



Blue-Cloud

Piloting innovative services for Marine Research & the Blue Economy

D3.4 - Blue Cloud Demonstrator Users Handbook V2

Work Package	WP3, Blue Cloud Pilot Demonstrators
Lead Partner	IFREMER
Lead Author (Org)	Dominique OBATON, Gilbert MAUDIRE (IFREMER)
Contributing Author(s)	<p>Demo 1: CABRERA Patricia (VLIZ), SCHEPERS Lennert (VLIZ), PINT Steven (VLIZ), PANNIMPULLATH REMANAN Renosh (SU), SAUZEDE Raphaëlle (SU), UITZ Julia (SU), BARTH Alexander (GHER/ULiège), TROUPIN Charles (GHER/ULiège).</p> <p>Demo 2: DEBELJAK Pavla (SU), IRISSON Jean-Olivier (SU), PESANT Stéphane (EMBL), SCHICKELE Alexandre (SU)</p> <p>Demo 3: DRUDI Massimiliano (CMCC), LECCI Rita (CMCC), PALERMO Francesco (CMCC), MARIANI Antonio (CMCC), BALEM Kevin (IFREMER), GARCIA JUAN Andrea (IFREMER), BACHELOT Loic (IFREMER), NOTEBOOM Jan Willem (KNMI), CASTAÑO-PRIMO Rocío (UiB), PFEIL Benjamin (UiB), JONES Steve (UIB)</p> <p>Demo 4: BARDE Julien (IRD), BLONDEL Emmanuel (FAO), ELLENBROEK Anton (FAO), GENTILE Aureliano (FAO), MARKETAKIS Yannis (FORTH)</p>
Reviewers	Pasquale Pagano (CNR), Dick Schaap (MARIS), Sara Pittonet Gaiarin (Trust-IT)
Due Date	December 2021 (M27)
Submission Date	2 February 2022
Version	2.0 DRAFT NOT YET APPROVED BY THE EUROPEAN COMMISSION

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)



DISCLAIMER

“Blue-Cloud, Piloting Innovative services for Marine Research & the Blue Economy” has received funding from the European Union's Horizon programme call BG-07-2019-2020, topic: [A] 2019 - Blue Cloud services, Grant Agreement n.862409.

This document contains information on Blue-Cloud core activities. Any reference to content in this document should clearly indicate the authors, source, organisation, and publication date.

The document has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the Blue-Cloud Consortium, and it cannot be considered to reflect the views of the European Commission. The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur because of using its content.

COPYRIGHT NOTICE



This work by Parties of the Blue-Cloud Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). “Blue-Cloud, Piloting Innovative services for Marine Research & the Blue Economy” has received funding from the European Union's Horizon programme call BG-07-2019-2020, topic: [A] 2019 - Blue Cloud services, Grant Agreement n.862409.

VERSIONING AND CONTRIBUTION HISTORY

Version	Date	Authors	Notes
1.0	17.12.2021	Cécile Nys (OceanScope/IFREMER)	First full draft version
1.1	4 January 2022	Dominique Obaton (IFREMER)	Complete version for internal review
1.2	12 January 2022	Dick M.A. Schaap (MARIS) ; Sara Pittonet Gaïarin (Trust-IT)	Internal review
2.0	27 January 2022	Dominique Obaton (IFREMER)	Final version for submission addressing reviewers' comments

Table of Contents

Executive summary	8
1 Introduction	9
2 Registration to the Blue Cloud VRE	11
3 Demonstrator # 1 – Zoo- and Phytoplankton Essential Ocean Variable products	12
3.1 Zooplankton EOVS.....	12
3.1.1 Short description of the service	12
3.1.2 Targeted users	13
3.1.3 Step by step guideline to use the service	13
3.1.4 Data sources	16
3.1.5 Scientific references.....	16
3.2 Phytoplankton EOVS	16
3.2.1 Short description of the service	17
3.2.2 Targeted users	18
3.2.3 Step by step guideline to use the service	18
3.2.4 Data sources	21
3.2.5 Scientific references.....	22
3.3 Modelling phyto & zoo –plankton interactions.....	22
3.3.1 Short description of the service	22
3.3.2 Targeted users	23
3.3.3 Step by step guideline to use the service	23
3.3.4 Data sources	25
3.3.5 Scientific references.....	25
4 Demonstrator # 2 – Plankton Genomics.....	26
4.1 Exploring genetic data & identifying clusters containing unknown genes	26
4.1.1 Short description of the service	26
4.1.2 Targeted users	26
4.1.3 Step by step guideline to use the service	26
4.1.4 Data sources	30
4.1.5 Scientific references.....	30
4.2 Mapping the geographical distribution of plankton functional gene clusters using habitat models.....	31
4.2.1 Short description of the service	31
4.2.2 Targeted users	31
4.2.3 Step by step guideline to use the service	32
4.2.4 Data sources	32
4.2.5 Scientific references.....	33
5 Demonstrator # 3 – Marine Environmental Indicators	34
5.1 Targeted users	34
5.2 Marine Environment Indication (MEI) generator	35
5.2.1 Short description of the service	35
5.2.2 Step by step guideline to use the service	36
5.2.3 Data sources	38

5.2.4	Scientific references.....	38
5.3	Ocean pattern indicator.....	39
5.3.1	Short description of the service	39
5.3.2	Targeted users	39
5.3.3	Step by step guideline to use the service	40
5.3.4	Data sources	46
5.3.5	Scientific references.....	46
5.4	Ocean regime indicator.....	47
5.4.1	Short description of the service	47
5.4.2	Targeted users	47
5.4.3	Step by step guideline to use the service	47
5.4.4	Data sources	56
5.4.5	Scientific references.....	56
5.5	Storm severity index	57
5.5.1	Short description of the service	57
5.5.2	Targeted users	57
5.5.3	Step by step guideline to use the service	57
5.5.4	Data sources	62
5.5.5	Scientific references.....	62
5.6	Simple access to carbon data.....	62
5.6.1	Short description of the service	62
5.6.2	Step by step guideline to use the service	63
5.6.3	Data sources	64
5.6.4	Scientific references.....	65
6	Demonstrator # 4 – Fish, a matter of scales	66
6.1	Fisheries atlas	66
6.1.1	Short description of the service	66
6.1.2	Targeted users	66
6.1.3	Step by step guideline to use the service	67
6.1.4	Data sources	70
6.1.5	Scientific references.....	71
6.2	Global record of stocks and fisheries (GRSF).....	71
6.2.1	Short description of the service	71
6.2.2	Targeted users	72
6.2.3	Step by step guideline to use the service	72
6.2.4	Data sources	75
6.2.5	Scientific references.....	76
7	Common useful generic IT services	77
7.1	Blue Cloud Data Discovery & Access Service.....	77
7.1.1	Short description of the service	77
7.1.2	References	79
7.2	Blue Cloud Notebook for CMEMS WEKEO data access.....	79
7.2.1	Short description of the service	79
7.2.2	References	80
8	Conclusions	81

List of figures

Figure 1 Blue-Cloud Gateway registration page	11
Figure 2 Jupyter interface with the BlueCloud Zooplankton Demonstrator, showing how to save a notebook.....	14
Figure 3 Spatial distribution of <i>Metridia lucens</i> for the year 2004 (top panels) and year 2013 (lower panels). The binned observation represents the average of the reported value for the corresponding grid cell and year.....	15
Figure 4 Left panel represents the geographical locations of Chla (BGC-Argo) profiles used for the present study. Right panel represents a density scatter plot for the validation of the model.....	18
Figure 5 Global Chla concentration at 0m depth for January 2018.	20
Figure 6 Global PFT at 0m depth for January 2018.	21
Figure 7 Average monthly relative contributions for each limitation factor in the growth of phytoplankton for the nearshore region.	24
Figure 8 Plankton size fractions from Sunagawa et al. 2020	27
Figure 9 Example output Notebook 1.1. from R (leaflet package)	28
Figure 10 Sequence similarity network analysis based on Forster & Bittner 2015.	29
Figure 11 ‘Generate new output’ User Interface.....	37
Figure 12 ‘My data’ – User Interface (UI).	37
Figure 13 ‘Show result’ in the ‘My data’ UI.	38
Figure 14 ‘Ocean patterns indicator’ workflow	40
Figure 15 Development notebook – Model parameters.....	41
Figure 16 Development notebook – Load training dataset.	42
Figure 17 . Development notebook – Create and train model.	42
Figure 18 Development notebook – Development plots.....	43
Figure 19 Development notebook – Refit and save model.....	43
Figure 20 Predict & Plot notebook – Load model and dataset.	44
Figure 21 Predict & Plot notebook – Predict labels.	45
Figure 22 Predict & Plot notebook – Plot results.....	45
Figure 23 Predict & Plot notebook – Save data.	46
Figure 24 ‘Ocean regimes indicator’ workflow	47
Figure 25 Development notebook – Model parameters.....	49
Figure 26 Development notebook – Load training dataset.	49
Figure 27 Development notebook – Preprocessing.....	50
Figure 28 Development notebook – Create model and train	51
Figure 29 Development notebook – Development plots.....	52
Figure 30 Development notebook – Refit and save model.....	53
Figure 31 Predict & Plot notebook – Load model and dataset.	54
Figure 32 Predict & Plot notebook – Predict labels.	55
Figure 33 Predict & Plot notebook – Plot results.....	55
Figure 34 Predict & Plot notebook – Save data.	56

Figure 35 – Storm Severity Index (SSI) user input.....	59
Figure 36 Calculation Input parameters.	60
Figure 37 Calculation in progress	60
Figure 38 User plotting input	60
Figure 39 Map plots and time series plots	61
Figure 40 Geographical distribution (left) and longitude-depth scatterplot of pH data (right) from two sources and through the ERDDAP servers used in the notebooks.....	64
Figure 41 A web snapshot of the fisheries atlas	66
Figure 42 User portal serving fisheries data management communities.....	67
Figure 43 Example of user selection panel on a dataset in the Fisheries Atlas.	68
Figure 44 User visualisation option (2D/3D) in the Fisheries Atlas.	69
Figure 45 Example of statistics in the Fisheries Atlas.	70
Figure 46 GRSF data preparation portal for registered users	72
Figure 47 GRSF Public Map interface	73
Figure 48 GRSF navigation bar	73
Figure 49 The GRSF VRE UI for the Global Record of Stocks and Fisheries (GRSF).	74
Figure 50 GRSF Interactive Map viewer.	74
Figure 51 a) GRSF Record editing environment; b) GRSF API List.	75
Figure 52 : HDA API steps	79

List of Tables

Table 1 Zooplankton EOVs data sources	16
Table 2 Phytoplankton EOVs data sources	22
Table 3 Modelling phyto & zoo –plankton interactions data sources.....	25
Table 4 Demo 2 - Exploring genetic data – Data sources.....	30
Table 5 Demo 2 - Mapping the geographical distribution of plankton - data sources	32
Table 6 Marine Environmental Indication (MEI) generator data sources	38
Table 7 Ocean Pattern Indicator data sources.....	46
Table 8 Ocean Regime indicator - data sources.....	56
Table 11 Fisheries Atlas data sources	71

Glossary

Achronym	Definition
ABAC	Attribute-based access control
API	Application Programming Interface
BPNS	Belgium part of the North Sea
CMEMS	Copernicus Marine Environment Monitoring Service
DDAS	Data Discovery & Access Service
DIAS	Data and Information Access Service (funded by EC COPERNICUS programme)
EcoTaxa	Web application dedicated to the visual exploration and the taxonomic annotation of images focused on planktonic biodiversity
EOV	Essential Ocean Variables
EOSC	European Open Science Cloud
GAM	Generalized Additive Models
GRSF	Global record of Stocks and Fisheries
HDA	Harmonised Data Access
KEGG	Kyoto Encyclopedia of Genes and Genomes
IAM	Access Management Service
IdM	Identity Management Service
MATOU	Marine Atlas of Tara Ocean Unigenes
MEI	Marine Environment Indication
MLP	Multi-Layer Perceptron
NPZ	Nutrient-Phytoplankton-Zooplankton
NPZD	Nutrient-Phytoplankton-Zooplankton-Detritus
PAR	photosynthetically available radiation
PFT	Phytoplankton Functional Types
RMSE	Root Mean Square Error
SDG	Sustainable Development Goal
SDI	Spatial Data Infrastructure
SLA	sea level anomaly
SNL	Social Networking Library
SSI	Storm severity index
UI	User Interface
UMA	User-Managed Access
VLab	Virtual Laboratory
VRE	Virtual Research Environment
WEkEO DIAS	WEkEO is one of the 5 Copernicus DIAS, bringing in the CMEMS, C3S and CAMS

Executive summary

This handbook describes the 12 thematic IT services developed and made available for users as part of the Blue Cloud platform and tested in real scenarios via the demonstrators, plus 2 additional generic IT services available directly from the platform. Users are able to discover and understand the services and then to re-run each of them as proposed by the developers.

To do this, they will need first to register as a Blue Cloud user. This will allow them to access to the Blue Cloud Virtual Research Environment (VRE) and the Virtual Laboratories (VLab) where the service they look for is provided. This is true except for 1 Vlab for which additional credentials provided by the scientists' developers are requested. Points of contact are also given in case the users have difficulties or questions regarding the service they want to try-out and use.

Some services give the possibility to the users to select other input data or to “play” with several options.

Services described are in a prototype stage (TRL6 – 7) and may be enhanced following feedback from the users. Therefore, please, as user, provide feedback to us!

This handbook will be corrected and updated during the year. It will be also improved with additional services, thematic and generic, to provide the most possible complete and up-to-date overview of the Blue Cloud services along the project life.

1 Introduction

In this version, 12 thematic IT services and 2 generic ones are described and proposed to be used and operated by users. The thematic services offer a large span of marine environmental subjects and the generic ones a good support to exploit and use them. We, at this stage, have a good showcase of services for web-based science, including the Blue Cloud Virtual Research Environment (VRE) and multiple Virtual Laboratories (Vlab) as well as a Data Discovery & Access Service (DD&AS) to facilitate discovery and retrieval of data sets and products in stand-alone mode.

Technical and scientific developments are done in parallel during the project although the second ones need the first ones to be ready, well tested, and available for operating by users. This leads to transitional workarounds and an “agile” process necessary to show the potential of Blue Cloud at this stage of the project.

Research and scientific developments are innovative and very often generic enough to support many users in their own developments and help them to save time. Several developments use AI and neural networks; others provide graphical interfaces to visualise large amount of data and to derive from them indicators and statistics.

Targeted users are mostly expert users, while some services target a larger public although intermediate users are required to present the services and results to end users, such as policy and decision makers.

For demonstrator #1 – Zoo- and Phytoplankton Essential Ocean Variable products, 3 services are provided:

- The *zooplankton essential ocean variable* service that provides interpolated maps of zooplankton abundances calculated from the minimization of a cost function using as input data in situ observations.
- The *phytoplankton essential ocean variable* service that calculates 3D distribution phytoplankton biomass and diversity proxy using a machine learning technique and in situ and satellite observations as input data.
- The *modelling phyto & zoo -plankton interactions* service that quantifies the relative contributions of the bottom-up and top-down drivers in phytoplankton dynamics thanks to a numerical NPZ (Nutrients-Phytoplankton-Zooplankton) model.

For demonstrator #2 – Plankton Genomics, 2 linked services are provided:

- The first one, *exploring genetic data and identifying clusters containing unknown genes*, enables the discovery of the unknown marine plankton genes (from annotation files) of the Tara Ocean expedition dataset and the building of gene clusters by similarities of sequences and larger metabolic pathways.
- The second one, *mapping the geographical distribution of plankton functional gene clusters using habitat models*, considers the previous result as input data and proposes tools, based on a machine learning method, to explore the relationship between the abundance of plankton genes and the environmental context.

For demonstrator #3 – Marine Environmental Indicators, 5 services are described:

- The *Marine environmental indicator generator* is a web graphical interface for the creation and the display of added-value environmental data (e.g. average over time) built upon generic ones.
- The *ocean pattern indicator* service allows building clusters of profiles, for any oceanographic variables (physical, biogeochemical; gridded, unstructured), according to their vertical structures using a machine learning approach.
- The *ocean regime indicator* service is based on the same previous clustering method applied to a dataset of ocean time series.
- The *storm severity index* service calculates maps and time series of exceptional atmospheric wind circumstances for a given area and period with respect to thresholds.
- The *simple access to carbon data* service shows how to search and retrieve subsets of pH data through different sources (thanks to the widely used ERDDAP servers) and merge them into a single dataset to make visualisations.

For demonstrator #4 – Fish, a matter of scales, 2 services are provided:

- The first one is an online *fisheries atlas* of EU waters and beyond that proposes catch location specific information scalable to offer various indicators, statistics and interactive maps.
- The second one, the *global record of stocks and fisheries*, is a data portal for fisheries with focus on assessment status and management of natural living resources. This service is currently only open for selected users.

For demonstrator #5 – Aquaculture monitor, the services are not ready yet. They will be added to the handbook as soon as there are finalised.

The handbook describes each of the above services; this is done by demonstrators and by services. After the name of authors and maintainer of the service, a short description of the service is given followed by the users targeted. Then a step by step description of the service is detailed with references (and links) to the Vlab allowing users to operate the services provided. This is followed by the list of data used for the service, of which a number could also be queried and accessed through the Blue Cloud Data Access Service (DD&AS): <https://data.blue-cloud.org/>. Finally, scientific and technical references are listed.

Some of the services provide uncertainties of the results, although there is still some work to do as services are not mature enough yet and need improvements and consolidation before working on this subject. However, information on the uncertainties of the results will increase with time to answer to a recurring request from users.

Concerning the general IT services, the first one explains how to use the DD&AS and lists the Blue Data Infrastructures it contains and gives access to. The second one details how to use the Harmonised Data Access (HDA), unique IT interface, to all Copernicus datasets through the e-infrastructure WekEO DIAS.

2 Registration to the Blue Cloud VRE

To be able to access Blue Cloud services, the interested user needs first to register to Blue Cloud platform by creating an account on the Blue-Cloud gateway at <https://blue-cloud.d4science.org/>:

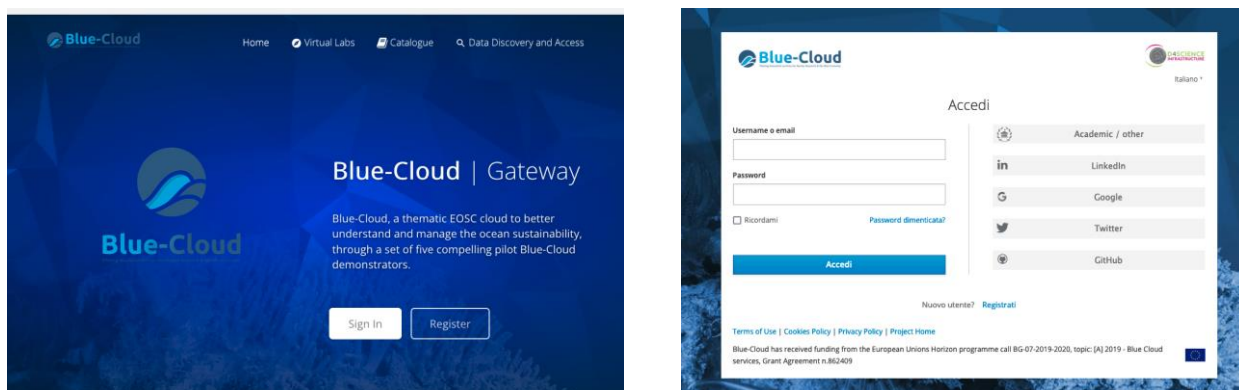


Figure 1 Blue-Cloud Gateway registration page

The registration process ends with a pop-up advertising on “email verification” sent by the Blue Cloud management team. After the confirmation of registration, the user can start using the Blue Cloud platform. Information on how to register on Blue-Cloud is also available here <https://youtu.be/qCa8zfyowF8>

The credentials allow accessing the Virtual Research Environment (VRE) as well as 3 demonstrators or virtual laboratories (VLabs):

- Demonstrator #1: Zoo and phytoplankton EOVS products and its 3 services
- Demonstrator #2: Plankton genomics and its 2 services
- Demonstrator #3: Marine environmental indicators and its 5 services

To access demonstrator #4, the VLab “Fish a matter of scale”, the user needs to ask additional access at the VLab level. Demonstrator #5 “aquaculture monitor” is still under development, not described yet in the handbook and not currently available.

3 Demonstrator # 1 – Zoo- and Phytoplankton

Essential Ocean Variable products

The Zoo- and Phytoplankton Essential Ocean Variable (EOV) demonstrator provides a **description** of the **current state** of the **plankton communities** and **forecasts** their evolution, representing valuable information for the modelling, assessment and management of the marine ecosystems.

For example, the service provided is of interest for ***fundamental research*** (e.g. researchers and consultants from environmental agencies) contributing to the understanding of the environmental conditions and top-down factors at different scales of observations (e.g. regional/global, seasonal and time series). This knowledge will help the ***marine policy officers*** to address threats such as food insecurity, as foreseen under the EU Biodiversity Strategy for 2030. Moreover, ***fisheries advisory organisations*** can use these plankton products to study the availability of food resources for fish stocks.

It consists of 3 different services that are described in the next sections with a detailed guideline on how to use them:

1. Zooplankton EOVs
2. Phytoplankton EOVs
3. Modelling phyto- and zooplankton interactions

To run these services, the registered user needs to go to https://blue-cloud.d4science.org/web/zoo-phytoplankton_eov

3.1 Zooplankton EOVs

Authors: Alexander Barth and Charles Troupin, GHER/ULiège (Belgium).

Corresponding author/maintainer: abarth@uliege.be

3.1.1 Short description of the service

The zooplankton EOV demonstrator provides a methodology to generate zooplankton products based on in situ observations of abundance of six zooplankton species in a region encompassing the North East Atlantic.

The service offered is a complete workflow using the DIVAnd software tool (Data Interpolating Variational Analysis in n dimensions) to create interpolated maps of zooplankton abundances. DIVAnd has been designed to interpolate sparse, in situ measurements onto a regular grid in an optimal way, considering constraints such as the presence of obstacles (coastlines, islands) or currents. The method is based on the numerical implementation of the Variational Inverse Model (VIM), which consists of a minimization of a cost function, allowing the choice of the analysed field fitting at best the data sets (Barth et al., 2014).

The service is provided as a set of Jupyter notebooks that describe the full procedure to create the final, gridded products by (1) data reading; (2) choice of analysis parameters; (3) spatial interpolation; (4) creation of plots; and (5) writing of a netCDF file storing the results.

3.1.2 Targeted users

The specific target users of this service are **researchers and modelers** who wish to reconstruct a continuous field from discrete or heterogeneous measurements. Results can be used by EOVs experts that wish to evaluate the state of primary and secondary consumers in the marine ecosystem.

3.1.3 Step by step guideline to use the service

The notebook files are available for users in the shared workspace under: workspace/VREFolders/Zoo-Phytoplankton_EOV/DIVAndNN/bluecloud-plankton-master/src. This directory contains the file `DIVAndNN_analysis.ipynb` for the analysis (including data reading, defining the analysis parameter and write the results in NetCDF files) and `DIVAndNN_plot_res.ipynb` for plotting the results (including preparing an animation over time). Note that the files in the shared workspace are read-only and a user should copy them in their home directory before running or modifying them.

Steps to run the notebook:

1. Navigate to Zoo-Phytoplankton_EOV/DIVAndNN/bluecloud-plankton-master/src
2. Open the notebook `DIVAndNN_analysis.ipynb` and save it to your home directory using File -> Save Notebooks As and just enter the name of the notebook without any leading file path (Figure 2). One should not attempt to run the notebooks from the shared workspace.
3. Select Run -> Run All Cells in the notebook interface to make the analysis
4. For plotting, open the notebook `DIVAndNN_plot_res.ipynb`. Likewise copy also the notebook `DIVAndNN_plot_res.ipynb` to your home directory.
5. Select Run -> Run All Cells in the notebook interface to make the visualizations

The notebook can make a single 2D analysis (longitude, latitude) using all years combined or a yearly 3D analysis (longitude, latitude, time) by setting the variable n dimensions to 2 or 3 (and defining the year range in the variable years). However, a 3D analysis would require much more memory (about 32 GB of memory for about 28 years).

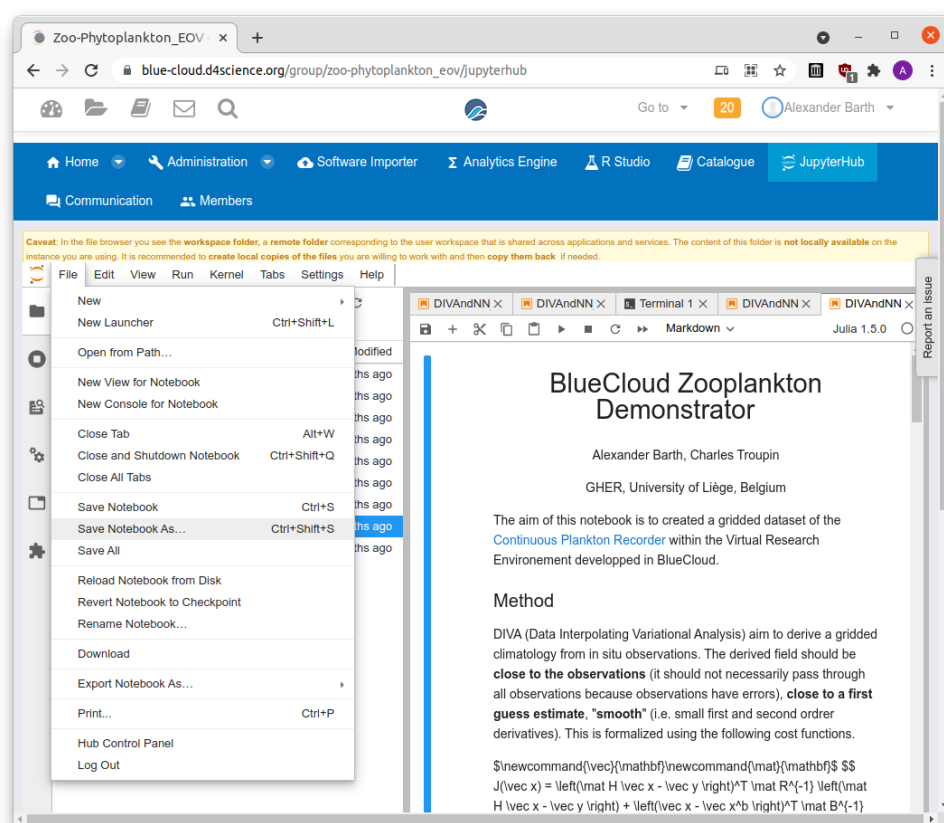
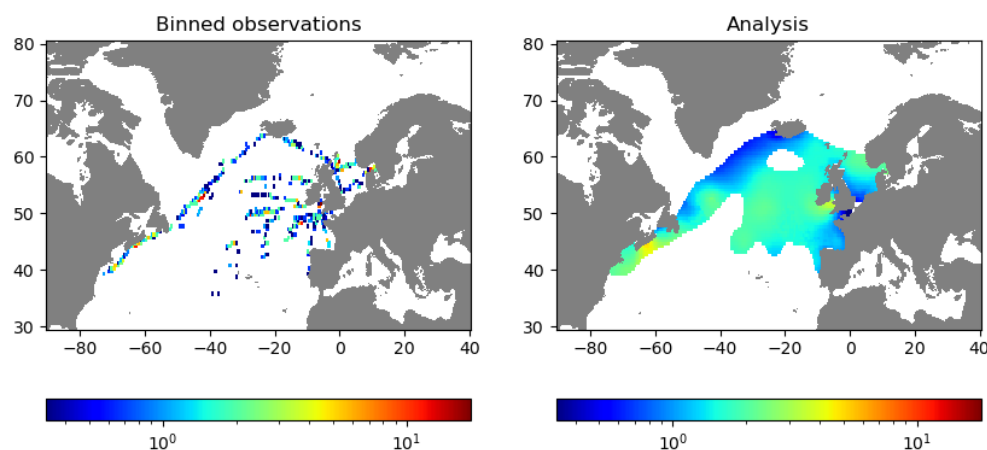


Figure 2 Jupyter interface with the BlueCloud Zooplankton Demonstrator, showing how to save a notebook

Finally, the results will be placed in the folder “BlueCloud-data” in the user’s home directory. Note that the home directory in BlueCloud is a temporary working space. The notebook, scripts and results should be downloaded. Figure 3 shows, as an example of one of the results, the increase of *Metridia lucens* which is particularly visible at the west of Ireland. The strong increase was consistent over several years in the analysis and in the binned observations.

Metridia lucens 2004



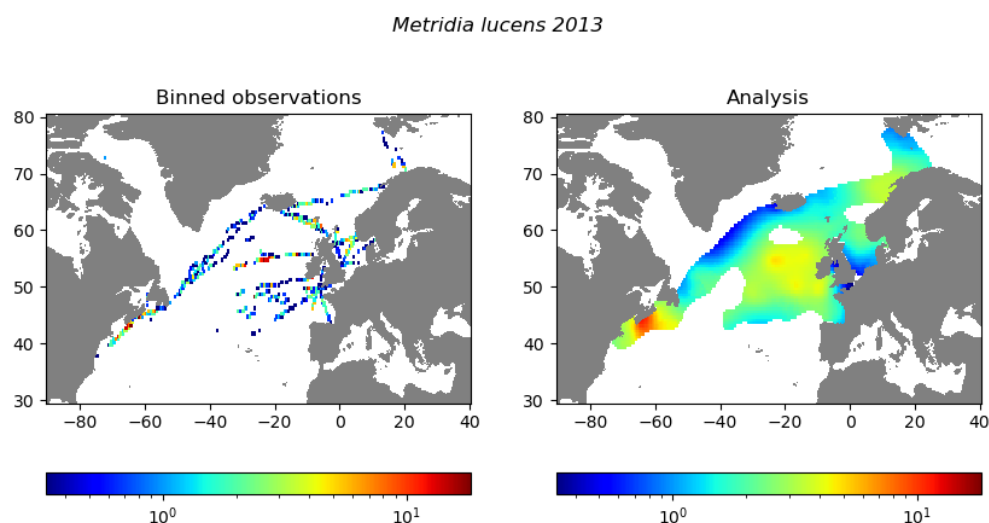


Figure 3 Spatial distribution of *Metridia lucens* for the year 2004 (top panels) and year 2013 (lower panels). The binned observation represents the average of the reported value for the corresponding grid cell and year

In addition to the results, the uncertainty of the output is captured by the relative error (stored in the output netCDF files) and the validation statistic (JSON files) that is the mean squared error between the analysis and the data set aside for validation. The validation statistics represents the accuracy of the results on average in absolute terms while the relative error indicates how much the analysis is influenced by observations close-by.

If a user would like to use these notebooks with other data, keep in mind following points:

The input data is a CSV file where strings are delimited by quotes. The columns with the following headers are used (with descriptive names):

- minimumDepthInMeters
- maximumDepthInMeters
- decimalLatitude
- decimalLongitude
- abundance
- scientificName
- eventDate (in the format y-m-d H:M:S.s, e.g. 2017-03-13 02:50:00.0)

If the data set has different column names, a user can directly load the data into the Notebook via the `DataFrames.jl/CSV.jl` packages; for example to extract a column named "longitude" in a CSV file "filename.csv" one can use:

```
using CSV, DataFrames
df = CSV.read("filename.csv", DataFrame);
lon = df.longitude
```

Alternatively, a user can adapt the function `read_data` in the `BlueCloudPlankton.jl` module (or simply reformat the data so that it has the identical format as the original CPR data).

The spatial domain is defined by the variable `gridlon` and `gridlat` in the file `grid.jl`. The notebook automatically downloads and prepares all necessary covariables on the defined grid (unless the relevant files already exist).

3.1.4 Data sources

Variables	Data sources	Infrastructure	Access through
Zooplankton abundances	https://www.emodnet-biology.eu/data-catalog?module=dataset&dasid=216	EurOBIS	Blue Cloud (VRE or DD&AS)
Bathymetry	https://www.gebco.net/	GEBCO	GEBCO or Blue Cloud Vlab
Sea water temperature and salinity	https://www.seadatanet.org/	SeaDataNet	Blue Cloud (DD&AS) or Blue Cloud Vlab
Nitrate, Silicate and Phosphate	https://www.ncei.noaa.gov/products/world-ocean-atlas	World Ocean Atlas, NOAA	World Ocean Atlas or Blue Cloud Vlab

Table 1 Zooplankton EOVS data sources

A temporary copy of these datasets will be downloaded when the notebook is run and saved in the “BlueCloud-data” folder (located in the user’s home directory). If the user re-runs the notebook, the datasets are not downloaded again and the previously downloaded files are reused.

3.1.5 Scientific references

Barth, A., Beckers, J.-M., Troupin, C., Alvera-Azcárate, A., and Vandenbulcke, L. (2014): [DIVAnd-1.0: n-dimensional variational data analysis for ocean observations](#), *Geosci. Model Dev.*, 7, 225-241, doi: 10.5194/gmd-7-225-2014.

Barth, A., Troupin, C., Reyes, E., Alvera-Azcárate, A., Beckers J.-M. and Tintoré J. (2021): [Variational interpolation of high-frequency radar surface currents using DIVAnd](#). *Ocean Dynamics*, 71, 293-308, doi: 10.1007/s10236-020-01432-x.

DIVAnd tool github repository: <https://github.com/gher-ulg/Divand.jl>

3.2 Phytoplankton EOVS

Authors: Renosh Pannimpullath Remanan, Raphaëlle Sauzède, Julia Uitz and Hervé Claustre, Laboratoire d’Océanographie de Villefranche (LOV), Institut de la Mer de Villefranche (IMEV), CNRS – Sorbonne Université (France).

Corresponding authors / maintainer: julia.uitz@imev-mer.fr and raphaelle.sauzede@imev-mer.fr

3.2.1 Short description of the service

The phytoplankton Essential Ocean Variables (EOV) demonstrator aims to provide a methodology to generate **global open ocean three-dimensional (3D) gridded products of (1) chlorophyll a concentration (Chla)**, which is a proxy of the total phytoplankton biomass, and (2) **Phytoplankton Functional Types (PFT)**, as a proxy for phytoplankton diversity, based on vertically-resolved *in situ* data of ocean physical properties (temperature and salinity) matched up with satellite products of ocean colour and sea level anomaly.

3.2.1.A Machine learning method

The phytoplankton EOV products are developed following the method of *Sauzède et al. (2016)*, which relies on machine learning, specifically an artificial neural network (Multi-Layer Perceptron, MLP), and retrieves the vertical distribution of biogeochemical properties from merged ocean colour and hydrological data. MLPs consist of several layers: one input layer, one output layer and one or several hidden layers. Each layer is composed of neurons, which are elementary transfer functions that provide outputs when inputs are applied.

Here, following the same philosophy as the method developed by *Sauzède et al. (2016)*, two different MLP-based algorithms are developed for the independent retrieval of the Chla and of the PFT EOV products.

A) The first MLP retrieves the depth-resolved Chla product and is trained using in-situ depth-resolved measurements of Chla, temperature and salinity (T/S), from the global BioGeoChemical-Argo (BGC-Argo) observation network (Coriolis Global Data Center), matched-up with global satellite-derived products. The MLP input layer is composed of three main components:

- Surface satellite-based inputs from CMEMS and GlobColour, such as the ocean colour remote sensing reflectance (R_{rs}) at five wavelengths, the photosynthetically available radiation (PAR), and the sea level anomaly (SLA);
- Depth-resolved ocean physical properties such as components derived from a principal component analysis (PCA) of the T/S vertical profiles and the mixed layer depth;
- Time (day of the year transformed in cycles) and geographical coordinates of the ocean colour and hydrological data.

B) The retrieval of the depth-resolved PFT product relies on two distinct MLP-based algorithms. The first MLP is an upgraded version of the method developed by *Sauzède et al. (2015)* that includes hydrological information as additional input. This MLP is trained using a database comprising concurrent shipborne measurements of phytoplankton pigments determined by High Performance Liquid Chromatography (HPLC), chlorophyll fluorescence and T/S profiles. This MLP is applied to the BGC-Argo database, with vertical profiles of chlorophyll fluorescence and T/S used as inputs, in order to infer phytoplankton community composition, expressed in terms of depth-resolved Chla associated with three PFTs (pico-nano- and microphytoplankton). This approach enables to enrich the global BGC-

Argo database with the PFT information, which would not be available otherwise. Then, a second MLP is trained using the “PFT-enriched” BGC-Argo database, matched-up with satellite-derived products, in an analogous manner as described in A.

3.2.1.B Model validation

From around 90,000 profiles of the Global BGC database, we selected the Chla profiles that had matchups with Remote sensing reflectance (Rrs) and Sea Level Anomaly (SLA) for the present study. The total number of Chla profiles considered was 26927, from which 80% of the profiles (21541 profiles) were selected for the training and 20% of the profiles (5386 profiles) were selected for the validation. The red points shown in the map are the locations of the Chla profiles used for the training and the cyan points are the locations of the profiles used for the validation (left panel in Figure 4). The validation of the model was performed with Chla from MLP versus Chla from floats for 4347 profiles using density scatter plot (right panel in Figure 4). The linear regression comparison between the Chla derived from the model versus Chla from floats shows satisfactory results with a coefficient of determination (R^2) value of 0.80, slope value of 0.92, and Mean Absolute Percentage Difference (MAPD) of 28.76%.

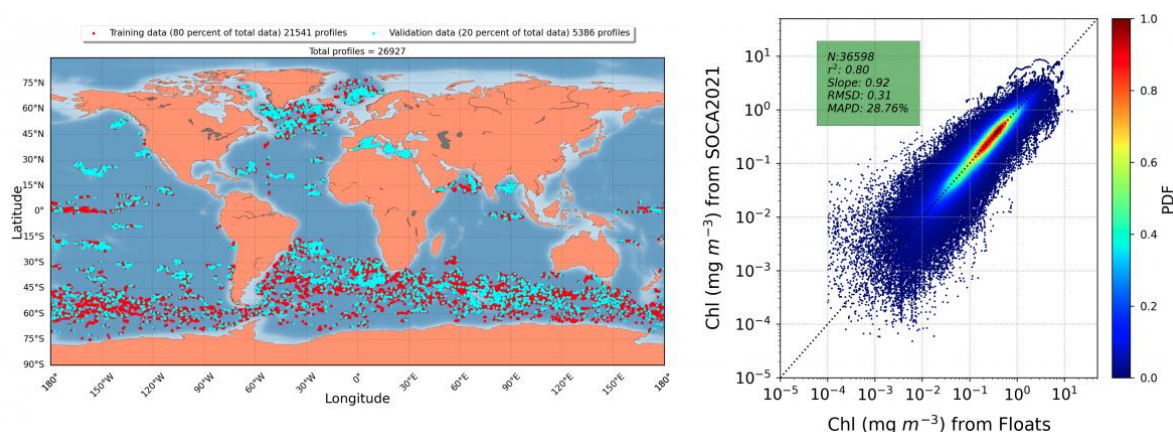


Figure 4 Left panel represents the geographical locations of Chla (BGC-Argo) profiles used for the present study. Right panel represents a density scatter plot for the validation of the model

3.2.2 Targeted users

The specific target users of this service are **researchers and modelers** who wish to create or use global 3D fields of phytoplankton. Results can be used by EOVS experts who wish to evaluate the state of primary producers in the marine ecosystem.

3.2.3 Step by step guideline to use the service

This section includes the steps to use the services explained in the previous section and considerations to take into account. Users, to be able to run these services, need to go to https://blue-cloud.d4science.org/web/zoo-phytoplankton_eov

All necessary files to calculate the Chla and PFT products for the year 2018 are located in the Workspace > VRE folders > Zoo-Phytoplankton_EOV > Phytoplankton_EOV. In this folder, there are 3 folders: Chla_Product, PFT_Product and Inputs. The inputs folder contains all the input data that is used for both products (monthly “.nc” files).

The Chla_Product and PFT_Product folders include themselves 3 subfolders:

- The “Programs” folder which contains 2 Jupyter notebooks and 2 folders:
 - The Functions folder contains all the necessary functions required to generate the 3D Global products, and
 - The Models folder contains the trained MLP models and PCA models.
- The Outputs folder that contains the output global 3D products generated for each month of the year 2018, as “.nc” files.
- The Plots folder that contains the visualisation of the outputs products, as “.png” files (2D spatial plots for 36 depths).

Steps to run the notebooks:

In the [Zoo-phytoplankton Vlab home](#) navigate to the shared workspace (VRE Folders) and make a copy of the folder Phytoplankton_EOV in your workspace (home directory). This can be done, in the JupyterHub > Terminal, using this code to automatically copy all this files:

```
cp-r /home/jovyan/workspace/VREFolders/Zoo-  
Phytoplankton_EOV/Phytoplankton_EOV/Chla_Product/ /home/jovyan
```

Before executing these notebooks: `CREATE_MONTHLY_FIELDS_Loop_ZNORM.ipynb`, `Output_spatial_plots.ipynb`, and `Functions/SOCA_CHLA_ZNORM_2020.ipynb`, the paths should be checked and modified accordingly.

To generate the Chla product, open and run the two Jupyter notebooks available in the Phytoplankton_EOV/Chla_Product/Programs folder from the Jupyter Lab of the VRE, in the following order:

- The first notebook: `CREATE_MONTHLY_FIELDS_Loop_ZNORM.ipynb` generates the global 3D Chla products in NetCDF format. For each month, the output is saved in the corresponding monthly folder, under Outputs.
- The second notebook: `Output_spatial_plots.ipynb` is used to generate the visualization plots based on the output NetCDF files obtained from the first notebook. For each month, the plots are saved in the corresponding monthly folder under Plots.

To generate the PFT product, open and run the two Jupyter notebooks available in the Phytoplankton_EOV/PFT_Product/Programs folder from the Jupyter Lab of the VRE, in the following order:

- The first notebook: `CREATE_MONTHLY_FIELDS_PFT_ZNORM_N1.ipynb` generates the global 3D PFT products (Micro-Chla, Nano-Chla, and Pico-Chla) in NetCDF format. For each month, the output is saved in the corresponding monthly folder, under Outputs.
- The second notebook: `Plots_output_spatial_monthly_PFT_2018.ipynb` is used to generate the visualization plots based on the output NetCDF files obtained from the first notebook. For each month, the plots are saved in the corresponding monthly folder under Plots.

The demonstrator generates global 3D Chla and PFT products for the year 2018. To reproduce this workflow with different data, users must have their input data in the same format as the data provided in the "Inputs" folder and change the paths on the 2 main Jupyter notebooks to read this data accordingly. The outputs NetCDF files and 2D spatial plots will be generated in the corresponding folders under Outputs and Plots. Figures 5 and 6 illustrate the output product generated for the surface (0m depth) for January, which is a typical winter month. The Chla product is in units of mg of chlorophyll a per cubic meter (m^{-3}); the PFT product provides the chlorophyll a concentration associated with the micro-, nano-, and picophytoplankton size classes in units of mg of chlorophyll a m^{-3} .

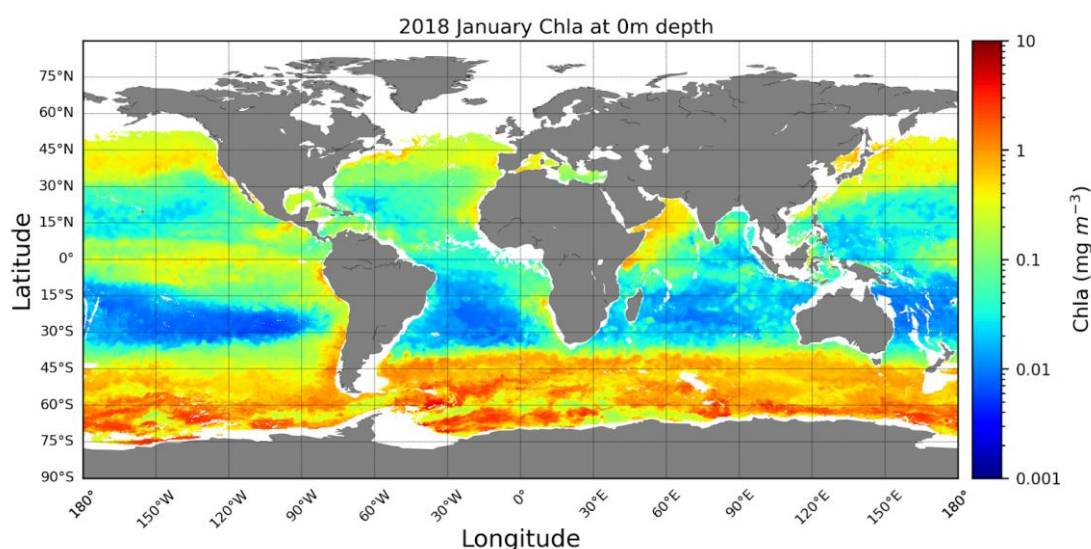


Figure 5 Global Chla concentration at 0m depth for January 2018.

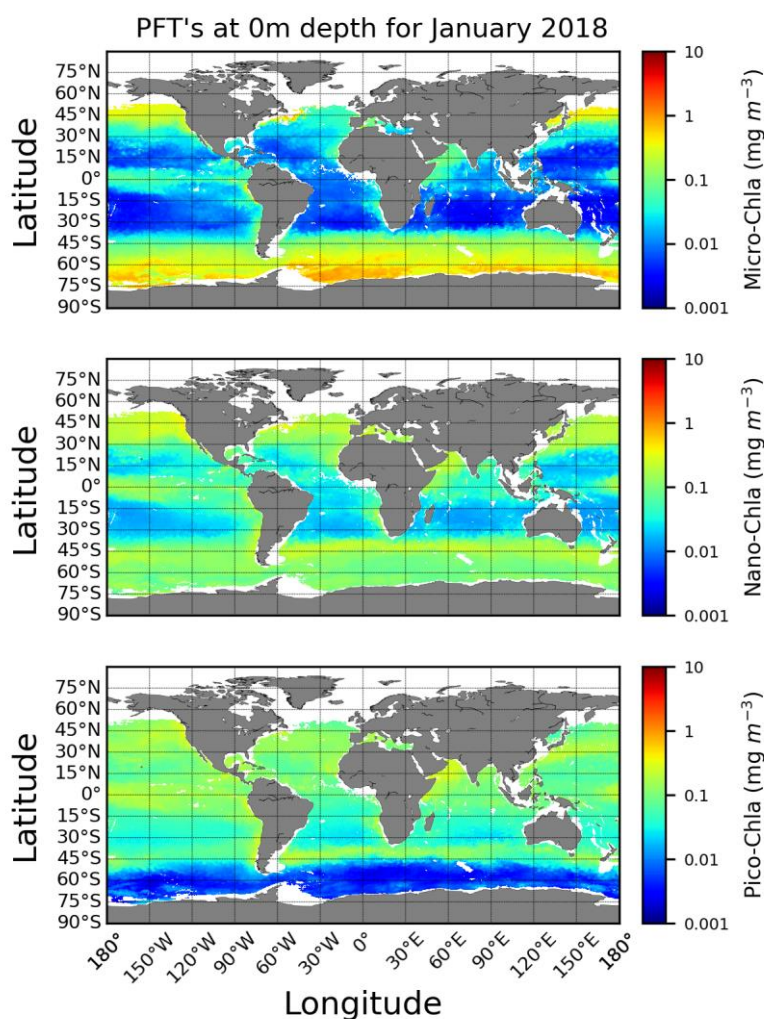


Figure 6 Global PFT at 0m depth for January 2018.

3.2.4 Data sources

Variables	Data sources	Infrastructure	Access through
Satellite-derived reflectance	OCEANCOLOUR GLO OPTICS L3 REP OBSERVATIONS 009 086	CMEMS (Copernicus marine service)	Blue Cloud (Notebook interface to WEKEO HDA API)
Satellite Sea Level Anomaly	SEALEVEL GLO PHY L4 REP OBSERVATIONS 008 047 product	CMEMS	Blue Cloud (Notebook interface to WEKEO HDA API)
Physical data: T, S, MLD Global ARMOR 3D products	CMEMS MULTIOBS GLO PHY REP 015 002	CMEMS	Blue Cloud (Notebook interface to WEKEO HDA API)
BGC-Argo Float NetCDF files (S-files)	ftp.ifremer.fr/ifremer/argo/ ; http://www.argo.ucsd.edu	Argo GDAC (Coriolis center)	Blue Cloud

Variables	Data sources	Infrastructure	Access through
			(Notebook interface to WEKEO HDA API)
Satellite-derived Photosynthetically Available Radiation	ftp://ftp.hermes.acri.fr	GlobColour	GlobColour or Blue Cloud Vlab
High-performance liquid chromatography (HPLC) data	http://www.obs-vlfr.fr/proof/cruises.php	LOV database	Blue Cloud Vlab
Bathymetry	https://www.gebco.net/data_and_products/gridded_bathymetry_data/	GEBCO	GEBCO or Blue Cloud Vlab

Table 2 Phytoplankton EOVS data sources

BGC-Argo floats and High-performance liquid chromatography (HPLC) data were only used to train the model, and bathymetry data was used to select open ocean waters with depths greater than 1500m.

3.2.5 Scientific references

Sauzède, R., H. Claustre, C. Jamet, J. Uitz, J. Ras, A. Mignot, and F. D’Ortenzio (2015). Retrieving the vertical distribution of chlorophyll a concentration and phytoplankton community composition from in situ fluorescence profiles: A method based on a neural network with potential for global-scale applications, *J. Geophys. Res. Oceans*, 120, 451–470, doi:10.1002/2014JC010355.

Sauzède, R., Claustre, H., Uitz, J., Jamet, C., Dall’Olmo, G., d’Ortenzio, F., Gentili, B., Poteau, A. and Schmechtig, C. (2016). *A neural network-based method for merging ocean color and Argo data to extend surface bio-optical properties to depth: Retrieval of the particulate backscattering coefficient*. *Journal of Geophysical Research: Oceans*, 121(4), pp.2552-2571.

3.3 Modelling phyto & zoo –plankton interactions

Authors: Viviana Otero, Steven Pint, Patricia Cabrera, Lennert Schepers and Gert Everaert, Flanders Marine Institute (Belgium).

Maintainer: stevenpint@vliz.be

3.3.1 Short description of the service

This service provides a workflow to run a **mechanistic model** using near real-time data to quantify the relative contributions of the bottom-up and top-down drivers in phytoplankton dynamics. The Nutrient-Phytoplankton-Zooplankton (NPZ) model used in this demonstrator was adjusted from the

Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) model of *Soetaert and Herman (2009)*. The NPZD model is commonly used and describes the four state variables of nutrients, phytoplankton, zooplankton and detritus.

Phytoplankton dynamics are simulated based on information from nutrient concentrations and zooplankton density. Based on these simulations using near real-time data, it is possible to calculate and visualise the relative contribution of each bottom-up or top-down driver, i.e. (1) nutrients, (2) Sea Surface Temperature (SST), (3) photosynthetically active radiation (PAR) and (4) zooplankton grazing, over time.

The validation of the model is performed by comparing the model predictions, i.e. phyto- and zooplankton biomass, with field observations. The Root Mean Square Error (RMSE) is calculated between prediction values and observational values. By doing so and by running the model for multiple parameterizations, we are able to select the best 10% simulations (lowest RMSE) to predict phyto- and zooplankton dynamics and define confidence intervals around the model predictions. To estimate the relative contributions of the drivers, we select the best 5% simulations to decrease computational effort.

3.3.2 Targeted users

The specific target users of this service are plankton researchers and ecosystem modelers. The methods and results can be useful for EOVS experts to understand the shifts in primary production and its spatiotemporal distribution and dynamics.

3.3.3 Step by step guideline to use the service

The workflow is provided in a R markdown document available in the shared workspace: `Zoo-Phytoplankton_EOV/Modelling_phyto_and_zooplankton_interactions/Manual_NPZ_model.Rmd`. This R markdown document contains all necessary information and code to (1) (re)calibrate the NPZ model, (2) simulate phyto-and zooplankton dynamics, (3) validate modelling results with observational data, (4) calculate the relative contribution of the bottom-up and top-down drivers on phytoplankton dynamics and (5) visualise the modelling results.

Steps to run the Rmarkdown:

1. Copy the document (`Manual_NPZ_model.Rmd`) to your personal workspace. This is the only document that you need.
2. Open the document from the VRE Rstudio and run it step by step, or select Run All if you are already familiar with the script and choose the region and iterations you want to run the model for.

There are necessary files to run the NPZ model in the shared workspace, such as .csv files and R scripts, but these files are automatically copied to your personal workspace when running the R markdown. Results are automatically stored in the folder Output > Final results in your personal workspace.

Figure 7 shows the main output of the model for one of the locations, where we can observe the relative contributions of each of the drivers in phytoplankton abundances.

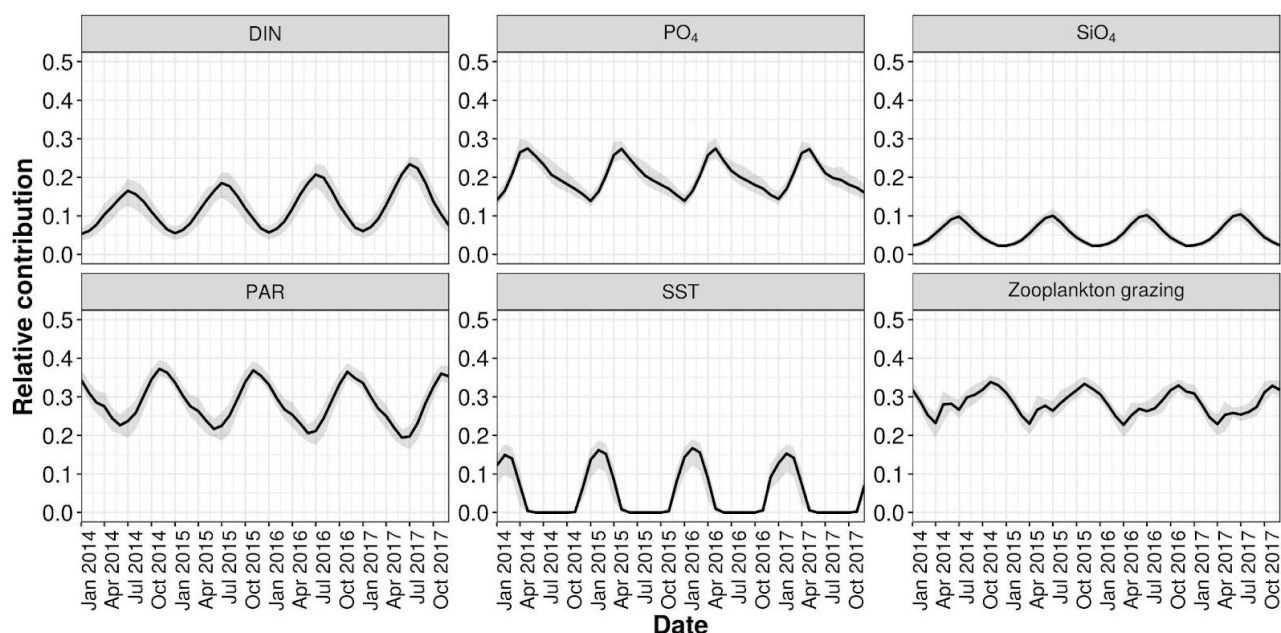


Figure 7 Average monthly relative contributions for each limitation factor in the growth of phytoplankton for the nearshore region.

If users would like to use the model with other data, note that the optimization of the parameters for the model is done for the Belgium part of the North Sea (BPNS). If other regional seas are considered with another biogeochemical cycling, consider that the model might need to be recalibrated. The code for this reparametrization is available in the R Markdown document. The input data (e.g. temperature, nutrients) are provided for the BPNS as a daily time series. If other sites are studied where the temporal resolution of the abiotic conditions is low (not on a daily basis), it is necessary to use interpolation methods such as Generalized Additive Models (GAM) or Generalized Linear Models or (GLM) to obtain a complete set of input data. The GAM's R scripts that we used here to create daily time series from available observed data (e.g. temperature, dissolved inorganic nitrogen, phosphate or silicate) are available on the [shared workspace](#), but are for our case study already pre-computed and available (Workspace > VRE Folders > Zoo-Phytoplankton_EOV > Modelling_phyto_and_zooplankton_interactions> Manual NPZ model > Rscripts). These are useful to prepare daily data for other areas.

There is an example of the NPZ model for one location in the North East Atlantic region, where the initial parameters were changed to calibrate the model. This can be found here: /Workspace/VRE Folders/Zoo-Phytoplankton_EOV/Modelling_phyto_and_zooplankton_interactions.

3.3.4 Data sources

Variables	Data sources	Infrastructure	Access through
Zooplankton abundances	https://www.emodnet-biology.eu/datacatalog?module=dataset&dasid=4687	EMODnet Biology	Blue Cloud (VRE or DD&AS)
Phytoplankton abundances (Chla)	http://rshiny.lifewatch.be/station-data/	LifeWatch	LifeWatch or Blue Cloud Vlab
Abiotic data (nutrients, PAR and temperature)	http://rshiny.lifewatch.be/station-data/	LifeWatch	LifeWatch or Blue Cloud Vlab

Table 3 Modelling phyto & zoo –plankton interactions data sources

3.3.5 Scientific references

Soetaert, K. and Herman, P.M.J. (2009). *A practical guide to ecological modelling. Using R as a Simulation Platform*. Springer-Verlag, New York, US, p. 372.

4 Demonstrator # 2 – Plankton Genomics

4.1 Exploring genetic data & identifying clusters containing unknown genes

Author: Pavla Debeljak (Sorbonne Université)

Contact person: pavla.debeljak@gmail.com

4.1.1 Short description of the service

Recent meta genomic studies have revealed that marine plankton is far more diverse than previously thought (Carradec et al. 2018, Duarte et al. 2020), with hundreds of thousands of genetically distinct taxa and more than 150 million genes documented. However more than half of the planktonic ‘omic’ sequences have still unknown taxonomy and/or function, especially in terms of sequences with eukaryotic origin. These unprecedented amounts of data on planktonic communities call for innovative data-driven methodologies to quantify and observe their biogeographic importance.

The key objective of the service is to enable the discovery of unknown genes using the large dataset collected during the Tara Oceans Expedition. The service allows retrieving unknown genes from annotation files for 4 different plankton size classes and then building gene clusters by similarities of sequences and larger metabolic pathways.

The output file is used for the second service of this demonstrator: mapping the geographic distribution of plankton functional gene clusters using habitat prediction models.

4.1.2 Targeted users

This service is designed for expert users with a strong genomic and bioinformatics background.

4.1.3 Step by step guideline to use the service

The codes are accessible to users from the Blue Cloud infrastructure and may also be duplicated and modified locally.

The data files for this service are available in a zip.file at: <https://data.d4science.net/PPhR>

The notebooks are self-explanatory in the sense that directions to use the functions and the choices to make are given alongside the code.

Information concerning the installation of this service are available in the readme.docx file at: <https://data.d4science.net/PPhR>

4.1.3.A Retrieving Unknowns from annotation files

In Notebook 1.1. sequence annotations from the Marine Atlas of Tara Ocean Unigenes (MATOU) are used for the exploration of non-annotated sequences. The sequences are available in FASTA format, which is a **text-based format for representing either nucleotide sequences or peptide sequences**, in which base pairs or amino acids are represented using single-letter codes.

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The FASTA sequence files containing over 116 million genes are retrievable through the EMBL ENA web service (<https://www.blue-cloud.org/data-infrastructures/european-nucleotide-archive-ena>, look into “Core services”).

Associated Data can be found in the publication (<https://www.nature.com/articles/s41467-017-02342-1>).

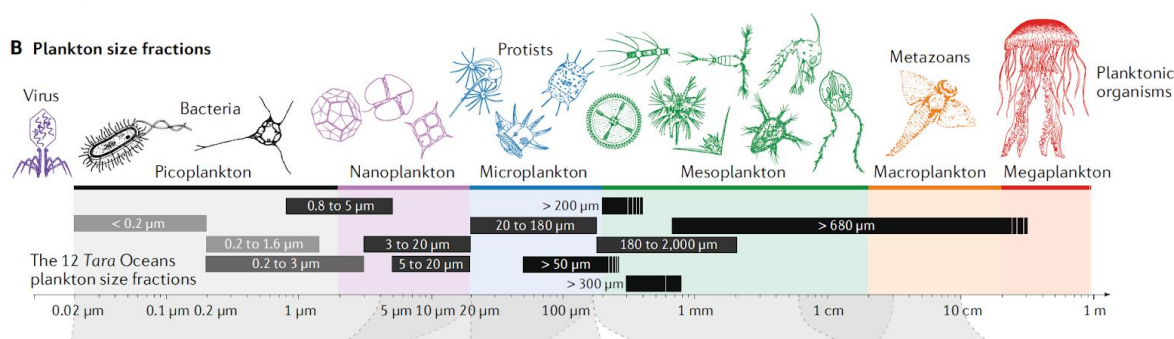


Figure 8 Plankton size fractions from Sunagawa et al. 2020

The dataset includes 4 different size classes (Figure 8) based on the different filter sizes used at on-board filtration: GGMM: 0.8-5 µm, MMQQ: 5-20 µm, QQSS: 20-180 µm, SSUU: 180-2000 µm (nanoplankton to mesoplankton). The Jupyter Notebook allows for the extraction of Unknowns based on Function (no annotation in the Protein Families catalogue (<http://pfam.xfam.org/>) or Taxonomy retrieved from the publication. The codes in the notebook find the unknown sequences and calculate the ratio of unknowns to knowns for different size fractions of the Tara Ocean data. Furthermore, giant scaffolds can be excluded and mean sequence length and standard deviation calculated and plotted in R (ggplot).

The different size classes ranging from Nanoplankton to Mesoplankton analysed together or separately can then be plotted (Figure 9) using environmental data in R (leaflet package).



Figure 9 Example output Notebook 1.1. from R (leaflet package)

4.1.3.B Creating protein functional clusters

Notebook 1.2. allows for the creation of protein functional clusters from FASTA files derived from Metagenomic and Metatranscriptomic sequencing. These clusters contain annotated as well as unknown sequences that can be passed on to the IT service: mapping the geographic distribution of plankton functional gene clusters using habitat prediction models (described in the next section).

The necessary data can be retrieved from <https://www.genoscope.cns.fr/tara> under “Tara Oceans Eukaryotic Genomes (“SMAGs”).

The provided peptide fast file contains 713 manually curated meta genome assembled genomes containing 10,207.435 proteins. With these, over 10 million sequences functional clusters of proteins can be built using a sequence similarity network (here using the igraph package in R). In this graph (Figure 10), nodes are protein sequences and edges represent the similarity and coverage between each pair of sequences.

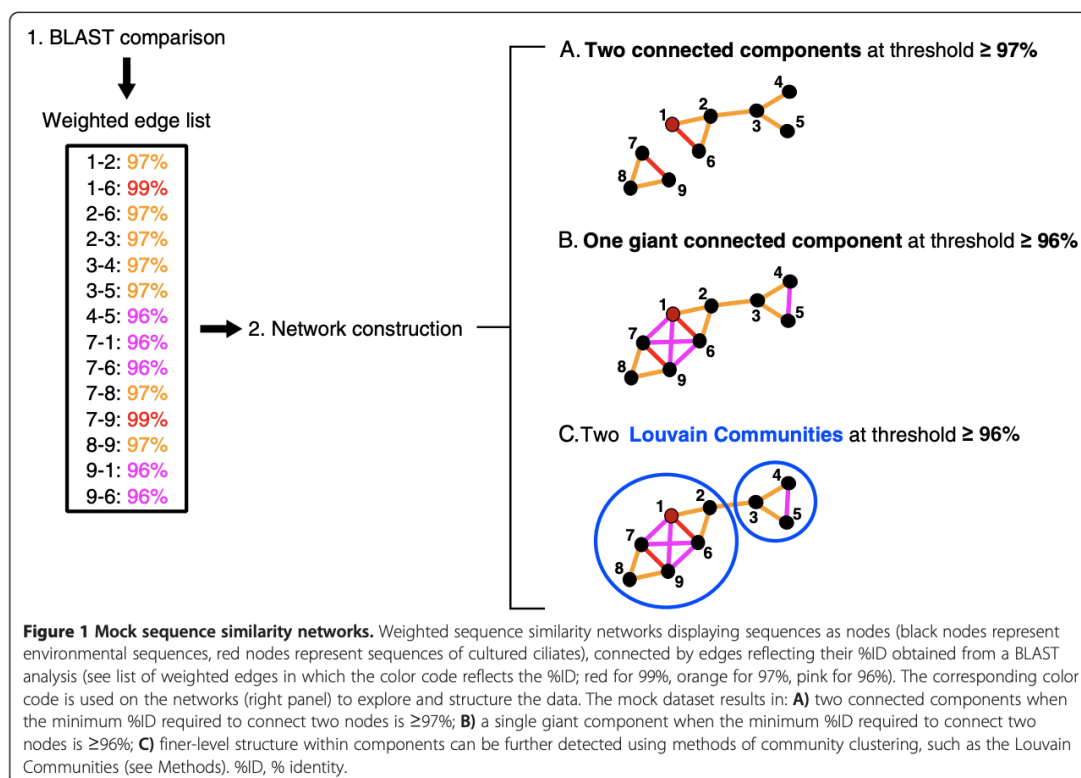


Figure 10 Sequence similarity network analysis based on Forster & Bittner 2015.

The weighted sequence similarity networks display sequences as nodes (black nodes represent environmental sequences, red nodes represent sequences of cultured ciliates), connected by edges reflecting their %ID obtained from a BLAST analysis (see list of weighted edges in which the colour code reflects the %ID; red for 99%, orange for 97%, pink for 96%). The corresponding colour code is used on the networks (right panel of the figure 10) to explore and structure the data. The mock dataset results in: **A)** two connected components when the minimum %ID required to connect two nodes is $\geq 97\%$; **B)** a single giant component when the minimum %ID required to connect two nodes is $\geq 96\%$; **C)** finer-level structure within components can be further detected using methods of community clustering, such as the Louvain Communities.

Such approaches allow for the observation of sequence clusters that are putatively homogenous in function and have recently highlighted the potential for deciphering global ocean biogeochemistry (Faure et al. 2021).

4.1.4 Data sources

Variables	Data sources	Infrastructure	Access through
MATOU V1. - Annotations (PFam Protein Families & Taxonomy), Metagenomic Read Abundance	https://www.genoscope.cns.fr/tara	ELIXIR / ENA	Blue Cloud (VRE or DD&AS)
Protein Family Clusters from SMAGs	https://www.genoscope.cns.fr/tara	ELIXIR / ENA	Blue Cloud (VRE or DD&AS)
Metagenomic read abundance for Protein Clusters	European Nucleotides Archive (ENA) https://www.ebi.ac.uk/ena/browser/view/ERZ480625	Blue Cloud	Blue Cloud PostgreSQL database

Table 4 Demo 2 - Exploring genetic data – Data sources

For the metagenomic read abundance for protein clusters, we recommend using the data set already pre-formatted in the Bluecloud PostgreSQL database accessed through the following notebook: <https://data.d4science.net/3hxx>. Note that data are not accessible in a "downloadable" sense; they are accessed by the notebook.

4.1.5 Scientific references

Carradec, Q., Pelletier, E., Da Silva, C. et al. (2018) *A global ocean atlas of eukaryotic genes*. Nat Commun 9, 373. <https://doi.org/10.1038/s41467-017-02342-1>

Duarte, C.M. et al. (2020) *Sequencing effort dictates gene discovery in marine microbial metagenomes*. SFAM 22:11 <https://doi.org/10.1111/1462-2920.15182>

Faure, E., Ayata, SD. & Bittner, L. (2021). *Towards omics-based predictions of planktonic functional composition from environmental data*. Nat Commun 12, 4361. <https://doi.org/10.1038/s41467-021-24547-1>

Forster D., Bittner L., Karkar S., Dunthorn M. , Romac S., Audic S., Lopez P., Stoeck T. & Baptiste E. (2015). *Testing ecological theories with sequence similarity networks: marine ciliates exhibit similar geographic dispersal patterns as multicellular organisms*. BMC Biology 13:16. <https://doi.org/10.1186/s12915-015-0125-5>

Sunagawa, S., Acinas, S.G., Bork, P. et al. (2020) *Tara Oceans: towards global ocean ecosystems biology*. Nat Rev Microbiol 18, 428–445. <https://doi.org/10.1038/s41579-020-0364-5>

4.2 Mapping the geographical distribution of plankton functional gene clusters using habitat models

Authors: A. Schickele¹, P. Debeljack², L. Guidi¹, S. Ayata^{1,3}, J.O. Irisson¹

Corresponding author/maintainer: alexandre.schickele@imev-mer.fr

¹ Sorbonne Université, CNRS, Laboratoire d'Océanographie de Villefranche (LOV), Villefranche-sur-Mer, FRANCE.

² Institut de Systématique, Évolution, Biodiversité (ISYEB), MNHN, CNRS, Sorbonne Université, EPHE, Université des Antilles, Paris, FRANCE

³ Sorbonne Université, CNRS, IRD, MNHN, Laboratoire d'Océanographie et du Climat: Expérimentations et Approches Numériques (LOCEAN-IPSL), Paris, FRANCE

4.2.1 Short description of the service

Planktonic organisms and by extension the composition and genetic potential of plankton in a given geographical area is influenced by the environmental context. In this notebook, we provide a series of tools to explore the relationship between the abundance of plankton genes and the environmental context, in order to project the biogeography of key metabolic pathways and as yet unknown plankton genes.

This notebook uses a machine learning regression method, namely multivariate gradient boosting (i.e. several gene clusters related to the same metabolic pathway are modeled at once), and is implemented on top of a Python library (MBTR; see <https://mbtr.readthedocs.io/en/latest/>). In line with best practices in machine learning, we perform a n -fold cross-validation: i.e. the data are split in n -equal sized groups, n -models are trained on $n-1$ splits, holding the last split for model evaluation only. In order to select the best model, we test different sets of hyperparameters, leading to one fitted model per cross-validation fold and set of hyperparameters. The quality of fit of each model is measured by a loss function at each boosting round and the best model is selected according to the hyperparameters and number of boosting rounds that produced the minimum loss averaged between all cross-validation folds.

The performance of our select model is evaluated by using the test split corresponding to each cross-validation fold and the following evaluation metrics: r-squared (R^2) and root mean squared error (rmse).

4.2.2 Targeted users

This service describes the entire modelling pipeline and codes. It is essentially designed for expert users with a strong genomic and modelling/machine learning background.

4.2.3 Step by step guideline to use the service

The codes are accessible for users from the Blue Cloud infrastructure and may also be duplicated and modified locally.

The files for this service available in a zip.file at: <https://data.d4science.net/3hwx>

The notebook is self explanatory in the sense that directions to use the functions and the choices to make are given alongside the code.

Information concerning the installation of this service are available in the readme.docx file at: <https://data.d4science.net/3hwx>

4.2.4 Data sources

Variables	Data sources	Infrastructure	Access through
Apparent Oxygen Utilization, Nitrate, Oxygen, Phosphate, Salinity, Silicate, Temperature	https://www.ncei.noaa.gov/products/world-ocean-atlas	World Ocean Atlas (NOAA)	NOAA or Blue Cloud VLab
Chlorophyll, Bathymetry	https://marine.copernicus.eu	CMEMS	Blue Cloud (Notebook interface to WEKEO HDA API) or Blue Cloud VLab
Mixed layer depth, Depth of euphotic zone, Distance to coast	Blue Cloud	Blue Cloud	Blue Cloud VLab
Metagenomic read abundance for Protein Clusters	Demonstrator #2 - Notebook 1	Blue Cloud	Blue Cloud PostgreSQL database

Table 5 Demo 2 - Mapping the geographical distribution of plankton - data sources

4.2.4.A Environmental data

First, we retrieved a large set of monthly and annual climatologies encompassing the 2005 to 2017 period, at 1° x 1° resolution and on a geographical domain ranging from -180 to +180°E and -90 to +90° for the following environmental variables, from the World Ocean Atlas (<https://www.ncei.noaa.gov/products/world-ocean-atlas>):

- **AOU**: Apparent Oxygen Utilization
- **nitrate**: Nitrate concentration
- **o2sat**: Percent Oxygen Saturation
- **oxygen**: Dissolved oxygen concentration
- **phosphate**: Phosphate concentration

- **salinity:** Sea surface salinity
- **silicate:** Silicate concentration
- **temperature:** Sea surface temperature

In addition, the following environmental variables were retrieved from (<https://marine.copernicus.eu/>):

- **CHL:** Chlorophyll A concentration
- **bathymetry:** sea bottom bathymetry

From the monthly climatologies and the variables above, we derived three additional environmental variables:

- **MLD:** Mixed Layer Depth, calculated using the method from Boyer de Montégut (2004)
- **ZE:** depth of the lower limit of the Euphotic Zone, calculated using the method from Lavigne et al. (2012)
- **distcoast:** calculated distance to the nearest coast

Finally, for all environmental variables, we calculated the mean (mean), standard deviation (sd), median (med), median average deviation (mad), monthly minimum (min) and monthly maximum (max) over the 2005 - 2017 period. **These data are accessible in the Blue Cloud dataspace from Jupyter Hub.**

4.2.4.B Genomic data

Second, we use MetaGenomic data retrieved from the Marine Atlas of Tara Ocean Unigenes (MATOU) described in Notebook 1. These data correspond to the number of reads (i.e. proxy of abundance) per genes and Tara Ocean station, for the surface and 0.5 to 3 μm organisms (i.e. pico-plankton). To be able to explore the genomic diversity of plankton, the data are annotated as following:

- If possible, each gene is functionally annotated using the KEGG (Kyoto Encyclopedia of Genes and Genomes) framework: <https://www.genome.jp/kegg/pathway.html>. This corresponds to a two-level annotation, i.e. the KEGG functional annotation, contained in KEGG metabolic pathways.
- Genes are grouped by similarity of sequences into clusters, containing annotated genes (i.e. KEGG + pathways) and as yet unknown genes.

In other words, we are able to model the biogeography of gene clusters related to a given metabolic pathway, and then the genes related to one of the clusters.

To meet the computing requirements and for better performance (e.g. avoid heaving memory usage), these data are stored in a PostgreSQL database within the Blue Cloud infrastructure. The latter is accessed during the first step of the modelling pipeline.

4.2.5 Scientific references

Nespoli, Lorenzo and Medici, Vasco (2020). *Multivariate Boosted Trees and Applications to Forecasting and Control* (nespoli2020multivariate). arXiv preprint arXiv:2003.03835

5 Demonstrator # 3 – Marine Environmental Indicators

5.1 Targeted users

The overall target audience of the services in the demonstrator (and VLab) Marine Environmental Indicators are **end users** such as environmental protection agencies and international stakeholders in the MSFD, in the UN SDG 14, in the UN SDG 13 and in the Blue Economy, themselves directly or through intermediate users who will work for them.

More precisely, each of the 5 available services is addressing specific features and functionalities as detailed:

- Marine environmental indicator generator

The service offers to environmental protection agencies a flexible capacity to perform statistical analyses of the quality and characteristics of the marine environment for the Mediterranean Sea region, with possibility to scale up to the Global.

- Ocean pattern and ocean regime indicators

The target audience of the service are the scientific users, providing to them a tool to facilitate the discovery of pattern/regime indicators based on machine learning and a simplified way to analyse oceanographic data.

- Storm severity index

The target audience of this Storm Severity Index (SSI) service are scientific users who are looking for a quantitative impact modelling of severe wind/storms for different areas in the Mediterranean Sea region and for different time periods, up to 40 years (1979 - 2020).

The SSI service offers the scientific users insight about impact of severe wind or storms and the ability to combine or correlate the calculated SSI with other marine environmental indicators.

- Simple access to carbon data

The target audience of this service that allows an easy access to carbon data are stakeholders in the UN SDG 14 (especially SDG14.3) and more precisely the agencies with interest for the implementation of indicators related to the inorganic carbon data.

Another audience are the scientific and technological users with interest to explore the actual capability of the current marine data infrastructures (i.e. ERDDAP) which are devoted to the dissemination of marine carbon data.

5.2 Marine Environment Indication (MEI) generator

Authors: Massimiliano Drudi (CMCC), Francesco Palermo (CMCC), Antonio Mariani (CMCC), Rita Lecci (CMCC)

Corresponding author / maintainer: oclab@cmcc.it

5.2.1 Short description of the service

The prototype MEI Generator service is a web graphical interface that allows the user to generate and display value-added environmental data from generic marine data.

These value-added environmental data can be an average over time, ocean patterns or regimes ... depending on the method chosen by the user; the output is proposed as a time series or a map.

The current prototype and the one described here, uses Copernicus Marine products (physical modelling products from the Mediterranean Sea) as input data. The method implemented allows various averages.

The flexibility of the service allows the user to specify the desired output type, among:

- timeseries
 - monthly mean time series
 - annual mean time series
 - monthly climatology time series
- maps
 - annual map
 - annual climatology map
 - monthly climatology map

For the available fields:

- temperature
- salinity
- water density
- currents
- kinetic energy

According to the selected output type and field, additional parameters (proposed by the MEI) are required for the finalization of a processing request. The current version of this service uses as input dataset a subset of the CMEMS product MEDSEA_REANALYSIS_PHYS_006_004, covering the period 1987-1989.

In 2022, a major upgrade of this service is planned to improve the flexibility and to implement the integration with more processing methods (i.e. new output types among the services described below) and more data sources, covering a longer period of time.

5.2.2 Step by step guideline to use the service

The MEI Generator service offers to the user an easy and transparent way to process data from external data sources, with the exploitation of methods available (existing and future) on the Data Miner service inside the Blue-Cloud VRE. In fact, in order to obtain value-added data, the interface presents to the user all the possible options, which are needed for the customization of the processing.

The functionalities currently provided are summarized as follow:

1. Generating new output starting from available data.
2. Access and visualize the value-added data previously calculated by the user.

5.2.2.A *Generating new output starting from available data*

In order to generate value-added data, please follow the instructions:

- open MEI Generator app from the VRE menu
- select the desired output, by providing the following information
 - the desired data source (Figure 11a) - now available only **MEDSEA_REANALYSIS_PHYS_006_004**
 - the output type (Figure 11b) - i.e. **monthly climatology timeseries**
- select the additional input parameters :
 - the **output field** (Figure 11c)
i.e. temperature, salinity, density, etc.
 - the **time range** (Figure 11d)
 - the **area** (of interest) in terms of boundaries (Figure 11e)
N.B. The points of minimum latitude and longitude and the points of maximum latitude and longitude build the area of interest. The chosen area (blue rectangle on map in Figure 11) is then visible on the map that is centred on the geographic domain of the selected data source
 - the **depth** layer (Figure 11f)
- and click on the “Execute process” button

Afterward the processing request for a new output is submitted to the DataMiner (WPS service of D4science) service for calculation.

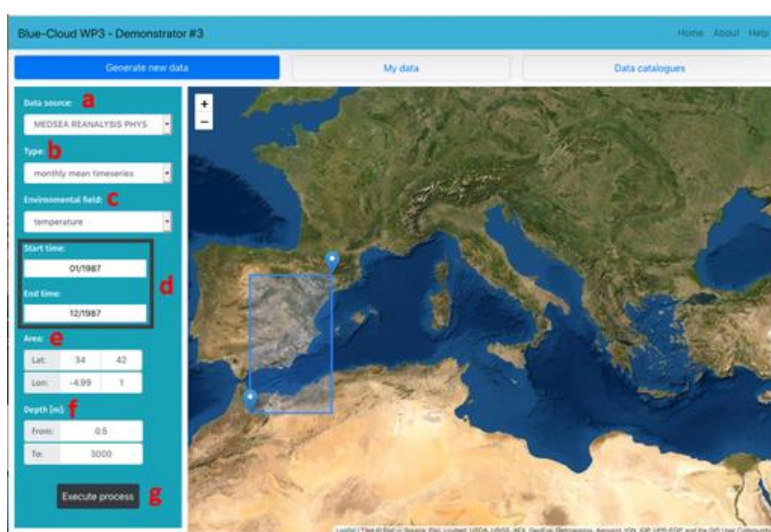


Figure 11 'Generate new output' User Interface.

The user can select data source (a), the output type (b), the environmental field (c), the time range (d), the area of interest (e) and the depth layer (f)

5.2.2.B Access and visualize the value-added data previously calculated by the user

By selecting the "My data" tab (Figure 12) the user can see the status of last launched processes, and after the completion, access the output. The shown information are:

- the creation time for each process (Figure 12a);
- the current status of their execution (Figure 12b & c);
- the data source (Figure 12d);
- the related parameters of the process (Figure 12e);
- the area of interest (Figure 12f);
- the depth range (Figure 12g) and
- the time range (Figure 12h).

Blue-Cloud WP3 - Demonstrator #3								
Generate new data			My data			Data catalogues		
Creation time a	Status b	Outputs c	Data source d	Type e	Area [lat,lon] f	Depth [m] g	Time range h	
2020-10-10T10:30:00	started	No results yet	MEDSEA_REANALYSIS_PHYS_006_004	annual mean timeseries - salinity	[34,-4.99] - [42,1]	[0.5,3000]	1988 - 1989	
2020-10-11T12:05:00	completed	Show i	MEDSEA_REANALYSIS_PHYS_006_004	monthly mean timeseries - temperature	[34,-4.99] - [42,1]	[0.5,3000]	1987-01 - 1987-06	
2020-10-12T15:20:00	completed	Show	MEDSEA_REANALYSIS_PHYS_006_004	monthly climatologic timeseries - density	[34,-4.99] - [42,1]	[0.5,3000]	1988 - 1989	
2020-10-10T10:30:00	error	Log	MEDSEA_REANALYSIS_PHYS_006_004	annual mean timeseries - salinity	[34,-4.99] - [42,1]	[0.5,3000]	1987 - 1989	

Figure 12 'My data' – User Interface (UI).

On the 'My data' tab (box in orange) of the UI, the user can see all his/her requests (past, present and ongoing) with their detailed info : creation time (a), status of process (b & c), original data source (d), parameters used in process (e), area of interest (f), depth range (g) and time range (h). Once a process is finished (status = completed), the user can visualize the end-product by clicking on the 'Show button' (i).

When the status of the process is presented as “completed”, then the user can visualize the data produced by the process by clicking on the “Show” button (Figure 12i). The user then obtains the screen as shown below (Figure 13). The user can select to download the image (Figure 13a - Download Image), the NetCDF file of the data (Figure 13b – Download Data) and eventually the execution log (Figure 13c – Download Log) of the process.



Figure 13 ‘Show result’ in the ‘My data’ UI.

The user can download the illustration of the results (Download Image – a), download the produced data in a NetCDF file (Download Data – b) and/or download the execution logs of the process (Download Log – c).

5.2.3 Data sources

Variables	Data sources	Infrastructure	Access through
temperature, salinity, currents	MEDSEA_REANALYSIS_PHYS_006_004	CMEMS	Blue Cloud (VRE, a 3-year subset dataset)

Table 6 Marine Environmental Indication (MEI) generator data sources

5.2.4 Scientific references

Coppini, G., Marra, P., Lecci, R., Pinardi, N., Cretì, S., Scalas, M., Tedesco, L., D'Anca, A., Fazioli, L., Olita, A., Turrise, G., Palazzo, C., Aloisio, G., Fiore, S., Bonaduce, A., Kumkar, Y., Ciliberti, S. A., Federico, I., Mannarini, G., Agostini, P., Bonarelli, R., Martinelli, S., Verri, G., Lusito, L., Rollo, D., Cavallo, A., Tumolo, A., Monacizzo, T., Spagnulo, M., Sorgente, R., Cucco, A., Quattrocchi, G., Tonani, M., Drudi, M., Panzera, L., Navarra, A., and Negro, G.: SeaConditions: a web and mobile service for safer professional and recreational activities in the Mediterranean Sea, Nat. Hazards Earth Syst. Sci. Discuss., doi:10.5194/nhess-2016-176, 2016

Lyubartsev, V., Borile, F., Clementi, E., Masina, S., Drudi, M., Coppini, G., Cessi, P., Pinardi, N. 2020. Interannual variability in the Eastern and Western Mediterranean Overturning Index. In: Copernicus

Marine Service Ocean State Report, Issue 4 – K. Von Schuckmann, P.Y. Le Traon, N. Smith, A. Pascual, S. Djavidnia, J.P. Gattuso, M. Grégoire, G. Nolan (eds). J. Oper. Oceanogr. 13 Suppl. 1., doi:10.1080/1755876X.2020.1785097

Mannarini G., Turrise G., D'Anca A., Scalas M., Pinardi N., Coppini G., Palermo F., Carluccio I., Scuro M., Cretì S., Lecci R., Nassisi P., Tedesco L. VISIR: technological infrastructure of an operational service for safe and efficient navigation in the Mediterranean Sea 2016, Natural Hazards and Earth System Sciences, 16, 1791-1806, DOI: 10.5194/nhess-16-1791-2016, 2016.

Mannarini G., L. Carelli, J. Orović, C. P. Martinkus, and G. Coppini. Towards Least-CO2 Ferry Routes in the Adriatic Sea. Journal of Marine Science and Engineering, 9(2), doi:10.3390/jmse9020115, 2021

Pinardi, N., Lyubartsev, V., Cardellicchio, N., Caporale, C., Ciliberti, S., Coppini, G., De Pascalis, F., D'Alti, L., Federico, I., Filippone, M., Grandi, A., Guideri, M., Lecci, R., Lamberti, L., Lorenzetti, G., Lusiani, P., Macripo, C. D., Maicu, F., Mossa, M., Tartarini, D., Trotta, F., Umgiesser, G., and Zaggia, L.: Marine Rapid Environmental Assessment in the Gulf of Taranto: a multiscale approach, Nat. Hazards Earth Syst. Sci., 16, 2623-2639, doi:10.5194/nhess-16-2623-2016, 2016

5.3 Ocean pattern indicator

Authors: Kevin Balem (Ifremer), Andrea Garcia Juan (Ifremer), Loic Bachelot (Ifremer)

Corresponding authors / maintainer: kevin.balem@ifremer.fr

5.3.1 Short description of the service

The Ocean Patterns Indicator is based on a machine learning approach. It consists in applying a clustering, or classification method called GMM (Gaussian Mixture Model), a probabilistic model, to a set of profiles, from a structured (model output, reanalysis) or an unstructured (set of observations) dataset. Any type of variable can be used: temperature, salinity ... The ocean profiles are automatically gathered into several clusters, or classes, depending on their vertical structure. When analysing the different clusters, spatial and temporal coherences can be revealed, that is what we define as the Ocean Patterns Indicator.

The service offers users a flexible and innovative approach to perform statistical analyses on ocean dataset.

5.3.2 Targeted users

The target audience of the service are the scientific users, providing to them a tool to facilitate the discovery of pattern indicators based on machine learning and a simplified way to analyse oceanographic data.

5.3.3 Step by step guideline to use the service

For the ‘Ocean patterns indicator’, the workflow (Figure 14) is structured in two notebooks: a model development notebook and a prediction notebook. In the model development notebook, the user will download a training dataset, parameter, optimize and train the model, and then save it in a NetCDF file. In the second notebook (prediction notebook), the user will upload the model generated in the first notebook, download the dataset to be predicted and plot the results. The figures and the dataset including the computed variables can be saved in user workspace, in a NetCDF file.

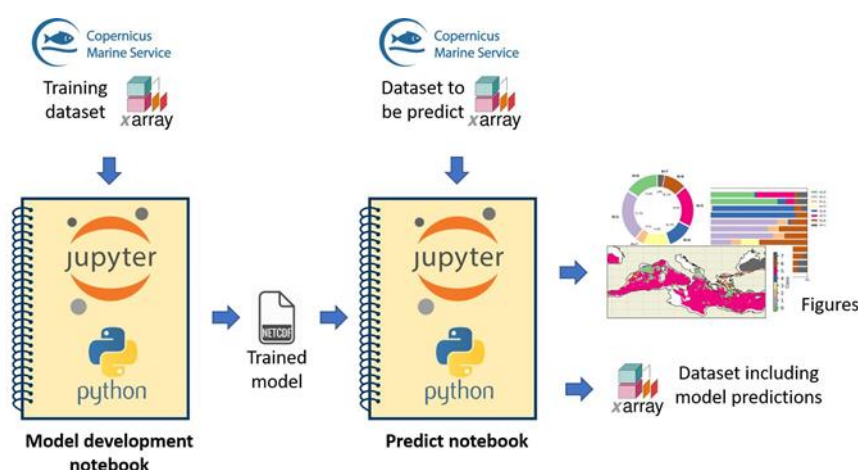


Figure 14 ‘Ocean patterns indicator’ workflow

In order to access one of the notebooks, please follow the instructions:

1. Open a JupyterLab instance using the JupyterHub tab inside the VREVLab
2. Open a Terminal inside the JupyterLab - from the menu > File > New > Terminal
3. Inside the terminal, copy the archive, change the permission and unzip, with the commands:
 - cp workspace/VREFolders/MarineEnvironmentalIndicators/notebooks/OceanPatternsIndicator/OceanPatternsIndicator.zip .
 - chmod 777 OceanPatternsIndicator.zip
 - unzip OceanPatternsIndicator.zip
4. Open the desired notebook - from the menu > Open from Path, using the following path:
 - For the development notebook: OceanPatternsIndicator/Develop_PCM_model.ipynb
 - For the predict and plot notebook: OceanPatternsIndicator/predict_PCMLabels_and_plot.ipynb

5.3.3.A Development notebook

The user should carefully read the descriptions and follow the instructions as presented in the ‘Development’ notebook. This notebook is structured as follows:

a. Model parameters

User should provide the following parameters to design the model: the number of classes (Figure 15 a) and the name of the chosen variable (Figure 15 b).

b. Load training dataset

User should provide CMEMS account user and password (Figure 16 a). The training dataset is downloaded from CMEMS and saved in a NetCDF file in datasets/ folder in the workspace. (Figure 16 b). Finally, the training dataset is loaded locally using xarray library. (Figure 16 c).

c. Create and train model

User can create (Figure 17 a) and train (fit) the model (Figure 17 b) following the instruction in the notebook.

d. Development plots

The user can produce some plots that will be used as a guide to choose the best model parameters. (Figure 18)

e. Refit and save model

Model is trained again with the correct number of classes and saved as a NetCDF file in the models/ folder (Figure 19). It will be used as an input in the next notebook (See following paragraph: Predict and plot notebook).

Model parameters

Model parameters

In this section you will provide the parameters you want to use for designing your model: you should choose the **number of classes** and provide the name that the variable (**feature**) will have in the model.

For the number of classes K you can choose a low number at the beginning (around 6). In the plot section you will optimize the number of classes using the **BIC plot**. Then you will use the optimized number of classes to train the model again.

```
[2]: # number of classes
K=6 a

# name of variable (feature)
var_name_md1 = 'temperature' # in model b
```

Figure 15 Development notebook – Model parameters

In the parameters, the user can introduce the number of classes (a) to train the model in at first and the chosen variable (b) in which he wants to work.

Load training dataset

Choose training dataset

The training dataset is downloaded from CMEMS servers, so you will need to have a **CMEMS account** (you can sign up [here](#)).

You should provide your CMEMS **user name** and **password** below.

```
[ ]: CMEMS_user = '#####'
      CMEMS_password = '#####'
```

a

Data comes from monthly mean fields of *GLOBAL_REANALYSIS_PHY_001_030* product, an eddy-resolving reanalysis with 1/12° horizontal resolution and 50 vertical levels (click [here](#) for getting all the information about the dataset). As an example, we propose to you a selection covering the Mediterranean sea during 2018.

If you feel confident you can modify downloading parameters (cell below) to test other dataset selections than the one we propose here (covering the Mediterranean). You can also test other variables, but do not forget to change variables names in the cell above. And be careful with memory limits: do not choose very big geographical extents or very long time series.

```
[3]: # geographical extent
      geo_extent = [-5, 35, 30, 46] # [min lon, max lon, min lat, max lat]
      # time extent
      time_extent = ['2018-01-01', '2018-12-31'] # ['min date', 'max_date']
      # variable to be predict
      var_name_ds = 'theta' # name in dataset
      # file name
      file_name = 'global-reanalysis-phy-001-030-monthly_med_2018.nc'
```

Load training dataset

Training dataset is download from CMEMS servers using a Motu client and saved as a NetCDF file in *datasets/* folder in your workspace. Downloading will take some minutes.

```
[ ]: !pip install motuclient --upgrade
      bashCommand = 'python -m motuclient -u ' + CMEMS_user + ' -p ' + CMEMS_password + ' -m "http://my.cmems-du.eu/motu-web/Motu" \
      -s GLOBAL_REANALYSIS_PHY_001_030-TDS -d global-reanalysis-phy-001-030-monthly \
      -x ' + str(geo_extent[0]) + ' -X ' + str(geo_extent[1]) + ' -y ' + str(geo_extent[2]) + ' -Y ' + str(geo_extent[3]) + ' \
      -t ' + time_extent[0] + ' -T ' + time_extent[1] + ' -z 0.0 -Z 2500.0 \
      -v so -v ' + var_name_ds + ' -o datasets -f ' + file_name
      sp = subprocess.call(bashCommand, shell=True)
      file_path = 'datasets/' + file_name
```

b

Training dataset is loaded from the NetCDF file using *xarray* library.

```
[5]: ds = xr.open_dataset(file_path)
```

c

Figure 16 Development notebook – Load training dataset.

User provides his/her CMEMS account user and password (a). User can then download the training dataset from CMEMS in a NetCDF file (b) on the local workspace. Finally the training dataset is loaded locally using the xarray library (c).

Create and train model

In this section, you can create your own model using the number of classes *K* and the feature given as input. Then, the model is trained (**fitted**) to the training dataset and profiles are classified (**predict**) in order to make some useful plots in the next section.

Create PCM

```
[8]: # pcm feature
      z = ds[z_dim][0:30]
      pcm_features = {var_name_md1: z}
      m = pcm(K=K, features=pcm_features)
      m

[8]: <pcm 'gmm' (K: 6, F: 1)>
      Number of class: 6
      Number of feature: 1
      Feature names: odict_keys(['temperature'])
      Fitted: False
      Feature: 'temperature'
      Interpolater: <class 'pyxpcm.utils.Vertical_Interpolator'>
      Scaler: 'normal', <class 'sklearn.preprocessing_data.StandardScaler'>
      Reducer: True, <class 'sklearn.decomposition_pca.PCA'>
      Classifier: 'gmm', <class 'sklearn.mixture_gaussian_mixture.GaussianMixture'>
```

a

Fit model

```
[9]: # Variable to be fitted (variable name in model: variable name in dataset)
      features_in_ds = {var_name_md1: var_name_ds}
      m.fit_predict(ds, features=features_in_ds, dim=z_dim, inplace=True)
      m

[9]: <pcm 'gmm' (K: 6, F: 1)>
      Number of class: 6
      Number of feature: 1
      Feature names: odict_keys(['temperature'])
      Fitted: True
      Feature: 'temperature'
      Interpolater: <class 'pyxpcm.utils.Vertical_Interpolator'>
      Scaler: 'normal', <class 'sklearn.preprocessing_data.StandardScaler'>
      Reducer: True, <class 'sklearn.decomposition_pca.PCA'>
      Classifier: 'gmm', <class 'sklearn.mixture_gaussian_mixture.GaussianMixture'>
      log likelihood of the training set: 19.558644
```

b

Figure 17. Development notebook – Create and train model.

User can create (a) and train/fit the model (b) following the instruction in the notebook (text in green preceded by a hashtag '#').

Development plots

The plots in this section will help you to **optimize** the model parameters (specially the number of classes) and to take a look on how the model is working.

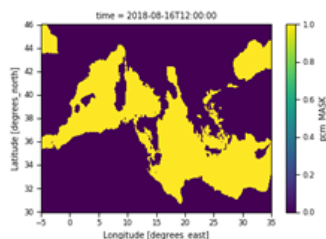
1. Mask

When fitting the model, the pyxpcm software **preprocessed** the data to use profiles without NaN values. NaNs can be found when the profile is not depth enough, for example. Profiles with NaN values are masked and they are not used for fitting the model.

You can plot the mask below to know how is the dataset which is actually used to fit the model. As we are working with a time series, you should choose the time slide to be plotted.

```
[10]: mask = ds.isel(time=7).pyxpcm.mask(m, features=features_in_ds, dim=z_dim)
      mask.plot()
```

```
[10]: <matplotlib.collections.QuadMesh at 0x7f943edec310>
```



2. BIC

The BIC (Bayesian Information Criteria) is used to **optimize the number of classes** in the model, trying not to over-fit or under-fit the data. For calculating this index, the model is fitted to the training dataset for a range of K values from 0 to 20, doing 10 runs each time to calculate the standard deviation. The **minimum** in the BIC curve will give to you the best number of classes to be used.

For each run, a sub-dataset of the training dataset is used, as profiles should not be correlated neither spatially nor temporally to get a minimum. Spatial and temporal correlation in the training dataset are user inputs that should be given below. For our example in the Mediterranean sea, values have been optimized. If you want to try another geographical selection or another variable you may change this numbers.

You can also choose the number of runs and the maximum number of classes, taking in to account that increasing this numbers will increase the computation time. Calculation is parallelized but it still take some minutes.

Figure 18 Development notebook – Development plots.

User can produce some plots that will be a guide to choose the best model parameters.

Refit and save model

As you know now which is the **best number of classes** to be used with your training dataset, you can train (fit) the model again with the good number of classes.

```
[ ]: # good number of classes
      K = 8
      m = pcm(K=K, features=pcm_features)
      m.fit_predict(ds, features=features_in_ds, dim=z_dim, inplace=True)
```

a

If you are happy with you model, you can save it in the *models/* folder and use it in the *predict_PCMlabels_and_plot.IPYTHON* notebook to classify (predict) a dataset and plot the results corresponding to **ocean patterns indicators**.

```
[ ]: m.to_netcdf('models/test_model_mediterranean_temp_2018.nc')
```

c

Figure 19 Development notebook – Refit and save model.

Model is trained again (a) with the correct number of classes (b) and saved as a NetCDF file in the folder (c). It can be used as an input in the next notebook

5.3.3.B Predict and plot notebook

The user should carefully read the descriptions and follow the instructions as presented in the ‘Predict and plot’ notebook. This notebook is structured as follow:

a. Load model and dataset

The user should provide the CMEMS account user and password (Figure 16 a) and the path to the trained model generated in the first notebook (Figure 19c).

User can also use one of the already trained models available in the folder (Figure 20 a). The input dataset is downloaded from CMEMS (a CMEMS user account is need – Figure 20 c) and the model is load from the NetCDF file (Figure 20 b).

b. Predict labels

Classes' prediction is done using pyxpcm library. User should follow notebook instructions (Figure 21).

c. Plot results

Results are plotted in different ways and figures are saved in the figures' folder on the user workspace (Figure 22).

d. Save data

User can also save the dataset including the new computed variables in a NetCDF file (Figure 23).

Load model and dataset

In this section you will upload the **model** and the **dataset** and you should provide some information about them.

You don't need to use the same dataset you used to train the model for making the prediction of labels. You can, for example, train the model with in-situ data and apply it to a numerical model dataset in order to evaluate the numerical model realism.

Load model

You can choose an already trained model, available for you in *models/* folder, or you can design your own model using the *Develop_PCM_model.ipynb* notebook.

In the cell below you should provide the model path and the name in the model of the variable (feature) to be predict.

```
[2]: # Model path
model_path = 'models/test_model_mediterranean_temp_2018.nc'
# Variable to be predict
var_name_md1 = 'temperature' # name in model
```

a

pyxpcm library is used to load the chosen model.

```
[3]: m = pyxpcm.load_netcdf(model_path)
m
```

b

```
[3]: <pcm 'gmm' (K: 10, F: 1)>
Number of class: 10
Number of feature: 1
Feature names: odict_keys(['temperature'])
Fitted: True
Feature: 'temperature'
Interpolator: <class 'pyxpcm.utils.Vertical_Interpolator'>
Scaler: 'normal', <class 'sklearn.preprocessing.data.StandardScaler'>
Reducer: True, <class 'sklearn.decomposition_pca.PCA'>
Classifier: 'gmm', <class 'sklearn.mixture._gaussian_mixture.GaussianMixture'>
log likelihood of the training set: 21.359517
```

Load dataset

Dataset is downloaded from CMEMS servers, so you will need to have a **CMEMS account** (you can sign up [here](#)).

You should provide your CMEMS **user name** and **password** below.

```
[ ]: CMEMS_user = '#####'
CMEMS_password = '#####'
```

c

Figure 20 Predict & Plot notebook – Load model and dataset.

User has to provide the CMEMS account user and password (c). User can use one of the already trained models available in the folder (a). The input dataset is downloaded from CMEMS (CMEMS account needed) and the model is load from the NetCDF file (b).

Predict labels

Classes labels and some statistics are computed using `pyxpcm` library. New variables with the results are added to the dataset (`inplace=True` option).

Predict class labels

Taking into account the characteristics of the classes already determine in the trained model, each profile in the dataset is classified (**predicted**) into one of the classes. A new variable `PCM_LABELS` is created, including one class label for each profile.

```
[8]: features_in_ds = {var_name_md1: var_name_ds}
m.predict(ds, features=features_in_ds, dim=z_dim, inplace=True);
```

Probability of a profile to be in a class (Posteriors)

As `pyxpcm` software is using a GMM (Gaussian Mixture Model) to determine clusters, it is possible to calculate the probability of a profile to belong to a class, also call **posterior**. This is the first step to determine the robustness of the model, that will be calculated below. A new variable `PCM_POST` is created.

```
[9]: m.predict_proba(ds, features=features_in_ds, dim=z_dim, inplace=True);
```

Classes quantiles

Class vertical structure can be represented using the quantiles of all profiles corresponding to a class. We advise you to calculate at least the **median profile** and the 5% and 95% quantiles (`q=[0.05, 0.5, 0.95]`) to have a good representation of the classes, but feel free to add other quantiles if you want. A new variable `outname=var_name_ds + '_Q'` is added to the dataset.

```
[10]: ds = ds.pyxpcm.quantile(m, q=[0.05, 0.5, 0.95], of=var_name_ds, outname=var_name_ds + '_Q', keep_attrs=True, inplace=True)
```

Robustness

Robustness represents the **probability** of a profile to belong to a class, as posteriors, but the value range is more appropriated for graphic representation. Two new variables are added to the dataset: `PCM_ROBUSTNESS` and `PCM_ROBUSTNESS_CAT`.

```
[11]: ds.pyxpcm.robustness(m, inplace=True)
ds.pyxpcm.robustness_digit(m, inplace=True)
```

```
[11]: xarray.Dataset
```

Dimensions: (depth: 41, latitude: 193, longitude: 481, pcm_class: 8, quantile: 3, time: 12)

Coordinates:

pcm_class	(pcm_class)	int64	0 1 2 3 4 5 6 7
depth	(depth)	float32	-0.494025 -1.541375 ...
latitude	(latitude)	float64	30.0 30.08 30.17 ... 4...
time	(time)	datetime64[ns]	2017-01-16T12:00:00...

Figure 21 Predict & Plot notebook – Predict labels.

Plot results

Plots are created using the `Plotter` class, which is instantiate below. Plots include the vertical structure and the spatial and the temporal distribution of classes. These plots would allow you to determine if classes show a spacial or temporal coherence: the **ocean patterns indicators**.

`save_BlueCloud` function save the figure and add dataset information and logos below.

Please, feel free to change plot options if you need it.

```
[12]: P = Plotter(ds, m)
```

1. Vertical structure of classes

The graphic representation of quantile profiles reveals the vertical structure of each class and how clusters are created, as the clustering method is base on finding similarities in the vertical structure of the feature (profiles). The median profiles will give you an idea of the **typical profile** representing each class and the rest of quantiles, the **variability** of the profiles within a class.

```
[13]: P.vertical_structure(q_variable = var_name_ds + '_Q', sharey=True, xlabel='Temperature (°C)')
P.save_BlueCloud('figures/vertical_struc_EX.png')
```

Figure saved in figures/vertical_struc_EX.png

Vertical structure of classes

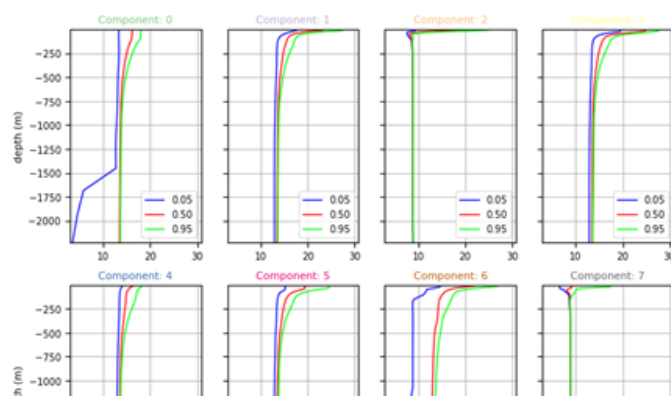


Figure 22 Predict & Plot notebook – Plot results.

Save data

If you are happy with the results and you want to work on the data by your own, you can save the dataset including the new PCM variables (PCM labels, robustness, ...) in the cell below.

```
[21]: ds.to_netcdf('datasets/tests_predicted_dataset.nc')
```

Figure 23 Predict & Plot notebook – Save data.

5.3.4 Data sources

Variables	Data sources	Infrastructure	Access through
Temperature Salinity ...	MEDSEA_REANALYSIS_PHYS_006_004	CMEMS	Blue Cloud VRE or Notebook interface to WEkEO HDA API
Temperature Salinity ...	GLOBAL_REANALYSIS_PHY_001_030	CMEMS	Copernicus (Motu Client)

Table 7 Ocean Pattern Indicator data sources

A subset of product MEDSEA_REANALYSIS_PHYS_006_004 in CMEMS catalogue is made available as a sample input dataset inside the VRE that the user can select from the web interface. The sample input dataset inside the VRE has the same format of the external data source (CMEMS).

For the notebooks, two examples of input datasets are available: a selection of monthly mean fields of GLOBAL_REANALYSIS_PHY_001_030 CMEMS product covering the Mediterranean Sea in 2018 and the same selection in 2017. Input dataset is downloaded from CMEMS using a CMEMS Motu client and saved in the user workspace. The code for downloading data is included in the notebooks. Some pre-trained models are also provided and can be used in the notebooks by the user.

5.3.5 Scientific references

Guillaume Maze et al., Coherent heat patterns revealed by unsupervised classification of Argo temperature profiles in the North Atlantic Ocean, Progress in Oceanography, Volume 151, 2017, Pages 275-292, ISSN 0079-6611, <https://doi.org/10.1016/j.pocean.2016.12.008>.

5.4 Ocean regime indicator

Authors: Kevin Balem (Ifremer), Andrea Garcia Juan (Ifremer), Loic Bachelot (Ifremer)

Corresponding authors / maintainer: kevin.balem@ifremer.fr

5.4.1 Short description of the service

The Ocean Regimes indicator is based on the same clustering method as for the Ocean Pattern indicator, a machine learning approach based on a Gaussian probabilistic method, but applied to a dataset of ocean time series (Chlorophyll-a, SST...). The time series are gathered into clusters depending on their seasonal variability. For this indicator, spatial coherences can be revealed when plotting the different classes in a map. The service offers users a flexible and innovative approach to perform statistical analyses on ocean dataset.

5.4.2 Targeted users

The target audience of the service are the scientific users, providing to them a tool to facilitate the discovery of regime indicators based on machine learning and a simplified way to analyse oceanographic data.

5.4.3 Step by step guideline to use the service

For the 'Ocean regimes indicator', the workflow (Figure 24) is structured in two notebooks: a model development notebook and a prediction notebook. In the model development notebook, the user will download a training dataset, parameter, optimize and train the model, and then save it in a NetCDF file. In the second notebook (prediction notebook), the user will upload the model generated in the first notebook, download the dataset to be predicted and plot the results. The figures and the dataset including the computed variables can be saved in user workspace, in a NetCDF file.

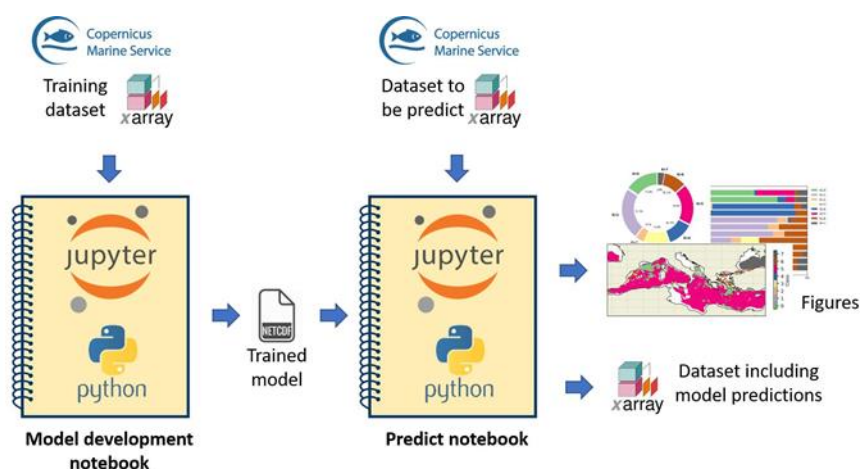


Figure 24 'Ocean regimes indicator' workflow

In order to access one of the notebooks, please follow the instructions:

1. Open a JupyterLab instance using the JupyterHub tab inside the VREVLab
2. Open a Terminal inside the JupyterLab - from the menu > File > New > Terminal
3. Inside the terminal, copy the archive, change the permission and unzip, with the commands:
 - `cp workspace/VREFolders/MarineEnvironmentalIndicators/notebooks/OceanRegimesIndicator/OceanRegimesIndicator.zip .`
 - `chmod 777 OceanRegimesIndicator.zip`
 - `unzip OceanRegimesIndicator.zip`
4. Open the desired notebook - from the menu > Open from Path, using the following path:
 - For the development notebook:
`OceanRegimesIndicator/ModelDevelopment_OceanRegimes.ipynb`
 - For the predict and plot notebook:
`OceanRegimesIndicator/PredictandPlot_OceanRegimes.ipynb`

5.4.3.A *Development notebook*

The user should carefully read the descriptions and follow the instructions as presented in the 'Development' notebook. This notebook is structured as follows:

a. **Model parameters**

Users should provide the following parameters to design the model: the number of classes (Figure 25).

b. **Load training dataset**

Users should provide a CMEMS account user and password (Figure 26 a). The training dataset is downloaded from CMEMS and saved in a NetCDF file in the datasets/ folder in the workspace. (Figure 26 b). Finally, the training dataset is loaded locally using the xarray library. (Figure 26 c).

c. **Preprocessing**

Some preprocessing is necessary in order to train a model. It is split in 5 steps (Figure 27):

- First 2 steps are some subsampling with a weekly average and a latitude/longitude subsampling.
- In the 3rd step, NaNs values are removed using a mask. Here, in the example of the notebook, we use our own Mediterranean mask for removing the Black Sea, the Atlantic Ocean and the lakes.
- The sklearn **StandardScaler** is applied (4th step): it standardizes the *feature* dimension by removing the mean and scaling to unit variance
- 5th step, A PCA (Principal Component Analysis) is applied. The PCA reduces the feature dimensions by projecting the data into a lower dimensional space. With the `n_components` option, you can choose the amount of variance that needs to be explained by the reduced components (default value is 0.99).

d. Create and train model

Users can create (Figure 28) and train (fit) the model following the instruction in the notebook.

e. Development plots

The user can produce some plots that will be used as a guide to choose the best model parameters (Figure 29).

f. Refit and save model

Model is trained again with the correct number of classes and saved as a NetCDF file in the models/ folder (Figure 30). It will be used as an input in the next notebook (See following paragraph: Predict and plot notebook).

Model parameters

In this section you must choose the **number of classes** (K) you want to use to design your model.

You can choose a low number at the beginning (around 6 is a good general start). In the plot section you will optimize the number of classes using the BIC plot. Then you will use the optimized number of classes to train the model again.

K=6

Figure 25 Development notebook – Model parameters.

In the parameters, the user can introduce the number of classes

Load training dataset

Choose training dataset

The training dataset is downloaded from CMEMS servers, so you will need to have a **CMEMS account** (you can sign up [here](#)).

You should provide your CMEMS **user name** and **password** below.

```
[ ]: CMEMS_user = '#####'
      CMEMS_password = '#####'
```

a

Data comes from monthly mean fields of GLOBAL_REANALYSIS_PHY_001_030 product, an eddy-resolving reanalysis with 1/12° horizontal resolution and 50 vertical levels (click [here](#) for getting all the information about the dataset). As an example, we propose to you a selection covering the Mediterranean sea during 2018.

If you feel confident you can modify downloading parameters (cell below) to test other dataset selections than the one we propose here (covering the Mediterranean). You can also test other variables, but do not forget to change variables names in the cell above. And be careful with memory limits: do not choose very big geographical extents or very long time series.

```
[3]: # geographical extent
      geo_extent = [-5, 35, 30, 46] # [min lon, max lon, min lat, max lat]
      # time extent
      time_extent = ['2018-01-01', '2018-12-31'] # ['min date', 'max_date']
      # variable to be predict
      var_name_ds = 'thetao' # name in dataset
      # file name
      file_name = 'global-reanalysis-phy-001-030-monthly_med_2018.nc'
```

Load training dataset

Training dataset is download from CMEMS servers using a Motu client and saved as a NetCDF file in datasets/ folder in your workspace. Downloading will take some minutes.

```
[ ]: !pip install motuclient --upgrade
      bashCommand = 'python -m motuclient -u ' + CMEMS_user + ' -p ' + CMEMS_password + ' -m "http://my.cmems-du.eu/motu-web/Motu" \
      -s GLOBAL_REANALYSIS_PHY_001_030-TDS -d global-reanalysis-phy-001-030-monthly \
      -x ' + str(geo_extent[0]) + ' -X ' + str(geo_extent[1]) + ' -y ' + str(geo_extent[2]) + ' -Y ' + str(geo_extent[3]) + ' \
      -t ' + time_extent[0] + ' -T ' + time_extent[1] + ' -z 0.0 -Z 2500.0 \
      -v so -v ' + var_name_ds + ' -o datasets -f ' + file_name
      sp = subprocess.call(bashCommand, shell=True)
      file_path = 'datasets/' + file_name
```

b

```
[4]: file_path = 'datasets/' + file_name
```

Training dataset is loaded from the NetCDF file using xarray library.

```
[5]: ds = xr.open_dataset(file_path)
```

c

Figure 26 Development notebook – Load training dataset.

User provides his/her CMEMS account user and password (a). User can then download the training dataset from CMEMS in a NetCDF file (b) on the local workspace. Finally the training dataset is loaded locally using the xarray library (c).

Preprocessing

Before using the training dataset, you need to format it. The preprocessing phase consists of 5 different steps that are explained below.

1. Weekly mean

For each time series (each pixel) a weekly mean is applied. It allows us to **smooth the time series**, focusing on the seasonal variability. The dimensions on the dataset are also reduced, so processing time will be shortened. If your dataset contains more than one year, all years are taken together in the weekly mean, creating weekly climatological time series. This "week" dimension will be called "feature" in the preprocessed dataset.

```
X = OR_weekly_mean(ds, var_name=var_name)
```

2. Reduce latitude and longitude to sampling dimension

The dataset is stacked into a **2D dataset** with dimensions "feature" (corresponding to weeks) and "sampling" (corresponding to latitude and longitude). Latitude and longitude information is not taken into account by the model: it only sees a set of "independent" time series and uses the similarities in temporal variability to make the clusters.

```
X = OR_reduce_dims(X)
```

3. Delete all NaN values

NaNs values are removed using a **mask**. Here we use our own Mediterranean mask for removing the Black Sea, the Atlantic Ocean, and the lakes. If you are using a different dataset, you can use your own mask (look at the *docstring* of the function to know mask requirements) or you can automatically produce your mask from the dataset (default option `mask_path='auto'`). The mask will be used below for unstacking the final dataset.

```
#X, mask = OR_delete_NaN(X, var_name=var_name, mask_path='datasets/Mediterranean_mask.nc')
X, mask = OR_delete_NaN(X, var_name=var_name)
```

Quick plot of the mask.

```
mask['mask'].plot()
```

4. Scaler

The sklearn **StandardScaler** is applied: it standardizes the *feature* dimension by removing the mean and scaling to unit variance. For more information, see [sklearn doc](#).

Depending on your dataset you may prefer to use other scalers. You can choose other options using the `scaler_name` input: **Normalizer** and **MinMax Scaler**.

```
X = OR_scaler(X, var_name=var_name)
```

5. Principal Component Analysis (PCA)

A PCA (Principal Component Analysis) is applied. The PCA **reduces the feature dimensions** by projecting the data into a lower-dimensional space. With the `n_components` option, you can choose the amount of variance that needs to be explained by the reduced components (default value is 0.99).

```
X = OR_apply_PCA(X, var_name=var_name)
```

The preprocessed dataset contains a new "**CHL_reduced**" variable with dimensions "feature_reduced" and "sampling" that will be used for training the model.

Figure 27 Development notebook – Preprocessing

Create and train a model

In this section, you can create your own model using the number of classes K . Then, the model is trained (**fitted**) to the training dataset and the time series are classified (**predict**) in order to make some useful plots in the next section.

Create model

We use a GMM (**Gaussian Mixture Models**) which decomposes the PDF (Probability Density Function) of the dataset into a sum of gaussians. For more information, see [Maze et al, Prg.Oc, 2017](#).

```
model = mixture.GaussianMixture(n_components=K, covariance_type='full')
model
```

Fit model

```
X_labels = model.fit_predict(X[var_name + "_reduced"])
X = X.assign(variables={"GMM_labels":(('sampling'), X_labels)})
```

Robustness

The classification robustness is a scaled version of the **probability** of a time series to belong to a class. A 0 value indicates the model is totally unsure of the classification result (all classes are equiprobable), while values close to 1 indicate the model is highly confident of the result.

Robustness will be used in the Development plots section to optimize the model parameters.

```
# calculate probability
X_proba = model.predict_proba(X[var_name + "_reduced"])
X = X.assign(variables={"GMM_proba":(('sampling', 'k'), X_proba)})

# calculate robustness
maxpost = X["GMM_proba"].max(dim="k")
nK = len(X["GMM_labels"])
robust = (maxpost - 1. / nK) * nK / (nK - 1.)
Plist = [0, 0.33, 0.66, 0.9, .99, 1]
rowl0 = ('Unlikely', 'As likely as not', 'Likely', 'Very Likely', 'Virtually certain')
robust_id = np.digitize(robust, Plist) - 1

X = X.assign(variables={"GMM_robustness":(('sampling'), robust), "GMM_robustness_cat":(('sampling'), robust_id)})
X["GMM_robustness_cat"].attrs['legend'] = rowl0

print(X)
```

Finally, the dataset including the results is **unstacked**.

```
ds_labels = OR_unstack_dataset(ds_attrs, X, mask)
print(ds_labels)
```

Figure 28 Development notebook – Create model and train

Development plots

The plots in this section will help you to **optimize** the model parameters (especially the number of classes) and to take a look at how the classification behaves. Some of the functions contained in the `Plotter_OR` class will be used in this section.

```
P = Plotter_OR(ds_labels, model)
```

1. Scatter plot

The histograms of the **first 2 components of the reduced variable** are plotted in the diagonal subplots and a scatter plot is shown in the upright subplot. Data is colored depending on the different classes. The 2 first components of the reduced variable contain most of the information in the dataset. So, this plot will help you to understand your dataset distribution and how the model is constructing the clusters. As we are using a GMM, the model separates the dataset PDF (Probability Density Function) into a sum of gaussians to define each cluster.

```
P.scatter_PDF(var_name = var_name + '_reduced')
P.save_BlueCloud('figures/scatter_PDF_EX_ch1.png')
```

2. BIC

The BIC (**B**ayesian **I**nformation **C**riteria) can be used to **optimize the number of classes** in the model, trying not to over-fit or under-fit the data. To compute this index, the model is fitted to the training dataset for a range of K values from 0 to 20. A **minimum** in the BIC curve will give you the optimal number of classes to be used.

Moreover, for each K range run, a subset of the training dataset is randomly selected in order to use *independent* time series. Indeed, the ocean exhibits spatial correlations that reduce the real information contained in the training dataset. This has to be taken into account. This turns to our advantage here because the grid of the dataset allows us to draw several subsets of uncorrelated time series, finally allowing us to compute several times each K range run and hence to compute a standard deviation on the BIC metric.

The spatial correlation scale to consider is determined by the user. Values in the cell below are ok for the Mediterranean sea example shown here. If you want to try another geographical selection or another variable you may change these numbers.

You can also choose the number of runs and the maximum number of classes, taking into account that increasing these numbers will increase the computation time.

User input

```
corr_dist = 40 # correlation distance in km
Nrun = 10 # number of runs for each k
NK = 20 # max number of classes to explore
```

BIC calculation

Calculation can take a few minutes.

```
BIC, BIC_min = BIC_calculation(X=X, coords_dict={'latitude':'lat', 'longitude':'lon'},
                              corr_dist=corr_dist,
                              feature_name='feature_reduced', var_name='CHL_reduced',
                              Nrun=Nrun, NK=NK)
```

BIC plot

```
plot_BIC(BIC, NK=NK)
P.save_BlueCloud('figures/BIC_EX_ch1.png', bic_fig='yes')
```

Optimized number of classes

```
BIC_min
```

If this is not the number of classes you chose in the beginning of the notebook, don't worry, you will fit your model again with the appropriate number of classes at the end of the notebook.

The BIC curve may not show a clear minimum. This can be an indication that some time series remained correlated in the training set, so try to adjust more precisely the correlation scale.

If the BIC curve has a clear minimum, don't forget to take into account the standard deviation. The BIC curve indicates a statistical optimum, so if the minimum is not above the standard deviation range, then it is indicative of an optimal **range** rather than a precise value. In this case, use your expertise to choose the number of classes (within the BIC allowed range) leading to ocean regimes that simply make the most sense to you.

3. Robustness

The model robustness represents a useful scaled **probability** of a time series to belong to a class. If a lot of time series show very low values you should maybe change the number of classes.

```
P.plot_robustness()
P.save_BlueCloud('figures/robustness_EX_ch1.png')
```

Figure 29 Development notebook – Development plots.

Users can produce some plots that will be a guide to choose the best model parameters.

Refit and save model

As you now know which is the **best number of classes** to classify your dataset profiles, you can train (fit) the model again with the appropriate number of classes.

```
# appropriate number of classes
#K = BIC_min
K = 7

model = mixture.GaussianMixture(n_components=K, covariance_type='full')
model = model.fit(X[var_name + "_reduced"])
```

If you are satisfied with your model, you can save it in the *models/* folder and use it with the `PredictandPlot_OceanRegimes.ipynb` notebook to classify (predict) a dataset and plot the results corresponding to **Ocean Regimes Indicator**.

```
joblib.dump(model, 'models/test_SST_k7.sav')
```

Figure 30 Development notebook – Refit and save model.

Model is trained again with the correct number of classes and saved in the folder. It can be used as an input in the next notebook

5.4.3.B Predict and plot notebook

The user should carefully read the descriptions and follow the instructions as presented in the ‘Predict and plot’ notebook. This notebook is structured as follows:

a. Load model and dataset (Figure 31)

The user should provide the CMEMS account user and password and the path to the trained model generated in the first notebook.

Users can also use one of the already trained models available in the folder. The input dataset is downloaded from CMEMS (a CMEMS user account is needed) and the model is loaded from the .sav file with Joblib.

b. Preprocessing

Some preprocessing is necessary in order to train a model. It is split in 5 steps (Figure 27, same preprocessing as the development notebook):

- First 2 steps are some subsampling with a weekly average and a latitude/longitude subsampling.
- In the 3rd step, NaNs values are removed using a mask. Here we use our own Mediterranean mask for removing the Black Sea, the Atlantic Ocean and the lakes.
- The sklearn **StandardScaler** is applied (4th step): it standardize the *feature* dimension by removing the mean and scaling to unit variance
- In step 5, A PCA (Principal Component Analysis) is applied. The PCA reduces the feature dimensions by projecting the data into a lower dimensional space. With the `n_components` option, you can choose the amount of variance that needs to be explained by the reduced components (default value is 0.99).

c. Predict labels

Classes’ prediction is done using the loaded model. Users should follow notebook instructions (Figure 32).

d. Plot results

Results are plotted in different ways and figures are saved in the *figures/* folder on the user workspace (Figure 33).

e. Save data

User can also save the dataset including the new computed variables in a NetCDF file (Figure 34).

Load model and dataset

In this section, you will download the **model** and the **dataset** and you should provide some information about them.

You don't need to use the same dataset you used to train the model for making the prediction of labels. You can, for example, train the model with in-situ data and apply it to a numerical model dataset in order to evaluate the numerical model realism.

Load model

You can choose an already trained model, available for you in `models/` folder, or you can design your own model using the `ModelDevelopment_OceanRegimes.ipynb` notebook.

In the cell below you should provide the model path.

```
# Model path
model_path = 'models/test_modelOR_mediterranean_chl_2019_k7.sav'
```

Load the chosen model.

```
model = joblib.load(model_path)
K = model.n_components
model
```

Load dataset

Dataset is downloaded from CMEBS servers, so you will need to have a **CMEBS account** (you can sign up [here](#)).

You should provide your CMEBS **user name** and **password** below.

```
CMEBS_user = '#####'
CMEBS_password = '#####'
```

Data comes from daily Chlorophyll-a fields of `OCEANCOLOUR_GLO_CHL_L4_REP_OBSERVATIONS_009_082` product, based on a space-time interpolation and a multi-sensors approach and with a spatial resolution of 4km (click [here](#) for getting all the information about the dataset). As an example, we propose to you a selection covering the Mediterranean sea during 2019.

If you feel confident you can modify downloading parameters (cell below) to test other dataset selections than the one we propose here (covering the Mediterranean). You can also test other variables. Please, be careful with memory limits: do not choose very big geographical extents or very long time series.

```
# geographical extent
geo_extent = [-5, 42, 30, 46] # [min lon, max lon, min lat, max lat]
# time extent
time_extent = ["2019-01-01", "2019-12-31"] # ["min date", "max date"]
# variable to be predict
var_name = 'CHL' # name in dataset
# file name
file_name = 'oceancolour_glo_chl_l4_rep_observations_009_082_2019.nc'
```

Dataset is downloaded from CMEBS servers using a Motu client and saved as a **NetCDF** file in the `datasets/` folder in your work space. Download will take a few minutes.

Figure 31 Predict & Plot notebook – Load model and dataset.

User has to provide the CMEBS account user and password. Users can use one of the already trained models available in the folder. The input dataset is downloaded from CMEBS (CMEBS account needed) and the model is loaded from the .sav file.

Predict labels

Results of the classification (e.g. labels and some other statistics) are computed in the cells below. These new variables are simply added to the preprocessed dataset `X`.

Predict class labels

The trained model instance (here called `model`) contains all the necessary information to classify time series from the dataset we just loaded. Each time series in this dataset will be attributed (**predicted**) to one of the model classes. A new variable `GMM_labels` is created to host this result in the preprocessed dataset `X`.

```
X_labels = model.predict(X[var_name + ".reduced"])
X = X.assign(variables={"GMM_labels": ("sampling", X_labels)})
print(X)
```

Probability of a profile to be in a class

As the GMM (**Gaussian Mixture Model**) is a fuzzy classifier, it is possible to calculate the probability of a time series to belong to a class, also called **posterior**. This is the first step to determine the robustness of the classification, which will be calculated below. A new variable `GMM_post` is created.

```
X_proba = model.predict_proba(X[var_name + ".reduced"])
X = X.assign(variables={"GMM_post": ("sampling", X_proba)})
print(X)
```

Classes quantiles

Class time series structure can be represented using the quantiles of all time series corresponding to a class. We advise you to calculate at least the **median profile** and the 5% and 95% quantiles (`q=[0.05, 0.5, 0.95]`) to have a minimal representation of the classes but feel free to add other quantiles if you want. You have the option to plot the scaled variable if you prefer. A new variable `VAR_q` `VAR_q = "CHL_q"` is added to the dataset.

```
# quantiles we want to calculate
q = [0.05, 0.5, 0.95]
# we can use the normal variable or the scaled variable
var_q = var_name # or var_q = "CHL_scaled"
```

```
k_values = np.unique(X[GMM_labels].values)
nan_matrix = np.empty((K, np.size(q), np.size(X.feature)))
nan_matrix[:] = np.NaN
m_quantiles = xr.DataArray(nan_matrix, dims=['k', 'quantile', 'feature'])
for yi in range(K):
    if yi in k_values:
        m_quantiles[yi] = X[var_q].where(X[GMM_labels]==yi, drop=True).quantile(q, dim='sampling')
X = X.assign(variables={var_q + ".0": ("k", 'quantile', 'feature'), m_quantiles})
X = X.assign_coords(coords={"quantile": q})
print(X)
```

Robustness

The classification robustness is a scaled version of the **probability** of a time series to belong to a class (i.e. the posterior) so that the value range is more appropriate to assess the **robustness** of a classification. A 0 value indicates the model is totally unsure of the classification result (all classes are equiprobable), while values close to 1 indicate the model is highly confident of the result. Note that this does not prevail the scientific mean of the classification, but rather indicates the ability of the model to attribute a time series to a specific class with confidence.

Two new variables are added to the dataset: `GMM_robustness` and `GMM_robustness_cat`. The 2nd variable is categorical and is based on the IPCC likelihood scale:

Robustness range	Category
0-33%	Unlikely
33%-66%	As likely as not
66%-90%	Likely
90%-99%	Very likely
99%-100%	Virtually certain

```
maxpost = X[GMM_post].max(dim='k')
nK = len(X[GMM_labels])
robust = (maxpost - 1. / nK) * nK / (nK - 1.)
Plist = [0, 0.33, 0.66, 0.9, .99, 1]
robust_id = np.digitize(robust, Plist) - 1
X = X.assign(variables={"GMM_robustness": ("sampling", robust), "GMM_robustness_cat": ("sampling", robust_id)})
X[GMM_robustness_cat].attrs['legend'] = rowid
print(X)
```

Finally, the dataset including results is **unstacked**.

```
ds_labels = OR_unstack_dataset(ds_attrs, X, mask)
print(ds_labels)
```

Figure 32 Predict & Plot notebook – Predict labels.

Plot results

Plots are created using the `Plotter_OR` class, which is instantiated below. Plots include the time series structure and the spatial distribution of classes. These plots will allow you to determine if classes show a spatial coherence: the **Ocean Regimes Indicator**.

The `save_BlueCloud` function saves the figure and adds dataset information and logos below.

Please, feel free to change plot options if you need it.

```
P = Plotter_OR(ds_labels, model)
```

1. Time series structure

The graphic representation of quantile time series reveals the seasonal structure of each class. The median time series will give you the best idea of the **typical time series** of a class and the other quantiles, the possible **spread** of time series within a class. You can choose the start month in the plot: here we use `start_month=6` to highlight the seasonal blooms.

```
P.series_structure(q variable = var_q + ".0", ylabel="Chlorophyll-a (mg m-3)")
P.save_BlueCloud('figures/tseries_struc_EX_chl.png')
```

Quantiles can also be plotted **together** to highlight differences between classes. Using the `plot_q` option you can choose the quantiles you want to plot together.

```
P.series_structure_comp(q variable = var_q + ".0", plot_q= 'all', ylabel="Chlorophyll-a (mg m-3)")
P.save_BlueCloud('figures/tseries_struc_comp_EX_chl.png')
```

2. Spatial distribution of classes

You can also plot the GMM labels in a map to analyse the spatial coherence of classes. The spatial information (coordinates) of time series is not used to fit the model, so spatial coherence appears naturally, revealing seasonal structure similarities between different areas of the ocean. If you detect any spatial coherence, well done, you have found an **Ocean Regimes Indicator**.

```
P.spatial_distribution()
P.save_BlueCloud('figures/spatial_distr_EX_chl.png')
```

3. Robustness

Robustness is a scaled **probability** of a time series to belong to a class. When looking at the spatial distribution of the robustness metric, and if classes have a spatial structure, you may encounter regions with high probabilities: these regions are the "core" of the class.

```
P.plot_robustness()
P.save_BlueCloud('figures/robustness_EX_chl.png')
```

4. Classes pie chart

Here you can plot a pie chart showing the percentage of profiles belonging to each class and the number of classified profiles.

```
P.pie_classes()
P.save_BlueCloud('figures/pie_chart_EX_chl.png')
```

Figure 33 Predict & Plot notebook – Plot results.

Save data

If you are happy with the results and you want to work on the data on your own, you can save the dataset including the new GMM variables (GMM labels, robustness, ...) in the cell below.

```
ds_labels.to_netcdf('datasets/OR_mediterranean_2019_CHL_predicted_dataset.nc')
```

Figure 34 Predict & Plot notebook – Save data.

5.4.4 Data sources

Variables	Data sources	Infrastructure	Access through
Temperature Salinity ...	MEDSEA_REANALYSIS_PHYS_006_004	CMEMS	Blue Cloud or Notebook interface to WEKEO HDA API
Temperature Salinity ...	GLOBAL_REANALYSIS_PHY_001_030	CMEMS	Copernicus (Motu Client)

Table 8 Ocean Regime indicator - data sources

A subset of product MEDSEA_REANALYSIS_PHYS_006_004 in CMEMS catalogue is made available as a sample input dataset inside the VRE that the user can select from the web interface. The sample input dataset inside the VRE has the same format of the external data source (CMEMS).

For the notebooks, two examples of input datasets are available: a selection of monthly mean fields of GLOBAL_REANALYSIS_PHY_001_030 CMEMS product covering the Mediterranean Sea in 2018 and the same selection in 2017. Input dataset is downloaded from CMEMS using a CMEMS Motu client and saved in the user workspace. The code for downloading data is included in the notebooks. Some pre-trained models are also provided and can be used in the notebooks by the user.

5.4.5 Scientific references

D'Ortenzio, F. and Ribera d'Alcalà, M.: On the trophic regimes of the Mediterranean Sea: a satellite analysis, Biogeosciences, 6, 139–148, <https://doi.org/10.5194/bg-6-139-2009>, 2009.

Mayot, N., D'Ortenzio, F., Ribera d'Alcalà, M., Lavigne, H., and Claustre, H.: Interannual variability of the Mediterranean trophic regimes from ocean color satellites, Biogeosciences, 13, 1901–1917, <https://doi.org/10.5194/bg-13-1901-2016>, 2016.

5.5 Storm severity index

Authors: Jan Willem Noteboom (KNMI)

Corresponding authors / maintainer: janwillem.noteboom@knmi.nl

5.5.1 Short description of the service

The Storm Severity Index (SSI) service calculates maps and time series of exceptional atmospheric wind or storm circumstances that can impact seas such as the Mediterranean Sea.

The SSI service can be used to study individual storms or storm/SSI distributions for a given area (in the Mediterranean Sea) and period of time (e.g. a winter season or 30 years of storm climatology). In addition, series of SSI distributions can be calculated using a time step (e.g. every year/month over the entire chosen period).

The level of wind speed above which impact is expected can be indicated using a wind speed threshold value. For this wind speed threshold value, percentiles (e.g. P98 with minimum value) can be selected. These percentiles use specific threshold values for each location (grid-cell). Alternatively, a fixed wind speed threshold value can be given for the entire area.

Each calculated map or time series can be plotted and saved to a file.

5.5.2 Targeted users

The target audience of this SSI service are scientific users who are looking for a quantitative impact modelling of severe wind/storms for different areas in the Mediterranean Sea region and for different time periods, up to 40 years (1979 - 2020).

The SSI service offers the scientific users insight about impact of severe wind or storms and the ability to combine or correlate the calculated SSI with other marine environmental indicators.

5.5.3 Step by step guideline to use the service

The Storm Severity Index (SSI) service consists of a notebook and uses hourly Copernicus C3S ERA5 reanalysis data (10 m wind data above sea) of the Mediterranean Sea area (0.5x0.5 deg) for the time period 1 January 1979 till 31 December 2020 as source data (input dataset file in NetCDF format). In addition, the notebook uses an input file (NetCDF format) that contains for each grid cell of the input dataset file the corresponding wind speed percentile values P90, P95, P98 and P99 (Mediterranean Sea area, 40 years time period).

Note: the input file that contains the wind speed percentiles data has been derived from the dataset input file using a special Python script that is not included in the SSI release.

The source or input data are used to calculate daily SSI grid data (1) that are subsequently aggregated to SSI maps and time series for a given time period, timestep and area (2).

$$SSI_{k,day} = \sum_{t=1}^{T=24h} [(max(0, \frac{v_{k,t}}{v_{threshold}} - 1)^3] \quad (1)$$

k is a single grid cell of the Mediterranean Sea area

$$SSI_{j,step} = \sum_{t=1}^{Stepsize} SSI_{j,day} \quad (2)$$

j is a single grid cell of the selected area, *stepsize* is the specified timestep (in days)

The *wind speed threshold* specifies the level above which the wind speed is expected to impact the marine environment ($SSI > 0$). The wind speed threshold can be a *percentile* or a *fixed value*.

- A **percentile** specifies a wind speed threshold value for each grid cell and represents the wind speed at a percentage level (P90, P95, P98 or P99) of a sorted (low-high) collection of wind speed values (ERA5). This results in higher threshold values for grid cells in more windy areas. Windspeed values above the threshold are regarded exceptional and therefore causing impact. The sorted collection of windspeed values of a grid cell consists of hourly windspeed values for the period 1 January 1979 till 31 December 2020. E.g. P98 means that only 2% of the windspeed values exceeded the threshold value for that area (grid cell) in the last 40 years.
- A **fixed value** specifies a fixed windspeed threshold value for all grid cells in the area.

In addition to SSI maps, SSI time series are calculated and contain per time step the total SSI value for the entire area.

To summarize, the calculated output data by the notebook consist of the following SSI map data (grid) for the area and time period specified per time step:

1. *SSItotal*, the total cumulative SSI value (grid cell)
2. *SSImax*, the maximum SSI value (day)
3. *SSIrte*, the average SSI value (zero values excluded)
4. *SSIday*, number of days the SSI value is greater than zero

In addition, the SSI time series data are calculated and consist (for the entire area) of *SSItotal*, *SSImax*, *SSIrte* and *SSIday*.

Both the calculated map data (grid data) and time-series data are stored in one output dataset file (NetCDF format). Plots for the maps and time series can be generated and are saved in separate files (PNG format).

Installing and running the SSI notebook

To **install** the SSI notebook, please follow the installation instructions in README in /workspace/VREFolders/MarineEnvironmentalIndicators/notebooks/StormSeverityIndicator/



To **execute** the SSI notebook, open the notebook and perform a **step by step execution**.

After the SSI initialization steps and the checking of the source data, the user is asked to provide input data using input widgets (figure 35)


Enter NAME of outputfile

OutputFile:


Select TIME INTERVAL between 1979-01-01 and 2021-01-01


StartDate:  EndDate: 

For SERIES of SSI maps, select unit and size of the timestep

StepUnit:  StepSize: 1

Select AREA for latitude 30.0 till 46.0 and longitude -6.0 till 36.0

Lat(deg):  32.0 – 43.5

Long(deg):  -5.0 – 34.0

SSI windspeed threshold – Percentile (with min. value) or Fixed value

Select: ☐ Percentile ☒ Percentile with min. ☐ Fixed

Percentile: ☐ P90 ☐ P95 ☒ P98 ☐ P99

Fixed threshold value (or min. perc value) in m/s

*** NOTE: Please MOVE to the NEXT CELL to activate your input setting ***

Figure 35 – Storm Severity Index (SSI) user input

The following user input data can be entered:

1. Name of the output file
Used for the output data file (NetCDF) and the plot files (PNG).
2. Time interval between 1 January 1979 and 31 December 2020.
3. Stepsize (optional). A timestep can be given in days, months or years.
4. Area bounding box: any subset of the Mediterranean Sea. This is currently limited to the Mediterranean Sea area only.
5. Wind speed threshold
Percentiles P90, P95, P98, and P99 per grid cell with a minimum value or a fixed windspeed threshold value for the entire area

After entering your input **please make sure that you move to the next notebook cell to activate it.**

An overview of the calculation input is given in the next step (figure 36) just before the calculation step.

```
Forbidden characters in outputfile replaced by underscore
Output filename: output/SSI_output12345.nc
SUMMARY OF SSI CALCULATION PARAMETERS (INPUT):
Start: 2010
End(including): 2016
Stepsize: 1 Years
Number of steps (maps) = 7
Latitude range: 32.0 43.5
Longitude range: -5.0 34.0
Select TIME INTERVAL between 1979-01-01 and 2021-01-01
Windspeed threshold percentile P98 with minimum value 15.0
```

Figure 36 Calculation Input parameters.

After you start the SSI calculation step a progressing bar is presented (figure 37)

Calculating: 

Figure 37 Calculation in progress

When the calculations are completed, the output data is saved in a NetCDF file

STORAGE OF CALCULATED SSI DATA
 SSI grid data (steps, cells latitude, cells longitude : (7, 23, 78)) and time series data saved to file output/SSI_output12345.nc

Now the user can specify which maps and time series plots to create and save in PNG files (figure 38)

SELECT SSI variables to plot

☒ SSItotal ☐ SSImax ☐ SSInrdays ☐ SSirate

SELECT type to plots

☒ Create map plots ☒ Create time serie plots

SET number of plots limit (0 is unlimited, 100 absolute maximum)

Limit:

*** NOTE: Please MOVE to the NEXT CELL to activate your plotting setting ***

Figure 38 User plotting input

After entering the plotting details, **please move to the next notebook cell to activate the plotting settings**. The plots are generated and saved in PNG files (figure 39).

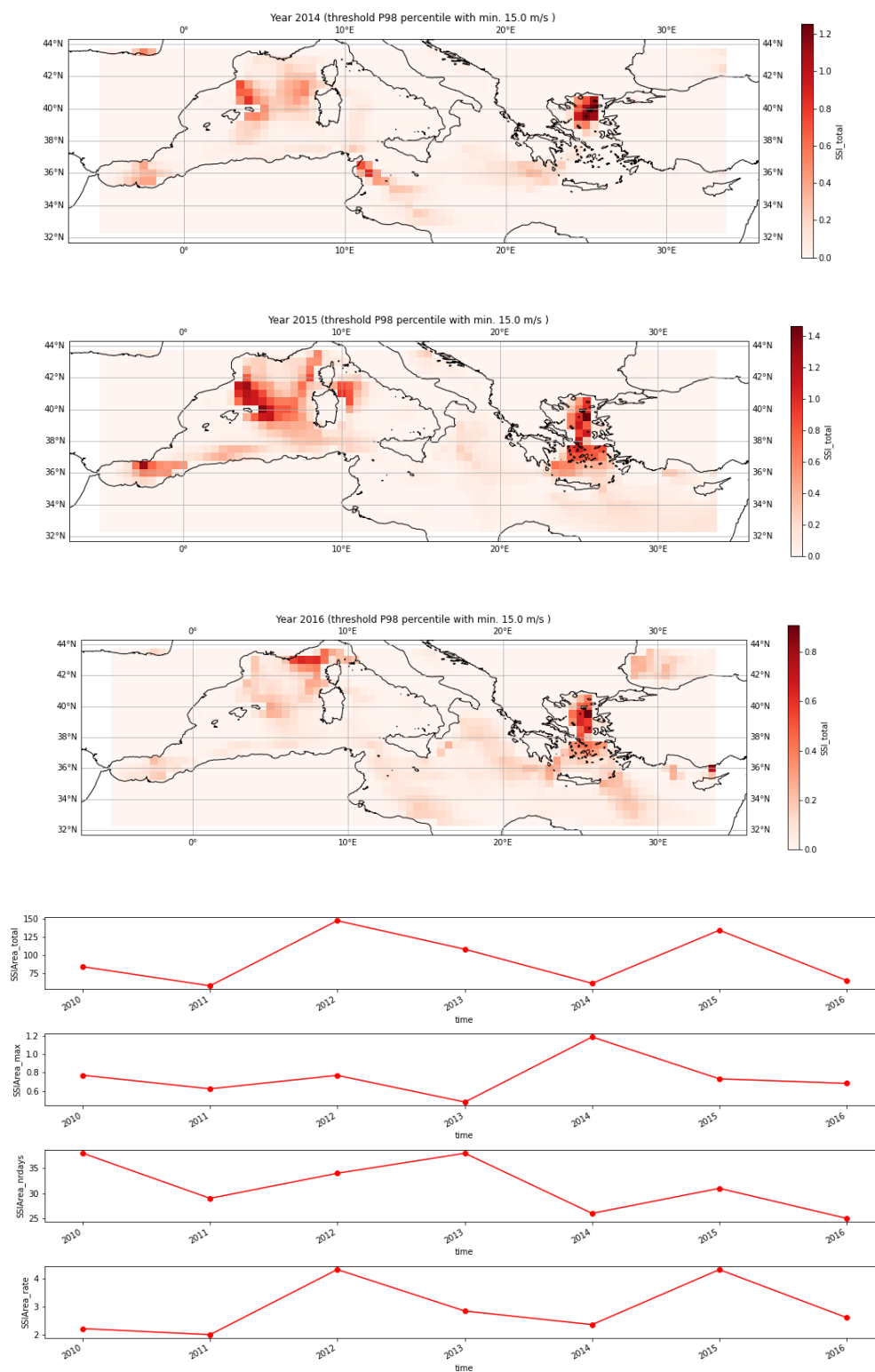


Figure 39 Map plots and time series plots

5.5.4 Data sources

Variables	Data sources	Infrastructure	Access through
<i>hourly windspeed data</i>	C3S_ERA5_Medsea_1979_2020_allmonths_alldays.nc (NetCDF file)	C3S (Copernicus Climate Service)	C3S or Blue Cloud VLab
<i>windspeed percentile values P90, P95, P98 and P99</i>	C3S_ERA5_Medsea_1979_2020_windspeed_P90959899.nc (NetCDF file)	KNMI	Blue Cloud VLab

Table 9 Storm severity index - data sources

NetCDF file *C3S_ERA5_Medsea_1979_2020_allmonths_alldays.nc* contains hourly windspeed data that have been downloaded from the Copernicus C3S Climate Data Store (CDS) to Blue Cloud VRE using dataset *ERA5 hourly data on single levels from 1979* where parameter *Ocean surface stress equivalent 10m neutral wind speed* has been selected for the Mediterranean Sea area (latitude 30.0 to 46.0, longitude -6.0 to 36.0, 0.5x0.5 deg grid) and for the time period 1 January 1979 till 31 December 2020.

NetCDF file *C3S_ERA5_Medsea_1979_2020_windspeed_P90959899.nc* contains the calculated windspeed percentile values P90, P95, P98 and P99 for the area and period that is contained in the hourly windspeed data file. The calculation of percentiles has been done by a special Python script that is not available on Blue Cloud VLab.

5.5.5 Scientific references

Walz MA, Kruschke T, Henning W, Ulbrich U, Leckebusch GC. (2017). Quantifying the extremity of windstorms for regions featuring infrequent events. <https://doi.org/10.1002/asl.758>

Leckebusch GC, Renggli D, Ulbrich U. (2008). Development and application of an objective storm severity measure for the Northeast Atlantic region. *Meteorologische Zeitschrift* 17(5): 575–587. DOI:[10.1127/0941-2948/2008/0323](https://doi.org/10.1127/0941-2948/2008/0323)

5.6 Simple access to carbon data

Authors: Rocío Castaño-Primo, Steve Jones, Benjamin Pfeil (University of Bergen)

Corresponding author / maintainer: rocio.primo@uib.no

5.6.1 Short description of the service

The service provides information on how to search for and retrieve subsets of inorganic carbon data without having to download the full file(s) by using a data server largely used by the oceanographic community.

The data server ERDDAP, developed by the NOAA, allows users to explore, select and download subsets of data, in their preferred format, regardless of the format of origin. It standardizes the variable names and units of position (latitude, longitude, altitude/depth) and time. It can be accessed via script and removes the need of downloading ("copy" or "replicate" in BlueCloud) multiple and/or large files that may or may not be of interest to the user. Any kind of data can be used either gridded (as model data) or discrete (as in situ data). Many marine research data holders currently have ERDDAP servers; some examples in Europe are EMODnet, IFREMER, ICOS, EMSO, Irish Marine Institute, BODC...

5.6.2 Step by step guideline to use the service

The carbon data Jupyter notebooks provide sample scripts that show how to explore and retrieve inorganic carbon data using ERDDAP servers, merge different sources in a single dataset and make simple visualization. For this particular example we picked pH data, since it is the variable most widely available and two European ERDDAP servers: EMODnet and IFREMER.

The service consists of two notebooks: one for searching, selecting, downloading and writing pH data into a (comma-separated) file and another one to further explore the data retrieved in the first notebook: plot, compare and identify duplicates. The README file and the notebook notes (<https://data.d4science.net/eZst>) provide instructions to run the notebooks and explanations to each block of code.

The code can be freely changed by the user to explore different servers, carbon variables and/or time and geographical constraints. Two main switches are set at the beginning of each notebook and explained.

Carbon_data_from_ERDDAP.ipynb: We show how to search datasets within an ERDDAP server following particular criteria; in this case, the CF standard name of pH. Before downloading the expected data, we retrieve some important metadata of the dataset by exploring the attributes of the variables or using external sources such as APIs. This is done for both ERDDAP servers. There is a switch at the beginning set to *False* because that particular piece of code is time consuming. However, the user can set it to *True* freely; it will not affect subsequent code. Finally, the data selected from a particular time frame and quality control flag (in the current version of the notebooks, only data that are good or probably good are picked) are downloaded and saved in two files (data from one and the other ERDDAP servers) at the same path as the notebooks.

Merging_plotting_carbon_data.ipynb: This notebook starts with a switch to either run the *Carbon_data_from_ERDDAP.ipynb* notebook or read from the files generated by that notebook. It merges the two dataframes (files), uses a few methods to identify and remove duplicates and plots maps and time series (Figure 40).

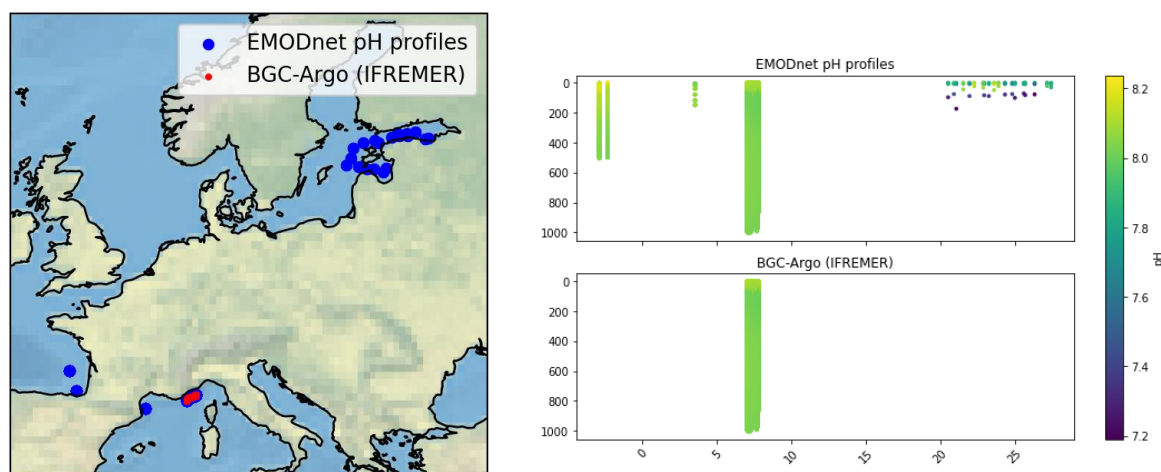


Figure 40 Geographical distribution (left) and longitude-depth scatterplot of pH data (right) from two sources and through the ERDDAP servers used in the notebooks.

Note to the users: due to how the BGC-Argo data is structured in the ERDDAP server, the data search as posed in the notebooks in rare cases take a bit longer and time out; just re-run the cell again.

5.6.3 Data sources

The service offers an example on how to access carbon data hosted in different ERDDAP servers and therefore no data files are provided with the notebooks.

The example developed in the Jupyter notebooks makes use of pH data retrieved from two European ERDDAP servers, the IFREMER and the EMODnet ones.

Variables	Data sources	Infrastructure	Access through
pH	Argo float synthetic vertical profiles : BGC data	Euro-ARGO	IFREMER (ERDDAP) or Blue cloud (DD&AS)
pH	EMODnet Physics - Collection of Ph (PHPH) Profiles - MultiPointProfileObservation	EMODnet	EMODnet (ERDDAP)

Table 10 Simple access to carbon data – data sources

There is a direct access to the data where they are (i.e. IFREMER, EMODnet) through ERDDAP as this has been one objective of the service to show the capabilities of ERDDAP on Blue Cloud (VRE).

5.6.4 Scientific references

Bittig HC, Maurer TL, Plant JN, Schmechtig C, Wong APS, Claustre H, Trull TW, Udaya Bhaskar TVS, Boss E, Dall’Olmo G, Organelli E, Poteau A, Johnson KS, Hanstein C, Leymarie E, Le Reste S, Riser SC, Rupan AR, Taillandier V, Thierry V and Xing X (2019) A BGC-Argo Guide: Planning, Deployment, Data Handling and Usage. *Front. Mar. Sci.* 6:502. doi: 10.3389/fmars.2019.00502

Simons, R.A. 2020. ERDDAP. <https://coastwatch.pfeg.noaa.gov/erddap> . Monterey, CA: NOAA/NMFS /SWFSC/ERD.

6 Demonstrator # 4 – Fish, a matter of scales

6.1 Fisheries atlas

Authors: Anton Ellenbroek, Emmanuel Blondel (FAO), Julien Bard (IRD)

6.1.1 Short description of the service

The service concerns an online **fisheries atlas** of EU waters and beyond, providing a harmonised time-series of catch, commodities and trade data. This atlas is an expansion of the FAO Tuna Atlas, it is scalable and offers to users features for data analysis using indicators, statistics, interactive maps, etc (Figure 41).

This is an online overview of fisheries data accessible through a map Viewer, ISO/OGC metadata and data services, analytical and reporting tools and R Shiny, Jupyter and Markdown reporting services.

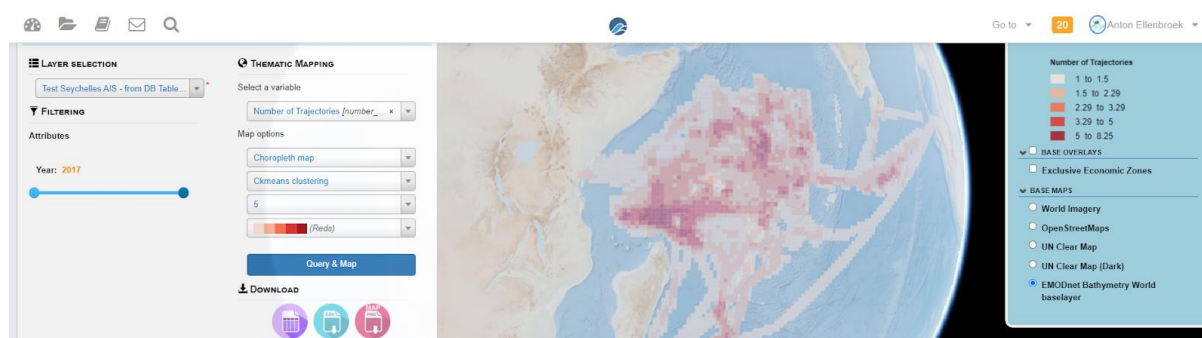


Figure 41 A web snapshot of the fisheries atlas

The objective is to have this service open and public, currently this is not the case and a user registered to the Blue Cloud VRE needs to ask additional login codes to the service manager to be able to access the fisheries atlas.

6.1.2 Targeted users

End-users: general public with an interest in fisheries dynamics, fish provenance, fisheries distribution, fisheries and SDG 2 and SDG 14 who can have access to the information through atlases, API's or QR codes.

Most of the demonstrator services operate through the OpenFairViewer and end-users are isolated from the data-preparation and validation phases.

Fisheries Managers: regional fisheries managers who require access to overviews of fisheries to inform their management decision making processes (Figure 42).

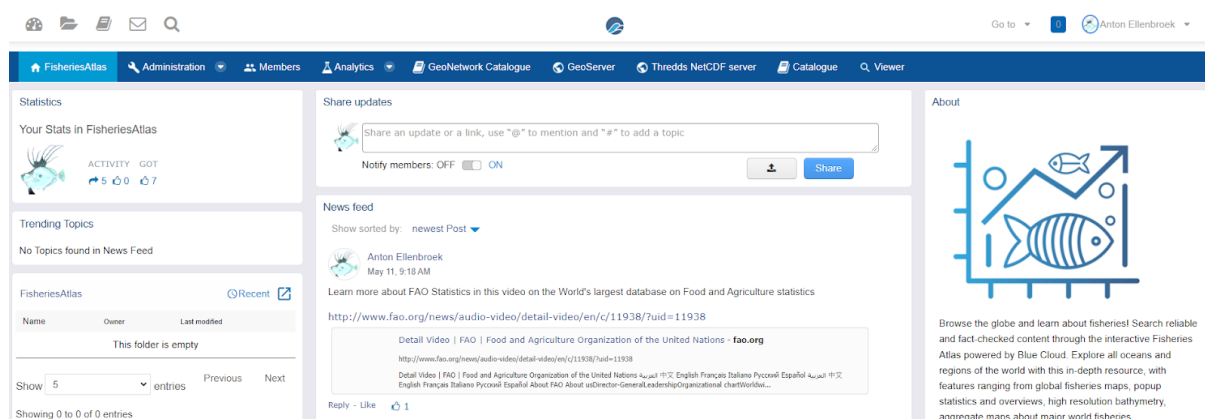


Figure 42 User portal serving fisheries data management communities

Advanced users: regional fisheries data analysts that need to show how fisheries in their area of interest develop over time and researchers modelling marine ecosystems or fisheries dynamics who need such data to be calibrated.

Developers: system developers in need of a generic solution for the management of any fisheries time-series that brings statistical data into a data harmonization and QA process.

We recommend familiarizing yourself with [geoflow](#) R package to assess the requirements on how to publish and manage fisheries data flows within Blue Cloud. Another starting point for data owners is to learn to use d4storagehub4R, an interface to the 'D4Science' 'StorageHub' API.

6.1.3 Step by step guideline to use the service

6.1.3.A Atlas

The Fisheries Atlas, with a focus on Statistical Data Management, is available here: <https://blue-cloud.d4science.org/web/fisheriesatlas> (User registration required at the VLabs level as well)

The Fisheries Atlas, through the public user interface, provides access to thematic maps of fisheries, but also contains data and statistics on fisheries production and trade through a nice and aesthetic interface (RShiny pop-up). The design aims to show trends in Fisheries indicators towards achieving the SDG's. It contains a map viewer to discover, display and access relevant layers and spatial information such as:

- FAO and IRD Tuna Atlas layers, providing global maps of Tuna species catches;
- FAO fisheries commodities and trade national statistics;
- Selected Regional Fisheries Organisations competence areas;
- Selected EMODNet data such as bathymetry, plastic occurrence or CMEMS temperature;
- Selected FAO species distribution maps;

This list will grow with the underlying Blue Cloud Spatial Data Infrastructure (SDI); the FAO VRE manager will decide which dataflow are mature enough and ready for demonstrator display. It can inherit other features from specialised fisheries dataflow later on.

6.1.3.B Developments

The SDI behind the Fisheries Atlas (made of GeoServer, GeoNetwork, OpenFairViewer components) is continuously enriched with FAIR compliant data. Depending on their metadata richness, they are interoperable across upgraded R and R Shiny based visualisation and analytical tools.

The Demonstrator goal is to provide a FAIR environment for fisheries related, geospatial explicit data and the analytics to manage FAIR data. Several key Blue Cloud services are used:



The **Analytics** part offers access to generic services such as RStudio, D4Science's DataMiner, JupyterHub and the Software Algorithm Importer (SAI).

The **GeoNetwork Catalogue** provides access to the FAIR geospatial data. In the case of the fisheries atlas, the focus is on fisheries catch statistics, fishing areas, and species distributions. The catalogue metadata are a key data resource for the viewer and allow managing any OGC compliant data.

A **Thredds** catalogue service is available for e.g. access to public NetCDF files.

The **Catalogue menu option** provides access to the gCube Catalogue (powered by CKAN) that will be filled with public data upon approval of the demonstrator by the data-owners.

The **Viewer** part provides access to data managed by the Fisheries Community through a rich plateau of data from FAO and IRD. Dependent on the data selected by the user, the demonstrator generates a metadata driven query panel that is self-explanatory (Figure 43)



Figure 43 Example of user selection panel on a dataset in the Fisheries Atlas.

6.1.3.C Map Viewer

The “map area” offers, depending on the selection and context of the viewer, interactive options including 2D/3D maps, applications and statistics (pop-up statistics, R-Shiny based query panels on statistical time series).

By selecting a dataset and one associated map layer, the user can apply filters based on the dataset dimensions. The list of dimensions is specific to each dataset and fetched from its data structure definition. It generally includes fisheries-domain specific dimensions (flagstate, region, commodities, catch type, species etc.) that can be selected by means of drop-down lists; the temporal dimension and an aggregation method to generate the statistical map.

Once data is mapped (Figure 44), the user can access data in tabular form and export data either as digital data formats or reproducible scripts (R script, Jupyter notebook).

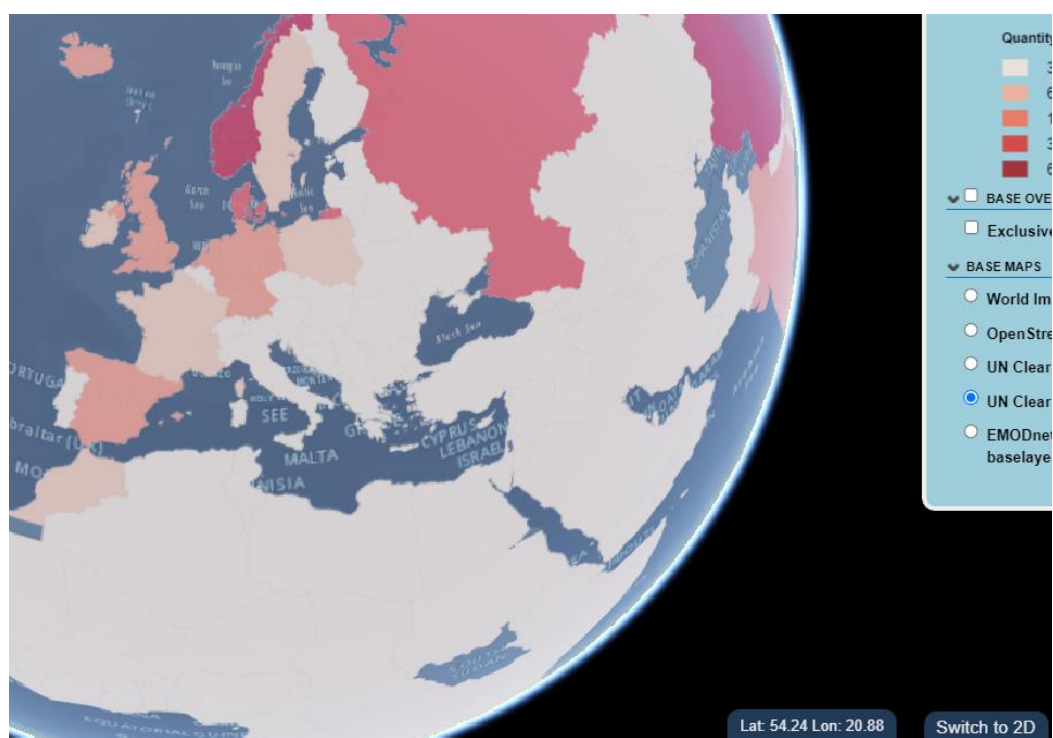


Figure 44 User visualisation option (2D/3D) in the Fisheries Atlas.

For selected cases, fisheries data are represented in a variety of visualisations (Figure 45) made using Rshiny and allowing FAIR statistical time-series, all of which can be downloaded and re-used.

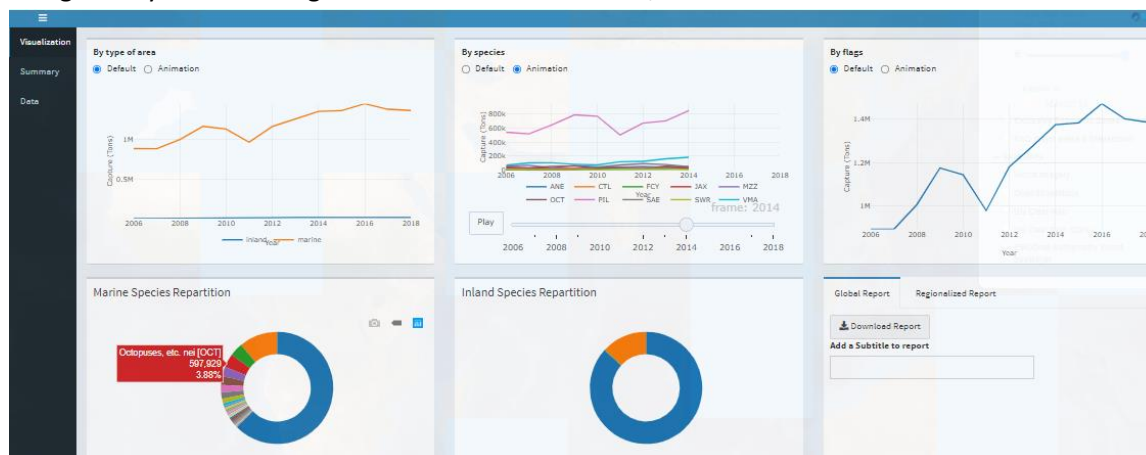


Figure 45 Example of statistics in the Fisheries Atlas.

6.1.4 Data sources

“In Blue Cloud” data: the FAO VRE is capable of finding, accessing and querying metadata and data through ISO/OGC services (CSW/WMS/WFS), analyse and display results. This data can be combined with fisheries specific data for further analysis in the WPS based Blue Cloud DataMiner. Furthermore, FAO VRE Cataloguing services (publish, discover and access) are exploited for exposing the demonstrator information through a VRE operated from Blue Cloud.

External data: access and display ancillary information using a variety of ad-hoc services. Data include Global catch Maps of fisheries and FAO global fisheries project inventory data.

“In Virtue” data: any VRE is designed to ingest, harmonize and standardize fisheries collated data on usual fisheries and related variables (e.g. catch and effort, size class, commodities & trade, AIS). The current global datasets have been provided by regional fisheries management organisations (RFMOs) and the FAO Fisheries Division. However, any compliant time-series could be included and loaded in a similar workflow. Global core information on Global Stocks and Fisheries is incorporated relying on semantic web technologies.

“FAIR compliant” data: this FAO VRE can use in a transparent manner any FAIR dataset as a native source of data, which complies with usual ISO-OGC standards (meta-data formats and access protocols). Also for non-fisheries related data, users can contact the FAO VRE Administrators if they wish to add relevant data. Data that are exposed as fully FAIR compliant ISO/OGC, then any additional data, can be added very easily showing the usefulness of the FAIR approach.

Data used for the service are listed in the table below; they are an example of possibilities offered to the user.

Variables	Data sources	Infrastructure	Access through
<i>Catch and effort, size class, commodities, trade of fisheries, AIS</i>	FAO and other fisheries monitoring and research organisations	Fisheries Atlas VRE Global Record of Stocks and Fisheries VRE	Fisheries Atlas Viewer and catalogues Global Record of Stocks and Fisheries Viewer and catalogue. Selected information through API and SPARQL endpoints

Table 9 Fisheries Atlas data sources

6.1.5 Scientific references

Blondel E., Julien Barde, Wilfried Heintz, & Alexandre Bennici. (2020). GeoFlow: Tools to orchestrate and run geospatial (meta)data workflows. Zenodo. <https://doi.org/10.5281/zenodo.4275926>

Blondel, Emmanuel. (2021). OpenFairViewer: a FAIR, ISO and OGC (meta)data compliant GIS data viewer for browsing, accessing and sharing geo-referenced data (2.7.2). Zenodo. <https://doi.org/10.5281/zenodo.5761650>

6.2 Global record of stocks and fisheries (GRSF)

Authors: Anton Ellenbroek (FAO)

6.2.1 Short description of the service

The objective of the Global Record of Stocks and Fisheries (GRSF) is to deliver a scalable and robust open data portal for fisheries data in EU waters and beyond, with a focus on assessment status and management of natural living resources. The service expands the existing GRSF VRE with new information for approved status assessments of fisheries, including those from other sources and demonstrators. The GRSF contains Unique Identifiers of Stocks and Fisheries and the descriptions of the area and management structure. It is thus a natural companion of the Fisheries Atlas (service described above) that has more catch location specific information.

This process is not open access, but private since the information needs to be first aligned and validated. The prime purpose of the service is to develop a public dataset using Blue Cloud services for harmonization, standardization, maintenance and publication of fisheries data from different sources.

6.2.2 Targeted users

End-users: general public with an interest in stocks and fisheries, fish provenance, fisheries distribution, fisheries and SDG2 and SDG 14, accessed through e.g. web-portals, atlases, API's or QR codes. FAO is negotiating a release plan with the rightful data-owners.

Advanced users: regional fisheries data analysts, who need to show how fisheries in their area of interest develop over time, can request to become a member of the GRSF core team. Only owners or collators of fisheries data with a vested interest in monitoring fisheries can be accepted.

Developers: system developers dealing with the management of fisheries can bring collated statistical data into a data harmonization and QA process through a collaboration with the Fisheries Demonstrator Managers FAO and FORTH. Desired skills include semantic technologies and REST-API's.

Fisheries Managers: regional fisheries managers that require access to overviews of fisheries to inform their management decision making processes (Figure 46).

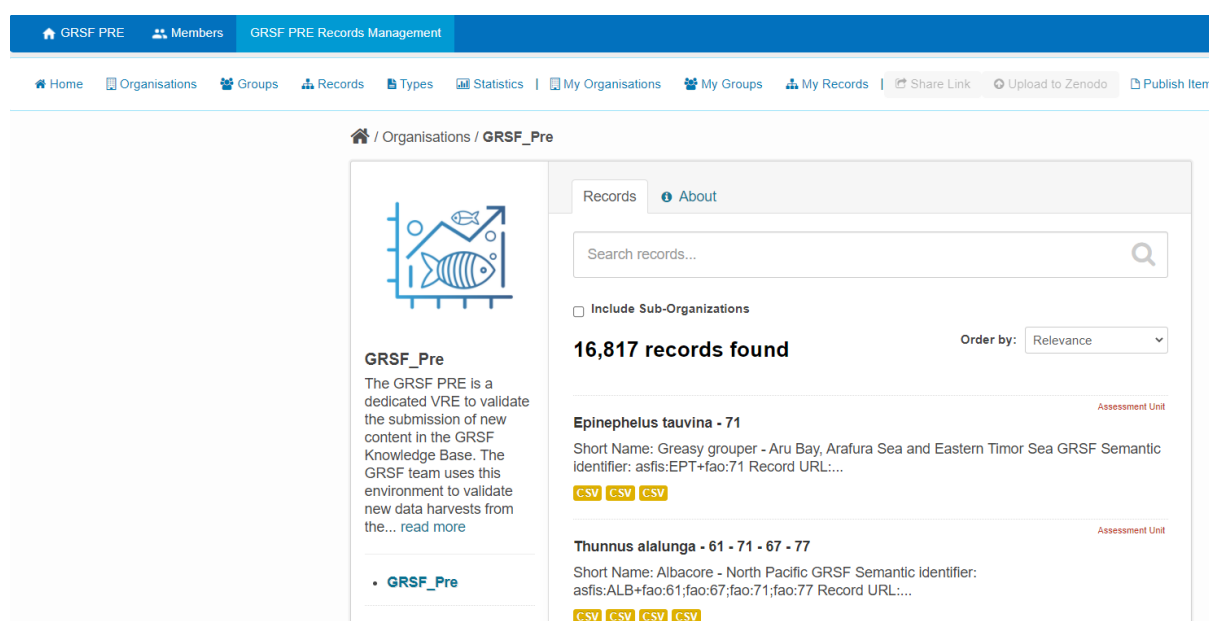


Figure 46 GRSF data preparation portal for registered users

6.2.3 Step by step guideline to use the service

6.2.3.A Global record of Stocks and Fisheries (GRSF)

The service is currently not opened and needs additional access codes at the VLab level.

The global reference repository for stocks and fisheries is accessible through:

- the Blue Cloud catalogue that enables the hierarchical organisation of those resources, with respect to several groupings (e.g. their corresponding types, exploiting resources, provenance

- information, etc.), which allows users discovering and accessing them, including QR codes for the visual identification and ease of sharing across different users and platforms,
- a set of APIs that allow retrieving particular information for stocks and fisheries in a programmatic manner,
 - a set of competency queries able to answer complex (and in many cases common) questions that are impossible to answer from the original data sources of GRSF. Each competency question is assigned a small description and a SPARQL query to be submitted to the GRSF Knowledge Base, which allows an answer and a presentation of the result to the user in a user-friendly manner (Figure 47).

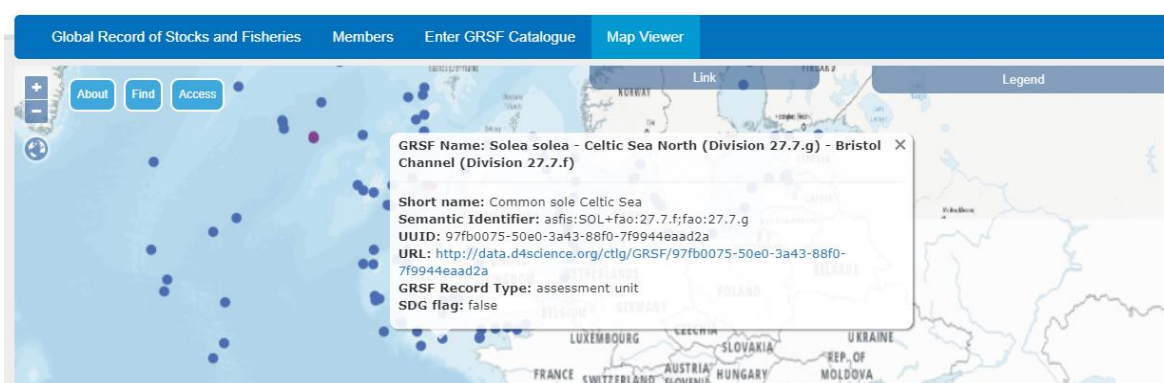


Figure 47 GRSF Public Map interface

6.2.3.B GRSF pre-release

These VREs include the GRSF with a focus on stocks and fisheries Information management and are available on the following link: GRSF PRE: https://blue-cloud.d4science.org/web/grsf_pre.

The service shares many aspects with the Fisheries Atlas, but has a focus more on information management and sharing of data at the level of stocks and fisheries. The GRSF PRE VLAB is used to validate new content in the GRSF Knowledge Base and thus is not a public service.

In 2020, this environment was used to validate data harvested from 3 global GRSF sources; FAO-FIRMS, the RAM Legacy Stock Assessment Database, and SFP's FishSource. The harmonized results are now published into the GRSF Admin and GRSF VREs. All GRSF environments are connected to the Blue Cloud Spatial Data Infrastructure (SDI) and all geospatial-referenced content can be viewed in context of other data accessible within the Blue Cloud (e.g. Bathymetry layers from EMODnet).

The GRSF is not yet fully released, as the approval process needs to ensure all providers agree on the proposed aggregations, terminology and presentation. A preview can be accessed after approval by the GRSF managers. The pre-release VRE for end-users combines several features (Figure 48):



Figure 48 GRSF navigation bar

The Member section is fully managed as a D4Science service.

The GRSF Catalogue is a gCube Catalogue powered by CKAN, and can be accessed through the user Interface (UI) or by using the gCat API's (Figure 49).

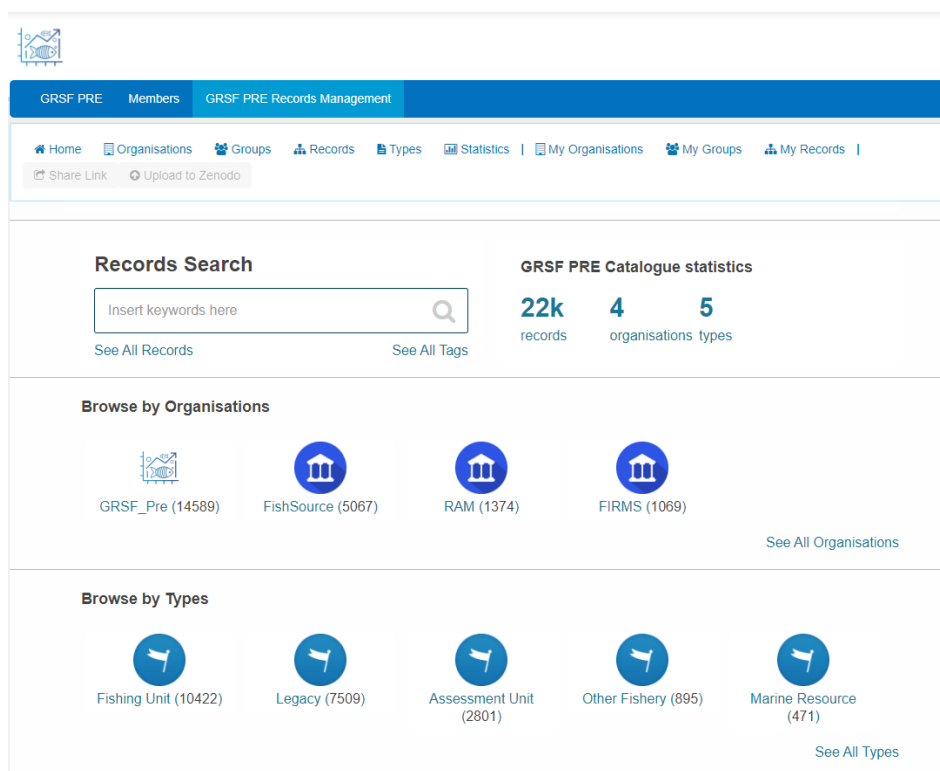


Figure 49 The GRSF VRE UI for the Global Record of Stocks and Fisheries (GRSF).

The Map viewer uses the same metadata driven OpenFairViewer as D4science (Figure 50).

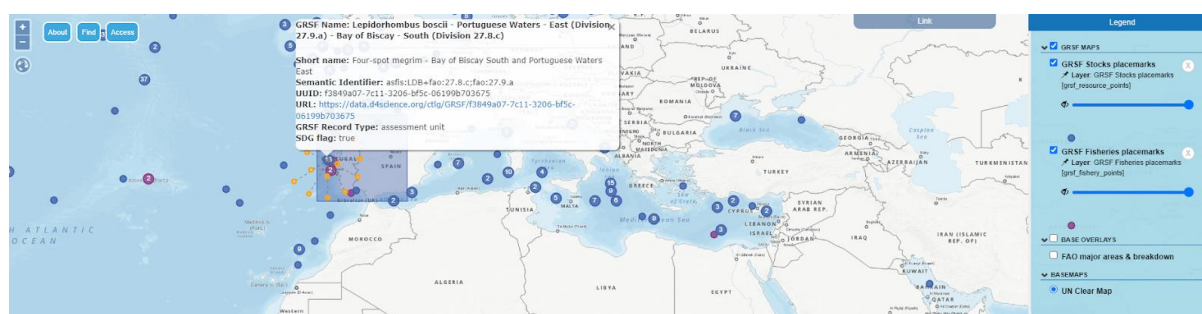
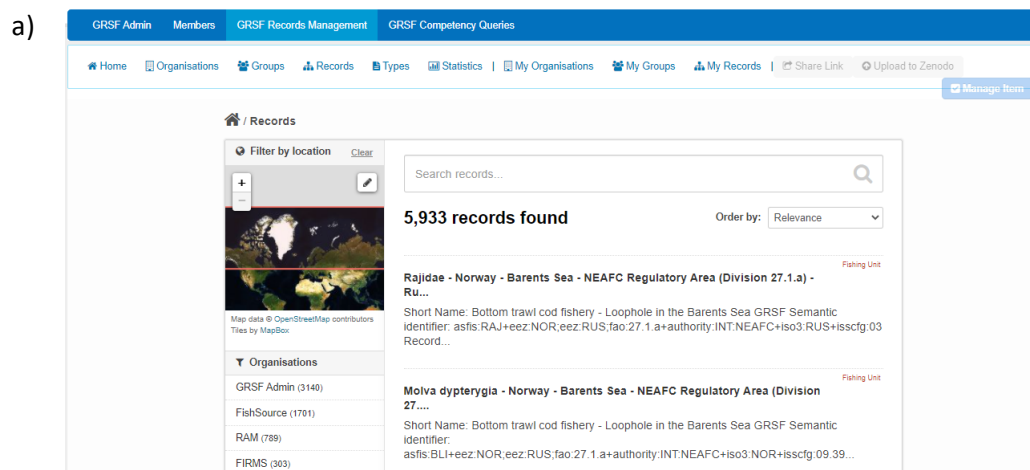


Figure 50 GRSF Interactive Map viewer.

The Competence queries (about 100 in total) directly access the semantic knowledge base behind the GRSF (Figure 51).



b) **Global Record of Stocks and Fisheries**

GRSF Queries

- | | |
|--|------------------------------|
| 1. List all GRSF Stock records | SPARQL query |
| 2. List all GRSF Fishery records | SPARQL query |
| 3. List all GRSF Stock records with type "Assessment Unit" | SPARQL query |
| 4. List all GRSF Fishery records with the following fields Species Name / Area / Management Entity / Flag State / Fishing Gear | SPARQL query |
| 5. List all GRSF Fishery records flagged for traceability purposes | SPARQL query |
| 6. List all GRSF Fishery records from "Atlantic" | SPARQL query |
| 7. List all GRSF Stock records for the genus "Thunnus" | SPARQL query |
| 8. List all GRSF Stock records with their status | SPARQL query |
| 9. List all GRSF Fishery records with their status | SPARQL query |
| 10. Retrieve all (most) available information for stocks (without indicators). | SPARQL query |
| 11. Retrieve all water areas and related records. | SPARQL query |
| 12. Retrieve all species and related records. | SPARQL query |
| 13. Count all waterareas. | SPARQL query |
| 14. Count all species. | SPARQL query |

Figure 51 a) GRSF Record editing environment; b) GRSF API List.

6.2.4 Data sources

The **core data** of GRSF are provided by FAO of the UN, the University of Washington and the Sustainable Fisheries Partnership.

External data: access and display Ancillary information from FNS Cloud on Fish composition and several bespoke datasets are ingested using a variety of ad-hoc services. Data include Global Effort Maps of fisheries, FAO global project data, and fish tagging data.

Data used for the demonstrator is listed in the table below. This gives an example of the possibilities of the service.

Variables	Data sources	Infrastructure	Access through
<i>FIRMS data</i>	FAO - FIRMS	FAO Fisheries and Aquaculture Division	Harmonized and integrated data accessible through Map UI, Registry, QR codes, and API

Variables	Data sources	Infrastructure	Access through
<i>Stock assessment</i>	Univ. Washington	RAM Legacy Stock Assessment Database	Harmonized and integrated data accessible through Map UI, Registry, QR codes, and API
<i>Fish stocks and fisheries</i>	Sustainable Fisheries Partnership (SFP)	FishSource	Harmonized and integrated data accessible through Map UI, Registry, QR codes, and API

Table 12 GRSF data sources

6.2.5 Scientific references

Marketakis, Y., Tzitzikas, Y., Gentile, A., Van Niekerk, B., and Taconet, M., 2020. On the Evolution of Semantic Warehouses: The Case of Global Record of Stocks and Fisheries. 14th International Conference on Metadata and Semantics Research, Special Track on Metadata & Semantics for Agriculture, Food & Environment (AgroSEM'20) Madrid, 2020

Tzitzikas, Y., Marketakis, Y., Minadakis, N., Mountantonakis, M., Candela, L., Mangiacrapa, F., Pagano, P., Perciante, C., Castelli, D., Taconet, M., Gentile, A. and Gorelli G., 2017, September. Methods and Tools for Supporting the Integration of Stocks and Fisheries. In International Conference on Information and Communication Technologies in Agriculture, Food & Environment (pp. 20-34). Springer, Cham.

Tzitzikas, Y., and Marketakis, Y., 2020. Reinforcing Fisheries Management through Semantic Data Integration. ERCIM News, 123.

7 Common useful generic IT services

7.1 Blue Cloud Data Discovery & Access Service





Authors: Dick M.A. Schaap (MARIS)

7.1.1 Short description of the service

The **Blue Cloud Data Discovery and Access service (DD&AS)** is one of the two main components of the Blue Cloud technical framework, next to the **Blue Cloud Virtual Research Environment (VRE)**. It facilitates discovery and retrieval of data sets by users, who need be registered to it to download data. Users can download data sets in stand-alone mode or decide to forward requested data sets to their VLab data pool in the **VRE** if they are registered (N.B. there is one registration for the DD&AS and another one for the VRE). The data sets concern measurement data and derived data products that are managed in Blue Data Infrastructures (BDIs) which are interacting machine-to-machine with the **DD&AS**.

A common **DD&AS** interface is provided for discovery and retrieval of data sets from each of the federated Blue Data Infrastructures. The interface also includes facilities for mapping and viewing the locations of data sets. Moreover, the interface has a “shopping” (nothing to pay) mechanism, facilitating registered users to compose and submit shopping baskets with requests for data sets from multiple Blue Data Infrastructures in one go.

Currently, the Blue-Cloud DD&AS gives access to > 10 million data sets as managed and provided by the following Blue Data Infrastructures:

SeaDataNet CDI service	Marine physics, bathymetry, chemistry, geology, geophysics, and biology observation data sets	
EMODnet Chemistry data products	Marine chemistry data collections and interpolated map products	
EurOBIS - EMODnet Biology	Marine biogeographic data collections with taxonomy and distribution	
Euro-Argo and Argo GDAC	Ocean physics and marine biogeochemistry observation data from Argo floats	







ELIXIR- European Nucleotide Archive (ENA)	Nucleotide sequencing data and information on marine species	
EcoTaxa	Taxonomic annotation data of images on planktonic biodiversity	
SeaDataNet data products	Aggregated marine data collections and climatologies, such as for Temperature & Salinity	
ICOS-Marine	Long-term oceanic observations of carbon uptake and fluxes for understanding the global carbon cycle	
SOCAT - Surface Ocean CO2 Atlas	SOCAT version 2020 with quality-controlled surface ocean fCO2 measurements from 1957 to 2020	
EMODnet Bathymetry	EMODnet Bathymetry World Base Layer is used as base map in the interface	

Table 13 Blue-Cloud Data infrastructures

The query mechanism has a two-step approach:

- The first step enables users to identify interesting data collections, with free search, spatial and temporal criteria on a common catalogue for all federated BDIs;
- The second step enables users to drill down per interesting BDI to get more specific data sets at detail level, using again free search, spatial and temporal criteria and additional search criteria, specific for a selected BDI;

Finally, users are able to compose and submit shopping requests for associated data sets, which can be downloaded from their MyBlueCloud dashboard after some processing time.

This dashboard also allows users to push the retrieved data sets to the VRE data pool where they can use these data sets as input for V Labs and their services.

For more info:

- <https://data.blue-cloud.org>
- <https://data.d4science.net/SiQT>

7.1.2 References

Schaap, D., Thijssse, P., Pagano, P. Assante, M., Candela, L., Boldrini, E., Buurman, M., d'Antonio, M., Ariyo, C., Maudire, G., Nys, C., D2.7 Blue Cloud Architecture (Release 2), 30 Sept 2021

7.2 Blue Cloud Notebook for CMEMS WEkEO data access

Authors: Pasquale Pagano, Massimiliano Assante (CNR)

WEkEO is one of the 5 Copernicus DIAS (Data and Information Access Services). It hosts and gives access to large data products, which are resulting from satellite observations, numerical forecast models and large in-situ collections. These products themselves are too big (GBs to TBs) to be downloaded as units and are increasing in volume in time. For that purpose, the WEkEO Harmonized Data Access (HDA) has been set-up as a unique Application Programming Interface (API) that implements a REST-based single protocol enabling users to issue requests for the data needed. This API functions as a subsetting service, which allows users to compose and run dedicated queries for extracting slices of data from the large products, which then can be downloaded.

This WEkEO API is also very fit for configuring specific data extraction tasks, for instance in a Jupyter notebook and this way being used in the Blue Cloud VRE, also as some of the Blue Cloud V Labs require regular input from WEkEO.

7.2.1 Short description of the service

A tailored notebook has been implemented in the Blue Cloud VRE to bridge it and the WEkEO DIAS service. This notebook exploits the HDA API and allows users to search for datasets, to select the one that is needed, to see the properties of it and to issue a subsetting request that is required to access the data and bring it to the Blue Cloud workspace for use in V Labs.

The HDA API allows performing six steps from the authentication to the download (figure 52).

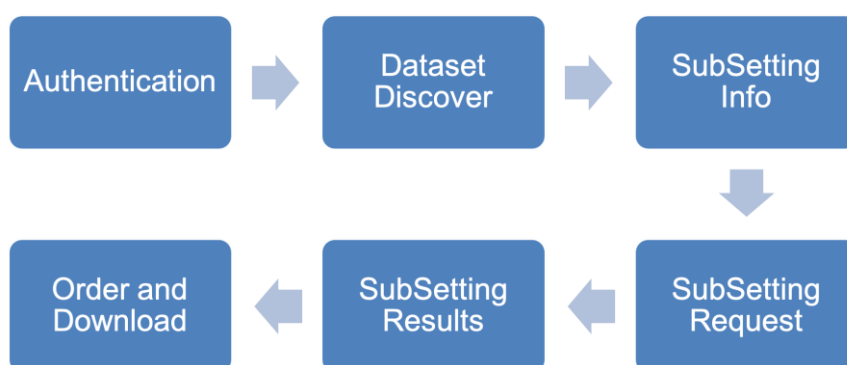


Figure 52 : HDA API steps

For each step, the Blue-Cloud notebook guides the user through the required actions.

For more info:

- <https://www.wekeo.eu/docs/harmonised-data-access-api>
- Blue Cloud VRE address [D4Science infrastructure](#)

7.2.2 References

Assante, M., Candela, L., Pagano, P., D4.4 Blue Cloud VRE Common Facilities (Release 2), 28 December 2021

8 Conclusions

The 12 target thematic services and the 2 additional generic services described give a good illustration of what is feasible and possible in terms of multi-disciplinary demonstrators and IT capacities (VRE, VLab, data discovery and access) through a large variety of examples (topics, scales, thematic). This will be improved over time until the end of the project.

This handbook will be enhanced during 2022 with 2 additional services on aquaculture (demonstrator #5 not described here) and other generic IT services, core of Blue Cloud. The described services might also be updated, for example, using additional data, which will allow improving their reproducibility and robustness.

User feedback is also very welcome to improve the Blue Cloud offer, services and this document which will be updated accordingly.

Demonstrators and services provided illustrate the offer and possibilities of Blue Cloud. It is definitively a step forward to new digital and numerical possibilities, which will be able to consider a very large amount and volume of data and its management.

These new capacities are required by the Green Deal of the European Union and by the Ocean Decade of the United Nations to support the sustainable development of the marine environment. Blue Cloud allows us to be on track although there is still lots of work to do, and for several years, especially on IT developments and data management to meet these EU and UN challenges. Existing and new demonstrators and services will continue to show and illustrate the various and numerous possibilities.