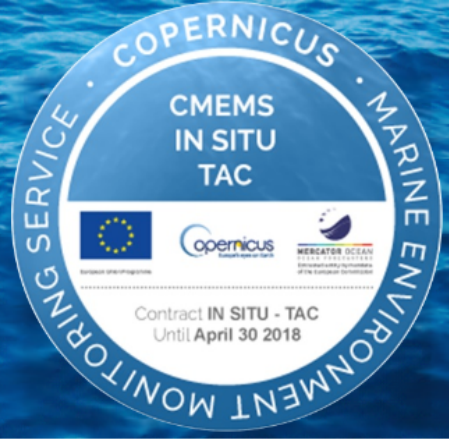


marine.copernicus.eu

1st CMEMS MED User & Training Workshop

In Situ TAC

Charles Troupin

SOCIB

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

”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?

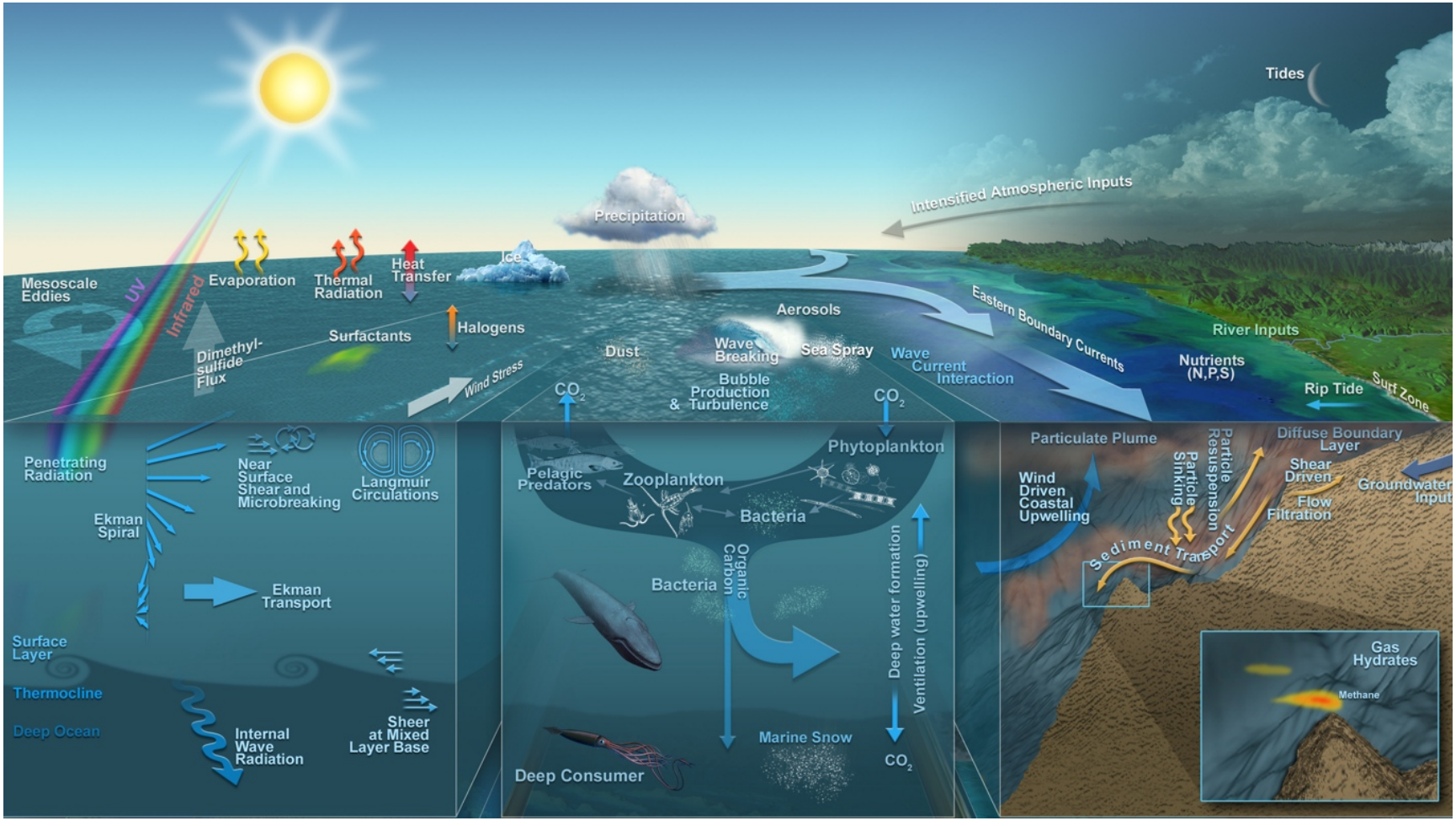
1. Model initialisation
2. Model validation
3. Data assimilation

models are idealisation of the reality

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

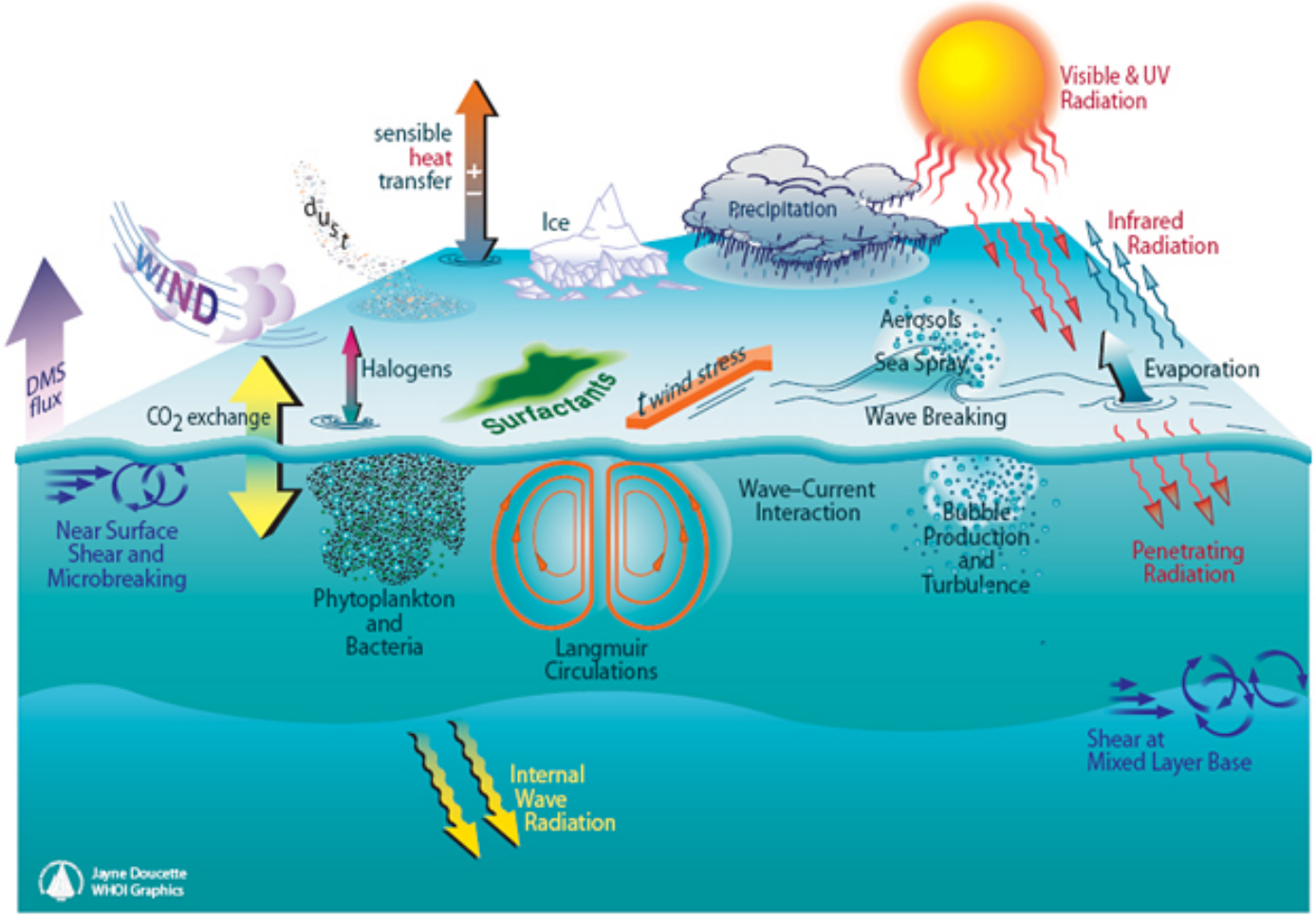
http://www.metoffice.gov.uk/learning/science/first-steps

The ocean is complex



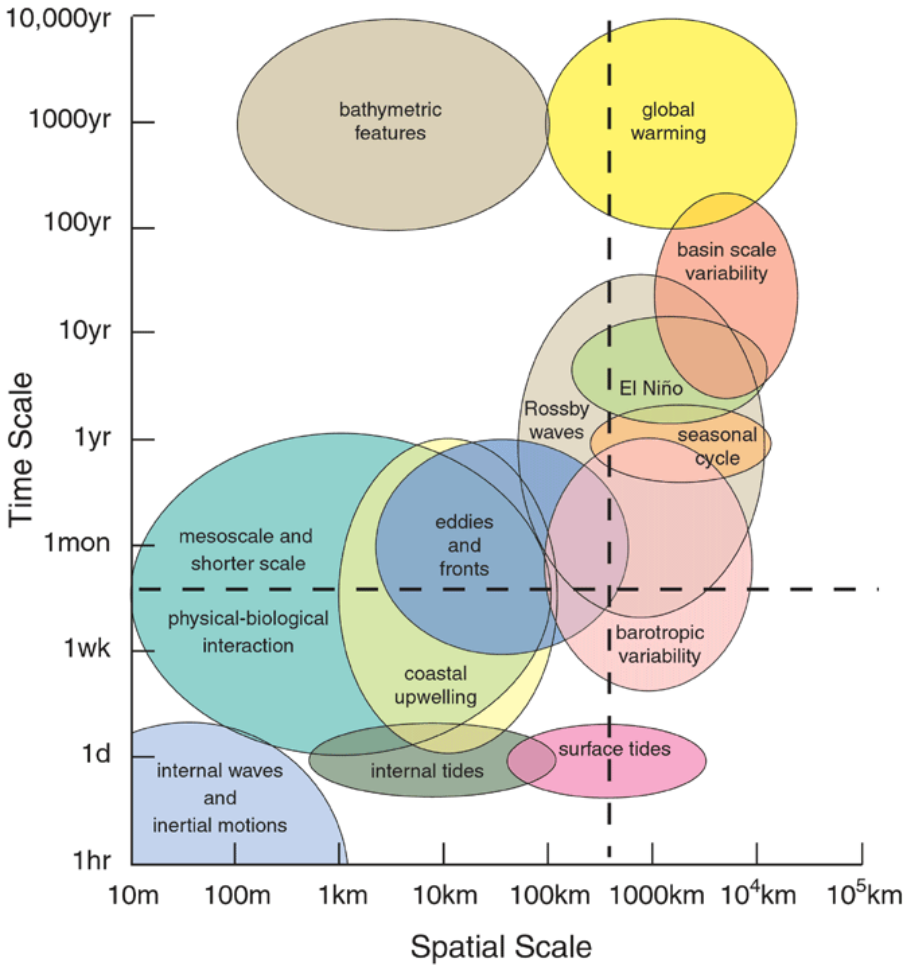
Many processes

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

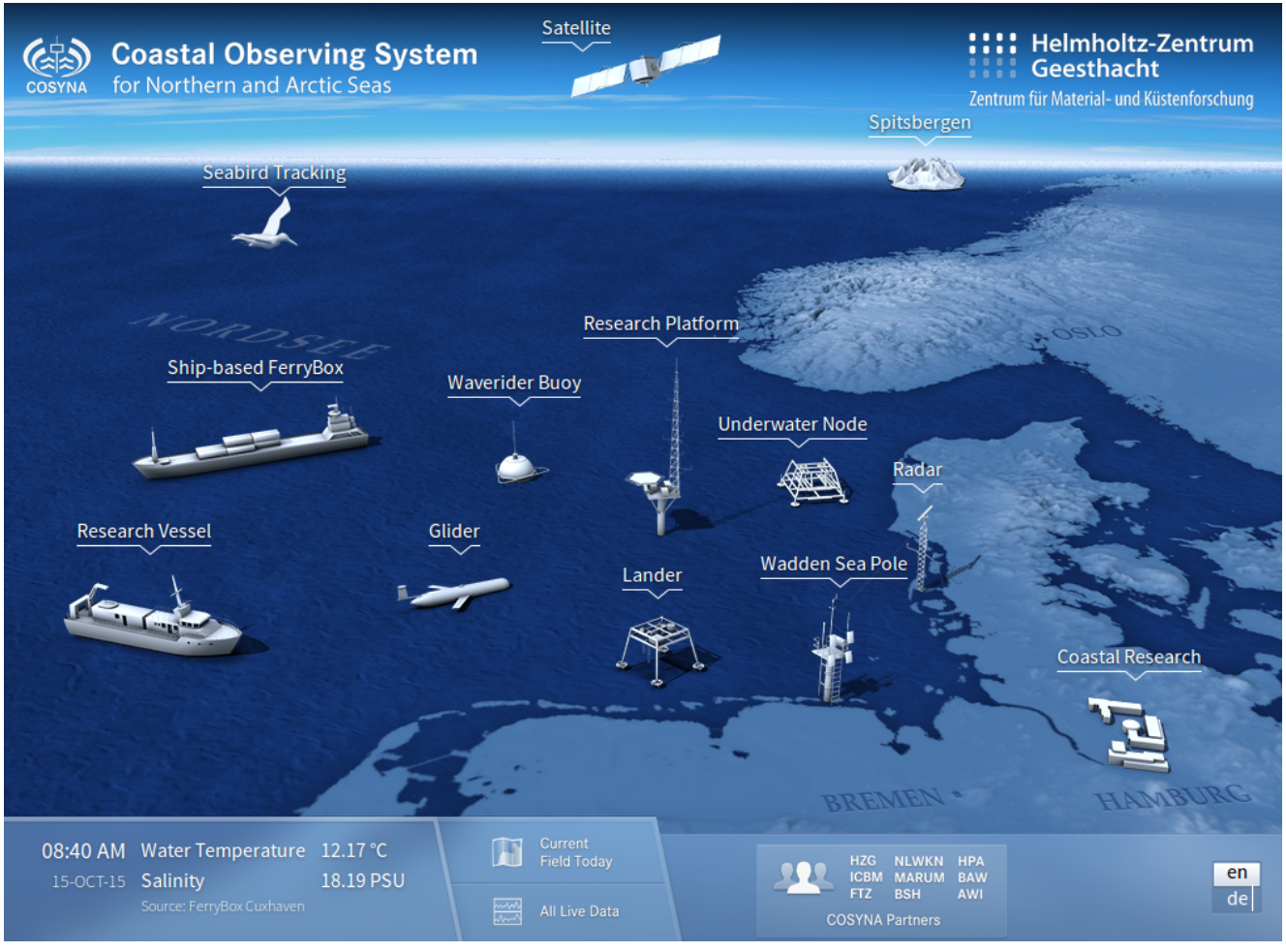
Credit: Global Ocean Observing System Office (IOC-GOOS)

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

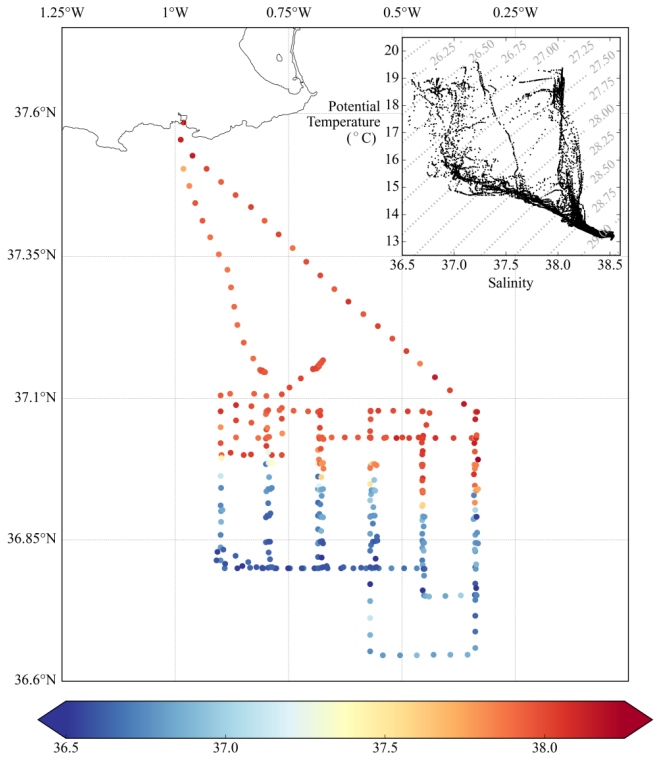
http://codm.hzg.de/codm/

Types of in situ data

Research Vessel

temperature, salinity, currents, oxygen, . . .

Feature type: trajectory of profiles for CTD trajectory for thermosalinograph

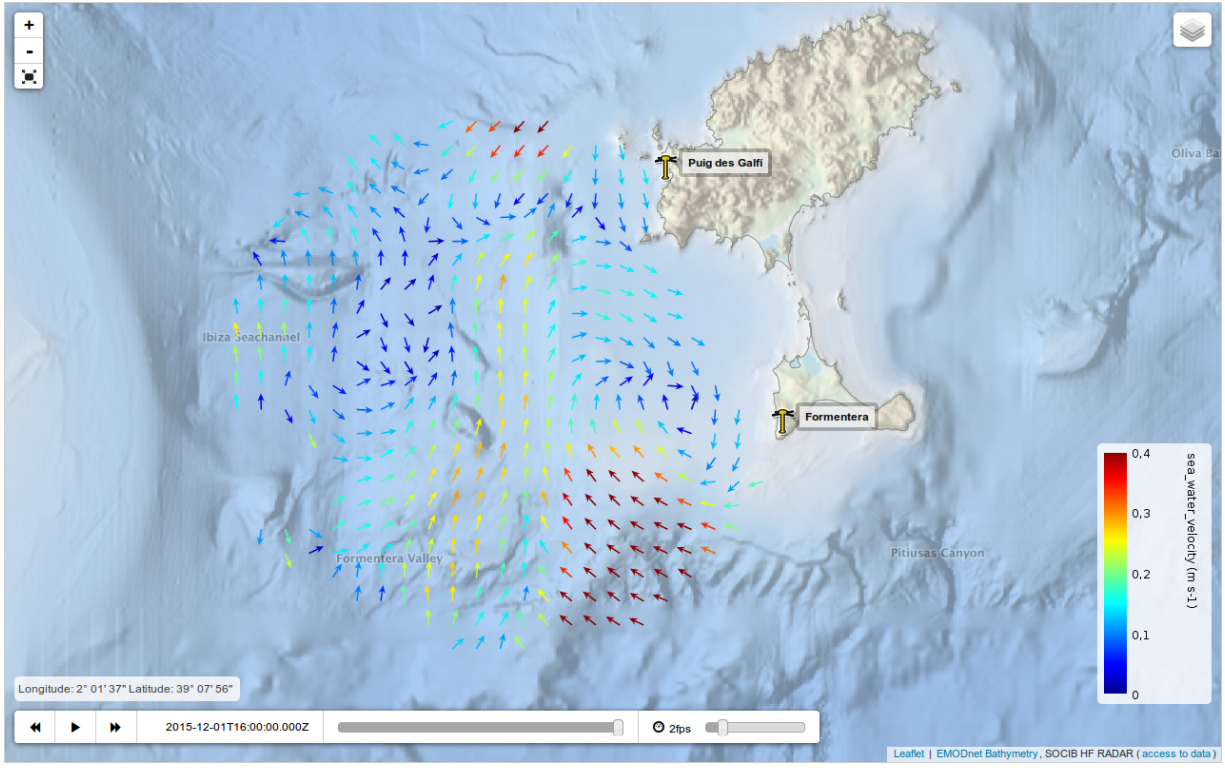


Types of in situ data

Coastal HF Radar

Current speed and direction

Feature type: grid

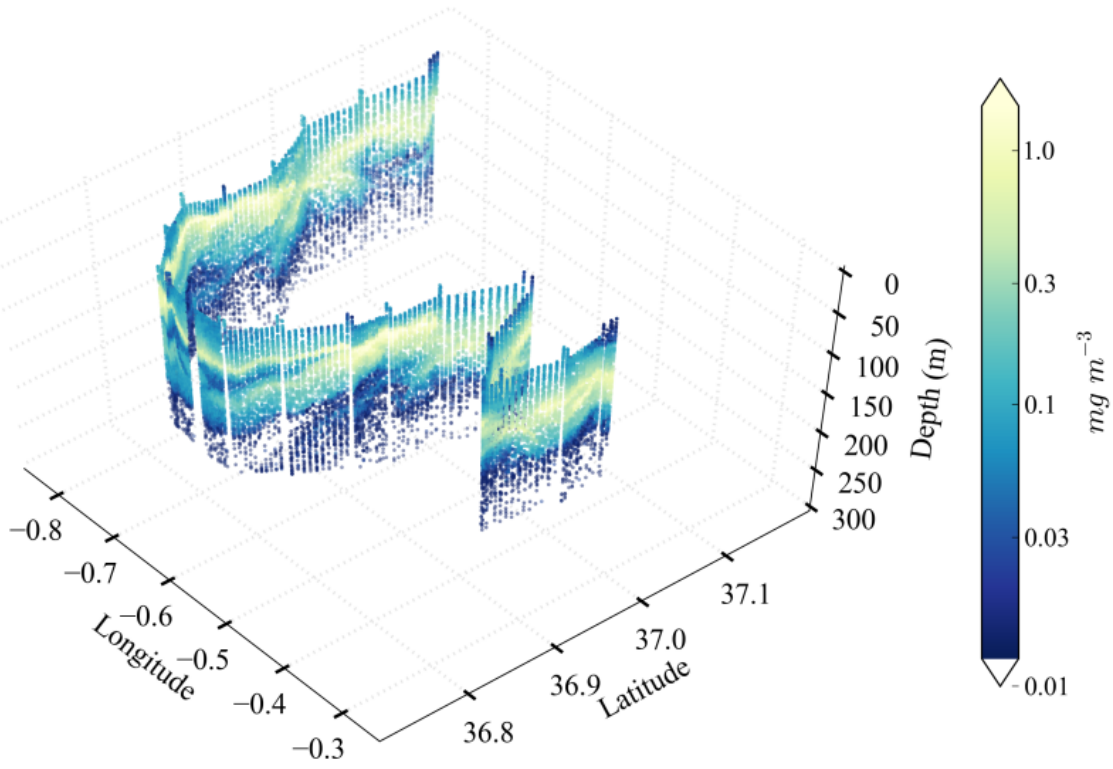


Types of in situ data

Glider

Temperature, salinity, currents, chlorophyll, . . .

Feature type: trajectory

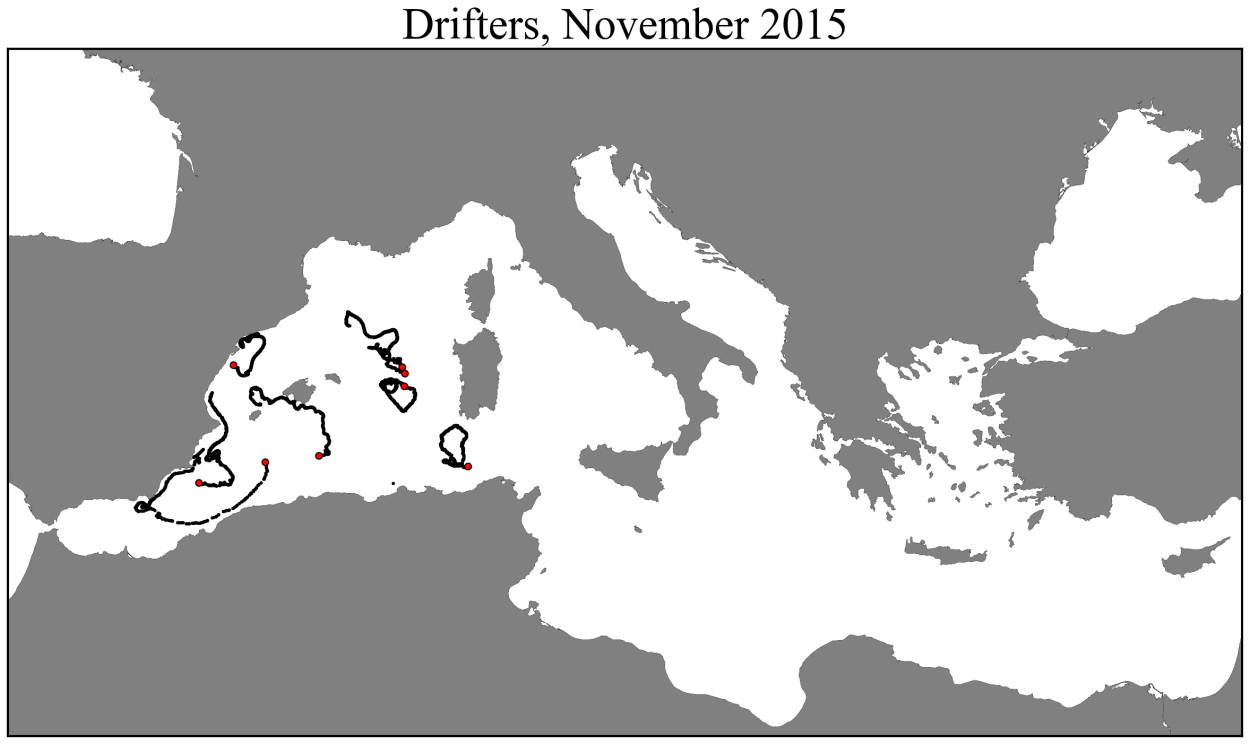


Types of in situ data

Drifting buoys and profilers

Temperature, salinity, currents, . . .

Feature type: trajectory and trajectory of profiles

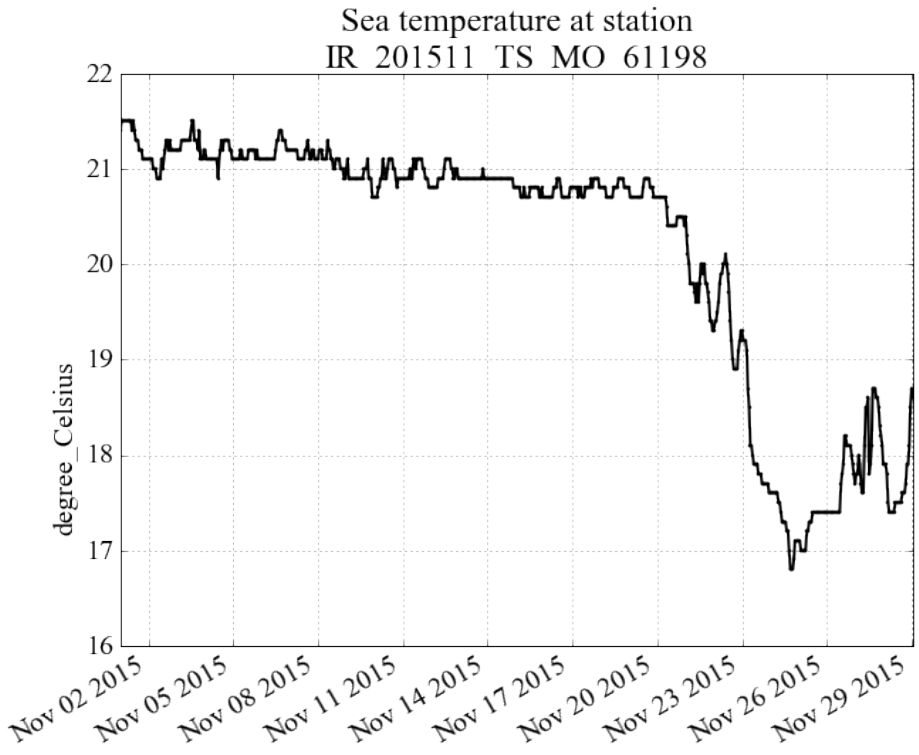


Types of in situ data

Fixed stations

Sea-level, weather/water column variables

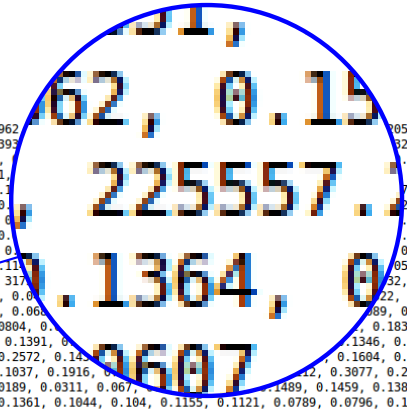
Feature type: time series



Data quality

Data quality

Example: platform velocity (drifter) in meters per second



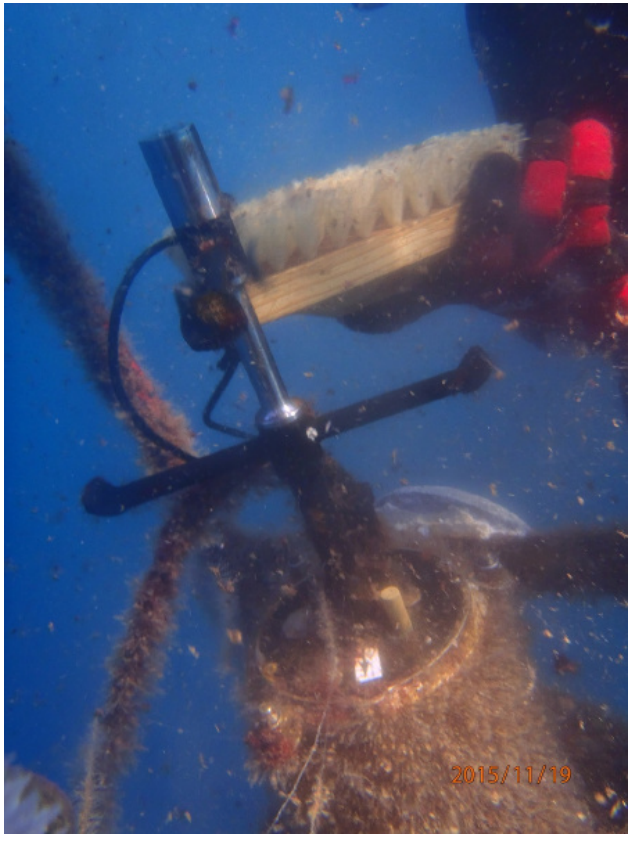
PSPEED[2252]

0.0, 0.2839, 0.3666, 0.4341, 0.3834, 0.3466, 0.3773, 0.3492, 0.323, 0.2699, 0.2069, 0.1519, 0.129, 0.1261, 0.1813, 0.1574, 0.2128, 0.2767, 0.2962, [illegible] 305, 0.176,
0.1328, 0.1152, 0.099, 0.0135, 0.0743, 0.1005, 0.1463, 0.1771, 0.1613, 0.1734, 0.1857, 0.177, 0.1765, 0.2041, 0.1783, 0.1565, 0.16, 0.155, 0.1395, [illegible] 32, 0.0826,
0.0826, 0.0473, 0.0527, 0.0953, 0.1482, 0.1619, 0.2156, 0.2329, 0.2573, 0.2657, 0.3225, 0.3258, 0.2855, 0.2609, 0.2247, 0.1748, 0.1183, 0.0904, [illegible] .2901, 0.3182,
0.2987, 0.2749, 0.2429, 0.2296, 0.0, 0.2104, 0.1823, 0.1785, 0.1757, 0.1991, 0.2107, 0.1944, 0.0, 0.1941, 0.2014, 0.1818, 0.1731, 0.1619, 0.161, [illegible] 0.2262,
0.2231, 0.2413, 0.2319, 0.2115, 0.1704, 0.1339, 0.0719, 0.0467, 0.0539, 0.0911, 0.1372, 0.1492, 0.1717, 0.2119, 8.6001, 0.2228, 0.204, 0.21, 0.1[illegible] 775, 0.0991,
0.1115, 0.1228, 0.1304, 0.1474, 0.1669, 0.1701, 0.1815, 0.1751, 0.1967, 0.1812, 0.1634, 0.1679, 0.184, 0.163, 0.1335, 0.1326, 0.1322, 0.1232, 0.1[illegible] 2682, 0.2399,
0.1896, 0.1623, 0.1812, 0.1661, 0.1484, 0.1413, 0.1672, 0.1762, 0.1673, 0.1812, 0.205, 0.2115, 0.1979, 0.1843, 0.1791, 0.1914, 0.2012, 0.2073, 0[illegible] .0878, 0.0924,
0.1246, 0.149, 0.168, 0.1739, 0.1758, 0.1818, 0.1694, 0.1447, 0.1203, 0.1001, 0.0993, 0.1013, 0.1031, 0.0916, 0.0916, 0.0912, 0.0929, 0.0953, 0.0[illegible] .1568, 0.1342,
0.1249, 0.1232, 0.1193, 0.1174, 0.1032, 0.0785, 0.0851, 0.125, 0.1797, 0.2034, 0.2226, 0.2132, 0.2064, 0.1872, 0.1679, 0.1503, 0.0851, 0.0501, 0[illegible] 0.1607,
0.1735, 0.173, 0.144, 0.108, 0.0866, 0.0571, 0.0602, 0.0588, 0.0321, 0.0576, 0.0608, 0.0657, 0.1028, 0.1181, 0.1349, 0.1466, 0.1662, 0.1528, 0.11[illegible] 0523,
16018.5532, 658.102, 0.1164, 0.1084, 0.1194, 0.1296, 0.1267, 0.1057, 0.124, 0.1272, 0.1143, 0.0812, 0.064, 0.0618, 0.034, 0.0623, 225557.2817, 317[illegible] 32, 0.1366,
0.1354, 0.1483, 0.0986, 0.1051, 0.1317, 0.1211, 0.0807, 0.0694, 0.0823, 0.0995, 0.1022, 0.0978, 0.1201, 0.1477, 0.1552, 0.1649, 0.1364, 0.0696, 0.0[illegible] 32, 0.0313,
0.0277, 0.0338, 0.0662, 0.0891, 0.0932, 0.1299, 0.1545, 0.1657, 0.0406, 0.0578, 0.0493, 0.0533, 0.0555, 0.0404, 0.0461, 0.0665, 0.0607, 0.0528, 0.068[illegible] 089, 0.0894,
0.1049, 0.097, 0.0899, 0.0644, 0.0623, 0.0774, 0.0537, 0.0969, 0.1311, 0.1573, 0.171, 0.1769, 0.163, 0.139, 0.1313, 0.1414, 0.1562, 0.0968, 0.0804, 0.1[illegible] , 0.183, 0.5928,
1.6349, 1.9252, 0.1694, 0.1655, 0.1368, 0.0768, 0.123, 0.1723, 0.2222, 0.2471, 0.2638, 0.2733, 0.2527, 0.2035, 0.1741, 0.1479, 0.1712, 0.1481, 0.1391, 0[illegible] .1346, 0.14, 0.1701,
0.1993, 0.1639, 0.1893, 0.2007, 0.1945, 0.1876, 0.1644, 0.1201, 0.1051, 0.1328, 0.143, 0.1399, 0.155, 0.1712, 0.1771, 0.1653, 0.1759, 0.3497, 0.2572, 0.143[illegible] 0607[illegible] , 0.1604, 0.2042, 0.2178,
0.1932, 0.1618, 0.1456, 0.114, 0.1065, 0.118, 0.0983, 0.1341, 0.1678, 0.2128, 0.2473, 0.2424, 0.2083, 0.1911, 0.149, 0.1015, 0.0421, 0.0215, 0.1037, 0.1916, 0[illegible] 12, 0.3077, 0.2796, 0.2717,
0.2144, 0.1772, 0.1398, 0.083, 0.0467, 0.1086, 0.1491, 0.2251, 0.328, 0.3323, 0.3237, 0.334, 0.3056, 0.2453, 0.208, 0.1704, 0.1105, 0.0657, 0.0189, 0.0311, 0.067[illegible] 1489, 0.1459, 0.138, 0.119,
0.0906, 0.0949, 0.1004, 0.0908, 0.0923, 0.079, 0.0637, 0.088, 0.0728, 0.0626, 0.0852, 0.0827, 0.1258, 0.1553, 0.1741, 0.1983, 0.1875, 0.1677, 0.1361, 0.1044, 0.104, 0.1155, 0.1121, 0.0789, 0.0796, 0.1078, 0.1198,
0.1155, 0.142, 0.1608, 0.1478, 0.1911, 0.1843, 0.1548, 0.1232, 0.0926, 0.1031, 0.069, 0.0211, 0.02, 0.0617, 0.0919, 0.0874, 0.0699, 0.0538, 0.0706, 0.1124, 0.1308, 0.1441, 0.1637, 0.1701, 0.1589, 0.1405, 0.1522,
0.1279, 0.0817, 0.0478, 0.0383, 0.017, 0.0456, 0.05, 0.0809, 0.1113, 0.1308, 0.1486, 0.152, 0.1467, 0.1232, 0.0843, 0.0842, 0.0691, 0.0855, 0.0999, 0.1144, 0.0963, 0.0526, 0.0156, 0.0269, 0.0647, 0.0986, 0.0882,
0.1142, 0.1382, 0.1207, 0.0835, 0.0963, 0.0976, 0.0997, 0.1121, 0.0944, 0.0708, 0.0393, 0.0273, 0.0221, 0.0302, 0.0588, 0.0936, 0.1036, 0.0872, 0.0633, 0.0785, 0.0806, 0.0802, 0.0859, 0.0727, 0.0867, 0.0758,
0.0708, 0.0595, 0.0408, 0.0351, 0.0446, 0.0787, 0.1025, 0.1249, 0.1427, 0.1506, 0.146, 0.1096, 0.099, 0.1009, 0.0838, 0.0947, 0.0847, 0.0819, 0.0859, 0.0892, 0.1274, 0.144, 0.1744, 0.1863, 0.1695, 0.174, 0.1458,
0.1447, 0.1402, 0.149, 0.1448, 0.1371, 0.1441, 0.1605, 0.1724, 0.1659, 0.1725, 0.1933, 0.1948, 0.1997, 0.2009, 0.1903, 0.1711, 0.1833, 0.1828, 0.1636, 0.1771, 0.2215, 0.2226, 0.225, 0.2437, 0.234, 0.2307, 0.2168,
0.1808, 0.1575, 0.1525, 0.1545, 0.1648, 0.1921, 0.2229, 0.2076, 0.2405, 0.2564, 0.244, 0.2196, 0.1872, 0.1856, 0.1863, 0.1822, 0.1455, 0.1147, 0.1031, 0.1598, 0.2011, 0.22, 0.2432, 0.2447, 0.2344, 0.227, 0.2052,
0.1666, 0.1555, 0.1355, 0.0946, 0.069, 0.0402, 0.0172, 0.0435, 0.0765, 0.089, 0.1045, 0.1318, 0.1585, 0.1904, 0.178, 0.1604, 0.1454, 0.1167, 0.1057, 0.0695, 0.0374, 0.0265, 0.0323, 0.0551, 0.0633, 0.1066, 0.1476,
0.1818, 0.2024, 0.2131, 0.2115, 0.1819, 0.1305, 0.0908, 0.0554, 0.0492, 0.0776, 0.1015, 0.1182, 0.1539, 0.1571, 0.1729, 0.1835, 0.1899, 0.209, 0.211, 0.1804, 0.1649, 0.1619, 0.2029, 0.2148, 0.2137, 0.2058, 0.212,
0.2225, 0.2342, 0.2301, 0.2219, 0.2123, 0.2265, 0.2521, 0.2623, 0.2773, 0.2972, 0.2928, 0.2932, 0.2931, 0.2798, 0.2585, 0.225, 0.1945, 0.147, 0.1143, 0.1083, 0.1103, 0.1231, 0.1284, 0.1308, 0.1632, 0.1797, 0.1833,
0.1806, 0.1614, 0.1447, 0.1397, 0.1313, 0.1525, 0.1609, 0.1801, 0.1853, 0.1783, 0.1725, 0.1724, 0.2157, 0.2236, 0.2255, 0.2091, 0.1929, 0.1766, 0.1567, 0.1772, 0.1766, 0.1811, 0.1934, 0.2102, 0.2313, 0.2447,
0.2594, 0.2475, 0.2059, 0.1574, 0.118, 0.0954, 343.2381, 0.1979, 0.1689, 0.2158, 0.2823, 0.3196, 0.3487, 0.3565, 0.3502, 0.3206, 0.2727, 0.2244, 0.1202, 0.0561, 0.0479, 0.0846, 0.1293, 0.1787, 0.2497, 0.2799,
0.2722, 0.283, 0.2831, 0.2907, 0.293, 0.2814, 0.2435, 0.2041, 0.1693, 0.1325, 0.0824, 0.0567, 0.0622, 0.1293, 0.1948, 0.249, 0.2656, 0.2987, 0.2914, 0.2622, 0.2099, 0.1835, 0.1507, 0.1178, 0.13, 0.1174, 0.1307,
0.1686, 0.2364, 0.2783, 0.2631, 0.2371, 0.2399, 0.23, 0.2327, 0.2531, 0.2196, 0.1939, 0.1596, 0.1204, 0.0739, 0.0715, 0.1097, 0.1343, 0.152, 0.1508, 0.1759, 0.1881, 0.2248, 0.2312, 0.2386, 0.2421, 0.2358, 0.2301,
0.2164, 0.1933, 0.1556, 0.1285, 0.1309, 0.1164, 0.1207, 0.1273, 0.1338, 0.1494, 0.1868, 0.1576, 0.1347, 0.1459, 0.1713, 0.2089, 0.2028, 0.174, 0.1807, 0.2331, 0.2501, 0.2779, 0.297, 0.289, 0.2777, 0.234, 0.237,
0.205, 0.1773, 0.1596, 0.169, 0.1725, 0.1484, 0.1535, 0.1837, 0.2143, 0.2413, 0.2401, 0.2446, 0.2101, 0.1806, 0.1583, 0.1384, 0.1252, 0.1442, 0.126, 0.1359, 0.1506, 0.1479, 0.1376, 0.1358, 0.1728, 0.1976, 0.2028,
0.1911, 0.1875, 0.1942, 0.1392, 0.1259, 0.1465, 0.1681, 0.1339, 0.1354, 0.1551, 0.1464, 0.1273, 0.0865, 0.0766, 0.0461, 0.0048, 0.0266, 0.0543, 0.0746, 0.0792, 0.0976, 0.1352, 0.1888, 0.1949, 0.2009, 0.1638,
0.1426, 0.133, 0.1232, 0.1417, 0.1497, 0.1169, 0.0774, 0.0412, 0.0072, 0.0408, 0.1091, 0.1163, 0.1133, 0.0835, 0.0957, 0.0899, 0.0569, 0.0456, 0.0442, 0.0647, 0.0799, 0.0706, 0.0742, 0.102, 0.0862, 0.0585, 0.0396,
0.0494, 0.0473, 0.0378, 0.029, 0.0158, 0.0421, 0.0972, 0.149, 0.1735, 0.1984, 0.1929, 0.1816, 0.15, 0.1155, 0.0951, 0.0685, 0.0638, 0.0684, 0.087, 0.1047, 0.1344, 0.1364, 0.1551, 0.1613, 0.1342, 0.1051, 0.0928,
0.0378, 0.0196, 0.0127, 0.0155, 0.0196, 0.0187, 0.0494, 0.0859, 0.111, 0.1314, 0.1305, 0.1727, 0.1775, 0.1646, 0.1434, 0.1432, 0.1448, 0.143, 0.1281, 0.0898, 0.0342, 0.0224, 0.0715, 0.0754, 0.0874, 0.1159, 0.141,
0.1459, 0.1574, 0.1593, 0.1864, 0.2001, 0.2039, 0.1573, 0.1362, 0.0935, 0.1133, 0.1325, 0.1357, 0.1471, 0.1988, 0.2272, 0.2221, 0.1874, 0.1613, 0.1803, 0.2175, 0.3854, 0.2967, 0.4033, 0.2804, 0.2663, 0.2483,
0.2002, 0.2249, 0.2337, 0.2962, 0.2935, 0.3084, 0.3166, 0.3389, 0.3289, 0.3649, 0.3964, 0.4465, 0.4657, 0.4601, 0.4619, 127.4443, 0.9106, 0.4454, 0.4066, 0.339, 0.2855, 0.1891, 0.1505, 0.1773, 0.2237, 0.2541,
0.277, 0.3202, 0.3681, 0.3536, 0.309, 0.3373, 0.3529, 0.3354, 0.3352, 0.3344, 0.3038, 0.2539, 0.1956, 0.168, 0.1138, 0.0787, 0.0585, 0.0806, 0.0873, 0.0867, 0.0779, 0.0951, 0.1347, 0.1887, 0.2229, 0.2343, 0.2723,
0.2939, 0.2897, 0.2901, 0.2902, 0.2868, 0.2832, 0.2609, 0.24, 0.2281, 0.1995, 0.1896, 0.1754, 0.1625, 0.1618, 0.1543, 0.1376, 0.1362, 0.1305, 0.1621, 0.2008, 0.1905, 0.1898, 0.2161, 0.2236, 0.2646, 0.3037, 0.3389,
0.3333, 0.3209, 0.2799, 0.2573, 0.2276, 0.2235, 0.2267, 0.222, 0.2155, 0.2017, 0.1709, 0.1308, 0.0962, 0.0812, 0.0977, 0.1572, 0.1979, 0.2271, 0.2524, 0.3012, 0.338, 0.352, 0.3336, 0.3048, 0.2954, 0.3142, 0.3362,
0.3362, 0.3263, 0.3143, 0.2838, 0.2697, 0.2938, 0.3195, 0.3, 0.2644, 0.2409, 0.2373, 0.2507, 0.2692, 0.292, 0.3199, 0.3433, 0.3697, 0.3585, 0.3174, 0.3097, 0.2936, 0.284, 0.2682, 0.2546, 0.2227, 0.1871, 0.1611,
0.1162, 0.1013, 0.0998, 0.0849, 0.0684, 0.1338, 0.1499, 0.1483, 0.1482, 0.1847, 0.1511, 0.2485, 0.2493, 0.2358, 0.2421, 0.2369, 0.2051, 0.1215, 0.1687, 0.1922, 0.2216, 0.2626, 0.2873, 0.2825, 0.2545, 0.2282,
0.1899, 0.1903, 0.1795, 0.157, 0.0911, 0.0725, 0.0638, 0.2964, 0.169, 0.1363, 0.1191, 0.122, 0.1472, 0.1601, 0.1695, 0.2052, 0.2156, 0.2061, 0.1948, 0.142, 0.1236, 0.1301, 0.1303, 0.1407, 0.1524, 0.178, 0.1938,
0.1975, 0.1787, 0.1596, 0.1527, 0.1773, 0.1702, 0.1797, 0.1869, 0.1561, 0.1879, 0.2337, 0.2011, 0.1943, 0.1778, 0.1885, 0.1854, 0.2141, 0.2265, 0.2139, 0.2068, 0.2015, 0.1807, 0.1743, 0.2106, 0.2631, 0.2983,
0.3142, 0.2848, 0.264, 0.2773, 0.2711, 0.2566, 0.2469, 0.2207, 0.1944, 0.1897, 0.197, 0.2274, 0.2563, 0.237, 0.2432, 0.2203, 0.248, 0.3008, 0.2676, 0.2829, 0.2339, 0.1676, 0.0789, 0.0644, 0.1576, 0.203, 0.2531,

Why data are not always good?

Why data are not always good?

Why data are not always good?



QC processing

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 vs. delayed mode quality control

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

Tests applied on Argo vertical profiles

- Deepest Pressure Test
- Platform Identification

More details: doi:10.13155/33951

Tests applied on Argo vertical profiles

- Deepest Pressure Test
- Platform Identification
- Impossible Date Test

More details: doi: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

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

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

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

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

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

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

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

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

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

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

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

More details: doi: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

More details: doi: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

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

Summary

1. Various types of platforms available

Summary

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

Summary

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

Summary

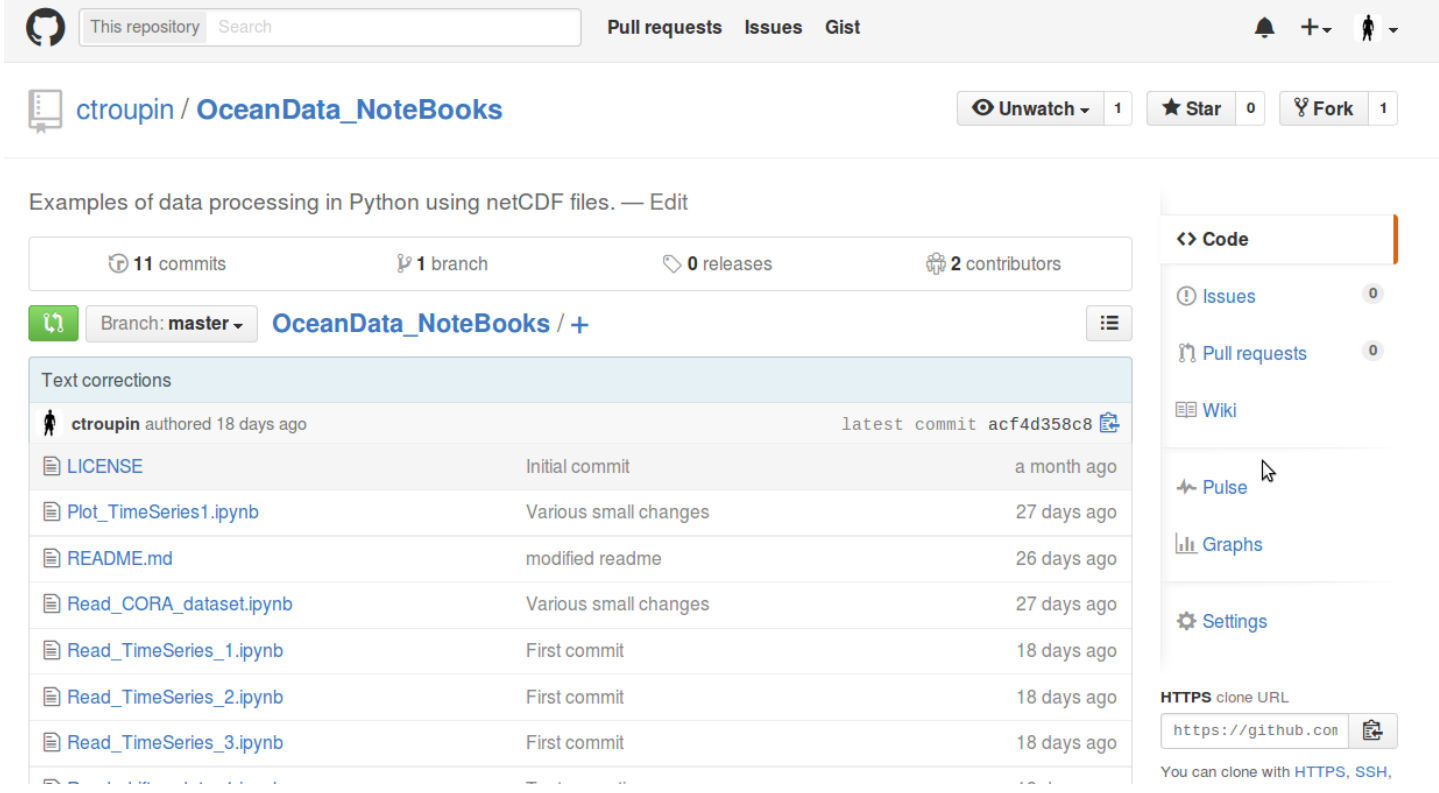
1. Various types of platforms available
2. Quality flags assigned to the measurements
3. In situ data are essential for numerical model
4. In situ observations are scarce

Training material

Training material

ipython notebooks distributed in github

https://github.com/ctroupin/OceanData_NoteBooks



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

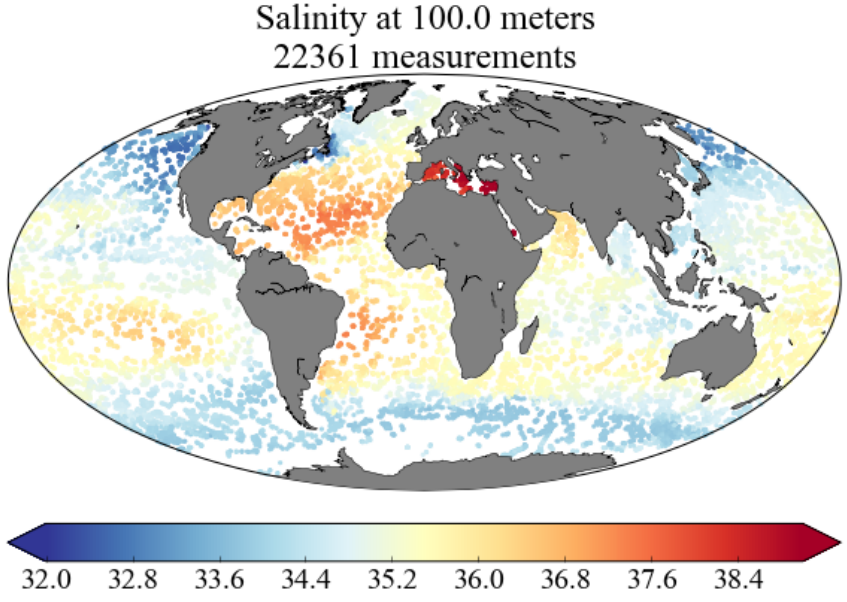
Why github?

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

How does it looks 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(scat, 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()
```



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

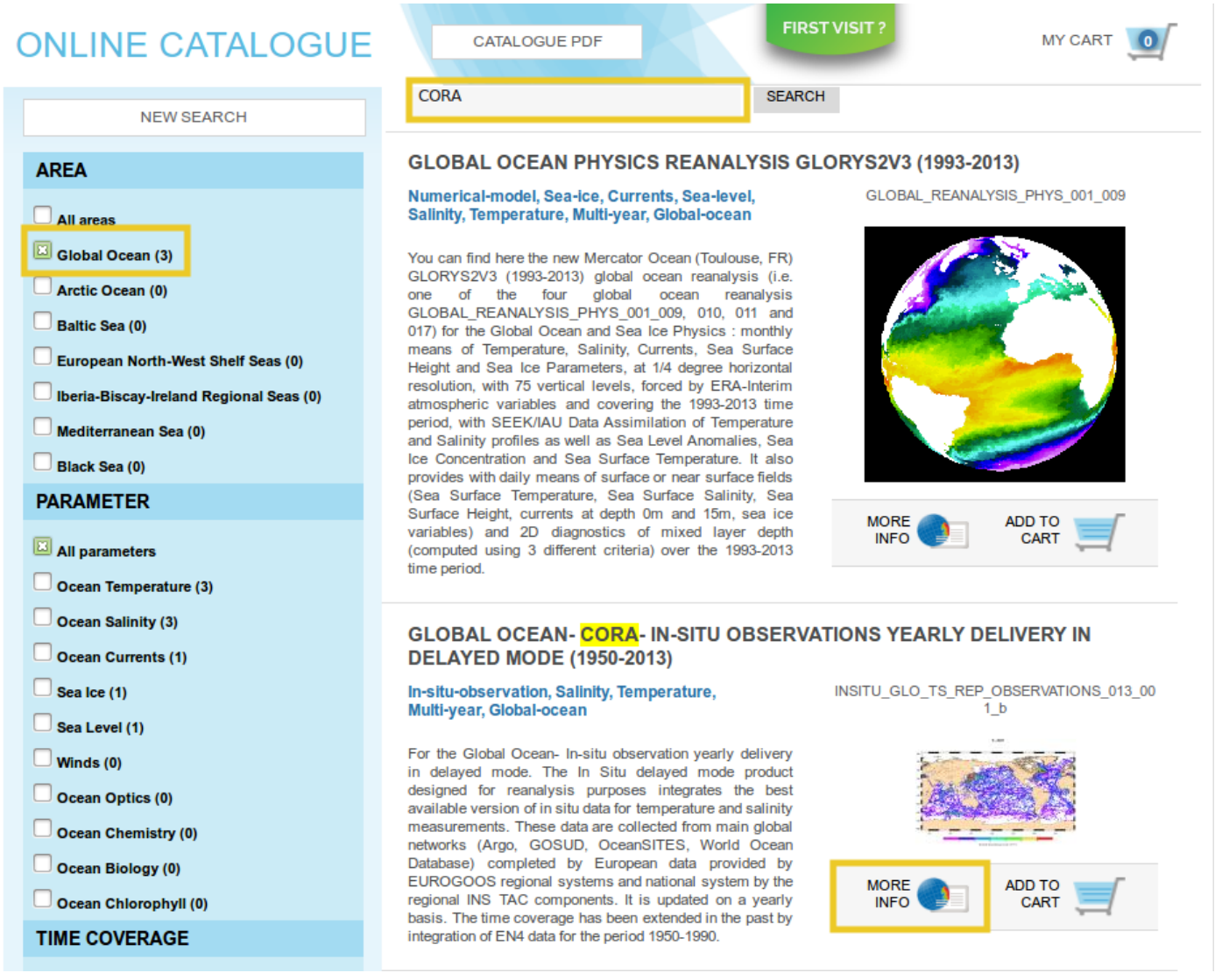
How to get the data?

Getting CORA dataset

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

Getting CORA dataset

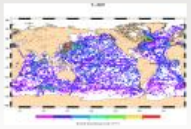
Select ”Global Ocean” and type ”CORA” in search box



Getting CORA dataset

Download product

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



BACK TO SEARCH

ADD TO CART

VIEW PRODUCT

DOWNLOAD PRODUCT

Metadata provided by CMEMS
Credits: Copernicus Marine Service

INFORMATION PDF XML | 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

90.0

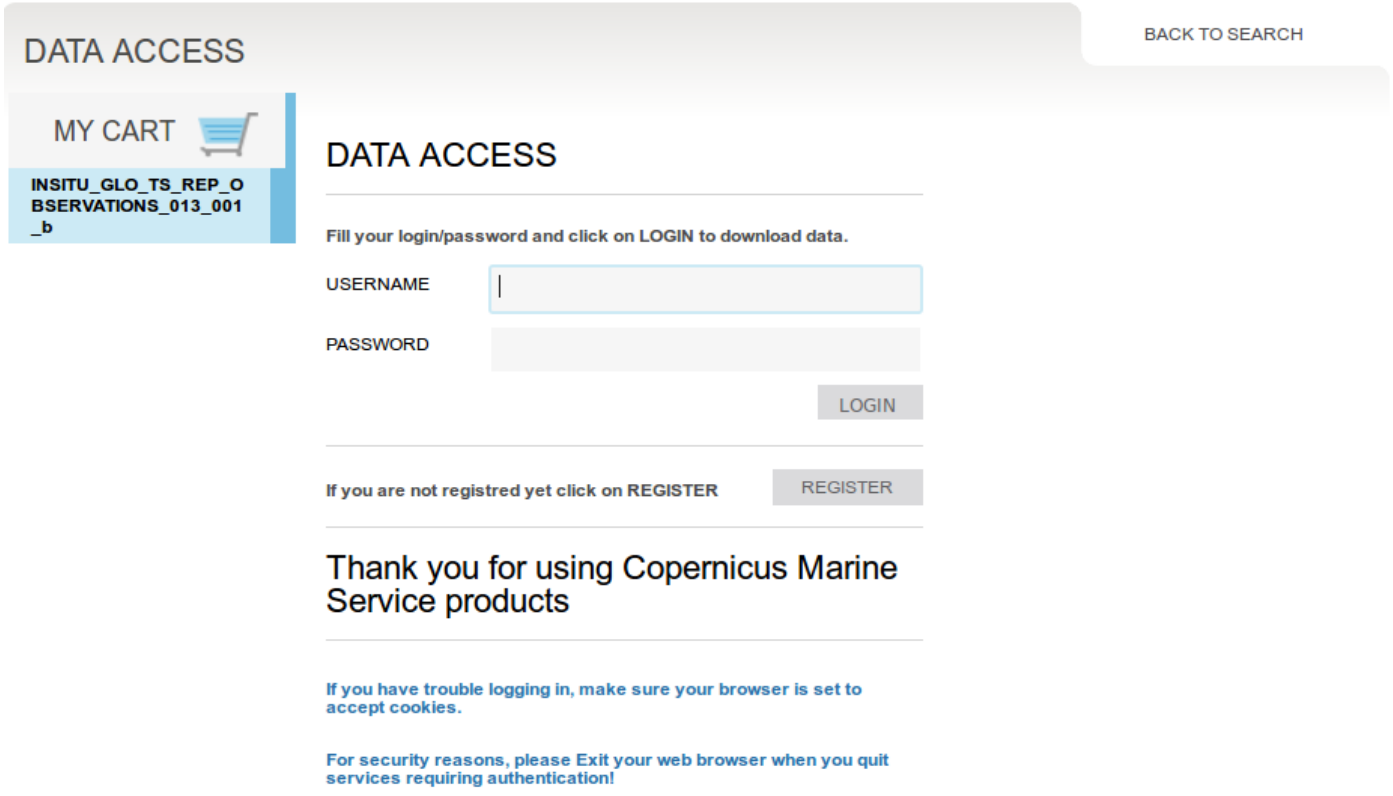
-180.0 180.0

-90.0

Areas:
global-ocean

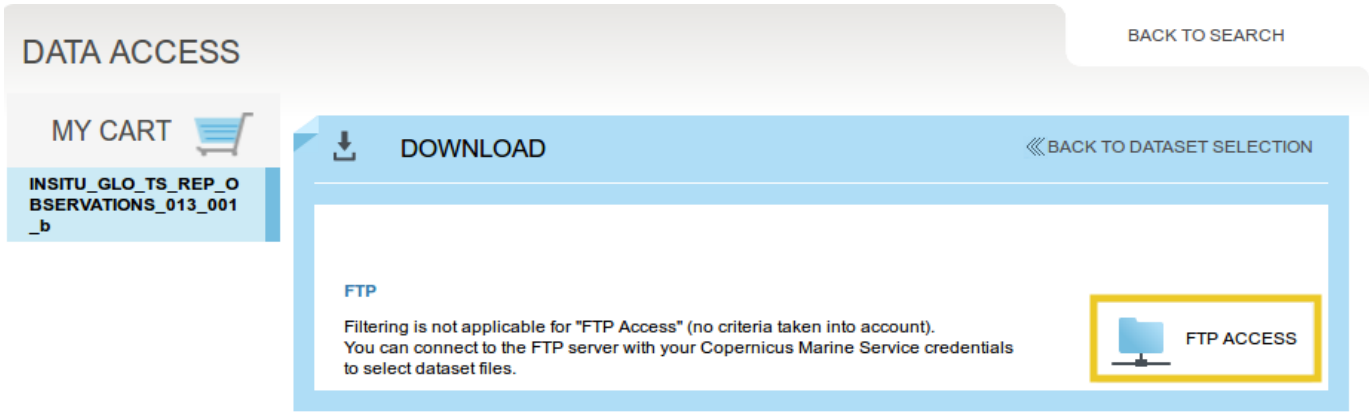
Getting CORA dataset

Use your username & password



Getting CORA dataset

FTP access (username & password)



Getting CORA dataset

OA directory

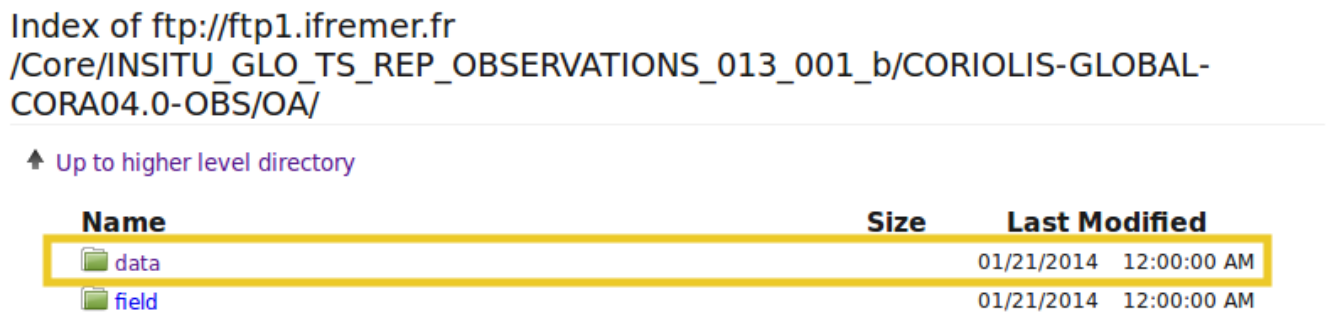
Index of ftp://ftp1.ifremer.fr
/Core/INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b/CORIOLIS-GLOBAL-CORA04.0-OBS/

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



Getting CORA dataset

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

Getting CORA dataset

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 is 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",
        "" ;
}
```

Ferret

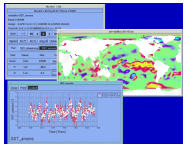
Home page: http://www.ferret.noaa.gov/Ferret/

What is does: visualization and analysis environment

Ferret to get basic info on file

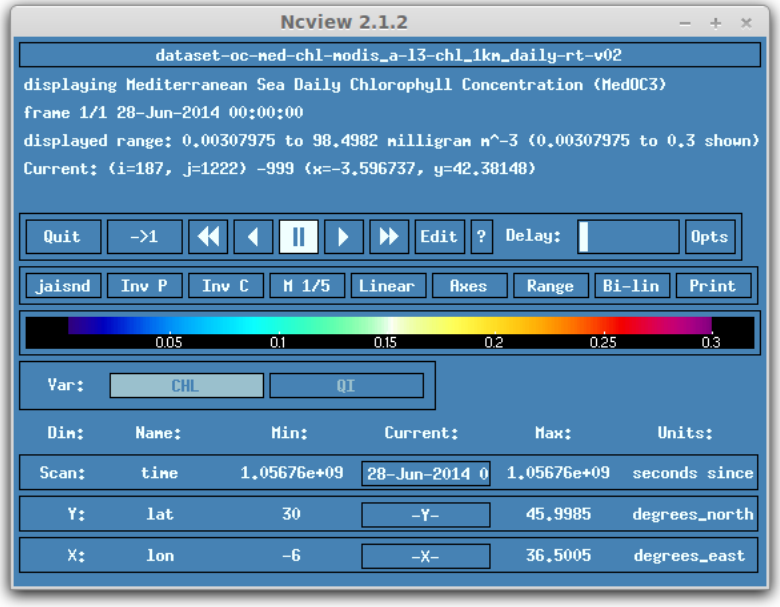
```
ctroupin@SCBD046 ~/Desktop $ ferret_c
     NOAA/PMEL TMAP
     FERRET v6.62
     Linux(gfortran) 2.6.9-89.0.20.ELsmp - 07/06/13
     25-Nov-15 12:23

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?
```

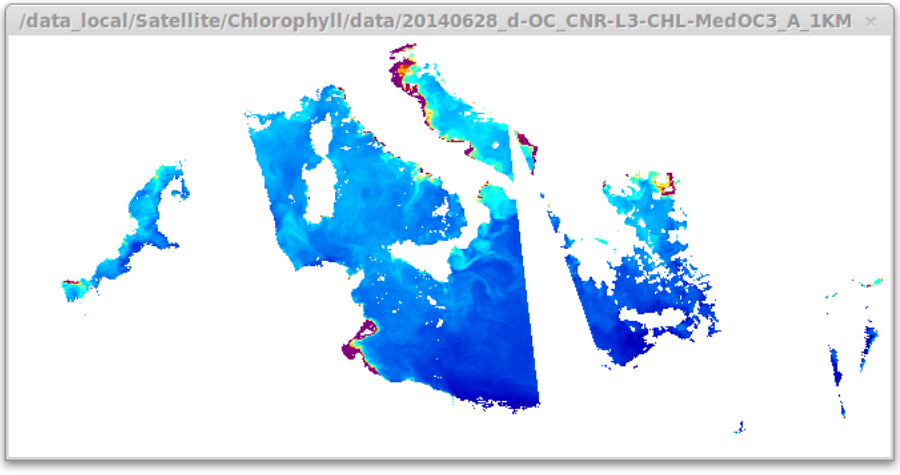
ncview



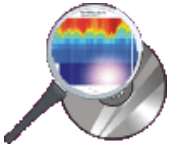
Home page: http://meteora.ucsd.edu/~pierce/ncview_home_page.html

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



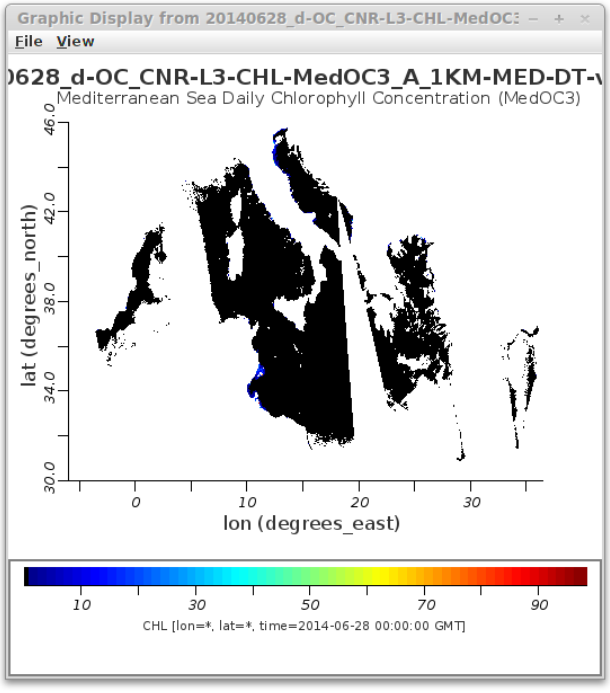


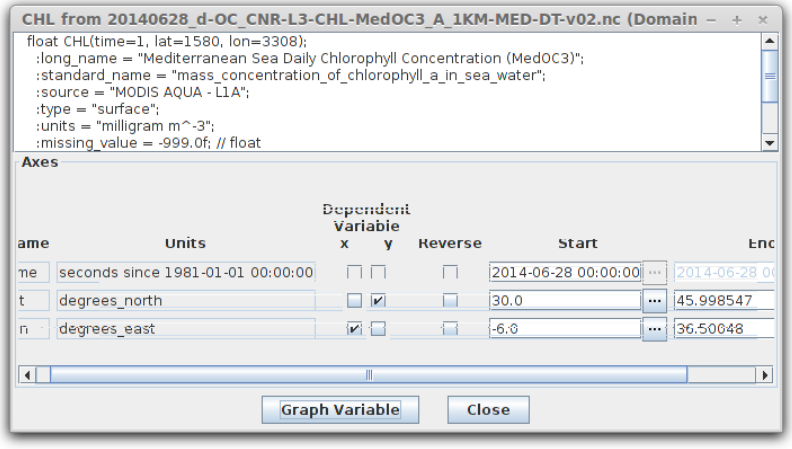
ncbrowse



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

What is does: interactive graphical display

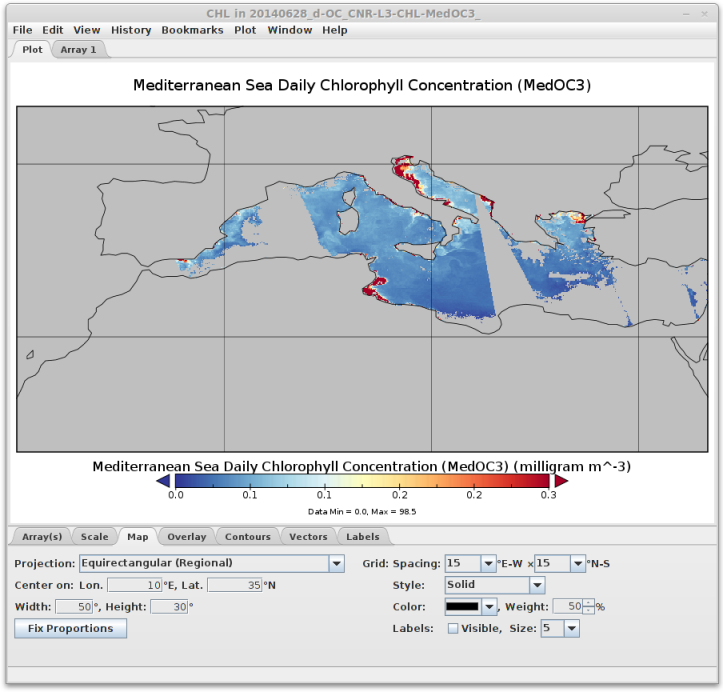


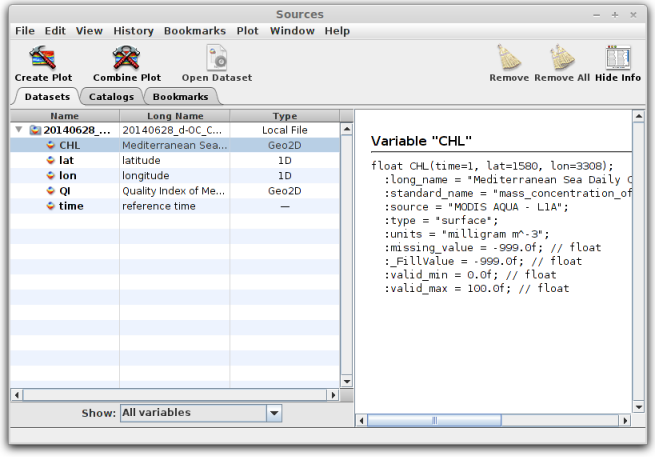


Panoply

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

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





cdo – Climate Data Operators

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
```

NCO – netCDF Operators

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
```

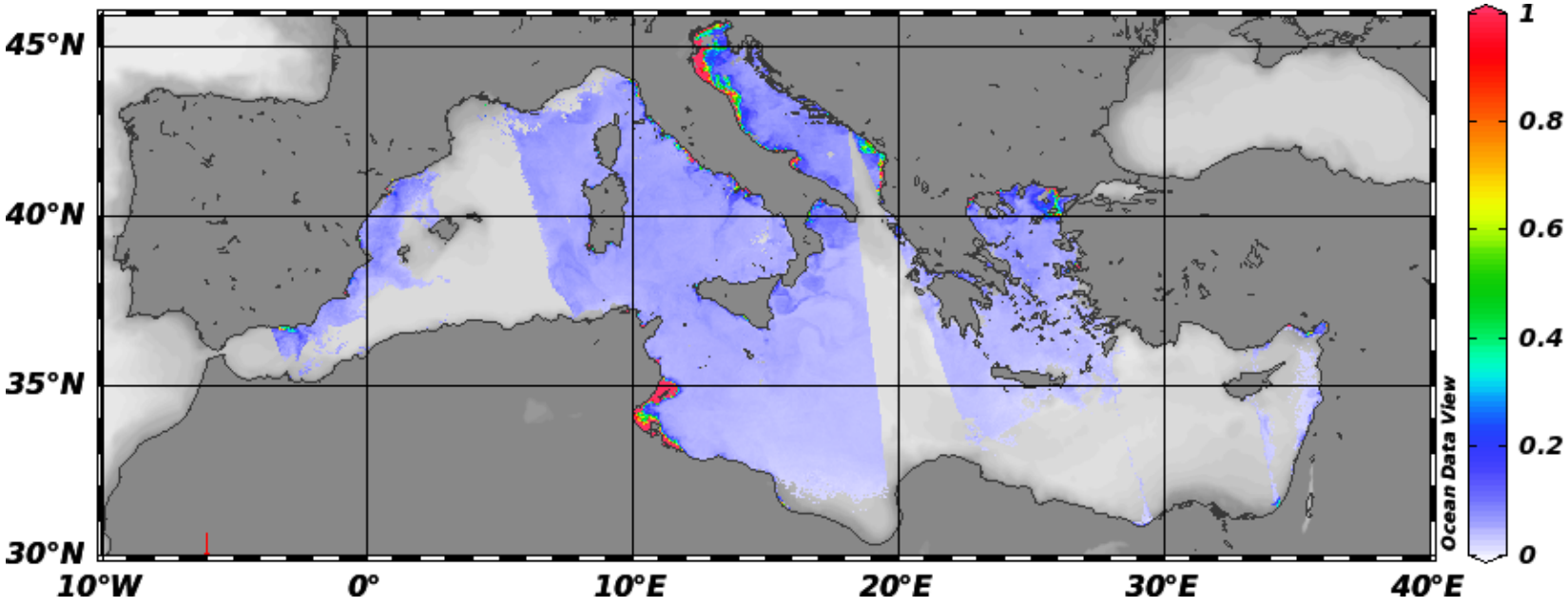
- Extract subregion:

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

ODV – Ocean Data View

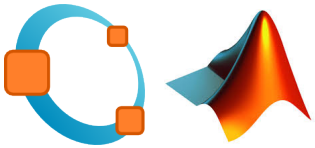
Home page: http://odv.awi.de/en/home/

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



Wanna know more? Click here

Octave / Matlab



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

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

ODV in 3 lines

> 40000 registered users

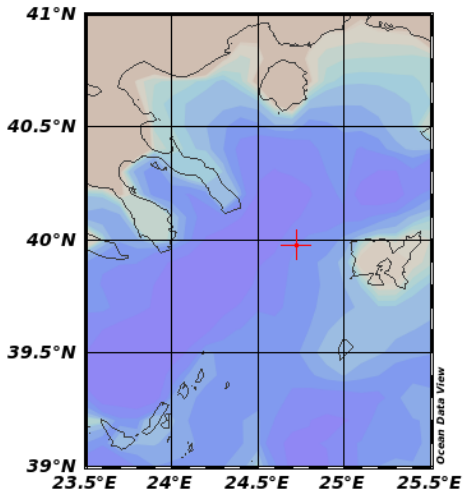
Data analysis + visualisation

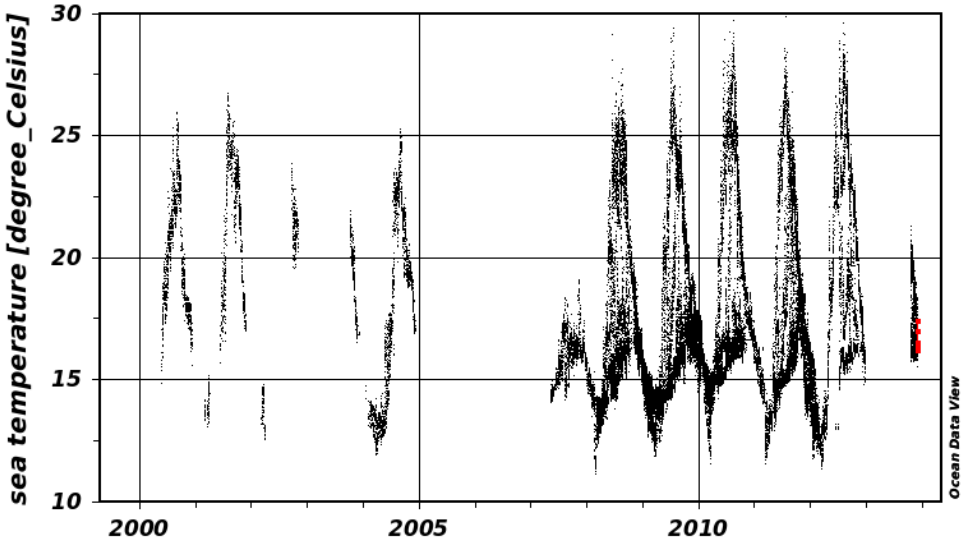
Almost every format supported

Working with ODV on Time Series

Objective: plotting time series

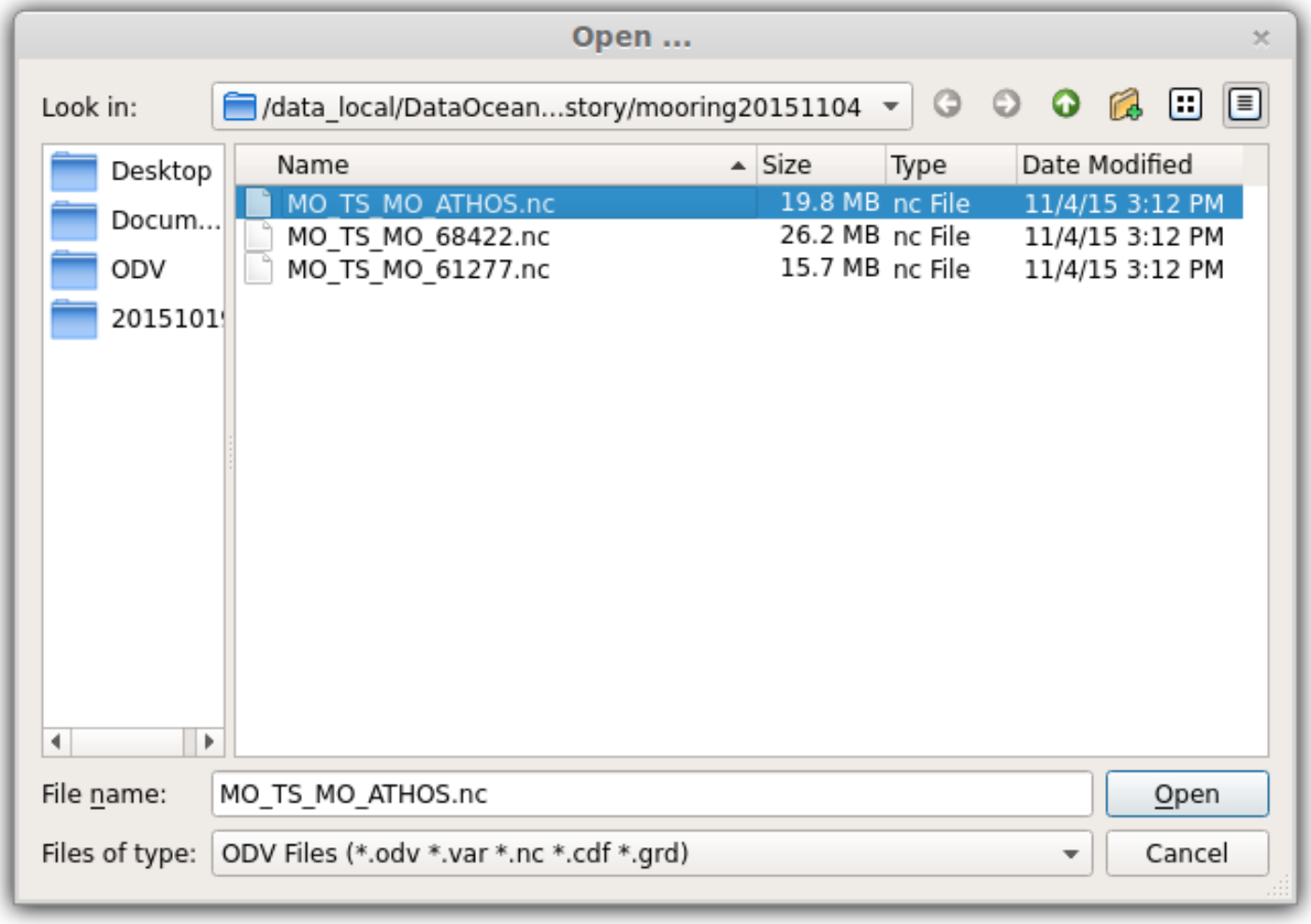
Temperature at mooring Athos





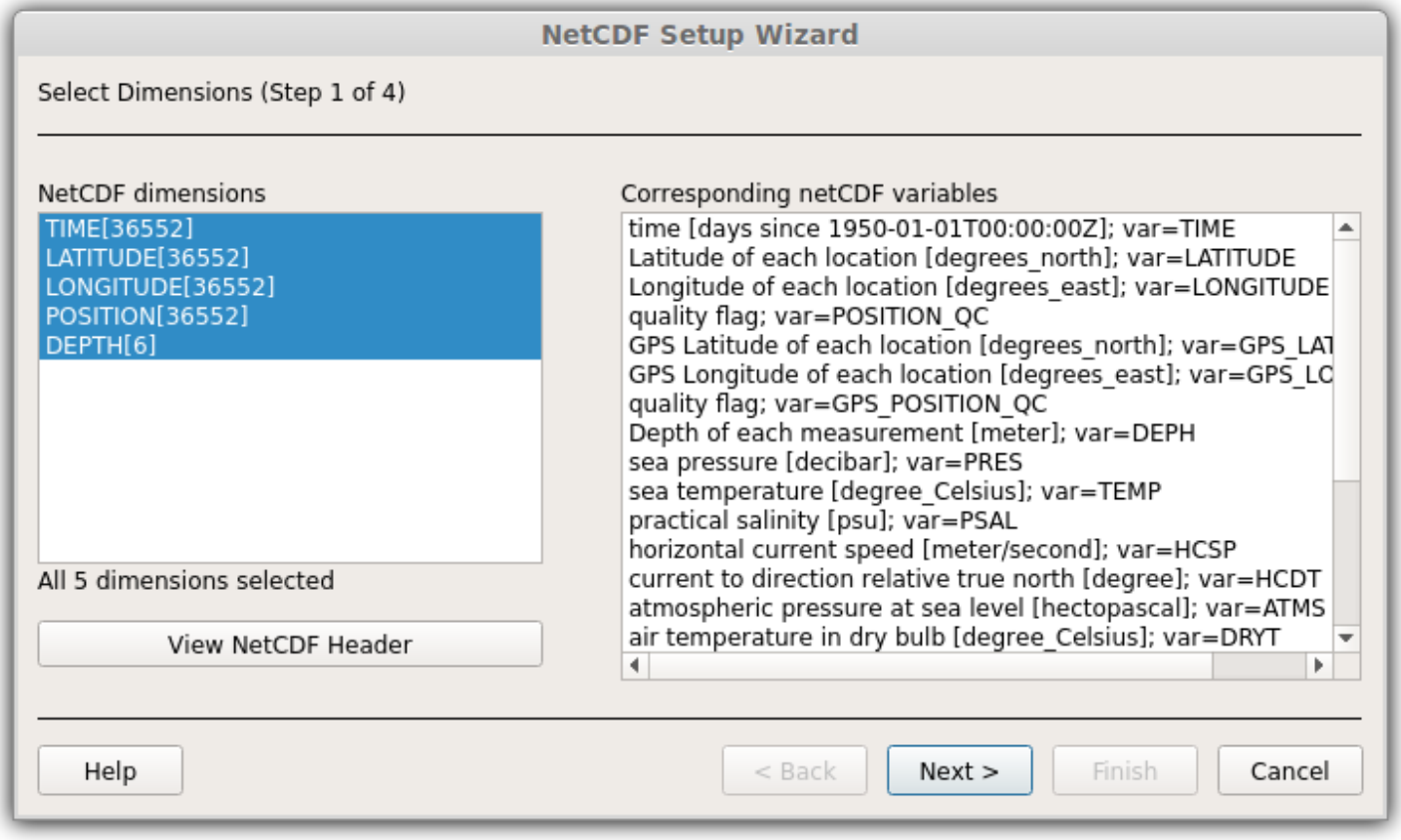
Opening the netCDF file

File → Open → netcdf



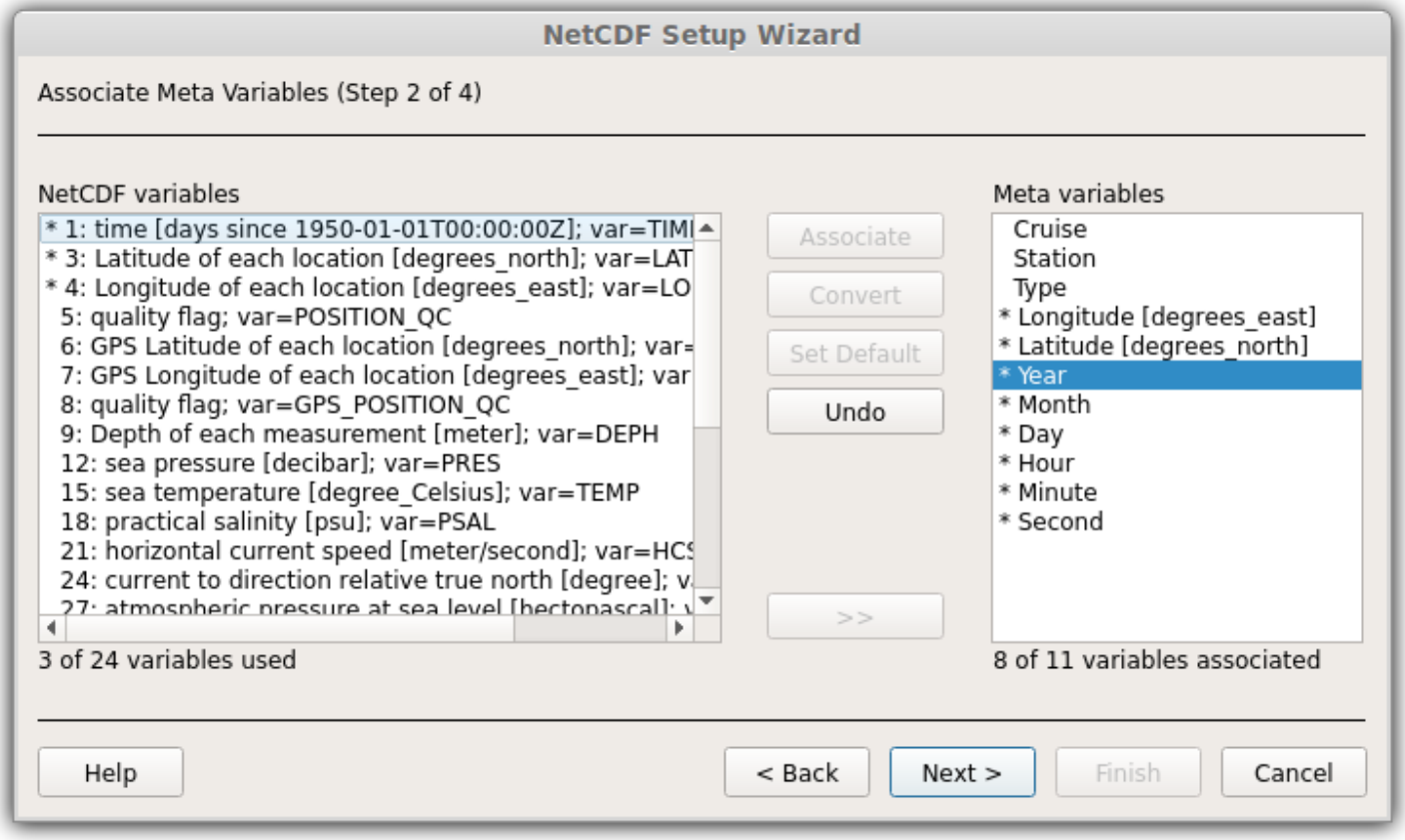
Opening the netCDF file

Dimension and variables : Next



Opening the netCDF file

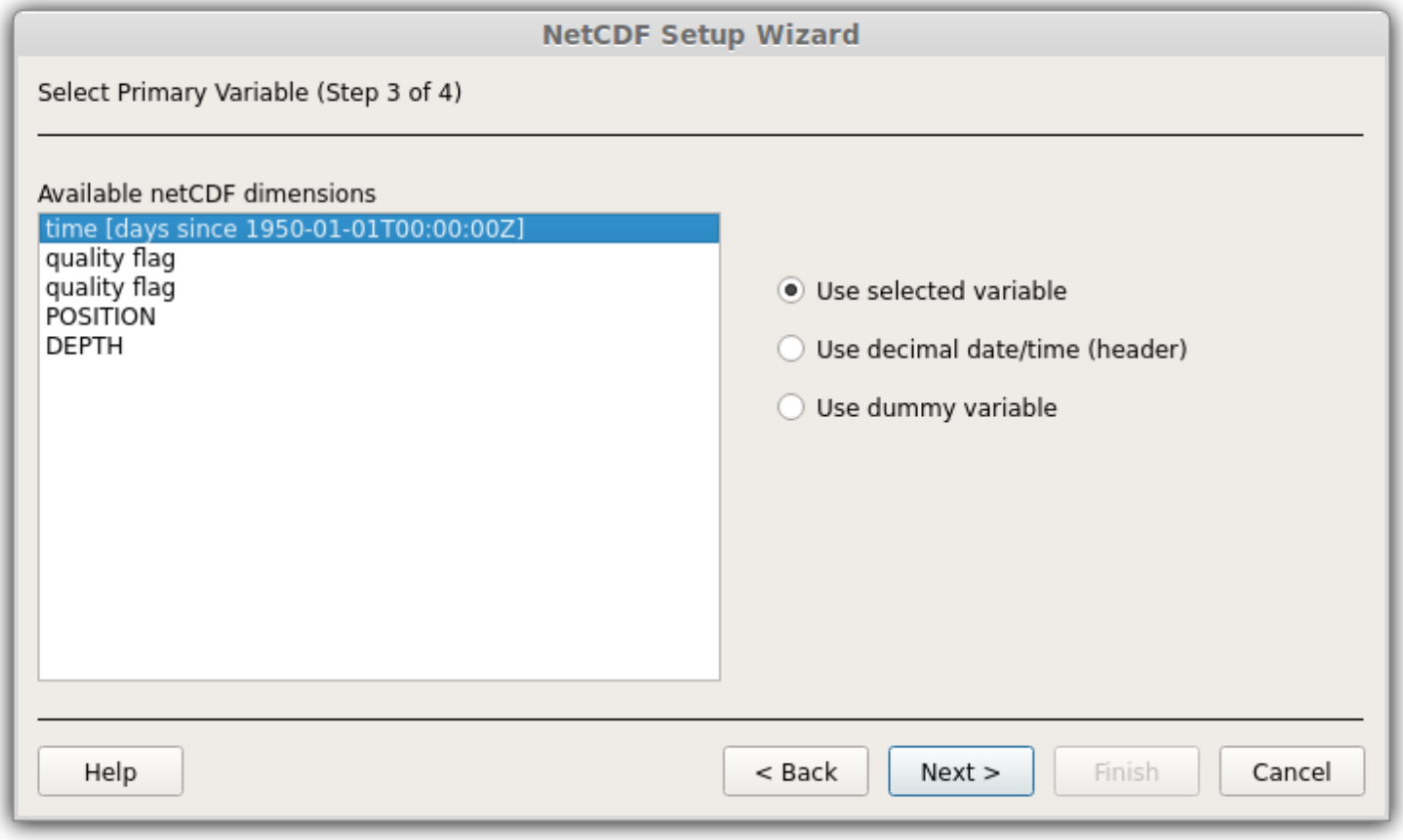
Variable association : Next



Quality Control variables not visible at this stage

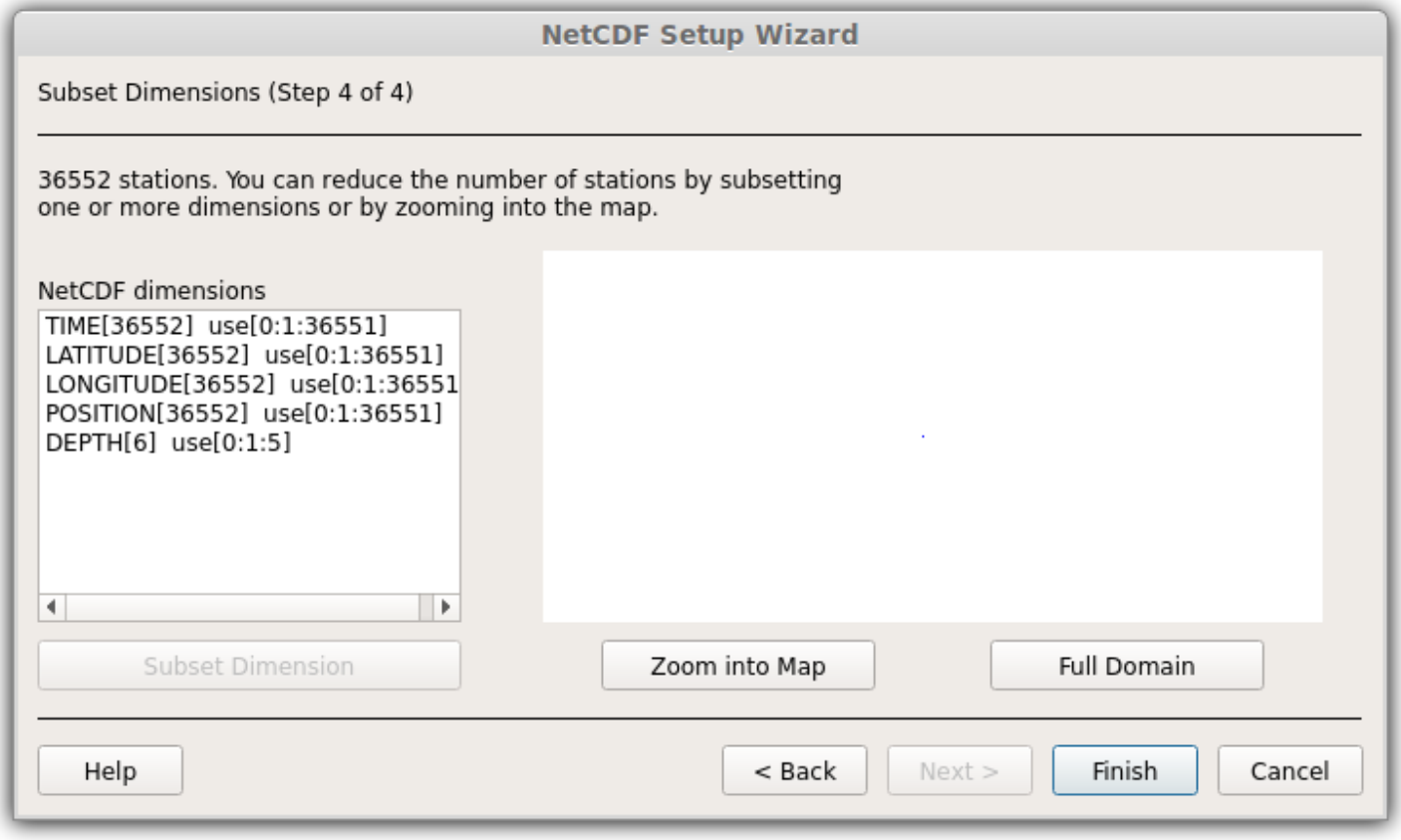
Opening the netCDF file

Primary variables : Next



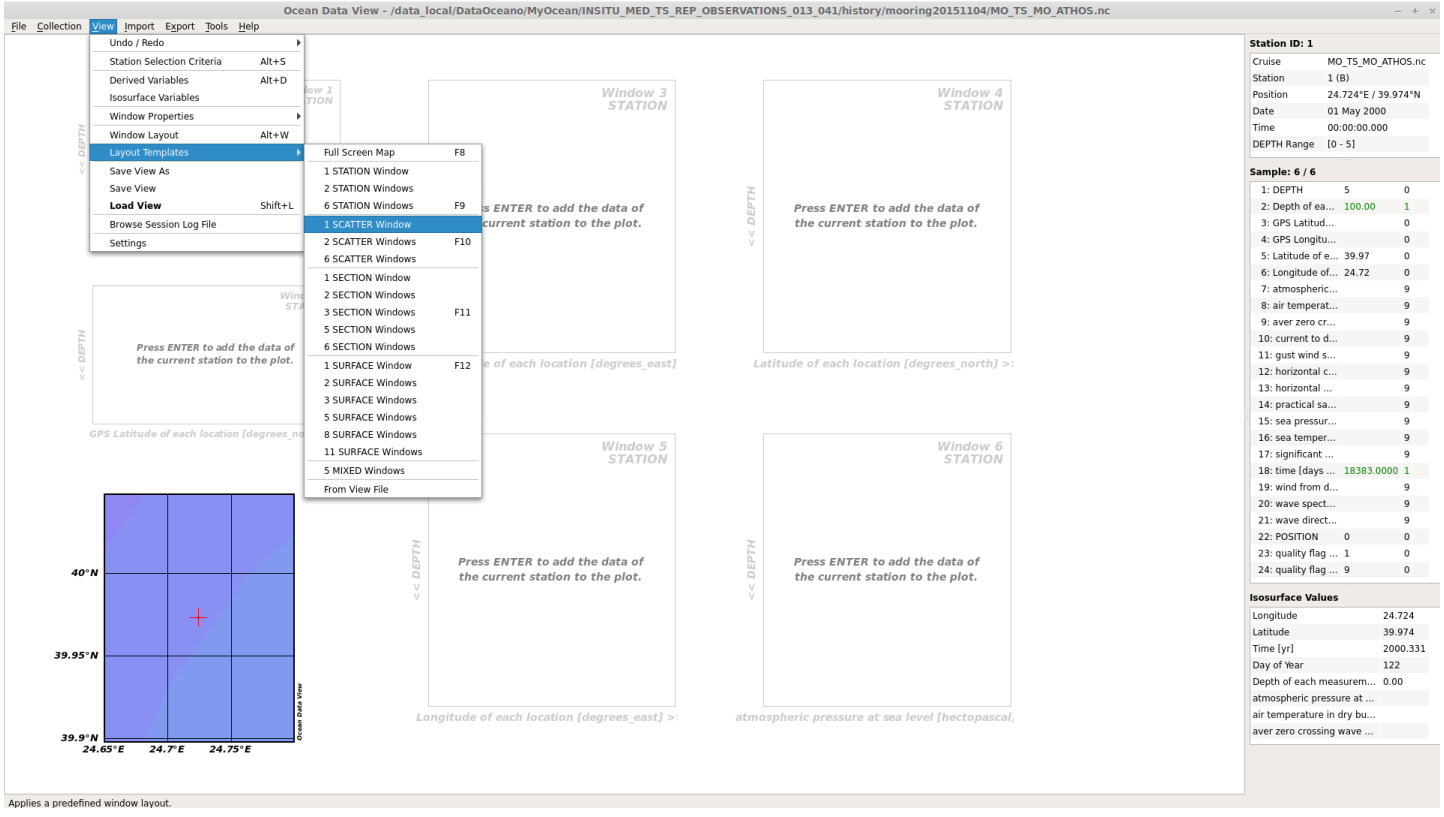
Opening the netCDF file

Subset dimensions : Finish



Plot the time series

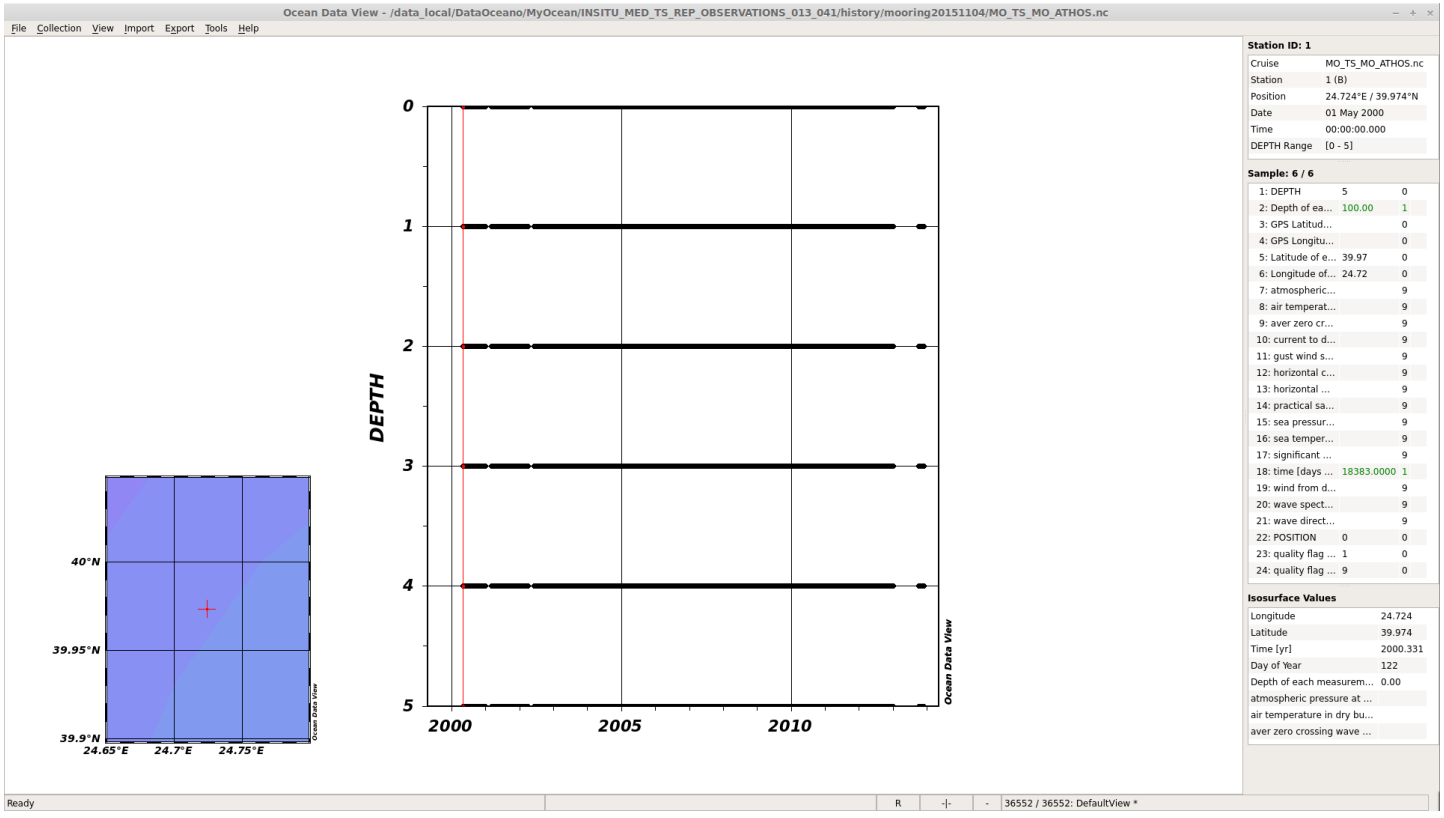
View → Layout template → Scatter window



Plot the time series

Right-click on plot:

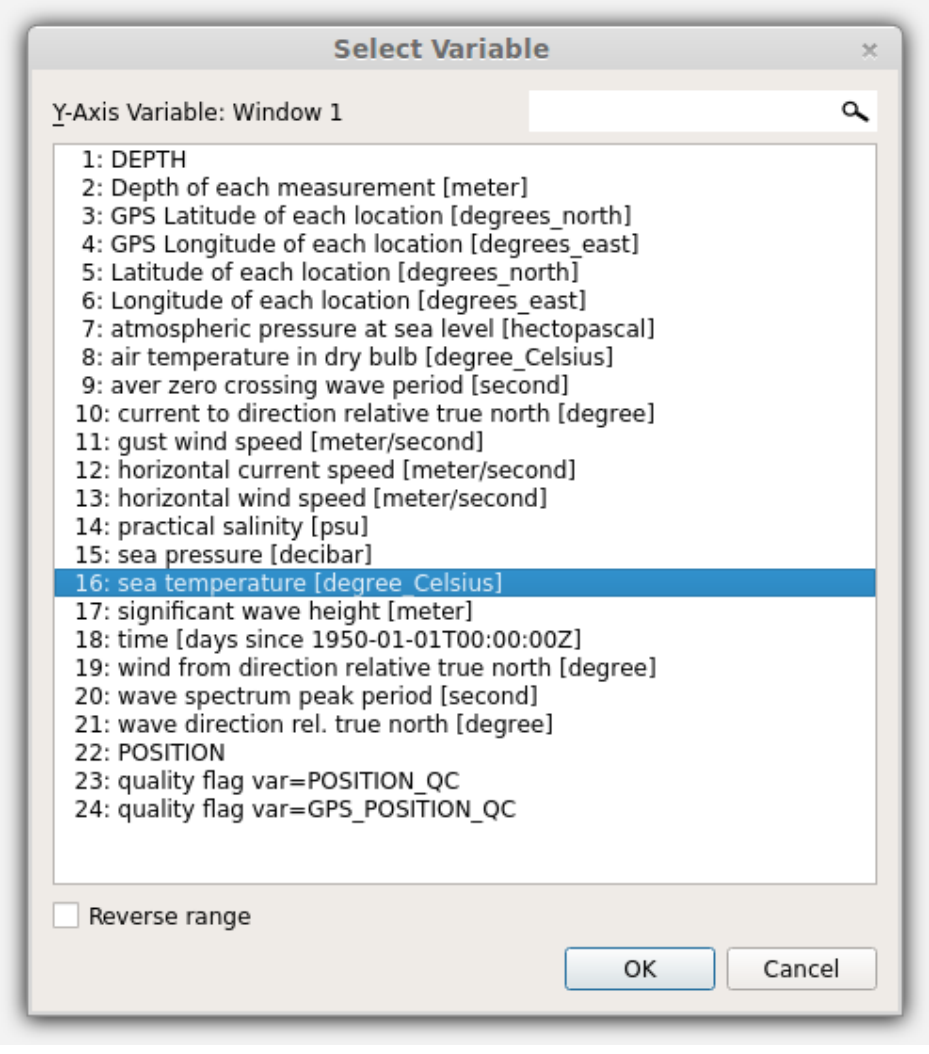
Change X and Y variable (temperature vs. time)



Plot the time series

Right-click on plot:

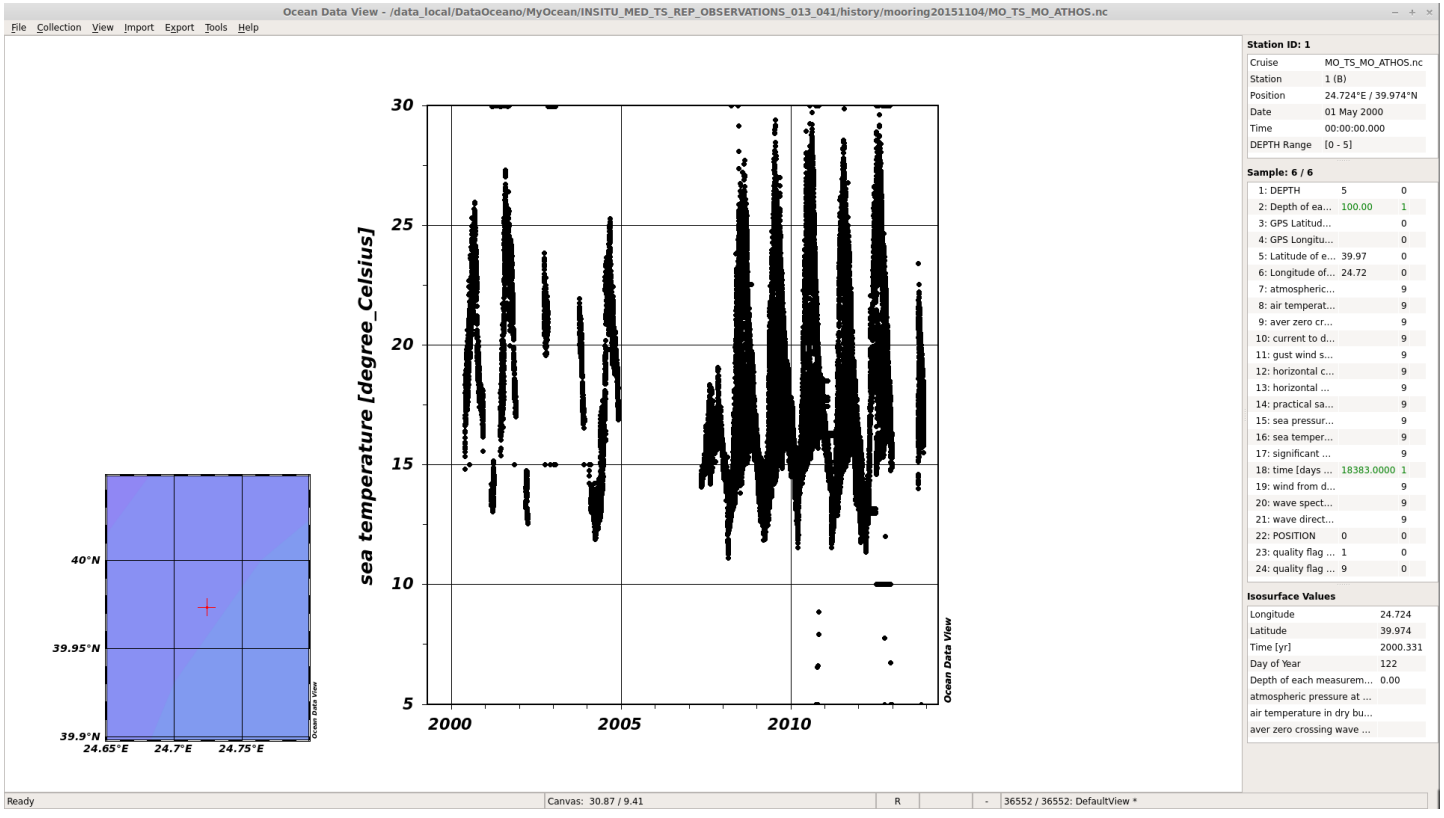
Change X and Y variable (temperature vs. time)



Plot the time series

Right-click on plot:

Change X and Y variable (temperature vs. time)

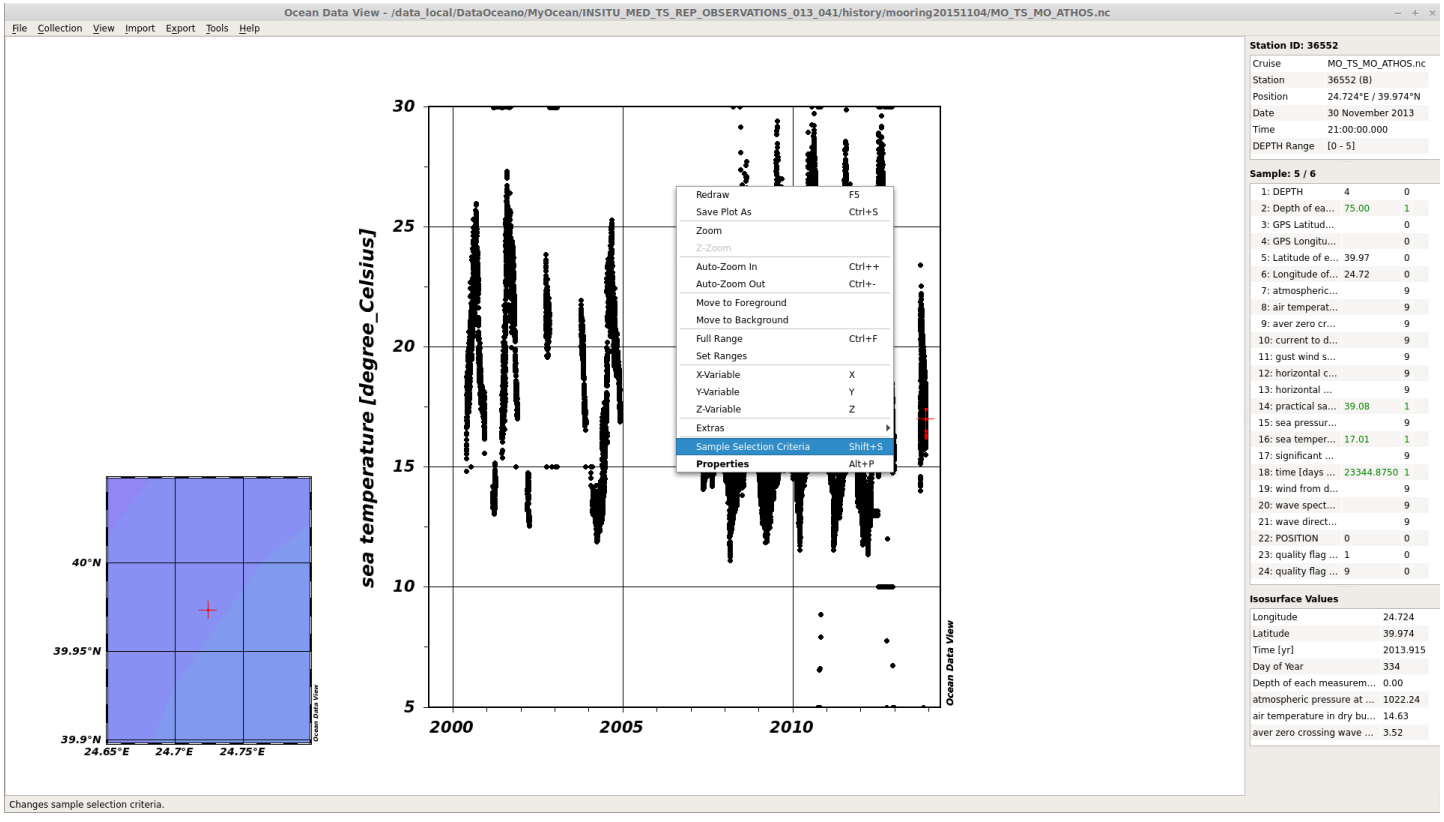


Bad values: will be treated later

Plot the time series

Right-click on plot: Sample Select Criteria

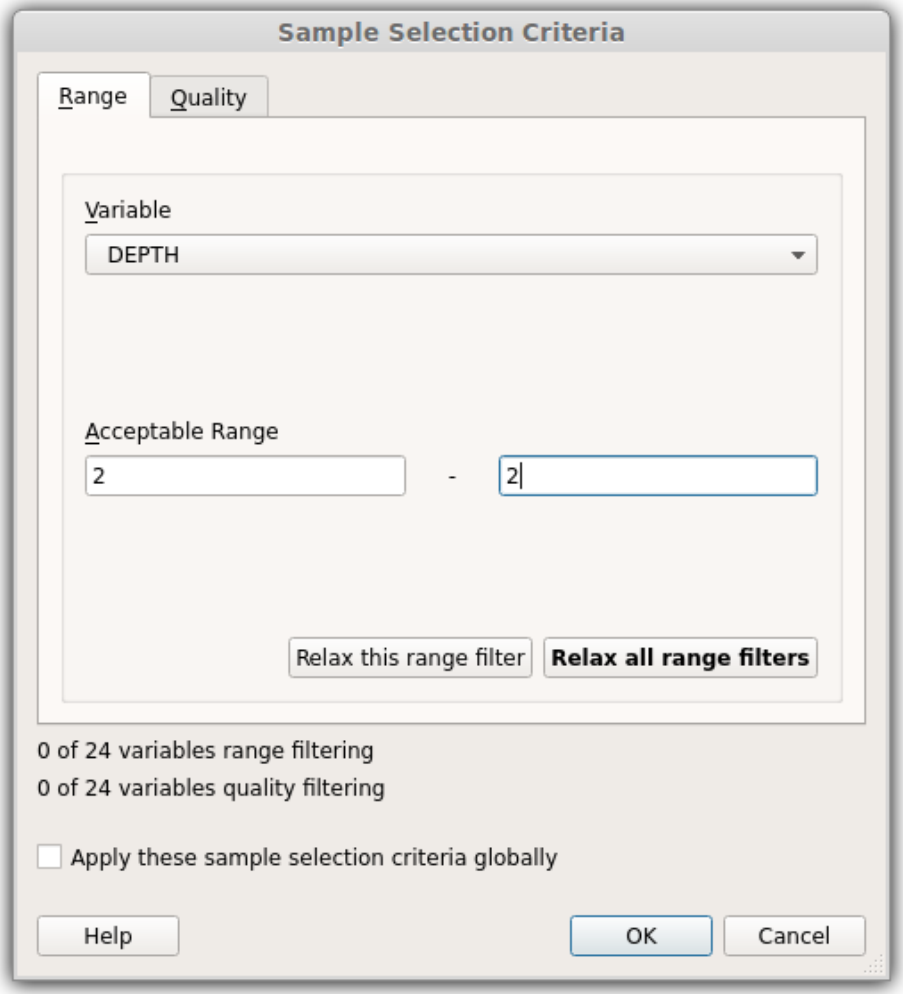
→ depth range



Plot the time series

Right-click on plot: Sample Select Criteria

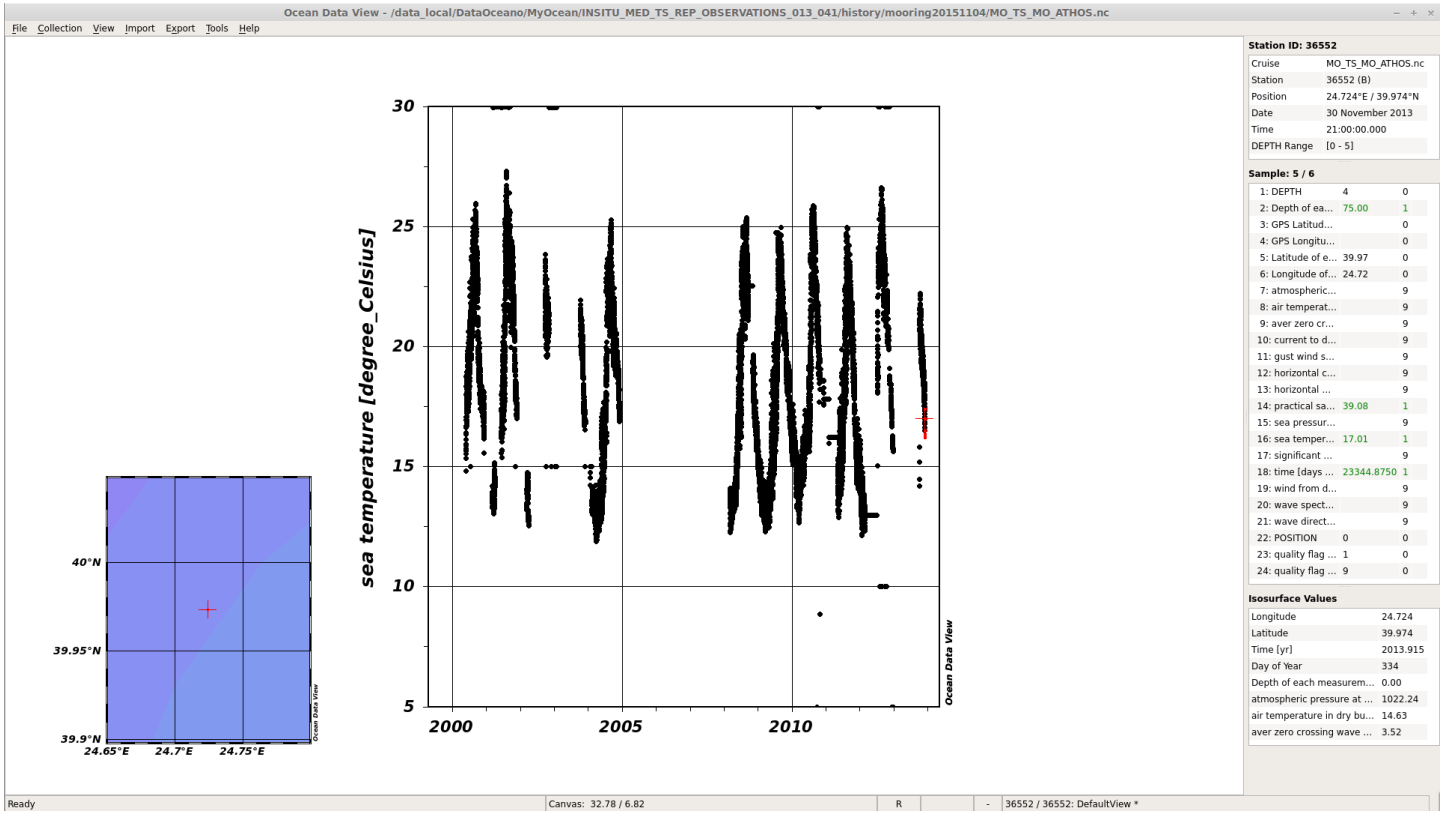
→ depth range



Plot the time series

Right-click on plot: Sample Select Criteria

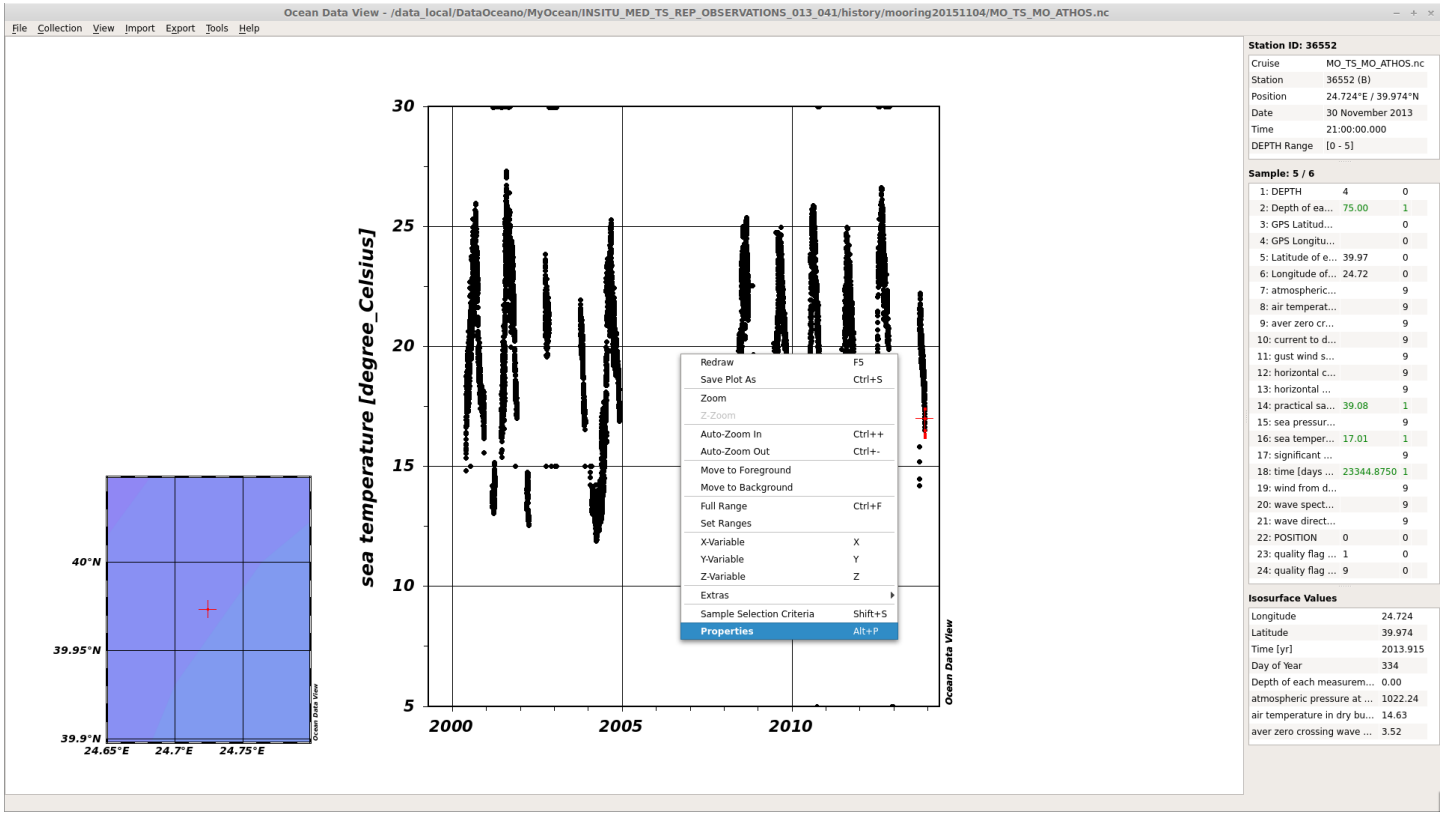
→ depth range



Now we have the series at 2 depth

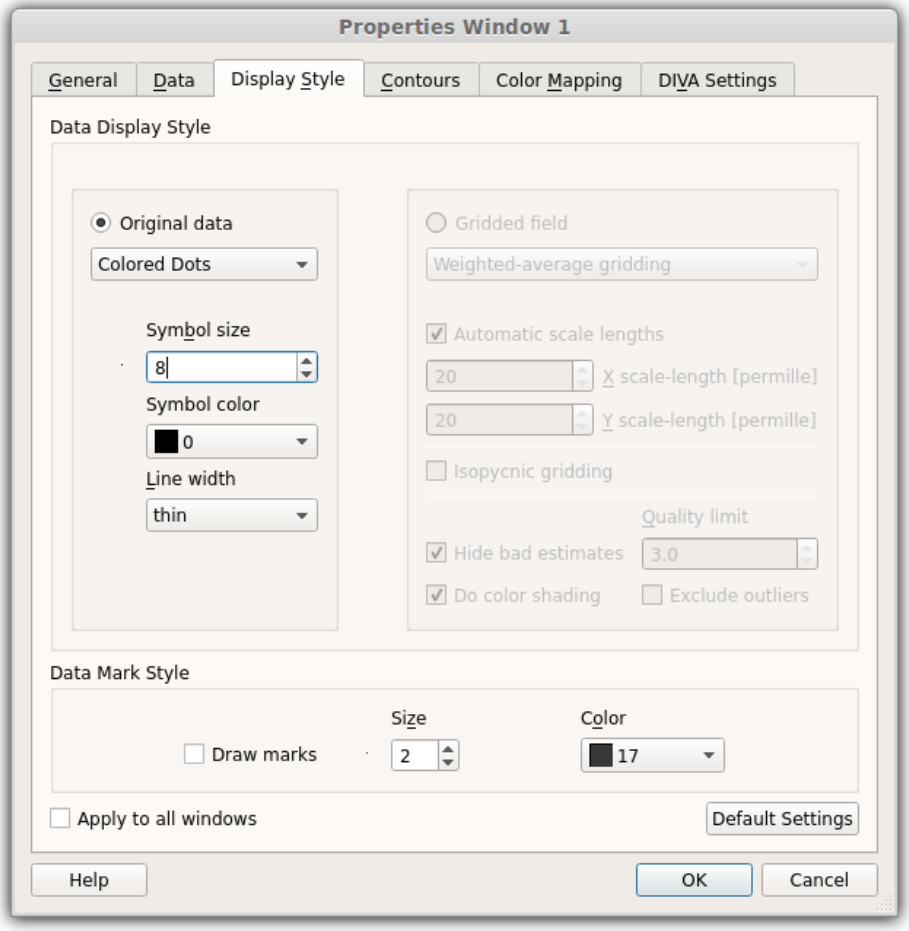
Improve the plot

- Right-click on plot: Properties



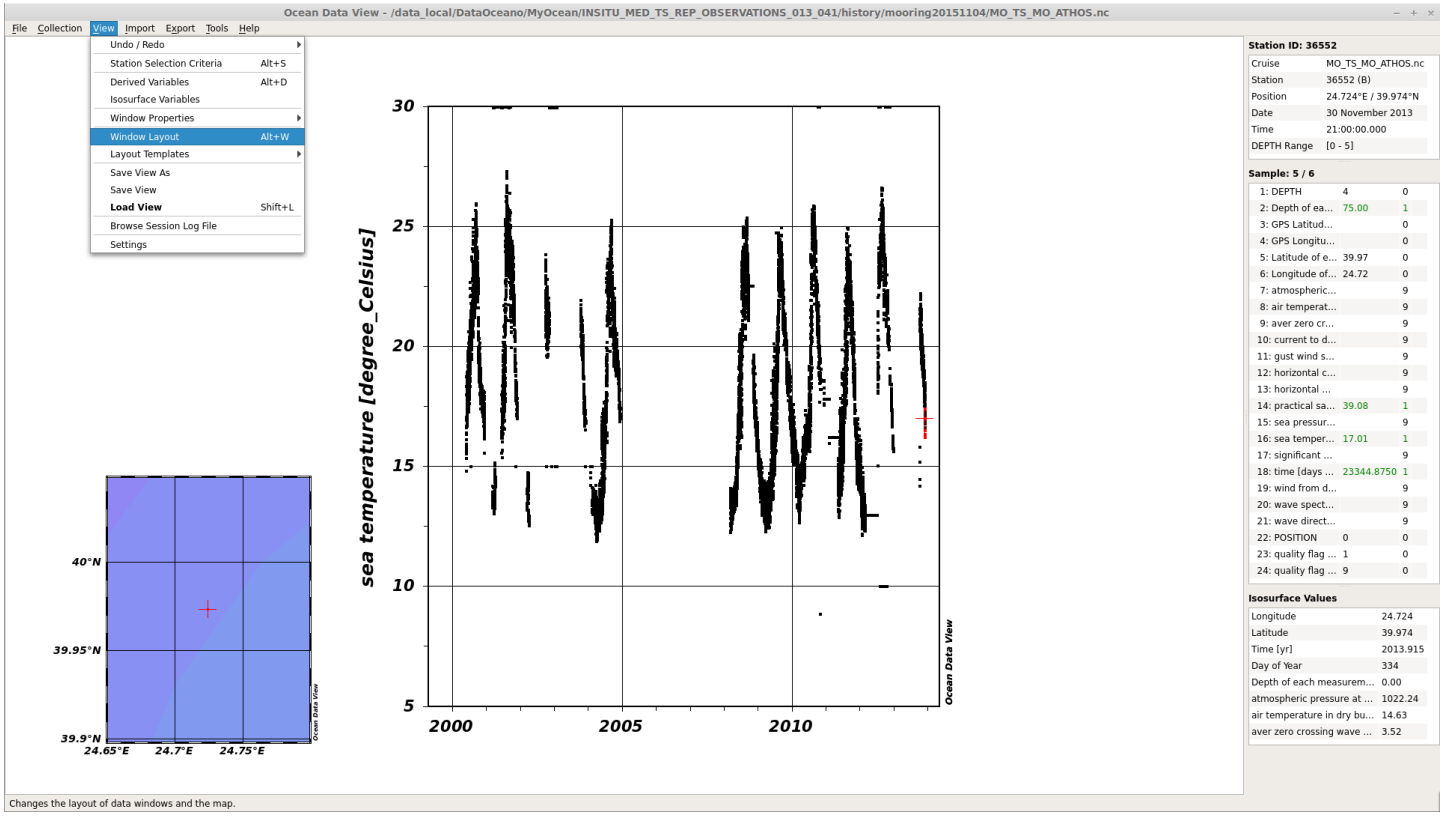
Improve the plot

Display Style: modify Symbols Size



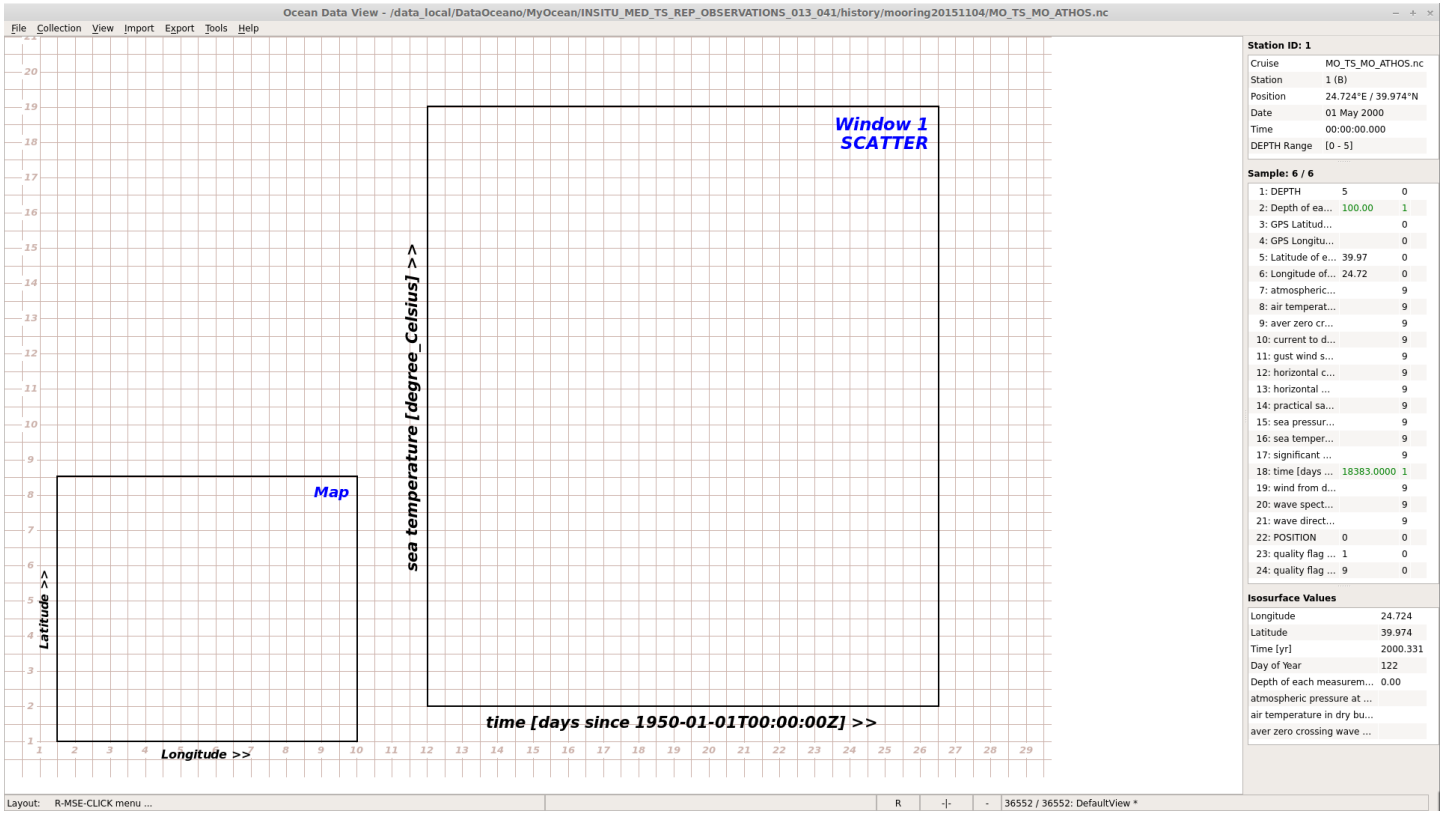
Improve the plot

View → Window Layout



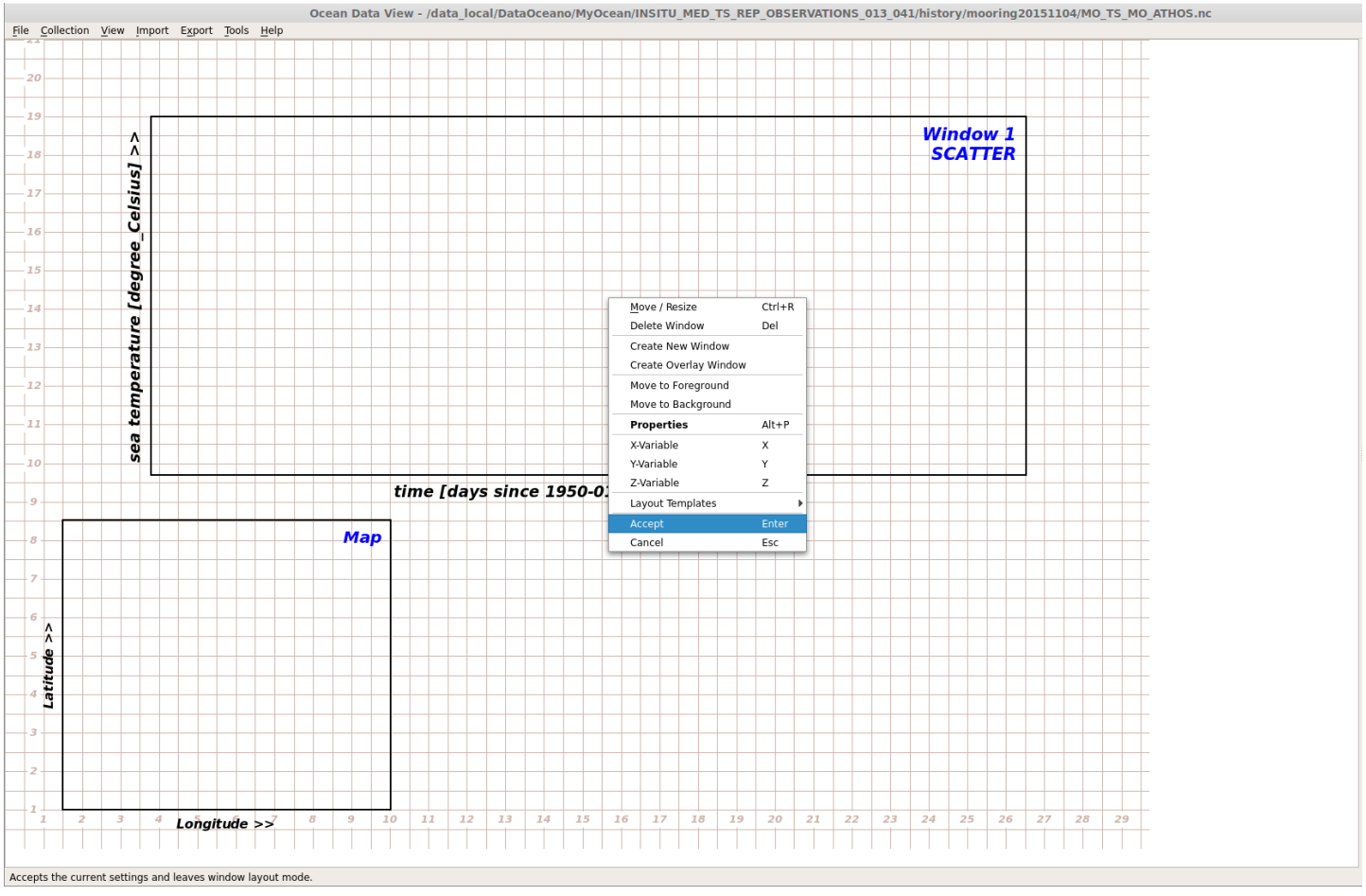
Improve the plot

Adapt size of the Scatter window



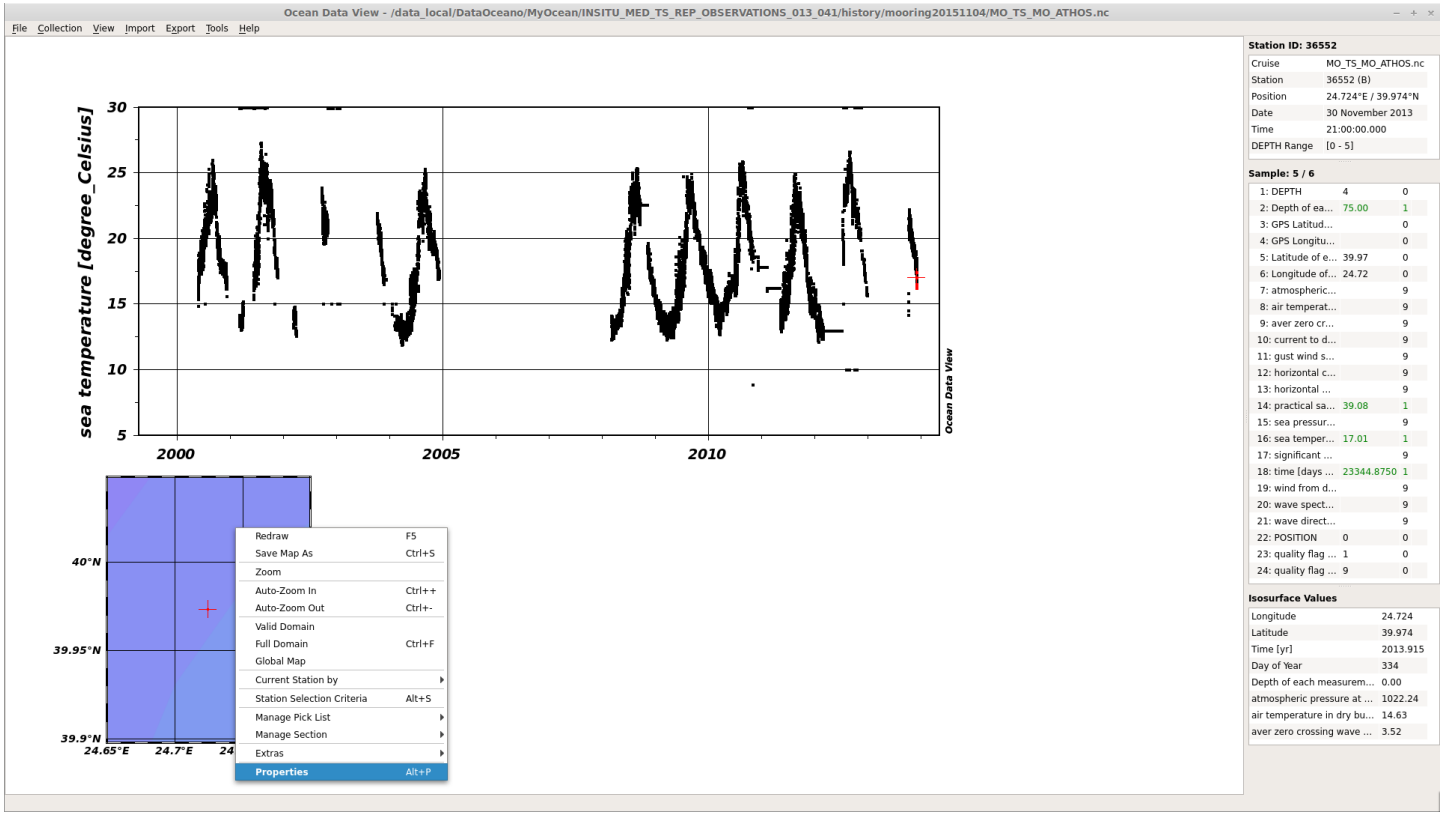
Improve the plot

Accept the change (Enter ↵)



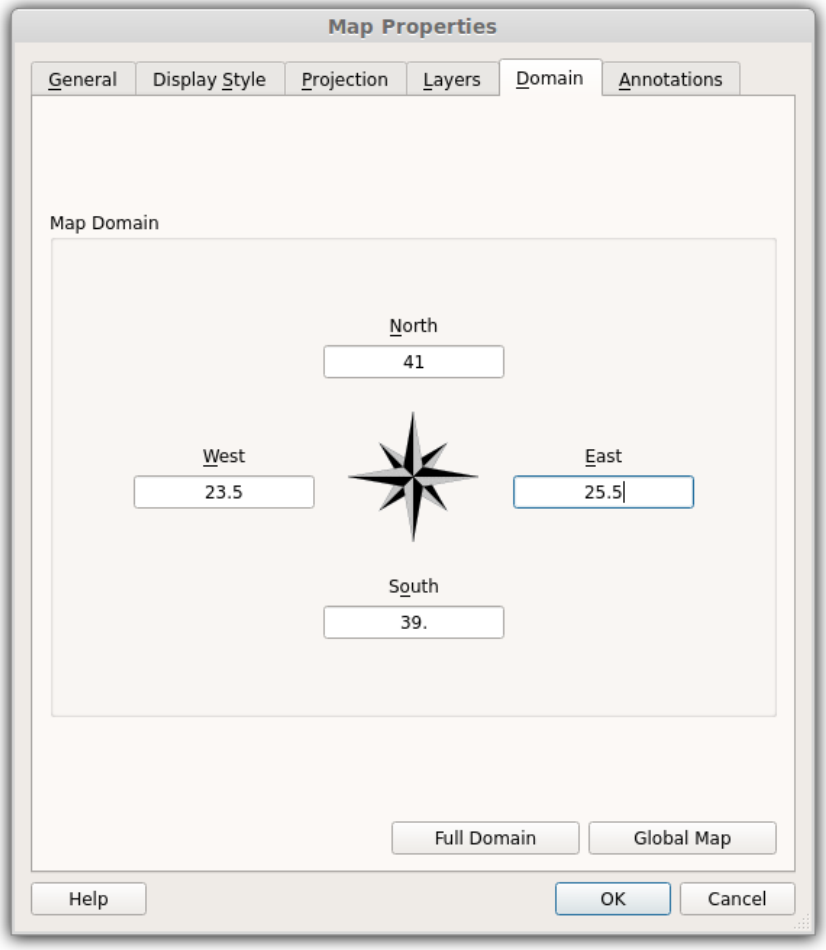
Improve the plot

- Right-click on plot: Properties → Domain



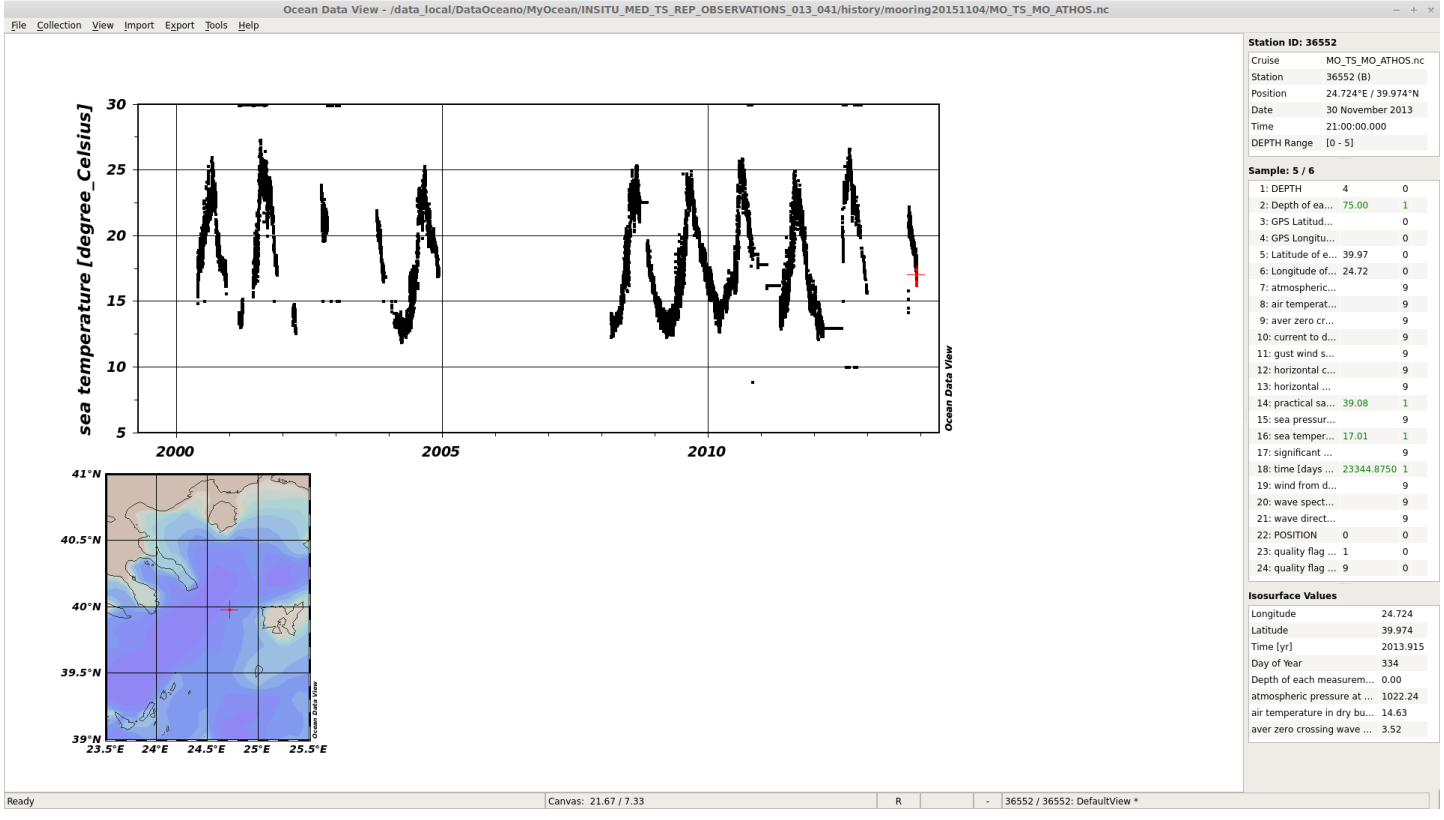
Improve the plot

Enlarge the map domain



Improve the plot

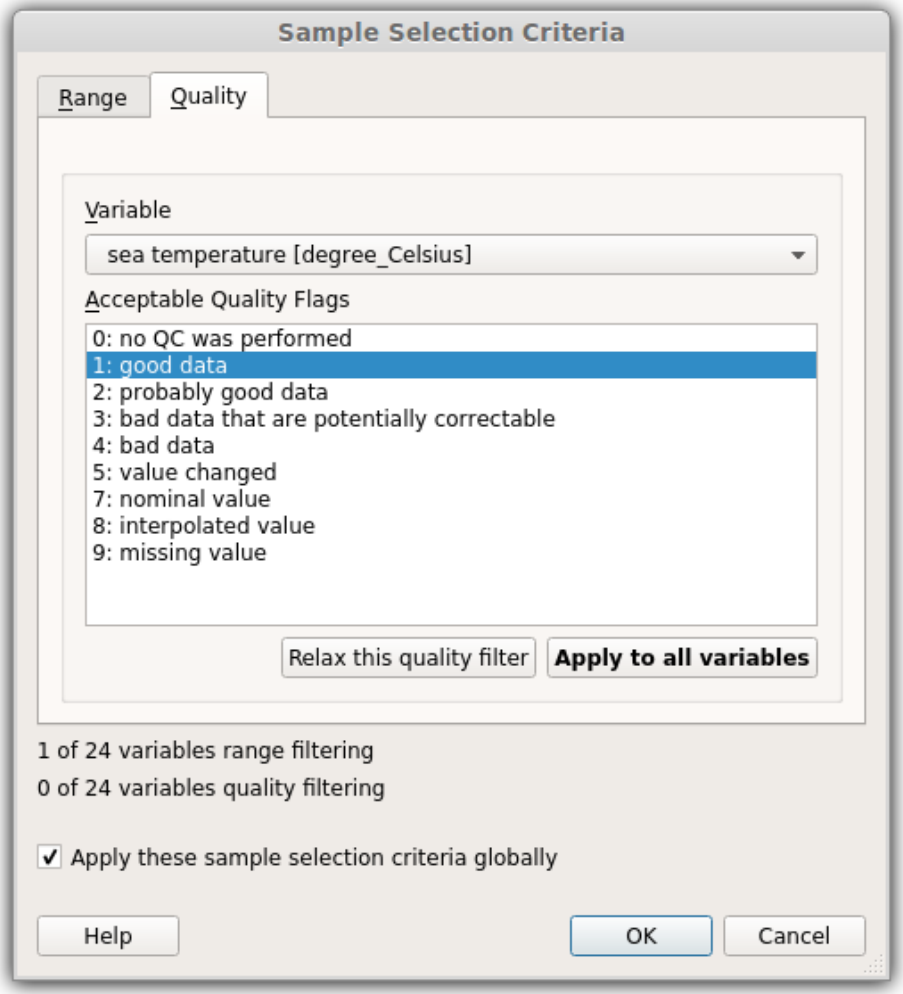
Enlarge the map domain



Apply quality flags

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

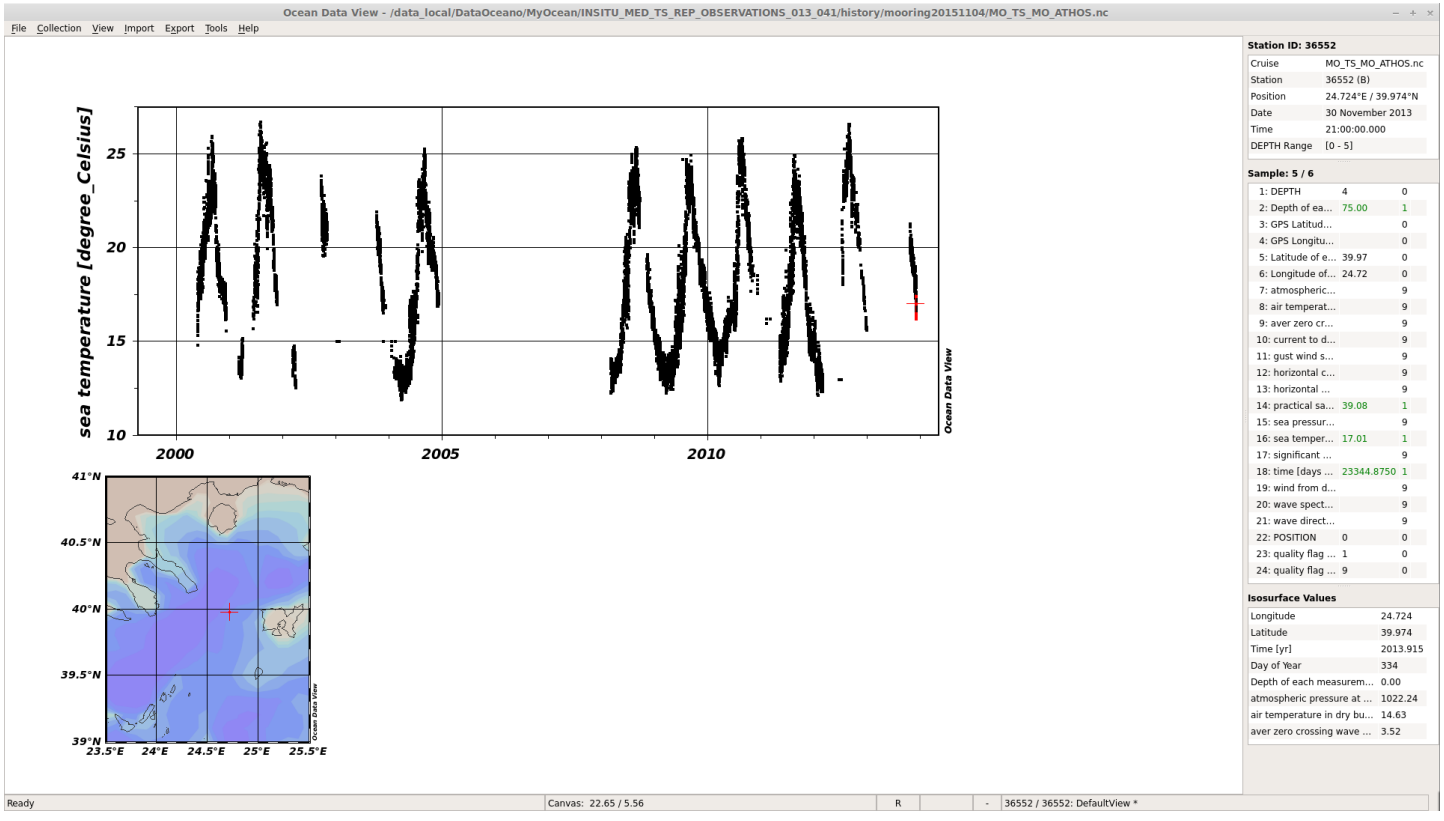
Select good data only



Apply quality flags

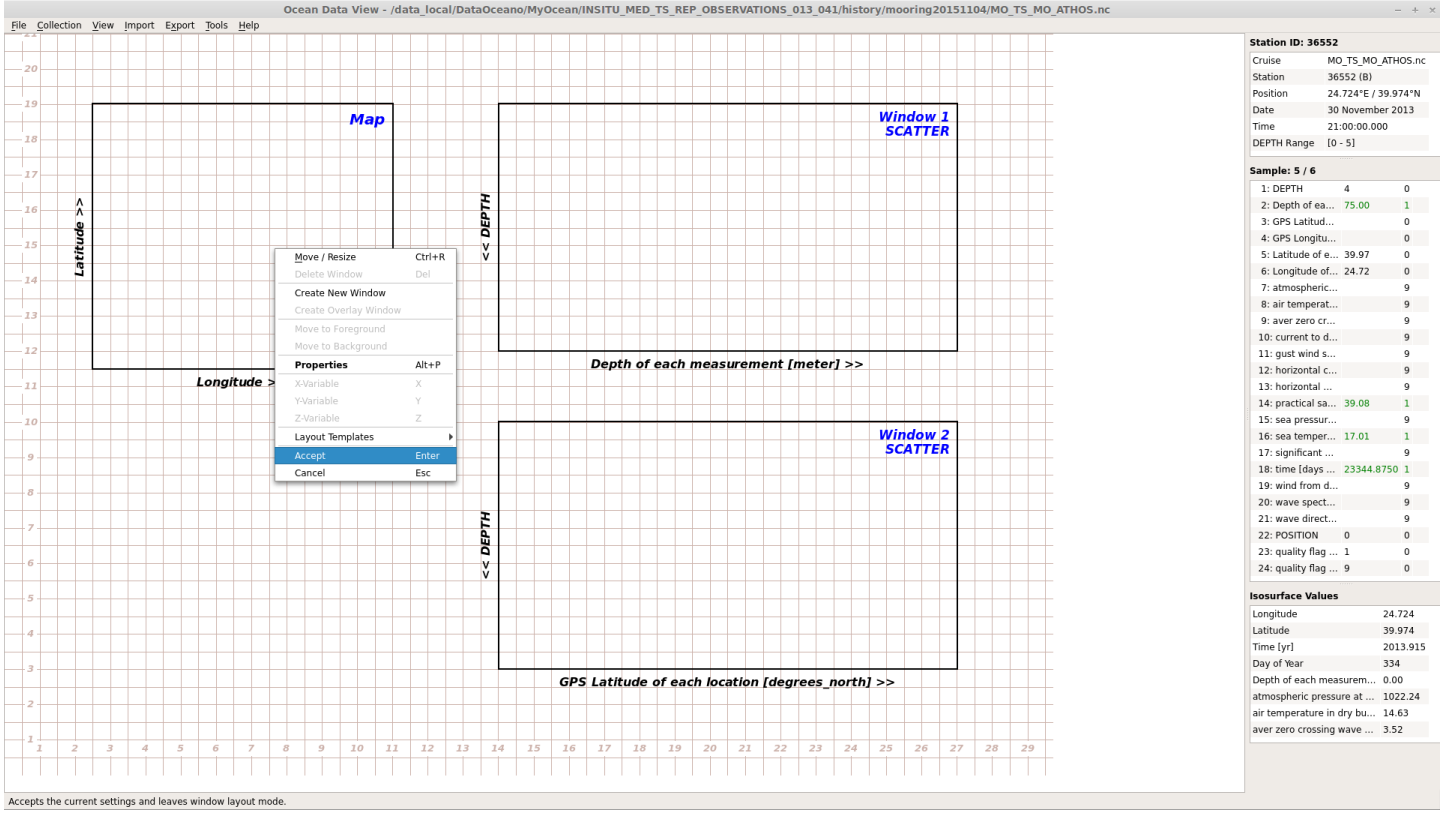
Right-click on plot: Sample Select Criteria → depth range

Select good data only



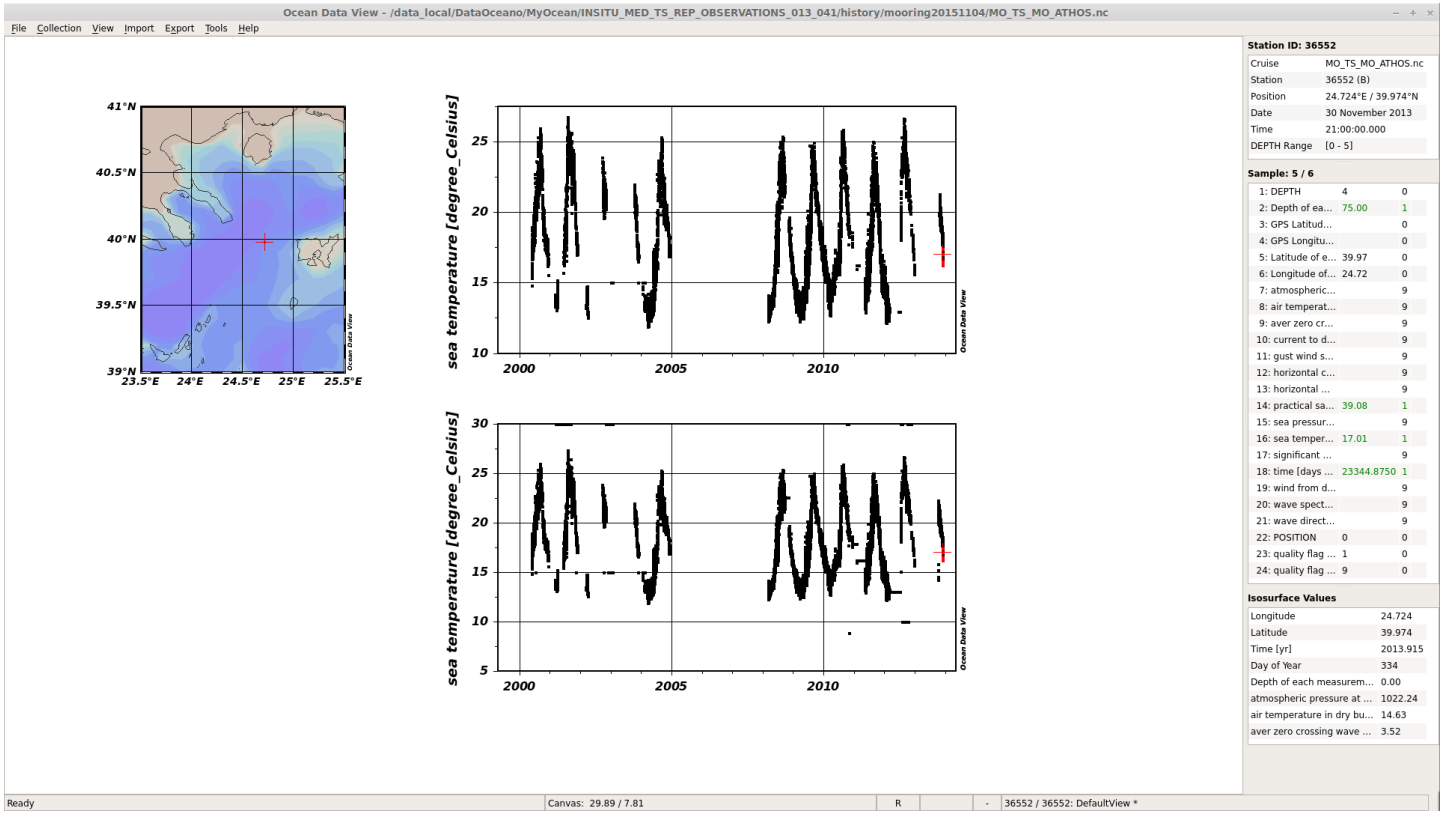
Apply quality flags

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

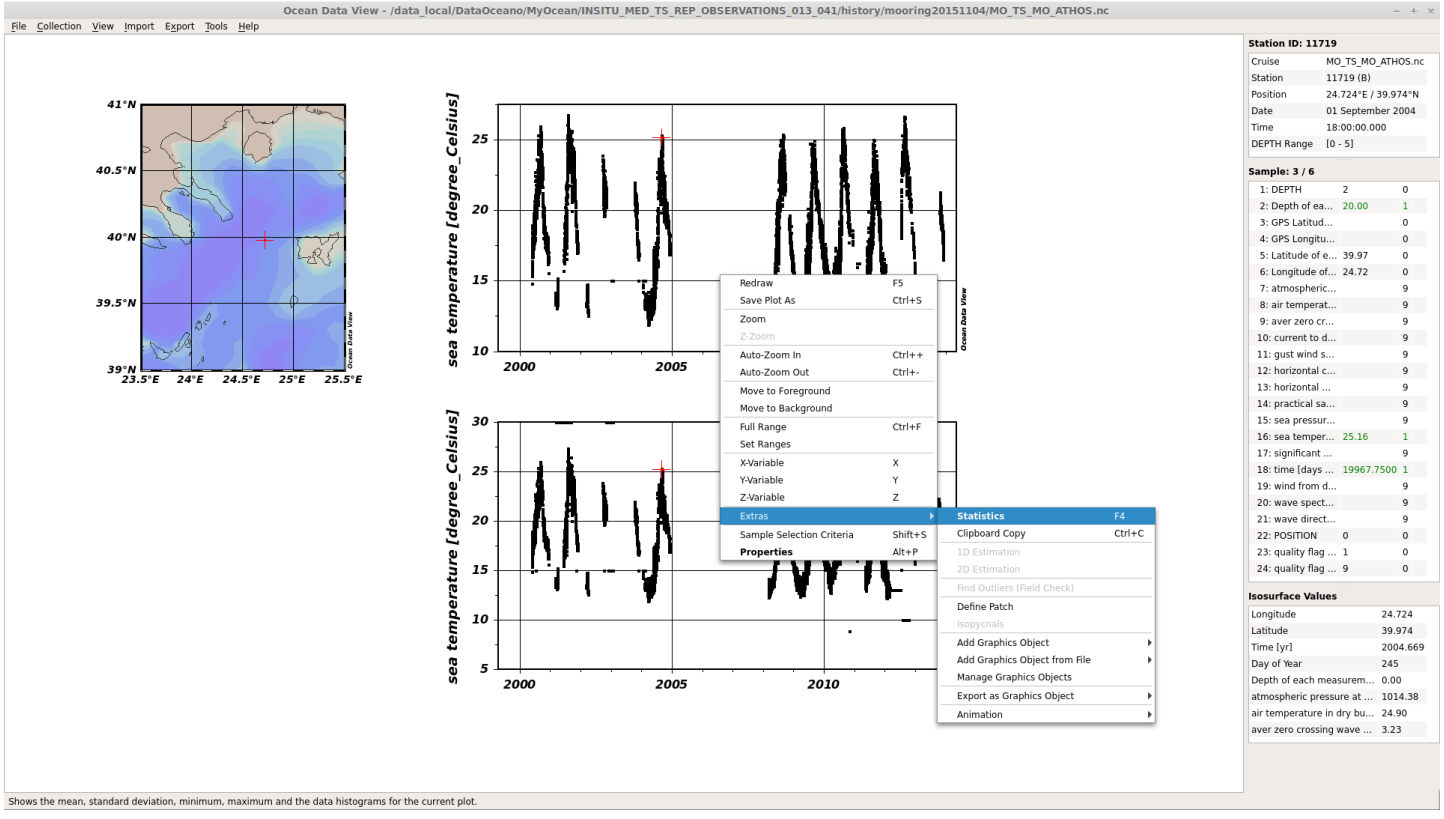


Extreme values are removed

Apply quality flags

Compare histograms:

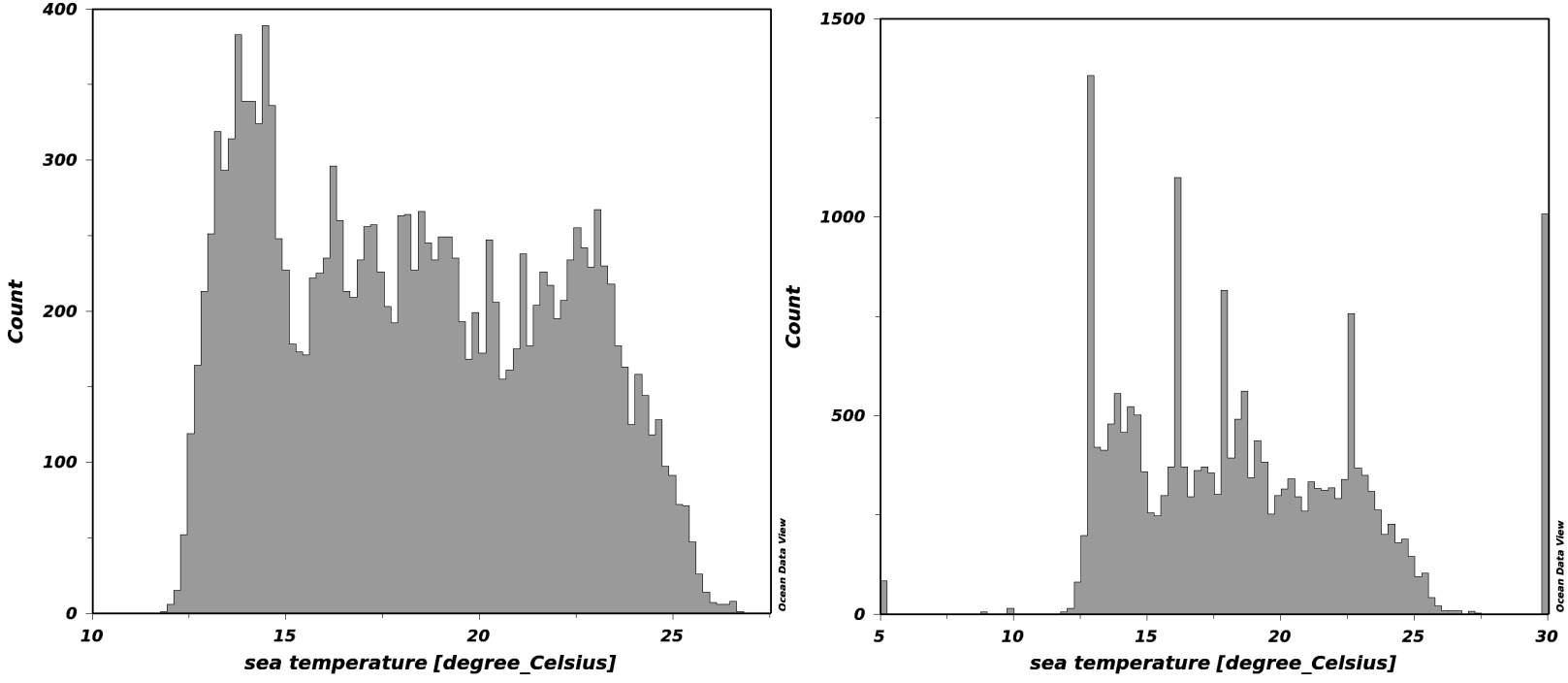
Right-click on plot → Extra → Statistics



Apply quality flags

Compare histograms:

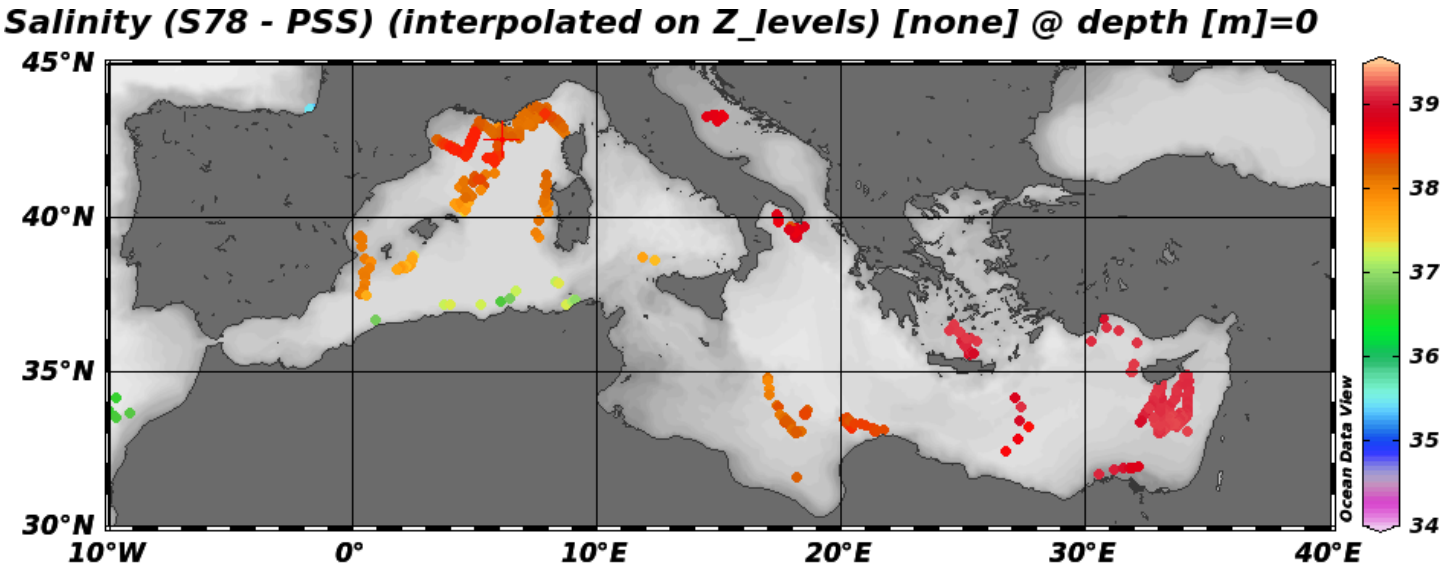
Right-click on plot → Extra → Statistics

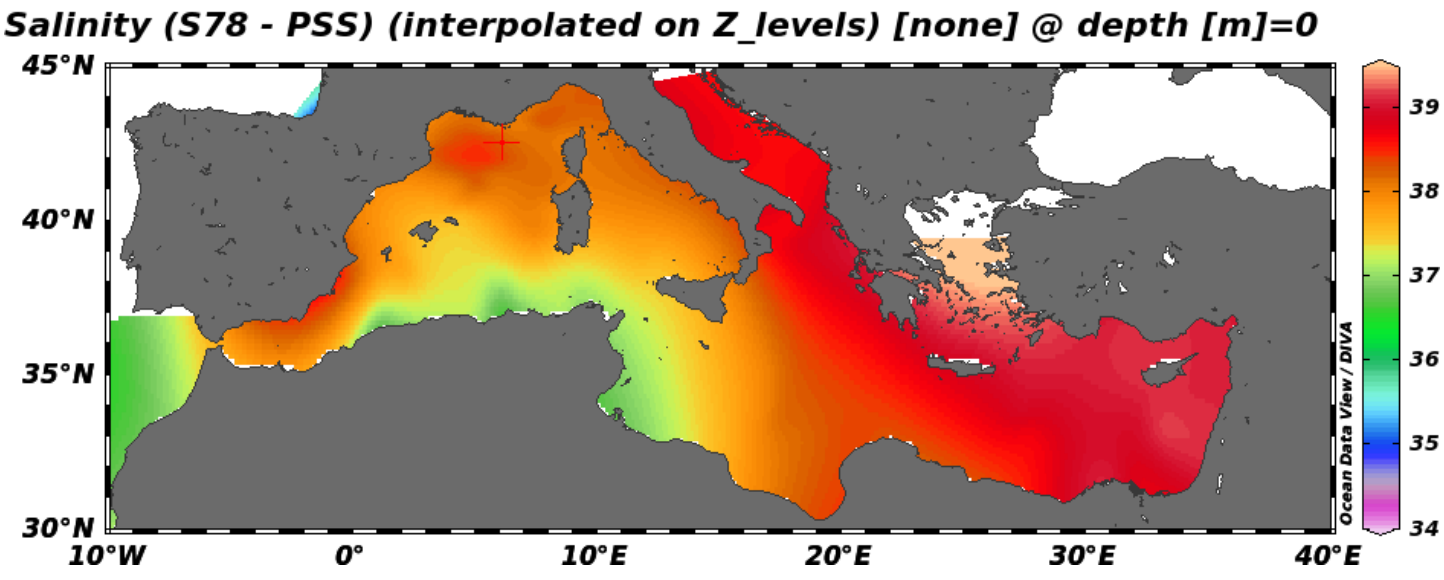


Distribution is improved

Working with ODV on CORA data set

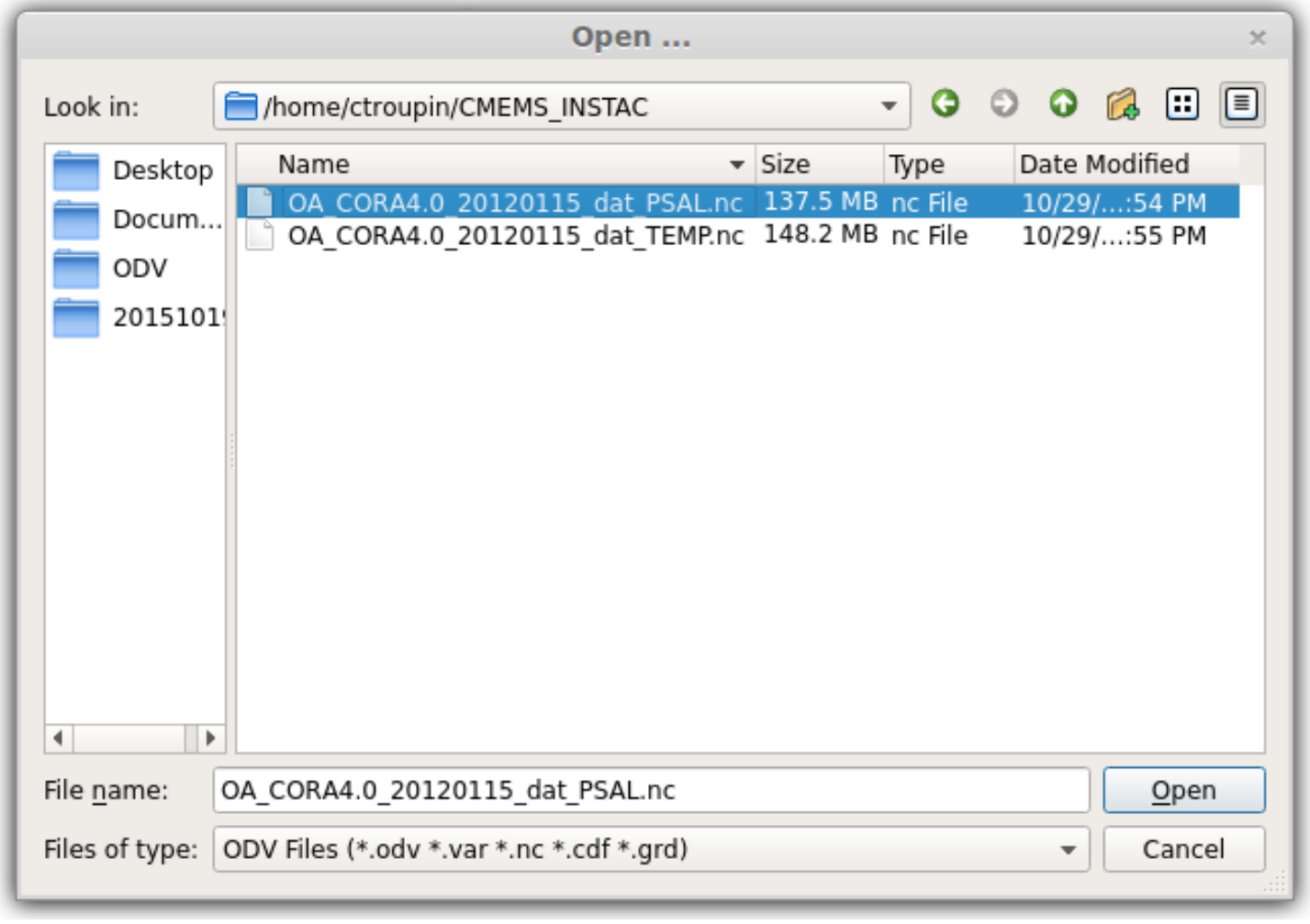
Objective: process CORA dataset





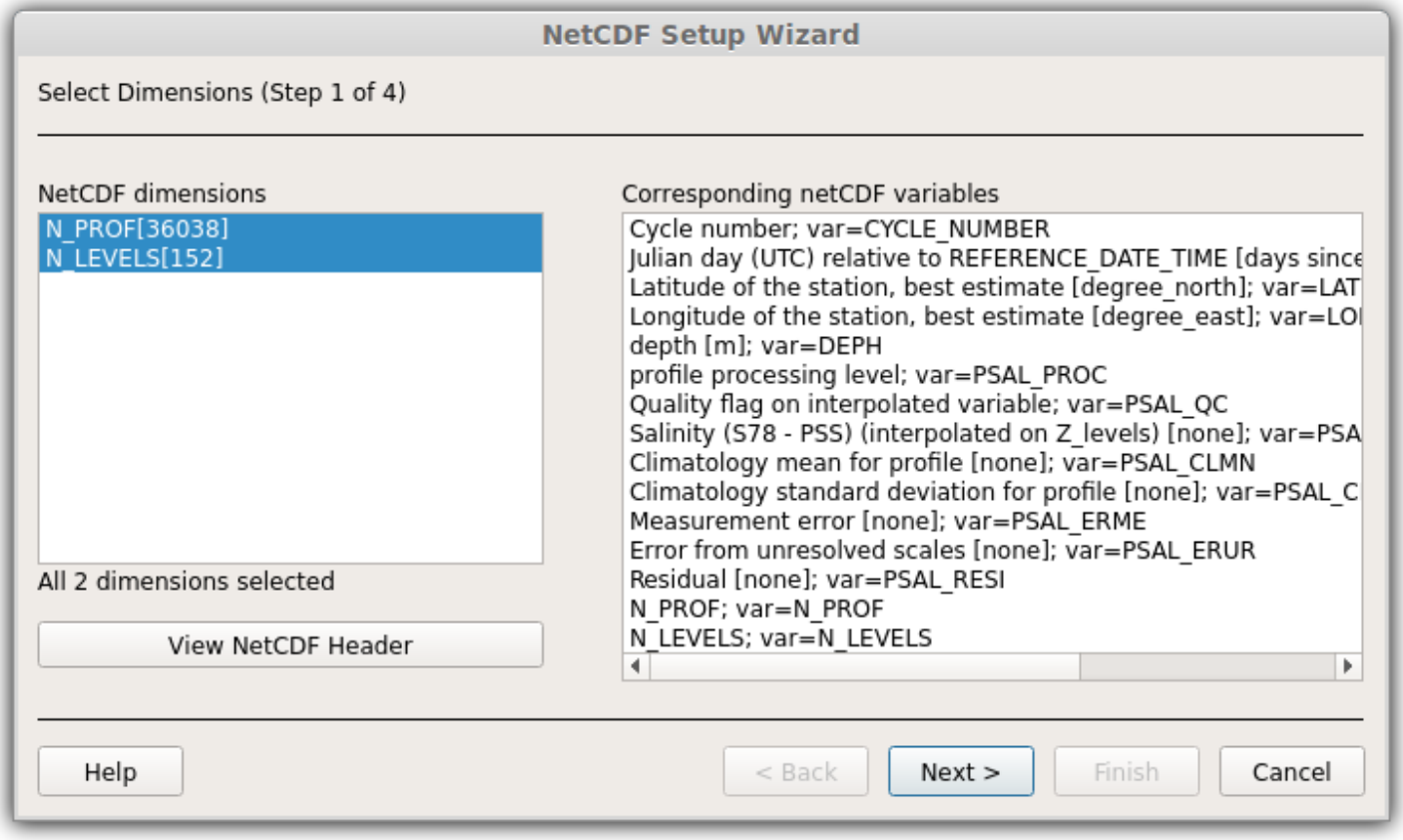
Opening the netCDF file

File → Open → Select the netCDF file



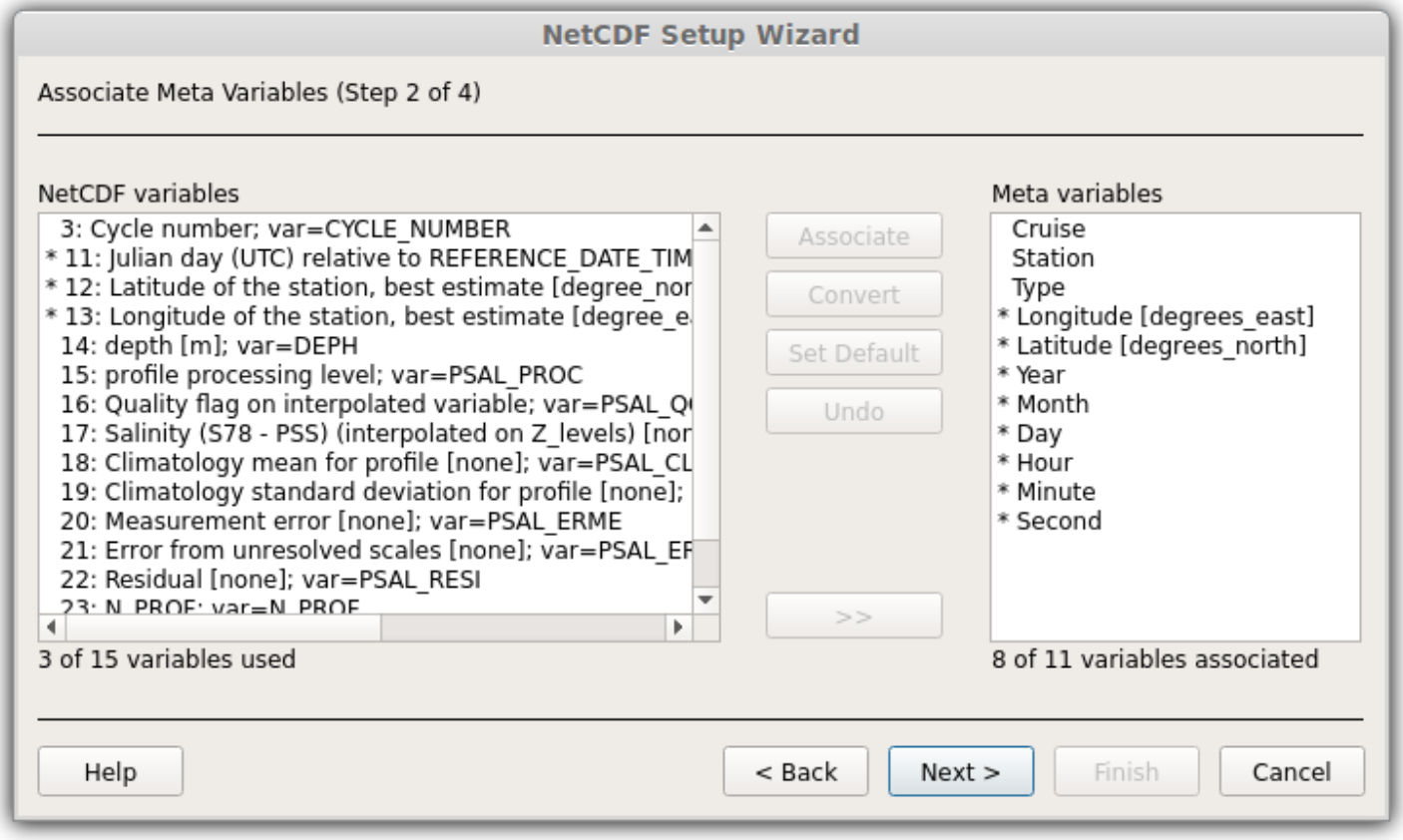
Opening the netCDF file

Dimension and variables : Next



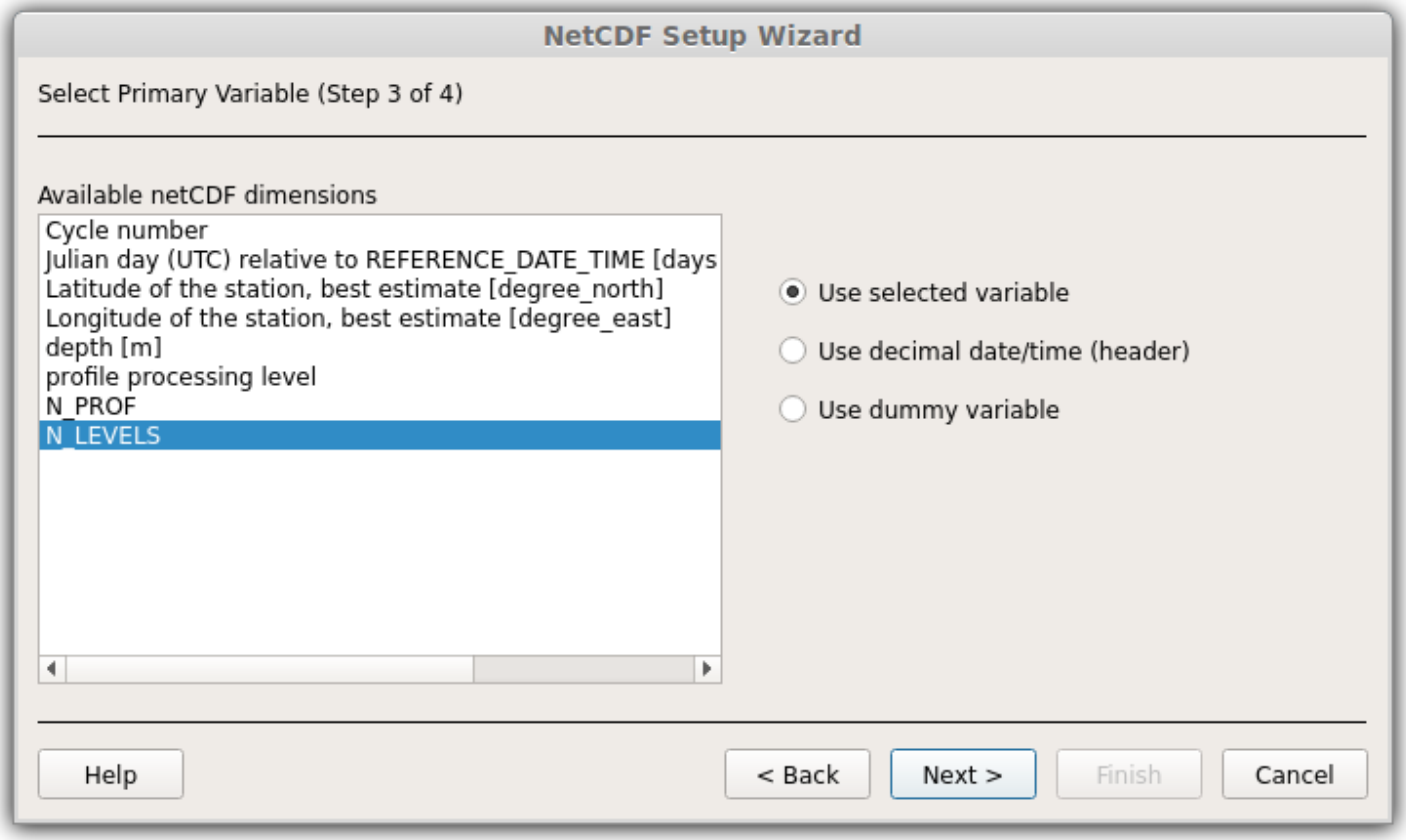
Opening the netCDF file

Variable association : Next



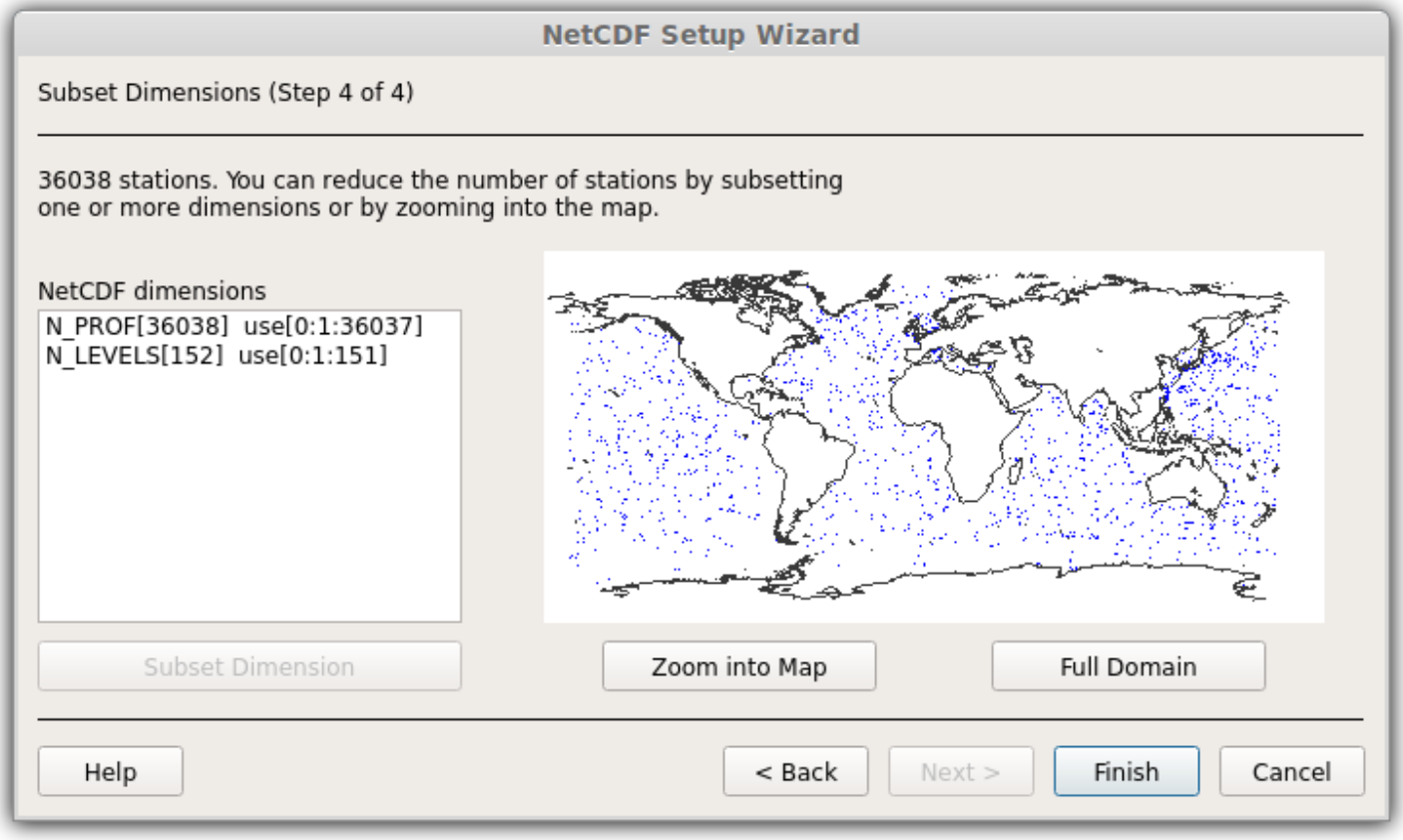
Opening the netCDF file

Primary variables : Next



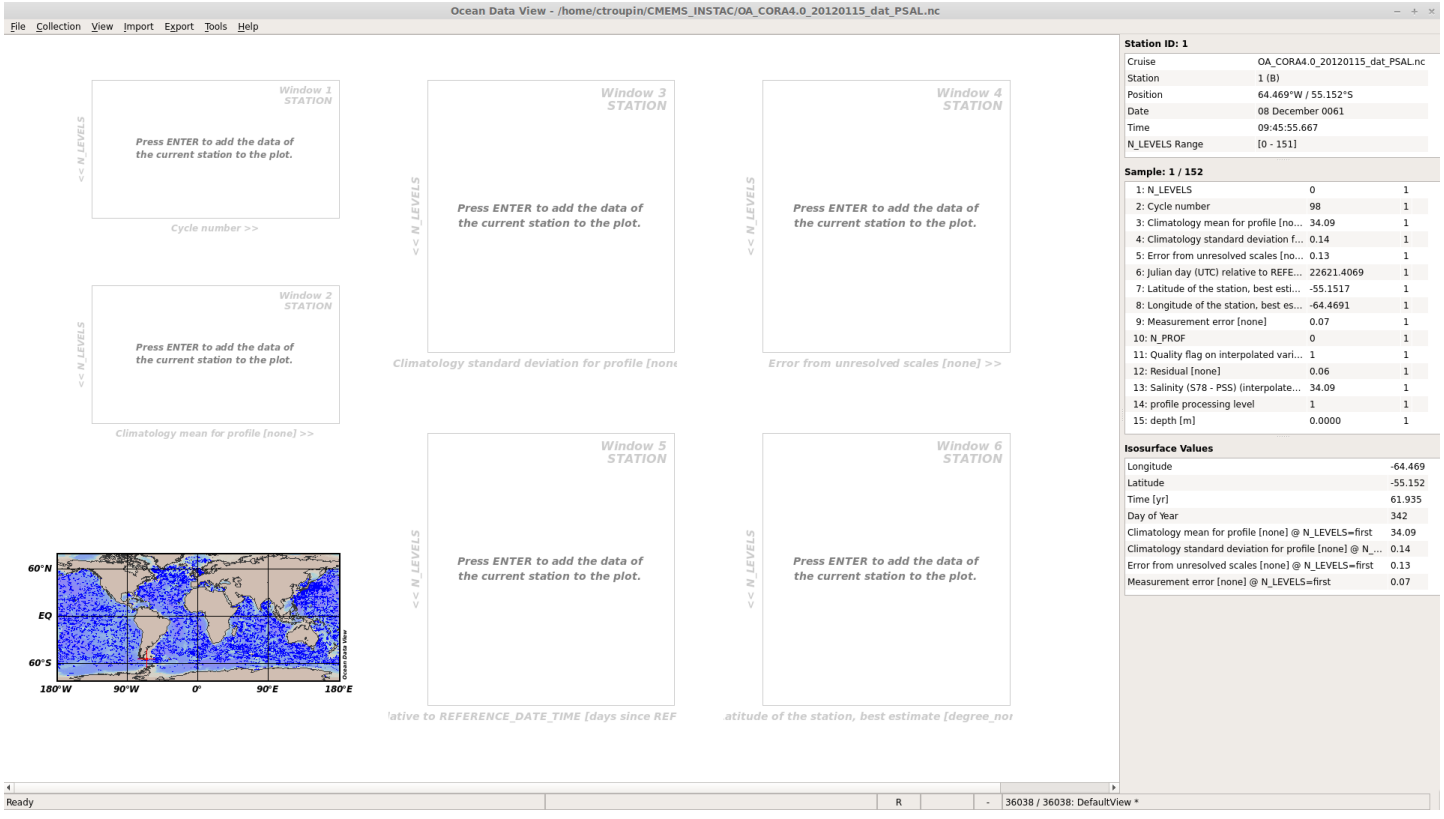
Opening the netCDF file

Subset dimensions : Finish



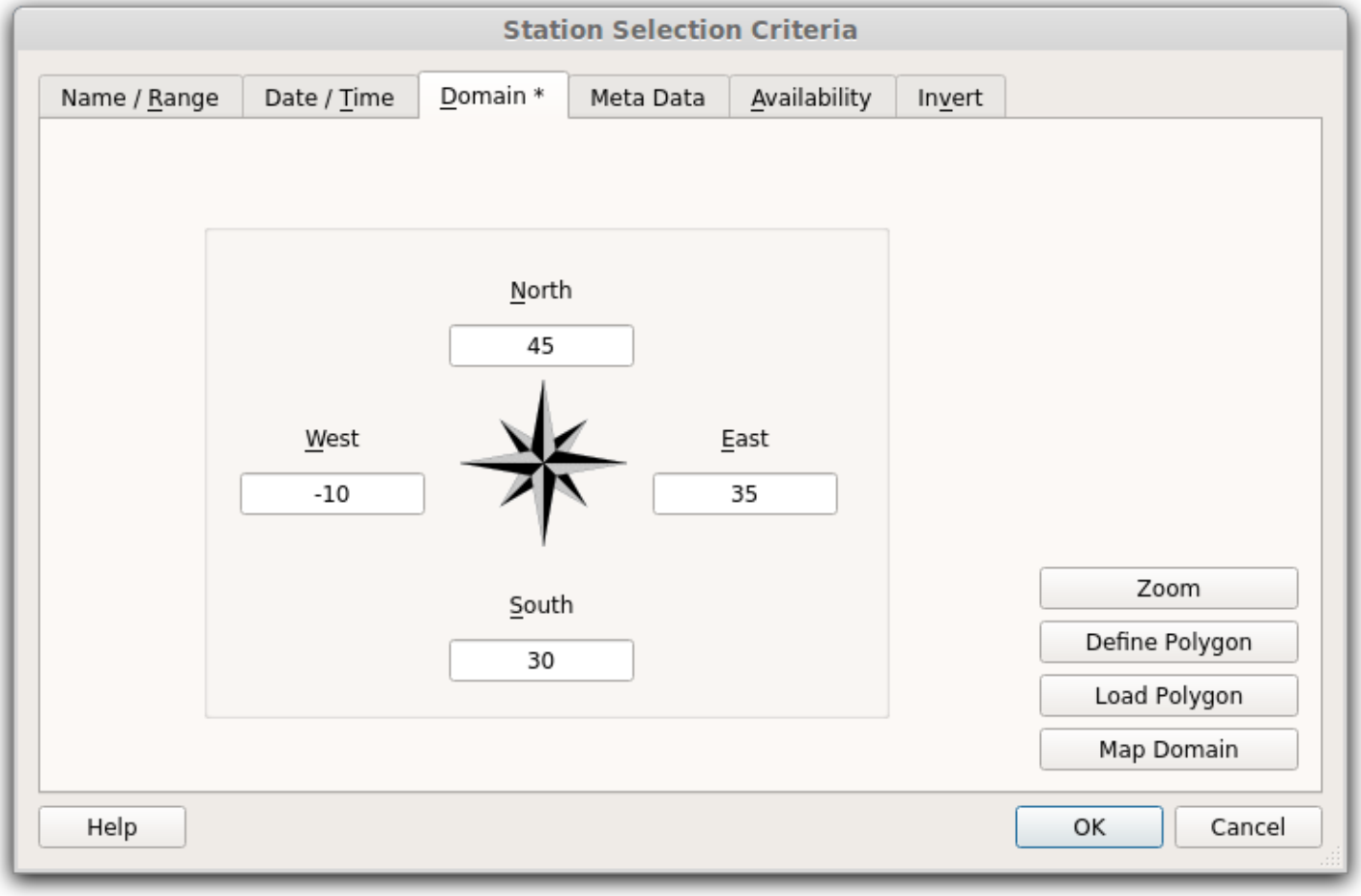
Opening the netCDF file

We get this window



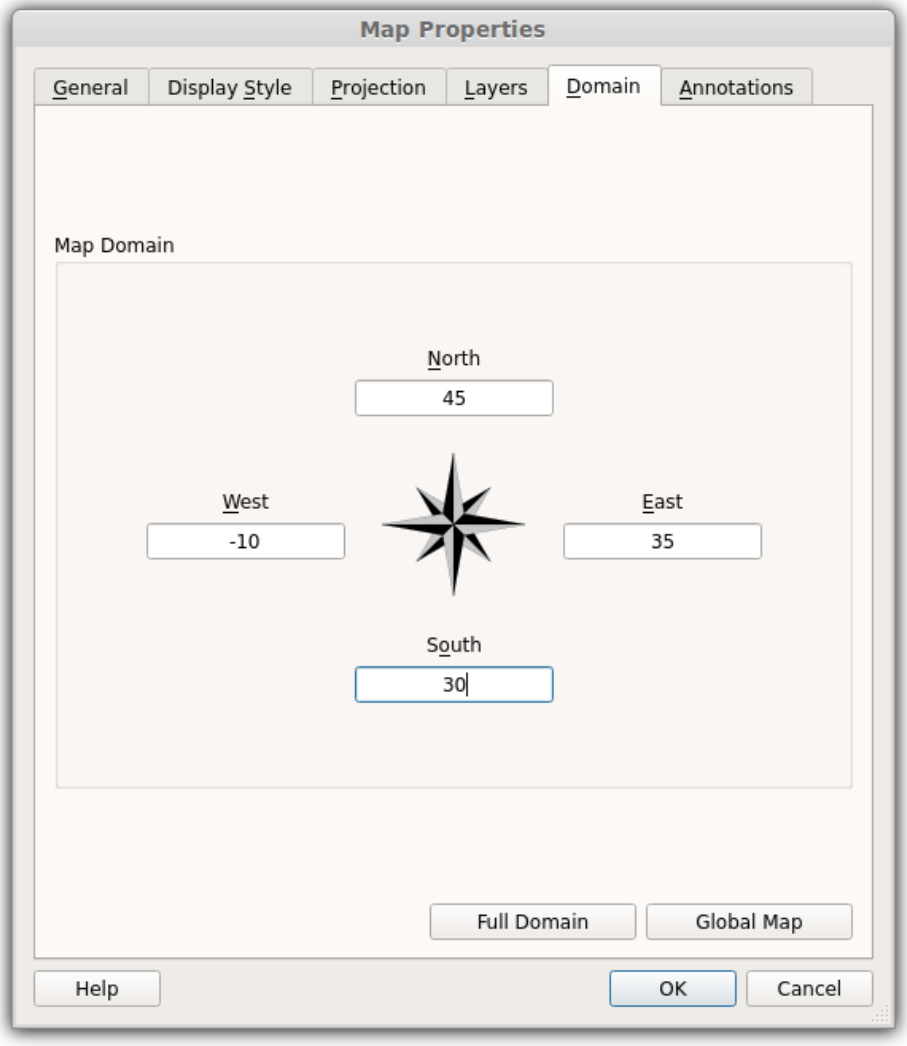
Region selection and basic statistics

- Right-click on image → Station Selection Criteria → Domain



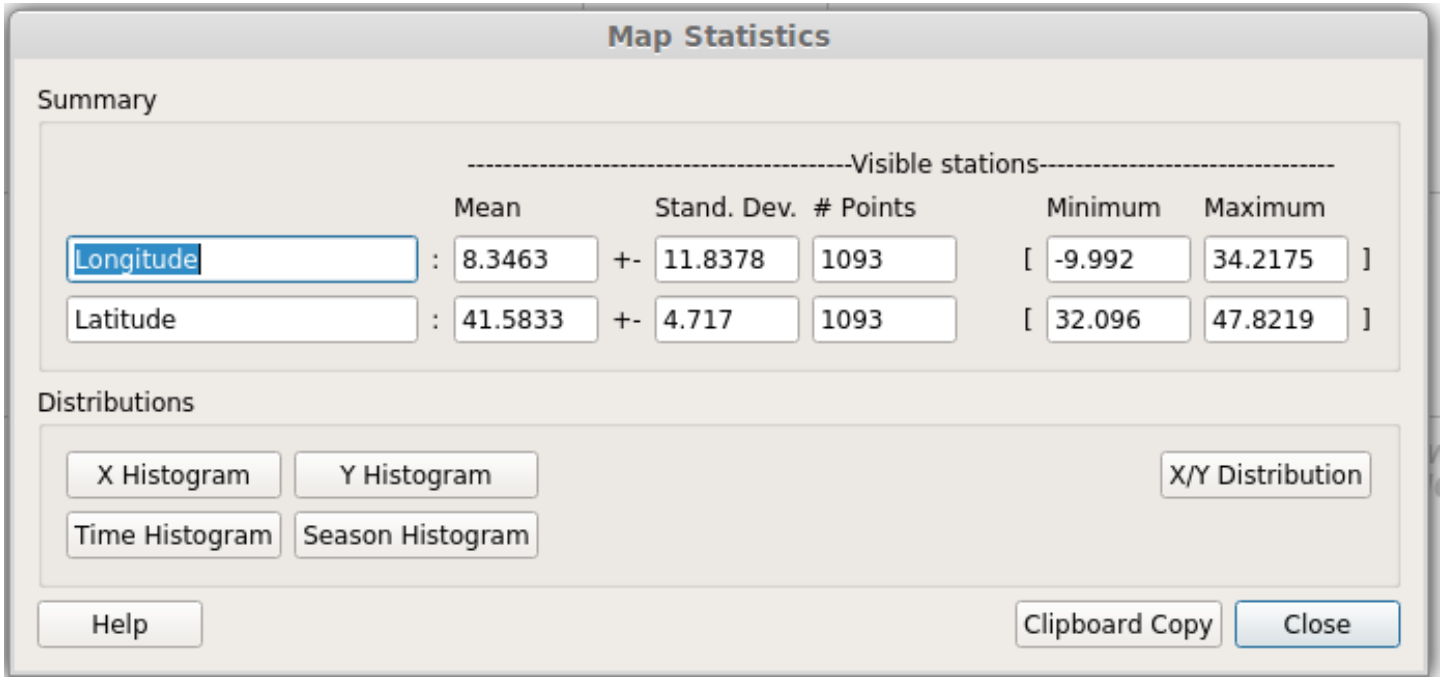
Region selection and basic statistics

Right-click on image → Properties → Domain



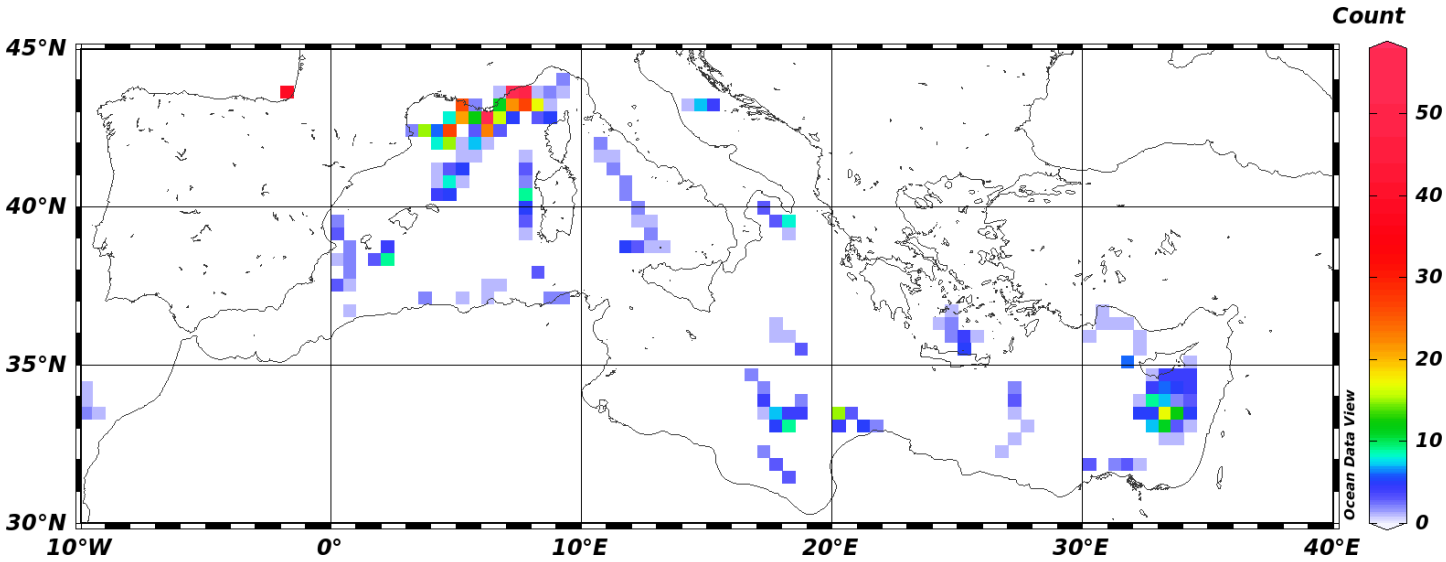
Region selection and basic statistics

Right-click on image → Extra → Statistics



Region selection and basic statistics

Figure → X/Y Distribution

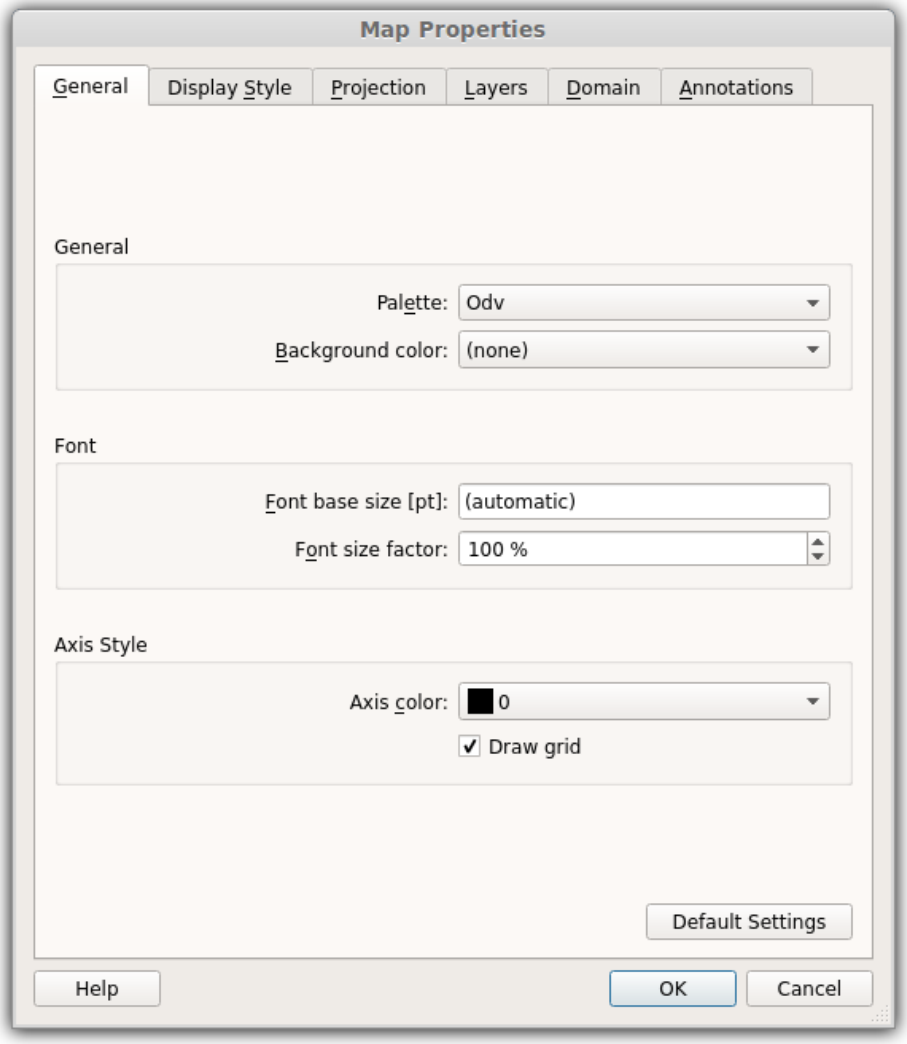


✉ Data scarcity and inhomogeneous distribution

Map improvement

Right-click on map → **Properties**

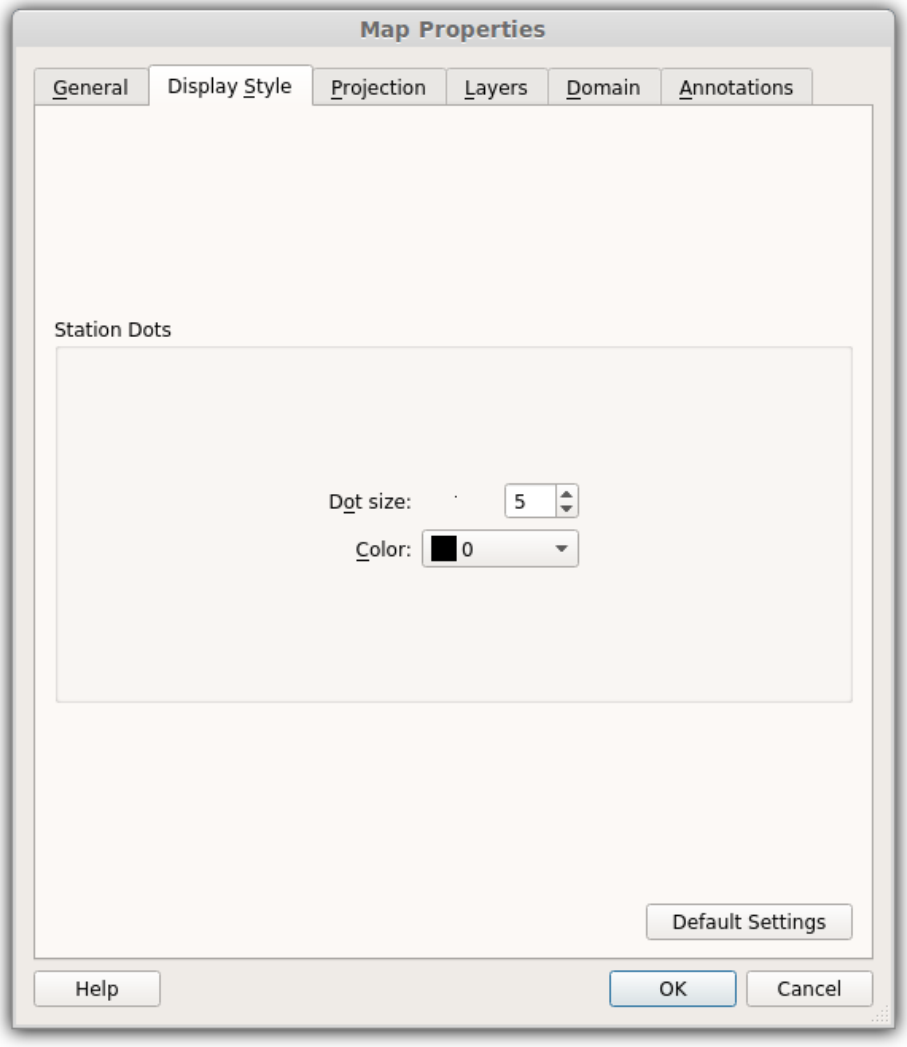
General: palette, colors etc



Map improvement

Right-click on map → **Properties**

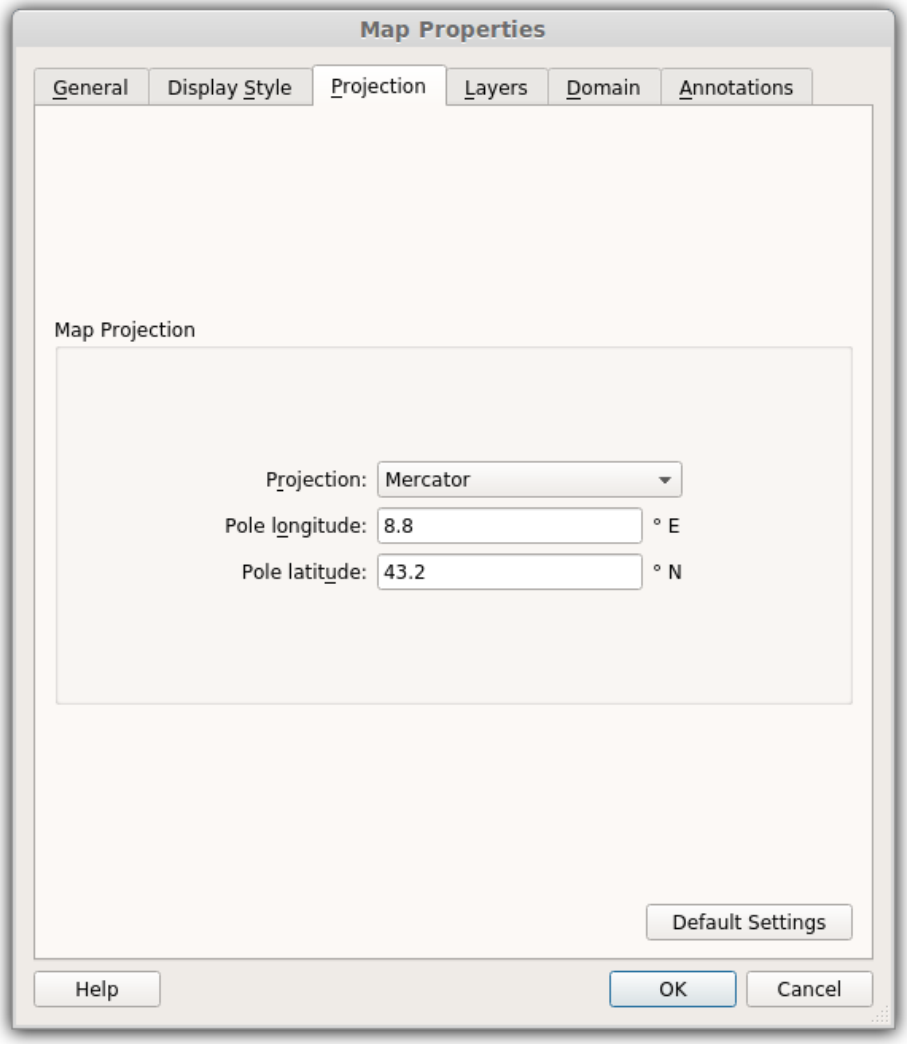
Display style: increase dot size, change color



Map improvement

Right-click on map → **Properties**

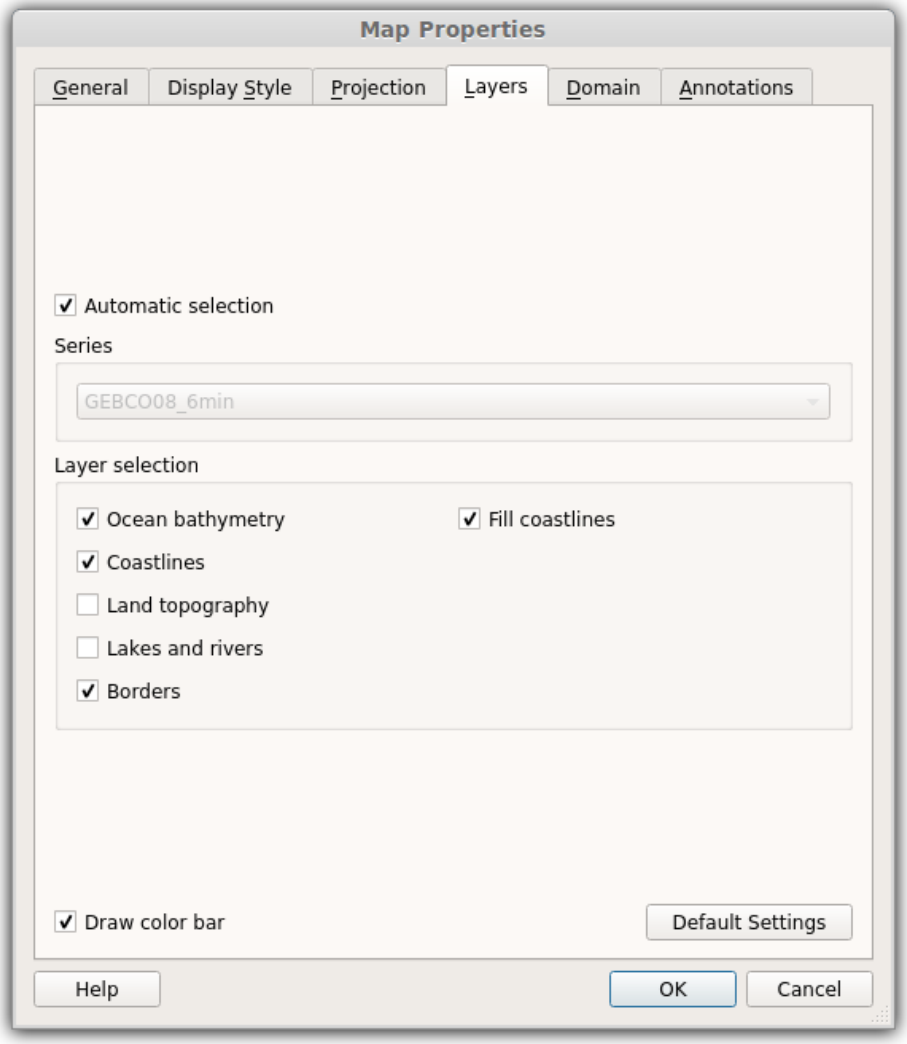
Projection: modify according to preference



Map improvement

Right-click on map → **Properties**

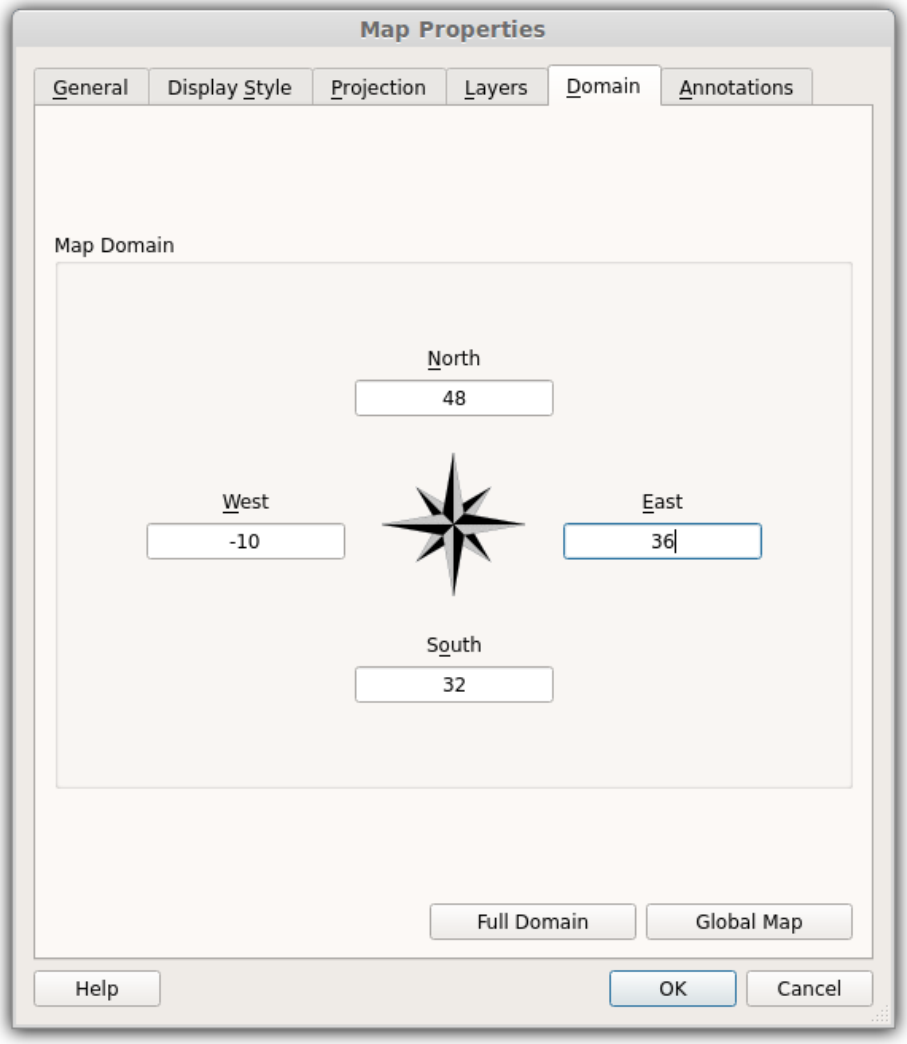
Layers: bathymetry + coastlines



Map improvement

Right-click on map → **Properties**

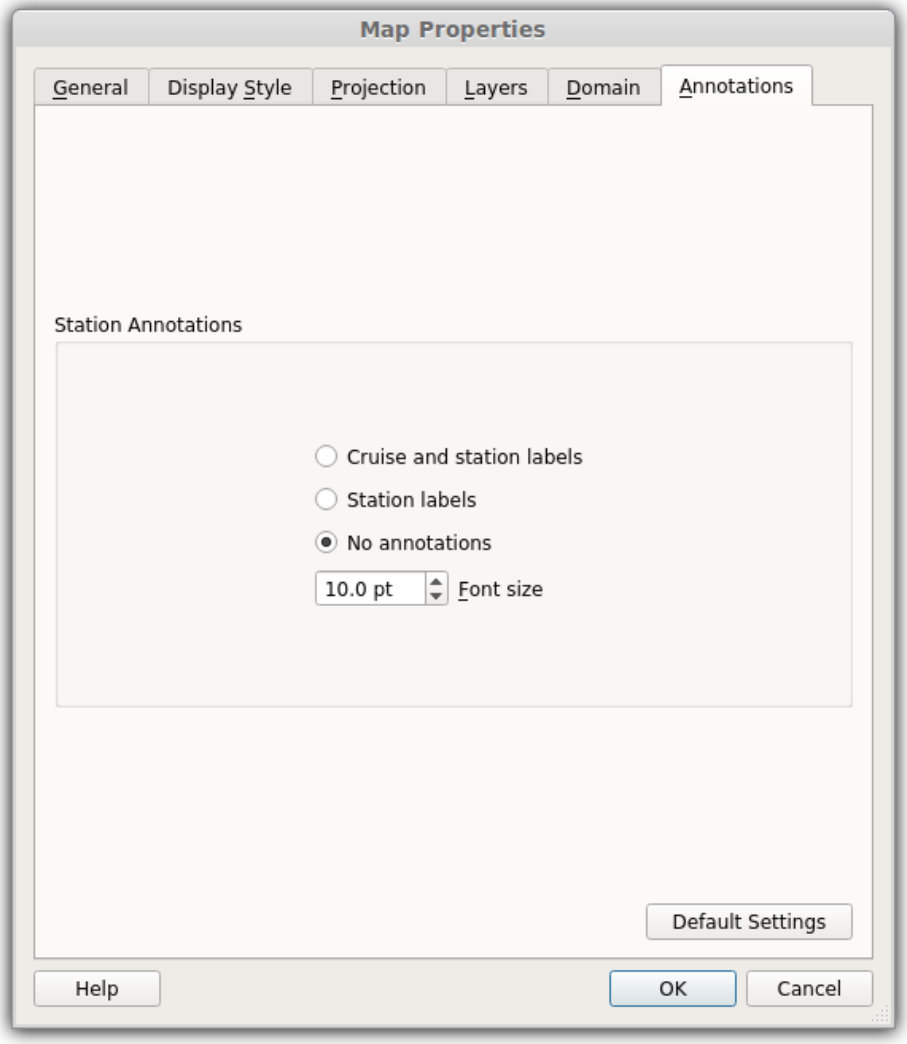
Domain: adjust limits (already done)



Map improvement

Right-click on map → **Properties**

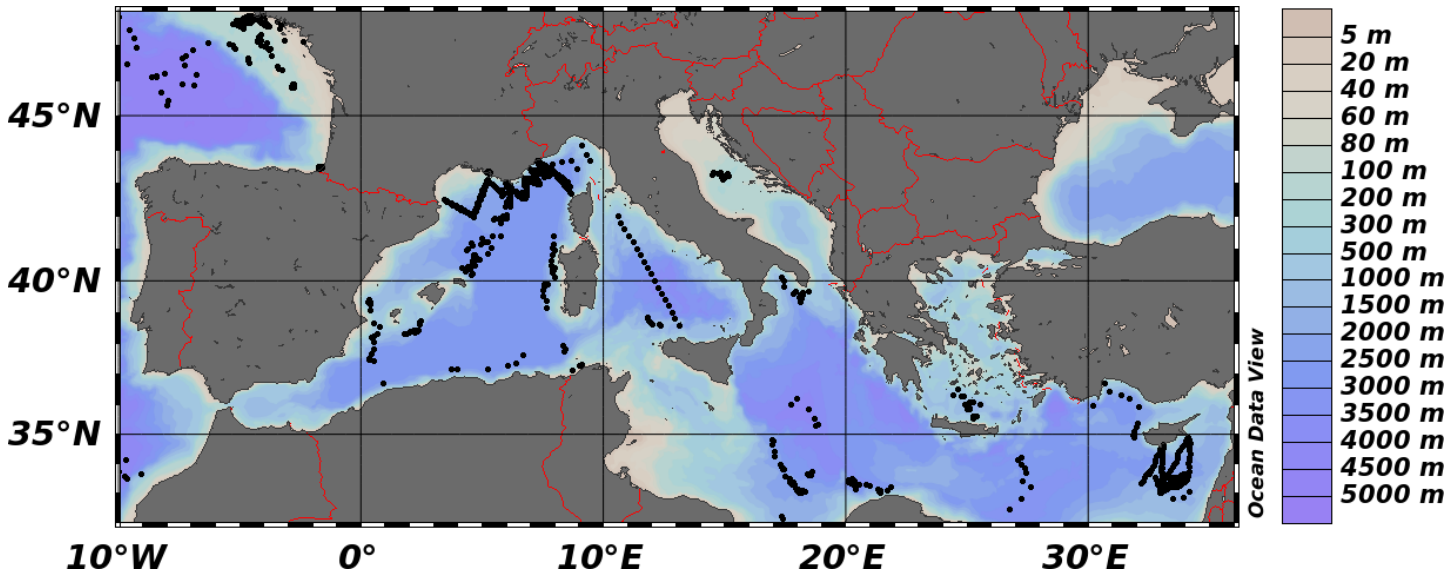
Annotations: not necessary



Map improvement

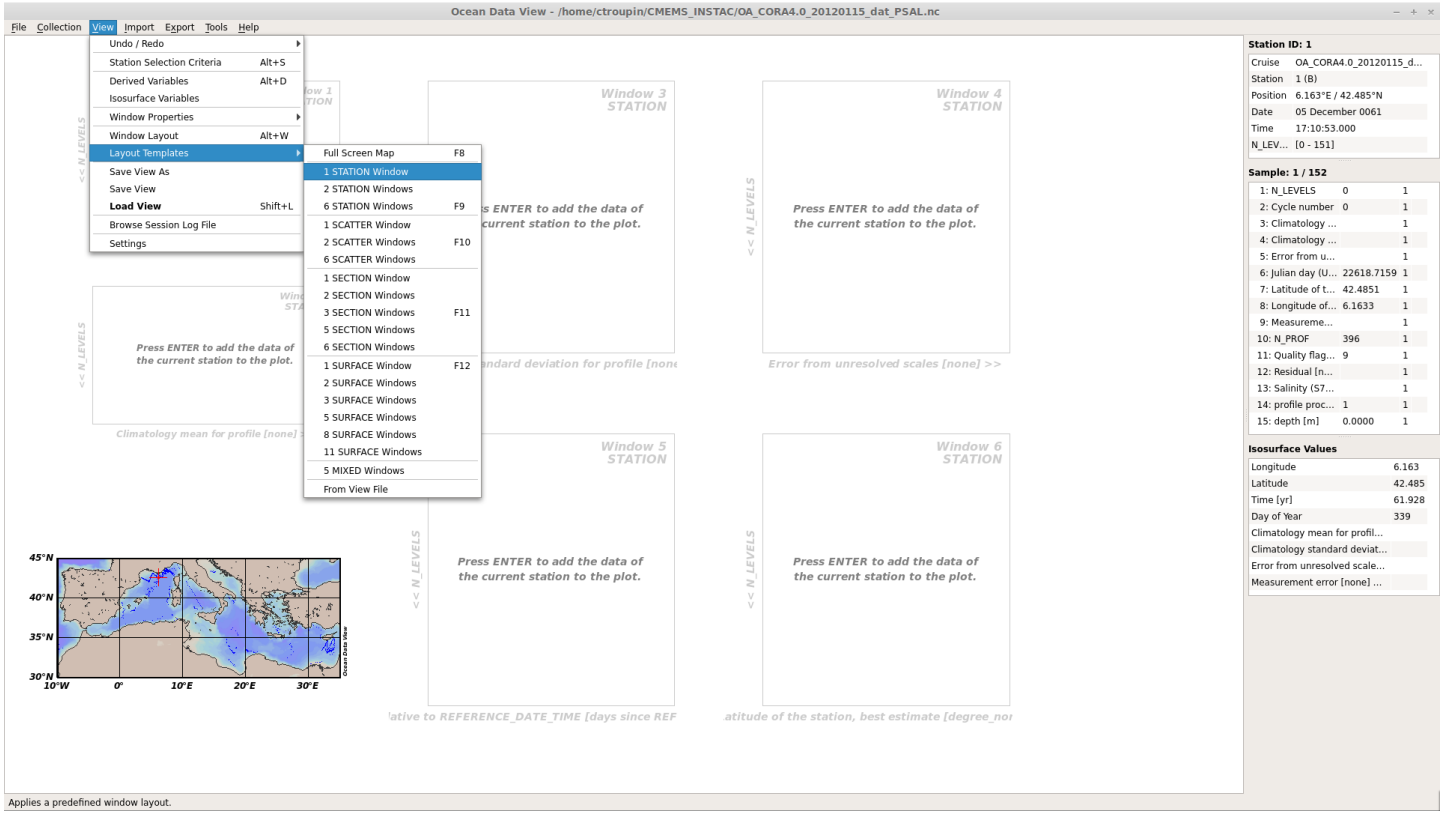
Right-click on map → **Properties**

View → Layout template → Full Screen Map F8



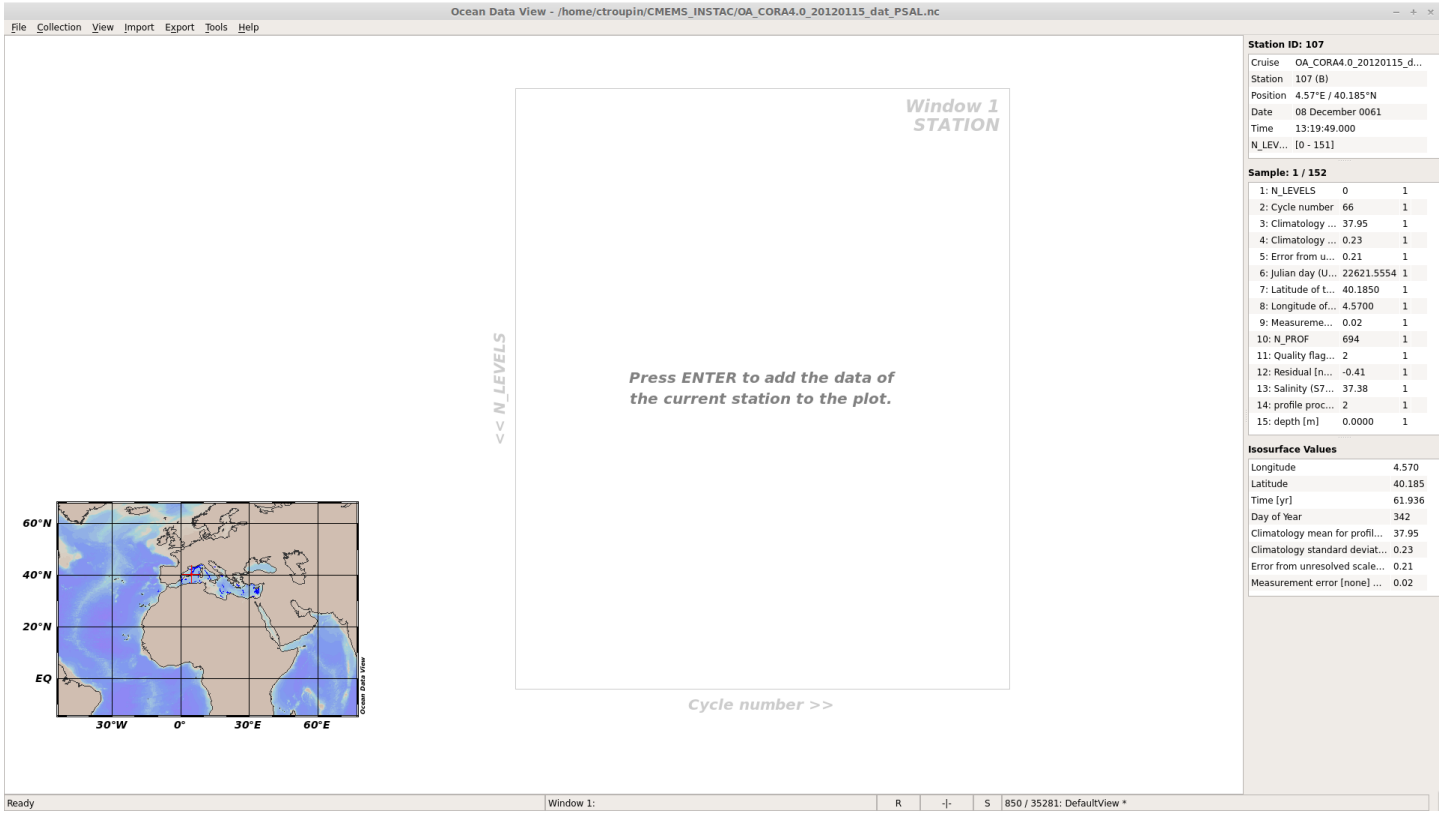
Station window

View → Layout template → Station window

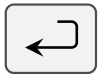


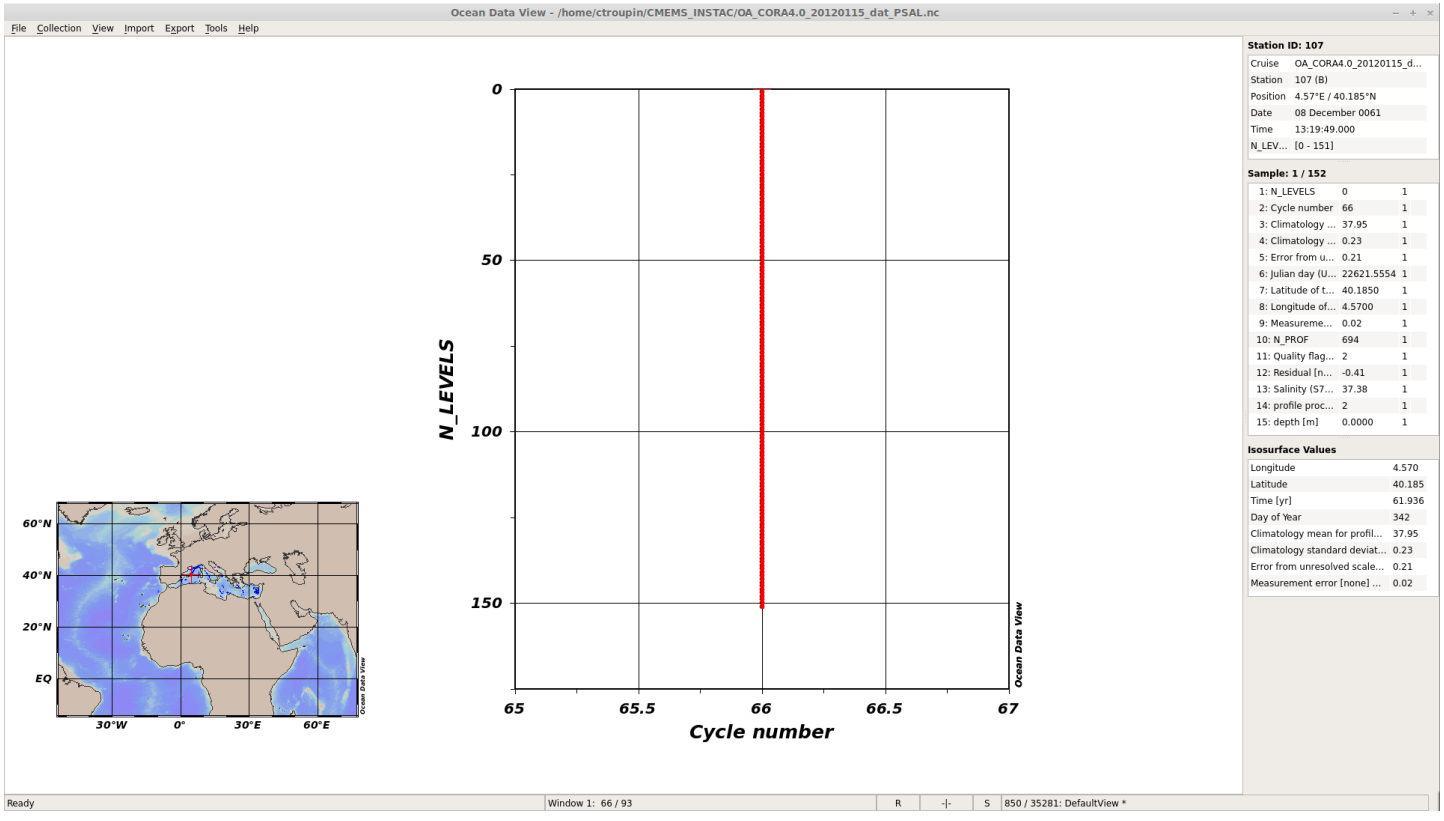
Station window

View → Layout template → Station window



Station window

Enter 

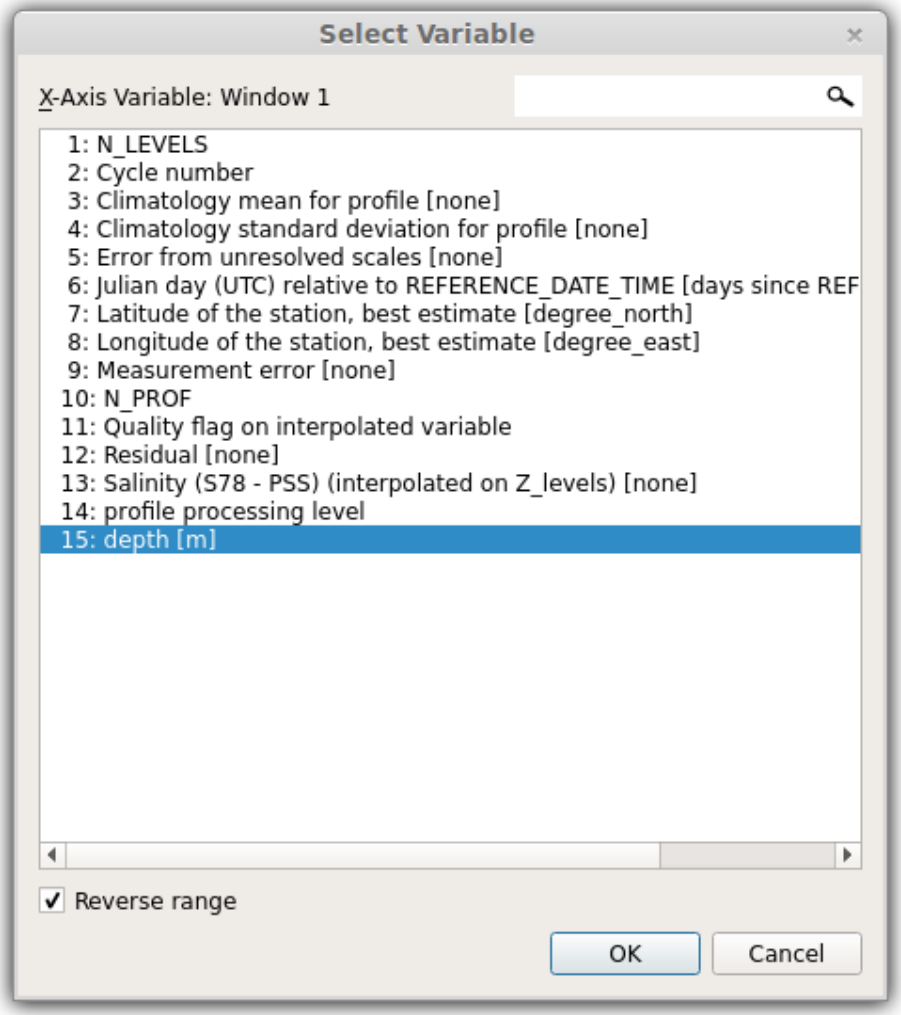


Station window

Right-click on map:

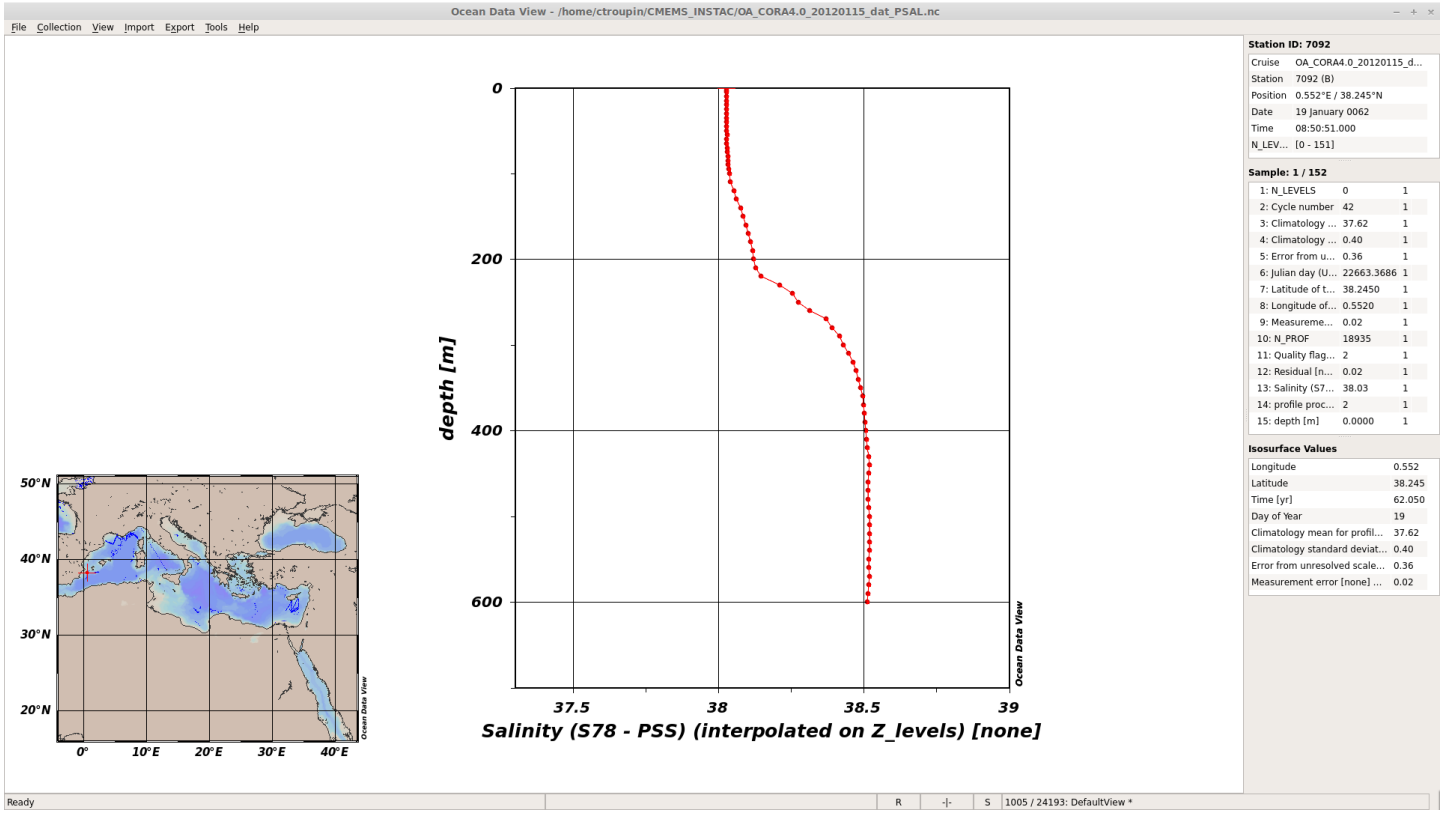
Change X and Y variable (salinity vs. depth)

Y variable → Check the reverse range box



Station window

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

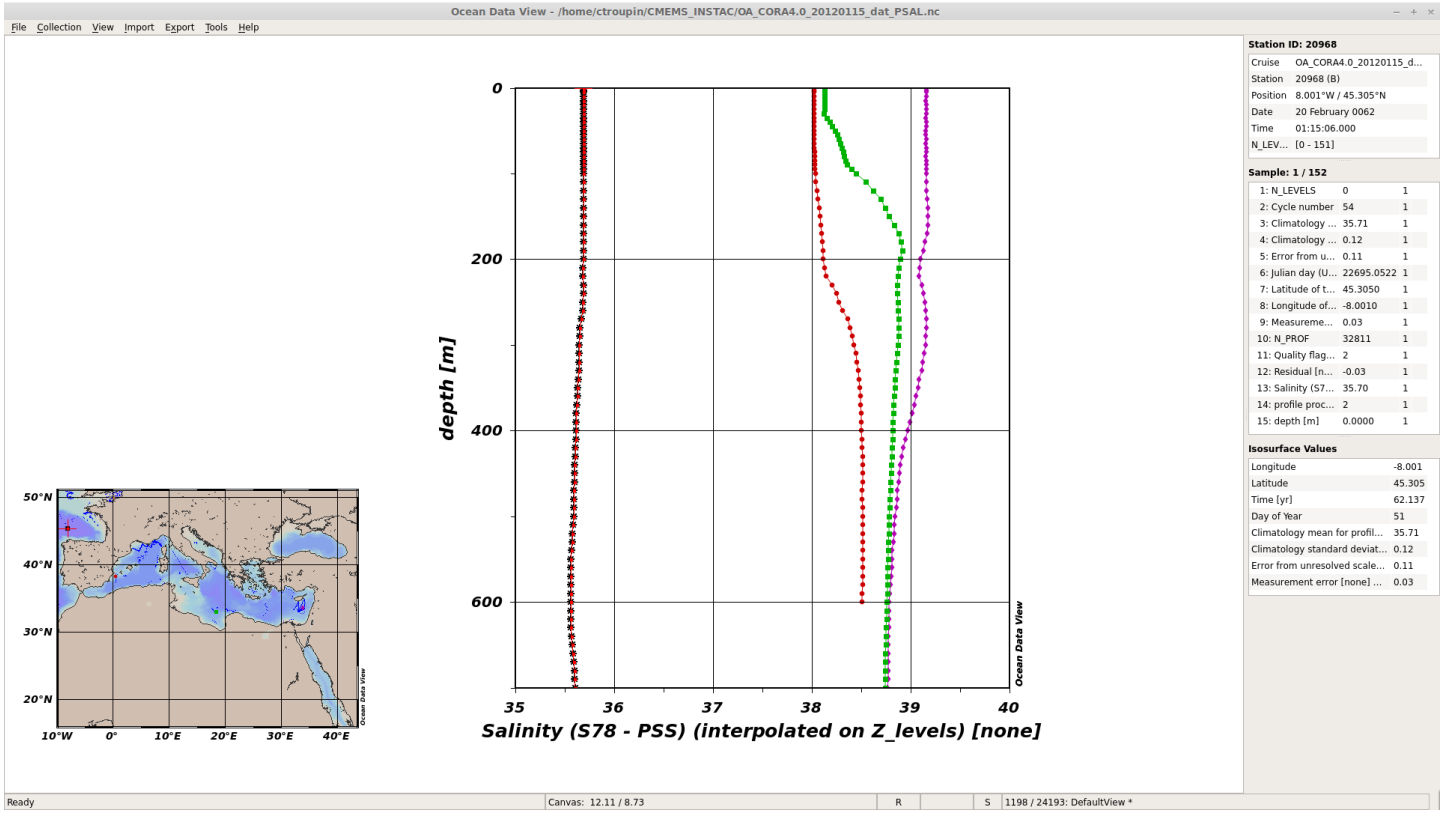


✉ Very different properties according to the basin

✍ Maybe needed to adjust range, otherwise not visible

Station window

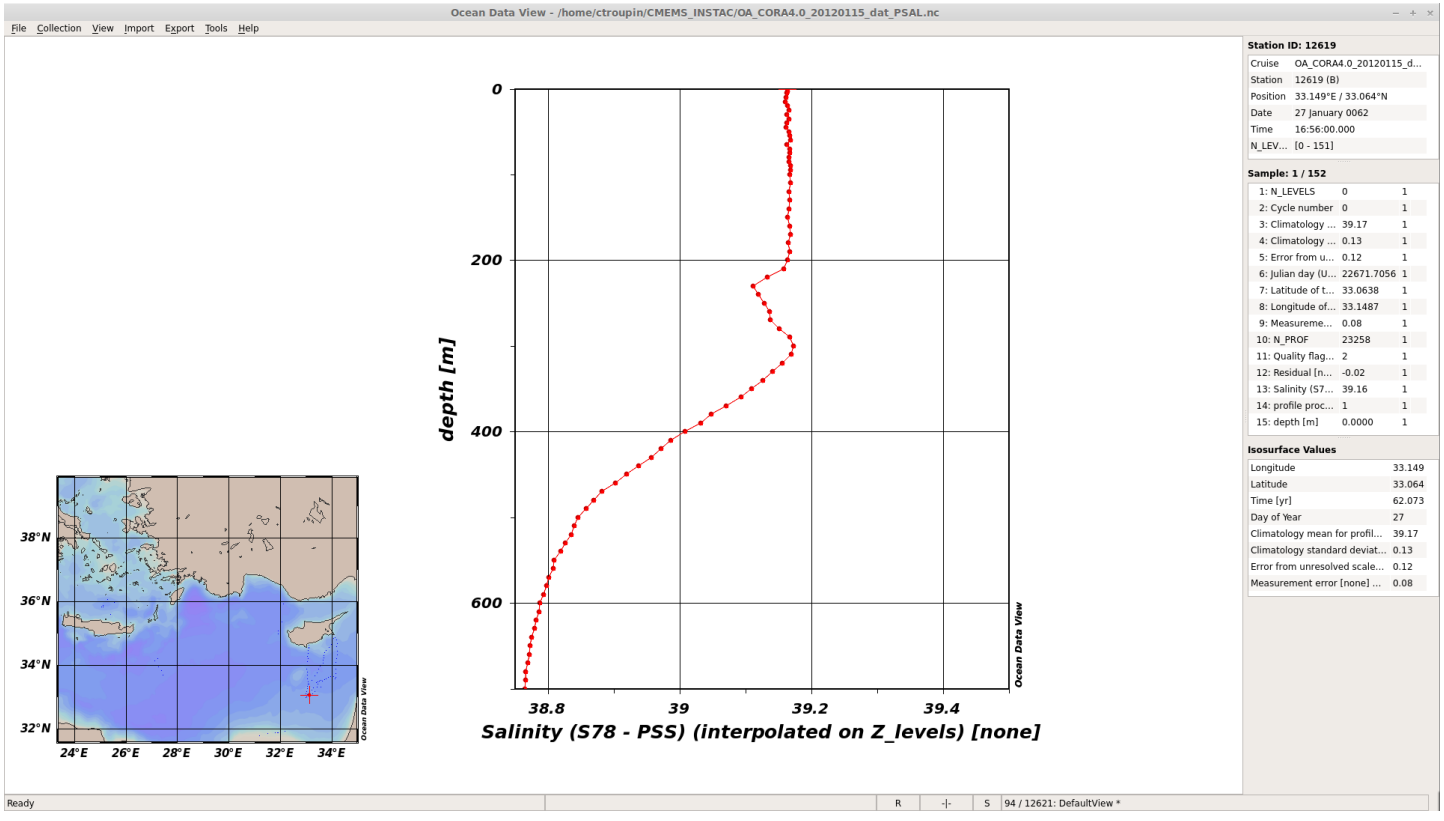
Compare profiles in different sub-regions



To remove stations: Manage Pick List → Remove all Stations

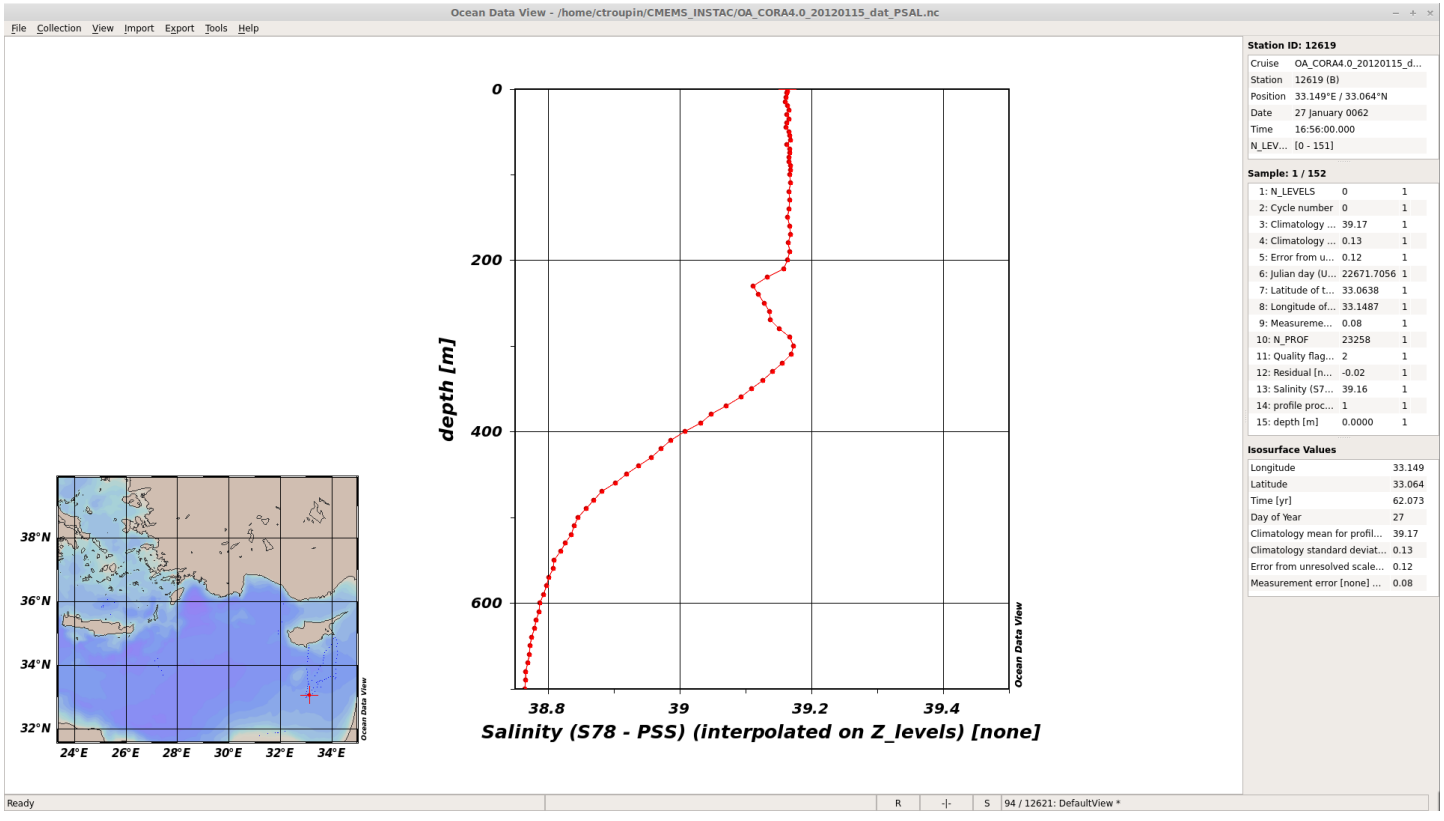
Station window

What happens with this profile south of Cyprus?



Station window

What happens with this profile south of Cyprus?



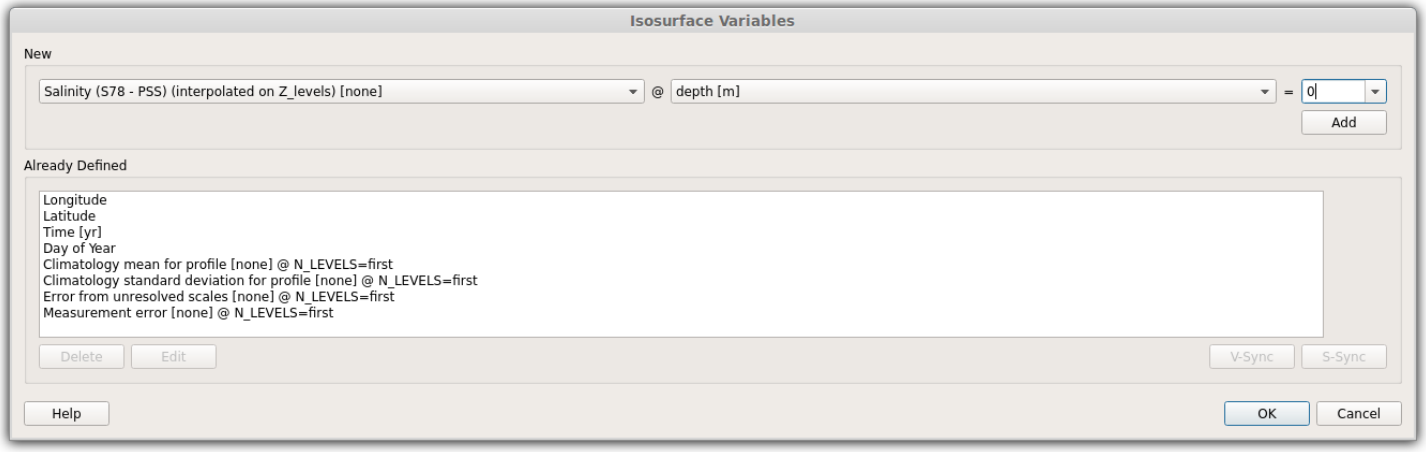
✉ Mixed-layer depth

Surface window

Define new isosurface variables:

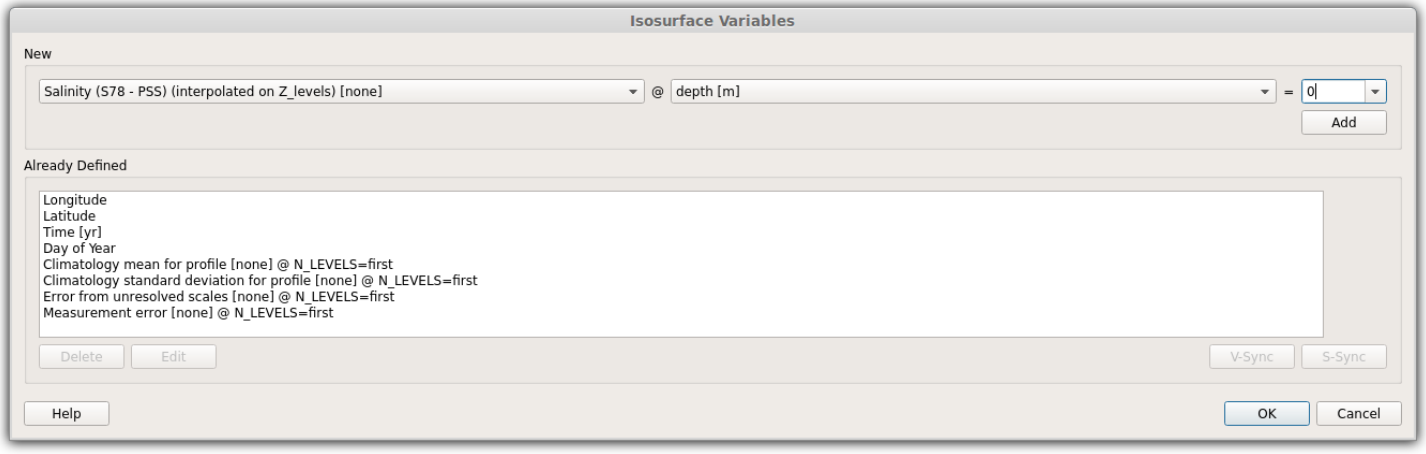
View → Isosurface Variables → salinity at depth = 0

Click on ”Add”



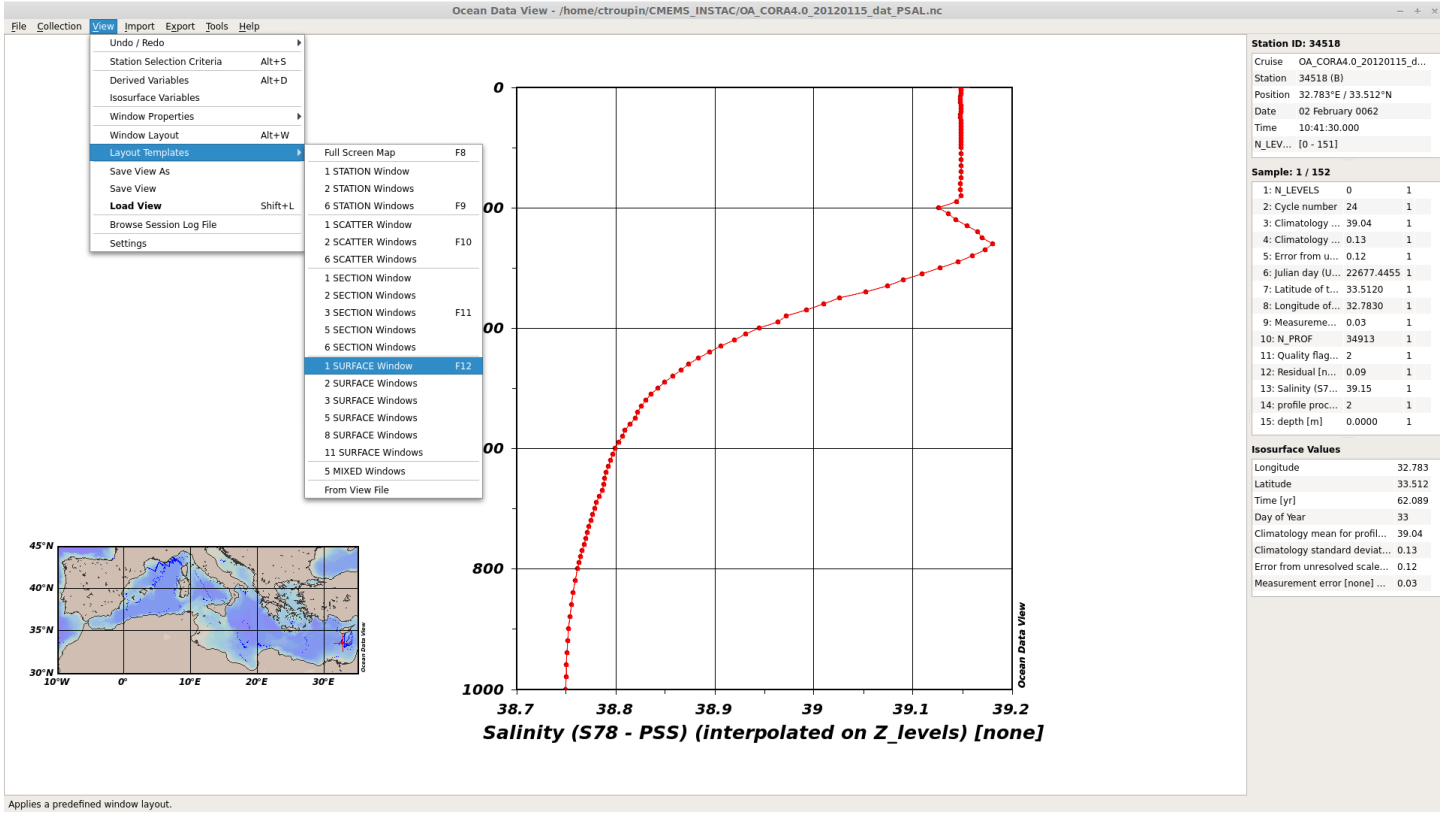
Surface window

Define new isosurface variables:
Same at depth = 200



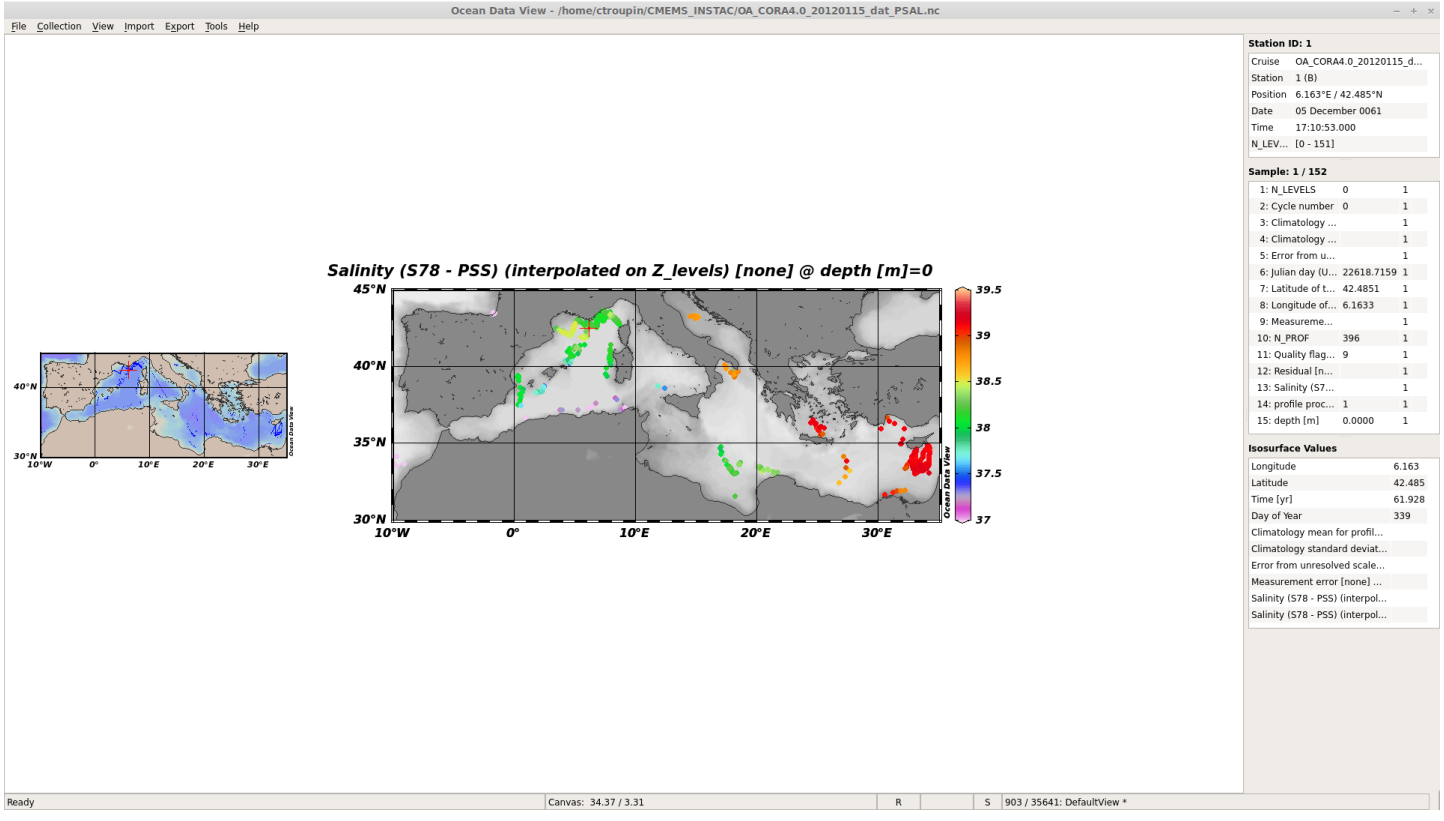
Surface window

View → Layout Template → SURFACE Window



Surface window

Right-click Z-variable → select newly created variable

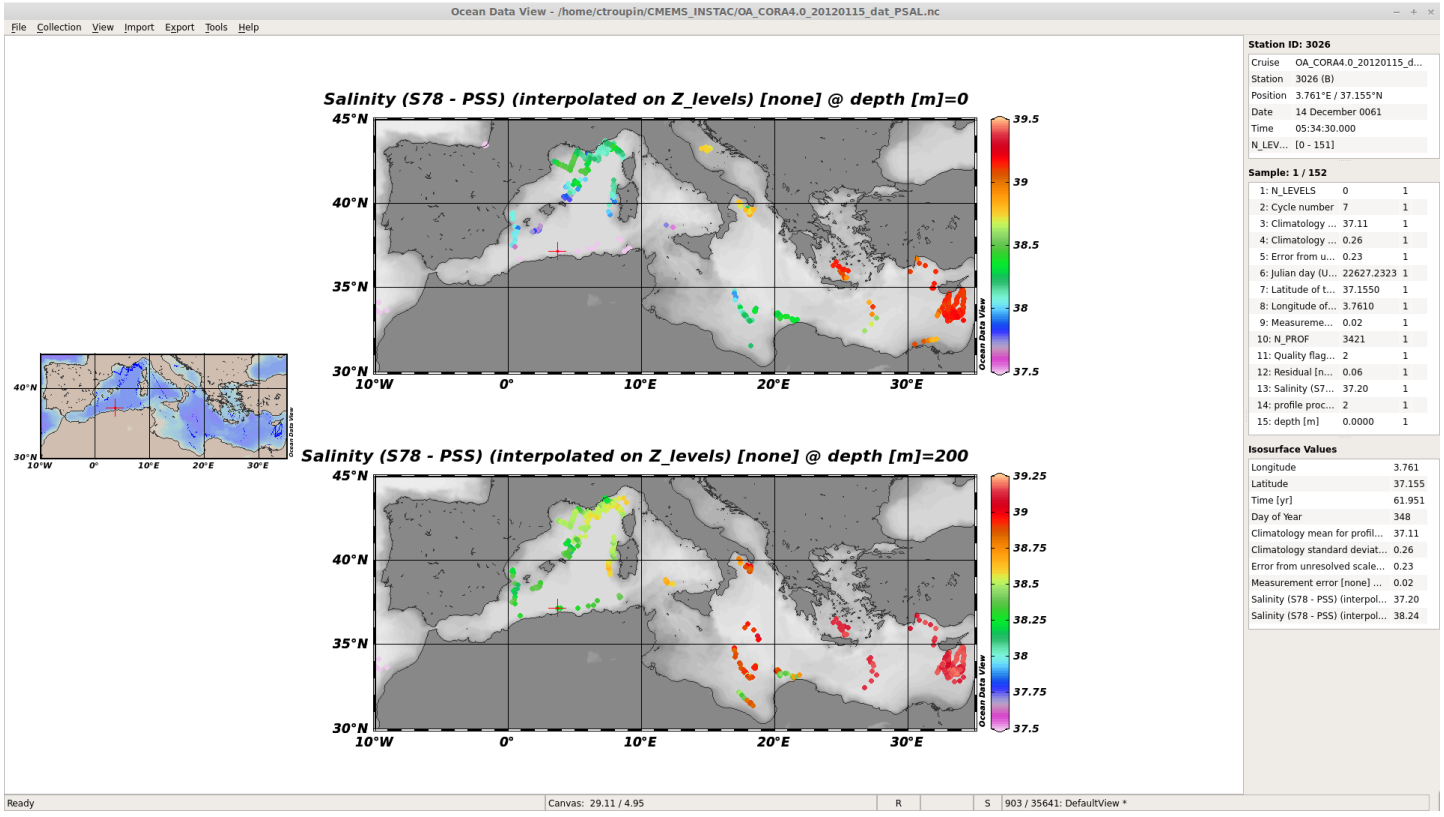


✉ higher salinity values in the Eastern Basin

✍ Adapt the range for the selected variable

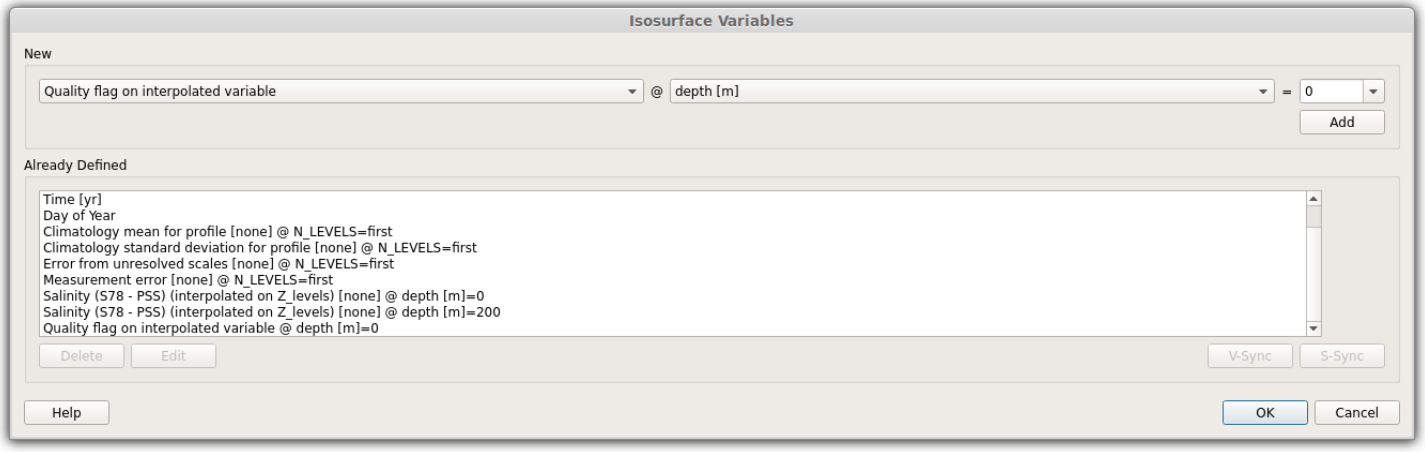
Surface window

Also possible to have several Surface Windows



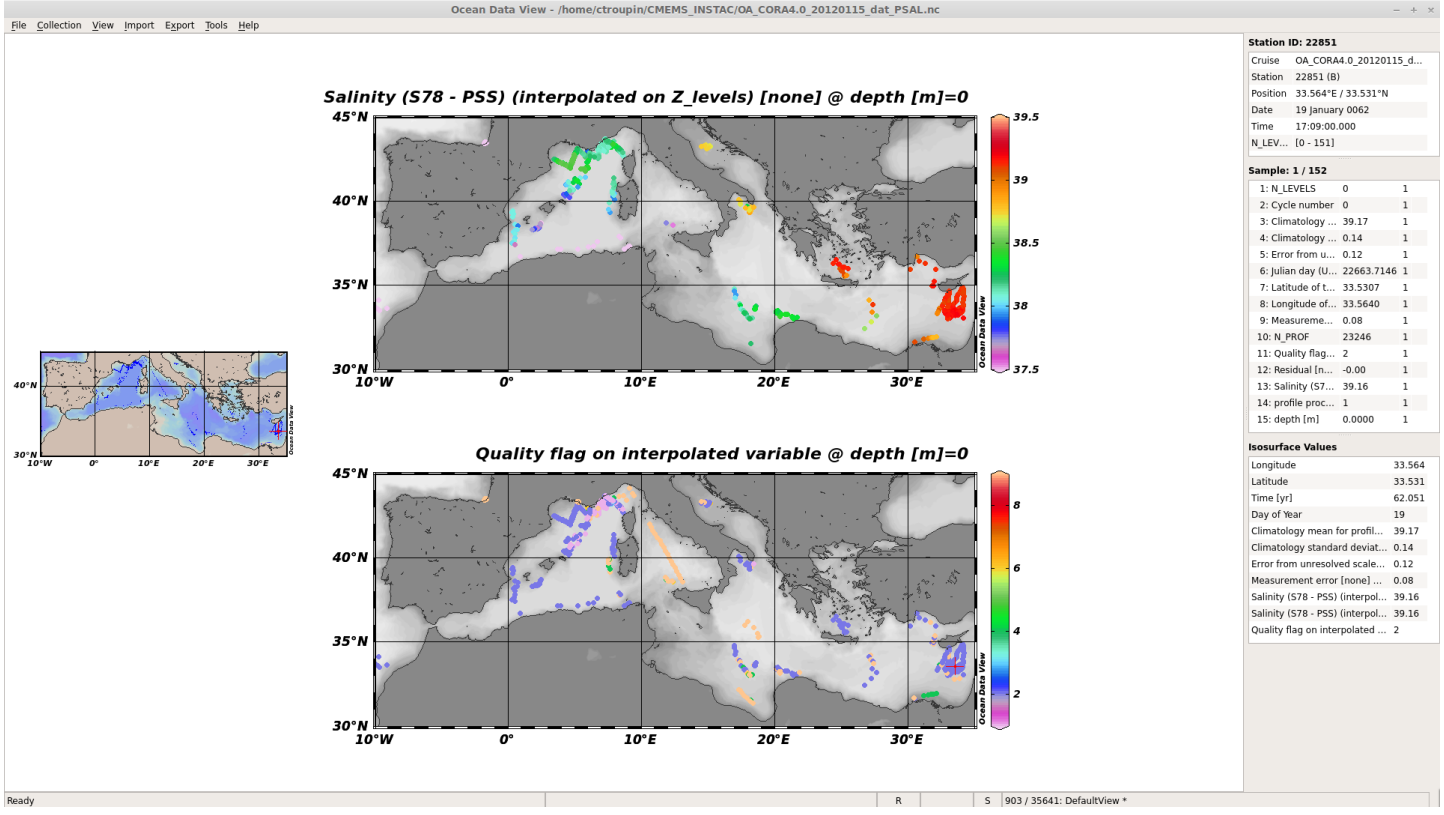
Surface window: quality flag

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



Surface window: quality flag

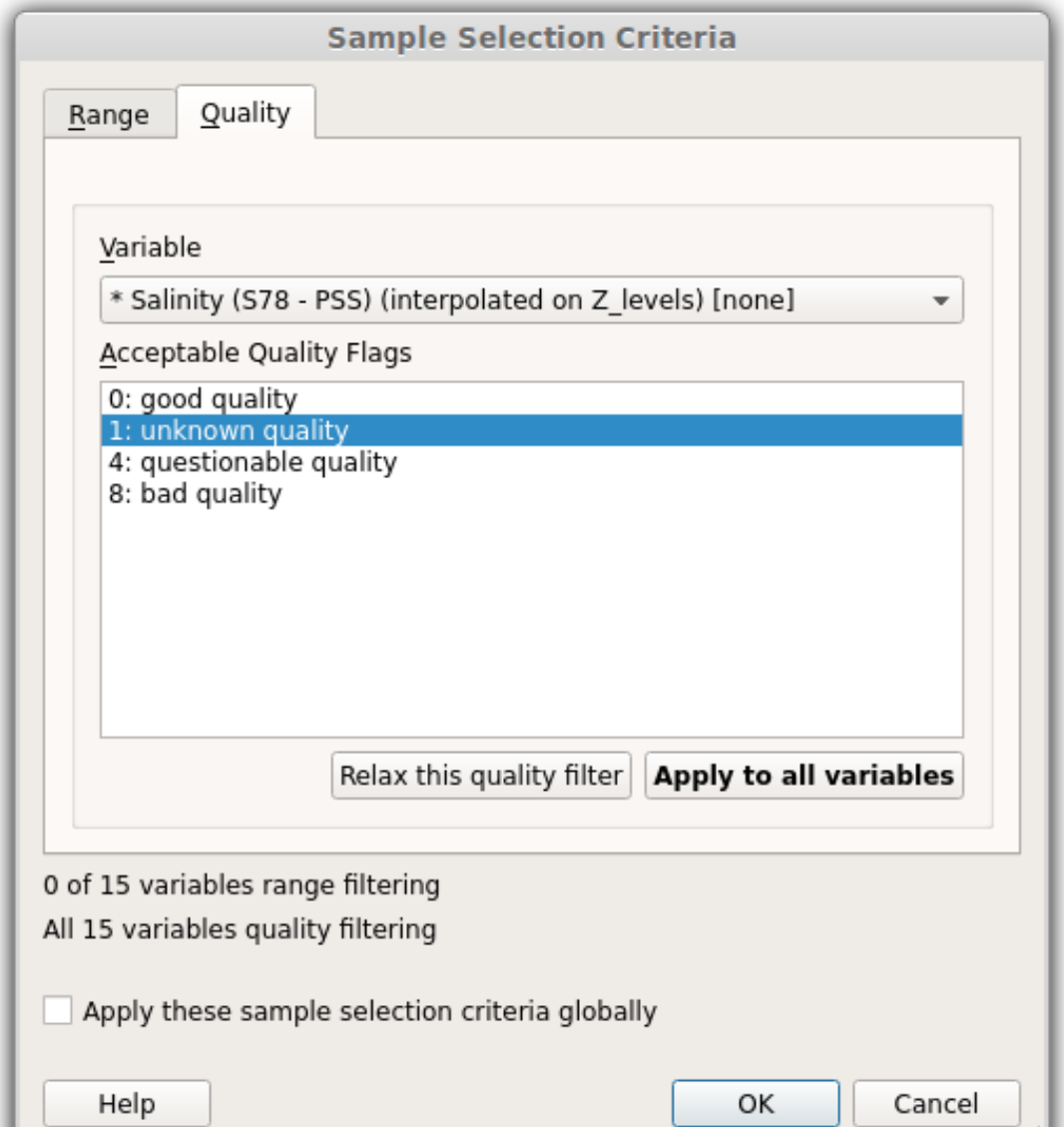
CORA Quality flags:

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!

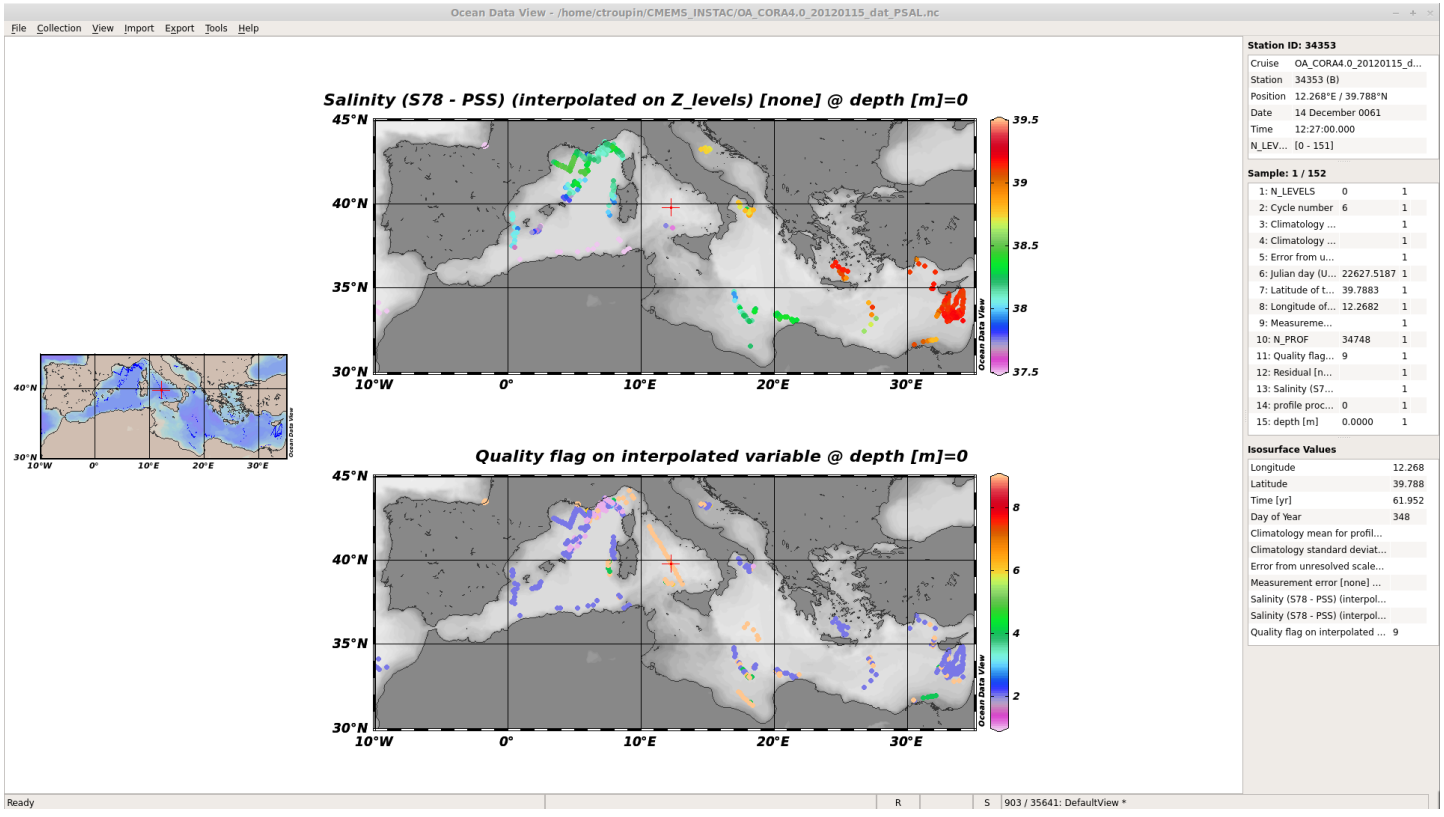
Surface window: quality flag

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



Surface window: quality flag

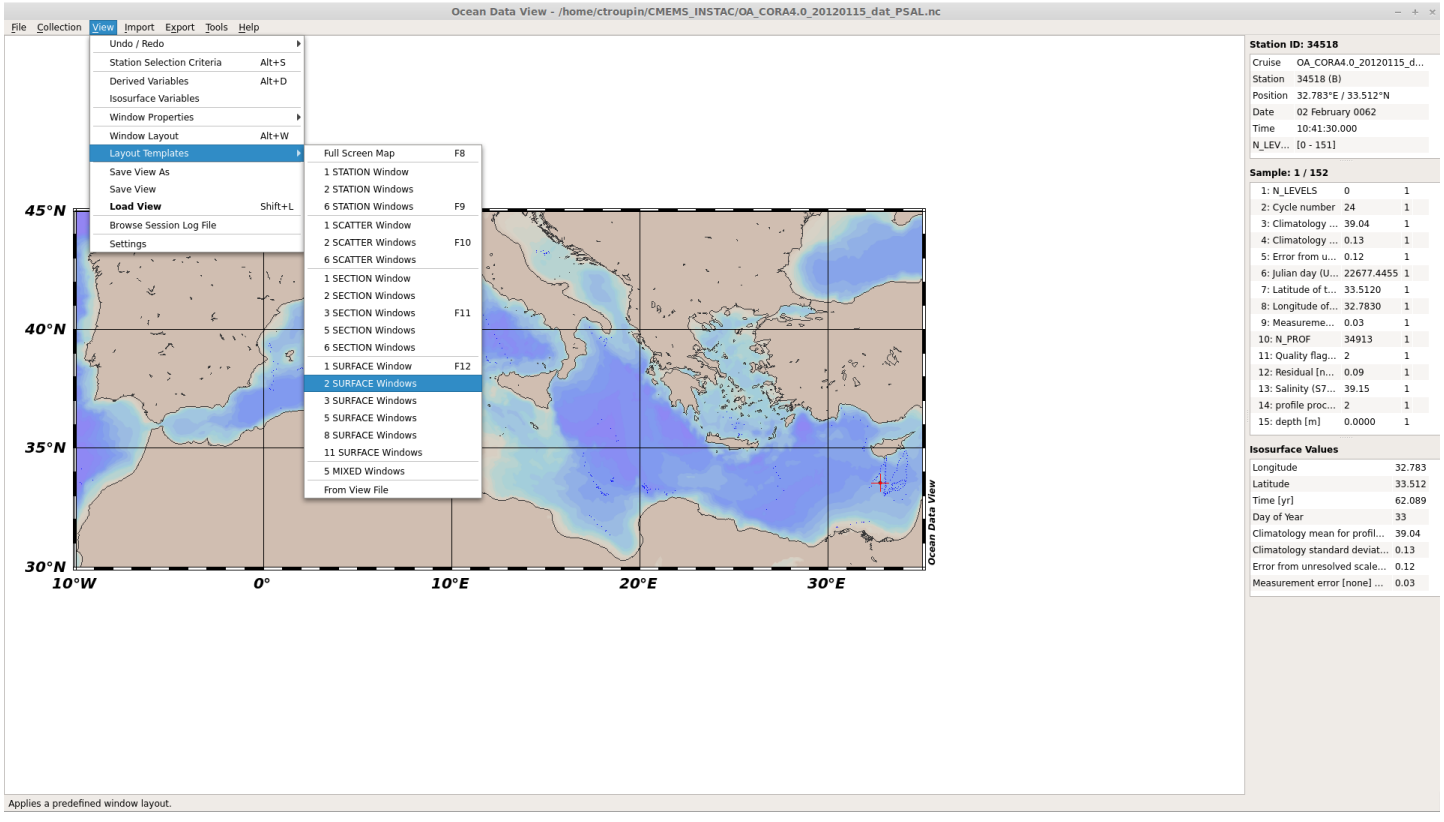
Right-click Sample Selection Criteria → Quality → Accepted quality flags = 1



✉ higher salinity values in the Eastern Basin

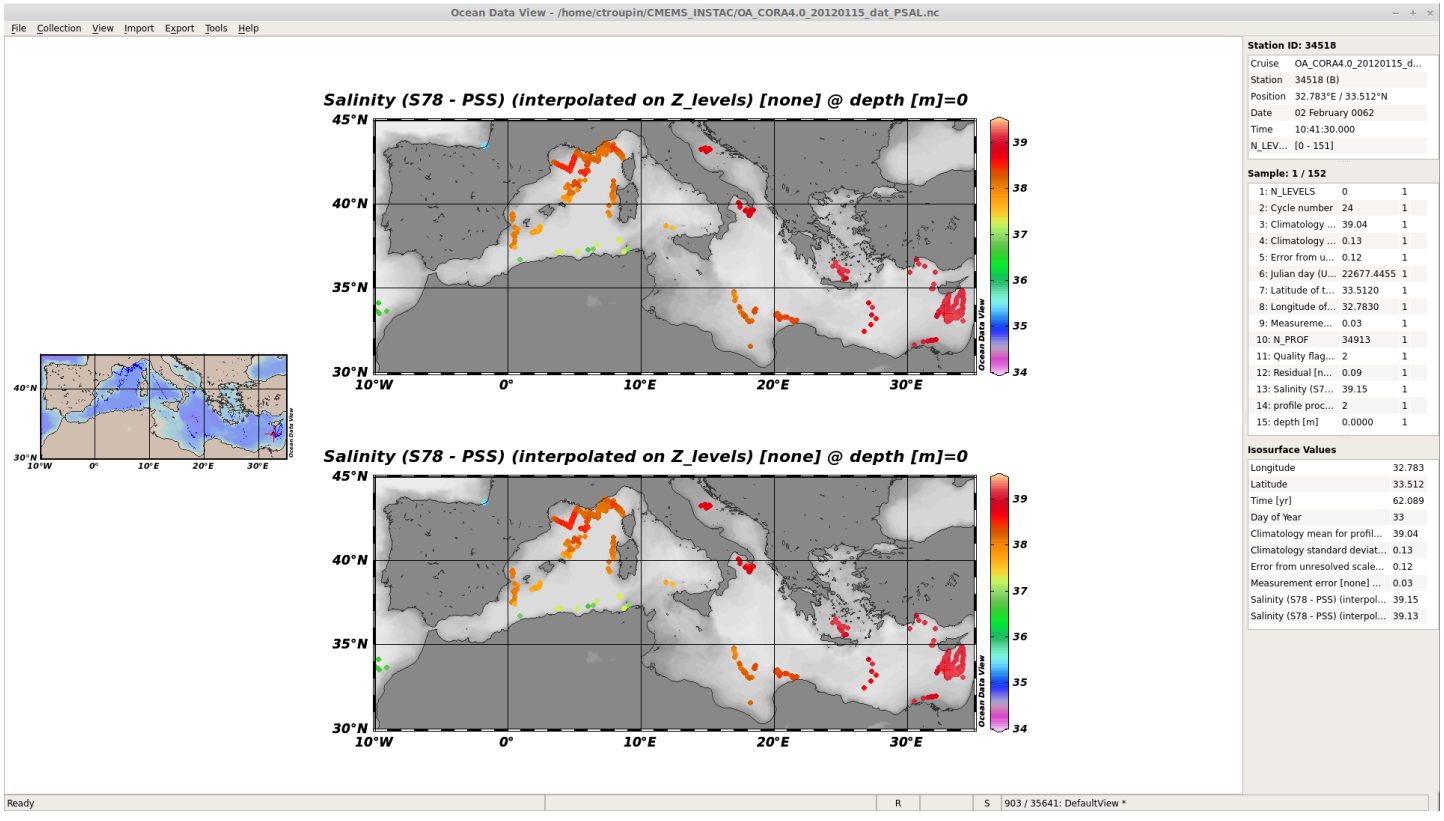
Surface window: gridding

View → Layout Template → SURFACE Window (× 2)



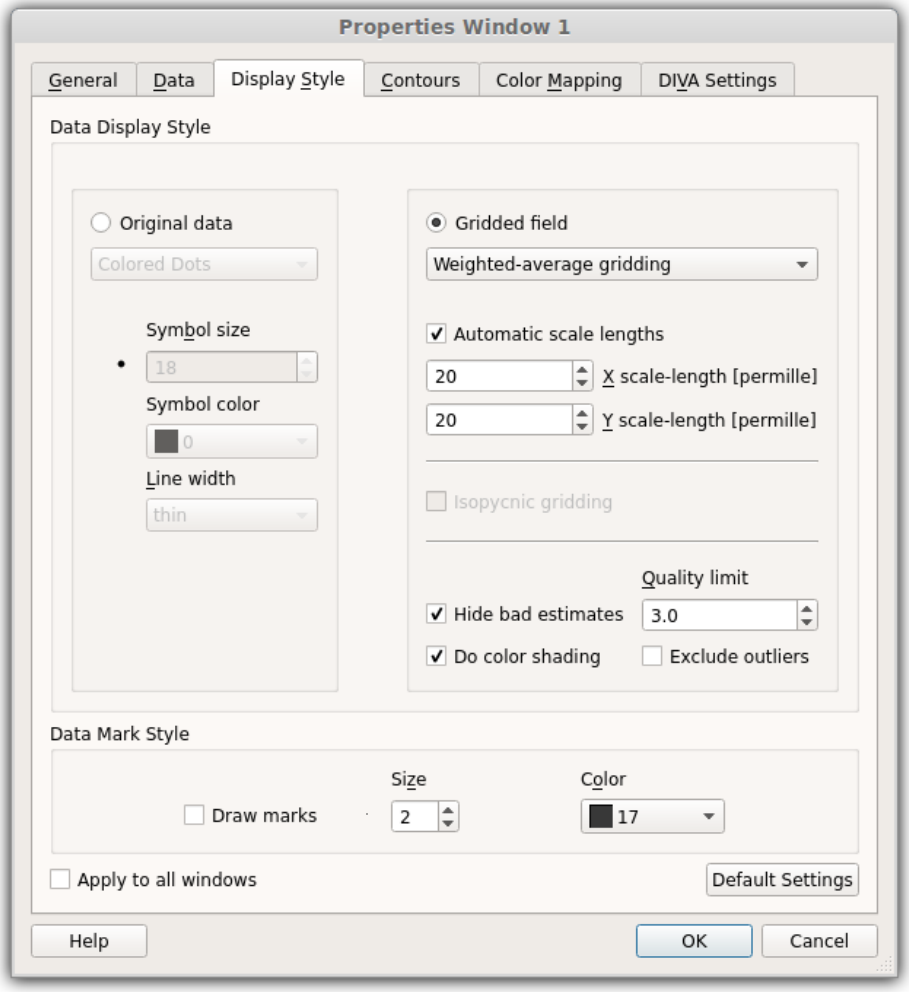
Surface window: gridding

Set Z variable to be Salinity at 0 m



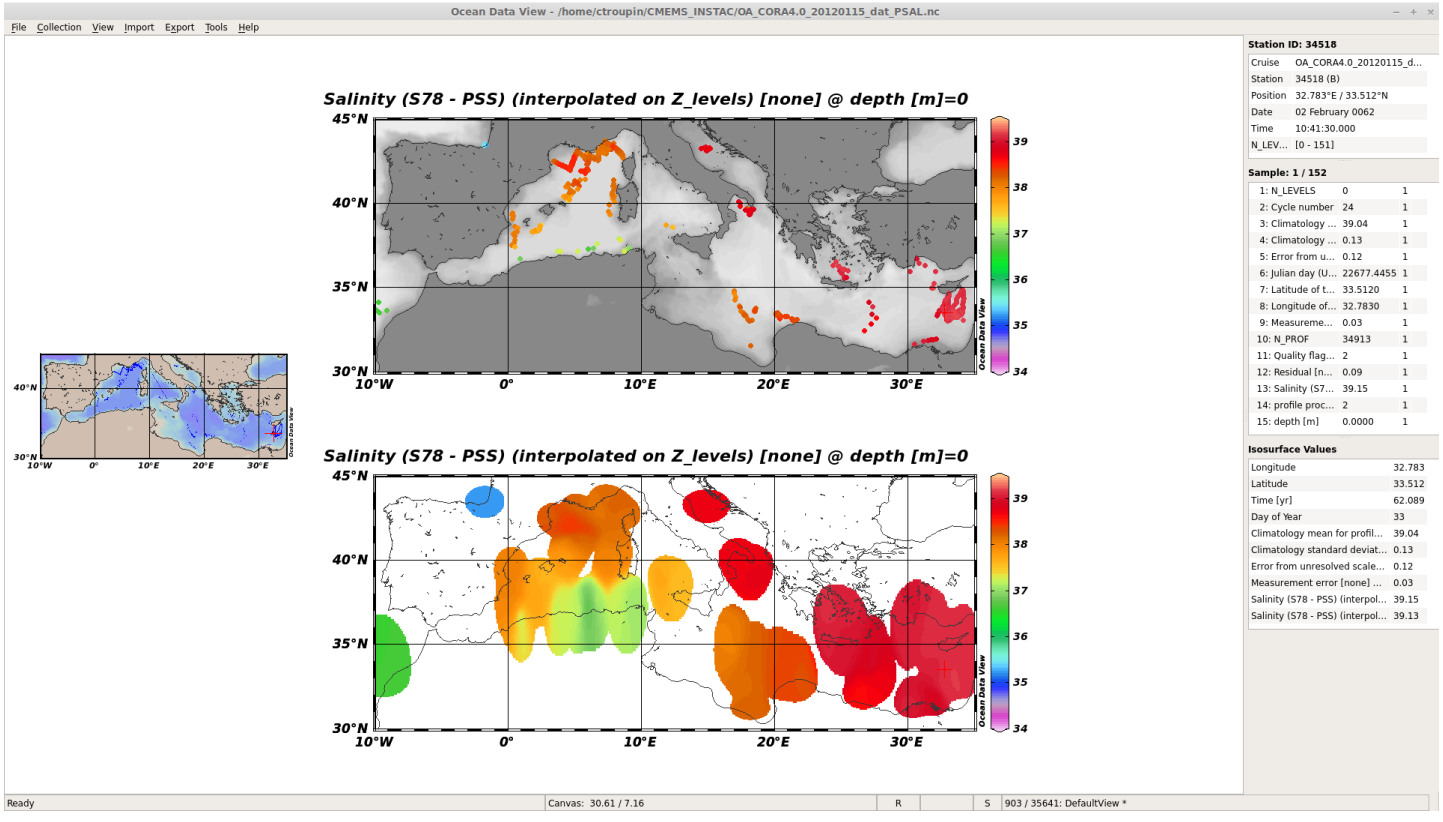
Surface window: gridding

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



Surface window: gridding

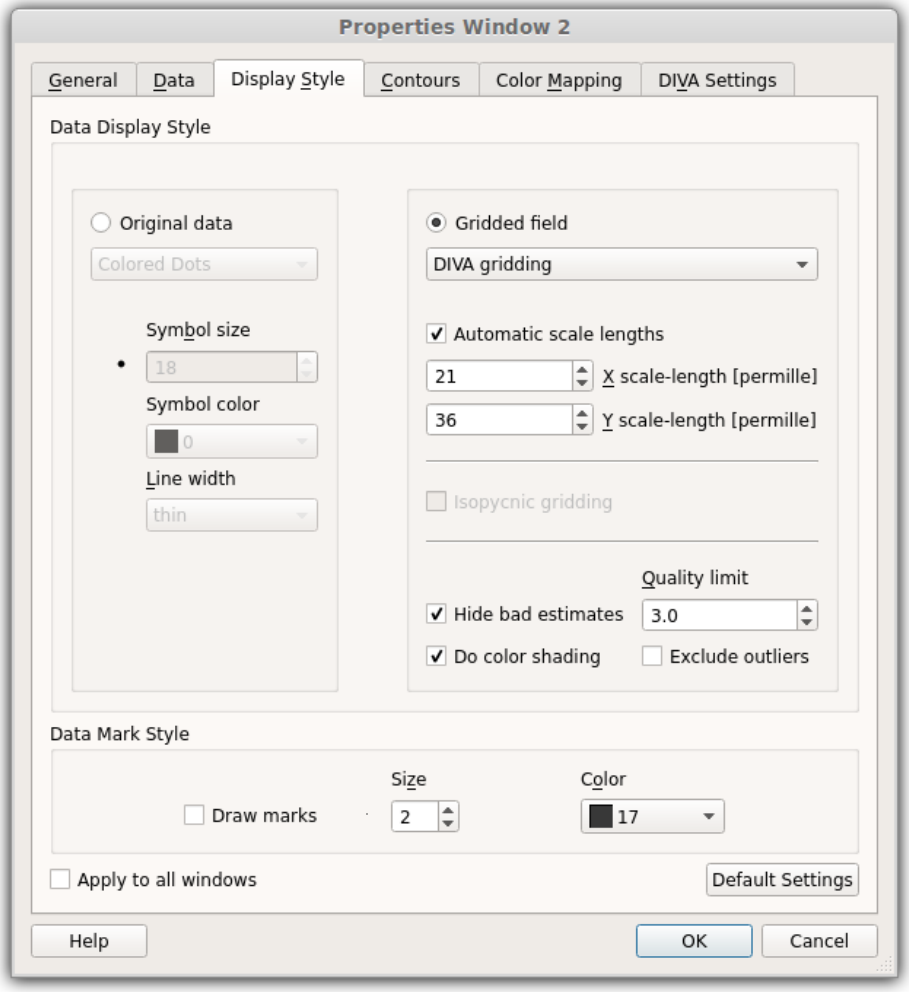
Gridded field of salinity



Normal interpolation does not consider boundaries!

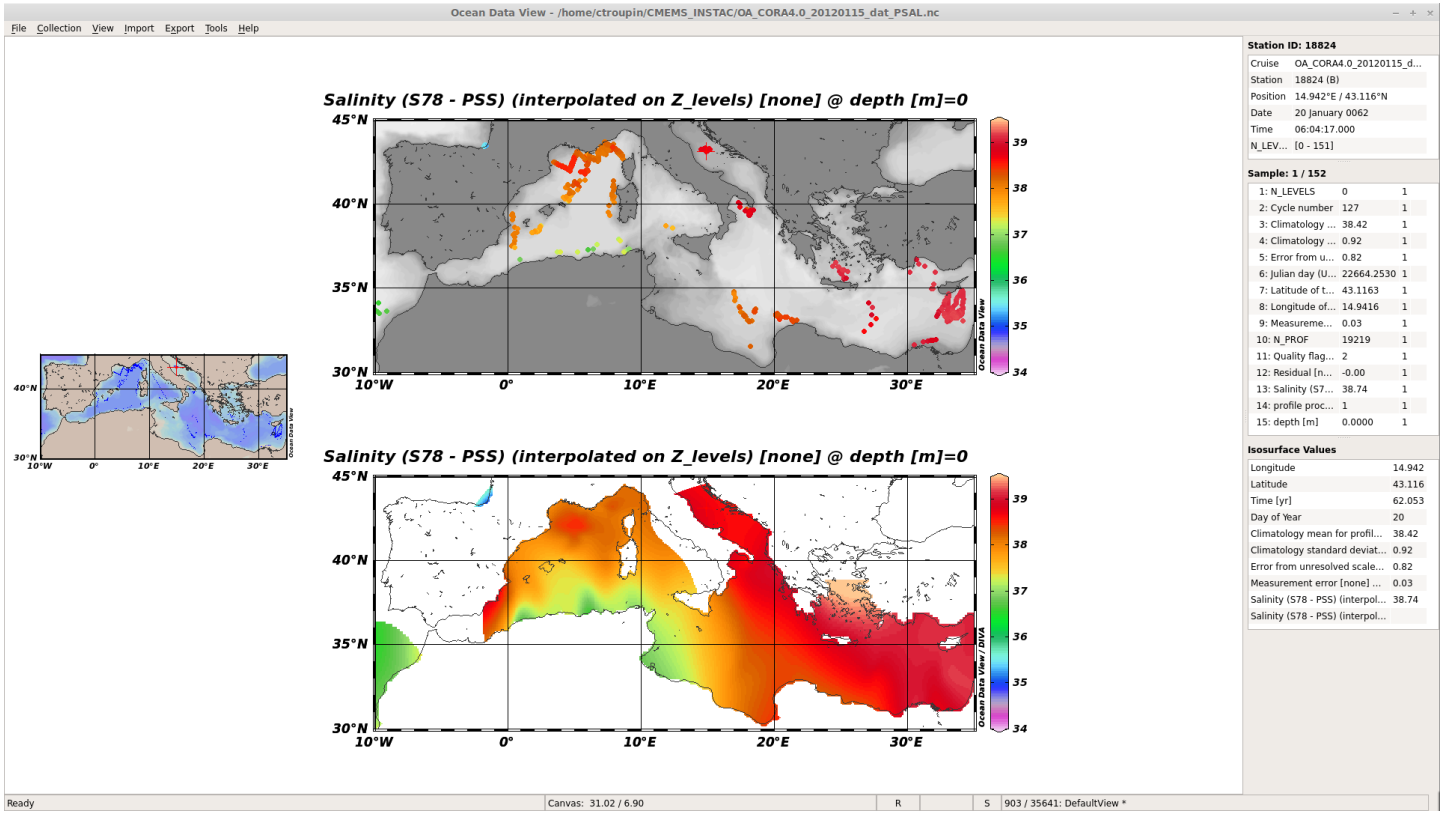
Surface window: gridding

Change Griddin method to DIVA Gridding



Surface window: gridding

DIVA gridded field of salinity

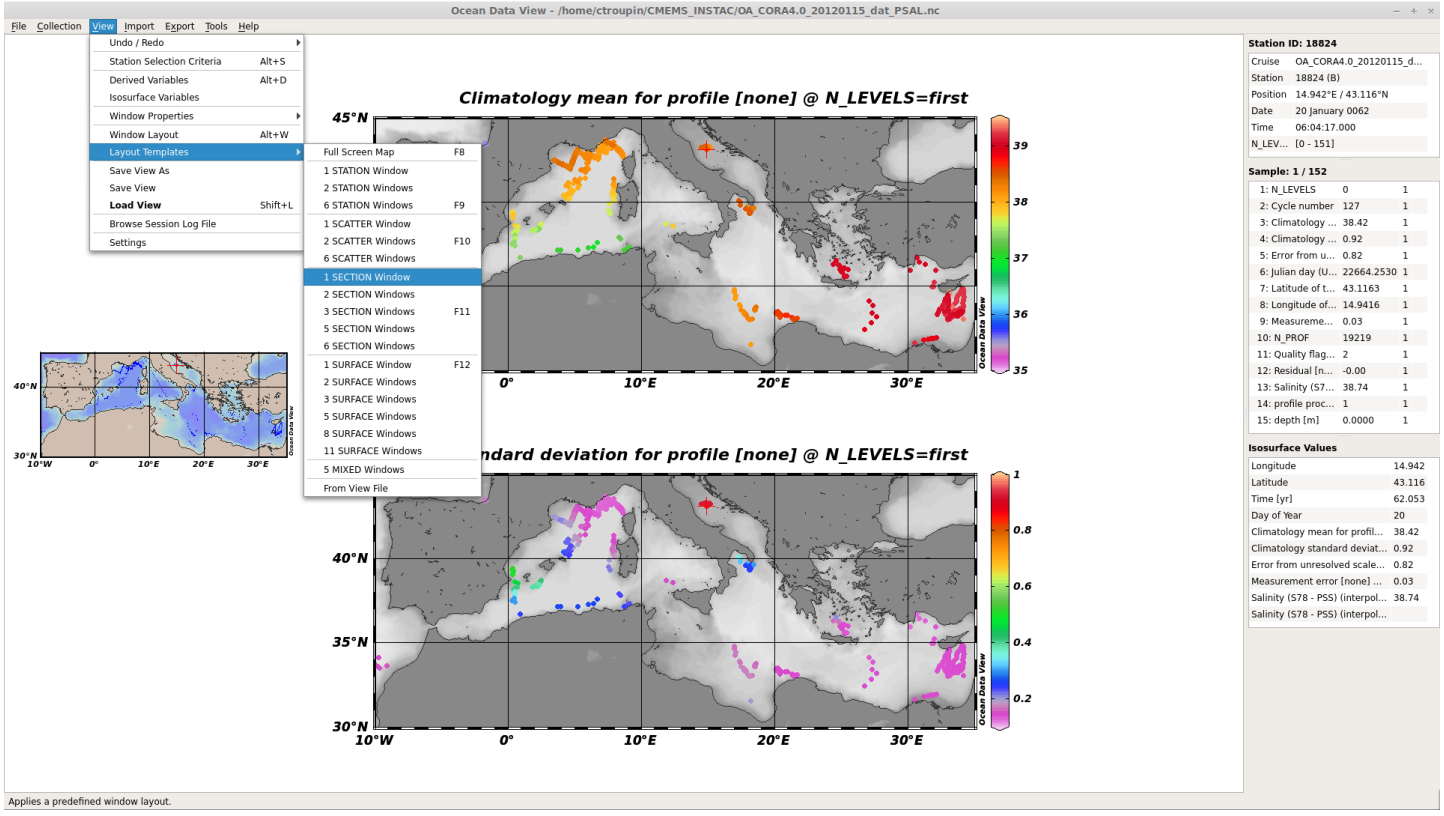


✍ Field with error above threshold is masked

✉ Interpolation technique is crucial with in situ data

Section window

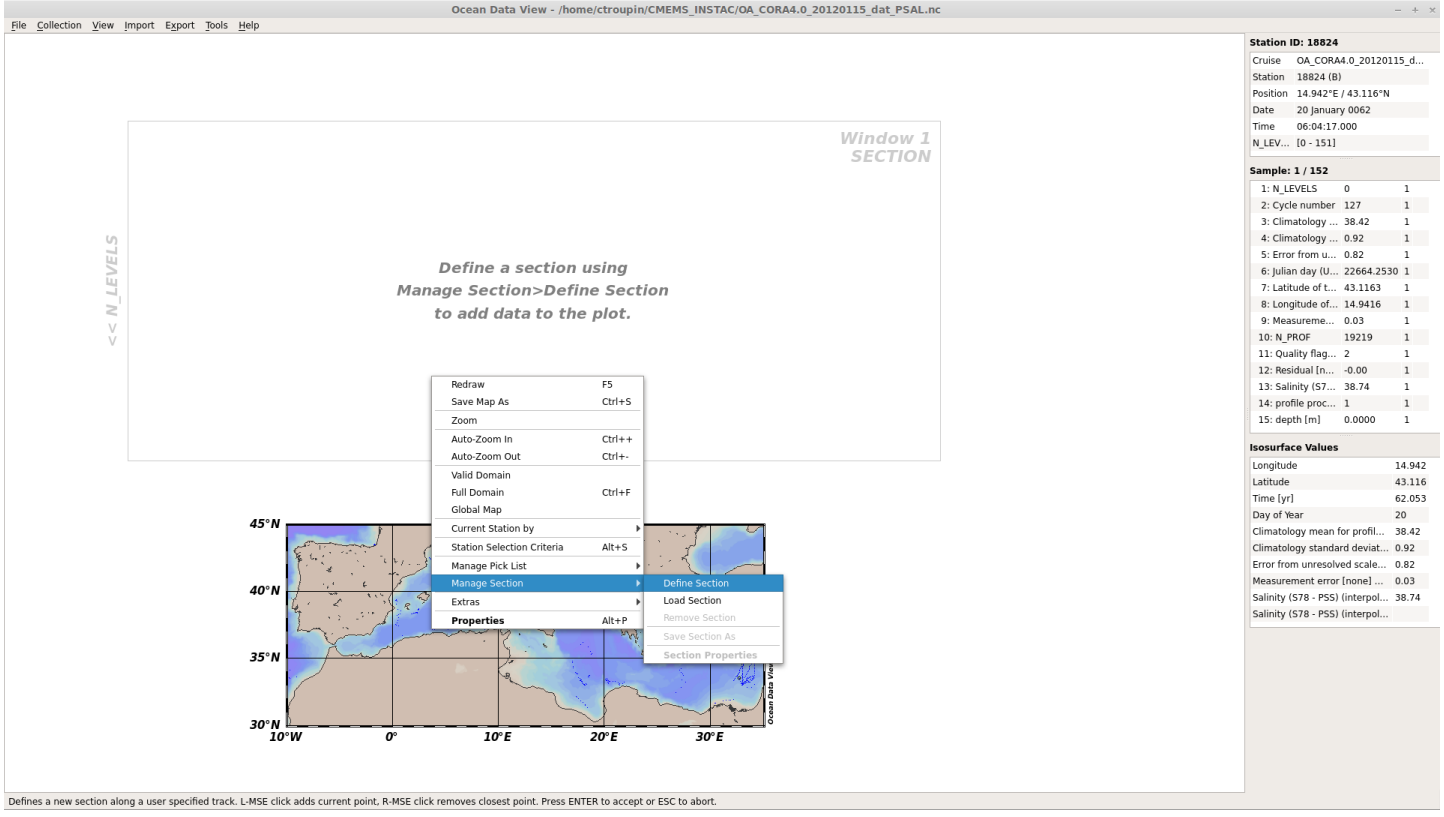
View → Layout Template → SECTION Window



Section window

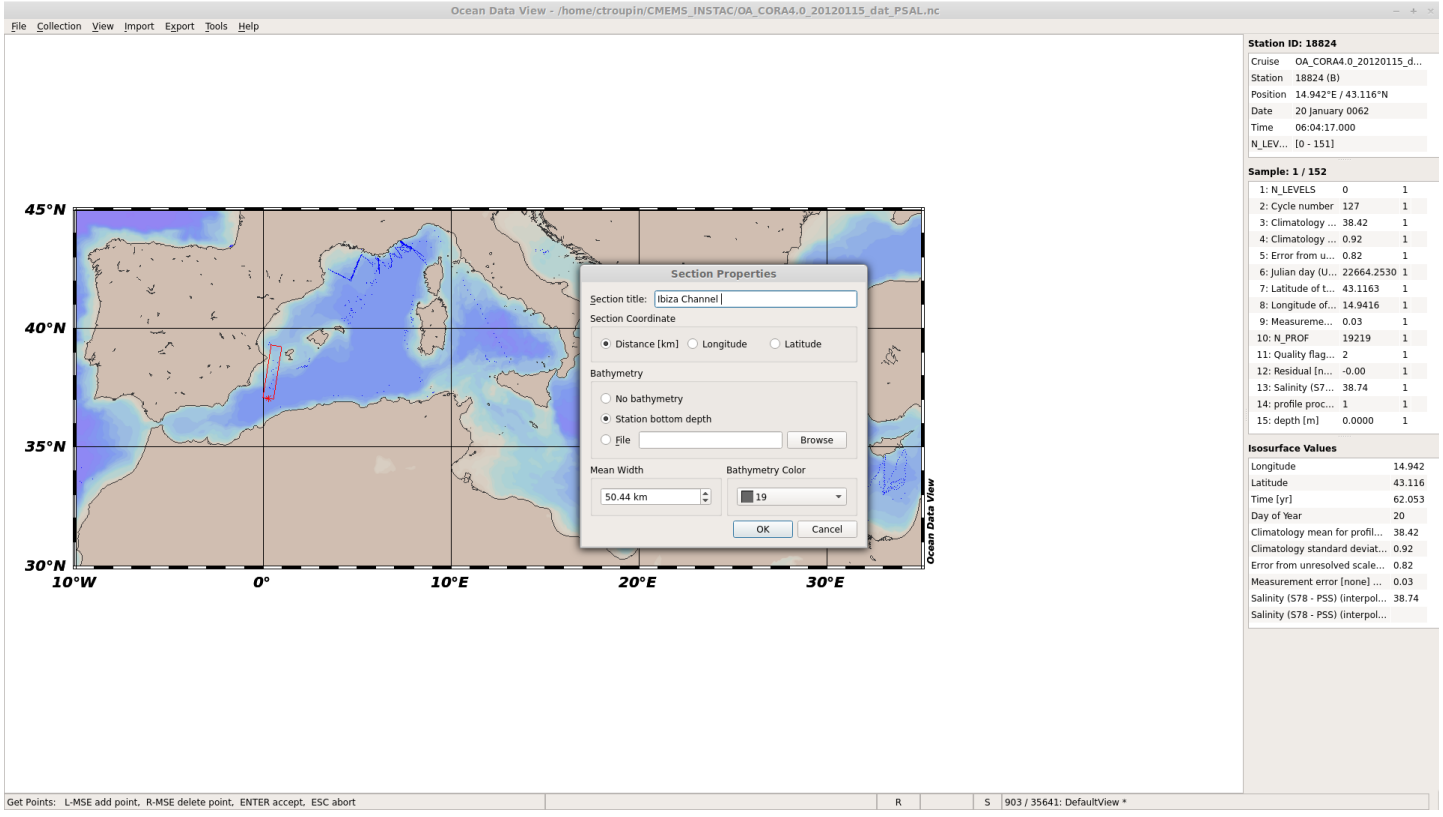
Right-click Manage Section → Define Section

Draw line along section



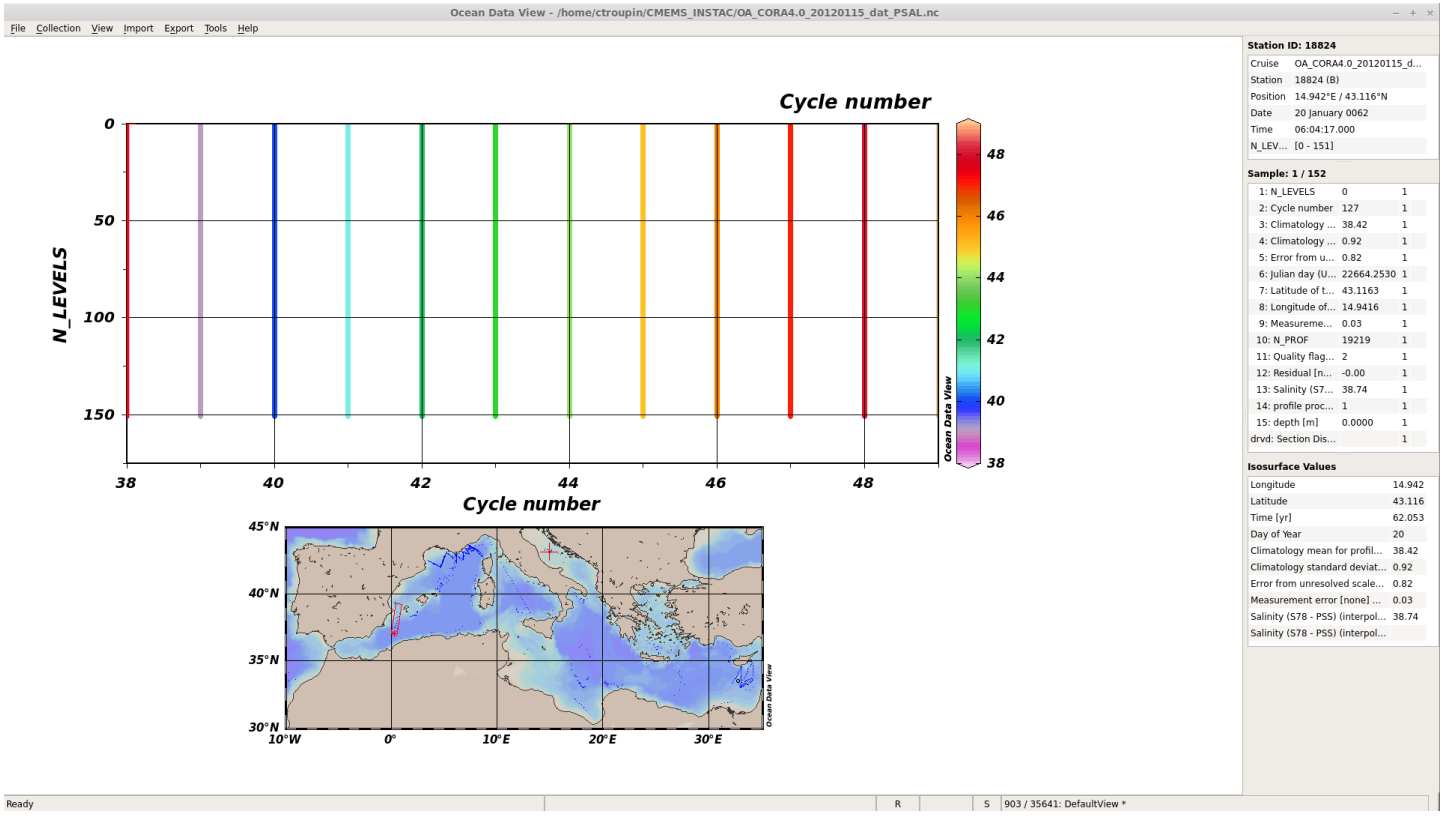
Section window

Edit Section Properties



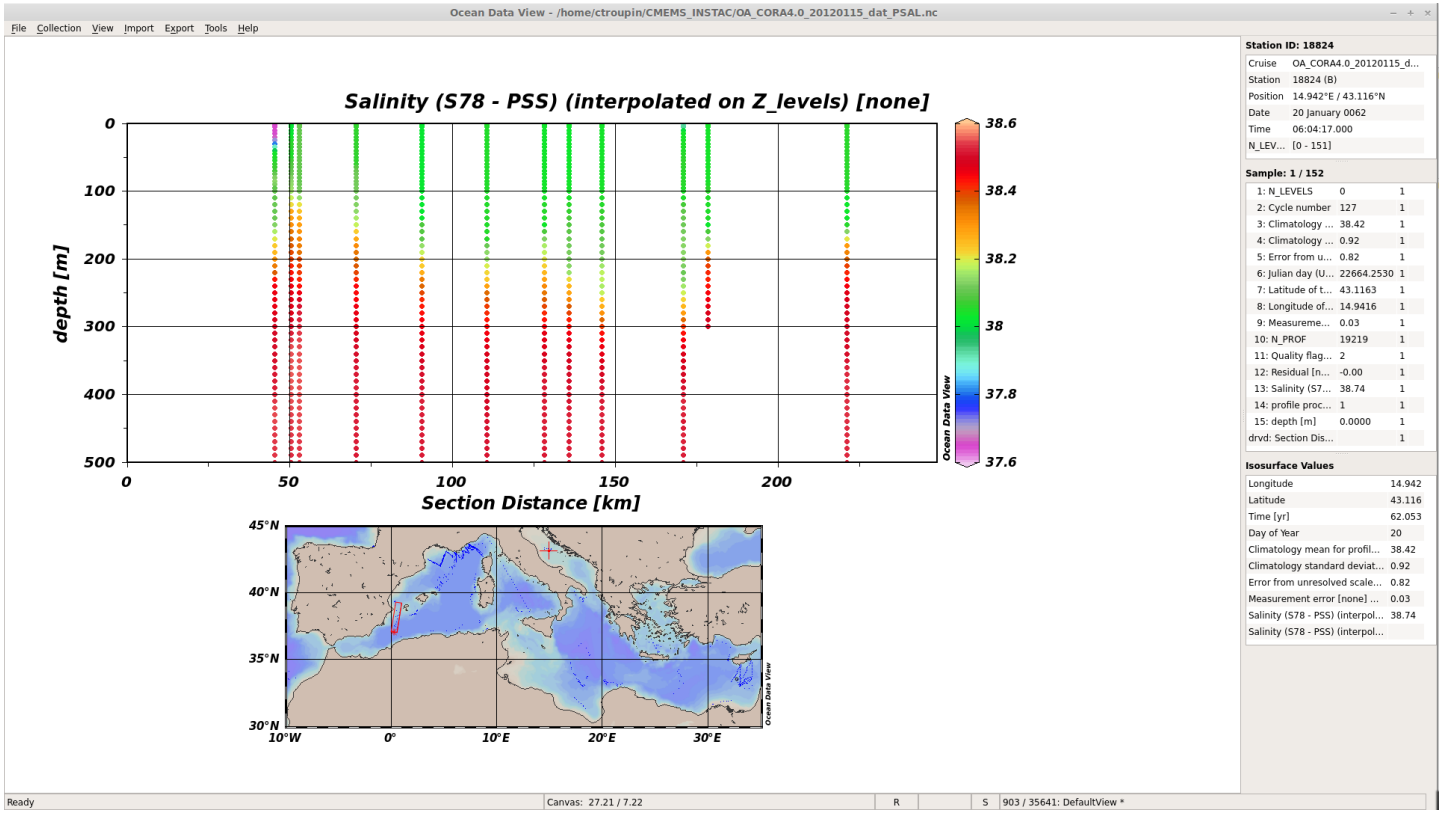
Section window

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



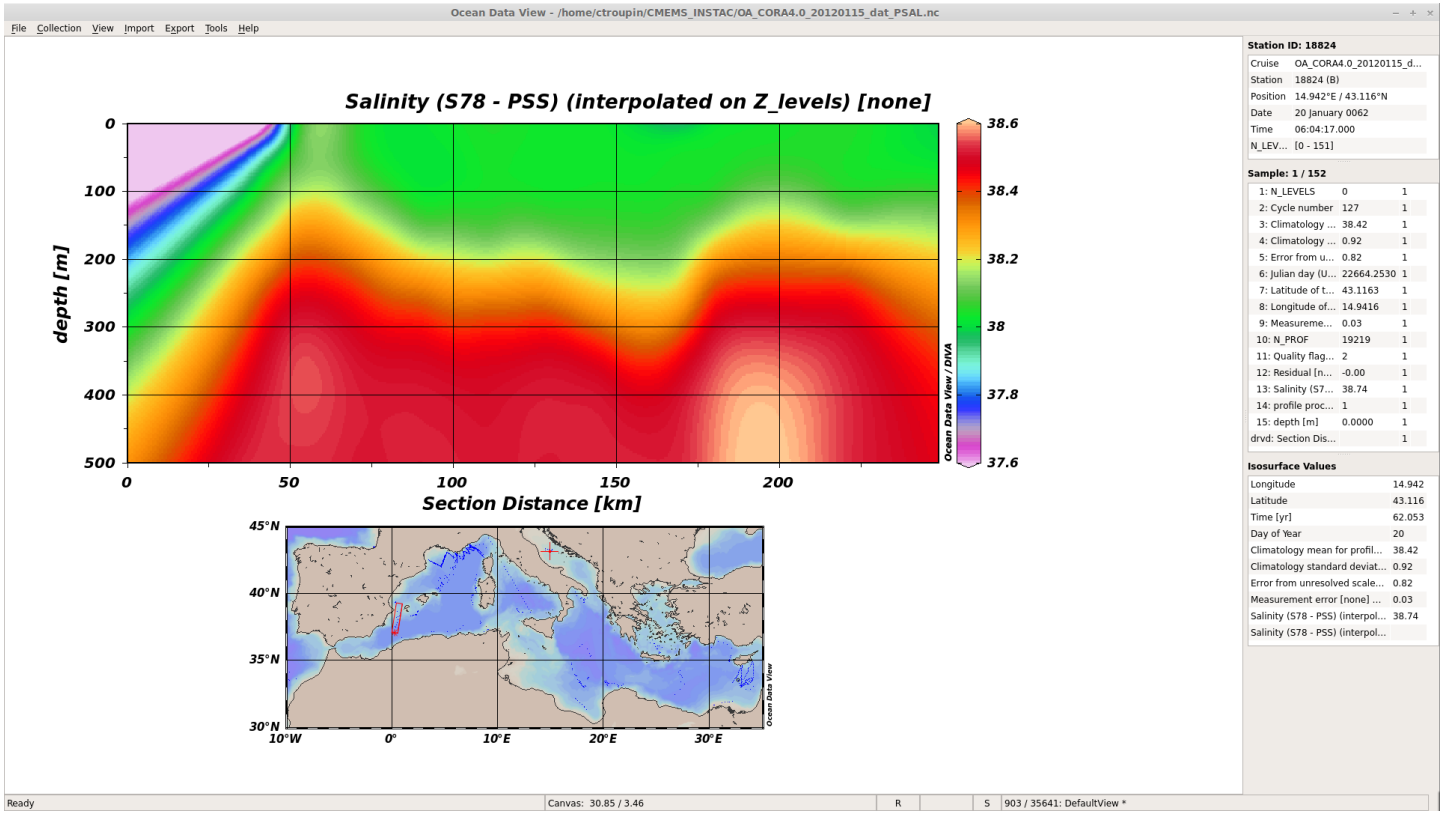
Section window

Set Z range between 0 and 500 m



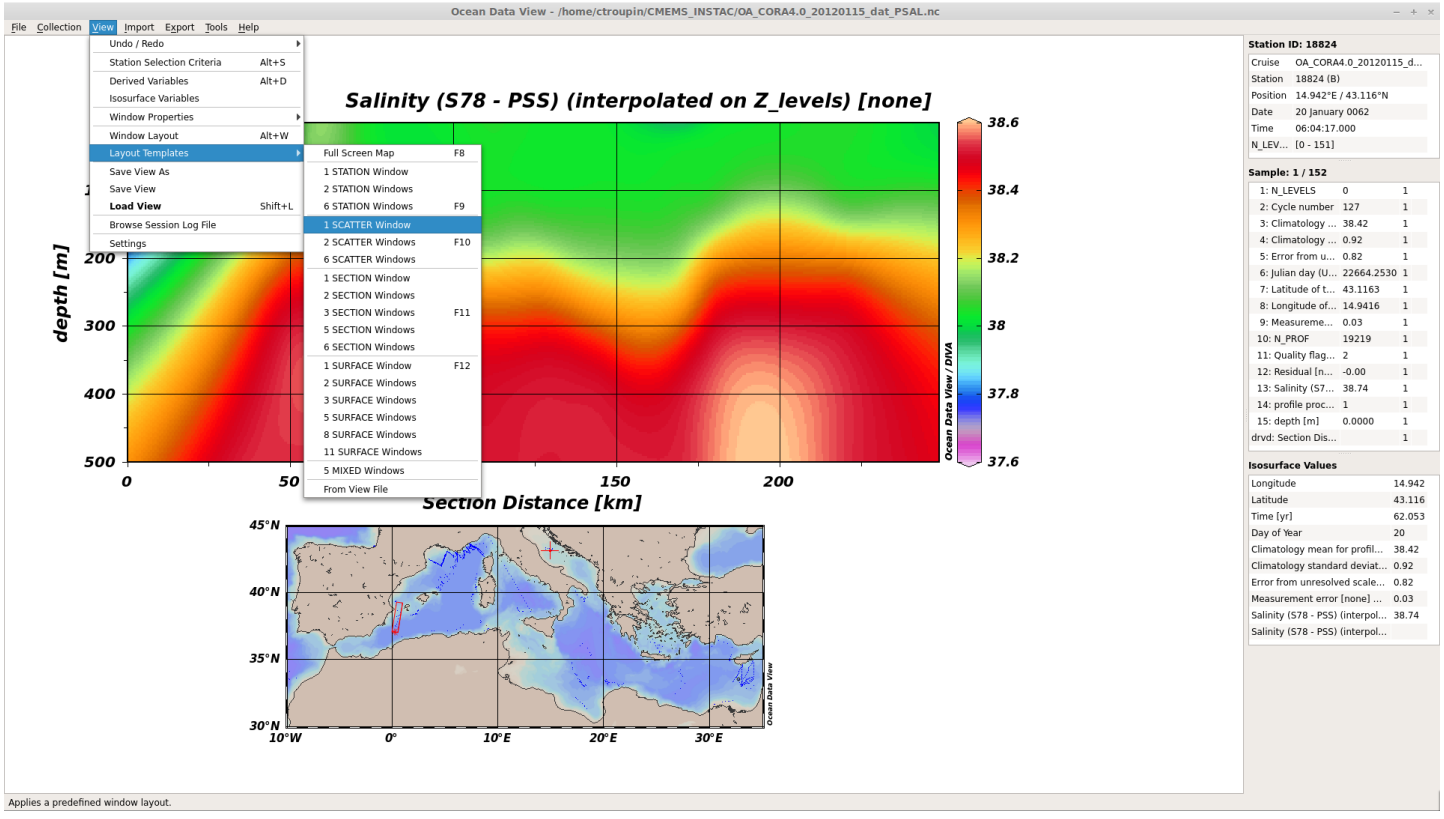
Section window

Grid using DIVA interpolation



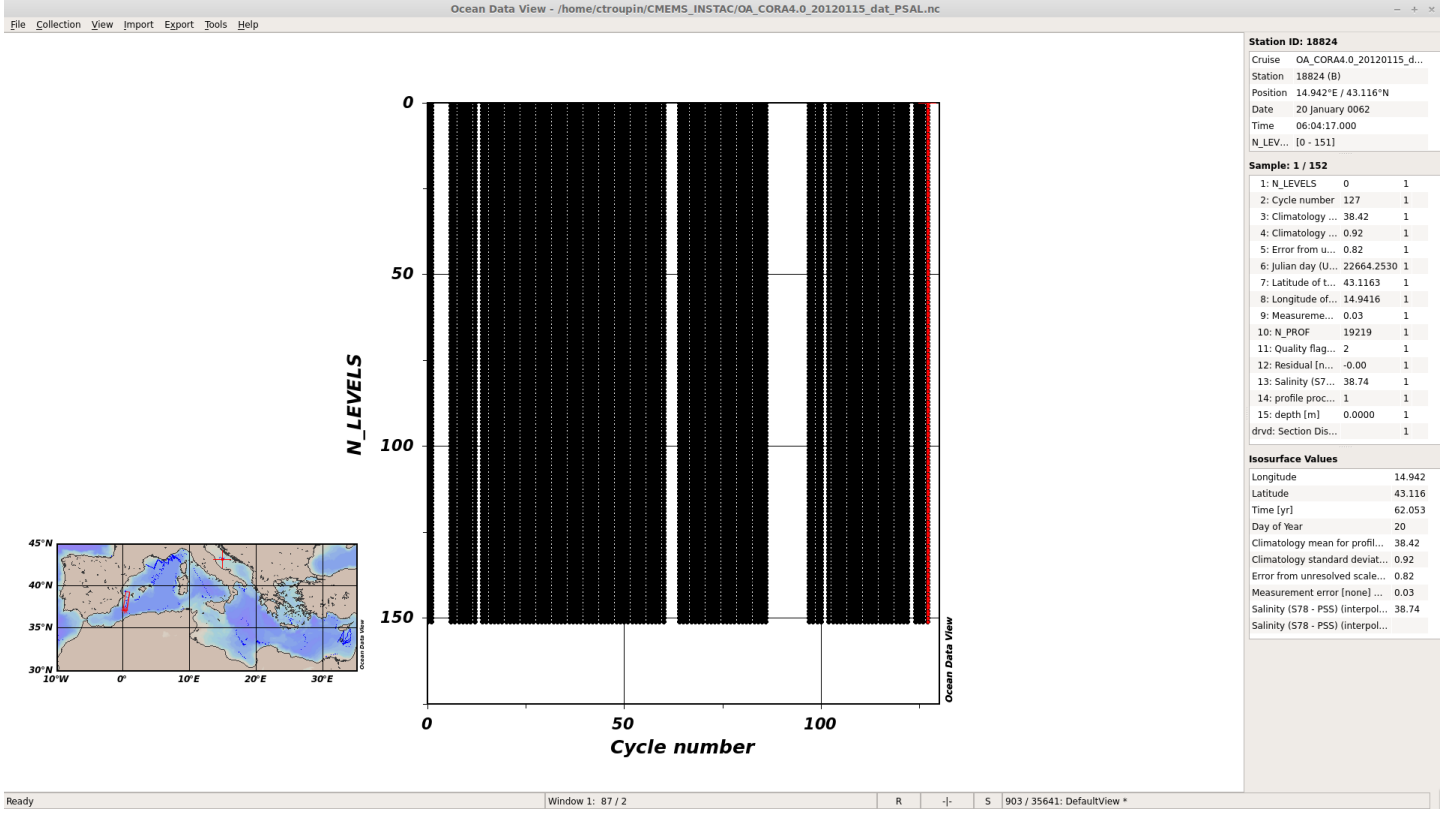
Scatter window

View → Layout Template → SECTION Window



Scatter window

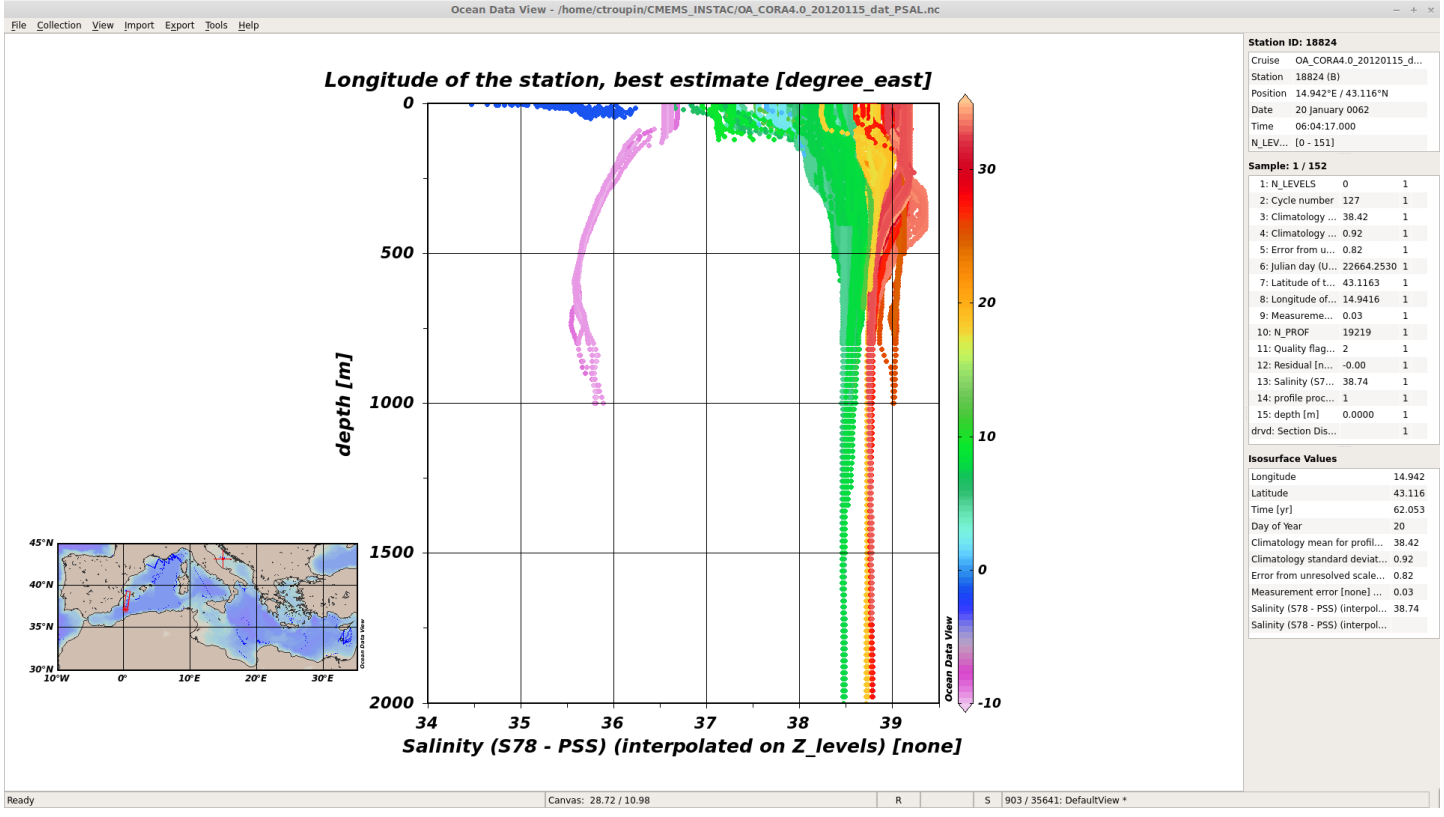
Change X, Y and Z variables



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

Scatter window

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

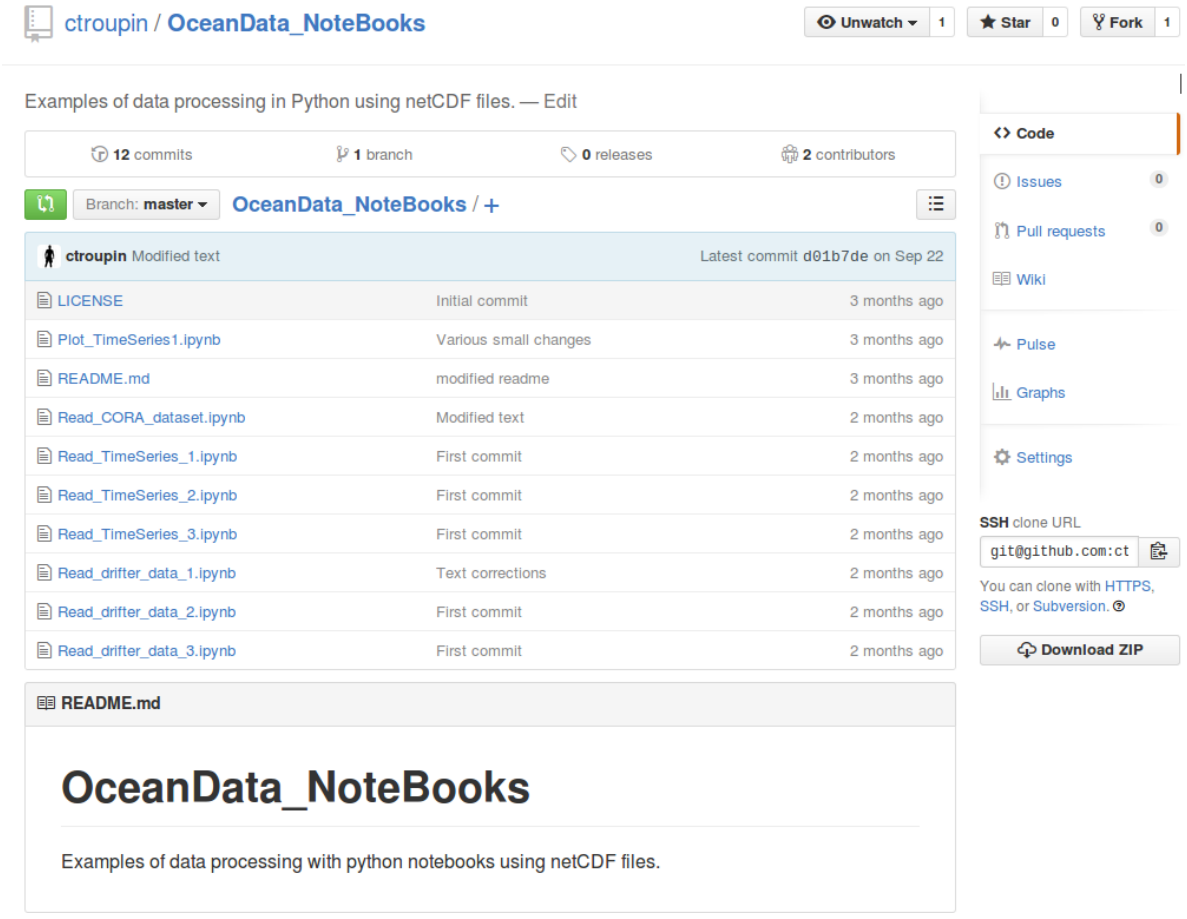
How to get the code?

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



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
```

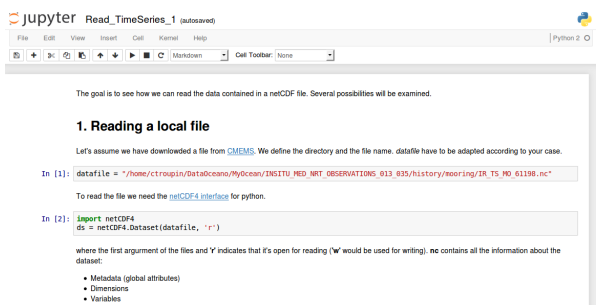
3. Go in the main directory

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

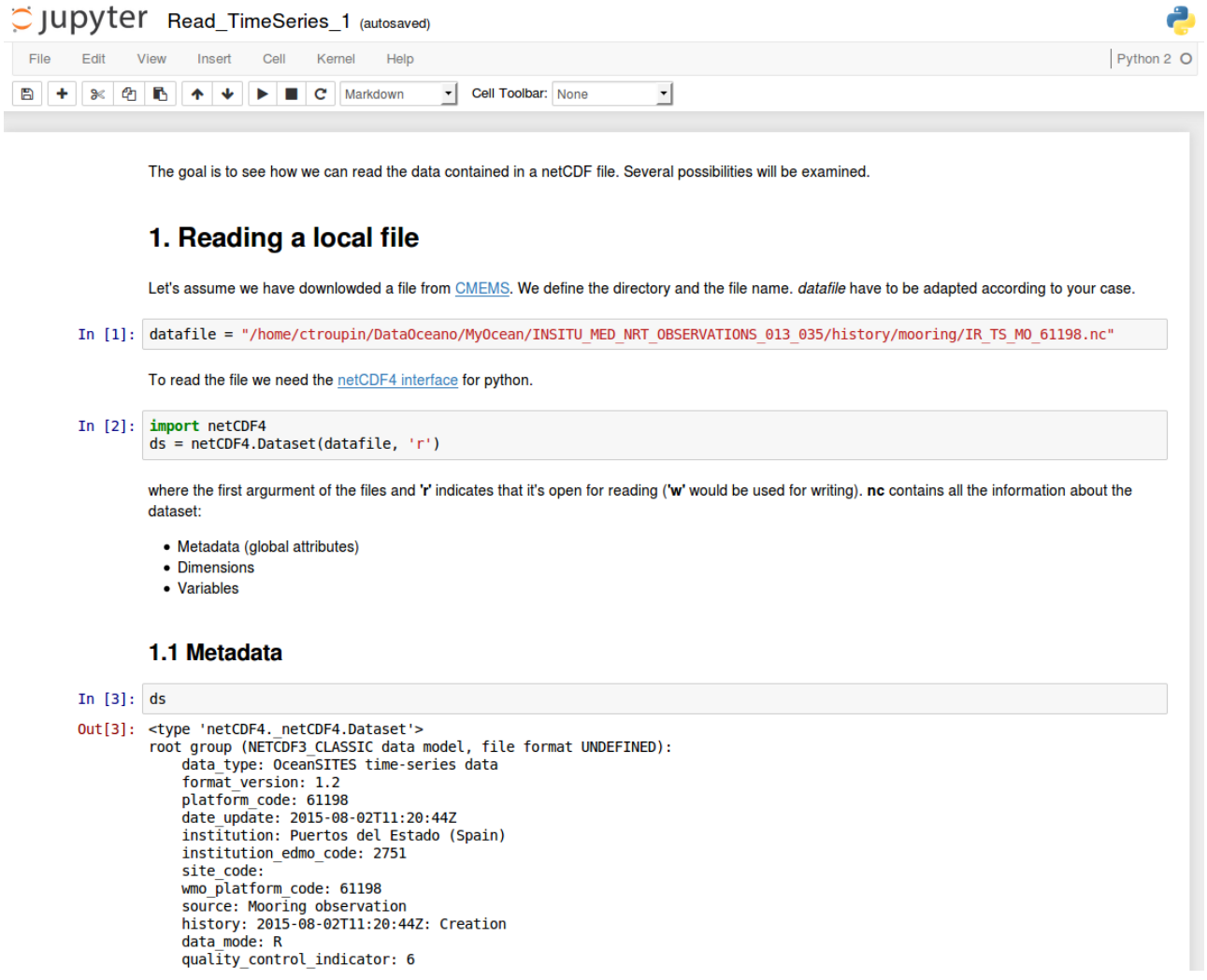
4. In a terminal, type

```
ipython notebook Read_TimeSeries_1.ipynb
```

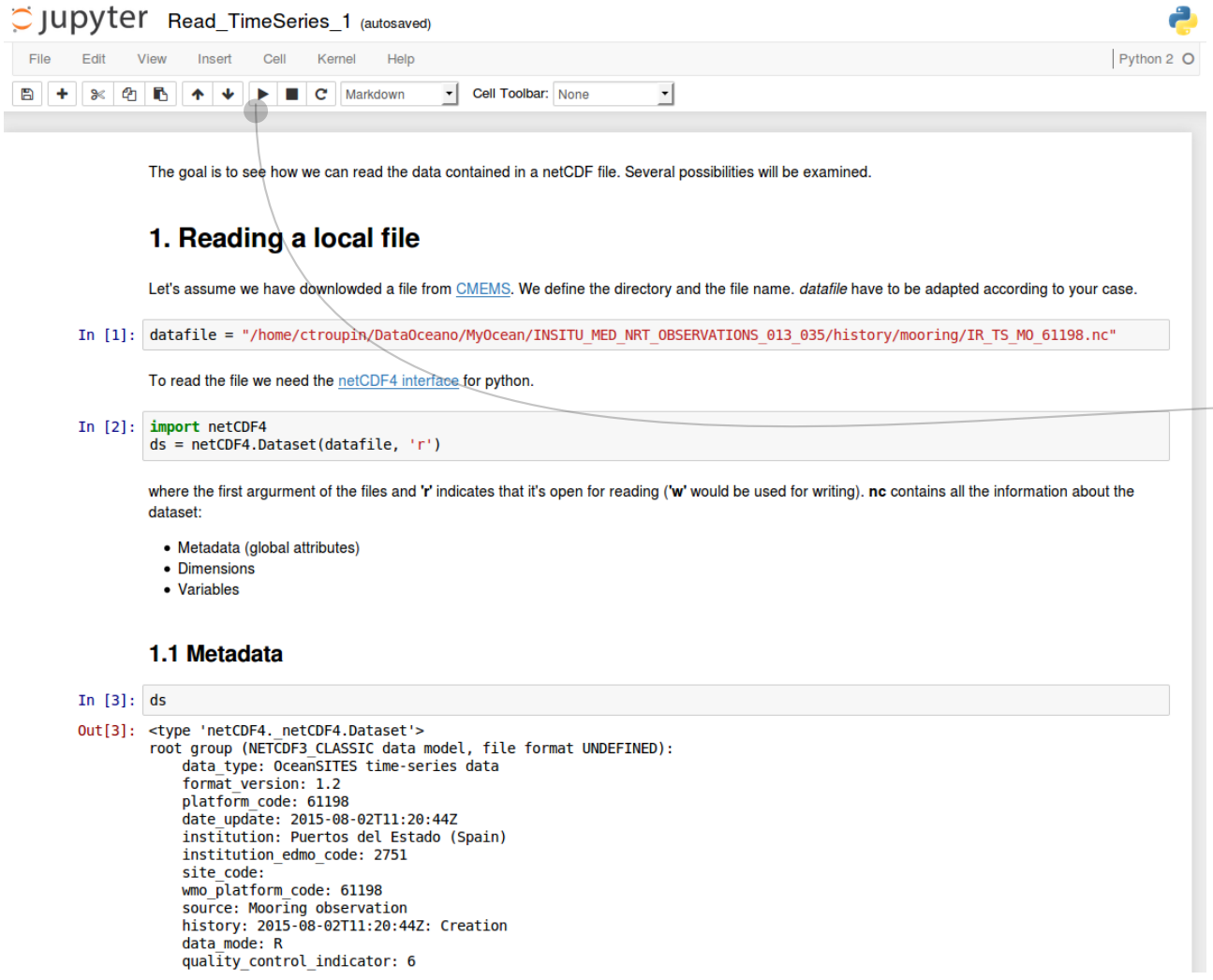
You should obtain something like that:



Structure of a notebook

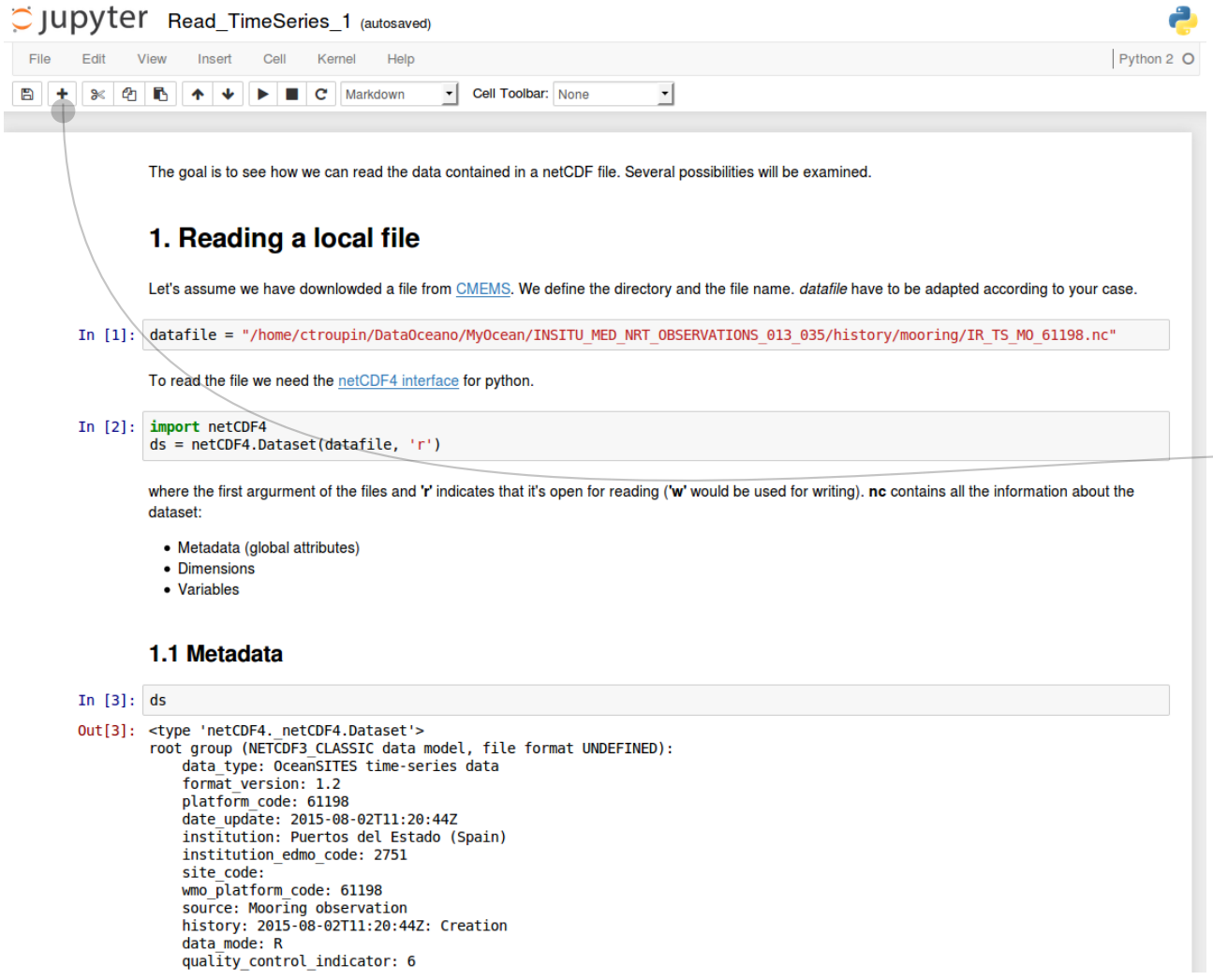


Structure of a notebook



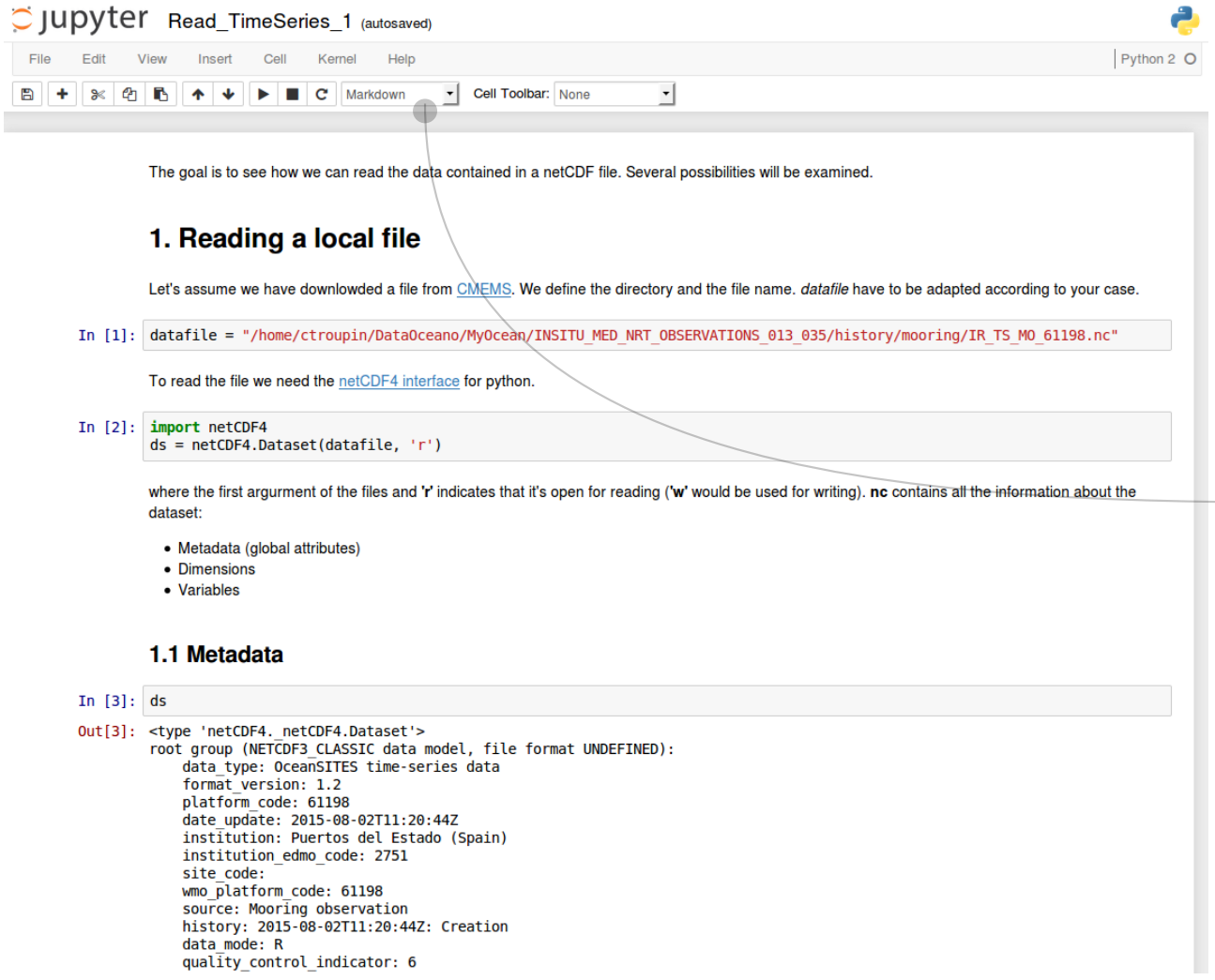
Run current cell

Structure of a notebook



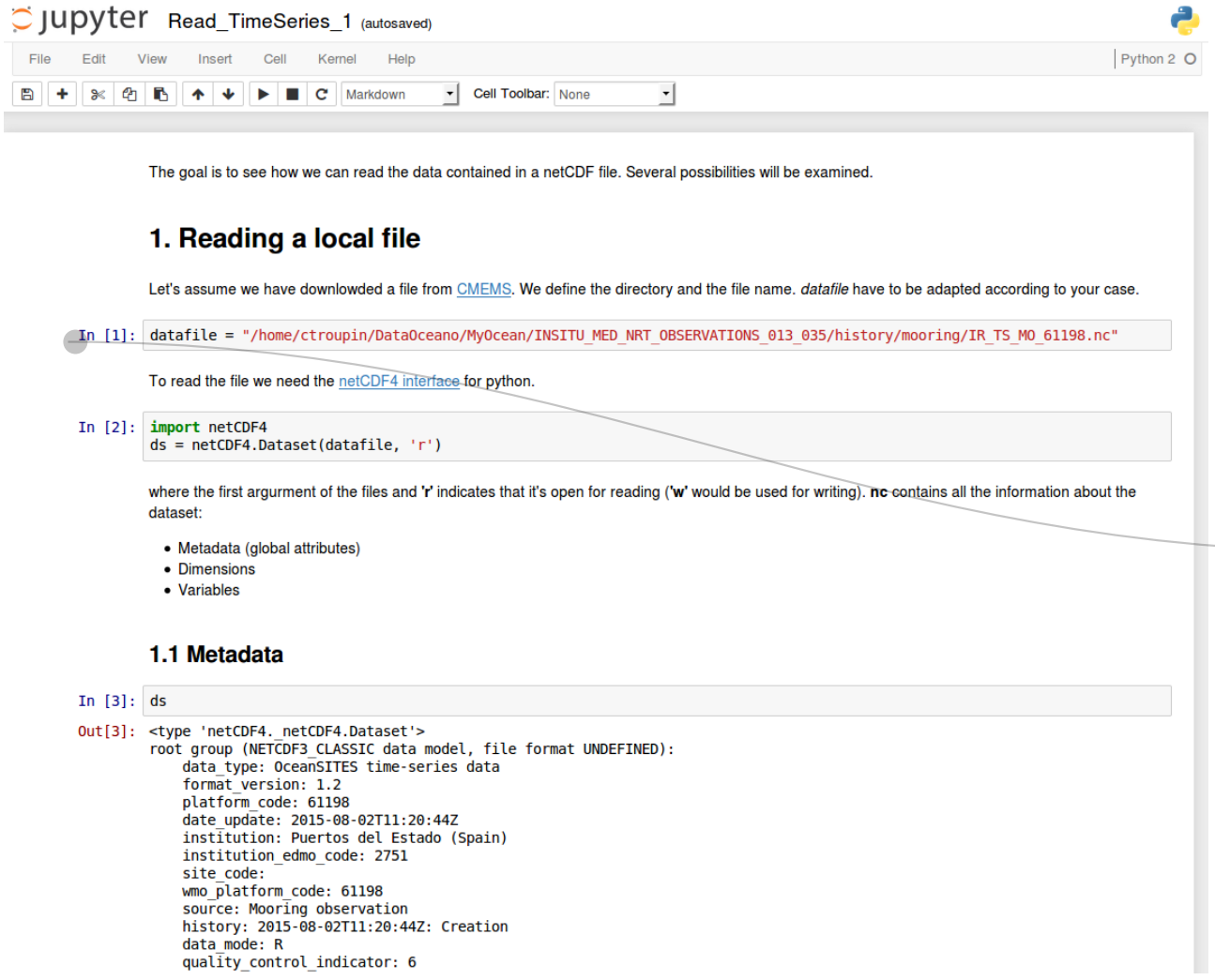
Run current cell
Add a new cell

Structure of a notebook



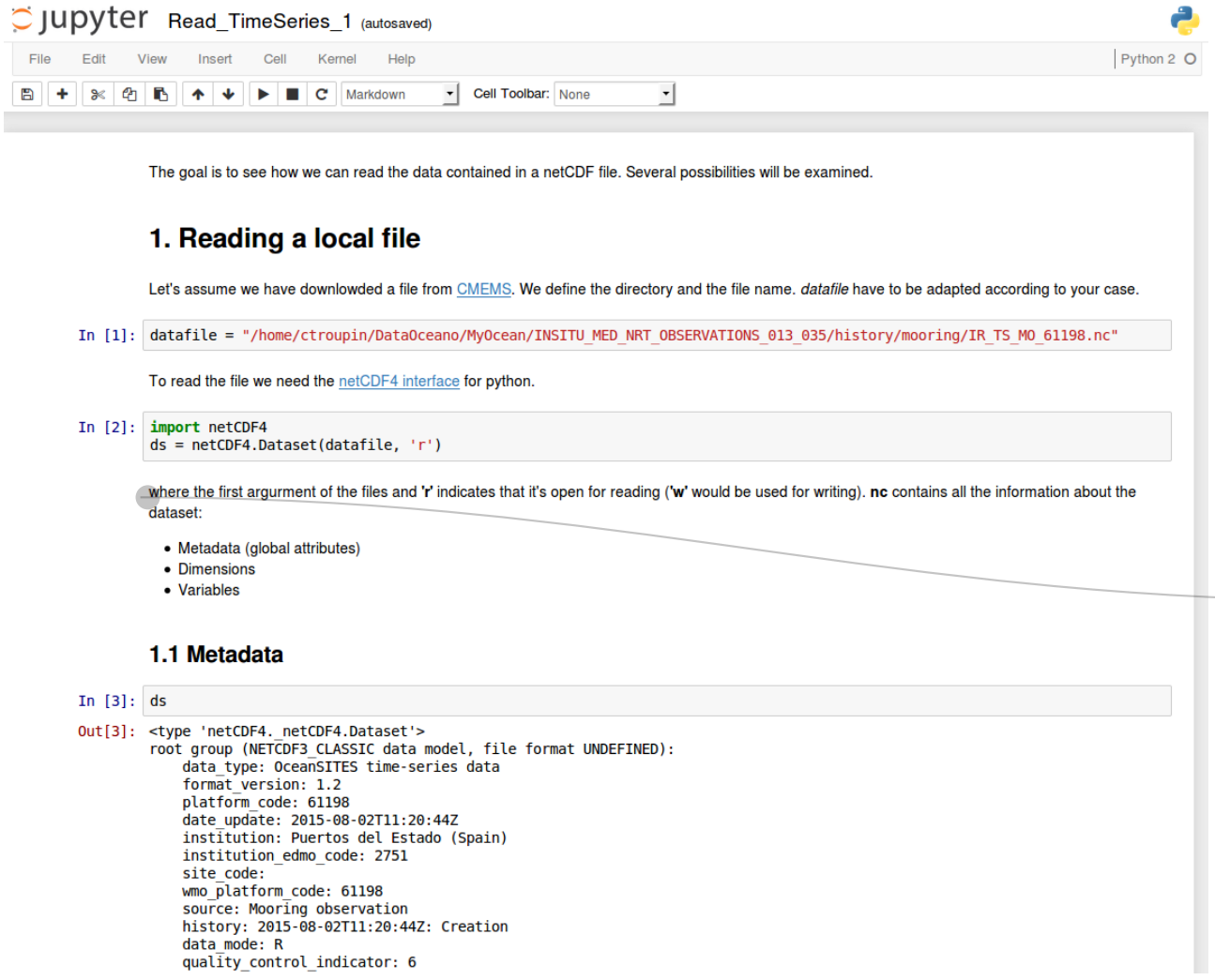
Run current cell
Add a new cell
Select type of cell

Structure of a notebook



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

Structure of a notebook



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:

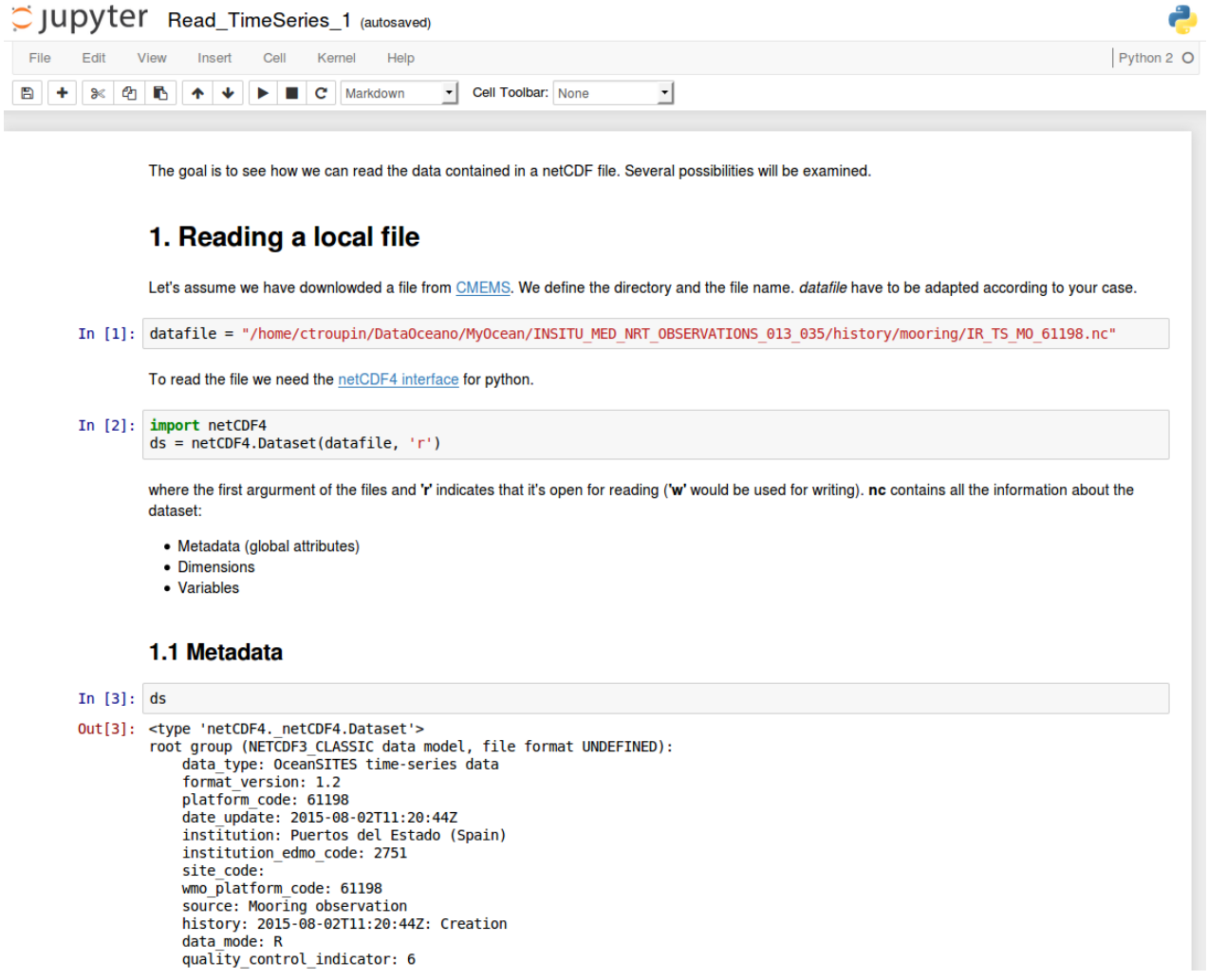
```
ipython notebook
```

Structure of a repository

In the directory containing the notebooks, type:

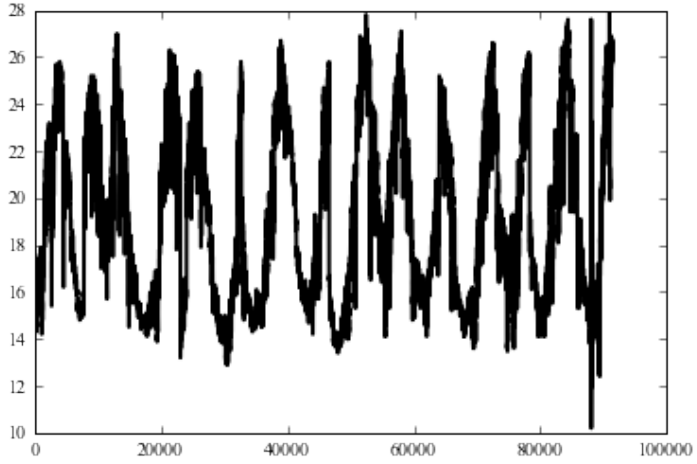
```
ipython notebook
```

You should get:



What's inside the repository?

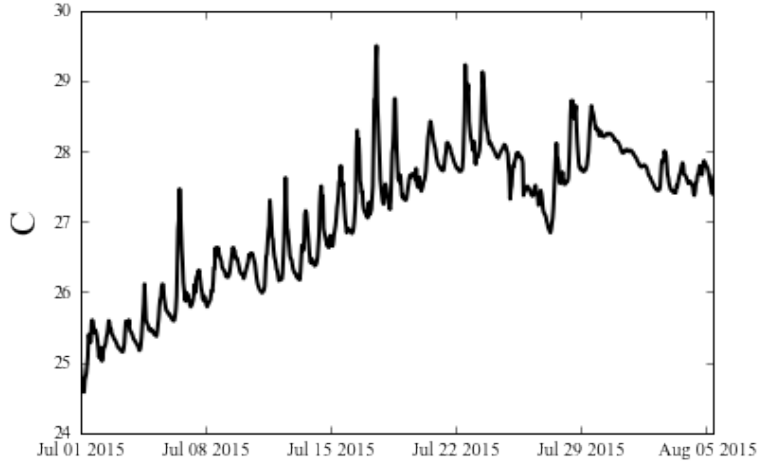
Read_TimeSeries_1.ipynb: reading a local netCDF file



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

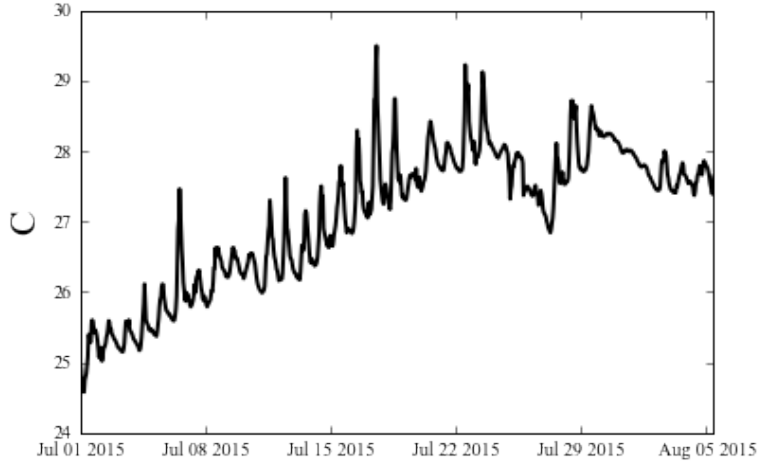


What's inside the repository?

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



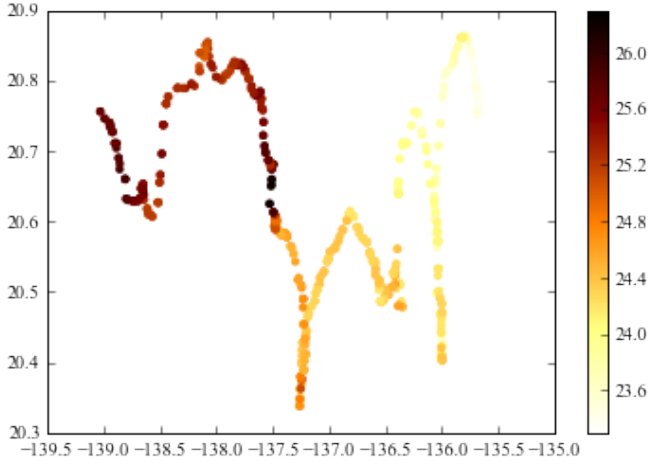
What's inside the repository?

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

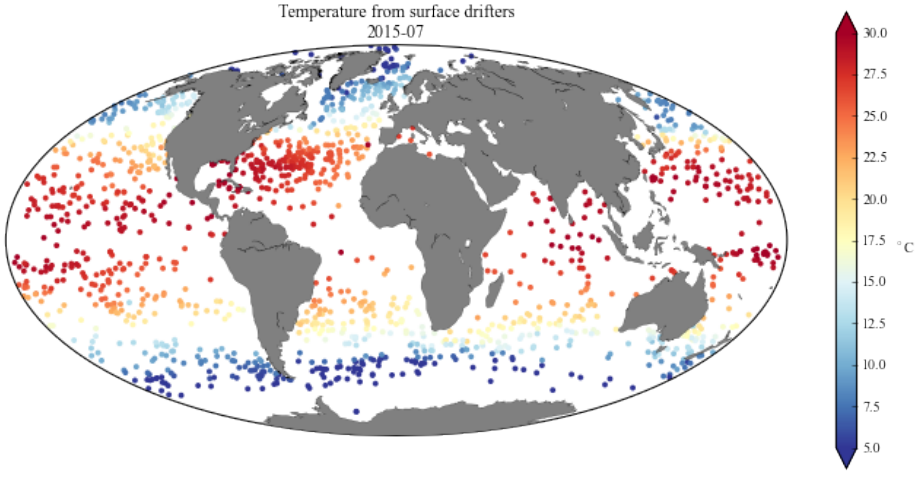
Read_drifter_data_1.ipynb: basic plot of a drifter trajectory



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

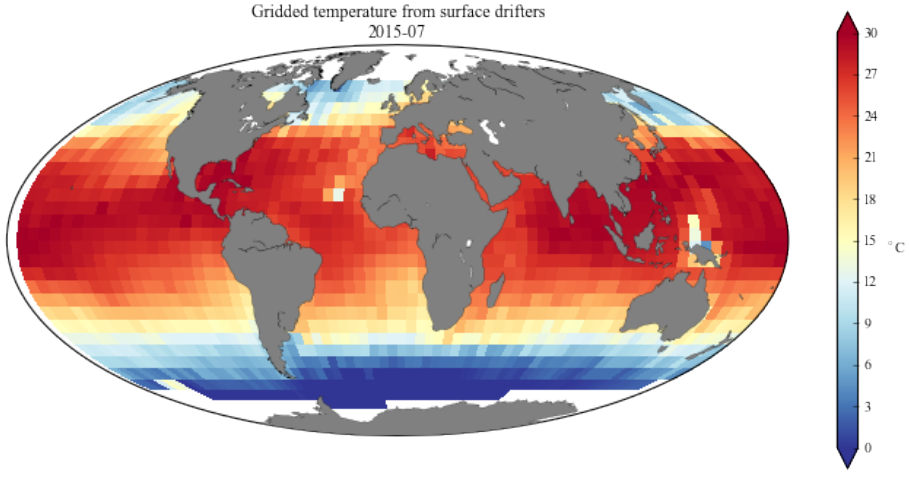
Read_drifter_data_1.ipynb: basic plot of a drifter trajectory
Read_drifter_data_2.ipynb: plotting temperature observations from drifters



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

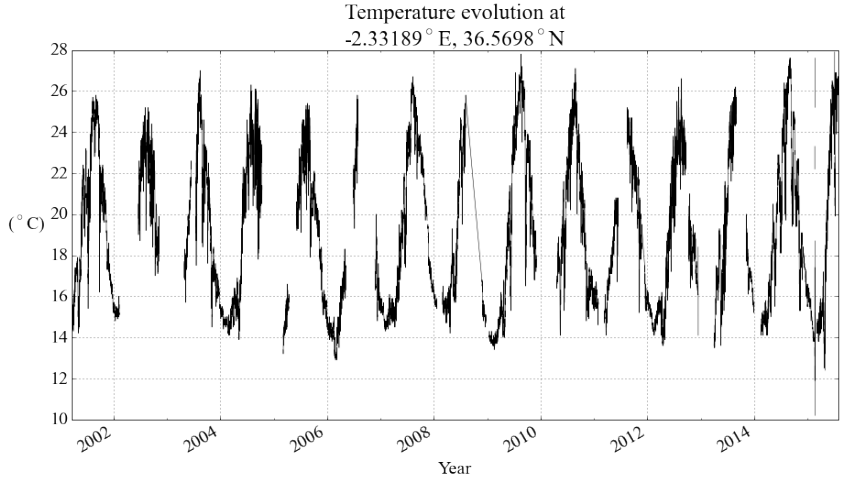


What's inside the repository?

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

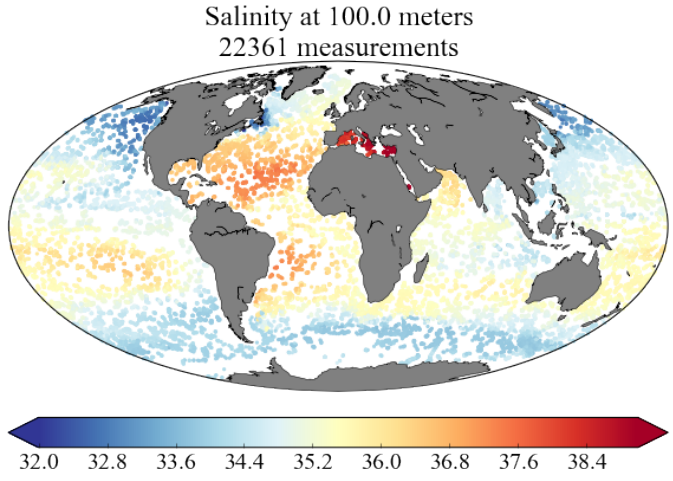


What's inside the repository?

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