The Seabottom Statistical Library in Python

Centro Nazionale Crisi, Emergenze Ambientali e Danno
Centro Operativo di Sorveglianza Ambientale

October 11, 2022

Roberto Inghilesi, Alessandro Mercatini

Description of the Seabottom Statistical Libraries v4.0

This software has been developed and upgraded in the framework of several different phases of "The European Marine Observation and Data Network (EMODnet) broad-scale seabed habitat map for Europe", known as EUSeaMap since the start of phase 2 in 2013. (see `EUSeaMap`). It started as a series of different FORmula TRANslator (FORTRAN) codes aimed to provide an estimate of the kinetic energy close to the sea bottom due to the effects of waves and currents in the Mediterranean Sea. This is a practical necessity, as wind-waves are calculated by wave models at the sea surface and currents are often calculated by oceanographic models at fixed depths (Z-models, as opposed to σ-models for which the levels start from the bottom topography). In time, codes were modified to cover also Black Sea, Iberia-Biscay-Ireland, and Macaronesia. The code for the evaluation of wave kinetic energy depends on time series of significant wave and peak period fields at the surface as obtained from high resolution statistical wave models like Wave Model (WAM) or Wave Watch III (WWIII) and a high resolution bathymetry in the area of interest. The estimate is obtained by a simplified method proposed by Soulsby in 2006, and described in the Annex "Compiling oceanographic layers" to the EUSeaMap 2019 - Technical Report and in the hereby cited specific literature. In order to obtain an estimate of the kinetic energy due to current at a small distance from the bottom, a simple adaptation of the boundary layer algorithm used in the oceanographic model used at Copernicus Marine Service (CMEMS) for the Mediterranean and Black Sea has been devised, with the idea that the application of the same scheme as a postprocessing of the 3-dimensional current field would provide a more consistent evaluation near the bottom. Also in this case, the method is highly dependent on the available bathymetry of the local area considered. All methods are clearly gross simplifications of the real physical processes, acceptable only in providing long term statistics. By 2016 CMEMS provides sufficiently high resolution simulations of physical characteristics of the European seas to be successfully used to evaluate the bottom kinetic energy. In 2022 it was decided to implement all software in the form of a library which would provide a way to easily extract the desired estimates using only CMEMS oceanographic outputs and EMODNET bathymetryes. The code is written in Python for several reasons. The first is that is a widely used language which has been adopted by almost all scientific communities. Second is that is particularly apt to work in object-oriented environments, has effective methods to deal with Network Common Data Form (NetCDF) and leads to surprisingly simple installation procedures for the libraries. Actually only the path of the library needs to be set in the script/notebook. On the other hand, even though some efforts have been spent to improve vectorialization, the execution of a long term statistic evaluation could take some time. Not too long though (something between a coffee and a compact lunch), so in order to keeps the installation of the library and the code as simple as possible, no parallelisation has been considered at the present stage. Another practical need is the evaluation of biochemical parameters (like salinity, O_2 or CHL) near the bottom. In this case no effort has been made to provide a boundary layer model, the user must be aware that the evaluation is based on the simple extraction of the parameter data at the level which is closest to the bottom. In the case of the temperature, CMEMS provides for the temperature at the bottom, so the library provides only the framework for the evaluation of the desired statistics. In order to keep the management of data sufficiently simple, the adopted bathymetry is the 2016 version, based on 12 data blocks. The library for interpolating the EMODNET 2020 product works on many blocks of high resolution data. Users may decide to use this version in order to gain some accuracy at the price of a longer pre-processing. The libraries have been devised to work on CMEMS oceanographic products (but some adaptation would be needed for products different from Mediterranaean, Black Sea and IBI), the simplicity of usage in terms of auto-consistency, means that the user is not required to specify the area of application or the time periods, the library simply works on all the files and reads the parameters from the netcdf imput files. This of course requires some extra care in the preparation of input data obtained from CMEMS. In the following sections the practical usage of the three main libraries is described as a tutorial to the work with the library implementations using notebooks and scripts. First section is the description of the package and the installing procedure. Second section is about the interpolation of bathymetry. Third is the evaluation of current-based kinetic energy. Fourth is the evaluation of wave-based kinetic energy. Fifth is the evaluation of biochemical parameters near the bottom. It is perhaps important to stress that as CMEMS is an active, still growing project with an increasing number of products, it is impossible to provide a code which work as a post-processor on every different (by domain area, forecast or reanalysis, ocean, or wave or biochemical) model output, and each data file possible organisation. Said that there is a reasonably high level of standardisation in all the CMEM products, so that only a small change in the code may be enough to work on an untested product. In most cases, to make it work it should be sufficient to add the name of the variable, the name of the missing value, the scale factor, the offset associated to the variable to the relative list, which are easy to find in the code. There is a section of the document where the possible problems are discussed.

Description of the package and implementation

This package contains 3 main directories: the directory 'bkenergy' which is the actual source code, the directory 'drivers' which contains some python notebooks and scripts which illustrate how to make the best and immediate use of the library, and a directory 'docs', containing the present file. No real installation is needed, just copy the directory 'bkenergy' contained in the package seabottom4 somewhere in the computer and be sure that the drivers are able to access the libraries and the data files. Some suggestions follows on how to do it quickly and painlessly. In order to put the libraries to use with the drivers, there are some operations that are in order. To make the drivers work is sufficient to

1. modify in each notebook/script the line:
   sys.path.append(r'/Users/uranus/workarea/bke/seabottom4') with the actual path of the directory seabottom4 in the user's computer

2. import the necessary libraries as in:
   from bkenergy.mkbathy.ibathy import Gridinfo, Sbathy
   from bkenergy.waves.iwke import Wbenergy

3. arrange the workspace modifying the lines: root='/Users/uranus/workarea/bke', where root is the directory containing both the 'data' and the 'seabottom4' library. (See fig.1). 'data' contains at least the directory 'emodnet' with the

EMODNET 2016 files, (or a directory emodnet_2020 with the netcdf files of the 2020 product) the directory ‘’outs for the output of the libraries and the directory ‘cmems’ with all separated subdirectories for the model data extractions.

4. bdata=root+’/data’+’/emodnet’

5. out=root+’/data’+’/outs’
   Different arrangements can, of course, be made, provided they are consistent with the initialization of the objects provided in the following sections. #emodnet bathymetry

6. bds=[”A2”,”A3”,”A4”,”B2”,”B3”,”B4”,”C2”,”C3”,”C4”,”D3”,”D4”]

7. nb=[bdata+”/”+bd+”.mnt” for bd in bds]

8. alternatively to the previous two points, in the case of EMODNET 2020, an alternative approach may be more effective: if fileList = os.listdir(bdata) is the directory where all netcdf files are stored, fileList is a list of all files present in that directory that can be read,verified and passed as argument to Sbathy. Examples of different possible approaches are provided in the drivers

   In both cases it is important that the files are read in lexicographic (dictionary) order. #cmems currents ibi

9. cdata=root+’/data’+’/cmems’+’/ibi/’

The remaining part of the driver is described in the following sections. The package bkenergy is made of four subpackages: mkbathy, currents, waves and mkbchem. The libraries are composed of object-oriented modules (ibathy, icke, iwke, ibchem) written in Python. To work properly, they necessitate only the standard python module NETCDF4. The code can be implemented easily as described before by referencing the path of the seabottom library directory in the driver notebook or in the scripts as shown in the driver examples. The packages mkbathy and mkbathy_2020 contain the classes and functions which provide the tools necessary to interpolate the EMODNET (2016 or 2020) bathymetry on the

```
#-----------------------------------------------------------------
#/root ----|
#       /data                       /sources
#          |                           |
#     /cmems /emodnet /outs          /bkenergy          /drivers
#        |                              |
#       /ibi  /med   /backsea ...        mkbathy -->ibathy
#                                       currents --> icke
#                                         tands  --> its
#                                         waves  --> iwke
#-----------------------------------------------------------------
    root = '/Users/uranus/workarea/bke'
    bdata = root + '/data' + '/emodnet'
    out = root + '/data' + '/outs'

    # emodnet bathymetry
    bds = ["A2", "A3", "A4", "B2", "B3", "B4", "C2", "C3", "C4", "D3", "D4"]
    nb = [bdata + "/" + bd + ".mnt" for bd in bds]

    # cmems currents ibi
    cdata = root + '/data' + '/cmems' + '/ibi/'
```

Figure 1: Suggested organization of files and data. left: data, right: library

selected model grid.The model grid is extracted directly from the cmems model output files, so the user is not required to explicitly define an area, but he is responsible for mantaining a consistent data pool (see the organisation of resources used in the drivers in fig.1). If in the data directory there are mixed extractions from different models or different subgrids, the evaluation will fail. Each data set pertaining to a certain grid/model should be keep separated. No other file should be present in the directories except for netcdf data files. Clearly exceptions can be controlled by modifying the code, but the cleaner the arrangement, the cleaner the code. Ideally, each chunk of data in the data files used to evaluate the long term statistics should contain the same number of times, but a certain amount of flexibility is allowed, and there can be a finite number of exceptions (as having 12 monthly data files with 31, 30 or 28 block daily data, instead of 12 files of 30 fixed days in a year). The structure of the data pool should be something like shown in fig.1. In order to facilitate the usage of the libraries, a set of drivers is provided in the form of simple scripts and notebooks. Simple python notebooks for the visualization of some of the products are also provided. Seabottom library has been tested with forecast and analysis, daily and monthly CMEMS products from the Mediterranean Sea, Black Sea and Iberian, Biscay and Irish (IBI) Seas.

Interpolation of Bathymetry

The package mkbathy (or, equivalently, mkbathy2020) contains the module 'ibathy' (or ibathy2020) with two main classes (Gridinfo and Sbathy) and several other classes and methods. The class Sbathy is used to provide an interpolated bathymetry upon which the other classes are based. The Class Gridinfo is written to provide an object with attributes defining the properties of the fields data read either from a model output or from the Emodnet bathymetry files. It is used by the Classes Sbathy, Sbenergy, Sbchem, Wbenergy. The Class Sbathy is written to provide an evaluation of bathymetry interpolated on the oceanographic model grid based on EMODNET bathymetry (2016). It is meant as an aid to the evaluation of seabottom kinetic energy or temperature and salinity obtained from CMEMS models with fixed z levels.usage- 1st step evaluate the bathymetry by using an instance of Sbathy taking input file of the oceanographic model describing the grid and a list of names of EMODNETbathymetry files. Not all files belonging to the EMODNET distribution need necessarily to be referenced, but the referenced files must completely cover the model domain. If in doubt,see EMODNET documentation or do reference all files. The program will consider only those that are necessary. Usage:

- $g = Sbathy(filename, 0, *nb)$, or
- $g = Sbathy.fromfile('namefile.nc')$ if a previous instance has produced the interpolated bathymetry, e.g. by using the method to_file as in: $g.to\_file("ibi\_batnn.nc")$.
- $g = Sbathy(filename, 0, *nb, printcheck = 1)$ prints the boundary index check for each eumodnet bathymetry block.

The only difference in usage between the library working with EMODNET 2016 and EMODNET 2020 is in the import of the library. EMODNET 2016 needs:

- from bkenergy.mkbathy.ibathy import Gridinfo, Sbathy while EMODNET 2020 needs:
- from bkenergy.mkbathy_2020.ibathy2020 import Gridinfo, Sbathy

From the coding point of view, Sbathy in ibathy2020 differs from Sbathy in ibathy only for the rule adopted to assign a EMODNET block data to the corresponding model domain area. Technical reference can be found in the Annex "Compiling

oceanographic layers" to the EUSeaMap 2019 - Technical Report
See examples in figs.2,4,8 and 10.

Evaluation of statistics (percentile, mean and stdev) of Sea bottom kinetic energy due to currents

The package currents contains the module 'icke' with the main class Sbenergy and several other methods necessary to access the data files, evaluate the kinetic energy at the bottom due to currents at a fixed depth of 1 m from the bottom and extract the n^{th} percentile. It depends on an instance of Sbathy for the bathymetry interpolated on the currents grid. Users can optionally give the appropriate value or the mean sea density in the area and the desired percentile. Example:
$Sbenergy(gmed, cdata, *args, density = 1036.0, percent = 90)$ The Class is written to provide an evaluation of current kinetic energy at the seabottom by means of self-contained objects. it needs a bathymetry interpolated on the grid and montly/daily data from oceanographic models i.e. for IBI files like for example dataset-ibi-analysis-forecast-phys-005-001-monthly_xxx.nc. All input files must be in the form of netcdf files. the class (Sbathy) is provided in order to interpolate Emodnet bathymetry on the specific model grid selected. Usage:

1. 1^{st} step:
   evaluate the bathymetry by using
   - $g = Sbathy(filename, 0, *nb)$ or,
   - $g = Sbathy.fromfile('namefile.nc')$ (see specific documentation).
2. 2^{nd} step: evaluate nth percentile of bottom energy
   - $h = Sbenergy(g.gmed, cdata, *nc,)$ g.gmed is an instance from emodnet 2016 bathymetry, or:
   - $h = Sbenergy(g.gmed, cdata, *nc, density = 1035., percent = 90)$
     where $percent = nn$ indicates the nn n^{th} percentile to be evalated, and $density = 1035.$ let the user to insert a value of mean density of the area to be used in the evaluation of the bottom sea kinetic enegy due to currents.
3. 3^{rd} step: output n^{th} percentile of seabottom energy

$h.bke\_to\_file('bke\_ibi\_out.nc')$.
For information on the actual algorithm used, Technical reference can be found in the Annex 1 "Compiling oceanographic layers" to the EUSeaMap 2019 - Technical Report. See examples in figs.3,5,8 and 11.

Evaluation of statistics (percentile, mean and stdev) of Sea bottom kinetic energy due to waves

The package waves contains the module 'iwke' with the main class Wbenergy and several other methods necessary to access the data files, evaluate the kinetic energy at the bottom due to waves and extract the n^{th} percentile. Usage:
$Wbenergy(gmed, cdata, *args, density = 1036.0, percent = 90)$ The Class Wbenergy is written to provide an evaluation of waves kinetic energy at the seabottom by means of self-contained objects. it needs a bathymetry interpolated on the grid and montly data from oceanographic models i.e. for med files like dataset-ibi-analysis-forecast-phys-005-001-monthly.nc, All inputs must be netcdf files. A class (Sbathy) is provided in order to interpolate Emodnet bathymetry on the specific model grid selected. Usage

1. 1^{st} step: evaluate the bathymetry by using

   - $g = Sbathy(filename, 0, *nb)$, or
   - $g = Sbathy.fromfile('namefile.nc')$

   (see specific documentation for Sbathy usage)

2. 2^{nd} step:
   evaluate the n^{th} percentile of bottom kinetic energy due to waves as:

   - $h = Wbenergy(g.gmed, wdata, *nw)$, where $g.gmed$ must be generated as an instance of class Sbathy), or

   - $h = Wbenergy(g.gmed, cdata, *nw, density = 1035., percent = 90)$, where $percent = n$ indicates the n^{th} percentile to be evaluated, and $density = 1035.$ lets the user to insert the appropriate value of mean density for the selected area.

3. 3^{rd} step:
   output the n^{th} percentile of seabottom energy as:

$h.wbke\_to\_file('bke\_ibi\_out.nc')$.

For information on the actual algorithm see the Annex 1 "Compiling oceanographic layers" to the EUSeaMap 2019 - Technical Report. See examples in fig.9.

Evaluation of statistics (percentile, mean and stdev) of Sea bottom temperature, salinity, oxygen and chlorophyll

The package mk_bchem contains the module 'bchem' with a main class Sbchem and several other methods necessary to access the data files, evaluate the biochemical parameter at the bottom and extract the n^{th} percentile. Usage: Sbchem(gmed, cdata, *args, parameter=xxxx, percent=50).
The Class Sbchm is written to provide an evaluation of temperature, salinity, o_2, chlorophyll at the seabottom by means of self-contained objects. It needs a bathymetry interpolated on the grid and montlydaily data from oceanographic models i.e. for IBI files like for example dataset-ibi-analysis-forecast-phys-005-001-monthly_xx.nc. All inputs must be netcdf files. A class (Sbathy) is provided in order to interpolate Emodnet bathymetry on the specific model grid selected. Usage:

1. 1^{st} step evaluate the bathymetry by using

   - $gts = Sbathy(filename\_ts, 0, *nb)$, or

   - $gts = Sbathy.fromfile('namefile.nc')$. (see specific documentation for the usage of Sbathy).

2. 2^{nd} step evaluate n^{th} percentile of the parameter xxxx

   - $hts = Sbchem(gts.gmed, tsdata, *nts, parameter = xxxx)$ (gts.gmed must be generated as an instance gts of class Sbathy),or
   - $hts = Sbchem(g.gmed, cdata, *nc, parameter = xxxx., percent = 90)$, where xxxx is 'btemperature','salinity','oxygen' or 'chlorophyll' and percent=nn indicates the n^{th} percentile to be evalated. Default parameter xxxx is 'salinity', default percent is 50. Mean and std are evaluated by default.

3. 3^{rd} step - output nth percentile of seabottom energy $hts.bts\_to\_file('bchem\_ibi\_out.nc')$.

For information on the actual algorithm used, see the Annex 1 ”Compiling oceanographic layers” to the EUSeaMap 2019 - Technical Report. See examples in fig.6.

List of Figures

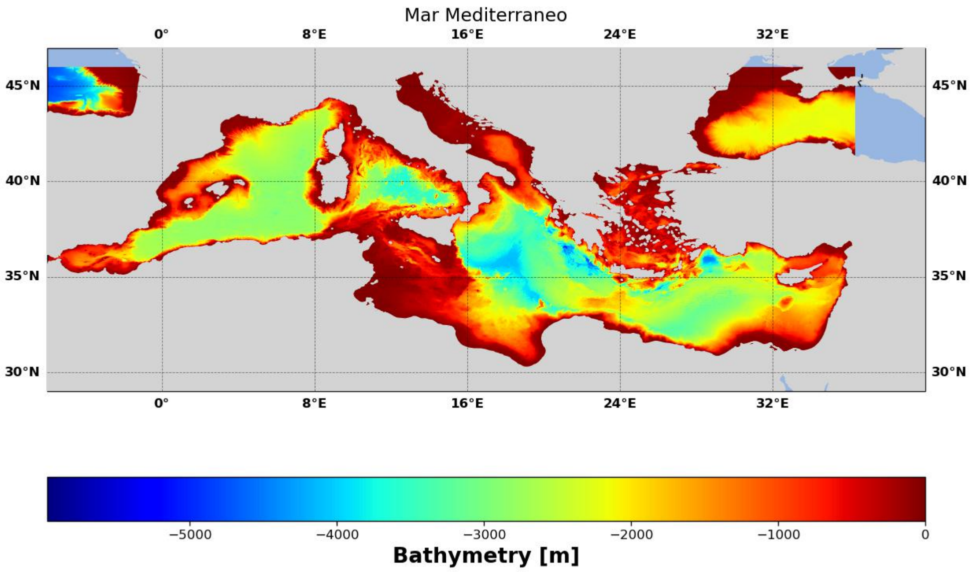

Figure 2: Emodnet Bathymetry (2016) interpolated on CMEMS INGV–RFVL-MFSeas3-MEDATL-b20160308_an-sv04.00 grid

WWW References

EUSeaMap: `https://www.emodnet-seabedhabitats.eu`

ACR Acronyms

CMEMS	Copernicus Marine Service
EMODnet	European Marine Observation and Data Network
FORTRAN	FORmula TRANslator
ISPRA	Istituto Superiore per la Protezione e la Ricerca Ambientale
NetCDF	Network Common Data Form
WAM	Wave Model
WWIII	Wave Watch III

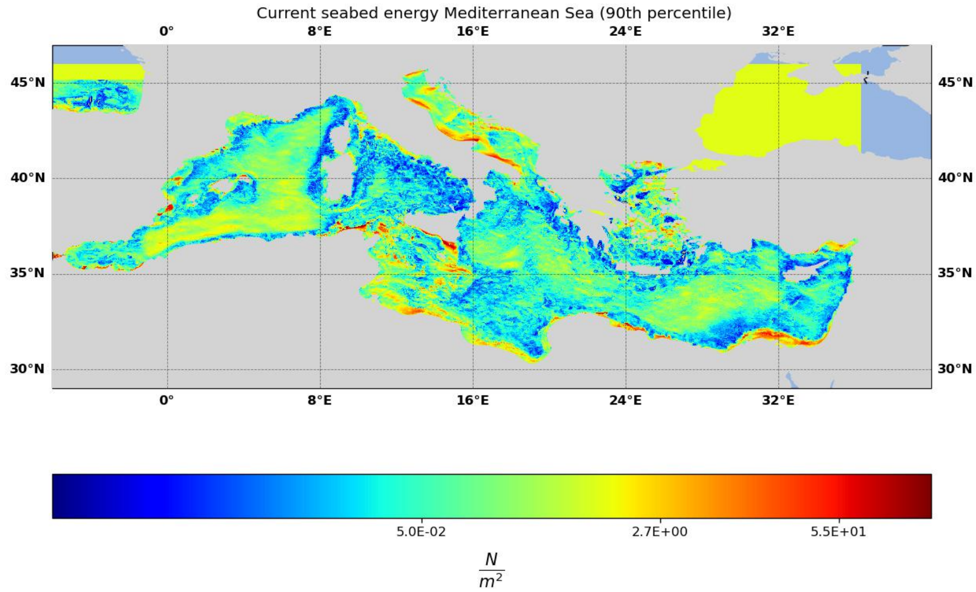

Figure 3: Emodnet Bathymetry (2016) interpolated on CMEMS INGV–RFVL-MFSeas3-MEDATL-b20160308_an-sv04.00 grid

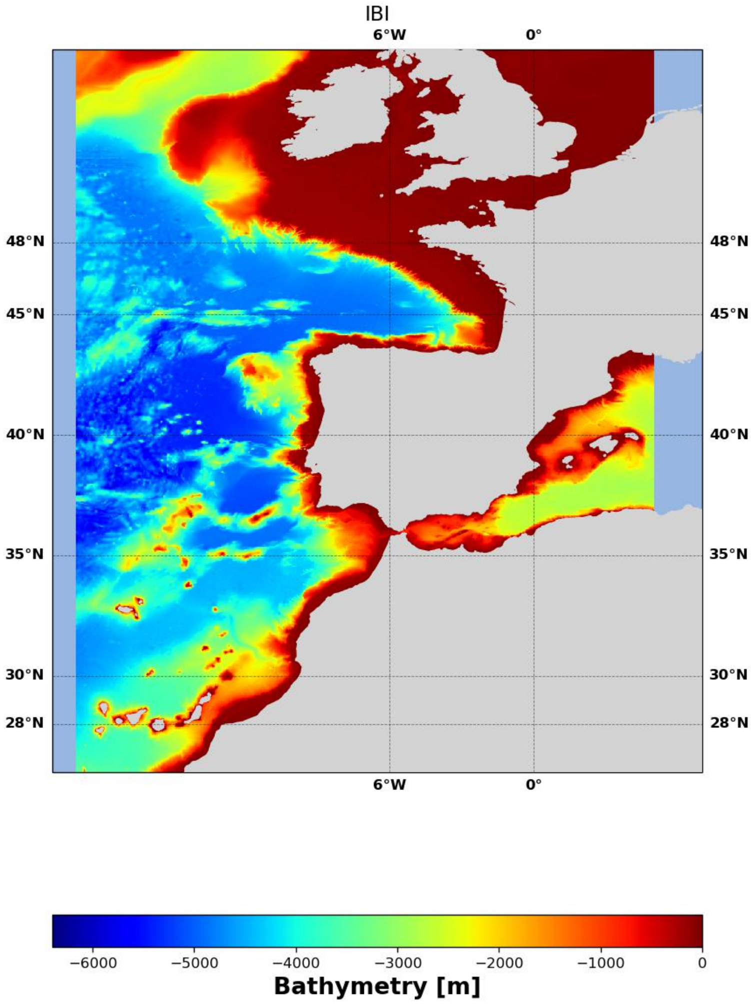

Figure 4: Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid

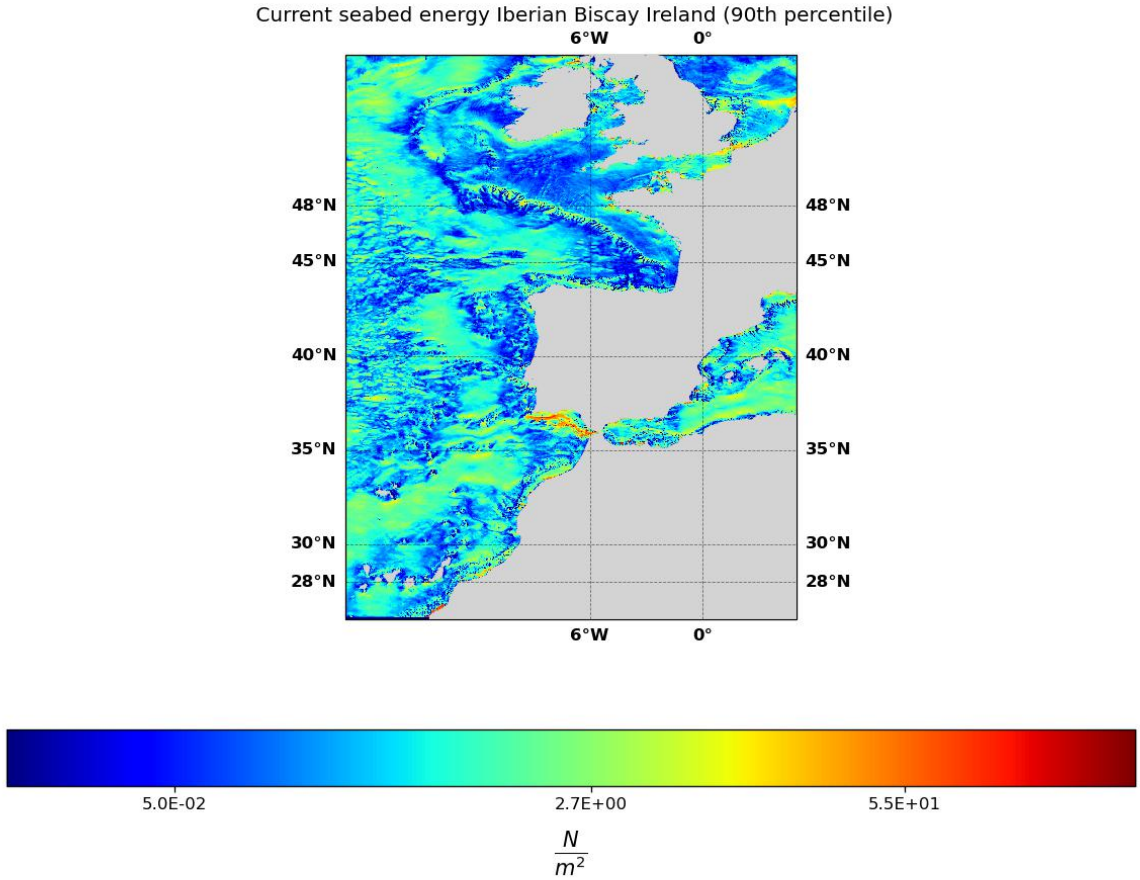

Figure 5: Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid

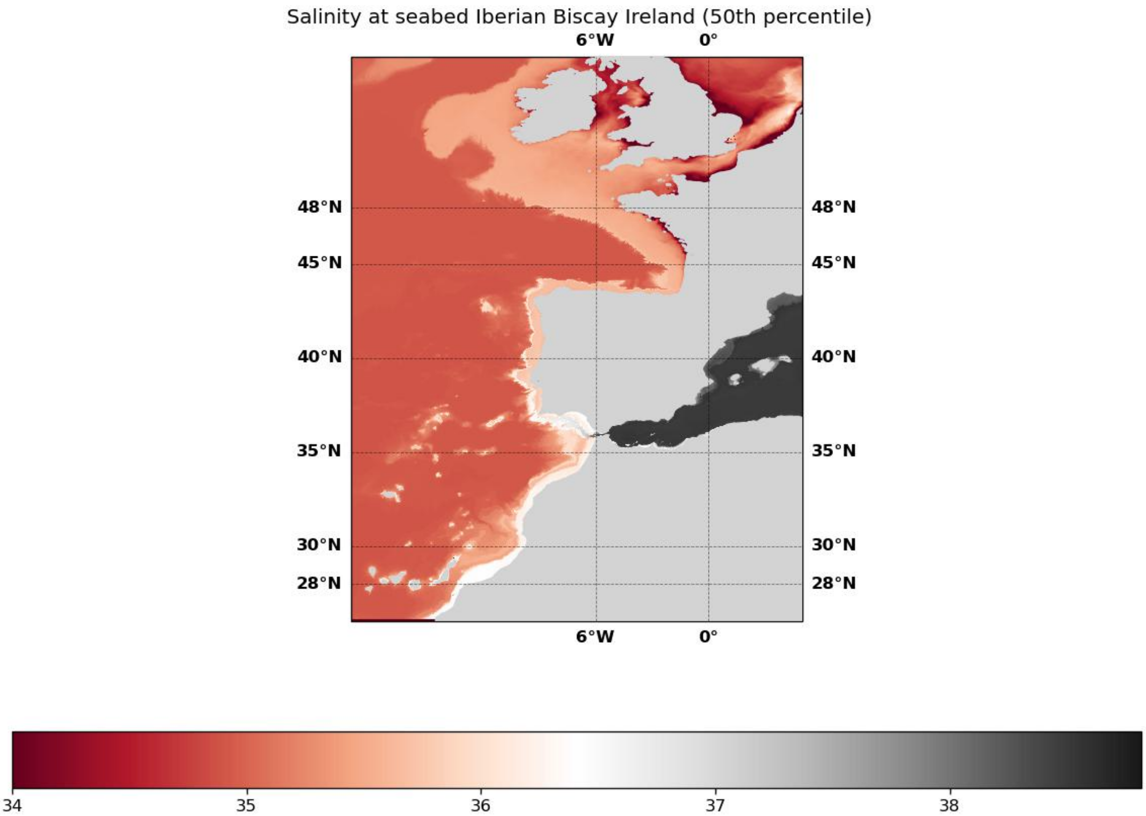

Figure 6: Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid

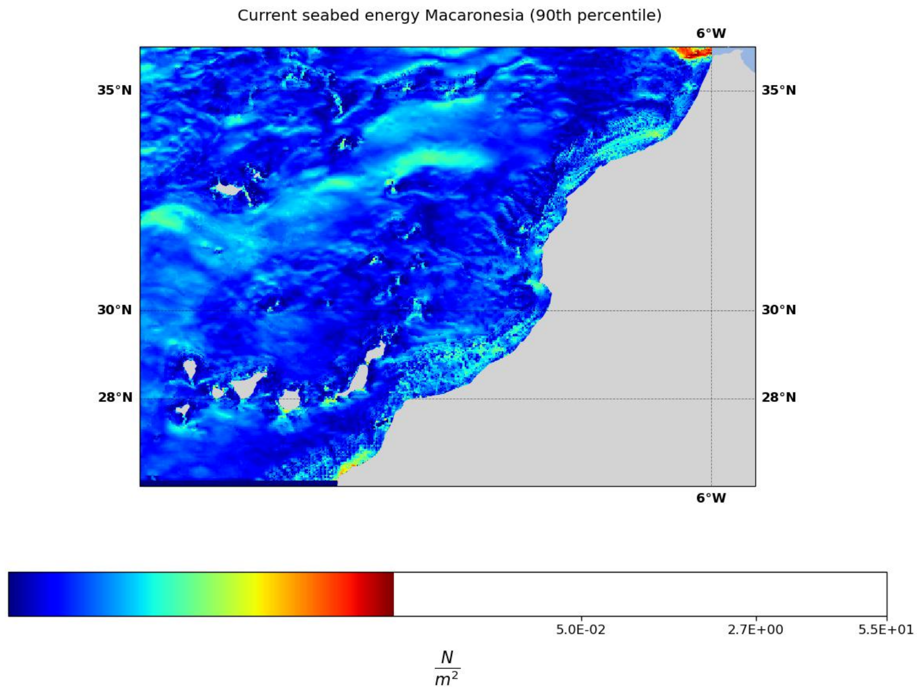

Figure 7: Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid

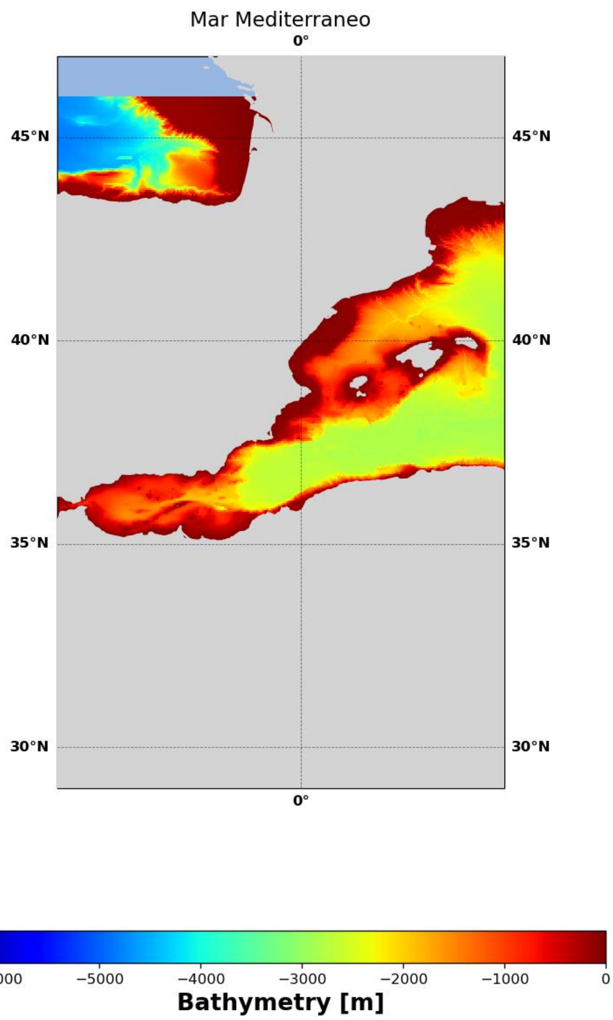

Figure 8: 'Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid'

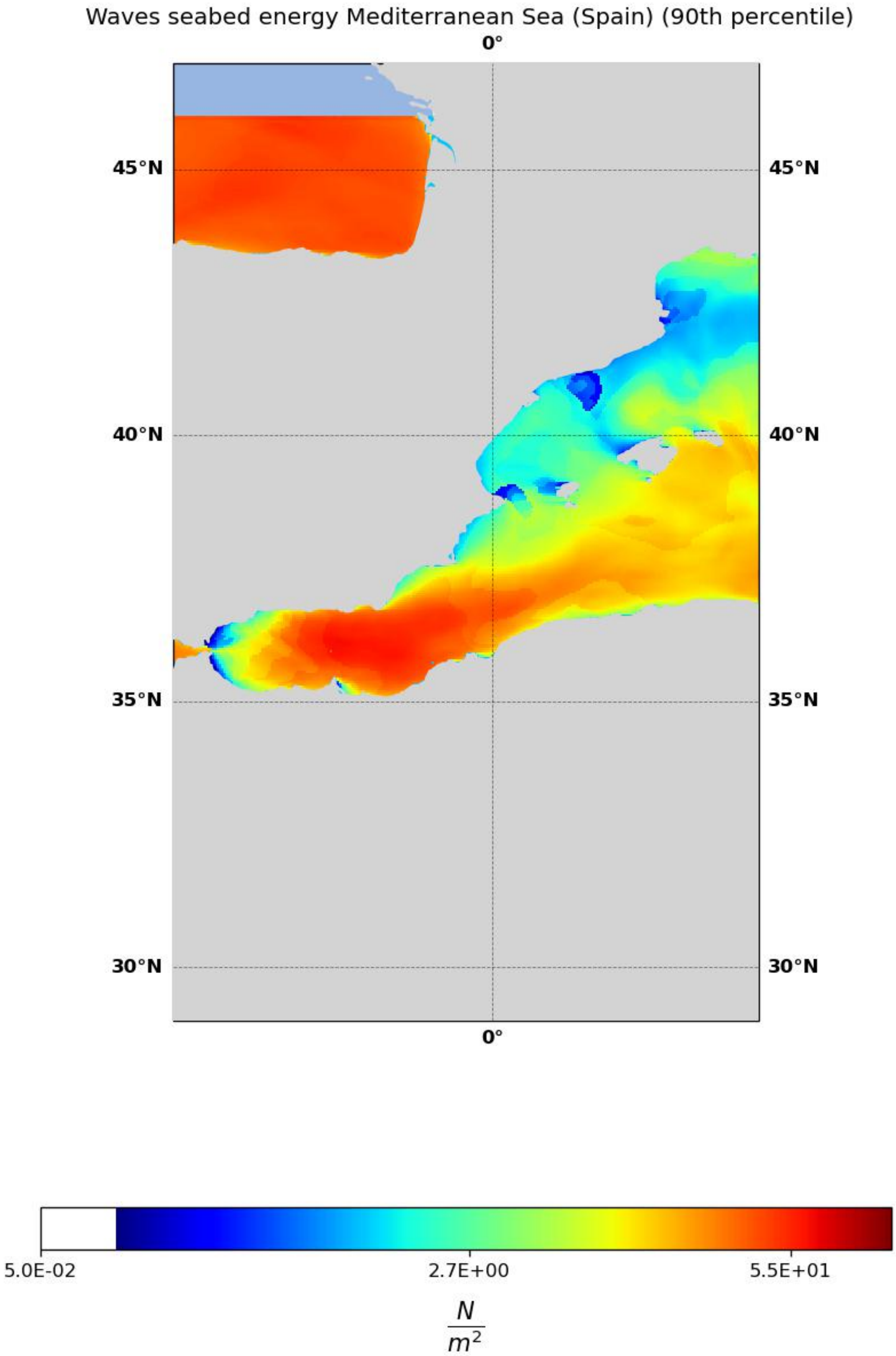

Figure 9: 'Emodnet Bathymetry (2016) interpolated on CMEMS dataset-ibi-analysis-forecast-phys-005-001-monthly grid'

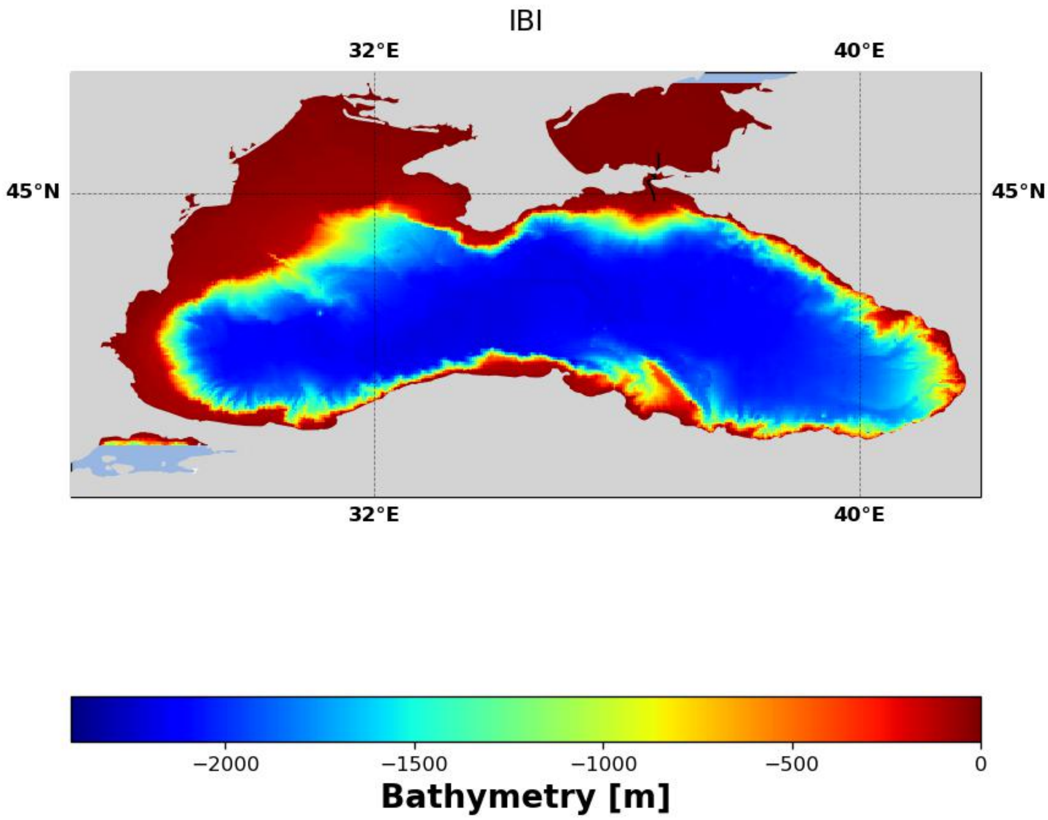

Figure 10: Emodnet Bathymetry (2016) interpolated on CMEMS sv03-bs-cmcc-cur-an-fc-d grid

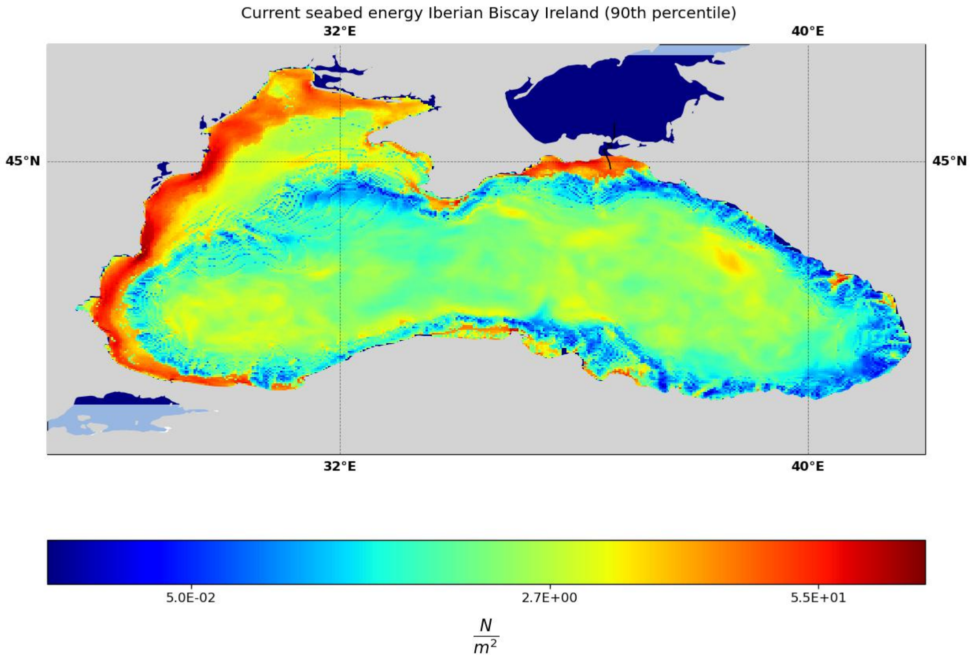

Figure 11: Emodnet Bathymetry (2016) interpolated on CMEMS sv03-bs-cmcc-cur-an-fc-d grid