

DIVA (Data-Interpolating Variational Analysis) in a virtual research environment

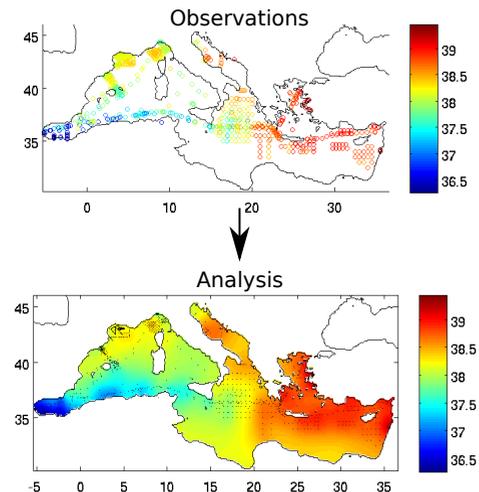
Alexander Barth¹, Charles Troupin¹, Sylvain Watelet¹, and Jean-Marie Beckers¹

¹GHER, University of Liège, Belgium



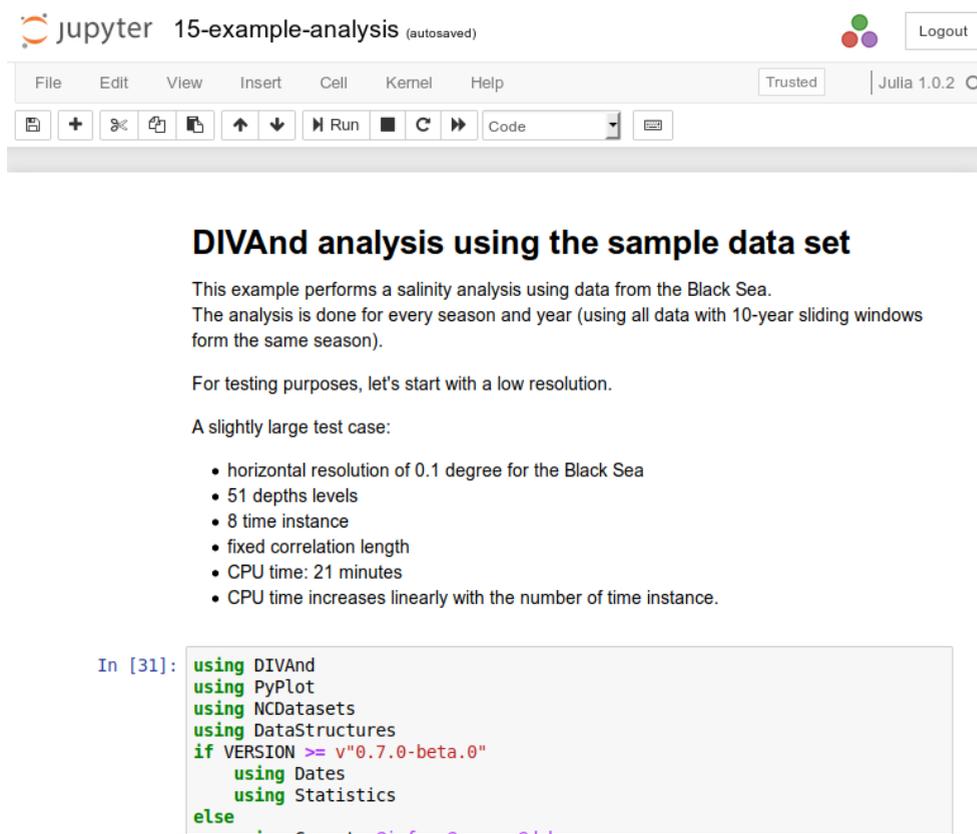
Data-Interpolating Variational Analysis

- DIVA: Data Interpolating Variational Analysis
- Objective: **derive a gridded climatology from in situ observations**
- The variational inverse methods aim to derive a continuous field which is:
 - **close to the observations** (it should not necessarily pass through all observations because observations have errors)
 - **"smooth"**



Jupyter and JupyterHub

- Overview of Jupyter and JupyterHub
- Jupyter is an interactive programming environment
 - **computing**
 - **documentation**
 - **visualization**
- JupyterHub is the multi-user variant for a remote server



jupyter 15-example-analysis (autosaved) Trusted | Julia 1.0.2

File Edit View Insert Cell Kernel Help

Code

DIVAnd analysis using the sample data set

This example performs a salinity analysis using data from the Black Sea. The analysis is done for every season and year (using all data with 10-year sliding windows from the same season).

For testing purposes, let's start with a low resolution.

A slightly large test case:

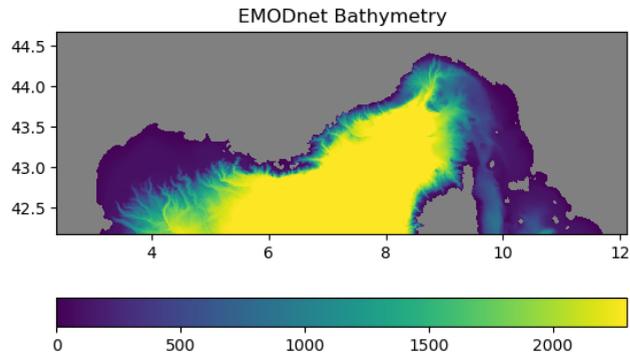
- horizontal resolution of 0.1 degree for the Black Sea
- 51 depths levels
- 8 time instance
- fixed correlation length
- CPU time: 21 minutes
- CPU time increases linearly with the number of time instance.

```
In [31]: using DIVAnd
using PyPlot
using NCDatasets
using DataStructures
if VERSION >= v"0.7.0-beta.0"
    using Dates
    using Statistics
else
```

Bathymetry

- Reading and processing of **bathymetry** from different sources (EMODnet Bathymetry, GEBCO) and preparation of land-sea mask.

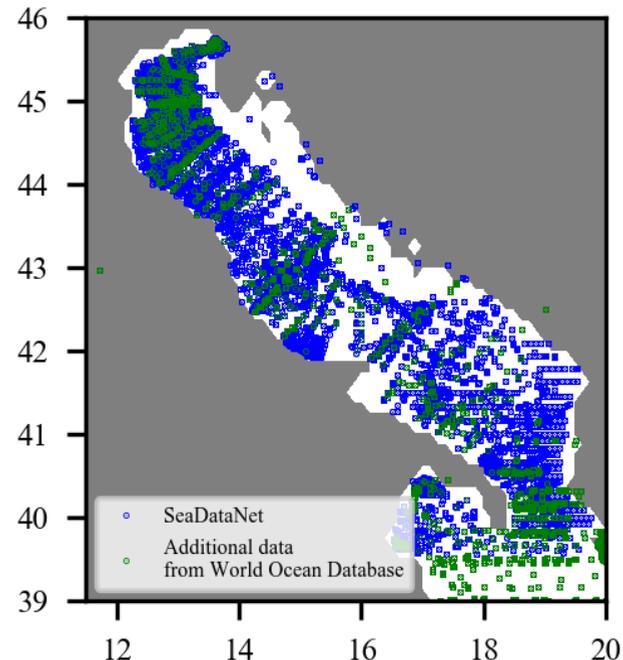
```
colorbar(orientation="horizontal")
contourf(bx[1:r:end],by[1:r:end],copy(b[1:r:end,1:r:end]'), levels = [-1e5,0],colors = [[.5,.5,.5]])
gca()[:set_aspect](1/cos(mean([ylim()...]) * pi/180))
title("EMODnet Bathymetry");
```



Observations

- Reading of different file formats: **netCDF** files and **ODV spreadsheets**
- Download data from external sources such as the **World Ocean Database** or from the **Copernicus Marine Environment Monitoring Service (CMEMS) In-Situ TAC**
- Check for **duplicates** (based on space-time distance)
- Use of the analysis to perform **additional quality checks** on the observations.

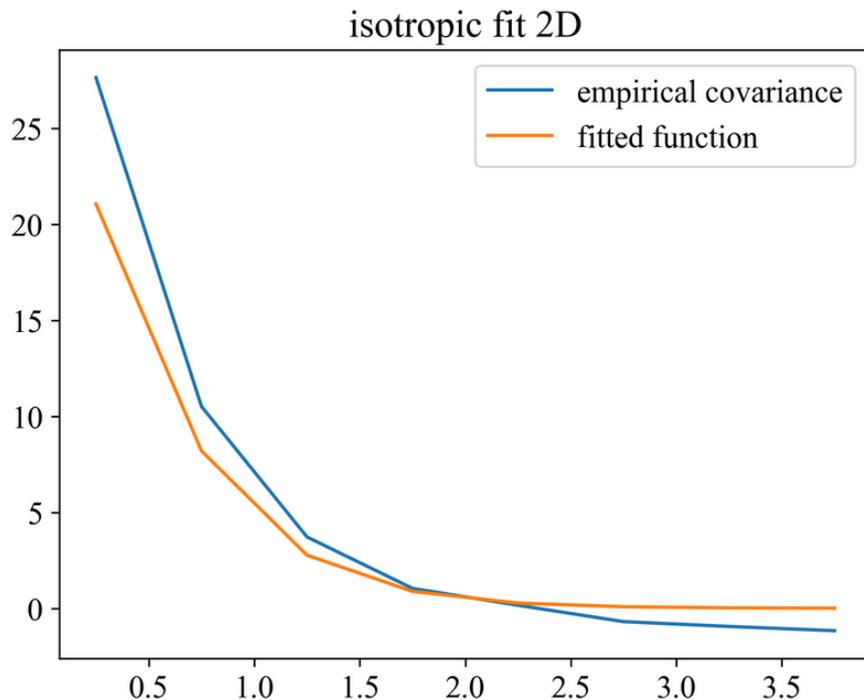
```
In [26]: figure("Adriatic-Additional-Data", figsize=(2,2))
ax = subplot(1,1,1)
ax[:tick_params]("both", labelsz=6)
ylim(39.0, 46.0);
xlim(11.5, 20.0);
contourf(bx, by, permutedims(Float64.(mask_edit[:, :, 1]), [2, 1]),
         levels=[-1e5, 0], cmap="binary");
plot(obslon, obslat, "bo", markersize=.2, label="SeaDataNet")
plot(obslonwod[newpoints], obslatwod[newpoints], "go",
     markersize=.2, label="Additional data\nfrom World Ocean Database")
legend(loc=3, fontsize=4)
gca()[:set_aspect](aspect_ratio)
```



Optimization

- Optimising the analysis parameters (**correlation lengths and noise-to-signal ratio**)
- Derivation of a **relative correlation length** based on the topography gradients.

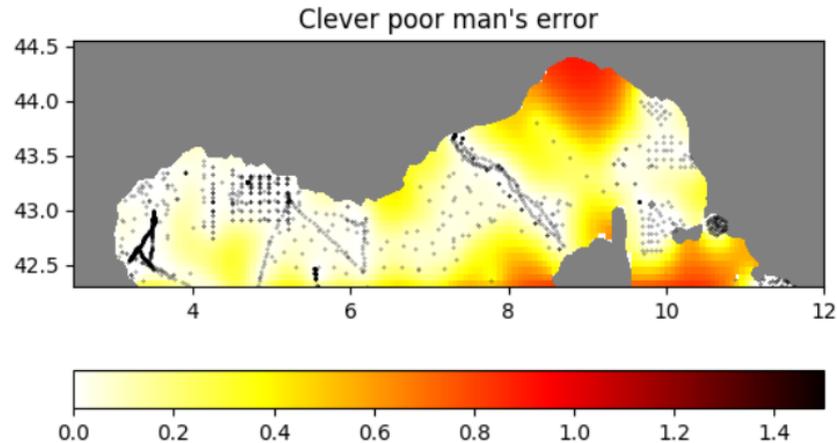
```
In [8]: figure()
plot(distx,covar, label = "empirical covariance")
plot(distx,fitcovar, label = "fitted function")
legend()
title("isotropic fit 2D")
```



Error field and results

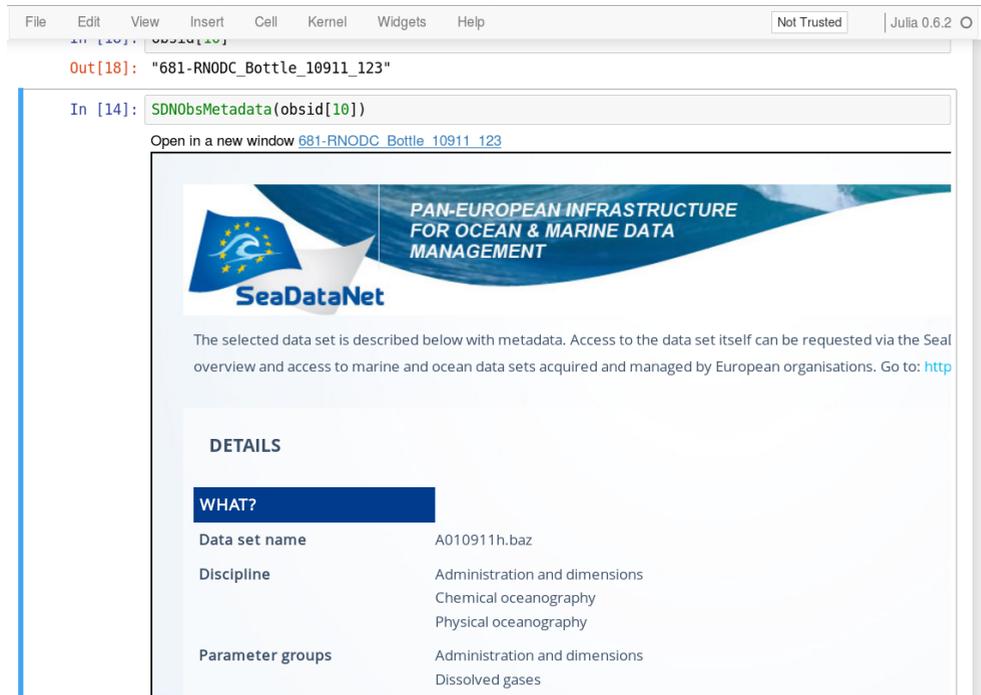
- **Error field** computation
- The error field determines how reliable the gridded field at a given location is
- Different algorithm to compute the error field are available
- Plot results and data.

```
ploterr(cpme)  
title("Clever poor man's error");
```



Metadata

- Query additional metadata for a given observation based on EDMO code and local CDI id from jupyter notebook



The screenshot shows a Jupyter Notebook interface with a Julia kernel. The output of a query is displayed as a web page from SeaDataNet. The page header includes the SeaDataNet logo and the text "PAN-EUROPEAN INFRASTRUCTURE FOR OCEAN & MARINE DATA MANAGEMENT". The main content area is titled "DETAILS" and contains a section "WHAT?" with the following information:

Data set name	A010911h.baz
Discipline	Administration and dimensions Chemical oceanography Physical oceanography
Parameter groups	Administration and dimensions Dissolved gases

REST API

- DIVAnd REST API is also intended to be a multi-user API
- Main functionalities
 - Extraction of a **bathymetry** (GEBCO or possibly EMODnet Bathymetry);
 - Generation of a **DIVA analysis** using data from the user and/or extracted from ODV;
 - Generation of a **XML description** for the Sextant catalogue (in process).
- The essential GET and POST calls will be implemented in a layer around DIVAnd.

annel=fs&q=julia+1.0+run+garbage+collector& 120% ... ☆ julia 1.0 run garbage

URL of the observation to force	
Name of the variable	
Bounding box (east, south, w, degrees)	
Comma separated list of depth levels (meters)	0,20,50
Correlation length in zonal and meridional direction (meters)	100000,100000
Error variance of observation (relative to the error variance of the background field)	1
Resolution in zonal and meridional direction (in degrees)	0.5,0.5
Start and end year	1900,2018
Month of every season	1,2,3 4,5,6 7,8,9 10,11,12
URL of the bathymetry file	sampledata:gebco_30sec

Metadata ▶

Run DIVAnd

[Download results](#)

Conclusions and outlook

- DIVAnd can be used in a jupyterhub-based work-flow
- Container images with all necessary software have been built
- Support of the NetCDF ODV format for more efficient data exchange
- REST API of DIVAnd is currently built
- Ontop of the REST API we are building a web user interface