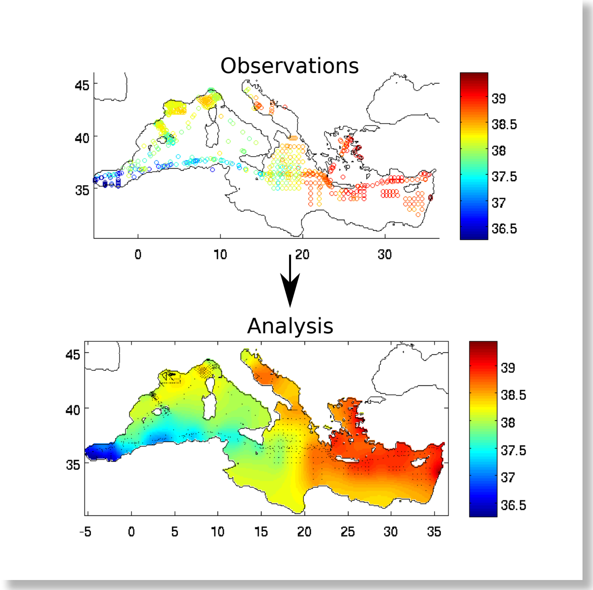
DIVA (Data-Interpolating Variational Analysis) in a virtual research environment

Alexander Barth[1], Charles Troupin[1], Sylvain Watelet[1], and Jean-Marie Beckers[1]

[1]GHER, University of Liège, Belgium

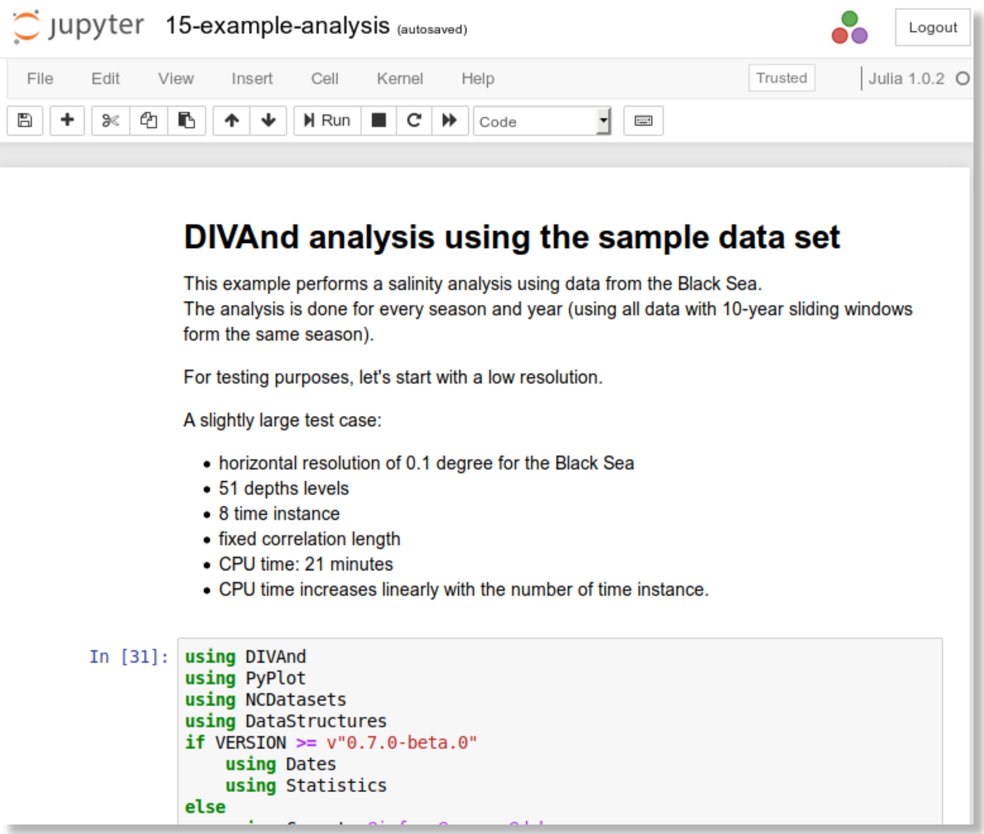
Data-Interpolating Variational Analysis

- 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**"



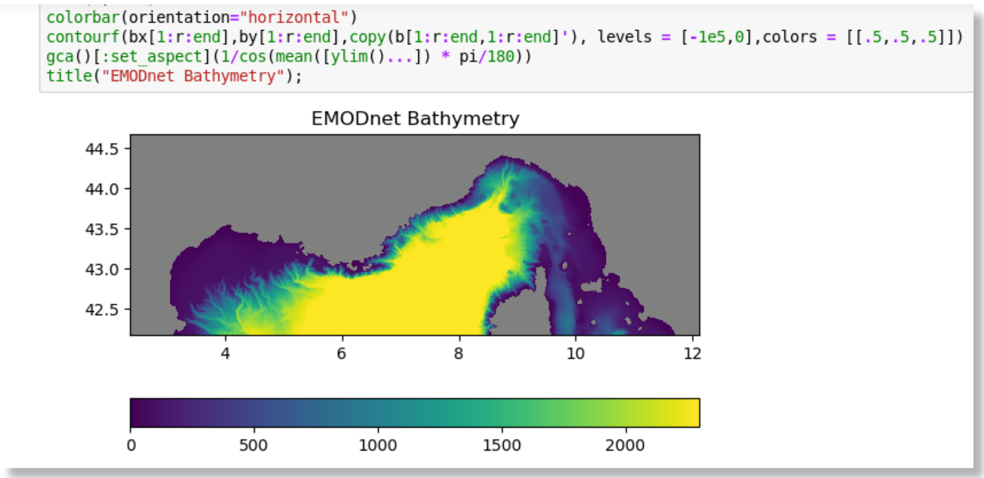
Jupyter and JupyterHub

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



Bathymetry

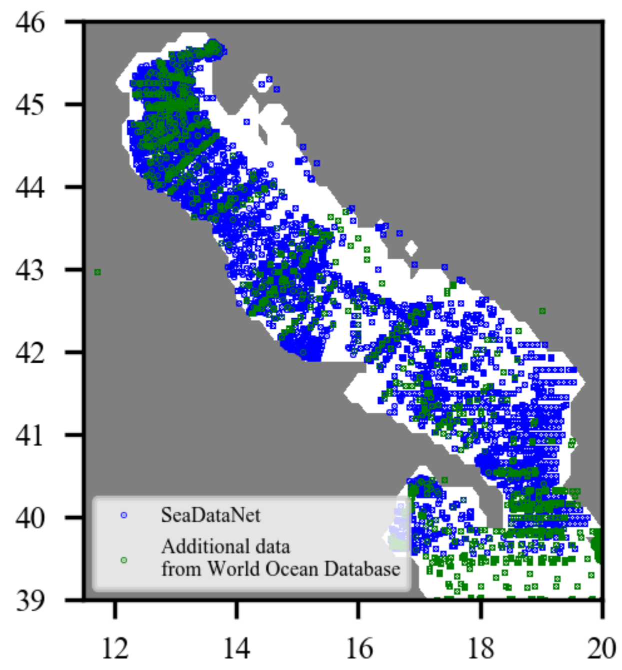
- Reading and processing of **bathymetry** from different sources (EMODnet Bathymetry, GEBCO) and preparation of land-sea mask.



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.

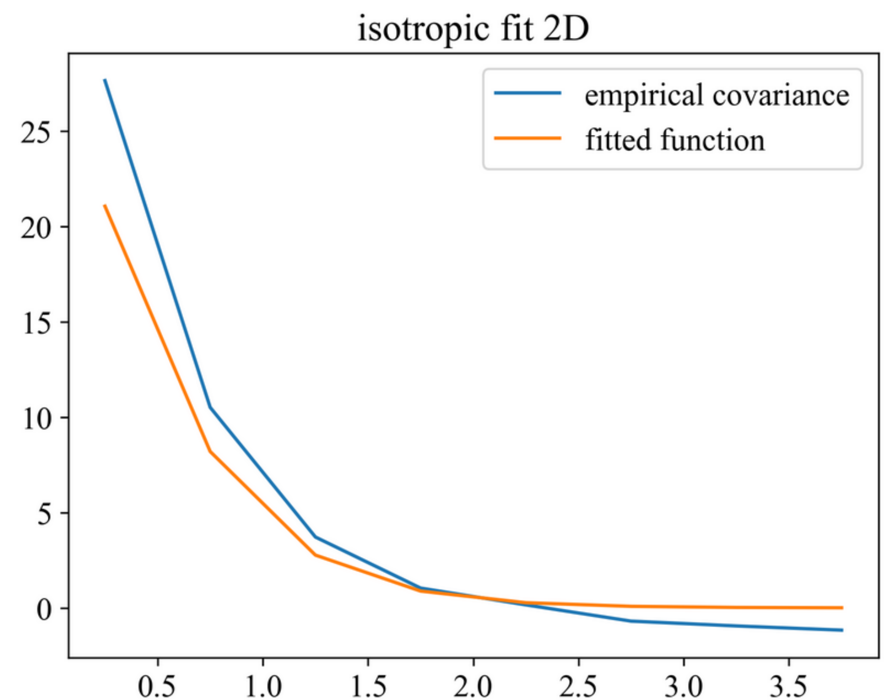
In [26]:
```
figure("Adriatic-Additional-Data", figsize=(2,2))
ax = subplot(1,1,1)
ax[:tick_params]("both",labelsize=6)
ylim(39.0, 46.0);
xlim(11.5, 20.0);
contourf(bx, by, permutedims(Float64.(mask_edit[:,:,1]),[2,1]),
    levels=[-1e5,0],cmap="binary");
plot(obslon, obslat, "bo", markersize=.2, label="SeaDataNet")
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.

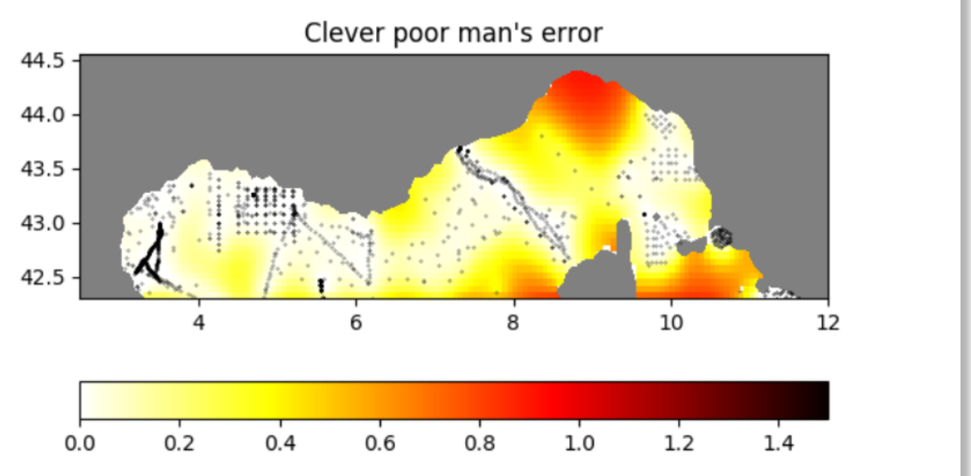
In [8]:
```
figure()
plot(distx,covar, label = "empirical covariance")
plot(distx,fitcovar, label = "fitted function")
legend()
title("isotropic fit 2D")
```



Error field and results

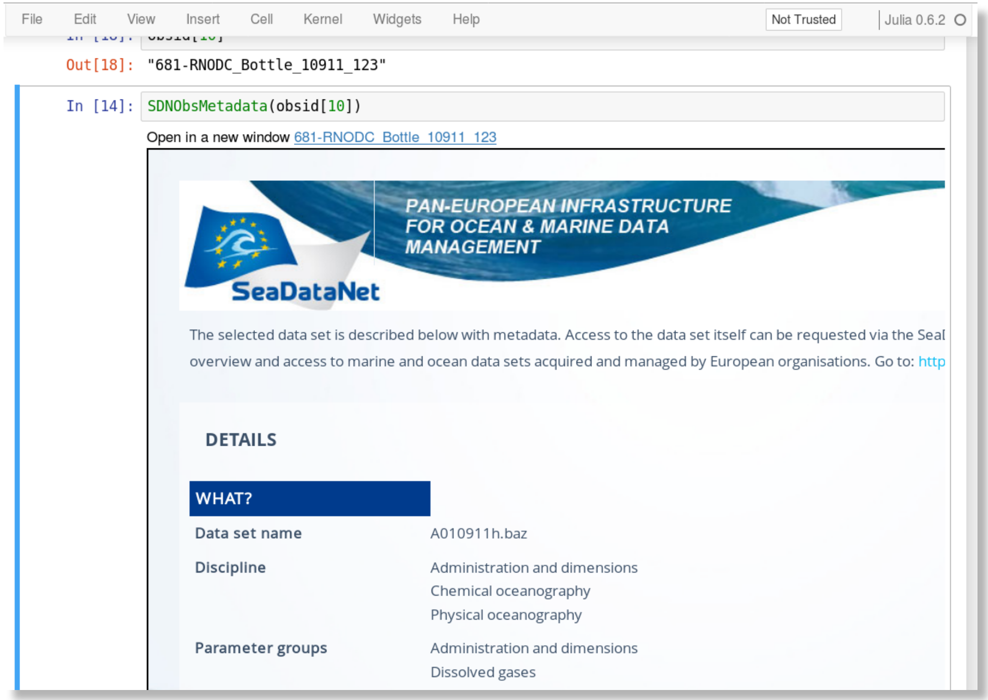
- **Error field** computation
- The error field determines how reliable the gridded field at a given location is
- Different algorithm to compute the error field are available
- Plot results and data.

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



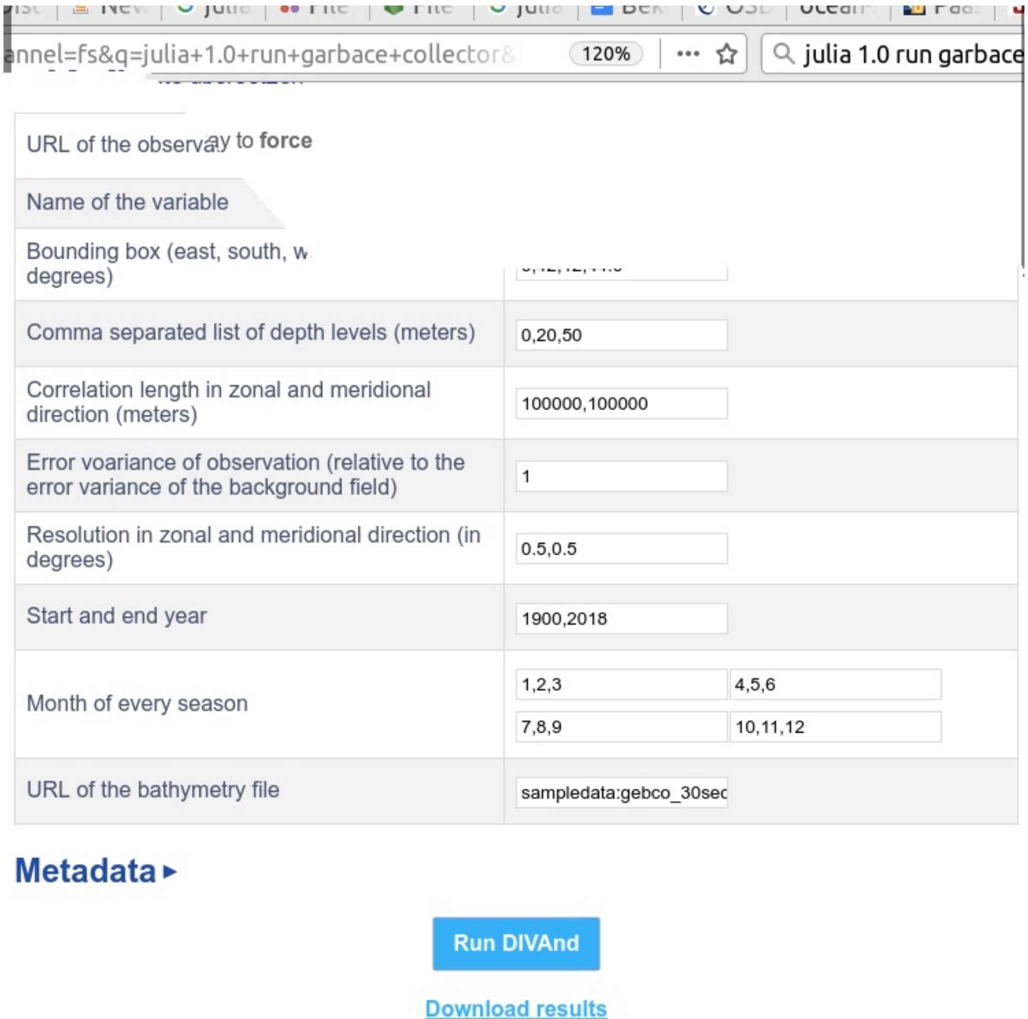
Metadata

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



REST API

- DIVAnd REST API is also intendent 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 implement in a layer around DIVAnd.


nnel=fs&q=julia+1.0+run+garbage+collector&
120%
julia 1.0 run garbace
URL of the observa
ay 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 voariance 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 jupterhub-based work-flow
- Container images will all necessary software have been build
- Support of the NetCDF ODV format for more efficient data exchange
- REST API of DIVAnd is currently build
- Ontop of the REST API we are building a web user interface