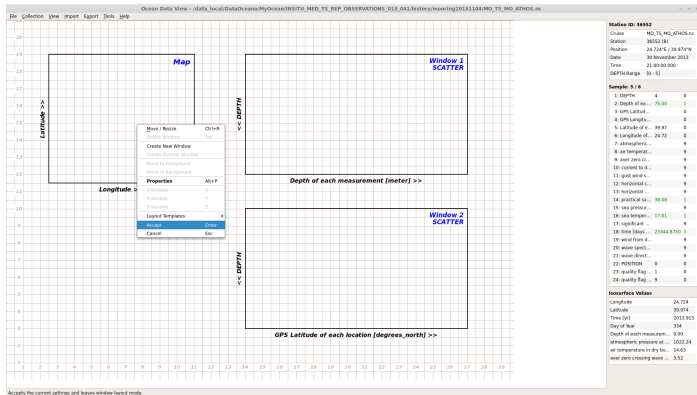
Right-click on plot: Sample Select Criteria → depth range
Select good data only



Apply quality flags



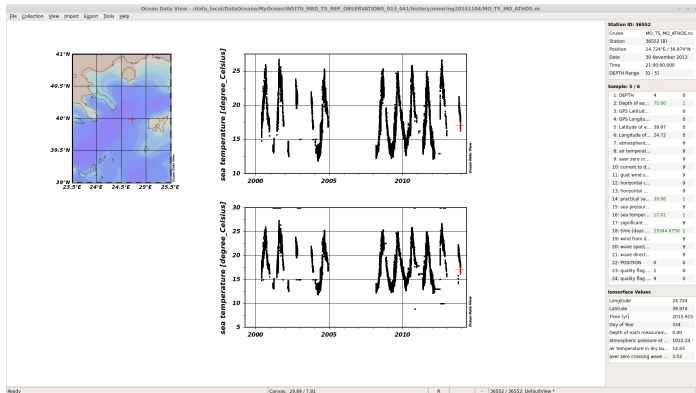
Configure Window Layout to have the 2 time series
(with and without QC)



Apply quality flags



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



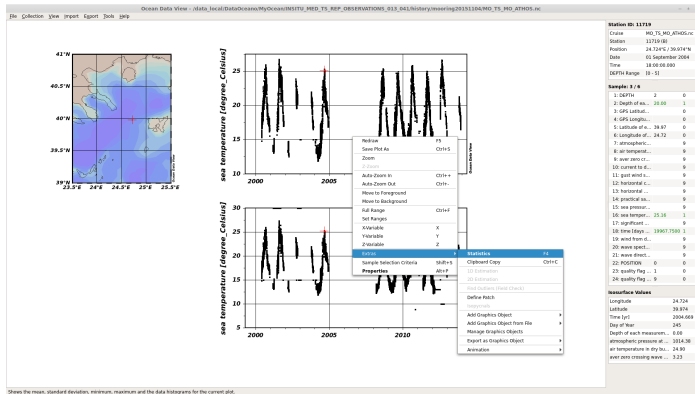
Extreme values are removed

Apply quality flags



Compare histograms:

📄 Right-click on plot → Extra → Statistics

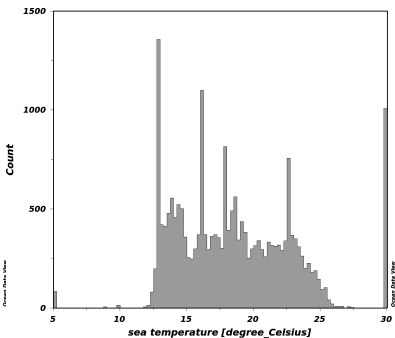
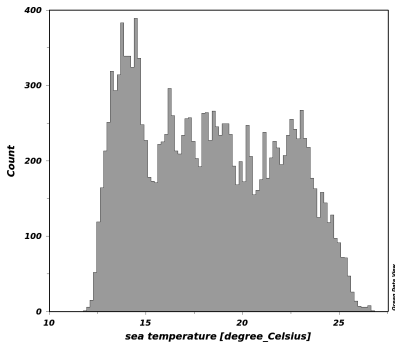


Apply quality flags



Compare histograms:

📄 Right-click on plot → Extra → Statistics



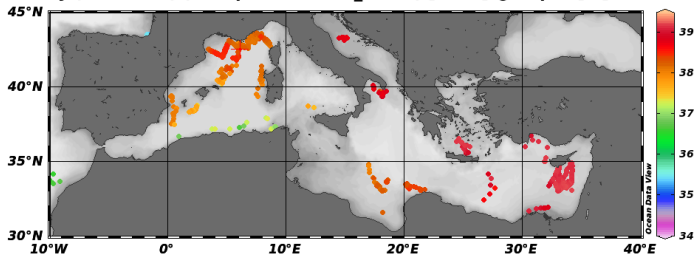
Distribution is improved

Working with ODV on CORA data set

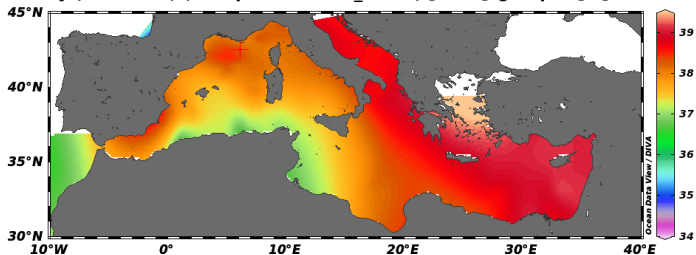
Objective: process CORA dataset



Salinity (S78 - PSS) (interpolated on Z_levels) [none] @ depth [m]=0



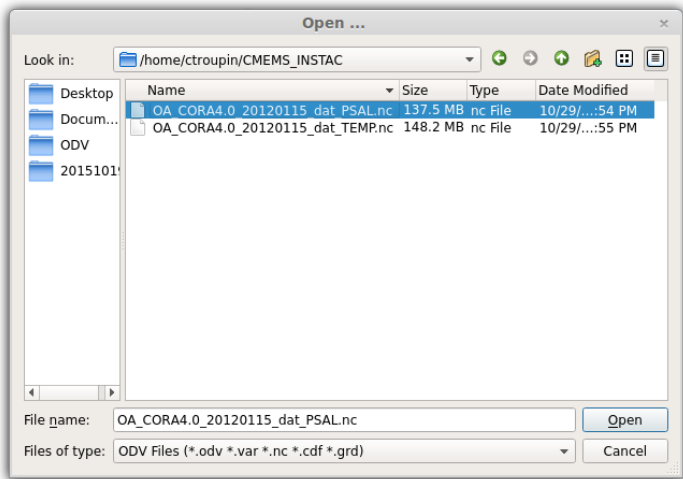
Salinity (S78 - PSS) (interpolated on Z_levels) [none] @ depth [m]=0



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">N_PROF[36038]N_LEVELS[152]	<ul style="list-style-type: none">Cycle number; var=CYCLE_NUMBERJulian day (UTC) relative to REFERENCE_DATE_TIME [days sinceLatitude of the station, best estimate [degree_north]; var=LATLongitude of the station, best estimate [degree_east]; var=LOdepth [m]; var=DEPHprofile processing level; var=PSAL_PROCQuality flag on interpolated variable; var=PSAL_QCSalinity (S78 - PSS) (interpolated on Z_levels) [none]; var=PSAClimatology mean for profile [none]; var=PSAL_CLMNClimatology standard deviation for profile [none]; var=PSAL_CMeasurement error [none]; var=PSAL_ERMEError from unresolved scales [none]; var=PSAL_ERURResidual [none]; var=PSAL_RESIN_PROF; var=N_PROFN_LEVELS; var=N_LEVELS

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**

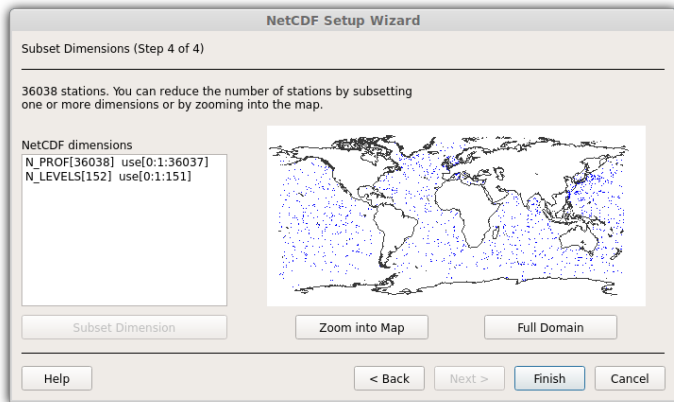
Use selected variable

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/electrosip/CHEM5_INSTAC/OA_COBRA4.0_20120115_dut_PSA.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

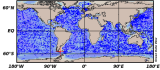
Press ENTER to add the data of the current station to the plot.

Window 6 STATION

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

Climateology standard deviation for profile [msec]

Error from unresolved scales [msec] =>



Climateology mean for profile [msec] =>

latitude to REFERENCE_DATE_TIME [days since REF]

altitude of the station, best estimate [degree_pos]

Station ID: 1

Crane	06_COBRA4_0_20120115_dut_PSA.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: Climateology mean for profile [msec]	34.09	1
4: Climateology standard deviation [msec]	6.134	1
5: Error from unresolved scales [msec]	6.133	1
6: Julian Day (UTC) relative to REF [..]	23221.04069	1
7: Latitude of the station, best est. [..]	-55.1517	1
8: Longitude of the station, best est. [..]	-64.4690	1
9: Measurement error [msec]	0.07	1
10: N_PROF	0	1
11: Quality flag on interpolated var. [..]	1	1
12: Residual [msec]	0.06	1
13: Satiny (519 - 952) Interpolated. [..]	34.09	1
14: profile processing level	1	1
15: depth [m]	0.0000	1

Resurface Values

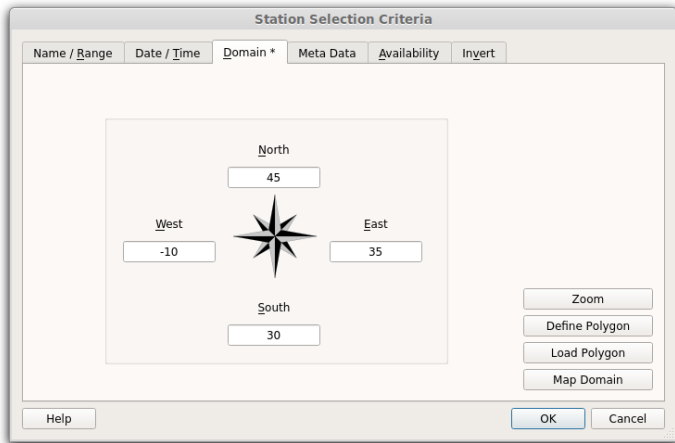
Longitude	-64.469
Latitude	-55.152
Time [yr]	63.931
Day of Year	342
Climateology mean for profile [msec] @ N_LEVELS=first	34.09
Climateology standard deviation for profile [msec] @ N_	6.14
Error from unresolved scales [msec] @ N_LEVELS=first	6.13
Measurement error [msec] @ N_LEVELS=first	0.07

Ready

Region selection and basic statistics



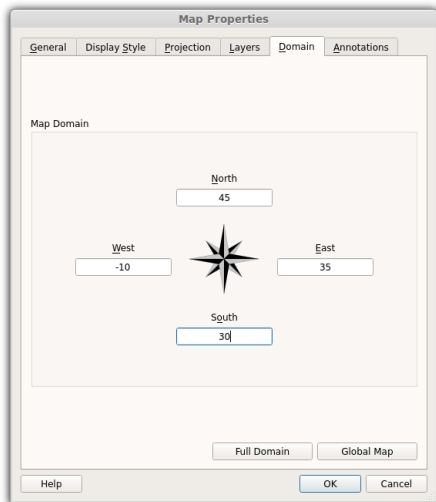
Right-click on image → Station Selection Criteria → Domain



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

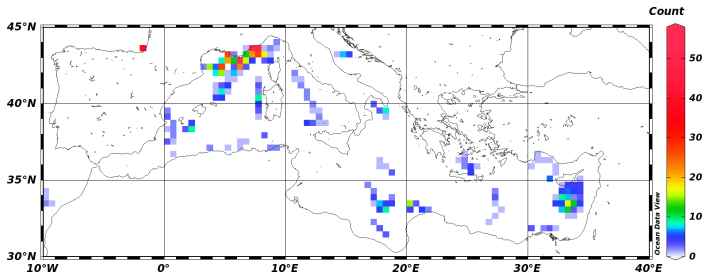
Time Histogram Season Histogram

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):
 - Palette: Odv
 - Background color: (none)
- Font**:
 - Font base size [pt]: (automatic)
 - Font size factor: 100 %
- Axis Style**:
 - Axis color: 0
 - Draw grid

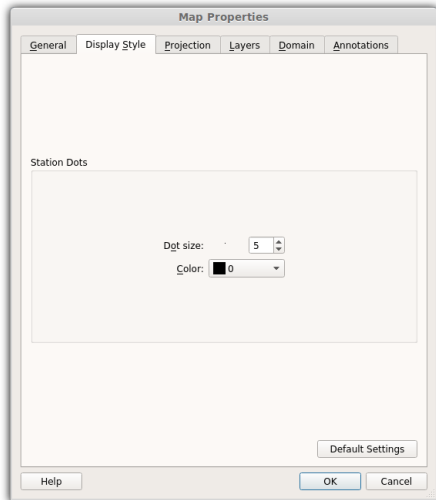
Buttons: Help, OK, Cancel, Default Settings

Map improvement



☞ Right-click on map → **Properties**

Display style: increase dot size, change color

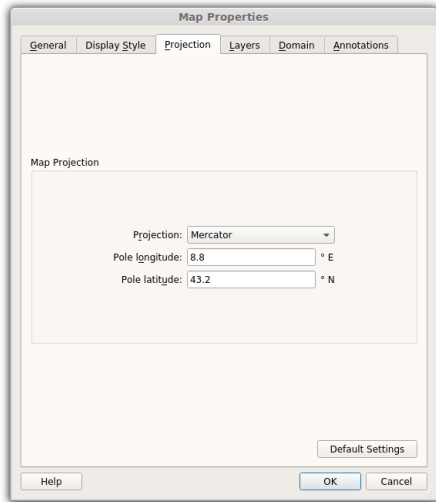


Map improvement



☞ Right-click on map → **Properties**

Projection: modify according to preference

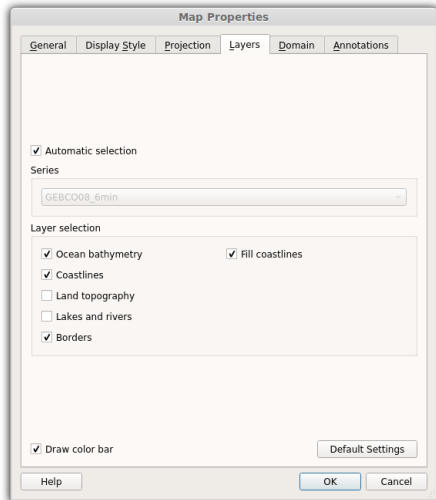


Map improvement



☞ Right-click on map → **Properties**

Layers: bathymetry + coastlines

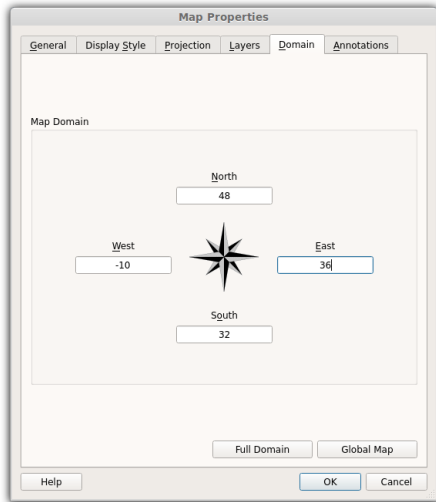


Map improvement



☞ Right-click on map → **Properties**

Domain: adjust limits (already done)

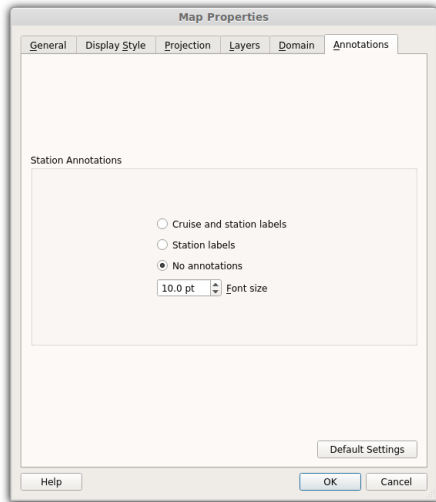


Map improvement



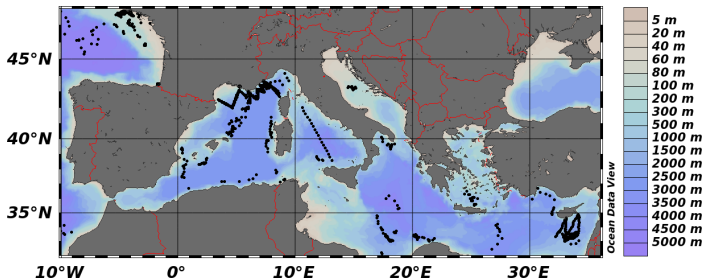
☞ Right-click on map → **Properties**

Annotations: not necessary



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, displaying options such as 'Full Screen Map', '1 STATION window', '2 STATION Windows', '4 STATION Windows', '8 STATION Windows', '16 STATION Windows', '32 STATION Windows', '64 STATION Windows', '128 STATION Windows', '256 STATION Windows', '512 STATION Windows', '1024 STATION Windows', and 'From View File'. The '1 STATION window' option is selected. The main window displays a map of the North Atlantic region, with a station location marked. The station information panel on the right shows the following details:

Station ID: 1

- Cruid: OA-CO944.0_20120115_A
- Station: 1100
- 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: Measurem...	...	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&A.0_20120115_dlat_PXAL.nc

File Collection View Import Export Tools Help

60°N
40°N
20°N
EQ
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 1961
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: Missing (v...	-841	1
13: Salinity STD...	37.28	1
14: profile proc...	2	1
15: depth (m)	0.0000	1

isocurface Values

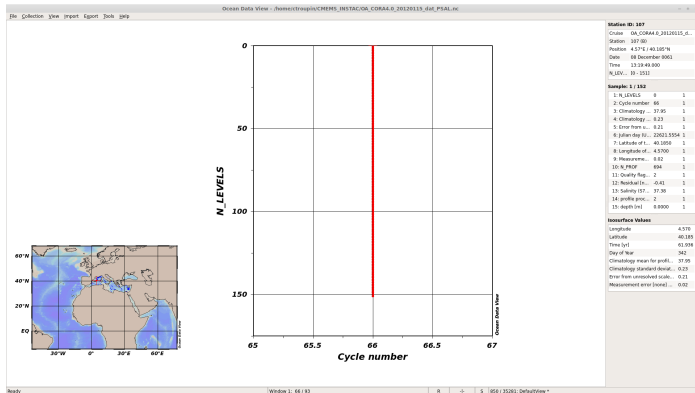
Longitude	4.570
Latitude	40.185
Time (yr)	61.998
Day of year	342
Climatology mean for profile	37.95
Climatology standard deviat...	0.23
Error from unrescaled scale	0.21
Measurement error (scaled) ...	0.02

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

Station window



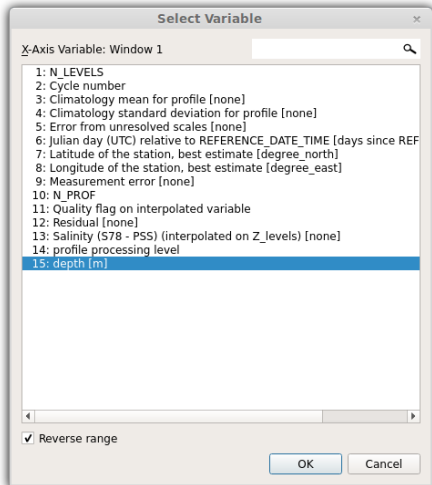
Enter 



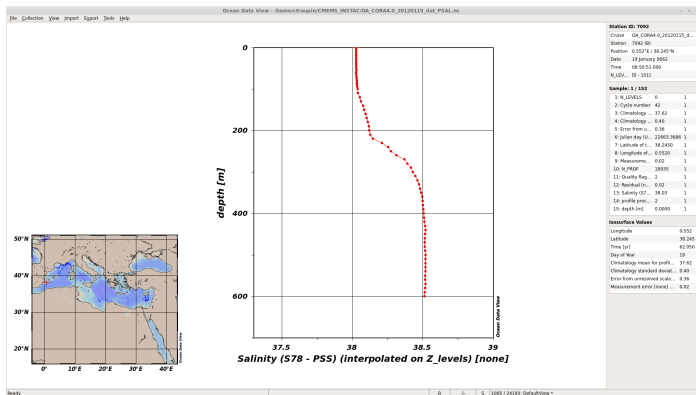
Right-click on map:

Change X and Y variable (salinity vs. depth)

Y variable → Check the reverse range box



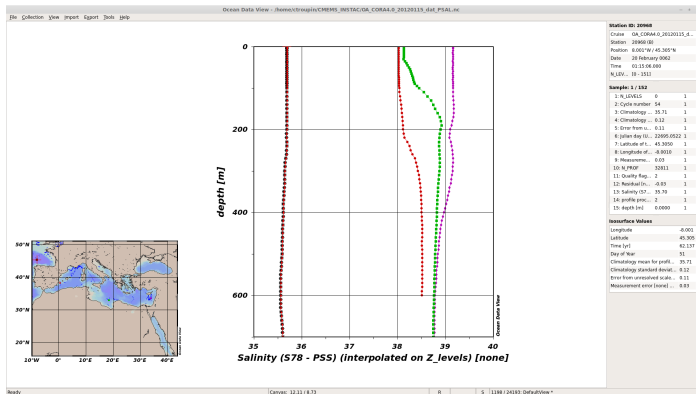
Double click on the map (left) to get profiles at different locations



☒ Very different properties according to the basin

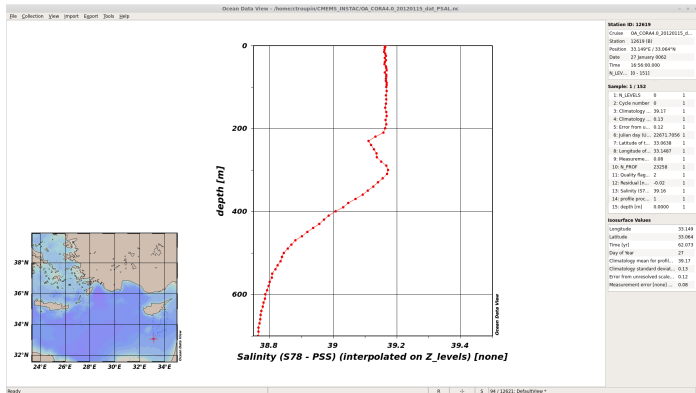
👉 Maybe needed to adjust range, otherwise not visible

Compare profiles in different sub-regions

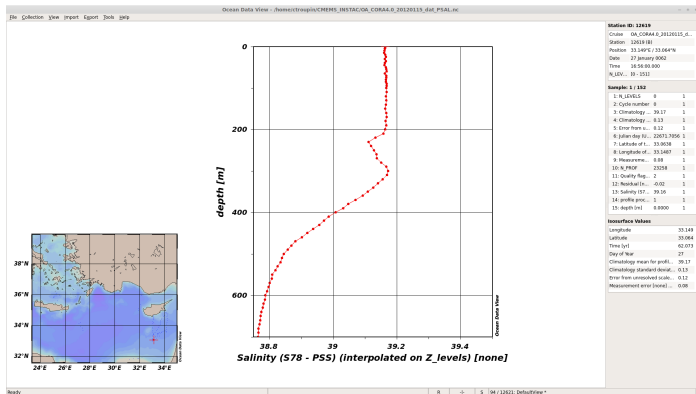


To remove stations: Manage Pick List → Remove all Stations

What happens with this profile south of Cyprus?



What happens with this profile south of Cyprus?

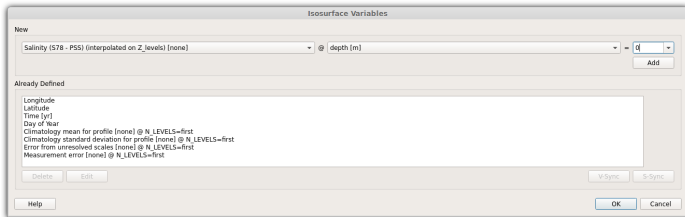


☒ Mixed-layer depth

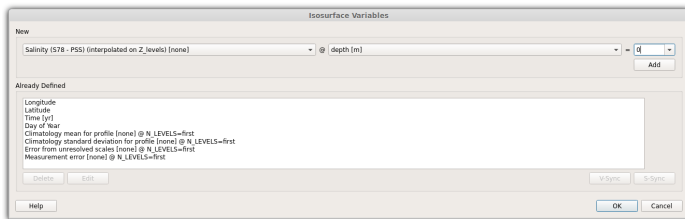
Define new isosurface variables:

View → Isosurface Variables → salinity at depth = 0

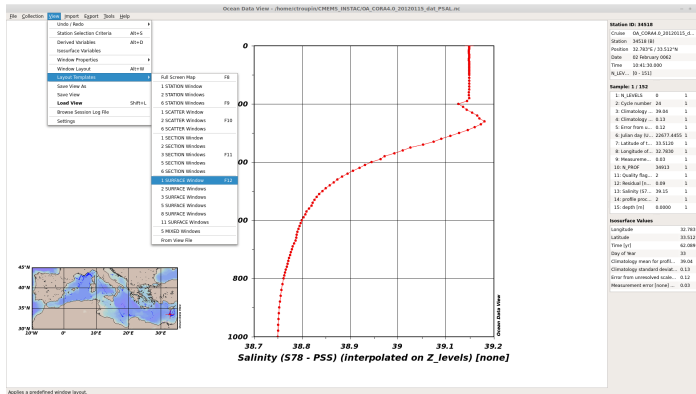
Click on "Add"



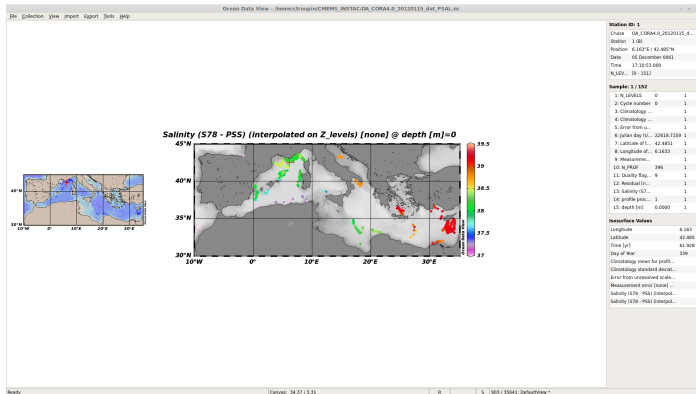
Define new isosurface variables:
Same at depth = 200



View → Layout Template → SURFACE Window



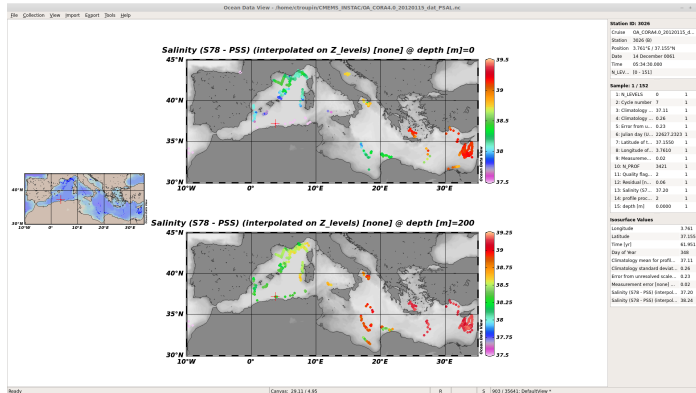
📄 Right-click Z-variable → select newly created variable



☒ higher salinity values in the Eastern Basin

👉 Adapt the range for the selected variable

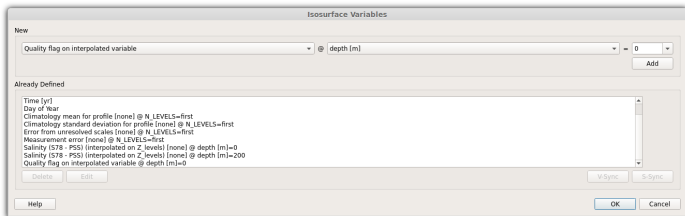
Also possible to have several Surface Windows



Surface window: quality flag



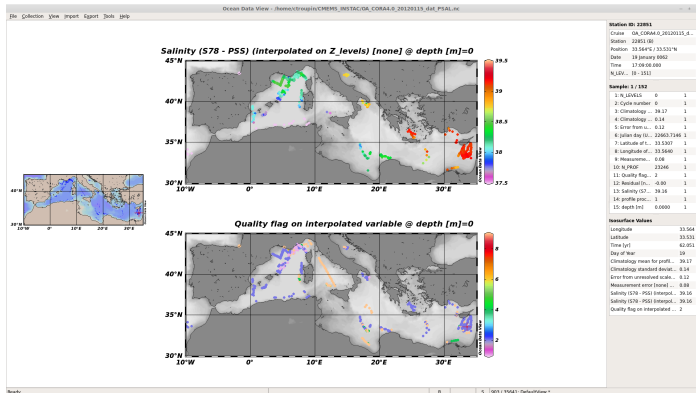
View → Isosurface Variables → Quality flag on interpolated variable at depth = 0



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

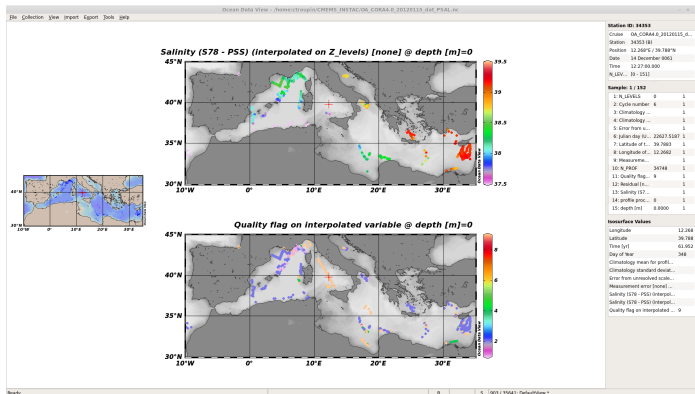
Apply these sample selection criteria globally

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)

File Collection View Import Export Tools Help

- Units / Axis
- Station Selection Criteria Alt+S
- Derived Variables Alt+D
- NonSurface Variables
- Window Properties
- Window Layout Alt+W
 - Layout Template
 - Save View As
 - Save View
 - Load View Shift+L
 - Browse Screen Log File
 - Settings

6-8 Screen Map F8

1 STATION Windows

2 STATION Windows F9

1 SCATTER Windows

2 SCATTER Windows F10

1 SECTION Windows

2 SECTION Windows

3 SECTION Windows F11

4 SECTION Windows

5 SECTION Windows

6 SECTION Windows

1 SURFACE Windows F12

2 SURFACE Windows

3 SURFACE Windows

5 SURFACE Windows

8 SURFACE Windows

11 SURFACE Windows

5 MIXED Windows

From View File

Station ID: 8533

Crutem DA_C0644_0_33120115_d_

Station 34518 (8)

Position 32.783°E / 33.512°N

Date 02 February 0862

Time 10:41:30.800

N_LEV... 10 - 1011

Sample 1 / 332

1. N_LEVELS 8 1
2. Cycle number 24 1
3. Climatology -39.84 1
4. Climatology -0.13 1
5. Error from ... 0.12 1
6. Julian day No. 20073.4852 1
7. Latitude of L. -33.5126 1
8. Longitude of... 32.7836 1
9. Measurement... 0.03 1
10. N_PROF 36613 1
11. Quality flag... 2 1
12. Bandwidth [m... 0.89 1
13. Salinity STD... 29.15 1
14. profile proc... 2 1
15. depth [m] 8.8900 1

NonSurface Values

Longitude 32.783

Latitude 33.512

Time [yr] 42.008

Day of Year 33

Climatology mean for prof... 36.04

Climatology standard deviat... 0.13

Error from unresolved scale... 0.12

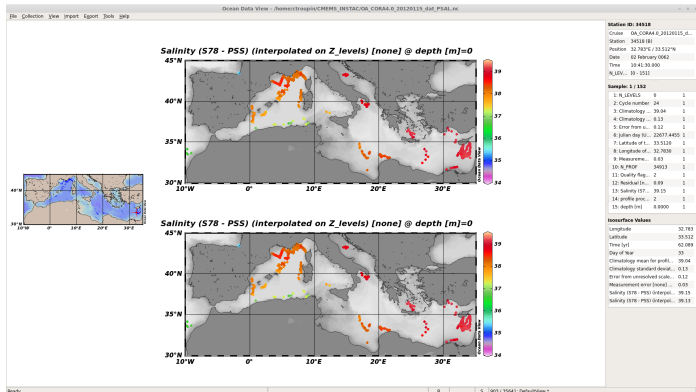
Measurement error [msec] ... 0.03

Applies a predefined window layout.

Surface window: gridding



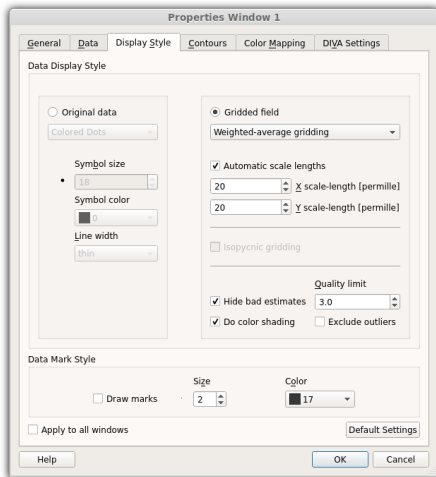
Set Z variable to be Salinity at 0 m



Surface window: gridding



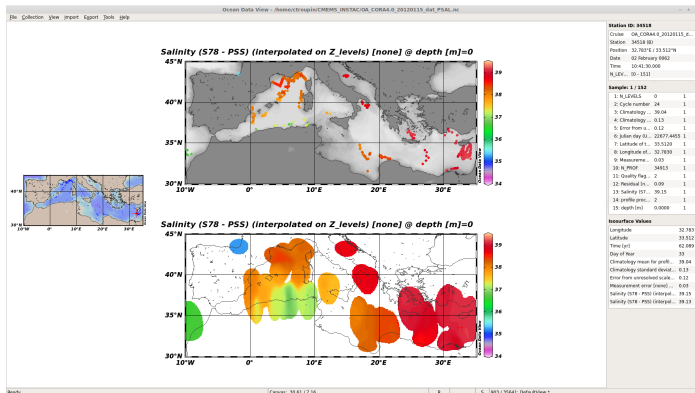
Right-click Properties → Display style → Gridded → Weighted-Average gridding (default parameters 20 X 20)



Surface window: gridding



Gridded field of salinity

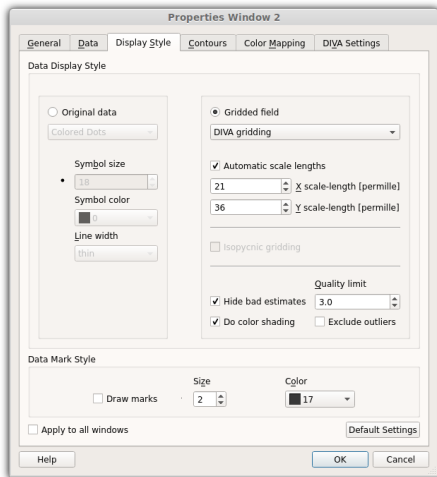


Normal interpolation does not consider boundaries!

Surface window: gridding



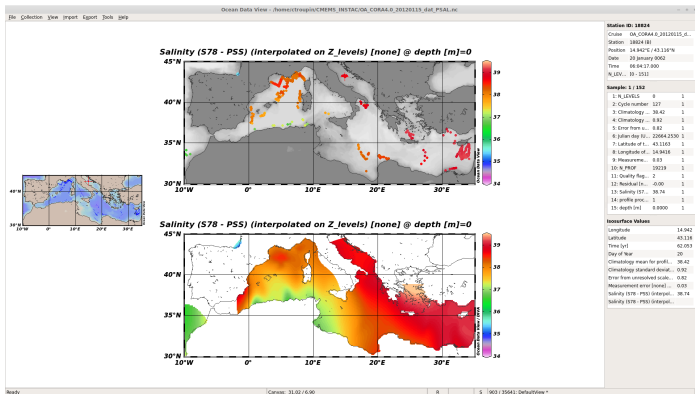
Change Gridding method to DIVA Gridding



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/ctraugiac/CMEMS_INSTAC/IO_CORAA_0_20120115_dlv_PSLA.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 001: 18824

Create: OIA_CORAA_0_20120115_dlv

Station: 18824 (0)

Position: 14.842°E / 43.116°N

Date: 20 January 0002

Time: 06:04:17.000

N_LEVELS: 18 - 1511

Sample: 3 / 592

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...)	22464.2533	1
7: Latitude of...	43.1163	1
8: Longitude of...	14.8416	1
9: Measurement...	-0.02	1
10: M_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)	62.855
Day of Year	20
Climatology mean for profil...	-18.42
Climatology standard deviat...	0.82
Error from unresolved scale...	0.82
Measurement error (Standa...	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

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

<< N_LEVELS

Window 1
SECTION

Newline	F5
Save Map As	Ctrl+S
Zoom	
Auto-Down In	Ctrl+I
Auto-Down Out	Ctrl+O
Valid 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

Define 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.

Station ID: 38024

Cruise	IO-CORAA 0_20120115_d...
Station	38024 (R)
Position	34.842°N / 43.116°E
Date	30 January 2002
Time	06:04:17.000
N_LEV...	(0 - 151)

Sample 1: 1302

1 N_LEVELS	0	1
2 Cyclic number	127	1
3 Climatology	36.42	1
4 Climatology	8.92	1
5 Error from ...	0.82	1
6 Julian Day No.	20944.2130	1
7 Latitude of ...	43.1164	1
8 Longitude of ...	34.8418	1
9 Measurement	0.03	1
10 N_PROF	19019	1
11 Quality flag	2	1
12 SeaLevel [m]	-0.80	1
13 Salinity [ST]	36.74	1
14 profile proc.	1	1
15 depth [m]	0.0000	1

Isosurface Values

Longitude	14.942
Latitude	43.116
Time [yr]	62.093
Day of Year	20
Climatology mean for profil.	36.42
Climatology standard deviat.	0.92
Error from unsmoothed scale...	0.82
Measurement error [seasal]	0.03
Salinity [ST] - PSS Interpol.	36.74
Salinity [ST] - PSS Interpol...	

Edit Section Properties

Ocean Data View - Network\traps\CHEM5\NETSC\DA_CORA_0_20120115_EUT_P5AL.nc

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...)	20464.2538	1
7: Latitude of ...	43.1163	1
8: Longitude of ...	14.9428	1
9: Measurement ...	0.03	1
10: H_PROF	150219	1
11: Quality Flag ...	2	1
12: Residual (m...)	-4.68	1
13: Salinity STD ...	36.74	1
14: profile prec...	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 (standard...	0.03
Salinity STD - PSS (interpol...	36.74
Salinity STD - PSS (interpol...	

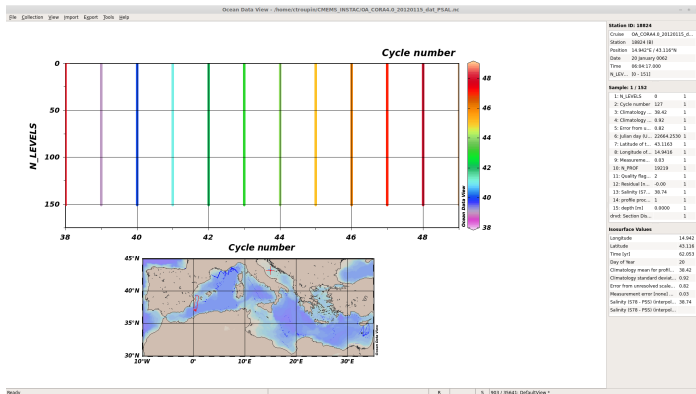
Get Point: L-RSG add point, R-RSG delete point, ONTCA accept, ESC abort

S | 843 / 3564: DefaultView *

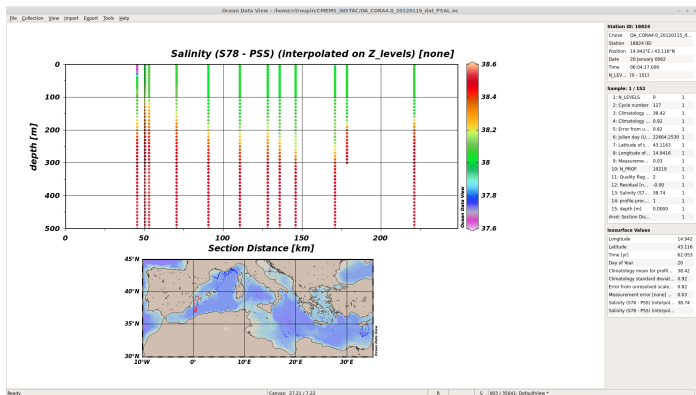
Section window



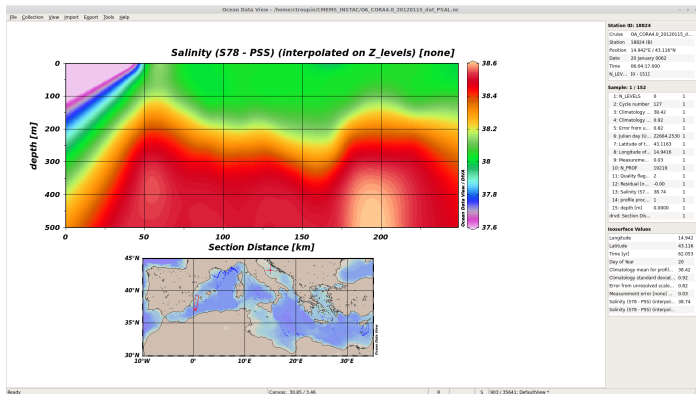
Change X, Y and Z variables
→ Distance, Depth and Salinity



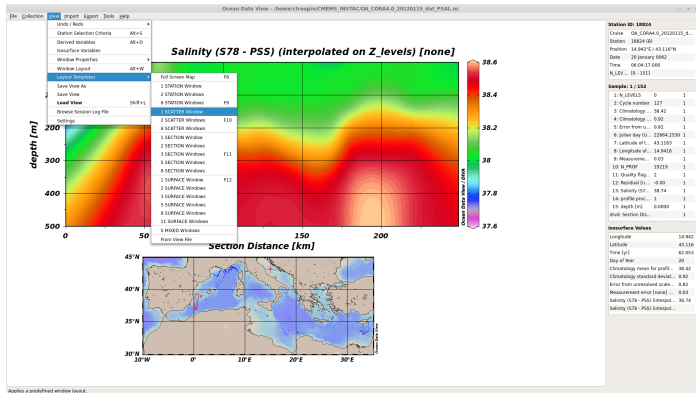
Set Z range between 0 and 500 m



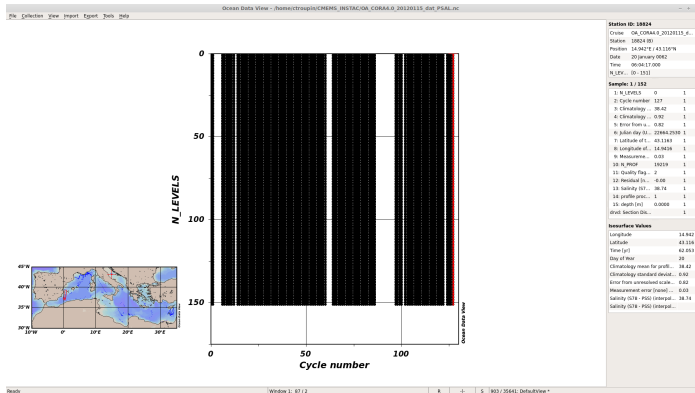
Grid using DIVA interpolation



View → Layout Template → SECTION Window

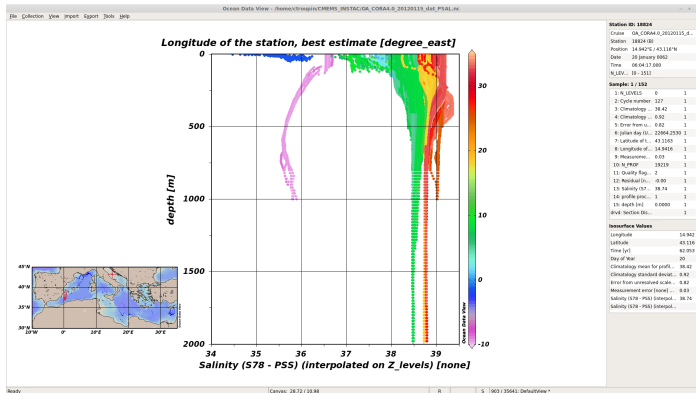


Change X, Y and Z variables



☒ 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

How to get the code?



The code is made available through github:
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 followed by 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
LICENSE	Initial commit	3 months ago
Plot_TimeSeries1.ipynb	Various small changes	3 months ago
README.md	modified readme	3 months ago
Read_CORA_dataset.ipynb	Modified text	2 months ago
Read_TimeSeries_1.ipynb	First commit	2 months ago
Read_TimeSeries_2.ipynb	First commit	2 months ago
Read_TimeSeries_3.ipynb	First commit	2 months ago
Read_drifter_data_1.ipynb	Text corrections	2 months ago
Read_drifter_data_2.ipynb	First commit	2 months ago
Read_drifter_data_3.ipynb	First commit	2 months ago

Below the file list, the `README.md` is expanded, showing the repository title "OceanData_NoteBooks" and the description "Examples of data processing with python notebooks using netCDF files." On the right side of the repository page, there is a sidebar with navigation links: "Code", "Issues" (0), "Pull requests" (0), "Wiki", "Pulse", "Graphs", and "Settings". At the bottom of the sidebar, there is an "SSH clone URL" section with the URL `git@github.com:ctroupin:OceanData_NoteBooks.git` 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
```

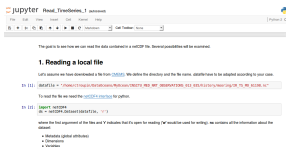
3. Go in the main directory



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


4. In a terminal, type











```
ipython notebook Read_TimeSeries_1.ipynb
```

You should obtain something like that:



 **Jupyter** Read_TimeSeries_1 (autosaved) 

File Edit View Insert Cell Kernel Help Python 2 

         Markdown 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 [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

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) 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 [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 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

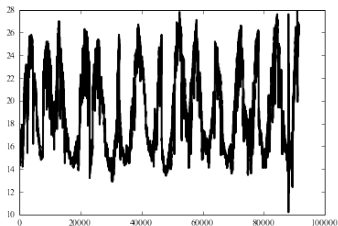
1.1 Metadata

```
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

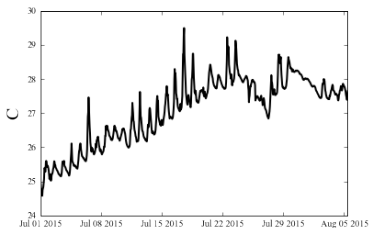


What's inside the repository?



Read_TimeSeries_1.ipynb: reading a local **netCDF** file

Read_TimeSeries_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol



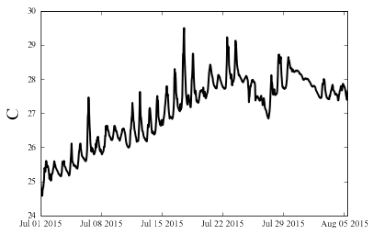
What's inside the repository?



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?

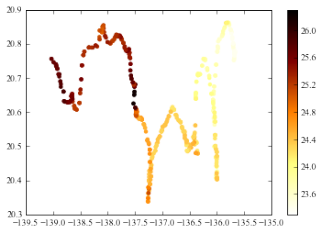


Read_TimeSeries_1.ipynb: reading a local netCDF file

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?



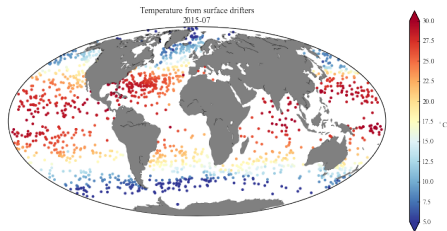
Read_TimeSeries_1.ipynb: reading a local **netCDF** file

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

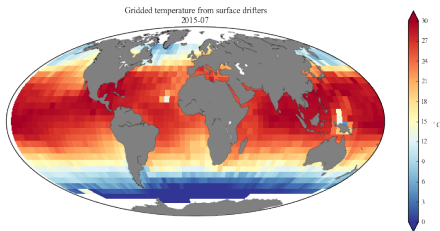
Read_TimeSeries_2.ipynb: reading a remote netCDF using **OPeNDAP** protocol

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

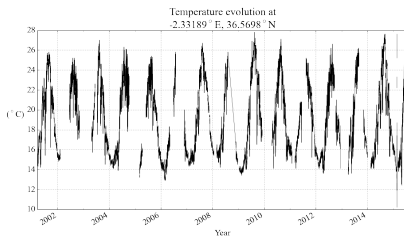
Read_TimeSeries_3.ipynb: reading a netCDF using the **CF module**

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**

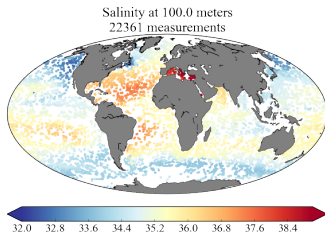
Read_drifter_data_1.ipynb: basic plot of a drifter trajectory

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

