



Offshore Low-trophic Aquaculture in Multi-Use Scenario Realisation

Grant agreement number: 101094065

Topic: HORIZON-MISS-2021-OCEAN-04-01

Starting date: 1 January 2023

Duration: 48 months

D5.2

Title: Service tools delivery – module v0 specifications

Date of delivery:

30 June 2024

Organization name of lead participant for this deliverable: DTU



"Funded by the European Union, grant agreement NO 101094065. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them."

Dissemination level		
PU	Public, fully open	x
Sen	Sensitive, limited under the conditions of the grant agreement (only for members of the consortium, incl the Commission Services)	

Deliverable number:	D5.2
Deliverable title:	Service tools delivery
Work package:	WP5
Lead participant:	DTU

Authors		
Name	Organisation	E-mail
Asbjørn Christensen *	DTU	asc@agua.dtu.dk
Giulia Dapuzo	ETT	giulia.dapuzo@grupposcai.it
Beatrice Maddalena Scotto	ETT	beatrice.scotto@grupposcai.it
Jun She	DMI	js@dmi.dk
Jens Murawski	DMI	jmu@dmi.dk
Vilnis Frishfelds	DMI	vifr@dmi.dk
Antonio Agüera Garcia	IMR	antonio.aguera@hi.no

*corresponding authors

Keywords
OWF, LTA, decisional support tools, service modules, event warning, operation planning, alien species, performance control, disease propagation, co-usage

Review History				
Date	Version	Review Author	Organisation	E-mail
11.06.2024		Carolina Maiorana	IMR	Carolina.maiorana@hi.no
7.06.2024		Øivind Bergh	IMR	Oeivind.bergh@hi.no
6.06.2024		Anita Jacobsen	IMR	Anita.jacobsen@hi.no



"Funded by the European Union, grant agreement NO 101094065. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them."

Executive Summary

This document provides a short overview of the version 0 implementation of the five backend service modules for the WP5 web service for aquaculture farms at OWFs described in the OLAMUR DoA. The aim of the service tool is to support individual aquaculture producers in the process of planning, deployment and operation, production, impact assessment, marketing, and capacity building. The modules described in this document are input to task 5.3 which eventually results in D5.3.

List of acronyms

Acronym	Meaning
AquaNIS	Aquatic Non-Indigenous Species
ASR	Alien Species Risk
BAL MFC	Baltic Monitoring Forecasting Center
CMEMS	Copernicus Marine Environment Monitoring Service
DoA	Description of Action
DPR	Disease Propagation Risk
EWS	Event Warning Service
FPC	Farm Performance Control
FTP	File Transfer Protocol
HAOP	Harmful Aquatic Organisms and Pathogens
HELCOM	Helsinki Commission
IC	Ice Chart
LTA	Low Trophic Aquaculture
MME	Multi-Model Ensemble
NWS MFC	North West Shelf Monitoring Forecasting Centre
OLAMUR	Offshore Low-trophic Aquaculture in Multi-Use Scenario Realisation
OOP	Offshore Operation Planner
OSPAR	Oslo and Paris Convention for the Protection of the Marine Environment of the North-East Atlantic
OWS	Offshore Wind Farms
SIC	Sea Ice Concentration
SIT	Sea Ice Thickness
WP	Work Package



Table of Content

Table of Content	4
1. Introduction.....	7
2. Module EWS - Event Warning Service (DMI)	8
3. Short introduction and background	8
4. Module functionality and structure diagram (visual)	8
5. Methodology and input data	10
6. Software/data requirements	11
7. Module implementation	12
8. Input data/parameters.....	12
9. Module output and example.....	13
10. Module output validation	13
11. Potential for future development/generalization	13
12. Module OOP - Offshore Operation Planner (DMI).....	13
13. Short introduction and background	13
14. Module functionality and structure diagram (visual)	14
15. Software/data requirements	16
16. Module implementation	17
17. Input data/parameters.....	17
18. Module output and example.....	18
19. Module output validation	18
20. Potential for future development/generalization	18
21. Module ASR - Alien Species Risk (DTU)	19
22. Short introduction and background	19
23. Module structure diagram	19
24. Module functionality	20
25. Software/data requirements	20



26.	Module implementation	20
27.	Input data/parameters	20
28.	Module usage	20
29.	Module output validation	21
30.	Potential for future development/generalization	21
31.	Module DPR - Disease Propagation Risk (DTU).....	21
32.	Short introduction and background	21
33.	Module structure diagram	24
34.	Module functionality	24
35.	Software/data requirements	28
36.	Module implementation	29
37.	Input data/parameters	30
38.	Module output	30
39.	Module output example.....	31
40.	Module output validation	32
41.	Potential for future development/generalization	32
42.	Module FPC - Farm Performance Control (IMR)	33
43.	Short introduction and background	33
44.	Module structure diagram (visual).....	33
45.	Module functionality	34
46.	Software/data requirements	35
47.	Module implementation	35
48.	Input data/parameters	35
49.	Module output	35
50.	Module output example.....	36
51.	Module output validation	36
52.	Potential for future development/generalization	36
53.	Discussion	36
54.	References.....	37
	Appendix I.....	38
1.	Technical details for operational data download and upload.....	38
1.1.	Atmospheric data from DMI Harmonie model	38
1.2.	Wave model data from DMI WAM model	40



1.3. Data processing of Copernicus Marine Service data, Scatterro winds	40
1.4. Data processing of Copernicus Marine Service data, Satellite SST and sea ice	40
1.4.1. Satellite SST	41
1.4.2. Ice model data	41
1.5. Data processing of Copernicus Marine Service data, CHL-a satellite observations and forecasts	41
1.5.1. Download and processing of satellite observed CHL-a	41
1.5.2. Download and processing of CHL-a modelled forecasts	41
1.6. Putting to FTP	41



1. Introduction

Task 5.2 develops a decision support functionality suite applied in T5.3/T5.4 to the WP5 web service for aquaculture farms at OWFs:

- T5.2: create scientific basis for service modules, passed to T5.3 as “algorithms”
- T5.3: technical integration of modules from T5.2
- T5.4: demonstration, validation, maturation of web service developed in T5.2 and T5.3

The present document provides a short overview of the initial (version 0) implementation of the five backend service modules for the WP5 web service for aquaculture farms at OWFs, as described in the OLAMUR DoA.

Task 5.2 organizes the data in support of WP2-WP4 and develops the decisional support methodology in the form of 5 modules and deploys the implementation to provide the added value products to OLAMUR stakeholders. Each module consumes data and data-products made available on infrastructure (see Annex 1 Deliverable D5.1.1 *OLAMUR data system and service platform v.1.0*) and run on-line/off-line algorithms (see WP2-WP4) to generate ad-hoc decision support information. The 5 modules are as follows:

- EWS - Event Warning Service: develop a warning module tool that identifies high impacts events for early warning based on forecast, satellite and local monitoring data such as storms, high seas, severe ice conditions, algae bloom.
- OOP - Offshore Operation Planner: the module offers an optimized local forecast based on multiple large-scale forecasts and local observations. The tool will be based on a multi-model ensemble developed in CMEMS and machine learning. The optimized local forecast will be further used to determine the working window in daily operations by combining user-defined criteria for offshore operations.
- ASR - Alien Species Risk: this tool will support farms to run alien species dispersal risk assessment. The module will adopt and adapt AquaNIS – information system on aquatic non-indigenous and cryptogenic species which serves HELCOM and OSPAR areas and other world regions. A recent update introduced a harmful aquatic organisms and pathogens (HAOP) early warning tool. This service will take input from WP3 and WP2.
- DPR - Disease Propagation Risk: this tool implements a spatial risk assessment and will be developed for aqua-farms for selected parasites. The goal of the module is to enable favorable site-layout and evaluation of alternative disease control options in relation to dynamic ocean conditions.
- FPC - Farm Performance Control - Farm condition assessment: this protocol will aggregate the data from regular farm area and its maintenance, to give an evaluation of crop status (e.g. growth, condition, risks of bacteria and environment).



The above modules will be linked to the OLAMUR map viewer that is currently being developed and tested. Specific visualisation and dashboard services will be developed. These connections will be made in coordination with the different task leaders in order to meet the different needs. For more details, please refer to the deliverable to be published in December 2024, *D5.3 OLAMUR Data Service Modules Description*.

2. Module EWS - Event Warning Service (DMI)

3. Short introduction and background

The event warning service module (EWS) is a warning tool that identifies high impacts events for early warning including storms, high seas, severe ice conditions, algae bloom, based on forecast and observations. The EWS provides important information for supporting decision-making in managing co-located aquaculture and offshore wind farms. As a demonstration service, the EWS module focuses on the three pilot sites in OLAMUR project (Table 1).

Please be noted that, as a demonstration, algae bloom warning is only made for German Bight (pilot A).

Pilot	Latitude (N)	Longitude (E)	Distance from shore (n.m.)	#of wind turbines	Aqua. farming	Type of warning events	Water depth (m)
A	54°23'0"	7°41'0"	12	80	Seaweed	Storm, high sea, algae bloom	28
B	55°1'34"	12°56'20"	8	72	Seaweed, mussels	Storm, high sea	16-30
C	58°27'41"	22°4'30"	2	0	Finfish, seaweed, mussels	Storm, high sea, severe sea ice	25

Table 1. Background information of the three pilot co-located farms and types of warning events

The algae bloom warning service (German Bight) includes the short-term prediction and warning of algal blooms including the proportion of blue-green algae in the German Bight area. The service runs on an unstructured triangular grid with particularly high resolution in the area of the wind farm of case study A of the project. The coupled physical-biogeochemical model is driven by operational oceanic data from CMEMS, which is supplemented with atmospheric operational data from the German Weather Service and discharge data from North Sea tributaries.

4. Module functionality and structure diagram (visual)

The functionality of the EWS module is to provide users with warning information in the next 24-48 hours on a set of hazard events in the farming area, including storms, high sea, severe sea ice and algae bloom. This can be realized via two types of visualizations: i) hazard overview presentation, which is a 2D map to show all warning events in the pilot areas (Figure 1); ii) forecast time series at pilot location (Figure 2).



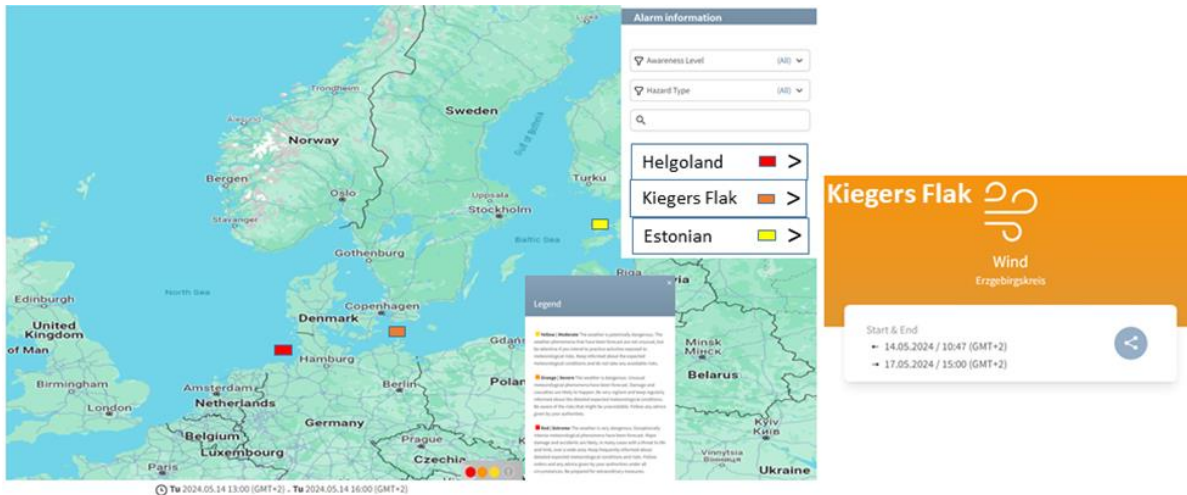


Figure 1. Schematic presentation of the overview map of hazard event information. The level of hazard will be displayed in yellow, orange or red, marked with a symbolic icon representing the hazard type. User has an option to select pilot site, awareness level (i.e., the hazard level) and hazard type. When a pilot site is chosen, all eligible warnings will be listed. For a specific warning, user will be able to see the warning text, as shown for the orange storm warning in Kiegers Flak.

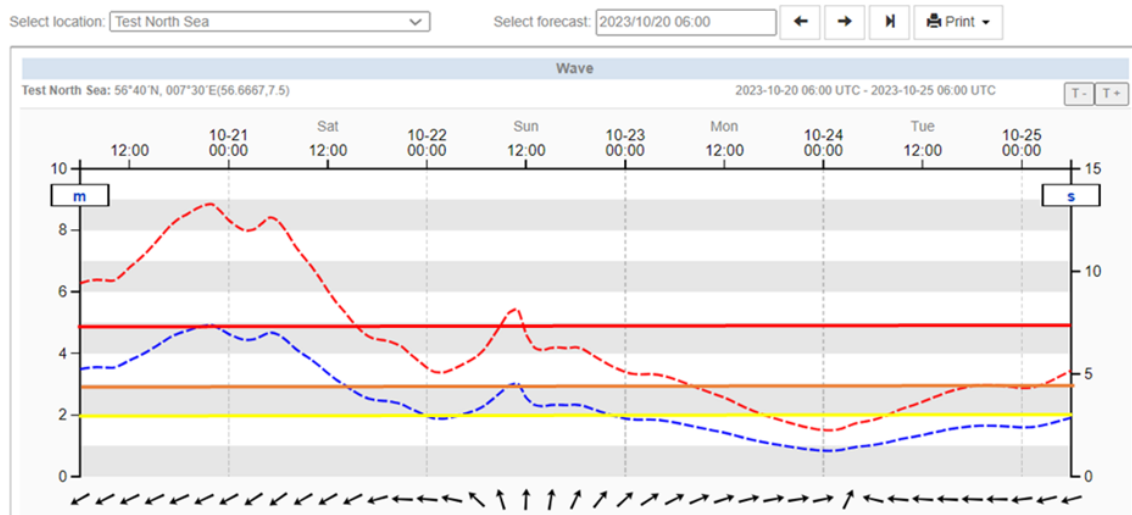


Figure 2. An illustration of a forecast time series at a pilot site: the blue line is the significant wave height forecast from 2023/10/20 06:00GMT to 2023/10/25 06:00GMT. The warning criteria are displayed with yellow, orange and red lines. Users can find that during 2023/10/20 06:00GMT-2023/10/23 00:00GMT, the significant wave height is above 2 m thus is a yellow warning period, while 2023/10/20 06:00GMT - 2023/10/21 12:00GMT is an orange warning period. Compared with the overview warning information, this plot gives more detailed warning information.

The algae bloom warning service (German Bight) module is based on an operational numerical model of ocean physics and biogeochemistry of the low trophic levels for the southern North Sea with high resolution of the German Bight, coasts and around the studied wind farm. A post-processing application reads the forecast data of up to three days and calculates the predicted average and peak algal load. These data are categorised in a scheme for evaluating the eutrophic situation based on chlorophyll or algal biomass concentrations. A warning is generated if the limit value is exceeded.



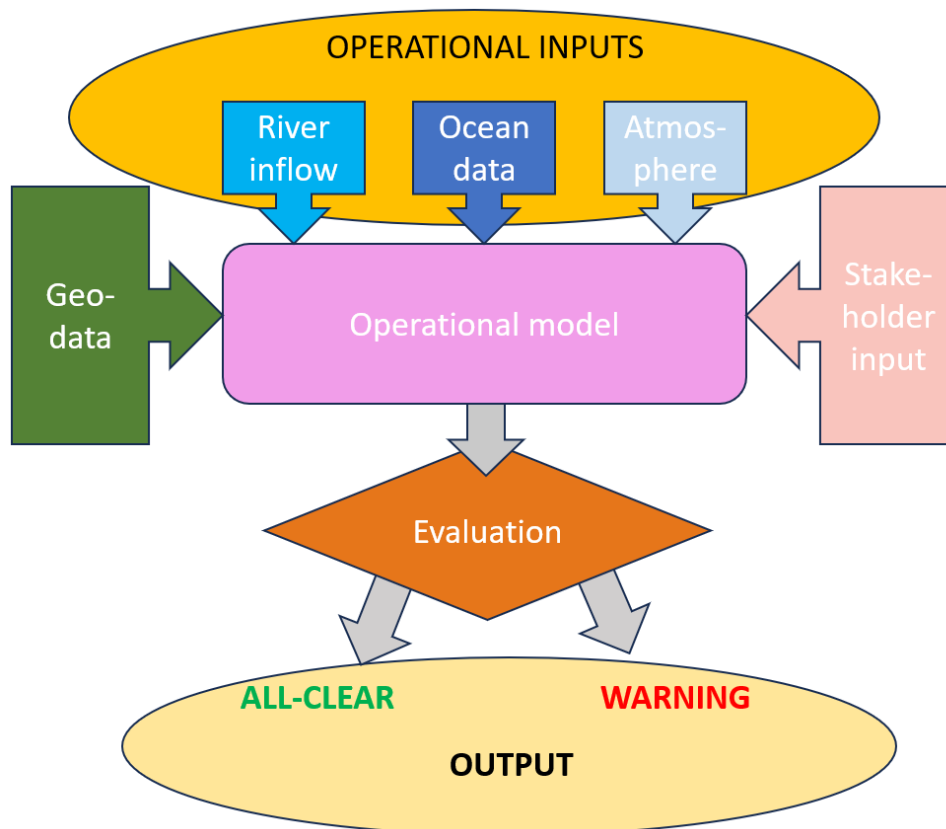


Figure 3. Data and information flow diagram of operational algae bloom warning service.

5. Methodology and input data

For each type of high impact event, 3 categories of warning will be issued. The warning criterion is based on a mixture of classic weather warning as defined in the weather service and for other thresholds also consulted with the farm operators. Input data are operational forecasts and observations, from CMEMS, national weather and other ocean services, if available. Details of the warning criteria and input data are given in Table 2.

Event type	Warning criterion	Warning categories	Timeframe of warning	Input data requirements
Sea storm	24.5 m/s > Sea surface wind speed (SSW) * >20.8 m/s	Yellow	24-48h	DMI operational weather forecasts
	32.6 m/s > SSW >24.5 m/s	Orange	24-48h	
	SSW >32.6 m/s	Red	24-48h	
High seas	3 m > Significant wave height (SWH) > 2 m	Yellow	24-48h	DMI operational wave forecast
	5 m > SWH > 3m	Orange	24-48h	
	SWH > 5 m	Red	24-48h	
Severe sea ice	Sea ice thickness (SIT) > SIT _c **	Level 1	24-48h	CMEMS sea ice observation and forecast (satellite and model)



Algae bloom	Total chlorophyll concentration > 15 mg/m ³ Chlorophyll cyanobacteria concentration > 10 mg/m ³	Red	24-48h	HEREON operational physical-biogeochemical model
-------------	--	-----	--------	--

Table 2. Event types, warning criterions and correspondent categories
 * the winds are defined as at 10m elevation, and 10min. Average
 ** SIT_c will be defined by farm operator UT

6. Software/data requirements

The flowchart of the EWS module is shown in Figure 4. First, the forecast and observation data should be automatically downloaded from data providers, i.e., DMI, HEREON and CMEMS. Second, warning information should be generated in text format for the three pilot sites, using the downloaded data. Third, all downloaded data products and the warning message should be uploaded to a specific folder in a server hosted by the platform host (ETT). Finally, ETT will present the 2D warning information (the mockup is illustrated in Figure 1) and forecast time series (the mockup is illustrated in Figure 2).

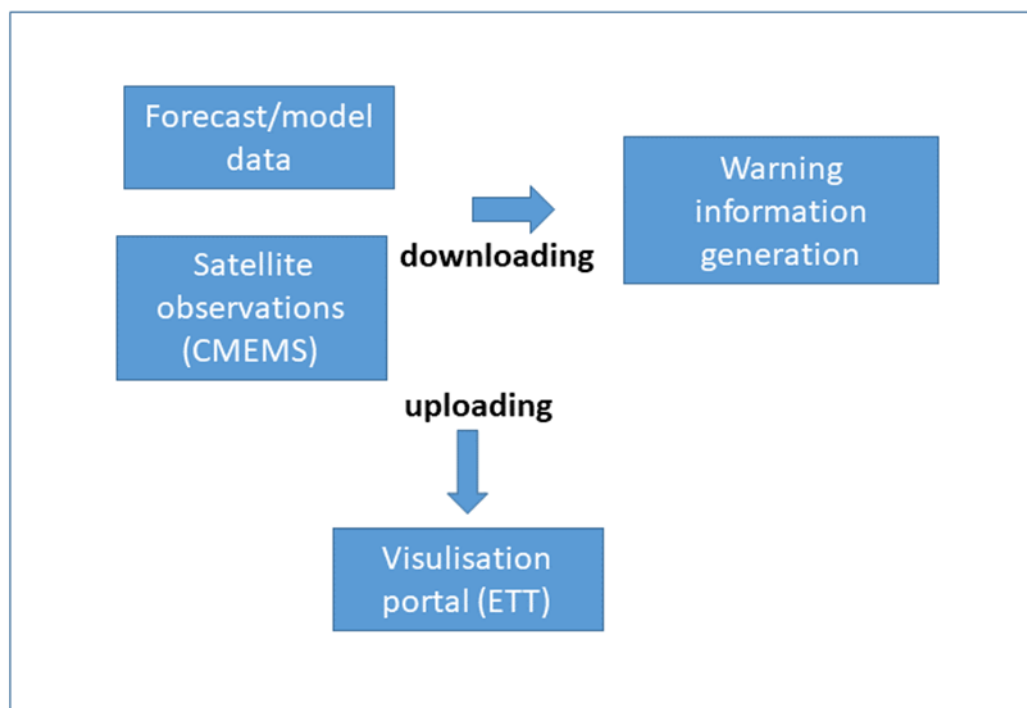


Figure 4. Flowchart of the Event Warning Service module.

All data used in the EWS are obtained either from project partners or CMEMS.

Software for auto-downloading, warning information generation, auto-uploading and visualization will be required to complete the EWS workflow.

To accomplish the EWS module, a variety of software tools are required, e.g. HPC, Fortran, intel fortran compilers, intel mpi compilers, python/Matlab/Jupyter.



7. Module implementation

The EWS service module is developed jointly by DMI, HEREON, KLU and ETT.

DMI is the module leader, responsible for overall designing, data provision/autodownloading/warning information generation/auto-uploading of the storm, high sea and sea ice hazard, HEREON/KLU is responsible for data provision/autodownloading/warning information generation/auto-uploading of the algae bloom hazard, ETT is responsible for visualization and establishing and maintaining the map viewer. The numerical operational model underlying the algae bloom EWS module uses an unstructured mesh of the southern North Sea with high resolution in the German Bight (Pein et al., 2023). The modelling framework is compiled and operated on local HPC resources. The access to operational forcing data happens via FTP or similar connectivity.

Operational data downloading and uploading: DMI has implemented automatic downloading/uploading of the required data. The operational schedule is currently defined as:

- Downloading CMEMS daily products at 9 UTC
- Downloading WAM products at 3, 9, 15, 21 UTC
- Downloading Weather prediction products at 3, 9, 15, 21 UTC
- Uploading to Olamur FTP at 3:54, 9:54, 15:54, 21:54 UTC

For more details, please see Appedix I - Technical details for operational data download and upload.

8. Input data/parameters

Sea surface wind forecast at 10 m elevation: DMI provides weather forecast based on HARMONIE model (<https://www.dmi.dk/vejro-og-atmosfaere/temaforside-vejro-modeller-step-by-step-for-de-nysgerrige-og-nordede/harmonie-dmis-vejro-model>), four times a day, with a forecast range of 56 hours and 2.5 km resolution. The product is hourly and covers the Baltic-North Sea region.

Significant wave height forecast: the wave forecast used in EWS is produced by DMI wave model WAM Version 4.5, four times a day, with a forecast range of 132 hours. The product is hourly and covers the Baltic-North Sea region with a resolution of about 5 km. A high resolution (in 1 km) forecast is available for inner Danish waters, covering the pilot 2 area. The DMI Baltic-North Sea region is nested in a North Atlantic region so that swell through the North Sea boundary from the Atlantic is resolved.

Sea ice analysis and forecast: the sea ice products are derived from satellite sea ice concentration (SIC) and sea ice thickness (SIT) from CMEMS level 4 products (product name SEAICE_BAL_SEAICE_L4_NRT_OBSERVATIONS_011_004, <https://doi.org/10.48670/moi-00132>). The SIC product is based on ice chart on daily basis during the Baltic Sea ice season in a 1 km grid. The SIT is a product based on the most recent available ice chart (IC) and a SAR image. The SAR data is used to update the ice information in the IC. For the forecast, CMEMS BALMFC product (BALMFC product (BALMFC_ANALYSISFORECAST_PHY_003_006,



<https://doi.org/10.48670/moi-00010>) is used. The forecast is made twice a day for a 120h range. Hourly data are available in a 1 n.m. grid.

Algae bloom warning service (German Bight): Physical boundary conditions such as daily sea surface height, current velocity, salinity, water temperature are derived from CMEMS product (e.g. GLOBAL_ANALYSISFORECAST_PHY_001_024). River nutrient load, algae biomass are obtained from EMODNET and national services. Atmospheric forcing with wind speed, surface pressure, surface temperature, radiation are retrieved from the German Weather Service COSMO-EU operational model (see Kaspar et al., 2020).

9. Module output and example

The module outputs are near real time warning of hazard events of storm, high waves, severe sea ice and algae bloom in the next 24-48 hours at the three pilot sites. At the same time, forecast time series of 10 m winds (in m/s), significant wave height (in meter), sea ice concentration (in %) and ice thickness (in meter), total algae biomass (in mg C / m³) and cyanoalgae biomass (in mg C / m³).

The figures 1 and 2 show a mockup of how the service visualisation within the map viewer could look like. The specific visualisation will be presented in deliverable D5.3.

10. Module output validation

The module output validation includes a product quality validation and a function validation. Product quality validation: this validation will evaluate quality of the forecast of the module products, i.e., forecast time series of 10 m winds, significant wave height, sea ice concentration and thickness, total algae biomass and cyanoalgae biomass against observations. Considering that the operational demonstration will start in year three, the product quality validation will be performed in year three.

The module function validation can only be made after the operational demonstration. This will include two parts: user feedback and statistics of system operational performance. The former will tell potential improvements of the service and the latter will tell the timeliness and stability of the system.

11. Potential for future development/generalization

Potential future improvements can be derived from the module output validation. This includes potential improvements on product quality (e.g. use of AI to merge the model simulation and underlying trends with satellite observations to generate the forecast on weighted combination from both sources), optimized service functions etc.

12. Module OOP - Offshore Operation Planner (DMI)

13. Short introduction and background

For both offshore wind farms and aquaculture farms, short-term operation and maintenance is “weather limited”. Strict limits exist for sea state parameters to allow maintenance ships to anchor at the offshore turbines and to transfer personnel. Typically, insurance companies define significant wave heights of approximately 1.5 m, below which maintenance operations



can be conducted. The exact limit depends on the ship size and type. As the costs for the operation of maintenance ships are a major factor, optimized planning of these operations is crucial. Decisions about whether to leave the harbour and go out to a farm are made based on sea state forecasts.

The Offshore Operation Planner (OOP) service module allows users to find out potential working window which meets user-defined length of the offshore operation and “weather limits”, based on the metocean predictions and/or local observations. The metocean datasets (winds and waves) used in OOP are from the same forecasts made for EWS. The prototype OOP module is developed for both OWF operations and aquaculture farm operations at the three pilot sites in OLAMUR project (Table 3). However, it can be used in any locations in the sea.

Pilot	Lat. (N)	Lon. (E)	Distance from shore (n.m.)	#of wind turbines	Aqua. farming	Type of operations	Water depth (m)
A	54°23'0"	7°41'0"	12	80	Seaweed	OWF, aquaculture farming	28
B	55°1'34"	12°56'20"	8	72	Seaweed, mussels		16-30
C	58°27'41" "	22°4'30"	2	0	Finfish, seaweed, mussels	Aquaculture farming	25

Table 3. Background information of the three pilot co-located farms and types of OOP

It is well-noted that accurate wind and wave forecast are essential for the quality of the OOP, and model forecasts may have significant errors offshore where the sea state is sensitive to the water depth and fetch. This tool will also provide optimised local forecast using two methods: multi-model ensemble and machine learning. The optimised local metocean forecast are then used to provide local predictions of weather windows for the offshore operation planner. The focus is on the prediction of continuous periods with favourable working conditions, i.e. low wind velocities and wave heights. For a given location, the tool will provide meteogram weather-and wave charts based on the optimized local forecasts along with a color-coded display of weather windows for user-defined critical thresholds. These thresholds represent safety ranges of marine operations and working conditions.

14. Module functionality and structure diagram (visual)

The functionality of the OOP module is to provide users with weather window information in the next 120 hours for user-defined pilot site, operation length and weather limits. The user-interface of the module consists of two parts: (1.) a data viewer, which displays either the bathymetry or surface maps of metocean forecasts (Figure 5) and (2.) the offshore operation planner, which displays forecast time series and weather windows for user specified locations, operation length and weather limits (Figure 6).



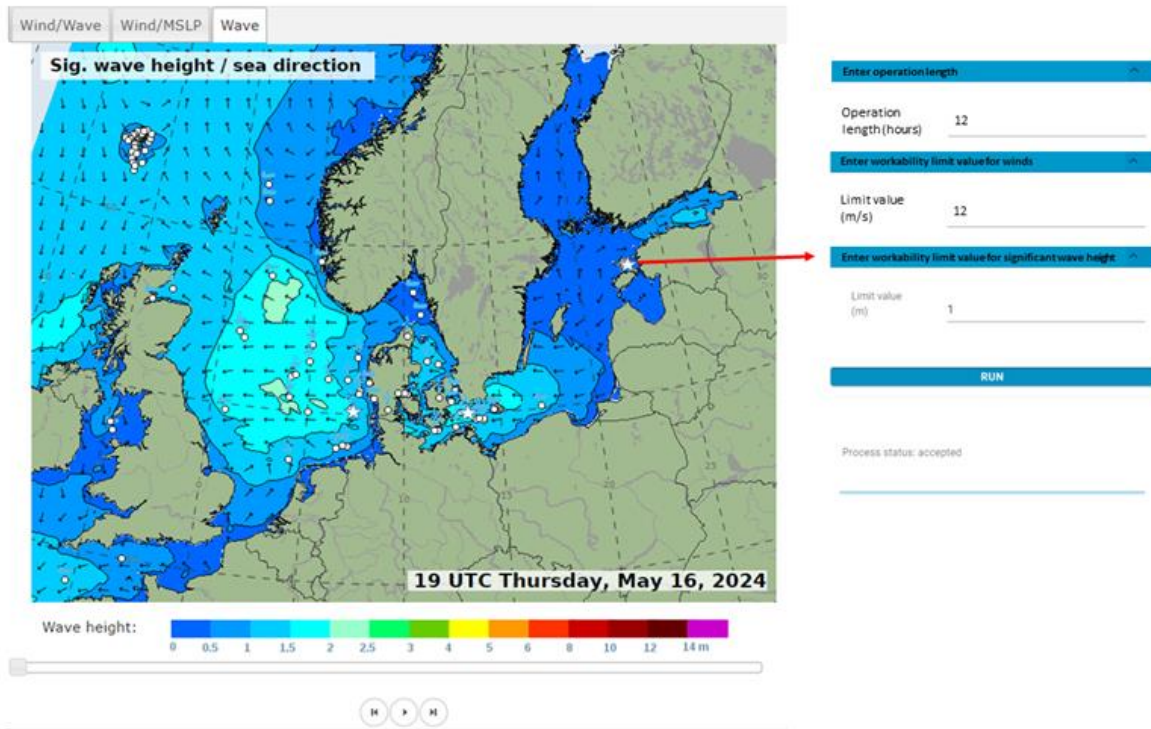


Figure 5. OOP Data viewer: the stars represent the pilot sites. By clicking a star, an interactive window will be activated in which the user will be asked to input the operation length and weather limit for the operation. After the input data are given, user can click the “RUN” button to get a forecast time series which marked qualified working windows.

In the data viewer, the user can view surface maps and animation of wind and wave forecast in next 120 hours. The user can also further specify the location for the offshore operations planner OOP in the data viewer. By doing this, the OOP allows the user to specify operation length, critical operation limits for wind speed and wave height (Figure 6). The module will calculate and visualize the time periods when the given criteria is fulfilled, i.e. the critical safety thresholds are not exceeded.

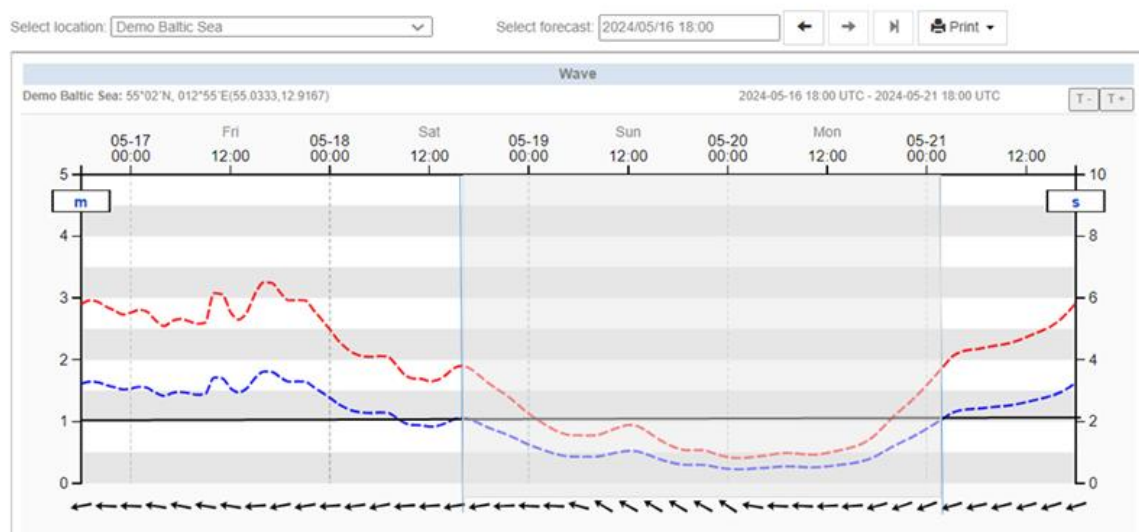


Figure 6. Weather window planner: this plot illustrates when select Pilot B, and with an operation length of 12 hours and a wave limit of 1 m, the qualified weather window is marked as grey.

Another function for the weather window planner is to add other value-added forecasts, e.g. the optimized multi-model ensemble forecast and observation data (if available), and local forecast based on machine learning.

The value-added local forecast is done during the data download, harmonization and pre-processing phase of the module. The user will not be able to interact with this part of the module. Developers will ensure that the user is provided with the best available forecast of wind and wave parameter. In the second part of the module, users can interact with the module and specify operation length, locations and critical thresholds for weather window calculation. These will be defined on the website, to which the user has access via a web interface.

The definition of working windows uses critical threshold values for wind velocity and significant wave height, to identify continuous periods with suitable metocean conditions for maritime and working operations. The critical threshold values depend on the size and category of the vessel that is involved and the type of work that is carried out. For this reason, users will be enabled to set the critical threshold value for their application. OWF maritime operations require larger vessels which can operate in rougher conditions. Aquaculture operations are usually carried out in smaller vessels that operate in the confines of the aquaculture farm. These vessels and working operations have much lower critical threshold values (Table 4).

Application	Critical threshold for wind velocity	Critical threshold for significant wave height
OWF maritime operations and navigation	15 m/s	1.5 m
Aquaculture farms, working conditions		0.5 m

Tabel 4: Typical threshold values for offshore wind farm (OWF) maritime operations and aquaculture working conditions, as currently used by Vattenfall for OWF operation and Seafarm for aquaculture farm operation in Kriegers Flak (Pilot B).

15. Software/data requirements

The flowchart of the OOP module is shown in Figure 7. First, the multiple forecasts of wind and waves, and satellite/in-situ wave observation data should be automatically downloaded from data providers, i.e., DMI, HEREON and CMEMS. Second, value-added local wave forecast will be made for three pilot sites, using the downloaded data, multi-model ensemble (MME) prediction and/or machine learning. Third, all downloaded data products and the value-added forecasts should be uploaded to a specific folder in an FTP server hosted by the platform host (ETT). Fourth, an interactive weather window calculator will be developed with the forecast data and user-defined operation length and workability limit of wind speed and significant wave height. Finally, ETT will present the 2D data viewer (mockup is illustrated in Figure 5) and weather window planner (mockup is illustrated in Figure 6).



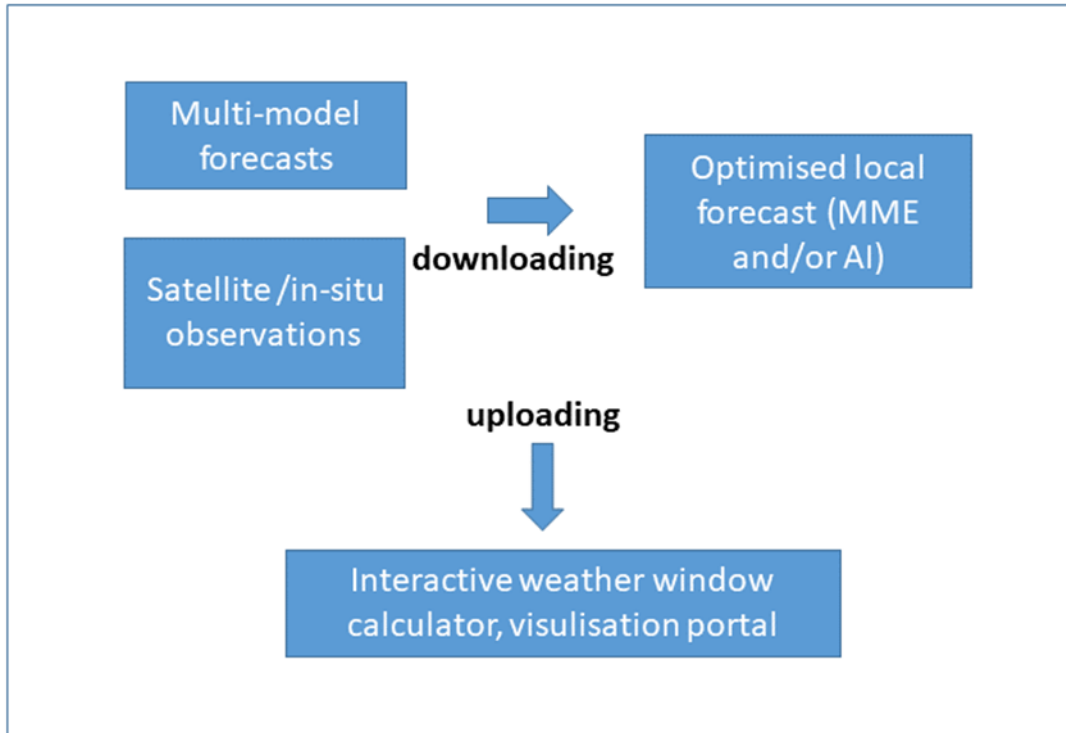


Figure 7. Workflow chart of the Offshore Operation Planner Service module.

Software for automatic downloading, MME local forecast (Frishfelds et al., 2023), ML-based local forecast, auto-uploading and interactive weather window calculator will be required to complete the OOP workflow.

16. Module implementation

The module runs as a web service using data from the OLAMUR data management system that is provided by DMI via FTP. DMI organizes the data download, harmonization and pre-processing of both model forecasts and observations, using data from different providers: CMEMS as well as project partners. The data is then uploaded to DMI's FTP server from which it is grabbed and uploaded to the OLAMUR data management system. The OOP web service utilizes the uploaded data for data visualization and local weather window calculation. The OOP web service runs on a dedicated server.

The OOP prototype will be based on single model forecast. The value-added local forecasts (MME and ML-based) will be added at a later stage.

17. Input data/parameters

For the prototype, DMI wind and wave forecast in the Baltic-North Sea will be used.

Sea surface wind forecast at 10 m elevation: DMI provides weather forecast based on HARMONIE model, four times a day, with a forecast range of 56 hours and 2.5 km resolution. The product is hourly and covers the Baltic-North Sea region. Between 56-120 hours, ECMWF wind forecast will be used, which has a 10 km resolution.

Significant wave height forecast: the wave forecast used in EWS is produced by DMI wave model WAM Version 4.5, four times a day, with a forecast range of 132 hours. The product is



hourly and covers the Baltic-North Sea region with a resolution of about 5 km. A high resolution (in 1 km) forecast is available for inner Danish waters, covering the pilot B area. The DMI Baltic-North Sea region is nested in a North Atlantic region so that swell through the North Sea boundary from the Atlantic is resolved.

At a later stage, MME and ML-based local forecast will be developed. The multi-model forecast data sets have different spatial resolutions and cover different forecast ranges. These are aggregated to one data set using pre-calculated weights, currently using statistical Multi-Model-Ensemble (MME) methods but planning to use machine learning methods (ML-MME) as well in the future. Higher weights are assigned to forecasts with higher quality. Currently, CMEMS model forecasts from BAL MFC, NWS MFC, GLO MFC, DMI and HEREON wave height forecasts are used during the multi-model aggregation. The aggregated data set is provided twice daily, using the forecasts at midnight and noon. Due to the processing chain and run-time of ocean models which follow atmospheric model runs when they have been completed, the upload time to the DMI FTP server and the OLAMUR data management system can be delayed by 4-6 hours.

18. Module output and example

The module outputs are qualified weather windows in the next 120 hours at the three pilot sites. At the same time, forecast time series of 10 m winds (in m/s) and significant wave height (in meter) will be available.

The figures 5 and 6 show the mockup of how the visualisation of the service within the portal could look like. The specific visualisation will be presented in deliverable D5.3.

19. Module output validation

The module output validation includes a product quality validation and a service function validation.

Product quality validation: this validation will evaluate the quality of the forecast of the module products, i.e., forecast time series of 10 m winds and significant wave height, against observations. Considering that the operational demonstration will start in year three, the product quality validation will be performed in year three.

The module service function validation can only be made after the operational demonstration. This will include two parts: user feedback and statistics of system operational performance. The former will tell potential improvements of the service and the latter will tell the timeliness and stability of the system.

20. Potential for future development/generalization

Potential future improvements can be derived from the module output validation. This includes potential improvements on product quality (e.g. use of MME and AI to merge the model simulation with satellite observations to generate the optimized local forecasts), optimized service functions etc.



21. Module ASR - Alien Species Risk (DTU)

22. Short introduction and background

The ASR tool will support farms to run alien species dispersal risk assessment. The module v0 will build on the AquaNIS interactive online searchable database information system on aquatic non-indigenous and cryptogenic species (Olenin et al, 2014):

<http://www.corpi.ku.lt/databases/index.php/aquanis/>

The screenshot shows the AquaNIS website interface. At the top, there is a navigation bar with 'Home', 'Staff', and 'Credits' buttons. Below this, a grid of six main tiles provides access to different data categories: 'species' (with sub-headers for native origin, taxonomy, biological traits), 'introduction events' (with sub-headers for impacts, habitats, vectors), 'search' (described as a flexible multi-criteria search engine), 'geography' (geographical information arrangement), 'EARLY WARNING SYSTEM' (regionally harmonized early warning system concept), and 'services' (most widespread species, new arrivals). To the right of these tiles is a large image of a sea slug with the caption 'Copyright: Sergej Olenin. Codium fragile fragile.' Below the main content area, there is a sidebar on the left with a 'Public domain' section containing links like 'Home', 'Species accounts', 'List of species', etc. The main content area below the tiles contains an 'About AquaNIS' section with descriptive text and a 'LAST UPDATED INTRODUCTION EVENTS' list with five entries, each including a species name, location, and date.

Figure 8. Screen shot of the interactive search portal AquaNIS.

The system contains observations on NIS recorded as events with biological metadata. AquaNIS allows users to view data along different axes, e.g. by species, time and locations. Currently, the system contains data on NIS introduced to marine, brackish and coastal freshwater of Europe and neighbouring regions, i.e. includes HELCOM and OSPAR areas.

The ASR module perspective is to assess the risk of lowered LTA production due to the emergence of alien species at the farm, not to assess the risk that multi-usage sites act as stepping stones for dispersal of LTA irrelevant HAOP that may damage local ecosystems.

23. Module structure diagram

The OLAMUR consortium does not have access to the raw database underlying AquaNIS. The ASR module therefore accesses AquaNIS data through the online interactive query interface. Technically this is implemented as automated polling (dynamic web-scraping) from AquaNIS portal using a Python API based on the Selenium library.



"Funded by the European Union, grant agreement NO 101094065. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them."

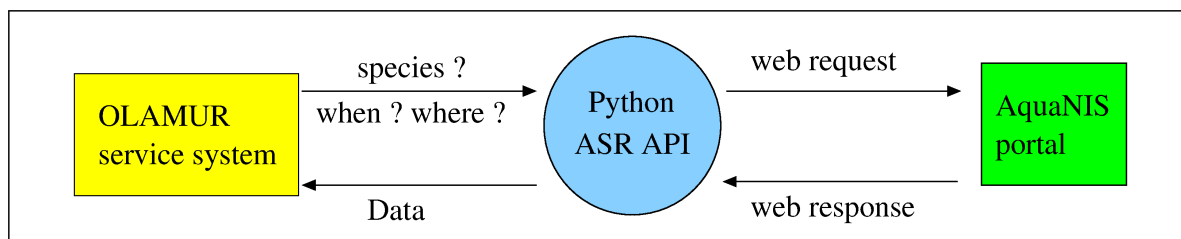


Figure 9. Information chain for the OLAMUR MU service system accessing the AquaNIS database via the AquaNIS online portal. The ASR API converts data queries to web requests and digest the web response from the AquaNIS server.

24. Module functionality

The ASR module allows to search introduction events dynamically as available in the AquaNIS database by various filter options.

25. Software/data requirements

Python3+ and the python-selenium on in service system.

26. Module implementation

The ASR module implements as a class *AquaNISbrooker* that facilitates HTML-based communication with the AquaNIS portal. An AquaNISbrooker instance instantiates a (by default) headless Firefox driver via the Selenium library that communicates with the AquaNIS portal via the advanced search feature, imitating an interactive session. Simpler web-scraping tools like *MechanicalSoup* were initially attempted, but given up, because AquaNIS is implemented using JavaScript, and the simpler web-scraping tools does not support JavaScript.

27. Input data/parameters

The ASR module requires no input data/parameters but is dependent on internet connection to the AquaNIS server.

28. Module usage

Instantiate an AquaNIS brooker to communicate with the AquaNIS portal:

```
import ASR
brooker = ASR.AquaNISbrooker()
```

The main method in version v0 is the `scan_introduction_list` that returns a list of tuples like (species, where, when) corresponding to the query; an empty list is returned if no records match the query. Examps of usage:

Scan AquaNIS for occurrences of a given species:

```
content = brooker.scan_introduction_list(species="Achnanthes pseudogroenlandica")
for item in content:
    print("%20s @ %20s @ %20s" % item)
```

Scan AquaNIS for occurrences at a given location:



```
content = brooker.scan_introduction_list(location="Latvia")
for item in content:
    print("%20s @ %20s @ %20s" % item)
```

29. Module output validation

The ASR module v0 only depend on the accuracy and timeliness of the AquaNIS database.

30. Potential for future development/generalization

The ASR module v0 is implemented as automated polling (dynamic web-scraping) from AquaNIS portal using a Python API based on the Selenium library. Web-scraping is prone to AquaNIS portal layout changes, and a direct dynamic access to the underlying database is preferable, however this is not supported currently. Further this portal-based coupling is dependent on the responsiveness of the AquaNIS portal, which has been an issue in the developmental phase of the coupling of AquaNIS to the OLAMUR service system.

AquaNIS is a useful tool to assess the dynamics of biological invasions at a regional and yearly time scale. However, for the purpose of early robust warning of LTA farms AquaNIS has shortcomings. The underlying database is updated on ad hoc basis by contributing scientists, i.e. a time lag between observation and accessibility is present. Observations are not derived from a dedicated, maintained, and comprehensive monitoring programme, but represent sparse samplings. AquaNIS entries has limited spatial resolution (LME / LME subregion / Port annotation) and yearly resolution. Therefore, future development efforts should target supplementary observational data source to complement AquaNIS. A new feature in AquaNIS is a harmful aquatic organisms and pathogens (HAOP) early warning. Hopefully, it will be possible to integrate this feature with the ASR module.

Currently a regional presence of a given species can be flagged. LTA farmers are likely more interested in a risk-oriented product. This requires considerations on how to weight AquaNIS AS observations at a certain distance/direction from LTA site to assess the propagation risk and propagation time scale, e.g. based on local connectivity metrics from Lagrangian simulation, if biology of AS is sufficiently confined. Such Lagrangian simulations are routinely conducted today (Hansen et al, 2024) with the access to real-time oceanography products from the Marine Copernicus service. One possible dissemination for traffic light classification of AS from LTA production perspective.

31. Module DPR - Disease Propagation Risk (DTU)

32. Short introduction and background

This tool implements a spatial risk assessment and will be developed for aquafarms for selected parasites. In relation to diseases in LTA farms three major questions are relevant:

- What is the risk that a given disease infects a farm and how large a fraction of the crops will be damaged?
- How can the farm be designed to lower the risk?
- What is the efficiency of alternative disease control options?



Answering these types of questions requires a quantitative approach to simulate the spatial disease propagation dynamics. The DPR module provides such a tool that enables users and planners to explore disease propagation dynamics for a generic ecologic template for single-stage parasitic epidemics for sedentary marine hosts. The goal of the module is to enable favourable site-layout and evaluation of alternative disease control options in relation to dynamic ocean conditions.

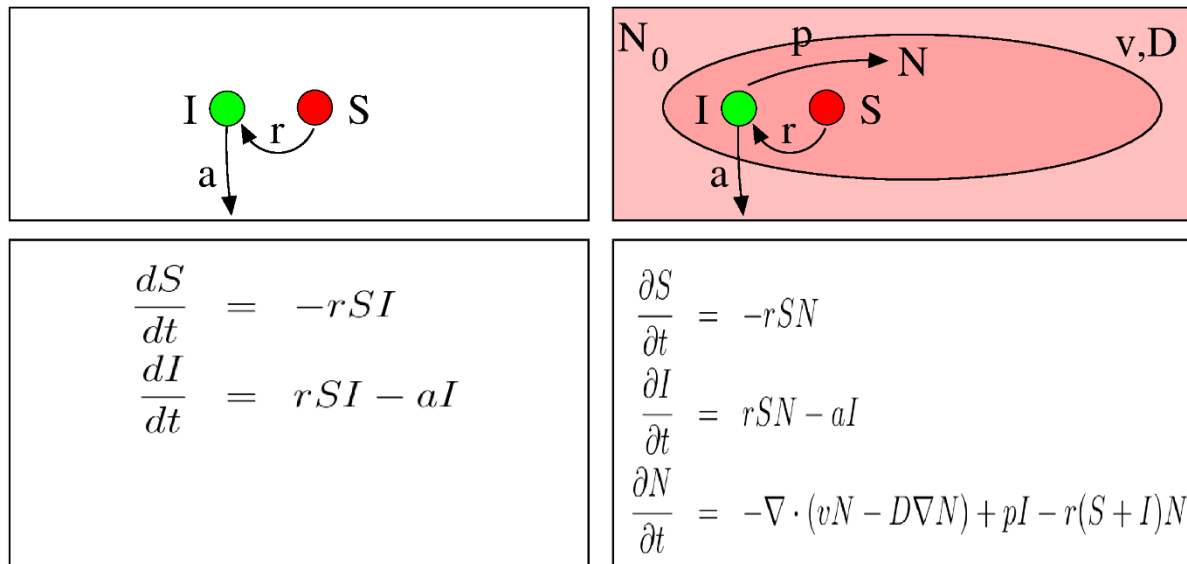


Figure 10. SIN model (left) compared to classical SI model (right) for simulating infectious dynamics.

The scientific basis for the model is the classical non-spatial SI model from (Kermack and McKendrick, 1991) for epidemics that also provided the starting point for epidemical forecasting in the Corona-period 2019-2022 (Fig 10, right side). That model has two state variables S (susceptibles) and I (infected), and two parameters r (specific infection rate) and a (recovery rate). When initial values (S_0 and I_0) are specified, the initial-value problem (IVP) is complete. Stage-structured extensions to represent biological details in epidemic processes follow from this skeleton model. Another direction of extension is to account for spatiality, which is implicitly boiled into the parameter r assumed constant. Whereas for terrestrial populations infection process can be considered local, whereby r scales with the overlap between $S(x,t)$ and $I(x,t)$, a pelagic transmission vector $N(x,t)$ is more common (Fig X.1, right). This extends the model from two to three state variable, which are now fields (depending on both position x and time t) describing density (per area) of susceptible, infected organisms and transmission vectors so that the mathematical problem becomes a boundary value problem (BVC) described by partial differential equations (PDE). Compared to the SI model, infection rate is proportional to $S \cdot N$ rather than $S \cdot I$. In the LTA situation organisms are stationary, so that dynamical equations for S and I do not have transport terms. Only the last dynamical equation for N describe the transport of the plume of transmission vectors, which includes both the ambient concentration (N_0) and locally generated contribution. The source term of the locally generated contribution is $p \cdot I$, where p is the production rate of transmission vectors by infected plants/bivalves. The first two terms on the right-hand side of the N equation are the transport rates by advection (v) and diffusion (D). In simplified simulations, $v(x,t)$ and $D(x,t)$ can be assumed constant at the spatial scale of the LTA farm $v(t)$



and $D(t)$, or even constant in time. The tool has provision for all options, from constant (v, D) to fully variable fields $v(x, t)$ and $D(x, t)$, obtained from an external data set (offline) or provided by online coupling with a hydrodynamic model describing water flow in the LTA farm. The last term in the N equation is the absorption of transmission vectors by infection processes (where infected organisms still has affinity for binding transmission vectors). The tool applies Dirichlet boundary condition $N=N_0$ at the simulation domain edge, where N_0 is the ambient concentration of transmission vectors. The simulation domain extends well beyond the LTA farm.

The tool is minimal and parsimonious including only an essential set of parameters that are SMART reflecting the common situation that finer details of infectious processes for a given parasite are not well confined. Compared to the classical SI model, just one new parameter p is introduced, along with a dynamical formulation $r(v, D)$ of the local infection rate:

$$r(v, D) = r_0 R v \left(1 + \frac{8}{3} \sqrt{\frac{DR}{\pi v^3}} \right)$$

where R is the cross-distance scale of the host organisms and r_0 is a sticking (and transmission) probability of a transmission vector, given that it hits the host organism. This expression is valid in the advective regime $DR \ll v^3$, and the second term express the leading contribution by diffusion processes in enhancing contact rate between host and transmission vector over the pure advective clearance volume rate. An expression in the diffusive regime $DR \gg v^3$ depends on the numerical forward integration step, since a micro-scale distribution limit is not established within the numerical forward integration step. The model above is implemented in 1D (downstream axis, for conceptual analysis and speedy simulations) and horizontal 2D (allows to access farm layout performance and directly couple to hydrographic data sets with varying current fields).



33. Module structure diagram

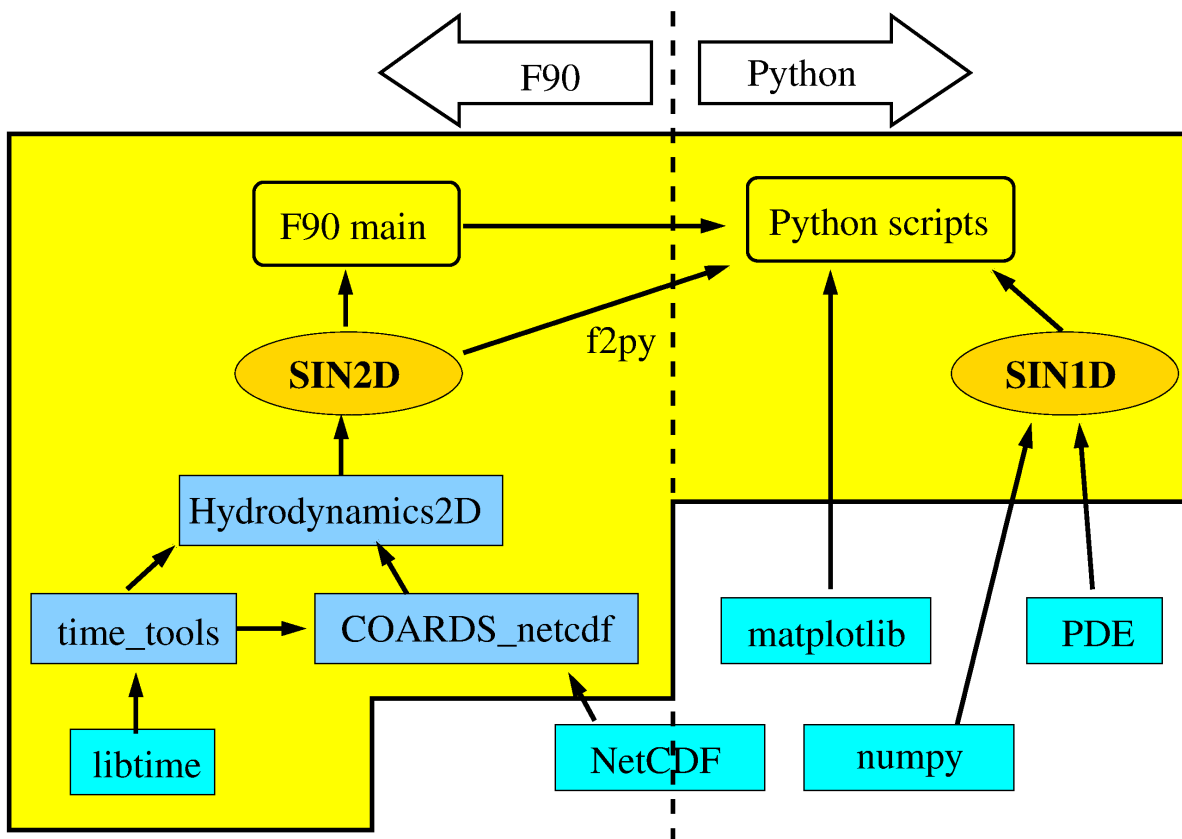


Figure 11 Components and relations in the DPR module. Dashed line separates F90 and Python components. SIN1D and SIN2D are main import points. The yellow area is packed in DPR. Cyan boxes are external standard code packages (dependencies) that must be installed prior to DPR. Light blue boxes are support modules. Optional hot coupling between F90/Python are facilitated by f2py.

The 1D (downstream axis) version is written in Python; the horizontal 2D version in Fortran 90 (hereafter F90) for increased speed. F90 is a frequent choice for performance code since its syntax allows for highly optimized numerical processing. Many leading high-performance codes are written in F90, e.g. NEMO (ocean circulation). The price for speed is that code needs to be compiled (pre-translated), which is slightly less convenient than interpreted code (e.g. Python or Matlab), however compilation is a standardized procedure initiated by a single command. The SIN2D side can be run independently of python, and conversely can SIN1D be run independently of the F90 components.

34. Module functionality

The main functionality of the DPR module is the ability to simulate the epidemic evolution of a simple parasitic disease with a pelagic dispersal phase. State variables and parameters are easily accessible in code modules allowing for easy experimentation. Data during or by the end of simulations can be exported in NetCDF format via the 2D COARDS_netcdf module which is an object-oriented API to the underlying NetCDF format. Auxiliary fields can also easily be read using this module.



Coupling to dynamic hydrography is facilitated by two fields: $v(x,t)$ (local currents at x and time t) and $D(x,t)$ (diffusivity, causing mixing due to subgrid scale eddies). In simplified simulations, $v(x,t)$ and $D(x,t)$ can be assumed constant at the spatial scale of the LTA farm. This option is supported both for SIN1D and SIN2D. Alternatively SIN2D v1.0 supports offline simulations where any COARDS compliant 2D hydrographic data can be loaded along the simulation. Any space filling hindcast/reanalysis oceanographic data products at e.g. CMEMS are stored in COARDS compliant format. The provided hydrographic data are interpolated in space and time to the simulation grid automatically, so data grids need not be coherent with SIN2D simulation grid. SIN1D v1.0 only support constant values of (v,D)

Forward simulations are conducted by tailoring a F90 template like `testrun.f90`, and compiling and linking this to other binaries, or tailoring a Python template like `testrun.py` and running it within a python environment as a script. For transparency, we have conducted same simulation in either way below, for comparison.

In the SIN2D examples below the simulation grid is 50x50 with 20 m resolution. A simulation with 50000 time steps is conducted, with timestep 8.64 sec, so the simulation corresponds to 5 days. We compared simulations with constant values $(v_x, v_y, D) = (0.5 \text{ m/s}, 0, 0.01 \text{ m}^2/\text{s})$ and realistic hydrography from the CMEMS product “Baltic Sea Physics Analysis and Forecast” (prodID `cmems_mod_bal_phy_anfc_PT15M-i`) with 2km horizontal resolution and 15-minute time resolution, subset to the Western Baltic Sea. The constant values (v, D) simulation took 0.76 sec, whereas the latter with variable hydrography took 6.47 sec on a laptop. This indeed indicates that the SIN1D/SIN2D tools are well-suited for interactive applications.

The three next figures show usage examples. Figure 12 shows running SIN2D using a driver F90 main program (`testrun.f90`) linked to the SIN2D library. `Init_sin2D` setup the simulation grid. `Set_Dirichlet_BC` sets boundary conditions to Dirichlet (constant value N_0 of $N(x,t)$) at the rim of the simulation domain; this corresponds to an assumption of a constant ambient spore concentration that may trigger an epidemic in the LTA farm. The subsequent lines set biological parameters, which are public in the F90 module scope. The next section sets the initial conditions for state variables (S, I, N) . S and I are non-movable. The LTA farm is outlined by setting the initial concentration of susceptible seaweed/bivalves $S_0 = S(x,0)$ with $S_0 = 0$ outside the farm. The main loop the conducts 50000 time steps of size $h=8.64$ seconds, calling the propagator subroutine `forward_step`; this name is the same for all PDE solvers, so PDE solver is currently chosen at compile time for SIN2D. Finally, the last parts show an example of saving output in plain text format, before invoking `finalize_SIN2D` that deallocates memory occupied by the execution.



```

testrun.f90
File Edit Options Buffers Tools F90 Help
! =====
! Basic test of forward integration for an LTA scene
! Feature hydrographic interpolation at Kriegers Flak @ (13,55) / constant hydrography
! =====
program testrun
  use SIN2D          ! solves PDEs
  use hydrodynamics2D ! interface to hydrography

  implicit none
  ! =====
  real(dpr),parameter :: N0 = 0.01      ! ambient spore concentration
  real(dpr),parameter :: h = 8.64      ! time step [sec]
  integer,parameter   :: nx0=10, nx1=25, ny0=20, ny1=30 ! LTA bounding box
  integer              :: ix,iy,it,j
  ! =====
  call init_sin2D(50, 50, 20.d0) ! set isotropic simulation grid: nx, ny, dx [meters]

  ! --- select hydrography setup ---
  !call init_hydrodynamics2D const(0.5d0, 0.0d0, 0.01d0) ! constant vx,vy,D
  call init_hydrodynamics2D_xyt_netcdf("../HydroData/KFhydro_test.nc", &
                                       (/13.d0,55.d0/), dx) ! dynamic hydrography

  call set_Dirichlet_BC(N0)

  r = 2.0e-6      ! sin2D scope: infection rate
  a = 1/10/86400. ! sin2D scope: curing rate
  p = 1000/86400. ! sin2D scope: spore production rate
  lamb = 0.d0     ! sin2D scope: background removal rate of N

  ! ===== initial conditions =====
  S = 0.d0
  I = 0.d0 ! no infected initially
  N = N0   ! N = ambient spore concentration initially

  ! ----- setup LTA farm -----
  do ix = nx0, nx1
    do iy = ny0, ny1
      S(ix,iy) = 1.0
    enddo
  enddo

  ! ===== main loop: integrate forward the system =====
  do it=1,50000
    call forward_step(h)
  enddo

  ! --- save data, example: line scan
  do ix = 1, nx
    write(*,'(i4.4,3x,3e15.7)') ix, S(ix,25), I(ix,25), N(ix,25)
  enddo

  call finalize_sin2D() ! close modules nicely
  !
end program
-:--- testrun.f90 All L41 (F90 -1)

```

Figure 12. Running SIN2D using a driver F90 main program (testrun.f90) linked to the SIN2D library.



```

testrun.py
File Edit Options Buffers Tools Python Help
#!/usr/bin/env python
# =====
# Redo testrun.f90 from python
# =====
from numpy import *
from sin2D import sin2d
from sin2D import hydrodynamics2d
nx,ny = 50, 60
sin2d.init_sin2d(nx,ny, 20.0) # nx,ny,dx
hydrodynamics2d.init_hydrodynamics2d_const(0.5, 0.0, 0.01)
hydrodynamics2d.init_hydrodynamics2d_xyt_netcdf("../HydroData/KFhydro_test.nc",
[13.,55.], 20)

sin2d.r = 2.0e-6 # sin2D scope: infection rate
sin2d.a = 1/10/86400. # sin2D scope: curing rate
sin2d.p = 1000/86400. # sin2D scope: spore production rate
sin2d.lamb = 0. # sin2D scope: background removal rate of N

N0 = 0.01
h = 8.64

# ===== initial conditions =====
sin2d.set_dirichlet_bc(N0)
sin2d.n = N0*ones((nx,ny), float)
sin2d.s[9:25,19:30] = 1.0

# ===== main loop: integrate forward the system =====
for it in range(50000):
    sin2d.forward_step(h)

# --- save data, example: line scan
for ix in range(nx):
    print(ix, sin2d.s[ix,24], sin2d.i[ix,24], sin2d.n[ix,24])

# ---- plot N distribution at end time ----
import matplotlib.pyplot as plt
plt.matshow(sin2d.n)
plt.show()

-:***- testrun.py All L3 (Python E1Doc)

```

Figure 13. Running SIN2D using a python script.

Figure 13 shows running same SIN2D job using a python script. This requires that the Python binding to SIN2D has been generated using f2py, so that SIN2D can be imported as a Python module. At the end of this script is also shown an example of plotting $N(x,t)$ from Python using Matplotlib



```

testrun_1D.py
File Edit Options Buffers Tools Python Help
1 #!/usr/bin/env python
2 # =====
3 #           Minimal example of using SIN1D
4 # =====
5 from numpy import *
6 import SIN1D
7
8 SIN1D.ng = 50      # number of grid points
9 SIN1D.L = 1       # farm length scale; grid scale dx = 2*L/ng
10 SIN1D.r = 1      # infection rate
11 SIN1D.a = 1.0    # curing rate
12 SIN1D.v = 0.5   # advection velocity
13 SIN1D.D = 0.01  # diffusivity
14 SIN1D.p = 0.1   # spore production rate
15 SIN1D.N0 = 0.01 # ambient spore concentration
16 SIN1D.grid = SIN1D.CartesianGrid([[0, 2*SIN1D.L]],
17                                   [SIN1D.ng], periodic=False) # simulation grid
18
19 from SIN1D import *
20 # initialize the equation and the space
21
22 xg = grid.cell_coords[:,0]
23 Si = ScalarField(grid, where(logical_and(xg>0.5*L, xg<1.5*L), 1, 0))
24 Ii = ScalarField(grid, 0)
25 Ni = ScalarField(grid, N0)
26
27 istate = FieldCollection([Si,Ii,Ni])
28
29 # solve the equation and store the trajectory
30 storage = MemoryStorage()
31 eq = SIN1D()
32 res = eq.solve(istate, t_range=1, method="scipy", tracker=storage.tracker(0.1))
33
34 # plot the trajectory as a space-time plot
35 spc = 80* " "
36 title = "S" + spc + "I" + spc + "N"
37 kymo = plot_kymographs(storage, title=title, filename="jj.png")
38
:--- testrun_1D.py All L39 (Python ElDoc)
(No changes need to be saved)

```

Figure 14. Running SIN1D using a python script.

Finally, figure 14 shows running SIN1D using a python script; SIN1D is written completely in python, based on the PDE solver environment PDE. Line 8-15 sets the biological and physical parameters. Line 16-17 sets the simulation grid. Line 19 reimport all SIN1D content to main scope, for simplicity. Line 22-27 sets initial conditions. Line 31 instantiates the simulation object, and line 32 solves (integrates forward in time) the SIN1D model. Line 37 shows an example of plotting the kymograph for the system dynamics.

35. Software/data requirements

SIN1D and SIN2D can be installed separately. The modules do not apply platform-specific functionality or features, so all parts should be installable on main platforms (*nixes, Mac and Windows), if dependencies are properly installed. All software dependencies below are open source or licensed for free.

SIN1D requirements:



"Funded by the European Union, grant agreement NO 101094065. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them."

- Python 3+
- py-pde: <https://py-pde.readthedocs.io>. Dependencies (including numpy and matplotlib) are described on this page.

SIN2D requirements:

- An F90+ compiler; Intels compiler (<https://www.intel.com/>) and GNUs gfortran (<https://gcc.gnu.org/fortran/>) are free, and we have compiled SIN2D with these. There are several licensed fortran compilers as well
- Usage NetCDF for I/O requires a version installed that is commensurable with the F90 compiler. Using COARDS compliant hydrography requires NetCDF (<https://www.unidata.ucar.edu/software/netcdf/>)
- F2py: to use python bindings and scripting requires f2py installed, which implies Python 3+ and numpy as well. The package NetCDF4-python (<https://github.com/Unidata/netcdf4-python>) is recommended as well in this context.
- Developer tools: gmake for easier building (<https://www.gnu.org/software/make/>); a basic makefile comes along with SIN2D.

36. Module implementation

SIN1D: this module uses the method of lines to solve the PDEs of the 1D SIN model. Spatial operators are discretized using finite element approximations which turns the PDEs into a high-dimensional coupled ODE system, which are integrated in time using high-order time algorithms. The module features the class SIN1D that represents the system. The process representation of the system rates is in the method *evolution_rate(self, state, t=0)* in a compact form making further experimentation easy. Grid and field classes are defined in py-pde.

SIN2D: this module is implemented with 2 PDE solvers:

1. A simple Euler-forward and upwind scheme for advection and centered 5-point formula for Laplacian
2. An implicit Crank-Nicolson solver based on a sparse-matrix algebra.

Due to the very high spatial resolution in the model, the hydrodynamic Courant condition mandate a quite small time integration step, which is significantly smaller than the intrinsic epidemic time scale; therefore, the gain of scheme 2 over 1 is not obvious for the nature of this problem, in the light of the algorithmic simplicity of scheme 1. Integration scheme is selected at compile time. Forward step algorithms follow same generic API:

Subroutine forward_step(real*8 :: h [sec])

 Call update_hydrodynamics(time + 0.5*h) ! Here mid point; other points may appl

 Propagate state variables S,I N by time step h, according to forward integration algorithm

 Call apply_BC() ! apply boundary conditions

 Time = time + h ! update time counter

End Subroutine



Data grid: regular Cartesian with arbitrary zonal (n_x) and meridional (n_y) number of mesh points and isotropic grid spacing (dx). Earth curvature effects are neglected at the scale of LTA applications. The hydrodynamics2D module provides time frame interpolation and time frame caching to minimize time consuming frame load.

Object oriented coding style has been adhered to allowing future development and updates to be conducted smoothly. In-code documentation and detailed API documentation along with code.

Support modules: describe types

37. Input data/parameters

Biological parameters		
r0	Sticking probability of infective vector	-
a	Loss rate of infected seaweed/bivalves	1/sec
p	Production rate of infective vectors by infected	1/sec/infected
lambda	Loss rate of pelagic infective vectors	1/sec
Technical parameters		
n_x	Zonal grid points	-
n_y	Meridional grid points	-
dx	Grid spacing	meter
h	Forward integration step	sec
N0	Ambient concentration of pelagic infective vectors	meter-2

Table 5. Biological and technical parameters of the SIN2Dmodel

Hydrographic data: offline data should be stored in COARDS compliant format (NOAA, 2024). Currently, unfortunately, most circulations models do not store horizontal diffusivity. Therefore, a constant value D_0 is assumed, or a dynamic value assessed *a posteriori* using the Smagorinsky expression (Smagorinsky 1963).

38. Module output

Product: light-weight on-demand PDE solver that computes relative infection risks for different LTA layout/orientations, given hydrodynamics (v, D). Useful for assessing

- Scenario response envelope
- Dynamical time scales
- Identify dynamical patterns



39. Module output example

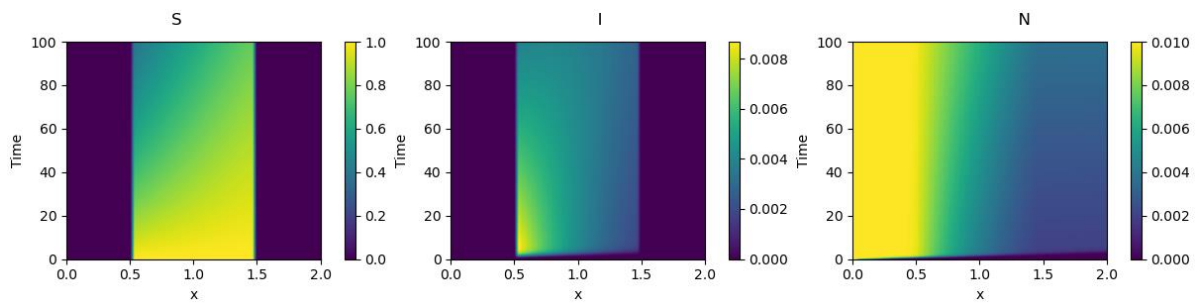


Figure 15. Kymograph example of state variables S (susceptibles), I (infected), and N (infectious spores) in a non-dimensional simulation of $SIN1D$. The LTA farm is situated between $0.5 < x < 1.5$, the time axis shows evolution of the system for $0 < t < 100$. The figures show how the epidemic start in the left (upwind) side of the farm, and spreads to the rest of the farm.

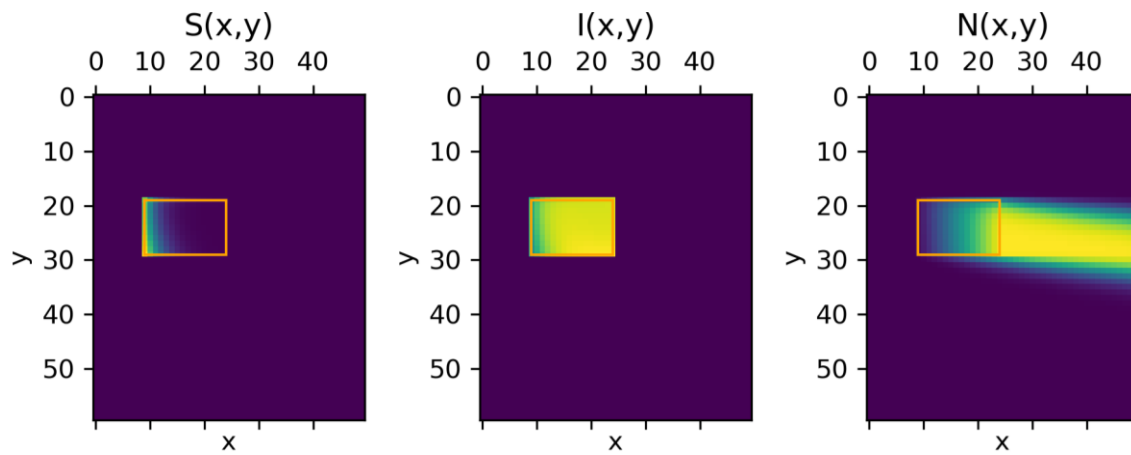


Figure 16. Simulation end state of a $SIN2D$ simulation using hydrography from CMEMS product ID `cmems_mod_bal_phy_anfc_PT15M-i`. The farm area is indicated by orange. The plot shows that the Eastern part of the farm is heavily infected. The right-most plot shows the plume of infective spores generated by infected seaweed/bivalves.

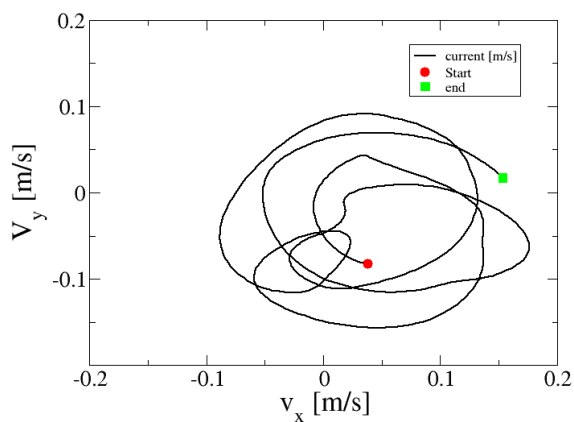


Figure 17. Average current (v_x, v_y) in the farm during the simulation in Figure 16. The tidal cycles are clearly visible, with an Eastward residual current in the simulation period. Data from 2024-05-09 to 2024-05-10 of CMEMS product ID *cmems_mod_bal_phy_anfc_PT15M-i*

40. Module output validation

Module validation is pending on further data gathered during the project. Data sets are not currently available.

41. Potential for future development/generalization

A list of most relevant parasites for seaweed and bivalves in LTA has been identified. For *Saccharina latissima*, the following are most relevant:

- *Pseudoalteromonas piscicida* (bleaching disease)
- *Iteromonas* sp.
- *Laminariocolax* sp.
- *Laminarionema* sp.

For bivalves, the most relevant parasite is:

- *Mytilicola orientalis*

For the rest of the project, a database of parameter ranges for SIN2D biological parameters (Table X.1) will be estimated in consultation with WP3 and available literature. On the process-oriented side there is some factors whose influence on model results and conclusions need to be checked. These are not limited to:

- Lag effects: in the current v0 formulation, infection and onset of spore production is immediate. There is a certain time lag associated with these processes. This can relatively easily be checked by shunting additional infected states, which do not produce spores.
- Organism infection severity: in the current v0 formulation, infection of individual organisms in Boolean and not associated with a severity. At a technical level, this can likely be addressed together with lag effects as a first approximation.
- Influence of vertical dimension: in the current v0 formulation, the model is vertically averaged. The vertical loss of transmission vectors is boiled into the chemostat loss term, where lambda accounts for vertical mixing effects. A logical extension of the model is to consider a 3-layer vertical approximation for N: above farm, within farm and below farm

On the technical side, some factors affecting model computational efficiency and model operational feasibility needs to be checked. These are not limited to:

- Online coupling of SIN2D: the model can relatively easily be coupled online to hydrodynamic models for water currents written in F90/C++, as the SIN2D model is well encapsulated and portable. In F90 implementations, basically a few lines need to be added (use association, initialization, state propagation, output and finalization) and a pass of hydrodynamic data to SIN2D



42. Module FPC - Farm Performance Control (IMR)

43. Short introduction and background

This module aims to provide stakeholders with a tool to track the crop status during the production cycle. The information provided by this module allows for planning in situ activities and aids the optimization of the production cycle and the maintenance and harvesting tasks during production. Cultivating low-trophic species in remote offshore areas means that shipping costs and meteorological conditions limit access to the facilities. Moreover, offshore environments are very dynamic, and aquaculture facilities will be subject to events that may impact the farm performance in a significant manner such as storms or algae blooms. Due to the logistic demand, it is expected that these aquaculture infrastructures will not be monitored directly very often; instead, stakeholder will rely on tools for remote monitoring and models. In this scenario the FPC module will provide the stakeholders a virtual status of the crop to help them plan maintenance and activities during the production cycle.

The Farm Performance Control module will rely on the offline ocean data from CMEM and other sources of updated time series of ocean conditions together with output from the models developed by WP4 and WP3 to assess the environmental conditions at the farm and the condition of the crop within the farm respectively. FPC module will allow the users to explore the information provided by the models on crop population structure, density and biomass, growth, mortality, individual condition, reproductive status and the environmental conditions at the farm location (oxygen, temperature, salinity, algae blooms, etc).

44. Module structure diagram (visual)

The aim of the module is to be a tool that can be used in all farms and it all pilots. The FPC functionality revolves around the Dynamic Energy Budget population models developed in WP3, integrating these within a farm structure using in-situ environmental conditions forcing to perform simulations of the evolution of the crop within the farm.

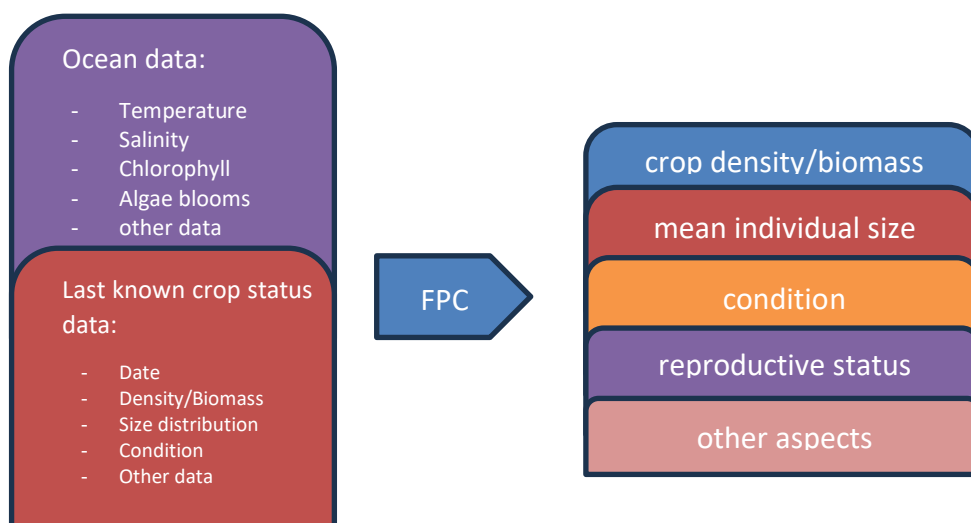


Figure 1. Module functionality diagram showing input and output data.



The FPC integrate the design of the farm: orientation, dimensions, distance between lines, type of low trophic farm and a given initial crop status to be used as a starting point of the simulation.

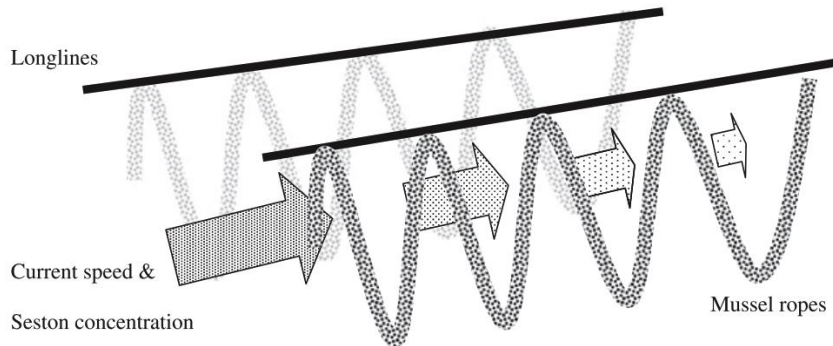


Figure 2. Example of farm structure for a mussel longline (Rosland et al. 2011)

45. Module functionality

The FPC module relies on a farm-scale model to simulate the biological activity of the crop under the farm location biotic and abiotic conditions. While ecosystem models, as such developed in WP3 allow for a larger scale simulation including large scale processes and ecosystem-level impacts, their use in short and almost real-time simulations of the crop become difficult due the needs of computation time to resolve processes at a large scale. Meanwhile, the use of a farm-scale model forced with location specific inputs allows for a fine spatial resolution within the farm which can then incorporate the complex nature of the interactions between the crop and the local hydrodynamics in an efficient manner. The inclusion of this interaction at this level allows for the detailed spatial description of the intraspecific competition for food and space within the farm allowing a more accurate determination of the feeding, growth, biomass and condition of the crop within the farm. As such a farm-scale model provides a better simulation tool for the crop population dynamics and its interaction with the environment.

The FPC module builds up on the farm model developed by Rosland et al. (2011). Currently the farm model is developed for a long-line farm design with a determined orientation and distance between the lines that are assumed to be parallel and identical. Further, it assumes that the friction caused by the interaction of the suspended canopy with water currents gradually decreases the water velocity downstream the farm. Considering the Chezy formula to represent the friction (see Aure et al. 2007) and Navier-Stokes equation for the conservation of momentum, the current velocity at any point of the longline (v , ms^{-1}) can be calculated as follows:

$$v = v_1 \cdot \exp(-\delta \cdot x) , \text{ with } \delta = \frac{2 \cdot f_k}{L_s}$$

where v_1 is the background velocity (ms^{-1}), f_k is the frictional constant, L_s is the spacing between longlines (m), and x is the distance along the longline direction (m). This approach to model current velocity along the longline is then integrated with the biological activity of the crop and thus the DEB-IBM model for the crop population by linking the effect of nutrients



capture by the crop population. Following Bacher et al. (2003), the former equation can be used to determine the nutrient concentration at any of point of the longline (S , mass m^{-3}) as:

$$S = S_1 \cdot \exp\left(\frac{C_t}{v_1 \cdot \delta} \cdot (1 - \exp(\delta \cdot x))\right)$$

where S_1 is the background nutrient concentration (mass m^{-3}), and C_t is the clearance rate (s^{-1}), which is a function of the individual clearance rate (C_i , $m^3s^{-1}ind^{-1}$), estimated with the Dynamic Energy Budget (DEB) model, and the population density ($ind\ m^{-3}$). The two-way coupling of the farm model with DEB occurs at the feeding/nutrient uptake functional response of DEB, where the clearance rate is calculated as a function of nutrient concentration (C_t). Moreover, the effect of other variables on the physiological responses of the cultivated species which are calibrated in the DEB models developed in WP3, such as temperature and salinity, are considered in the FCP. Other population dynamics mechanisms, such as natural mortality, which are included in the WP3 DEB-IBM models are also applied in the simulations performed by the FCP.

46. Software/data requirements

Version 0 of the FPC is developed in MatLab for blue mussel longline farming using NetCDF files containing the ocean data forcing (currently only temperature, chlorophyll concentration and current velocity and direction) which are extracted for the location of the low trophic farm, and input variables describing the farm orientation, a constant friction coefficient, distance between lines and their length, and the initial status of the crop as a minimum: crop density (numbers per meter of longline) and individual size.

47. Module implementation

The module will run as a web service using data from the OLAMUR data management system provided by WP3 and WP4. The service will organise the sourcing of data from different providers (CMEMS or others) to be available for the demonstration sites.

48. Input data/parameters

The FCP model will run using the parameters calibrated for the DEB and the other mechanistic models developed for the cultivated species in the WP3.

The FCP also requires parameters regarding the farm design meters of line, orientation, distance between lines, depth of the lines, friction coefficient for the farm infrastructure (longlines vs SmartFarm®, low vs larger density, size dependent)

The FCP will be forced by inputs of salinity, temperature, chlorophyll and hydrodynamics available at the OLAMUR data services or other data sources for the demonstration sites. The FCP module will require a starting crop status value, this can be just a seeding density at the start of the production cycle or in situ recordings of crop status done remotely (sensors/robotics) or during a maintenance event. The FCP will then use the data available to update that crop status to the current running date with the latest available data on the environmental conditions for the farm location provided by WP4 prediction models and in-situ recordings from deployed sensors.

49. Module output

The output of the FPC farm-scale model is a spatially explicit (along the longlines) assessment of the crop population and the environmental conditions of the farm, including:

- Mean individual size (shell length and weight)



- Condition (a weight to shell length ratio)
- Reproductive output (gonad index)

And a whole farm assessment of the competition of food. Density and biomass and growth during simulated period vs environmental conditions.

50. Module output example

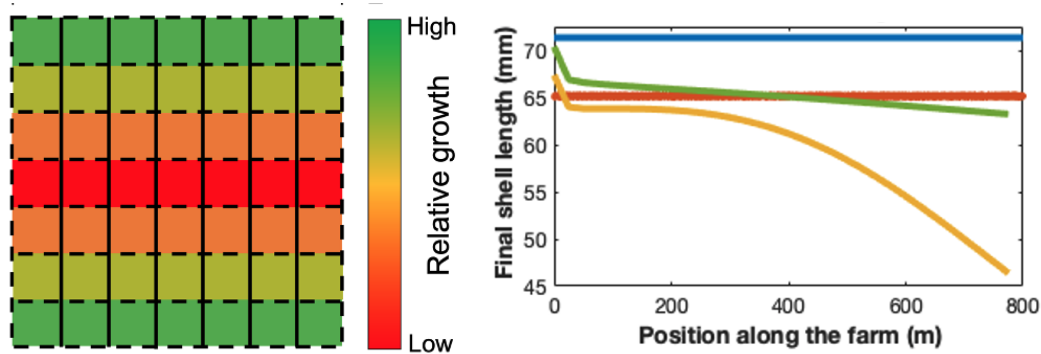


Figure 3. Some examples of possible outputs after running the FPC. Left: spatial evaluation of relative growth in a 800 by 800 meter farm with vertical longlines separated by 100m and assuming a bidirectional vertical water flow) that could be used to improve farm design or to assess the need to redistribute the crop biomass. Right: an estimation of shell length along the longlines of different farms in a real variable flow intensity and direction after 2 years simulation from seeded lines.

51. Module output validation

Model output will be validated against crop data from the different pilots. These data are collected in WP1 and available for WP3 to calibrate/validate the models and for the validation of this module.

52. Potential for future development/generalization

The potential development of the FCP will depend mainly on the availability of extra data sources. The implementation of the models in WP4 and the forecast for ocean conditions, including the prediction of phytoplankton and bacteria blooms could be used in the FCP to provide feedback of the crop to those events.

Moreover, the FCP will benefit from the further development of the models in WP3, adding extra mechanisms such as calcification, ecotox mechanisms and population dynamics mechanism such as improved natural mortality, mortality due to storms and predation risk evaluations.

53. Discussion

The description above of the five service modules constitute a skeleton for a future open tool environment for aquaculture operation in a multi-usage context. The web service framework is extensible with further service modules when needs are identified in collaboration with stakeholders and a technical implementation compatible with the T5.3 docking is available. The design choice of a web service framework avoids reduces the necessity of installing



additional software components and downloading data on the client side and avoid licensing issues of redistributing third party components.

The next steps in the path toward D5.3, integrating service modules described in this synopsis, initially involves the subtasks below across all five modules:

- Technical optimization of tool components: this involves code/dataflow profiling and identification of potential bottlenecks.
- Consolidate portability of the tool; this involves designating a list of supported platforms (browser and client OS versions), and collect list of dependencies, including minimal/problematic versions for internal use.
- Formulating a stakeholder testing plan; this includes setting up a stakeholder beta test group.
- Initiate user-oriented online documentation in an efficient format (Searchability, tutorials, videos, FAQs, installation guides etc).
- Conduct tool coherence/usability analysis – assess whether there is holes in functionality?
- Setup an easy installation flow at ETT for internal convenience.
- Setup tool in a versioning system.
- Discuss the possibility of presenting the tool concept as a peer-reviewed paper in an ecological software-oriented journal to create attention about the tool and possibly attract commercial partners.

After these initial steps, the tool development should enter an update cycle, involving these steps below:

- Development tracking system, incl. issue flow system (issue reporting, feedback, actions, archiving).
- Tool test suite for new releases.
- Living development priority list.
- Version tagging schedule.

54. References

Aure, J., Strohmeier, T., & Strand, Ø. (2007). Modelling current speed and carrying capacity in long-line blue mussel (*Mytilus edulis*) farms. *Aquaculture research*, 38(3), 304-312.

Bacher, C., Grant, J., Fang, J., Zhu, M., & Besnard, M. (2003). Modelling the effect of food depletion on scallop growth in Sungo Bay (China). *Aquatic Living Resources*, 16(1), 10-24.

Frishfelds V, J She, J Murawski, JW Nielsen (2023). Aggregating Sea Surface Hydrodynamic Forecasts From Multi-Models for European Seas. *TransNav: International Journal on Marine Navigation & Safety of Sea*. 17(3). DOI: 10.12716/1001.17.03.04

Hansen, F.T., Pastor, A., Christensen, A. and Stuer-Lauridsen, F. (2024). Using biophysical modelling and marine connectivity to assess the risk of natural dispersal of non-indigenous



species to comply with the Ballast Water Management Convention. *Biol Invasions*
<https://doi.org/10.1007/s10530-024-03327-0>

Kaspar, F., Niermann, D., Borsche, M., Fiedler, S., Keller, J., Potthast, R., ... & Tinz, B. (2020). Regional atmospheric reanalysis activities at Deutscher Wetterdienst: review of evaluation results and application examples with a focus on renewable energy. *Advances in Science and Research* 17, 115-128.

Kermack, W; McKendrick, A (1991). "Contributions to the mathematical theory of epidemics – I". *Bulletin of Mathematical Biology*. 53 (1–2): 33–55

NOAA 2024: COARDS NetCDF Conventions
<https://ferret.pmel.noaa.gov/Ferret/documentation/coards-netcdf-conventions>

Olenin S, Narščius A, Minchin D, David M, Galil B, Gollasch S, Zaiko A (2014) Making non-indigenous species information systems practical for management and useful for research: an aquatic perspective. *Biological Conservation*, 173, 98-107.

Pein, J., Staneva, J., Mayer, B., Palmer, M. D., & Schrum, C. (2023). A framework for estuarine future sea-level scenarios: Response of the industrialised Elbe estuary to projected mean sea level rise and internal variability. *Frontiers in Marine Science* 10, 1102485.

Rosland, R., Bacher, C., Strand, Ø., Aure, J., & Strohmeier, T. (2011). Modelling growth variability in longline mussel farms as a function of stocking density and farm design. *Journal of Sea Research*, 66(4), 318-330.

Smagorinsky J. (1963). General circulation experiments with the primitive equations. I. The basic experiment. *Monthly Weather Review*. 91(3):99–164.

Appendix I

1. Technical details for operational data download and upload

This appendix applies to both modules EWS and OOP.

1.1. Atmospheric data from DMI Harmonie model

Atmospheric data from DMI Harmonie model is produced on a rotated lat-lon grid covering and area from Greenland to the Eastern Baltics. The data is available as hourly forecast files in grib1 format. In a first step, the grid is de-rotated and the the 10m winds are extracted using CDO (Climate Data Operators, <https://code.mpimet.mpg.de/projects/cdo>).

Step1: Select wind components:



```
grib_copy -w indicatorOfParameter=33/34,level=10,indicatorOfTypeOfLevel=103 input
output
```

Step2: de-rotate and map to regular lat-lon grid

```
cdo rotuvNorth,33,34 input output
cdo genbil,remapwght infile remapweights.nc
```

where the remapwght is text file stating the desired lat-lon output grid:

```
gridtype = lonlat
xsize = 500
ysize = 350
xfirst = 5
xinc = 0.04
yfirst = 53
yinc = 0.02
```

Step3: Afterwards the hourly grib files are converted to netcdf using cdo. The hourly files are merged and the time axis is corrected.

```
cdo -f nc4c -z zip_6 copy input output
cdo -f nc4c mergetime input output
cdo -f nc4c settunits,hours input output
cdo -f nc4c -z zip_6 settaxis,gribdate0,1hour input output
```

where gribdate0 is start time of the forecast.

Step4: Next the global attributes to NetCDF are added:

```
fun_st='${The OLAMUR project is funded by the European Union. Grant no. 1011094065. Views
and opinions expressed are however those of the author only and do not necessarily reflect
those of the European Union or the CINEA. Neither the European Union nor CINEA can be held
responsible for them.}'
tit="DMI - bal - NEA - DMI"
cdo -f nc4c -z zip_6
setattribute,dataset_id=olamur_wp5_bal_NEA,funding_ID=1011094065,funding_statement
="${fun_st}",institution_edmo_code=469,Project_DOI=doi,QC_indicator=2,qc_manual=0,q
c_method=0,title="${tit}" input output
```

Note: the funding statement cannot include commas if using cdo attribute command.

Step5: Next the forecast is filtered to 3 Olamur sites:

```
cdo -f nc4c -z zip_6 sellonlatbox,7,9,53,55 input output
```



```
cdo -f nc4c -z zip_6 sellonlatbox,12,14,54,56 input output
cdo -f nc4c -z zip_6 sellonlatbox,21,23,58,59 input output
```

1.2. Wave model data from DMI WAM model

The data is provided by the DMI: North Sea – Baltic Sea model (NSB), and the model for the North Sea/Baltic Sea transition zone (IDF - Inner Danish Waters) with higher resolution.

1. WAM results for IDF are used for Olamur sites in Helgoland and Kriegers flak.
2. WAM results for North Sea – Baltic Sea are used for Tagalaht Olamur site.

Step1: Parameter selection: significant wave height, mean wave period and mean wave direction, which are extracted from DMI WAM grib files:

```
cdo -f nc4c select,name=swh,mwp,mwd input output
```

Step2: Add global attributes to the NetCDF files

```
cdo -f nc4c -z zip_6 setattribute,...
```

Step3: Finally the output result is filtered to respective sites with

```
cdo -f nc4c -z zip_6 sellonlatbox,...
```

1.3. Data processing of Copernicus Marine Service data, Scatterro winds

Download and processing of satellite observed global scattered winds. All Copernicus Marine Service data are downloaded and processed once a day.

Step1: Data download from Copernicus Marine Service

```
copernicusmarine subset --force-download --overwrite-metadata-cache --dataset-id
cmems_obs-wind_glo_phy_nrt_l4_0.125deg_PT1H $europa --start-datetime $dat4 --end-
datetime $dat3 --variable number_of_observations --variable eastward_wind --variable
northward_wind -o . -f in_nc/global_scatterro.nc
```

Step2: Then filtering to Olamur sites as for other variables above.

1.4. Data processing of Copernicus Marine Service data, Satellite SST and sea ice

Download and processing of satellite observed SST and sea ice. All Copernicus Marine Service data are downloaded and processed once a day.

Step1: Data download from Copernicus Marine Service



1.4.1. Satellite SST

```
copernicusmarine subset --force-download --overwrite-metadata-cache --dataset-id DMI-BALTIC-SST-L3S-NRT-OBS_FULL_TIME_SERIE $balsstn --start-datetime $dat4 --end-datetime $dat1s --variable sea_surface_temperature --variable mask --variable sea_ice_fraction -o . -f in_nc/bal_sat.nc
```

1.4.2. Ice model data

```
copernicusmarine subset --force-download --overwrite-metadata-cache --dataset-id cmems_mod_bal_phy_anfc_P1D-m $balsstn --start-datetime $dat1 --end-datetime $dat3 --variable siconc --variable sithick -o . -f in_nc/bal_ice.nc
```

Step2: Then filtering to Olamur sites as for other variables above.

1.5. Data processing of Copernicus Marine Service data, CHL-a satellite observations and forecasts

1.5.1. Download and processing of satellite observed CHL-a

Step1: Data download from Copernicus Marine Service

```
copernicusmarine subset --force-download --overwrite-metadata-cache --dataset-id cmems_obs_oc_bal_bgc_tur-spm-chl_nrt_l3-hr-mosaic_P1D-m $balsstn --start-datetime $dat4 --end-datetime $dat1s --variable CHL -o . -f in_nc/bal_chl.nc
```

Step2: Then filtering to Olamur sites as for other variables above.

1.5.2. Download and processing of CHL-a modelled forecasts

Step1: Data download from Copernicus Marine Service

```
copernicusmarine subset --force-download --overwrite-metadata-cache --dataset-id cmems_mod_bal_bgc_anfc_P1D-m $balsstn --start-datetime $dat1 --end-datetime $dat3 --variable chl -o . -f in_nc/bam_chl.nc
```

Step2: Then filtering to OLAMUR sites as for other variables above.

1.6. Putting to FTP

The data files are put to OLAMUR FTP with lftp reverse mirroring command.

Currently the removing of old files is switched off in mirroring command.

