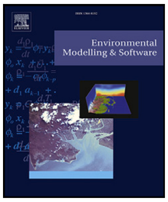
Environmental Modelling and Software 172 (2024) 105901

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



# An Open Science oriented Bayesian interpolation model for marine parameter observations

Gianpaolo Coro

*Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" del Consiglio Nazionale delle Ricerche (ISTI-CNR), Via Moruzzi 1, Pisa, 56124, Italy*

ARTICLE INFO

*MSC:*
68T01
41A05
86A05

*Keywords:*
Geospatial interpolation
Marine science
Advection–diffusion
Markov chain Monte Carlo
Bayesian models
Artificial intelligence

ABSTRACT

Ecological and ecosystem modellers frequently need to interpolate spatiotemporal observations of geophysical and environmental parameters over an analysed area. However, particularly in marine science, modellers with low expertise in oceanography and hydrodynamics can hardly use interpolation methods optimally. This paper introduces an Open Science oriented, open-source, scalable and efficient workflow for 2D marine environmental parameters. It combines a fast, efficient interpolation method with a Bayesian hierarchical model embedding the stationary advection–diffusion equation as a constraint. Our workflow fills the usability gap between interpolation software providers and the users' communities. It can run entirely automatically without requiring expert parametrisation. It is also available on a cloud computing platform, with a Web Processing Service compliant interface, supporting collaboration, repeatability, reproducibility, and provenance tracking. We demonstrate that our workflow produces comparable results to a state-of-the-art model (frequently used in oceanography) in interpolating four environmental parameters at the global scale.

*Software and data availability*

The source code and all experiments' input and output are available on the GitHub at

https://github.com/cybprojects65/BIMACInterpolator

The software was tested with R 4.1.0.

In particular, all used input and output data of BIMAC and DIVA are available as ESRI-GRID data at

https://github.com/cybprojects65/BIMACInterpolator/tree/main/scientific_paper_data

The Web service interface and WPS access point is available on the D4Science e-Infrastructure (https://services.d4science.org/). Free registration is required.

Subscription to the (free-to-use) BiodiversityLab Virtual Research Environment is required to properly size the computational resources to the users' request load (https://services.d4science.org/group/d4science-services-gateway/explore).

After subscription, the BIMAC Web interface will be freely accessible at

https://services.d4science.org/group/biodiversitylab/data-miner?OperatorId=org.gcube.dataanalysis.wps.statisticalmanager.synchserver.mappedclasses.transducerers.BIMAC

No fee is required to use the service.

## 1. Introduction

Observations of geophysical and environmental parameters in a geographic area are available from several prominent collectors of *in situ* or survey data (Argo, 2023; El Serafy, 2020; Zheng et al., 2018; Pouliquen et al., 2012, 2010). Ecological and ecosystem models often require processing these data to produce regular grids of parameter values in space and time over the analysed area (scalar *fields*, in physics terms). Producing a realistic gridded field from non-uniform scattered parameter observations is crucial for environmental and ecological analyses. For example, it allows discovering trends in geophysical and environmental parameters over time (Nishimura et al., 2022; Coro et al., 2020; Evans et al., 2020; Ilinca et al., 2022; Srivastava et al., 2019; Sun et al., 2018; Li and Heap, 2014; Li et al., 2011), estimating environmental suitability to species subsistence (Coro et al., 2022; Pradhan and Setyawan, 2021; Stampoulis et al., 2016; Bregaglio et al., 2011; Hansen and Ines, 2005; Parra et al., 2004), studying complex relations between different parameters (Coro et al., 2023, 2020; Costabile and Macchione, 2015; Capet et al., 2014; Troupin et al., 2010), and understanding ecosystem functions (Paudel et al., 2022; Peters et al., 2019; Willcock et al., 2018; Hunter et al., 2013).

Data gridding is a set of approximate methods to produce a regular gridded field for a parameter $c$ in a spatiotemporal reference coordinate system $x, y, z, t$ over a delimited area $A$. These methods usually start

*E-mail address:* gianpaolo.coro@isti.cnr.it.

https://doi.org/10.1016/j.envsoft.2023.105901
Received 16 March 2023; Received in revised form 17 November 2023; Accepted 21 November 2023
Available online 28 November 2023

from the analysis of the samples of the continuous field (observations) associated with $c(p)$ at positions $p_i = (x_i, y_i, z_i, t_i)$ (with $i = 1, \ldots, n$, and $n$ the number of observations) to infer field values for all positions in the regular grid. Most data gridding techniques assume that observation values are not exact because of intrinsic measurement errors or noise in the data. Therefore, they assume that the $c(p_i)$ samples are approximations of the true values, and thus, that the estimated field does not correspond exactly to the observations. These approaches are named *approximate interpolation* techniques as opposed to *exact interpolation* techniques that assume no error on the data (Lam, 1983). In several approaches, time information is either neglected or treated with a separate interpolation (Troupin, 2023; Beckers et al., 2014; Troupin et al., 2012).

In this paper, we are interested in techniques for 2D ($x$-$y$) approximate interpolation of marine parameters, assuming that 3D interpolations can be approximated by applying data gridding to different horizontal data layers separately, across the $z$ dimension (depth), and eventually stacking the grids. This hypothesis is reasonable for ideal fluids (with low or zero viscosity coefficient and a low frictional force between adjacent layers) and is often adopted for marine water (Troupin et al., 2008).

The motivation that brought us to this study is that large communities of users of high-quality data-gridding algorithms have requirements that still need to be met by modern implementations (Hojati et al., 2022; Coro, 2020b; Assante et al., 2019). Interpolation algorithms with good estimation accuracy usually have low efficiency, and their users are frequently non-oceanographers with limited access to powerful hardware for processing large observation datasets (Section 2.2). Many algorithms require complex parametrisations – to which they are very sensible – that only domain experts can actually handle (e.g., oceanographers and hydrodynamics experts). Therefore, communities of practice of ecological and ecosystem modellers and related-science scholars (e.g., computer scientists working in ecological modelling) need help for using these algorithms correctly. Attempts to automatise these algorithms currently have several limitations and adopt unrealistic assumptions (Section 2.2). These algorithms often require expertise in diverse programming languages – especially to run concurrent executions with different parametrisations – that users can only achieve through long training courses and investments. Moreover, this scenario does not meet the modern requirements of Open Science about the repeatability, reproducibility, re-usability, and openness of Science that multiple scientists working on different domains than oceanography need (Assante et al., 2022; Coro, 2020b; Assante et al., 2019).

This paper introduces an Open Science-oriented, open-source interpolation workflow (the Bayesian Interpolation Model with Advection-diffusion Constraint - **BIMAC**) suitable for processing observations of marine environmental parameters. BIMAC can scale from small areas to the global scale while considering the bulk motion of water and the diffusion of the analysed parameter during the interpolation process. The workflow combines a standard interpolation method based on inverse distance weighting with a Bayesian model that embeds hydrodynamic constraints. The workflow operates a 2-dimensional interpolation at a user-defined water depth while assuming the parameter concentration to be stationary. One main novelty is that it meets crucial requirements by ecological and ecosystem modellers, such as (i) the possibility to automatically infer all workflow input parameters without requiring expert knowledge in oceanography, (ii) the support of Open Science features of repeatability, reproducibility, and re-usability of the process, and (iii) the availability of a free-to-use, standardised, and easily integrable cloud-computing Web service supporting collaborative experimentation. We demonstrate that its results are comparable with those of a state-of-the-art interpolation model in the global-scale interpolation of four global-scale marine parameters commonly used in ecological modelling.

This paper is organised as follows: Section 2, reports an overview of general interpolation methods frequently used for marine environmental parameters and generally describes Bayesian graphical models. Section 3, describes our Bayesian workflow, its Open Science-oriented Web service version, and the evaluation case studies. Section 4, reports an effectiveness and efficiency comparison of our method with a state-of-the-art method on the case study. Finally, Section 5 draws the conclusions.

## 2. Overview of common interpolation methods for marine environmental parameters

A widely employed method for approximating and projecting an interpolated field onto a grid is the Optimal Interpolation (OI) technique (Kaplan et al., 2000; Shen et al., 1998; Bretherton et al., 1976; Gandin, 1963). OI is designed to minimise the expected square prediction error between the estimated (interpolated) field and the actual (unknown) field. The underlying assumption of OI is that the analytical form of the interpolated field can be expressed as a linear combination of the observation data. OI determines the optimal parameters for this linear combination through an analytical approach relying on the inversion of the data covariance matrix. For a comprehensive understanding, detailed mathematical explanations are provided in Appendix. While OI stands out as a robust interpolation technique, its primary drawback lies in its considerable algorithmic complexity, particularly for the inversion of the covariance matrix, leading to reduced efficiency (Section 2.2).

As an alternative to the Optimal Interpolation (OI) method, Kriging (Krige, 1951) is a widely adopted analytical interpolation algorithm. It distinguishes itself from OI by incorporating the distance between observed values into the estimation process of the interpolation linear combination coefficients from the covariance matrix.

Another frequently employed interpolation approach is the Variational Inverse Method (VIM) (Brasseur et al., 1996). Initially developed for climatology analyses, VIM addresses the inefficiency of OI when dealing with a large number of data values, a scenario often encountered in global-scale oceanic *in situ* observations. VIM utilises a finite-element method to minimise the disparity between the estimated and true fields.

A prominent implementation of VIM is the Data-Interpolating Variational Analysis (DIVA) (Beckers et al., 2014; Troupin et al., 2012). Initially designed for interpolating 2D fields and subsequently extended to accommodate 3D and temporal dimensions (Barth et al., 2014), DIVA minimises an error function (variational principle) based on the calculus of variations. This function depends on the distance between observations and the true field across the analysis area, as well as the norm of the field. A more detailed explanation of this powerful methodology is provided in Appendix. The constants involved in the minimisation process encompass (i) a characteristic length ($L$), determining the distance over which a data point influences neighbouring values, (ii) a term penalising solutions that produce values too different from the observations, which is estimated based on the signal-to-noise ratio ($SNR$), and (iii) three terms penalising solutions with significant anomalies, gradients, and variability, respectively. The key parameters for configuring DIVA are $L$ and the $SNR$. DIVA achieves error function minimisation through a finite-element algorithm on a regular grid with predefined extent and spatial resolution. Currently tailored for marine parameters, DIVA defines the estimated field within marine areas and confines it within the coastlines, primarily catering to communities in oceanography. The DIVA software incorporates *Divafit*, a functionality capable of automatically generating estimates for $L$ and $SNR$. However, Divafit operates under the strong assumption of an infinite, isotropic, and homogeneous area, making it beneficial for non-expert users but occasionally yielding sub-optimal results, especially when considering advection–diffusion.

Another frequently employed interpolation method is Inverse Distance Weighted (IDW), a deterministic approach estimating a gridded parameter field through the weighted average of observation values. For each grid point, weights are determined by the distances from observations to that point (Appendix). A weight-decay parameter ($\gamma$) allocates greater influence to observation values closer to the interpolated point, with a large $\gamma$ indicating dependence solely on neighbouring points. Often, $\gamma$ is set to 2 to align with various physics laws and expedite calculations (Lu and Wong, 2008). IDW is commonly used for swift coarse interpolations in spatial autocorrelation analyses due to its lower computational complexity ($O(n)$) compared to other interpolation methods such as OI (Chen, 2021). In geology and ecology data mining, as well as pattern recognition applications, IDW results are frequently deemed acceptable, particularly for small areas (Coro et al., 2023, 2022; Coro and Trumpy, 2020; Neissi et al., 2020; Yang et al., 2020; Srivastava et al., 2019; Santilano et al., 2019; Chowdhury and Maiti, 2016).

In our workflow, we used a modified version of IDW for an initial interpolation stage that later fed a probabilistic graphical model (Section 3.2).

### 2.1. Advection–diffusion constraint

The movement of particles in a fluid in regime conditions follows the *streamlines*, imaginary lines tangent to the fluid velocity. If no sources or sinks are present, the fluid mass in the time unit passing through a closed curve in the streamline is constant, i.e., the fluid mass is *conserved*. Advection is the phenomenon where a quantity $q$, whether dissolved or suspended in a fluid, moves together with the fluid volume. Accompanying advection, there might be *diffusion*, a transport mechanism linked to the gradient of dissolved/suspended quantity concentration. Diffusion involves the transfer of the quantity from regions with higher concentration to those with lower concentration, thereby working to homogenise the concentration value over time. Mathematical details can be found in Appendix.

If a constant diffusion occurs together with advection and the fluid is incompressible (i.e., the velocity has zero divergence and thus does not change in magnitude along the streamlines), and if the quantity is conserved and its concentration does not change over time, then the *steady-state advection–diffusion* equation subsists:

$$D\,\nabla^2 c - \mathrm{v}\cdot\nabla c \;=0$$

where $c$ is the volumetric density of $q$, $\nabla c$ is its gradient, v is the velocity vector field of the fluid, and $D$ is the *diffusion coefficient* that regulates the diffusion speed. In these conditions, the fluid density and velocity can be different from one point to the other but all particles pass through one point with the same velocity. Spatial interpolations frequently reproduce stationary conditions and can incorporate the *steady-state advection–diffusion* equation (*stationary advection–diffusion* equation) as a constraint during the interpolation process. This constraint enhances interpolation accuracy, particularly when the interpolated field represents a quantity subject to advection and diffusion.

DIVA provides users with the option to activate a stationary advection–diffusion equation constraint in the minimisation of the variational principle. This constraint serves to introduce anisotropy into the interpolation, aligning it more closely with fluid dynamics. The square integral of the advection–diffusion term ($D\,\nabla^2\,c - \mathrm{v}\cdot\nabla c$ ) is added to the variational principle, prompting the search for a new minimising field. Implementing this operation necessitates the inclusion of a velocity-field NetCDF file as an additional input from the user. The significance of the advection–diffusion term to the minimisation can be fine-tuned by adjusting a weight parameter. When applying the advection constraint, it is advisable to decrease the automatically estimated characteristic length ($L$) to compensate for the increase of correlation length along the streamlines.

### 2.2. Interpolation methods' drawbacks

The OI method requires the inversion of an $n\times n$ sized data covariance matrix. This operation has an algorithmic complexity of $O(n^3)$ although some efficiency improvements have been proposed (Zhang and Wang, 2010; Hartman and Hössjer, 2008). For this reason, OI might be unsuitable for processing large observation datasets. Moreover, OI is very sensitive to the choice of the correlation function (Gomis et al., 2001). Simple functions depending on point distance (e.g., a Gaussian decay) or isotropic functions are commonly used but are usually lowly suitable for oceanographic applications (Tandeo et al., 2011). Moreover, the model is very sensitive to the variance parameter, whose configuration requires domain expertise.

DIVA requires fewer calculations but also requires sizeable in-memory storage and calculations for large datasets. It is also very sensitive to the characteristic length $L$ and the signal-to-noise ratio parameter values. The Divafit estimates should be revised for applications to small basins and when the advection–diffusion constraint is enabled. Adjustment of these parameters usually requires domain expertise in oceanography and several attempts and tests. Another potential drawback for user communities in ecological and ecosystem modelling is that DIVA should be used through a computational notebook written in the Julia programming language (Barth et al., 2014), which requires attending specific workshops to use the software correctly (SeaDataCloud, 2023b). For some user communities, these notebooks can be executed on cloud infrastructures providing suitable hardware for processing small-medium size datasets (Blue Cloud Consortium, 2023). A fruition-paradigm shift has been recently adopted with respect to a previous distribution of DIVA as a Web service (GHER research group, 2023) endowed with an interface compliant with the Open Geospatial Consortium (OGC) specifications (Pagano and Napolitano, 2016). This paradigm shift has potentially changed the DIVA user community by requiring expertise in computational notebooks in the (still-niche (Tuychiev, 2022)) Julia programming language, and has made batch processing more difficult. The expertise required to the users is uncommon for scientists working on different domains than oceanography (Coro, 2020b).

Usability hindrances for diverse user communities could be softened by introducing new features such as (i) a robust automatic estimation of the input parameters, (ii) the support of big data processing, e.g., through cloud computing, and (iii) the availability of a Web service endowed with a standardised interface, instead of a computational notebook. Moreover, to meet the Open Science requirements of several ecological and ecosystem modelling communities, interpolation software should possibly support process-transparency by enabling result reproducibility, repeatability, computational provenance tracking, and re-usability across multiple application domains. In the next section, we will explain how we added these features to our interpolation method.

### 2.3. Probabilistic graphical models for solving the advection–diffusion equation

A probabilistic graphical model is a graph constituted by nodes representing the random variables of a complex statistical model (Bishop and Nasrabadi, 2006) (Fig. 1). Nodes can represent random variables, deterministic parameters (e.g., analytical functions), and constants. Nodes depending on other nodes have these as *parents*. Therefore, the graph defines the conditional dependencies between the random variables. For example, in Fig. 1 the $\{\theta_1,\theta_2,\theta_3,\theta_4\}$ random variables correspond to the following joint probability distribution:

$$p(\theta_1,\theta_2,\theta_3,\theta_4) = p(\theta_1)p(\theta_2)p(\theta_4|\theta_1,\theta_2,\theta_3)p(\theta_3|\theta_2)$$

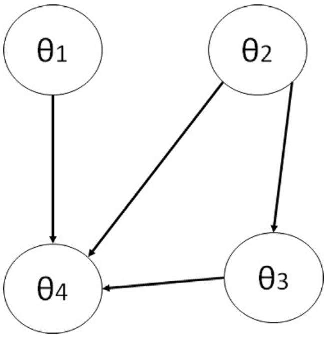


**Fig. 1.** A probabilistic graphical model with four variables.

This formula combines all conditional and prior distributions. If real data were available for $\theta_4$, then $p(\theta_4|\theta_1, \theta_2, \theta_3)$ would be a likelihood function used as a constraint to search for the expected values of the other variables (in compliance with Bayesian approaches). For example, the stationary advection–diffusion term can be embedded into a likelihood function forced to zero (Section 3.3). A Bayesian hierarchical model is a probabilistic graphical model in which the edges have a *causal interpretation*, established by the hierarchical dependencies between the nodes (Clauset et al., 2008). Software for building graphical models usually provides mechanisms to estimate the expected values of random variables by iteratively generating more and more correct samples of the joint probability distribution. One of these techniques is Gibbs sampling (Coro, 2017), which aims at sampling the posterior probability density $p(\bar{\theta}|\bar{y})$ of model parameters $\bar{\theta} = \theta_1, \dots, \theta_m$ given the observation data $\bar{y}$. This technique allows estimating the expected values of the $\bar{\theta}$ parameters based on the samples drawn from the posterior probability. The analytic forms of the prior and likelihood functions should be defined at the configuration time in the graphical model. The posterior probability would then be a multiplication of these functions. Gibbs sampling uses a Markov chain process to draw samples from the conditional distributions of the $\theta_i$ variables given all the other variables (*full conditionals*), which are linked to the posterior probability (Casella and George, 1992; Neal, 1993; Lyle Gurrin and Ekstrom, 2013; Chib, 1995; Resnik and Hardisty, 2010). We report details about this method in Appendix. Gibbs sampling produces a 1st-order Markov chain of samples (with the full conditionals being the transition functions) because, at each step, it estimates new values using the values of the previous iteration. The Markov chain usually begins to converge after generating many samples. Therefore, discarding the first produced samples (*burn-in iterations*) is good practice. The number of burn-in iterations depends on the model convergence speed. If the last samples of a variable are mutually independent, their mean approximates the expected value and their standard deviation measures uncertainty (Monte Carlo Integration (Walsh, 2004; MacKay, 1998)). To break the dependency between the draws in the Markov chain, one draw every $d$ draws should be kept, with $d$ heuristically chosen (*thinning*) (Lyle Gurrin and Ekstrom, 2013; Froese et al., 2014). A computational method that generates a Markov Chain of samples from a posterior distribution to estimate random variables' expected values is named Markov Chain Monte Carlo (MCMC) method. Several software implementations exist (Depaoli et al., 2016; Lunn et al., 2012; Robert and Ntzoufras, 2012), among which the Just Another Gibbs Sampler (JAGS) is one of the most frequently used (Plummer et al., 2003; Froese et al., 2018).

MCMCs are often used to solve the advection–diffusion equation in several application domains. For example, to reconstruct contaminant release history (Zhou and Tartakovsky, 2021), assess groundwater flow and mass transport prediction uncertainty (Fu and Gómez-Hernández, 2009), identify hazardous and polluting gas sources in urban areas (Öttl et al., 2003; Guo et al., 2009), and model erosion and sediment transport (Panday et al., 2014; Edge et al., 2022). The next section will explain how we used an MCMC-based model to interpolate environmental parameter observations, using the advection–diffusion equation as a constraint.

## 3. Methods

This section describes our method (BIMAC) for interpolating a 2D scalar field representing the distribution of a marine environmental parameter in an area. The method can scale from small to large analysis areas. It is also suitable for estimating the transport and diffusion of the parameter within the bulk motion of water. BIMAC is entirely written in R and open-source ("Software and data availability" section). It uses a Bayesian hierarchical model (solved through Gibbs sampling) to re-estimate a prior interpolation result based on IDW while constraining the re-analysis to satisfy the stationary advection–diffusion equation.

Although it has a more limited scope than alternative multi-dimensional interpolation methods (Section 2), it overcomes significant issues of the other solutions (Section 2.2) by (i) requiring a minimal parametrisation, with the possibility of full automatism, (ii) addressing users working in different domains than oceanography (e.g., geology, ecology, climate, etc.) using R libraries or via a Web service and interface, (iii) residing on a long-term sustainable cloud computing environment that supports parallelisation over several time and depth layers and different areas, (iv) complying with the Open Science directives of repeatability, reproducibility, and re-usability of the process through the use of standards to describe the Web service input (Web Processing Service (Schut and Whiteside, 2007)) and keep track of the computational provenance (Prov-O (Lebo et al., 2013)), (v) being enough computationally efficient to manage large datasets with modest hardware.

This section describes all the components of the BIMAC workflow, following the schema reported in Fig. 2, i.e., data provisioning and pre-processing (Section 3.1), prior parameter distribution calculation (Section 3.2), and final interpolated distribution estimation (Section 3.3). Moreover, it describes the Open Science-oriented Web service associated (Section 3.4) and the evaluation case study (Section 3.5).

### 3.1. *Data provisioning and pre-processing*

In the data preparation phase, the user should prepare two raster files containing the horizontal ($u$) and vertical ($v$) components of the water currents' velocity in the study area. Moreover, the user should specify the water depth level of the analysis. The two raster files should define the $u - v$ components of the currents at the specified depth, averaged over a specific time frame. In this time frame the currents will be assumed stationary. The files should be provided in the standard ESRI-GRID (ASCII) format defined by the Open Geospatial Consortium (OGC) (ArcMap, 2023), commonly used by ecosystem and ecological modellers (Coro et al., 2023). The files' spatial resolution will correspond to the interpolated field's grid resolution (*analysis resolution*). Retrieving and preparing these data is usually simple through common GIS software. Velocity data are frequently available on open data infrastructures (e.g., Copernicus (Copernicus, 2020)), as provided by different research groups through oceanographic analyses, forecasts, and re-analyses at detailed resolutions, and from regional to global scales and several temporal aggregations (from hourly to yearly). Pattern recognition experiments working with big data often aim to estimate macro patterns and could indeed prefer coarser time resolutions (Coro et al., 2022; Coro and Bove, 2022; Coro, 2020a; Coro and Trumpy, 2020).

The BIMAC user should also provide a comma-separated-values (CSV) file containing coordinate pairs and observation values for the analysed parameter at the specified input depth. This file should report three data columns: *longitude, latitude, value*. The data could come from

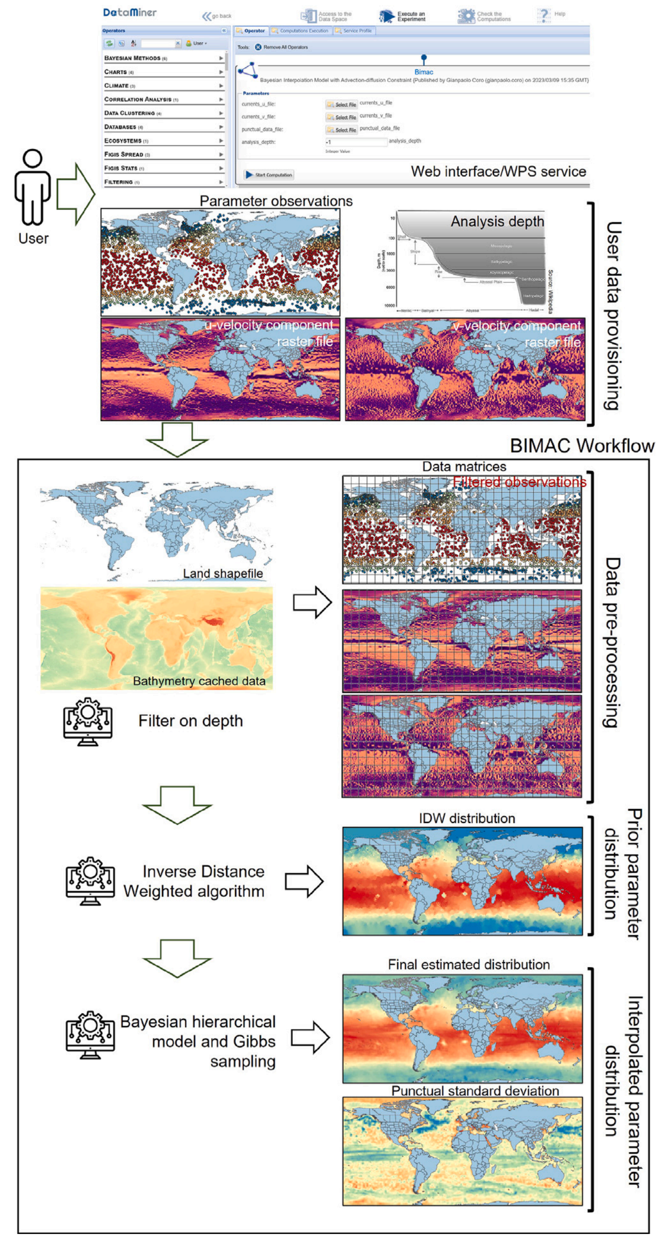


**Fig. 2.** General schema of our workflow.

*in situ* observations such as those of the global Argo network (Coro et al., 2018; Argo, 2023), which are valuable for marine ecological models (Coro et al., 2022).

Behind the scenes, BIMAC embeds a bathymetry raster file and automatically cuts it out on the study area extent. This file and a vector shapefile of global land distribution are used to set the analysis geographic boundaries. The bathymetry data resolution is automatically re-sampled at the *analysis resolution*. A caching mechanism avoids re-sampling the same region and resolution multiple times across different executions on the same machine. BIMAC uses the GEBCO 2022 global-scale dataset with a 0.004° spatial resolution as the default bathymetry file (GEBCO, 2022). This file can easily be substituted with a user-provided file for higher-resolution analyses, e.g., the EMODNET bathymetry dataset with a 0.001° resolution for European sea regions (EMODNET, 2020).

Given the requirements of our system, our principal aim was to obtain a reasonably accurate reconstruction while asking the users to provide only basic input information (i.e., currents' velocity components, depth, and observations). This way, we could offer an advanced interpolation method to large communities of users with low expertise in oceanography and hydrodynamics.

The data preparation algorithm can be summarised as follows (see Moyroud and Portet (2018)):

**Algorithm 1** Data preparation and pre-processing

```
Common preliminary user actions:
    Identify the analysis area extent $A$, $depth$, and time frame $T$. The BIMAC default $depth$ value is 0 (surface)
    Download $u - v$ velocity files for $T$ from an open data infrastructure (e.g., Copernicus)
    (Optional) Download bathymetry information from an open data infrastructure (e.g., EMODNET (EMODNET, 2020)). The BIMAC default bathymetry file is GEBCO 2022 (GEBCO, 2022)
    Cut the files at the same spatial extent and resolution $R$ with a GIS software (e.g., QGIS (Moyroud and Portet, 2018))
    Save the $u$-velocity, $v$-velocity, and (optionally) the bathymetry files as ESRI-GRID (ASCII) files
    Download the punctual, scattered observations of an environmental parameter from an open data infrastructure (e.g., Argo (Argo, 2023))
Workflow start - data preparation:
    Read the $u$-velocity file
    Read the $v$-velocity file
    Read the bathymetry file
    Read the observation points' dataset
    Infer the resolution parameter $R$ and the area extent $A$ from the velocity files
    Create a uniform grid with an $R$ spatial resolution, referring to area $A$, as a *data matrix* to report input and interpolation data
    Project the $u$-velocity onto a data matrix $\rightarrow$ $u$-velocity matrix
    Project the $v$-velocity onto a data matrix $\rightarrow$ $v$-velocity matrix
    Project and cut the bathymetry file onto a data matrix $\rightarrow$ bathymetry matrix
    Exclude observation points falling outside of $A$
    Using the bathymetry matrix and a land shapefile, mark the points falling on land or with bathymetry higher than $depth$ as *data to be excluded* from the subsequent analyses (*excluded points*).
```

The requested input data are similar to those required by the 2D version of DIVA (GHER research group, 2023), which simplifies the adoption of BIMAC by users familiar with this software. The required velocity file format is a simple ASCII format – instead of the NetCDF format used by DIVA – which is more familiar for ecological and ecosystem modellers (Christensen et al., 2005).

### 3.2. Prior parameter distribution

The second step of the workflow focuses on estimating a prior distribution for the reconstructed parameter field. In this phase, BIMAC estimates the parameter values over a 2D projection grid using a similar approach to the IDW interpolation method (Section 2.3). A first IDW-like process is conducted by assigning the weighted sum of the closest observation values to each grid point falling within a resolution-sized circle around the grid point. The weights are the inverse distances of

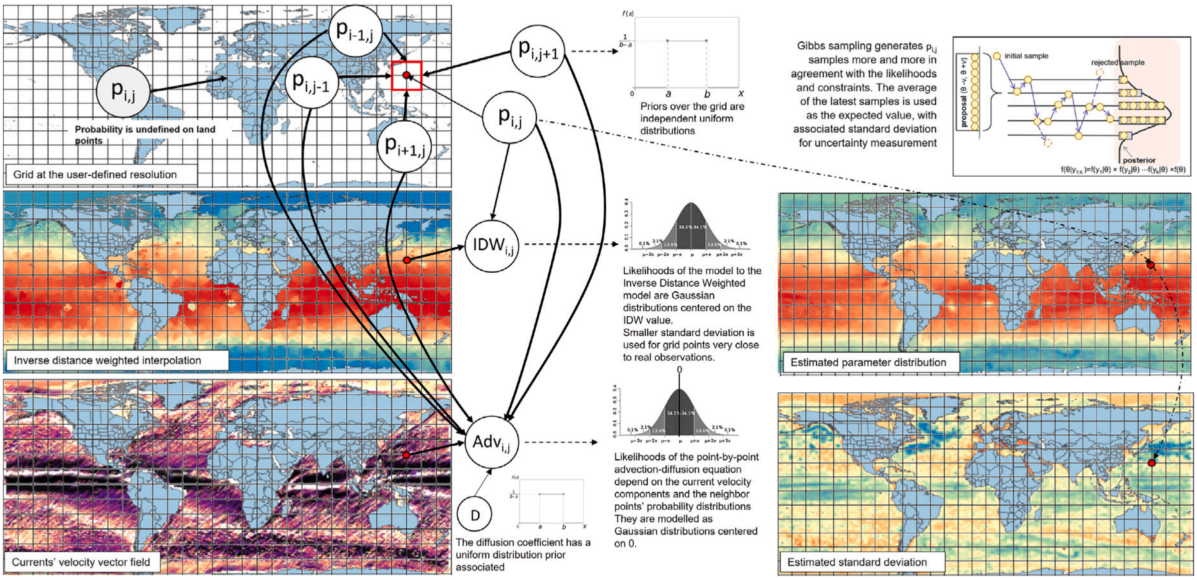

$p_{i,j}$
$p_{i-1,j}$
$p_{i,j-1}$
$p_{i+1,j}$
$p_{i,j+1}$
$p_{i,j}$
Probability is undefined on land points
Grid at the user-defined resolution
f(x)
1/(b−a)
0
a
b
x
Priors over the grid are independent uniform distributions
Gibbs sampling generates $p_{i,j}$ samples more and more in agreement with the likelihoods and constraints. The average of the latest samples is used as the expected value, with associated standard deviation for uncertainty measurement
proposal (θ −ν, θ +ν)
initial sample
rejected sample
posterior
$f(\theta|y_{1:k})=f(y_1|\theta) \times f(y_2|\theta) \cdots f(y_k|\theta) \times f(\theta)$
$IDW_{i,j}$
Inverse distance weighted interpolation
0.1%
2.1%
13.6%
34.1%
34.1%
13.6%
2.1%
0.1%
μ−3σ
μ−2σ
μ−σ
μ
μ+σ
μ+2σ
μ+3σ
Likelihoods of the model to the Inverse Distance Weighted model are Gaussian distributions centered on the IDW value.
Smaller standard deviation is used for grid points very close to real observations.
Estimated parameter distribution
0
$Adv_{i,j}$
D
Currents' velocity vector field
The diffusion coefficient has a uniform distribution prior associated
Likelihoods of the point-by-point advection-diffusion equation depend on the current velocity components and the neighbor points' probability distributions They are modelled as Gaussian distributions centered on 0.
Estimated standard deviation

**Fig. 3.** Conceptualisation of the Bayesian hierarchical model used in our workflow. The small image representing the Gibbs sampling convergence process was taken from Dong et al. (2019).

these neighbour points to the grid point. The grid points associated with weighted observations are used as new observations (grid-snapped observations). This operation merges multiple ($n$) overlapping points at the grid resolution (with $R \ll n$) and thus reduces the computation time of the next steps.

The grid-snapped observations are then interpolated through a second IDW process. IDW has a lower computational complexity than other interpolation algorithms (Section 2.3) but requires indicating a coefficient regulating the decay of the influence of each point-value from the other values. IDW is very sensitive to this parameter and might end in different results at its variation. However, this sensitivity can be tolerated if the IDW distribution is only used to estimate a *prior* (preliminary) interpolation. We modified the IDW process to estimate this parameter fully automatically. Our IDW process first identifies a *proximity radius* within which each real observation might be influenced by the other observations. In BIMAC, this length is statistically estimated based on a heuristic approach we developed by analysing several Argo datasets (Argo, 2023). First, the minimum distance ($d_m$) of each observation to the other observations is calculated. A log-normal distribution likely approximates this distribution because the *in situ* buoys typically present major concentration areas separated by larger zones without buoys. The median of the log-normal distribution is the geometric mean of the $d_m$ values. The upper confidence limit ($d_{m-up}$) can be used as a threshold over which points are too far to be influential. The user can also overwrite this $d_{m-up}$ limit. This technique is similar to other IDW algorithm enhancements (Lu and Wong, 2008) but is particularly suitable for the Argo network data.

The second IDW interpolation is conducted for the grid points surrounded by observations falling within the *proximity radius*. Each grid point is assigned the inverse-distance-weighted average of the surrounding observation values. After this calculation, some grid points (among the *non-excluded*) could remain without an assignment because there were no data around them within the proximity radius. In such a case, the IDW process is re-iterated by including the previously estimated points among the observations. This way, each estimated point value will always depend on the surrounding values within the *proximity radius*, which avoids unreal long-distance relations (Lu and Wong, 2008).

The IDW interpolation algorithm can be summarised as follows:

**Algorithm 2** Prior parameter distribution estimation

```
Create a uniform grid $G_{prior}$ with an $R$ spatial resolution, referring to area $A$, as a matrix for the interpolated data. The spatial points of this grid will be referred to as $p_i$ (with $i$ going from 0 to the unfolded-grid length), and their associated values as $v(p_i)$
For each grid point $p_i$:
    Retrieve the real observations $o_k$ falling within an $R$-distance from $p_i$
    If observations exist → produce an inverse-distance-weighted observation value $v(p_i) = \frac{\sum_k v(o_k)/d(p_i,o_k)}{\sum_k 1/d(p_i,o_k)}$; save $p_i$ among a new observation point collection ($\{o_{snap}\}$)
For each observation point $o_{snap}$
    Calculate the minimum distance $d_m(o_{snap})$ from the other grid-snapped observations
Estimate the upper confidence limit $d_{m-up}$ of a log-normal distribution over $d_m(o_{snap})$
a. For each $G_{prior}$ point $p_i$ (not being among the excluded points of Algorithm 1):
    Retrieve the grid-snapped observation values $v(p_j)$ falling within a $d_{m-up}$ distance
    If values exist:
        Calculate the inverse-distance-weighted value as $v(p_i) = \frac{\sum_{j \neq i} v(p_j)/d(p_i,p_j)}{\sum_{j \neq i} 1/d(p_i,p_j)}$, where $d(p_i,p_j)$ ($\leq d_{m-up}$) is the distance between points $i$ and $j$
Assign the newly calculated $v(p_i)$ values to $G_{prior}$
If points without a value associated still exist → re-iterate the process from point a., including the new $v(p_i)$ among the observations.
```

This algorithm automatically creates a uniform $G_{prior}$ spatial grid containing a first, coarse version of the final interpolated parameter distribution. This grid is used as the data reference in the likelihoods of a subsequent Bayesian model (Section 3.3). This model re-calculates the values by hypothesising a random observation error around the grid-snapped observations and a larger uncertainty on the IDW-interpolated values, and also uses the stationary advection–diffusion equation as a constraint.

### 3.3. Interpolated parameter distribution

As the final processing step, a new uniform spatial grid $G_{post}$ is calculated based on $G_{prior}$. This grid contains, at each point, the expected value of a posterior distribution calculated on the corresponding $G_{prior}$ point. BIMAC produces this distribution through a statistical processing of the IDW values using the stationary advection–diffusion equation as a constraint between the re-estimations. This model is thus a Bayesian model, and was implemented through JAGS (Plummer et al., 2003) as a Bayesian hierarchical model based on MCMC simulation (Section 2.3). The model calculates posterior-distribution samples for each grid point and then averages these samples to find the optimal punctual value expectations and uncertainties.

The model-internal prior distributions' analytical forms were selected by testing different functions in the processing of a *gold* example of Argo temperatures used by DIVA (SeaDataCloud, 2023a), a global-scale dataset of 8531 curated *in situ* observations. This dataset allowed us to fine-tune the prior distributions' forms and parametrisations. Eventually, the following hypotheses were adopted for the Bayesian model (represented in Fig. 3, which includes a small image representing the MCMC convergence process from Dong et al. (2019)):

- The parameter field is undefined on the*excluded* points, and these points are also excluded from the advection–diffusion equation constraint;
- The prior distribution of each grid point is a uniform distribution ranging between the minimum and maximum observed value (i.e., no prior value polarisation was given);
- The likelihood of a grid point value corresponding to an original grid-snapped observation ($o_{snap}$) is a normal distribution centred on the prior IDW value, with a small standard deviation. This constraint drives the final estimation from a uniform distribution towards the (inverse-distance-weighted) real observation values;
- The likelihood of all other point values is a normal distribution centred on the prior IDW value with a standard deviation equal to the overall standard deviation of the grid-snapped observation values. This function drives the final estimation from a uniform distribution towards a value within an uncertainty range around the prior IDW value. The uncertainty range will therefore depend on the area size and the observation values' variance. The final value will depend on the constraints to the other data and the convergence speed of the model;
- For each grid point, the likelihood of the stationary advection–diffusion term value is a normal distribution centred on zero (the expected value) with a small standard deviation. This choice forces each point value to satisfy the equation and interconnects one point to its surrounding points through derivatives and water velocity. The prior distribution of the diffusion coefficient is a uniform distribution between $10^{-10}$ (~ slow diffusion) and $3 \cdot 10^{-9}$ m$^2$ s$^{-1}$ (small, uncharged molecules) which imposes few prior assumptions on the parameter diffusion speed (Multiphysics Cyclopedia, 2017). The user can also overwrite these settings.

The algorithm managing the Bayesian hierarchical model can be summarised as follows (see Geweke (1996)):

**Algorithm 3** Interpolated parameter distribution estimation

```
Configure the JAGS model (priors, likelihoods, advection–diffusion constraint)
Run the Gibbs sampler to generate posterior probability samples for each $G_{post}$ point. Use 1000 iterations for convergence (with 100 burn-in iterations) while retaining one sample every 10 samples to lower their inter-dependency (thinning)
For each $G_{post}$ point
    Average the samples to generate the optimal estimation of the point (according to Monte Carlo Integration (Geweke, 1996))
    Report the standard deviation of the samples as the uncertainty on the value
(Optional) Smooth the $G_{post}$ values using a $3 \times 3$ moving-average matrix.
```

Using the standard deviation of the posterior distribution samples is a reasonable measure of uncertainty because fast-converging distributions are associated with samples closer to accumulation points (the expected values) and usually indicate that the samples are consistent with the priors, likelihoods, and constraints (Coro, 2017). The additional smoothing passage is similar to the *average pooling* made by several machine learning approaches (especially deep-learning models (Coro et al., 2021)) and adds further correlation between adjacent point values.

The $G_{post}$ dataset with associated values and uncertainties is saved as two separate ESRI-GRID (ASCII) raster files reporting the interpolated parameter distribution and uncertainty (as the punctual standard deviation), respectively. Together with another raster file containing the $G_{prior}$ interpolation, these files are the final result of our workflow.

### 3.4. Open science-oriented web service

We developed BIMAC as an open-source R script using JAGS for MCMC modelling (Plummer et al., 2003) ("Software and data availability" section). We published the workflow as a Web service supporting secure cloud and parallel processing and Open Science-oriented features (Hey et al., 2009). In particular, we integrated BIMAC with the DataMiner cloud computing platform of the D4Science e-Infrastructure (Coro et al., 2015; Candela et al., 2016; Coro et al., 2017; Assante et al., 2019), which published the process under the OGC-Web Processing Service standard (WPS (Schut and Whiteside, 2007)) ("Software and data availability" section). This standard interface allows directly integrating the process with widely used geospatial data processing software supporting this standard (e.g., QGIS and ArcGIS) (Coro, 2020b). The DataMiner automatically produces a graphic user interface based on the BIMAC input/output definitions. These features helped meet our crucial requirement that scientists of heterogeneous disciplines and competencies could easily use the software.

The BIMAC Web interface requires uploading two ESRI-GRID (ASCII) raster files onto the platform-integrated distributed storage system (Assante et al., 2019). These files should contain the $u - v$ velocity components of the currents in the analysis area at the depth of the analysis. The geospatial extent and location of these files identifies the analysis area. The online software internally uses the GEBCO 2022 bathymetry file at 0.004° resolution for depth filtering (Section 3.1). The user should also upload a CSV file containing the observations and specify the analysis depth level. The workflow can be executed either through a WPS-HTTP (POST/GET) call (Coro et al., 2017; Schut and Whiteside, 2007) or the online Web interface.

The open-source R software allows customising several data and parameters used by the workflow, e.g., the bathymetry data, the diffusion coefficient, the standard deviation of the advection–diffusion likelihood function, the prior distribution ranges, the error on the observations, the distance upper confidence limit, etc.

Hosting BIMAC on the DataMiner also gives the advantage of distributing the executions on a cloud of 15 machines equipped with Ubuntu 18.04.5 LTS x86 64 operating system, 16 virtual cores, 32 GB of Random Access Memory, and 100 GB of disk for each machine. Each machine can manage up to 4 executions simultaneously (i.e., $15 \times 4 = 60$ concurrent executions). This integration allows to concurrently process the data of different depths and time frames to produce 4D datasets, with each layer being independent of the other. Furthermore, the parameters, input and output data of each execution (computational provenance) are tracked in a user's private data space as XML documents following the Prov-O ontological specifications (Lebo et al., 2013). Provenance tracking is crucial for computational repeatability, reproducibility, and experimental history tracking (Koop et al., 2011; Freire et al., 2012). Through D4Science, the users can also share the computational provenance, compare and merge different results, and conduct experiments together (Assante et al., 2022).

Overall, this integration allowed us to make the workflow compliant with the Open Science features of repeatability, reproducibility, re-usability, collaboration, and interoperability (Hey et al., 2009).

### 3.5. Case study

We compared the results of BIMAC and DIVA at reconstructing surface-level ($depth = 0$) global-scale environmental parameters when used fully automatically. We used a one-month time frame consistent with large-scale climatology models that use DIVA (Troupin et al., 2010). We selected parameter measurements and averaged oceanic-current velocities in January 2018, one of the years with the largest amount of data in the Argo network repository (with 130,906 parameter observations overall). We used a 1° spatial resolution for the global-scale interpolation grid. Therefore, the selected January-2018 velocity components datasets from Copernicus (Copernicus Marine Service, 2018) were re-sampled at this resolution. The projection grid contained 61,200 points.

We interpolated the observations of four global-distribution parameters that are important for marine ecological models (particularly for ecological niche models (Scarcella et al., 2022; Coro et al., 2022)) and widely and frequently measured by the Argo buoys, i.e.:

- Seawater temperature (SST), in °C
- Seawater practical salinity (SAL), in $PSU$
- Mass concentration of chlorophyll-a in seawater (CHL), in $\mathrm{mg/m^3}$
- Moles of oxygen per unit of mass in seawater (DOX), in $\mathrm{\mu mol/kg}$

We downloaded the *delayed-mode* data for these parameters from Argo, which underwent systematic-error adjustments (Argo, 2018). Eventually, the point values retrieved for each parameter were 6,143,302 for SST, 4,697,151 for SAL, 30,190 for CHL, and 231,701 for DOX. We compared the accuracy (correct predictions over total predictions) of BIMAC with that of DIVA at predicting global-scale distributions for these parameters. DIVA was used through a Julia computational notebook. The total computation time from workflow/notebook start to end was also recorded to make an efficiency comparison. Full automation of the DIVA notebook was simulated by automatically estimating the parameters through Divafit (Section 2). Although this strategy produced sub-optimal results, it allowed us to compare DIVA with BIMAC when satisfying the requirement to be fully automatic. For the evaluation, we used 8000 randomly selected point values to interpolate one parameter at a time and the remaining points to test whether their associated values were correctly predicted within uncertainty. The test points were snapped to the interpolation grid through inverse distance weighting, and the fraction of correctly predicted grid points was used as the accuracy measure. This strategy allowed us to stabilise the test point values with respect to the large data variability in one month in each grid cell. The cross-validation process was repeated 20 times, and the average accuracy was reported. To run the tests, we used a D4Science virtual machine equipping Ubuntu 18.04.5 LTS x86 64 operating system, 8 virtual cores (although each run used one core only), and 32 GB of Random Access Memory.

## 4. Results

A quantitative comparison between BIMAC and DIVA showed a generally small accuracy discrepancy ( Table 1). The average absolute value of the discrepancy was 7.43%. On DOX, BIMAC had the largest relative accuracy *loss* with respect to DIVA (−4%), whereas on CHL it had a relative accuracy *gain* of +24%. The BIMAC accuracy loss was −3% on SST and −0.3% on SAL. The CHL observations were mainly concentrated in the southern hemisphere, which likely decreased the DIVA accuracy in the northern hemisphere. This comparison indicates that the two models present complementary aspects, qualitatively visible in Fig. 4. The advection–diffusion component was more accentuated in BIMAC than in DIVA and was likely responsible for increasing the accuracy on CHL prediction through missing-data compensation. However, BIMAC requires the observations to be consistent with the advection–diffusion, which is less probable in a one-month time frame, especially for DOX, because observations can have very high punctual variability in time. The DIVA SST and SAL distributions seemed blurred versions of the BIMAC distributions. Generally, the two models agreed especially when many uniformly-distributed observations were present. This is demonstrated by the lower accuracy discrepancy on SST and SAL. The DOX observations had a generally high punctual variability over time, which created a highly anisotropic distribution. Therefore, there was a less direct relation between the observations and the stationary advection–diffusion equation, which decreased the BIMAC performance.

The similarity between BIMAC and DIVA is particularly visible in Fig. 5, which reports the results on a *gold* example of Argo temperatures used by DIVA that presents low spatial and temporal variability (SeaDataCloud, 2023a) (Fig. 5a). In this case, the IDW distribution (Fig. 5b) was already a good approximation of the final distribution (Fig. 5d). The DIVA distribution was consistent with the areas with a high density of observations (Fig. 5c). However, the higher uncertainty in the Arctic Ocean (Fig. 5e) translated into a too-high average temperature (∼5–6 °C) (Seatemperatu.re, 2023). Instead, the BIMAC iterative approach for building the IDW distribution reconstructed a more likely temperature distribution in the Arctic Ocean (∼1–6 °C). The final BIMAC distribution in this area also presented a high uncertainty (Fig. 5f), as well as in all locations with reduced observation density. However, the Bayesian model produced a realistic lower average temperature (∼1–6 °C) by correctly revising the IDW distribution.

One interesting comparison between BIMAC and DIVA regards the execution times (Table 1 and Fig. 6, with the $X$-axis in logarithmic scale). The average BIMAC time across the case study parameters (180 s) was slightly lower than that of the DIVA notebook (210 s). The DIVA computational time included notebook initialisation (and data pre-processing) and model processing time, which had an equal weight in the total time (∼50%–50%). A detailed analysis of the computation times at the increase of the number of training observations showed almost linear trends in both models, with comparable times. This behaviour comes from the fact that the grid size was constant (1° resolution), whereas data pre-processing had a $O(n)$ computational complexity. It is worth noting that DIVA exhausted the machine memory before 10,000 observations, whereas BIMAC could interpolate even 1M points in 240 s.

**Table 1**
Performance comparison across our case study parameters, i.e., seawater temperature (SST), seawater practical salinity (SAL), mass concentration of chlorophyll-a in sea water (CHL), and moles of oxygen per unit of mass in seawater (DOX). The table reports the total number of observation points per parameter, the amount of data randomly selected for interpolation and testing, and the final grid resolution and size. The accuracy columns compare the effectiveness of our model (BIMAC) vs DIVA. The average computation time indicates the time required by the entire workflow (BIMAC) or notebook (DIVA) to interpolate one parameter.

| | | | | | Accuracy (%) | | Average total workflow/ notebook execution time per case study parameter (s) | |
|---|---|---|---|---|---|---|---|---|
| | No. of observations | No. Of Training observations | No. Of Test observations | Evaluation global-scale grid resolution | BIMAC | DIVA | BIMAC | DIVA |
| SST | 6,143,302 | 8,000 | 6,135,302 | 1° × 1° (61,200 points) | 79.05 | 81.40 | 180 | 210 |
| SAL | 4,697,151 | 8,000 | 4,689,151 | 1° × 1° (61,200 points) | 99.47 | 99.80 | | |
| CHL | 30,190 | 8,000 | 22,190 | 1° × 1° (61,200 points) | 97.20 | 73.52 | | |
| DOX | 231,701 | 8,000 | 223,701 | 1° × 1° (61,200 points) | 78.94 | 82.29 | | |

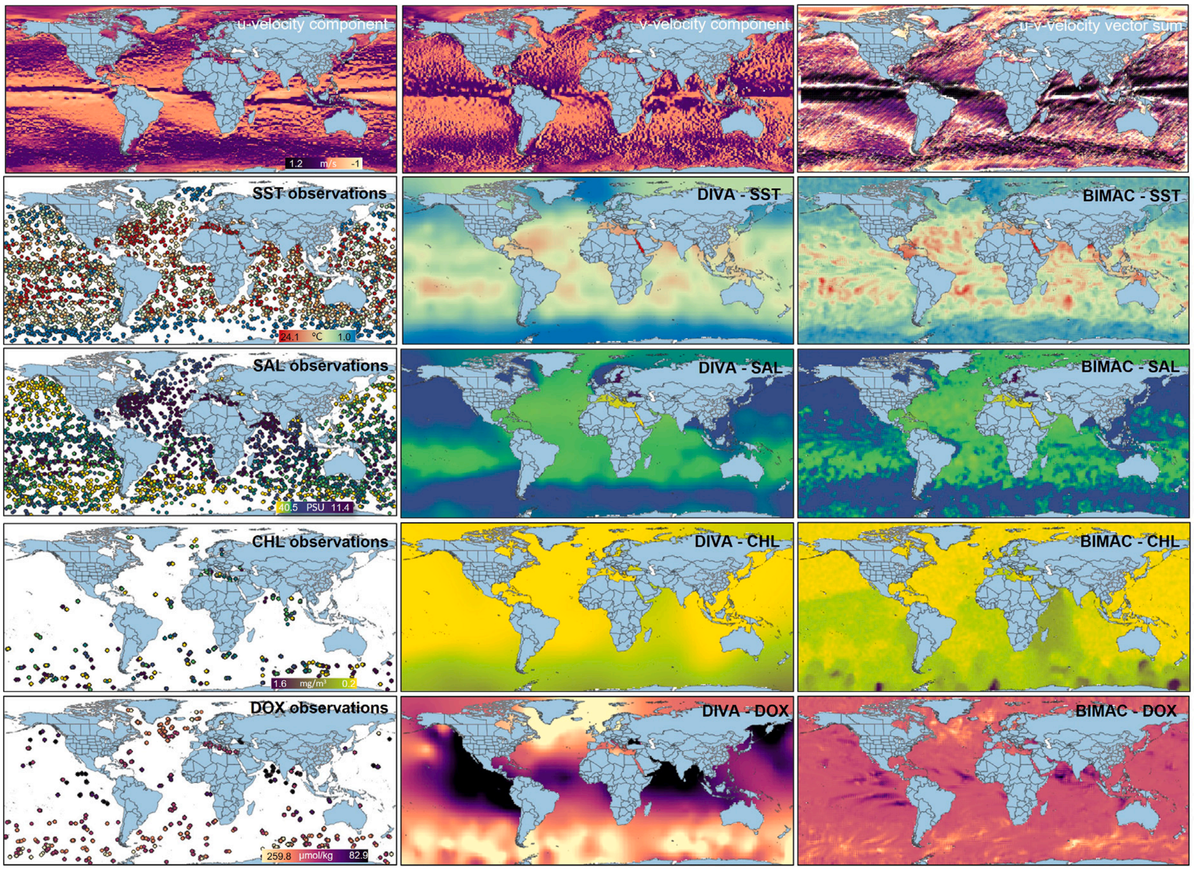


**Fig. 4.** Comparison between the outputs of DIVA and our workflow (BIMAC). The first row reports the horizontal ($u$), vertical ($v$), and vector sum of the oceanic current velocity components averaged over January 2018, as available on Copernicus. The lower images report the point-value distributions and the DIVA and BIMAC outputs produced for our case study parameters, i.e., seawater temperature (SST), seawater practical salinity (SAL), mass concentration of chlorophyll-a in seawater (CHL), and moles of oxygen per unit of mass in seawater (DOX). All distributions in one row share the same legend.

## 5. Conclusions

This paper has presented a workflow (BIMAC) to interpolate marine environmental-parameter observations such as those collected by international programs like the Argo network through fleets of global-scale distributed drifting robotic instruments. The workflow estimates the values of a marine environmental parameter over an area, on a regular grid, by processing punctual, scattered observations of the parameter in the area. As far as the underlying stationarity assumption is satisfied, BIMAC is general enough to work on other aquatic areas than marine areas. Unlike other approaches, BIMAC first conducts a prior interpolation of the values using an automatic, iterative modification of the Inverse Distance Weighted interpolation method. Finally, it re-estimates the values using a Bayesian hierarchical model that combines the prior interpolation with the observation data while using the stationary advection–diffusion equation as a constraint. This way, it models neighbour value inter-relations and admits errors in the observation values.

We demonstrated that on the interpolation of global-scale observations of marine environmental parameters from the Argo network, our results were comparable with those of DIVA (using 2D processing). The prediction accuracy within the uncertainty limits was reasonably high (between ~79% and 99.5%). The processing time was also satisfactory (~180 s) and comparable to that of DIVA. BIMAC processed a million observations in 240 s. The statistical re-analysis of the IDW prior distribution required most of the BIMAC computation time. Overall, we demonstrated that BIMAC is an efficient method. The computational complexity order of the Bayesian re-analysis depends on the grid size $Ng$ and the number of MCMC iterations $Ni$ (i.e., it is $\sim O(NgNi)$). The overall BIMAC computational complexity, including the IDW interpolation, is $O(n + NgNi)$. This complexity can reduce to $O(n)$ when

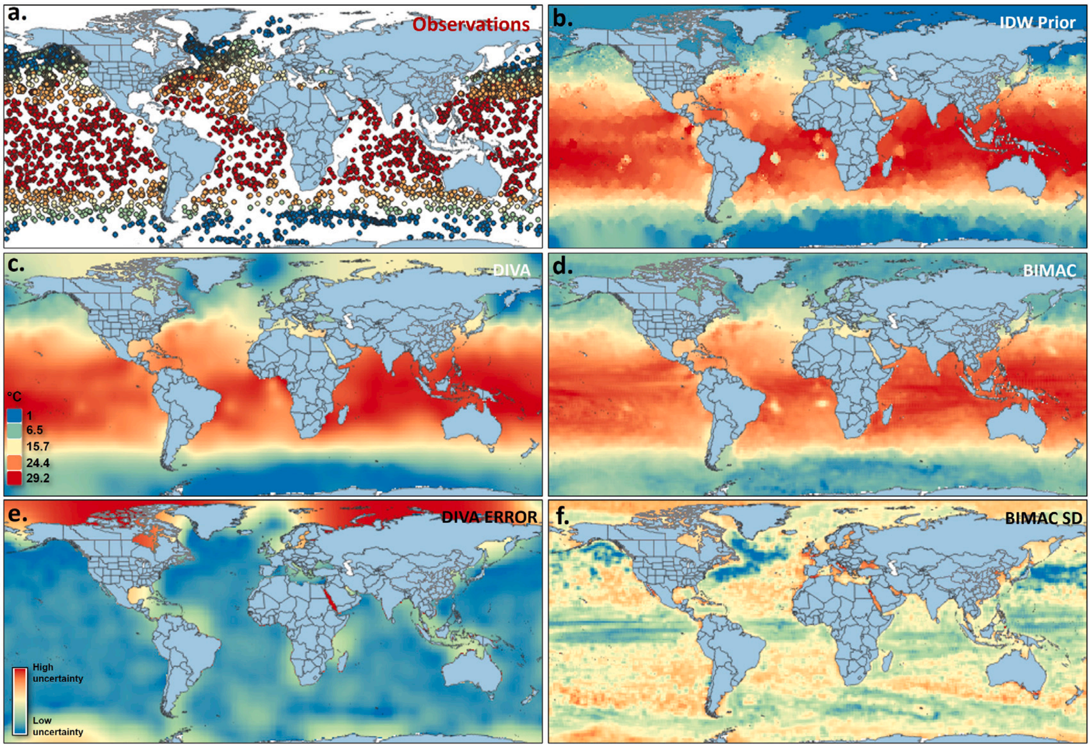


**Fig. 5.** Comparison between the DIVA and our workflow (BIMAC) outputs on a reference data set of global-scale sea-surface temperatures from the Argo network. The charts report (a) the point-values' distribution (observations), (b) the Inverse Distance Weighted interpolation produced by BIMAC as a prior distribution, (c) the DIVA interpolation result, (d) the BIMAC final interpolation, (e) the per-grid-cell error distribution produced by DIVA, (f) the per-grid-cell standard deviation (SD) produced by BIMAC. The temperature colour scale is the same across the temperature points and interpolations. Warmer colours refer to higher uncertainty areas for the error and standard deviation distributions (e and f).

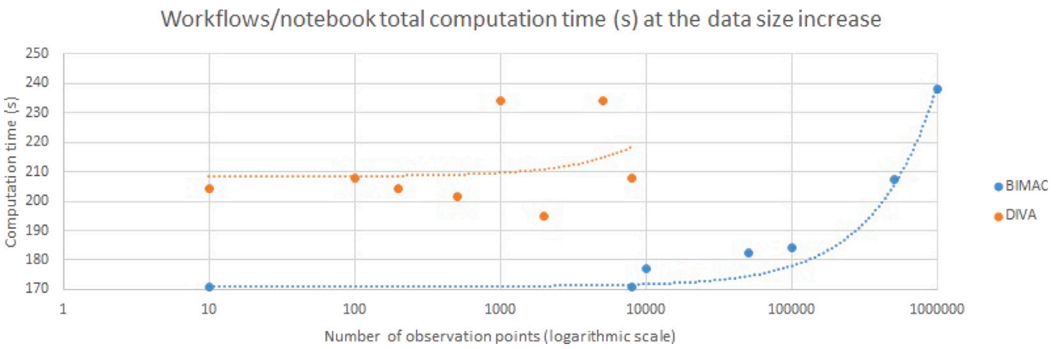


**Fig. 6.** Comparison between the total computation times of our workflow (BIMAC) and DIVA at the increase of the number of observation points analysed. The $X$-axis is reported in the logarithmic scale. The dotted lines are data-trend lines whose exponential-like shapes indicate linear increases. The DIVA performance is reported up to the largest size manageable by the used computing machine.

$n \gg Ng, Ni$ (e.g., in small areas or with less iterations), and is generally lower than the $O(n^3)$ of the OI interpolation method.

One novelty of BIMAC is its different scope with respect to alternative solutions. It addresses the analysts of marine parameters who need to estimate spatial distributions unavailable from large data collectors in specific temporal or spatial frames. These potential users might be ecological and ecosystem modellers needing to extract macro-patterns from estimates of parameter distributions based on local, private observations. These modellers usually have expertise in Open Science-oriented e-infrastructures, the R programming language, ESRI-GRID files, and GIS software. Moreover, they often need to quickly modify code or reduce/enhance features to adapt the models to their cases (Coro et al., 2015; Tsikliras et al., 2023). Since our target was this type of user community, our workflow particularly addressed Open Science features that are crucial in this context. Moreover, we used probabilistic graph models that are more easily interpretable and modifiable than the Artificial Neural Networks used by other solutions and are also frequently used by our target community. BIMAC proposes a fast and fully automatic solution that is also easily modifiable. The software is entirely open source and uses standard R libraries. A free-to-use WPS-based Web service allows easy integration into GIS software supporting this standard. A simple Web interface allows users with no programming skills to execute the interpolations. The used cloud computing infrastructure allows for concurrently processing different depth and time layers and obtaining 3D/4D representations of an environmental parameter distribution. The provenance tracking (standardised in Prov-O) allows sharing, repeating, and reproducing each experiment.

Enhancements of BIMAC will include further refinements of the prior distributions and likelihood functions within the Bayesian hierarchical model, e.g., to possibly find functions suitable for diverse areas and environmental parameters. Tests will also be conducted by substituting the IDW prior estimation with a DIVA interpolation to verify if combining finite-element and Bayesian models can improve the prediction accuracy. Finally, we will explore the possibility of adding 3D/4D processing. Applications of BIMAC are currently being conducted in the context of the EcoScope European project (EcoScope, 2023) in the fields of ecological niche modelling, biodiversity monitoring, and ecosystem modelling. Moreover, BIMAC is used in data gap analyses for the European Commission and the General Fisheries Commission for the Mediterranean (Palermino, 2023; General Fisheries Commission for the Mediterranean, 2023b; European Commission, 2023; General Fisheries Commission for the Mediterranean, 2023a) and within the ITINERIS Italian project (Italian Ministry of University and Research, 2023) in the fields of geology and geothermal energy.

## CRediT authorship contribution statement

**Gianpaolo Coro:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gianpaolo Coro reports financial support was provided by H2020 Food Security Sustainable Agriculture and Forestry Marine Maritime and Inland Water Research and the Bioeconomy. Gianpaolo Coro reports administrative support was provided by Italian PNRR.

## Data availability

All data are openly accessible and are linked in the manuscript Appendix.

## Acknowledgements

This work has been conducted in the context of the EcoScope European Project (H2020-EU.3.2.3.1 - Grant Agreement No. 101000302) and the ITINERIS PNRR Italian project (project code No. IR0000032 - ESFRI Environment). The author also thanks Antonio Palermino and Giuseppe Scarcella (IRBIM-CNR) for using BIMAC in data gap analyses producing new knowledge for the General Fisheries Commission of the Mediterranean and the European Commission as cited in the text.

## Appendix

This Appendix reports mathematical details about the methodologies mentioned in the paper.

### *Interpolation methods for marine environmental parameters*

**Optimal Interpolation (OI) method**: A popular technique for estimating the analytical form of an approximate interpolated field and then projecting it on the interpolation grid (Kaplan et al., 2000; Shen et al., 1998; Bretherton et al., 1976; Gandin, 1963). OI aims at minimising the expected square prediction error between the estimated (interpolated) field $c_e(p)$ and the real (unknown) field $c_r(p)$:

$$e^2(p) = E[(c_e(p) - c_r(p))^2]$$

OI assumes that the analytical form of the interpolated field is a linear combination of the observation data, i.e., of the approximate samples $c_i$ of $c_r(p)$ in the points $p_i$ (with $i = 1, \dots, n$) of a regular grid:

$$c_e(p) = \sum_{i=1}^{n} w_i(p) c_i$$

which leads to

$$e^2(p) = E[(\sum_{i=1}^{n} w_i(p) c_i - c_r(p))^2]$$

OI uses an analytical approach to find the optimal $w_i(p)$ functions that minimise $e^2(p)$. When the sum $\sum_{i=1}^{n} w_i(p) c_i$ is expressed as a vector product $\mathrm{w}^T(p)\mathrm{c}$, the error expression becomes

$$\begin{aligned} e^2(p) &= E[(\sum_{i=1}^{n} w_i(p) c_i - c_r(p))^2] = E[(\mathrm{w}^T(p)\,\mathrm{c} - c_r(p))^2] \\ &= E[c_r(p)^2] + E[\mathrm{w}^T(p)\,\mathrm{c}\,\mathrm{c}^T\,\mathrm{w}(p)] - 2E[c_r(p)\,\mathrm{c}^T\,\mathrm{w}(p)] \end{aligned}$$

The expectation $\mathrm{D} = E[\mathrm{c}\,\mathrm{c}^T]$ is, by definition, the ($n \times n$ sized) data covariance matrix, with $\mathrm{g}(p) = E[c_r(p)\,\mathrm{c}^T]$ being the covariance of the data with the true field. These two matrices (of which $g$ is unknown) allow simplifying the error definition as:

$$\begin{aligned} e^2(p) &= E[c_r(p)^2] + \mathrm{w}^T(p)\,\mathrm{D}\,\mathrm{w}(p) - 2\mathrm{g}^T(p)\,\mathrm{w}(p) \\ &= E[c_r(p)^2] - \mathrm{g}^T(p)\,\mathrm{D}^{-1}\,\mathrm{g}(p) + (\mathrm{w}(p) \\ &- \mathrm{D}^{-1}\,\mathrm{g}(p))^T\,\mathrm{D}\,(\mathrm{w}(p) - \mathrm{D}^{-1}\,\mathrm{g}(p)) \end{aligned}$$

which has its minimum for

$$\mathrm{w}(p) = \mathrm{D}^{-1}\,\mathrm{g}(p)$$

Consequently, the analytical form of the interpolated field becomes:

$$c_e(p) = \sum_{i=1}^{n} w_i(p) c_i = \mathrm{g}(p)^T\,\mathrm{D}^{-1}\,\mathrm{c}$$

The covariance matrix can be used to estimate $\mathrm{g}(p)$ if each $c_i$ is assumed to correspond to $c_r(p_i) + \epsilon_i$, with $\epsilon_i$ being a random observation error uncorrelated with the real field and the other observation errors (Troupin, 2023). This assumption allows expressing the D and $\mathrm{g}(p)$ internal elements in terms of the theoretical data variance and a correlation function $k$ (two user-provided input parameters):

$$\mathrm{D}_{ij} = \sigma^2 k(p_i, p_j) + \epsilon_i \delta_{ij}$$

$$g_i(p) = \sigma^2 k(p, p_i)$$

These values directly allow for calculating the $c_e(p)$ values. OI is a powerful interpolation technique, but its main drawback is its high algorithmic complexity to invert D, i.e., its low efficiency (Section 2.2).

**Kriging** (Krige, 1951): A widely adopted analytical interpolation algorithm, which differs from OI in the fact that the process estimating the linear-combination weights from the covariance matrix also considers the distance between the observed values.

**Variational Inverse Method (VIM)** (Brasseur et al., 1996): Initially conceived for climatology analyses, VIM overcomes the drawback of the low OI efficiency when a large number of data values are available, which might occur for global-scale oceanic *in situ* observations. VIM uses a finite-element method to minimise the difference between the estimated and the true fields. One of the most widely used implementations of VIM is the **Data-Interpolating Variational Analysis (DIVA)** (Beckers et al., 2014; Troupin et al., 2012), originally conceived to interpolate 2D fields but recently extended to manage 3D and time dimensions (Barth et al., 2014). Instead of addressing the minimisation of the expected error, DIVA minimises a function that depends on the calculus of variations (variational principle).

This function depends on the distance between the observations and the true field over the analysis area $A$, and the norm of the field:

$$J(c_r) = \sum_{i=1}^{n} \mu_i\, L^2\, (c_i - c_r(p_i))^2 + \|c_r\|^2$$

with

$$\|c_r\| = \int_A (\alpha_0 L^4\, c_r^2 + \alpha_1 L^2\, \nabla c_r \cdot \nabla c_r + \alpha_2\, \nabla\nabla c_r : \nabla\nabla c_r)\, dA$$

The constants involved can be interpreted as follows: $L$ is a characteristic length that sets the distance over which a data point influences its neighbour values; $\mu_i$ penalises the solutions producing values that are too different from the observations; $\alpha_0$ penalises fields with large anomalies; $\alpha_1$ penalises fields with large gradients; and $\alpha_2$ penalises fields with large variability. The $\alpha$ constants are scaled to depend on $L$. Normally, they are set to $\alpha_0 L^4 = 1$, $\alpha_1 L^2 = 2$, and $\alpha_2 = 1$ (to prefer homogeneous functions). As a further simplification, the penalty $\mu_i$ is defined based on the signal-to-noise ratio ($SNR$): $\mu_i \ L^2 = \mu \ L^2 = 4\pi \ SNR$. Therefore, the principal parameters to configure when using DIVA are $L$ and $SNR$. DIVA finds the $c_e(p)$ field that minimises $J(c_r)$ through a finite-element algorithm on a regular grid of predefined extent and spatial resolution. Currently, DIVA is principally conceived to process marine parameters; thus, the field is defined on marine areas and bounded within the coasts. DIVA computes a triangular mesh within the marine area, whose characteristic length is $L$. The variational principle is solved by each mesh triangle $s$ (=1, … , $N_s$) and then reassembled as $J(c_r) = \sum_{s=1}^{N_s} J_s(c_{rs})$. The triangular solutions are found by introducing a Kernel function $K(p_1, p_2)$ as the data-field covariance, i.e., by defining $g_i(p) = K(p, p_i)$ instead of the corresponding OI term $\sigma^2 k(p, p_i)$ (variance by correlation function). Using this definition, it can be demonstrated (Troupin, 2023) that the minimising field $c_e(p)$ is

$$c_e(p) = \mathrm{g}(p)^T \ \mathrm{D}^{-1} \ \mathrm{c}$$

with

$$\mathrm{D}_{ij} = K(p_i, p_j) + (1/SNR) \ \delta_{ij}$$

$K$ is usually represented as a very spare matrix, which can be computationally and memory demanding for high-resolution interpolation area and large observation datasets. The DIVA software embeds a functionality (Divafit) that can automatically produce estimates for $L$ and $SNR$. Divafit uses an analytical Kernel function (based on the Bessel function) that minimises the variational principle with the (strong) assumption that the area is infinite, isotropic, and homogeneous. Therefore, it is very valuable for non-expert users but frequently produces sub-optimal results, especially if advection–diffusion is considered.

**Inverse Distance Weighted (IDW)**: A deterministic method that estimates a gridded parameter field through the weighted average of the observation values. For each grid point, the weights depend on the distances of the observations from this point. Specifically, the estimated value at each grid point $p_i$ is

$$v(p_i) = \frac{\sum_{j=1}^{n} c_j / d(p_i, c_j)^\gamma}{\sum_{j=1}^{n} 1 / d(p_i, c_j)^\gamma}$$

Where $d(p_i, c_j)$ is the distance of $p_i$ from the point with value $c_j$, and $\gamma$ is a weight-decay parameter that assigns more significant influence to observation values closer to the interpolated point, i.e., a large $\gamma$ indicates dependency only on neighbouring points. Often, $\gamma$ is set equal to 2 to resemble several physics laws and speed-up calculations (Lu and Wong, 2008). This method is often used to obtain fast coarse interpolations for spatial autocorrelation analyses (Chen, 2021) because its computational complexity ($O(n)$) is much lower than the one of other interpolation methods (e.g., OI). In several geology and ecology data mining and pattern recognition applications, especially for small areas, the IDW results are considered acceptable (Coro et al., 2023, 2022; Coro and Trumpy, 2020; Neissi et al., 2020; Yang et al., 2020; Srivastava et al., 2019; Santilano et al., 2019; Chowdhury and Maiti, 2016).

### A.1. Advection–diffusion equation

The movement of particles in a fluid in regime conditions follows the *streamlines*, i.e., imaginary lines within the fluid to which the fluid velocity is always tangent. If no sources or sinks are present, the fluid mass is conserved, i.e., the fluid mass in the time unit passing through a closed curve in the streamline is constant. By defining $c$ as the volumetric density of a quantity $q$ transported by the fluid, the variation of $q$ in the time frame $\mathrm{d}t$ in a volume $V$ within the fluid is

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \int_V \frac{\partial c}{\partial t} \mathrm{d}V$$

If the quantity is conserved, this variation is equal to the total *flux* of the quantity $c$ through a surface $\partial V$ enclosing $V$, i.e.

$$\frac{\mathrm{d}q}{\mathrm{d}t} = -\oint_{\partial V} f \cdot \mathrm{dS}$$

where dS is an elementary surface over $\partial V$, and $f = c\mathrm{v}$, with v being the fluid's velocity vector field. The *flux* is thus a measure of the quantity flow through the surface. According to the divergence theorem:

$$\oint_{\partial V} f \cdot \mathrm{dS} = \int_V \nabla \cdot f \mathrm{d}V$$

hence

$$\frac{\partial c}{\partial t} = -\nabla \cdot f = -\nabla \cdot c \ \mathrm{v}$$

This is the *local* form of the *continuity equation*, i.e., it is associated with the Eulerian description of the transport phenomenon that focuses on one location at a time and observes the transport variations in that position.

*Advection* is the phenomenon for which a quantity $q$ dissolved or suspended in a fluid moves together with the fluid volume. Together with advection, there might be *diffusion*, a transport mechanism associated with a gradient of dissolved/suspended quantity concentration. Diffusion indicates a quantity transfer from points with a higher concentration to points with a lower concentration. Therefore it tends to uniform the concentration value over time.

If diffusion occurs together with advection, the *continuity equation* of a conserved quantity becomes the *advection–diffusion* equation:

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \ \nabla c) - \nabla \cdot c \ \mathrm{v}$$

where $\nabla c$ is the gradient of the volumetric density of $q$, and $D$ is the *diffusion coefficient* that regulates the diffusion speed. In this equation's most common usage scenarios, $D$ is assumed to be constant and the fluid to be incompressible (i.e., the velocity has zero divergence and thus does not change in magnitude along the streamlines). Consequently, the *advection–diffusion* equation becomes:

$$\frac{\partial c}{\partial t} = D \ \nabla^2 \ c - \mathrm{v} \cdot \nabla c$$

In a *steady-state* condition, the quantity concentration does not change over time, thus $\frac{\partial c}{\partial t} = 0$ and the *advection–diffusion* equation becomes:

$$D \ \nabla^2 \ c - \mathrm{v} \cdot \nabla c \ = 0$$

In this case, the fluid density and velocity can be different from one point to the other, but all particles pass through one point with the same velocity.

Spatial interpolations often reproduce stationary conditions and can embed the steady-state advection–diffusion equation as a constraint during the interpolation. This constraint improves the interpolation accuracy if the interpolated field corresponds to a quantity subject to advection and diffusion.

### Probabilistic graphical models and gibbs sampling

**Probabilistic graphical model:** A graph constituted by nodes representing the random variables of a complex statistical model (Bishop and Nasrabadi, 2006) (Fig. 1). The nodes are associated to random variables, deterministic parameters (e.g., analytical functions), and constants. Nodes can depend on other nodes, which will be considered *parent* nodes. Overall, the graph defines the conditional dependencies between the random variables. For convenience, we report again the example of Fig. 1, where the $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ random variables correspond to the following joint probability distribution: $p(\theta_1, \theta_2, \theta_3, \theta_4) = p(\theta_1)p(\theta_2)p(\theta_4|\theta_1, \theta_2, \theta_3)p(\theta_3|\theta_2)$. The joint probability distribution combines all conditional and prior distributions. If real reference-data were available for $\theta_4$, then $p(\theta_4|\theta_1, \theta_2, \theta_3)$ would be a likelihood function used as a constraint to search for the expected values of the other variables (in compliance with Bayesian approaches). A probabilistic graphical model in which the edges have a *causal interpretation* is named Bayesian hierarchical model, which establishes a hierarchical dependency between the nodes (Clauset et al., 2008).

**Gibbs sampling**: Software for building graphical models includes techniques to estimate the expected values of random variables by iteratively generating more and more correct samples of the joint probability distribution. One of these techniques is Gibbs sampling (Coro, 2017). It aims at sampling the posterior probability density $p(\bar{\theta}|\bar{y})$ of the model parameters $\bar{\theta} = \theta_1, \ldots, \theta_m$ given the observation data $\bar{y}$. Eventually, it estimates the expected values of the $\bar{\theta}$ parameters from the samples drawn from the posterior probability. The analytic forms of the prior and likelihood functions should be defined at the configuration time in the graphical model. The posterior probability would then be a multiplication of these functions. Gibbs sampling uses a Markov chain process (Casella and George, 1992; Resnik and Hardisty, 2010) to draw samples from the conditional distributions of the $\theta_i$ variables given all the other variables (*full conditionals*), i.e., $p(\theta_i|\theta_1, \ldots, \theta_i - 1, \theta_i + 1, \ldots, \theta_m, \bar{y})$. The samples from the full conditionals are linked to those of the posterior probability density as follows

$$p(\theta_1, \theta_2, \ldots, \theta_m|\bar{y}) = p(\theta_1|\theta_2, \ldots, \theta_m, \bar{y})p(\theta_2, \ldots, \theta_m|\bar{y})$$

The same rule holds for all variables. The first term on the right-hand side is the full conditional of $\theta_1$. Hence, sampling each full conditional in turn gives values proportional to those of the posterior distribution. Gibbs sampling uses this property by iteratively sampling the full conditional of one variable at a time, leaving the other variables at their preceding sampled values. When a full conditional is sampled this way, a new value for the conditioned variable is picked and then immediately used to sample the other variables that have not been sampled yet. For example, in the case of three variables, the first iteration will use random $\{\theta_1^{(1)}, \theta_2^{(1)}, \theta_3^{(1)}\}$ values; the second iteration will draw one sample $\theta_1^{(2)}$ from $p(\theta_1|\theta_2^{(1)}, \theta_3^{(1)}, \bar{y})$, then a $\theta_2^{(2)}$ sample from $p(\theta_2|\theta_1^{(2)}, \theta_3^{(1)}, \bar{y})$, and so on. A full conditional is usually easier to sample than the complete posterior density because it likely has a well-defined analytical form; otherwise, approximation techniques can be used for sample drawing (Neal, 1993; Lyle Gurrin and Ekstrom, 2013; Chib, 1995).

After $T$ iterations, the sampler will have generated $T$ samples for each variable. It can be demonstrated that the samples produced in the last iterations likely converge to the samples of the posterior probability density (Neal, 1993). The Gibbs sampler produces a 1st-order Markov chain of samples (with the full conditionals being the transition functions) because, at each step, it estimates new values using the values of the previous iteration.

The convergence of the Markov chain typically occurs after generating a substantial number of samples. Therefore, a common practice is to discard the initially produced samples, referred to as *burn-in iterations*. The number of burn-in iterations is contingent on the convergence speed of the model. If the final samples of a variable exhibit mutual independence, their mean serves as an approximation of the expected value, and their standard deviation quantifies uncertainty (Monte Carlo Integration) (Walsh, 2004; MacKay, 1998). To mitigate the dependency between successive draws in the Markov chain, a *thinning* strategy can be employed, where only one draw every $d$ draws is retained, with $d$ being heuristically chosen (Lyle Gurrin and Ekstrom, 2013; Froese et al., 2014). The computational approach that generates a Markov Chain of samples from a posterior distribution to estimate expected values of random variables is known as the Markov Chain Monte Carlo (MCMC) method. Numerous software implementations are available (Depaoli et al., 2016; Lunn et al., 2012; Robert and Ntzoufras, 2012), with Just Another Gibbs Sampler (JAGS) standing out as one of the most frequently utilised options (Plummer et al., 2003; Froese et al., 2018).

## References

ArcMap, 2023. Esri ASCII raster format. Available at https://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/esri-ascii-raster-format.htm.

Argo, 2018. Argo float data and metadata from Global Data Assembly Centre (Argo GDAC) – Snapshot of Argo GDAC of January 2018. Available at https://dataselection.euro-argo.eu/.

Argo, 2023. Argo float data and metadata from Global Data Assembly Centre (Argo GDAC). http://dx.doi.org/10.17882/42182, SEANOE.

Assante, M., Candela, L., Castelli, D., Cirillo, R., Coro, G., Dell'Amico, A., Frosini, L., Lelii, L., Lettere, M., Mangiacrapa, F., et al., 2022. Virtual research environments co-creation: The D4science experience. Concurr. Comput.: Pract. Exper. e6925.

Assante, M., Candela, L., Castelli, D., Cirillo, R., Coro, G., Frosini, L., Lelii, L., Mangiacrapa, F., Pagano, P., Panichi, G., et al., 2019. Enacting open science by D4science. Future Gener. Comput. Syst. 101, 555–563.

Barth, A., Beckers, J.-M., Troupin, C., Alvera-Azcárate, A., Vandenbulcke, L., 2014. Divand-1.0: n-dimensional variational data analysis for ocean observations. Geosci. Model Dev. 7 (1), 225–241.

Beckers, J.-M., Barth, A., Troupin, C., Alvera-Azcárate, A., 2014. Approximate and efficient methods to assess error fields in spatial gridding with data interpolating variational analysis (DIVA). J. Atmos. Ocean. Technol. 31 (2), 515–530.

Bishop, C.M., Nasrabadi, N.M., 2006. Pattern Recognition and Machine Learning, Vol. 1. Springer New York, New York, USA.

Blue Cloud Consortium, 2023. The Blue Cloud European project - e-Infrastructures. Available at https://blue-cloud.org/e-infrastructures.

Brasseur, P., Beckers, J.-M., Brankart, J., Schoenauen, R., 1996. Seasonal temperature and salinity fields in the Mediterranean Sea: Climatological analyses of a historical data set. Deep Sea Res. I 43 (2), 159–192.

Bregaglio, S., Donatelli, M., Confalonieri, R., Acutis, M., Orlandini, S., 2011. Multi metric evaluation of leaf wetness models for large-area application of plant disease models. Agricult. Forest Meteorol. 151 (9), 1163–1172.

Bretherton, F.P., Davis, R.E., Fandry, C., 1976. A technique for objective analysis and design of oceanographic experiments applied to MODE-73. In: Deep Sea Research and Oceanographic Abstracts, Vol. 23. Elsevier, pp. 559–582.

Candela, L., Castelli, D., Coro, G., Pagano, P., Sinibaldi, F., 2016. Species distribution modeling in the cloud. Concurr. Comput.: Pract. Exper. 28 (4), 1056–1079.

Capet, A., Troupin, C., Carstensen, J., Grégoire, M., Beckers, J.-M., 2014. Untangling spatial and temporal trends in the variability of the Black Sea Cold Intermediate Layer and mixed Layer Depth using the DIVA detrending procedure. Ocean Dyn. 64, 315–324.

Casella, G., George, E.I., 1992. Explaining the Gibbs sampler. Amer. Statist. 46 (3), 167–174.

Chen, Y., 2021. An analytical process of spatial autocorrelation functions based on moran's index. PLoS One 16 (4), e0249589.

Chib, S., 1995. Marginal likelihood from the Gibbs output. J. Amer. Statist. Assoc. 90 (432), 1313–1321.

Chowdhury, A., Maiti, S.K., 2016. Assessing the ecological health risk in a conserved mangrove ecosystem due to heavy metal pollution: A case study from sundarbans biosphere reserve, India. Hum. Ecol. Risk Assess.: Int. J. 22 (7), 1519–1541.

Christensen, V., Walters, C.J., Pauly, D., et al., 2005. Ecopath with Ecosim: A User's Guide, Vol. 154. Fisheries Centre, University of British Columbia, Vancouver, p. 31.

Clauset, A., Moore, C., Newman, M.E., 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453 (7191), 98–101.

Copernicus, 2020. Copernicus-Marine environment monitoring service. Available at https://marine.copernicus.eu.

Copernicus Marine Service, 2018. Global ocean ensemble physics reanalysis - Low resolution. Available at https://data.marine.copernicus.eu/product/GLOBAL_REANALYSIS_PHY_001_026/download?dataset=global-reanalysis-phy-001-026-grepv1-uv-monthly.

Coro, G., 2017. Gibbs sampling with JAGS: Behind the scenes. Available at https://www.researchgate.net/publication/313905185_Gibbs_Sampling_with_JAGS_Behind_the_Scenes.

Coro, G., 2020a. A global-scale ecological niche model to predict SARS-CoV-2 coronavirus infection rate. Ecol. Model. 431, 109187.

Coro, G., 2020b. Open science and artificial intelligence supporting blue growth. Environ. Eng. Manag. J. (EEMJ) 19 (10).

Coro, G., Bove, P., 2022. A high-resolution global-scale model for covid-19 infection rate. ACM Trans. Spatial Algorithms Syst. (TSAS) 8 (3), 1–24.

Coro, G., Bove, P., Ellenbroek, A., 2022. Habitat distribution change of commercial species in the Adriatic Sea during the COVID-19 pandemic. Ecol. Inform. 69, 101675. http://dx.doi.org/10.1016/j.ecoinf.2022.101675, URL https://www.sciencedirect.com/science/article/pii/S157495412200125X.

Coro, G., Bove, P., Kesner-Reyes, K., 2023. Global-scale parameters for ecological models. Sci. Data 10 (1), 7.

Coro, G., Candela, L., Pagano, P., Italiano, A., Liccardo, L., 2015. Parallelizing the execution of native data mining algorithms for computational biology. Concurr. Comput.: Pract. Exper. 27 (17), 4630–4644.

Coro, G., Massoli, F.V., Origlia, A., Cutugno, F., 2021. Psycho-acoustics inspired automatic speech recognition. Comput. Electr. Eng. 93, 107238.

Coro, G., Pagano, P., Ellenbroek, A., 2020. Detecting patterns of climate change in long-term forecasts of marine environmental parameters. Int. J. Digit. Earth 13 (5), 567–585.

Coro, G., Panichi, G., Scarponi, P., Pagano, P., 2017. Cloud computing in a distributed e-infrastructure using the web processing service standard. Concurr. Comput.: Pract. Exper. 29 (18), e4219.

Coro, G., Scarponi, P., Pagano, P., 2018. Enhancing ARGO floats data re-usability. Boll. Geofis. Teor. Appl. (Testo stamp.) 59, 53–55.

Coro, G., Trumpy, E., 2020. Predicting geographical suitability of geothermal power plants. J. Clean. Prod. 267, 121874.

Costabile, P., Macchione, F., 2015. Enhancing river model set-up for 2-D dynamic flood modelling. Environ. Model. Softw. 67, 89–107.

Depaoli, S., Clifton, J.P., Cobb, P.R., 2016. Just another Gibbs sampler (JAGS) flexible software for MCMC implementation. J. Educ. Behav. Stat. 41 (6), 628–649.

Dong, T., An, D., Kim, N.H., 2019. Prognostics 102: efficient Bayesian-based prognostics algorithm in Matlab. Fault Detect. Diagn. Progn. 5–25.

EcoScope, 2023. The EcoScope European project. Available online at https://ecoscopium.eu/.

Edge, W.C., Rayson, M.D., Jones, N.L., Ivey, G.N., 2022. In-situ estimation of erosion model parameters using an advection-diffusion model and Bayesian inversion. Authorea Preprints.

El Serafy, G., 2020. Aim, activities and early outcomes of the coastal working group of the european global ocean observing system (EuroGOOS). In: EGU General Assembly Conference Abstracts. p. 22401.

EMODNET, 2020. EMODnet Bathymetry. Available at https://emodnet.ec.europa.eu/en/bathymetry.

European Commission, 2023. STECF - EWG 23-12: Stock assessment in the Adriatic, Ionian and Aegean Sea. https://stecf.jrc.ec.europa.eu/ewg2312.

Evans, S.W., Jones, N.L., Williams, G.P., Ames, D.P., Nelson, E.J., 2020. Groundwater level mapping tool: An open source web application for assessing groundwater sustainability. Environ. Model. Softw. 131, 104782.

Freire, J., Bonnet, P., Shasha, D., 2012. Computational reproducibility: state-of-the-art, challenges, and database research opportunities. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. pp. 593–596.

Froese, R., Coro, G., Kleisner, K., Demirel, N., 2014. Revisiting safe biological limits in fisheries. Fish Fish.

Froese, R., Winker, H., Coro, G., Demirel, N., Tsikliras, A.C., Dimarchopoulou, D., Scarcella, G., Probst, W.N., Dureuil, M., Pauly, D., 2018. A new approach for estimating stock status from length frequency data. ICES J. Mar. Sci. 75 (6), 2004–2015.

Fu, J., Gómez-Hernández, J.J., 2009. Uncertainty assessment and data worth in groundwater flow and mass transport modeling using a blocking Markov chain Monte Carlo method. J. Hydrol. 364 (3–4), 328–341.

Gandin, L.S., 1963. Objective analysis of meteorological fields. Isr. Program Sci. Transl. 242.

GEBCO, 2022. GEBCO gridded bathymetry data. Available at https://www.gebco.net/data_and_products/gridded_bathymetry_data/.

General Fisheries Commission for the Mediterranean, 2023a. Data preparation meetings for hake and sardine in the Adriatic Sea (GSA 17 - 18). https://www.fao.org/gfcm/meetings/info/en/c/1652442/.

General Fisheries Commission for the Mediterranean, 2023b. Report on the data preparation meeting for demersal stocks in the Adriatic Sea. https://www.fao.org/gfcm/technical-meetings/detail/fr/c/1631884/.

Geweke, J., 1996. Monte Carlo simulation and numerical integration. In: Handbook of Computational Economics, Vol. 1. Elsevier, pp. 731–800.

GHER research group, 2023. Diva on Web. Available at https://ec.oceanbrowser.net/emodnet/diva.html.

Gomis, D., Ruiz, S., Pedder, M.A., 2001. Diagnostic analysis of the 3D ageostrophic circulation from a multivariate spatial interpolation of CTD and ADCP data. Deep Sea Res. I 48 (1), 269–295.

Guo, S., Yang, R., Zhang, H., Weng, W., Fan, W., 2009. Source identification for unsteady atmospheric dispersion of hazardous materials using Markov chain Monte Carlo method. Int. J. Heat Mass Transfer 52 (17–18), 3955–3962.

Hansen, J.W., Ines, A.V., 2005. Stochastic disaggregation of monthly rainfall data for crop simulation studies. Agricult. Forest Meteorol. 131 (3–4), 233–246.

Hartman, L., Hössjer, O., 2008. Fast kriging of large data sets with Gaussian Markov random fields. Comput. Statist. Data Anal. 52 (5), 2331–2349.

Hey, A.J., Tansley, S., Tolle, K.M., et al., 2009. The Fourth Paradigm: Data-Intensive Scientific Discovery, Vol. 1. Microsoft research Redmond, WA.

Hojati, M., Robertson, C., Roberts, S., Chaudhuri, C., 2022. GIScience research challenges for realizing discrete global grid systems as a Digital Earth. Big Earth Data 6 (3), 358–379.

Hunter, E.A., Gibbs, J.P., Cayot, L.J., Tapia, W., 2013. Equivalency of Galápagos giant tortoises used as ecological replacement species to restore ecosystem functions. Conserv. Biol. 27 (4), 701–709.

Ilinca, V., Şandric, I., Jurchescu, M., Chiţu, Z., 2022. Identifying the role of structural and lithological control of landslides using TOBIA and weight of evidence: case studies from Romania. Landslides 19 (9), 2117–2134.

Italian Ministry of University and Research, 2023. The ITINERIS PNRR Project. Available online at https://www.itineris.cnr.it/.

Kaplan, A., Kushnir, Y., Cane, M.A., 2000. Reduced space optimal interpolation of historical marine sea level pressure: 1854–1992. J. Clim. 13 (16), 2987–3002.

Koop, D., Santos, E., Mates, P., Vo, H.T., Bonnet, P., Bauer, B., Surer, B., Troyer, M., Williams, D.N., Tohline, J.E., et al., 2011. A provenance-based infrastructure to support the life cycle of executable papers. Procedia Comput. Sci. 4, 648–657.

Krige, D.G., 1951. A statistical approach to some basic mine valuation problems on the witwatersrand. J. South. Afr. Inst. Min. Met. 52 (6), 119–139.

Lam, N.S.-N., 1983. Spatial interpolation methods: a review. Am. Cartogr. 10 (2), 129–150.

Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., Zhao, J., 2013. Prov-o: The prov ontology. W3C Recomm. 30.

Li, J., Heap, A.D., 2014. Spatial interpolation methods applied in the environmental sciences: A review. Environ. Model. Softw. 53, 173–189.

Li, J., Heap, A.D., Potter, A., Daniell, J.J., 2011. Application of machine learning methods to spatial interpolation of environmental variables. Environ. Model. Softw. 26 (12), 1647–1659.

Lu, G.Y., Wong, D.W., 2008. An adaptive inverse-distance weighting spatial interpolation technique. Comput. Geosci. 34 (9), 1044–1055.

Lunn, D., Jackson, C., Best, N., Thomas, A., Spiegelhalter, D., 2012. The BUGS Book: A Practical Introduction to Bayesian Analysis. CRC Press.

Lyle Gurrin, S.H., Ekstrom, C., 2013. Practical data analysis with JAGS using R. http://bendixcarstensen.com/Bayes/Cph-2012/pracs.pdf.

MacKay, D.J., 1998. Introduction to Monte Carlo methods. In: Learning in Graphical Models. Springer, Berlin, Germany, pp. 175–204.

Moyroud, N., Portet, F., 2018. Introduction to QGIS. QGIS Generic Tools 1, 1–17.

Multiphysics Cyclopedia, 2017. Diffusion coefficient. Available at https://www.comsol.it/multiphysics/diffusion-coefficient?parent=diffusion-0402-392-422.

Neal, R.M., 1993. Probabilistic inference using Markov chain Monte Carlo methods. https://www.cs.princeton.edu/courses/archive/fall07/cos597C/readings/Neal1993.pdf.

Neissi, L., Golabi, M., Gorman, J., 2020. Spatial interpolation of sodium absorption ratio: A study combining a decision tree model and GIS. Ecol. Indic. 117, 106611.

Nishimura, R., Jones, N.L., Williams, G.P., Ames, D.P., Mamane, B., Begou, J., 2022. Methods for characterizing groundwater resources with sparse in situ data. Hydrology 9 (8), 134.

Öttl, D., Almbauer, R., Sturm, P.-J., Pretterhofer, G., 2003. Dispersion modelling of air pollution caused by road traffic using a Markov chain–Monte Carlo model. Stoch. Environ. Res. Risk Assess. 17, 58–75.

Pagano, P., Napolitano, U., 2016. Bridging environmental data providers and Sea-DataNet DIVA service within a collaborative and distributed e-infrastructure. Boll. Geofis. 23.

Palermino, A., 2023. Survey gaps software. https://github.com/CNRFisheries/Survey_gap.

Panday, P.K., Williams, C.A., Frey, K.E., Brown, M.E., 2014. Application and evaluation of a snowmelt runoff model in the Tamor River basin, Eastern Himalaya using a Markov chain Monte Carlo (MCMC) data assimilation approach. Hydrol. Process. 28 (21), 5337–5353.

Parra, J.L., Graham, C.C., Freile, J.F., 2004. Evaluating alternative data sets for ecological niche models of birds in the andes. Ecography 27 (3), 350–360.

Paudel, D., Boogaard, H., de Wit, A., van der Velde, M., Claverie, M., Nisini, L., Janssen, S., Osinga, S., Athanasiadis, I.N., 2022. Machine learning for regional crop yield forecasting in europe. Field Crops Res. 276, 108377.

Peters, M.K., Hemp, A., Appelhans, T., Becker, J.N., Behler, C., Classen, A., Detsch, F., Ensslin, A., Ferger, S.W., Frederiksen, S.B., et al., 2019. Climate–land-use interactions shape tropical mountain biodiversity and ecosystem functions. Nature 568 (7750), 88–92.

Plummer, M., et al., 2003. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vol. 124. Vienna, Austria, pp. 1–10.

Pouliquen, S., Carval, T., Loubrieu, T., von Schuckmann, K., Wehde, H., Sjur-Ringheim, L., Hammarklint, T., Harman, A., Soetje, K., Gies, T., et al., 2012. Real Time In Situ data management system for EuroGOOS: A ROOSes-MyOcean joint effort. In: EGU General Assembly Conference Abstracts. p. 10061.

Pouliquen, S., Carval, T., Loubrieu, T., et al., 2010. Real time in-situ data management system for EuroGOOS: A ROOSs–MyOcean joint effort. Sustain. Oper. Oceanogr. 197.

Pradhan, P., Setyawan, A.D., 2021. Filtering multi-collinear predictor variables from multi-resolution rasters of WorldClim 2.1 for Ecological Niche Modeling in Indonesian context. Asian J. Forestry 5 (2).

Resnik, P., Hardisty, E., 2010. Gibbs Sampling for the Uninitiated. Tech. Rep., DTIC Document.

Robert, C., Ntzoufras, I., 2012. Bayesian Modeling using WinBUGS. Taylor & Francis.

Santilano, A., Trumpy, E., Gola, G., Donato, A., Scrocca, D., Ferrarini, F., Brozzetti, F., de Nardis, R., Lavecchia, G., Manzella, A., 2019. A methodology for assessing the favourability of geopressured-geothermal systems in sedimentary basin plays: A case study in Abruzzo (Italy). Geofluids 2019.

Scarcella, G., Angelini, S., Armelloni, E.N., Costantini, I., De Felice, A., Guicciardi, S., Leonori, I., Masnadi, F., Scanu, M., Coro, G., 2022. The potential effects of covid-19 lockdown and the following restrictions on the status of eight target stocks in the adriatic sea. Front. Mar. Sci. 1963.

Schut, P., Whiteside, A., 2007. OpenGIS Web Processing Service. OGC project document http://www.opengeospatial.org/standards/wps.

SeaDataCloud, 2023a. Diva reference and demonstrative dataset of temperatures from the Argo network. Available at https://ec.oceanbrowser.net/emodnet/Data/temperature_argo.txt.

SeaDataCloud, 2023b. Diva workshops and training. Available at https://github.com/gher-uliege/Diva-Workshops.

Seatemperatu.re, 2023. Temperature of the water of the Arctic Ocean. Available online at https://www.seatemperatu.re/seas-and-oceans/arctic-ocean/.

Shen, S.S., Smith, T.M., Ropelewski, C.F., Livezey, R.E., 1998. An optimal regional averaging method with error estimates and a test using tropical Pacific SST data. J. Clim. 11 (9), 2340–2350.

Srivastava, P.K., Pandey, P.C., Petropoulos, G.P., Kourgialas, N.N., Pandey, V., Singh, U., 2019. GIS and remote sensing aided information for soil moisture estimation: A comparative study of interpolation techniques. Resources 8 (2), 70.

Stampoulis, D., Andreadis, K.M., Granger, S.L., Fisher, J.B., Turk, F.J., Behrangi, A., Ines, A.V., Das, N.N., 2016. Assessing hydro-ecological vulnerability using microwave radiometric measurements from WindSat. Remote Sens. Environ. 184, 58–72.

Sun, Q., Miao, C., Duan, Q., Ashouri, H., Sorooshian, S., Hsu, K.-L., 2018. A review of global precipitation data sets: Data sources, estimation, and intercomparisons. Rev. Geophys. 56 (1), 79–107.

Tandeo, P., Ailliot, P., Autret, E., 2011. Linear Gaussian state-space model with irregular sampling: application to sea surface temperature. Stoch. Environ. Res. Risk Assess. 25, 793–804.

Troupin, C., 2023. Diva user guide. Available at https://github.com/gher-ulg/Diva-User-Guide/raw/master/DivaUserGuide.pdf.

Troupin, C., Barth, A., Sirjacobs, D., Ouberdous, M., Brankart, J.-M., Brasseur, P., Rixen, M., Alvera-Azcárate, A., Belounis, M., Capet, A., et al., 2012. Generation of analysis and consistent error fields using the Data Interpolating Variational Analysis (DIVA). Ocean Model. 52, 90–101.

Troupin, C., Machín, F., Ouberdous, M., Sirjacobs, D., Barth, A., Beckers, J.-M., 2010. High-resolution climatology of the northeast Atlantic using Data-Interpolating Variational Analysis (Diva). J. Geophys. Res.: Oceans 115 (C8), http://dx.doi.org/10.1029/2009JC005512, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009JC005512, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2009JC005512.

Troupin, C., Ouberdous, M., Machín, F., Rixen, M., Sirjacobs, D., Beckers, J.-M., 2008. Three-dimensional analysis of oceanographic data with the software DIVA. In: EGU General Assembly. pp. 1–2.

Tsikliras, A.C., Coro, G., Daskalov, G., Grémillet, D., Scotti, M., Sylaios, G., 2023. Editorial: Ecocentric fisheries management in European seas: Data gaps, base models and initial assessments, volume I. Front. Mar. Sci. 10, NA. http://dx.doi.org/10.3389/fmars.2023.1295733, URL https://www.frontiersin.org/articles/10.3389/fmars.2023.1295733.

Tuychiev, B., 2022. The rise of Julia — Is it worth learning in 2022? Available at https://www.datacamp.com/blog/the-rise-of-julia-is-it-worth-learning-in-2022.

Walsh, B., 2004. Markov chain Monte Carlo and Gibbs sampling. http://web.mit.edu/~wingated/www/introductions/mcmc-gibbs-intro.pdf.

Willcock, S., Martínez-López, J., Hooftman, D.A., Bagstad, K.J., Balbi, S., Marzo, A., Prato, C., Sciandrello, S., Signorello, G., Voigt, B., et al., 2018. Machine learning for ecosystem services. Ecosyst. Serv. 33, 165–174.

Yang, W., Zhao, Y., Wang, D., Wu, H., Lin, A., He, L., 2020. Using principal components analysis and IDW interpolation to determine spatial and temporal changes of surface water quality of xin'anjiang river in huangshan, China. Int. J. Environ. Res. Public Heal. 17 (8), 2942.

Zhang, H., Wang, Y., 2010. Kriging and cross-validation for massive spatial data. Environmetrics 21 (3–4), 290–304.

Zheng, F., Tao, R., Maier, H.R., See, L., Savic, D., Zhang, T., Chen, Q., Assumpção, T.H., Yang, P., Heidari, B., et al., 2018. Crowdsourcing methods for data collection in geophysics: State of the art, issues, and future directions. Rev. Geophys. 56 (4), 698–740.

Zhou, Z., Tartakovsky, D.M., 2021. Markov chain Monte Carlo with neural network surrogates: application to contaminant source identification. Stoch. Environ. Res. Risk Assess. 35, 639–651.