Generating ocean climatologies from in situ observations

gher-ulg
@GHER_ULiege
0000-0002-0265-1021

Alexander Barth, Charles Troupin, Sylvain Watelet,
Aida Alvera-Azcárate and Jean-Marie Beckers

Collect once,

Use many times

And create products with DIVA

Conclusions

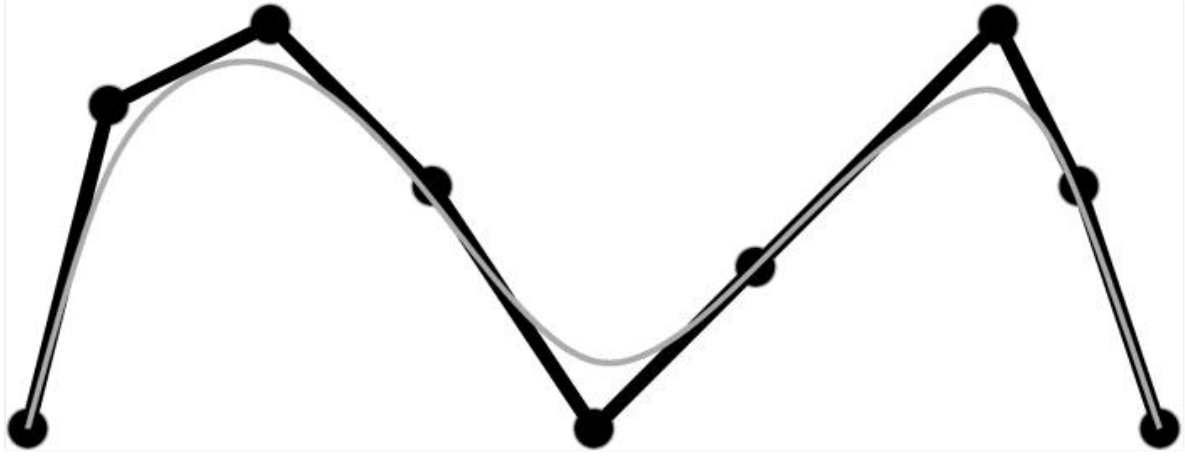
1. `DIVA` is a software tool written in Fortran
2. `DIVAnd` is a software tool written in Julia
3. Both are designed for the spatial interpolation of data

Methodology: spatial interpolation

Gridding problem

Constraints

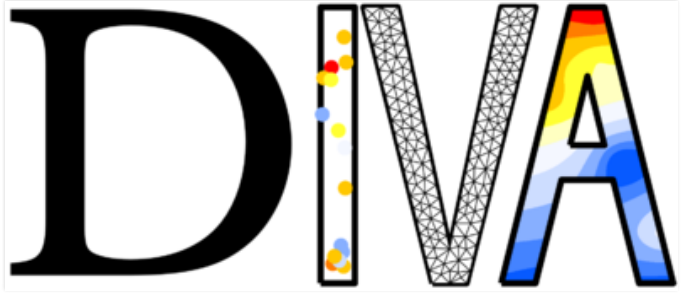
1. **Closer** observations have a **stronger** influence
2. Different **confidence** in some measurements
3. **Physical** barriers and currents
4. Deal with up to **millions** of points
5. Many sources of **errors** on observations
6. Need an associated **error field**



DIVA

Data-Interpolating Variational Analysis
https://github.com/gher-ulg/DIVA

DOI 10.5281/zenodo.1407062



DIVAnd

n-dimensional generalisation of DIVA
https://github.com/gher-ulg/DIVAnd.jl

DOI 10.5281/zenodo.1466985

README.md

DIVAnd

build passing build passing

coverage 82% codecov 84%

docs latest

`DIVAnd` performs an n-dimensional variational analysis of arbitrarily located observations. Observations will be interpolated on a curvilinear grid in 2, 3 or more dimensions.

Please cite this paper as follows if you use `DIVAnd` in a publication:

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

DIVAnd

https://www.geosci-model-dev.net/7/225/2014/

Geosci. Model Dev., 7, 225–241, 2014
www.geosci-model-dev.net/7/225/2014/
doi:10.5194/gmd-7-225-2014

Geoscientific Model Development
Open Access

`divand`-1.0: *n*-dimensional variational data analysis for ocean observations

A. Barth[1,*], J.-M. Beckers[1], C. Troupin[2], A. Alvera-Azcárate[1], and L. Vandenbulcke[3,4]

[1]GHER, University of Liège, Liège, Belgium
[2]IMEDEA, Esporles, Illes Balears, Spain
[3]seamod.ro/Jailoo srl, Sat Valeni, Com. Salatrucu, Jud. Arges, Romania
[4]CIIMAR, University of Porto, Porto, Portugal

[*] *Invited contribution by A. Barth, recipient of the EGU Arne Richter Award for Outstanding Young Scientists 2010.*

Correspondence to: A. Barth (a.barth@ulg.ac.be)

How to use it?

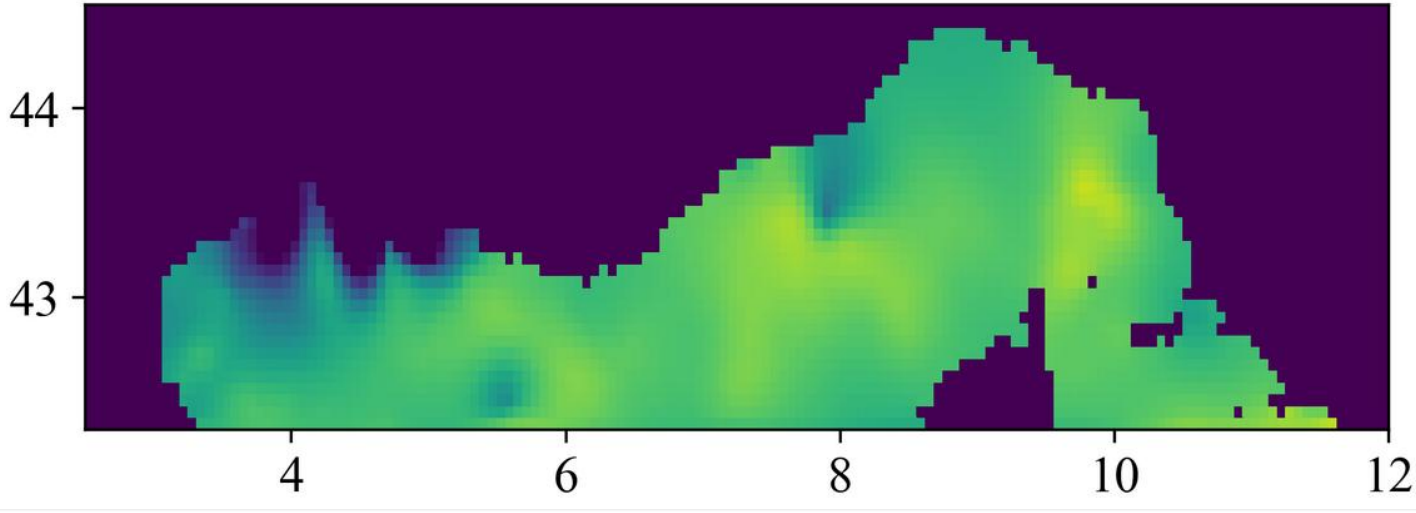
Jupyter notebooks as a guideline for the climatologies

Analysis `fi` using mean data as background.
Structure `s` is stored for later use in error calculation.

In [10]:
```
fi,s = DIVAndrun(mask,(pm,pn),(xi,yi),(obslon,obslat),obsval.-mean(obsval),len,epsilon2);
```

Create a simple plot of the analysis

In [11]:
```
pcolor(xi,yi,fi.+mean(obsval),vmin=37,vmax=38.5);
colorbar(orientation="horizontal")
gca()[:set_aspect](1/cos(mean([ylim()...]) * pi/180))
```



https://github.com/gher-ulg/Diva-Workshops

Fortran - MATLAB - Julia...

Creation: 2012
1.0.0 released: Aug 9, 2018
Simplicity of Python + speed of C or Fortran

http://julialang.org/
https://github.com/JuliaLang/julia

Who is Julia?

Julia Child (1912-2004)

By Lynn Gilbert - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=51678880

Why did we chose Julia?

Source: http://daftpunk.wikia.com, No copyright infringement is intended

Better...

Multiple dispatch
Math-friendly syntax
Unicode support: π, η, ∫ϵα

```
julia> 🚀 = 1.
julia> 🐢 = 2.
julia> N = 🚀 + 🐢
3.0
```

Faster

Just-in-time (JIT) compiled
Parallelism

```
function fib(n::Int)
  f=Vector{Int}(undef, n+1)
  f[1]=f[2]=1;
  for i=3:n+1
    f[i]=f[i-1]+f[i-2]
  end
  return f
end
ff = @time fib(400000000);
1.158971 seconds (18.52 k allocations: 2.981 GiB, 0.84% gc time)
```

Stronger

Metaprogramming:
Julia programs can read, analyse, generate other Julia programs

"Easy" interfacing: R, Python, ...

```
@pyimport numpy.random as nr
nr.rand(3,4)
```

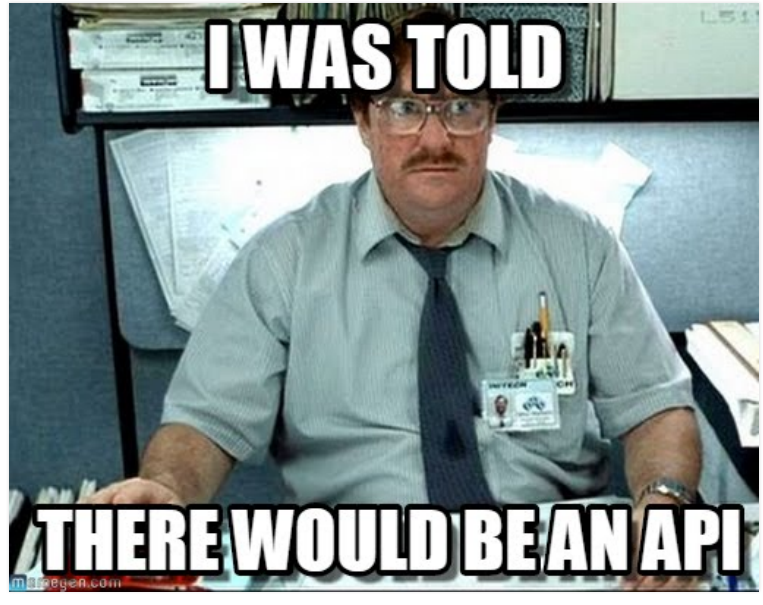
Harder

Learning a new and evolving language
Transition from 0.6 to 1.0

DIVAnd in the VRE

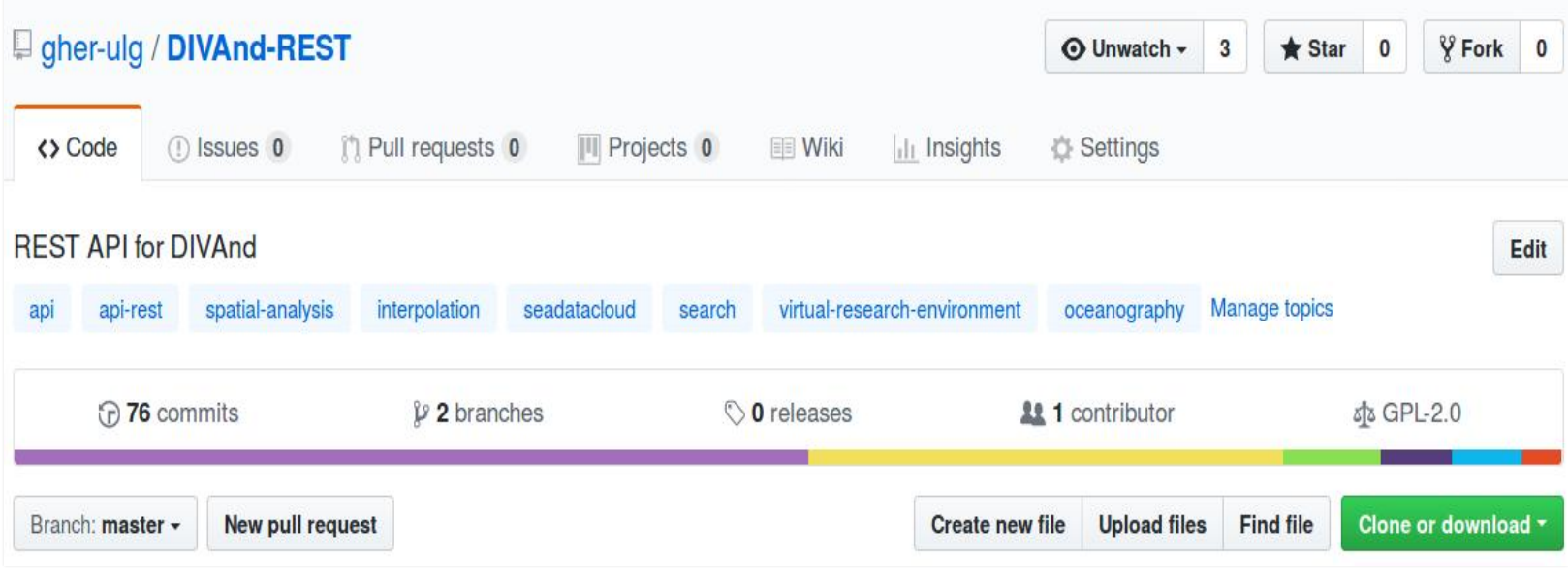
In short...

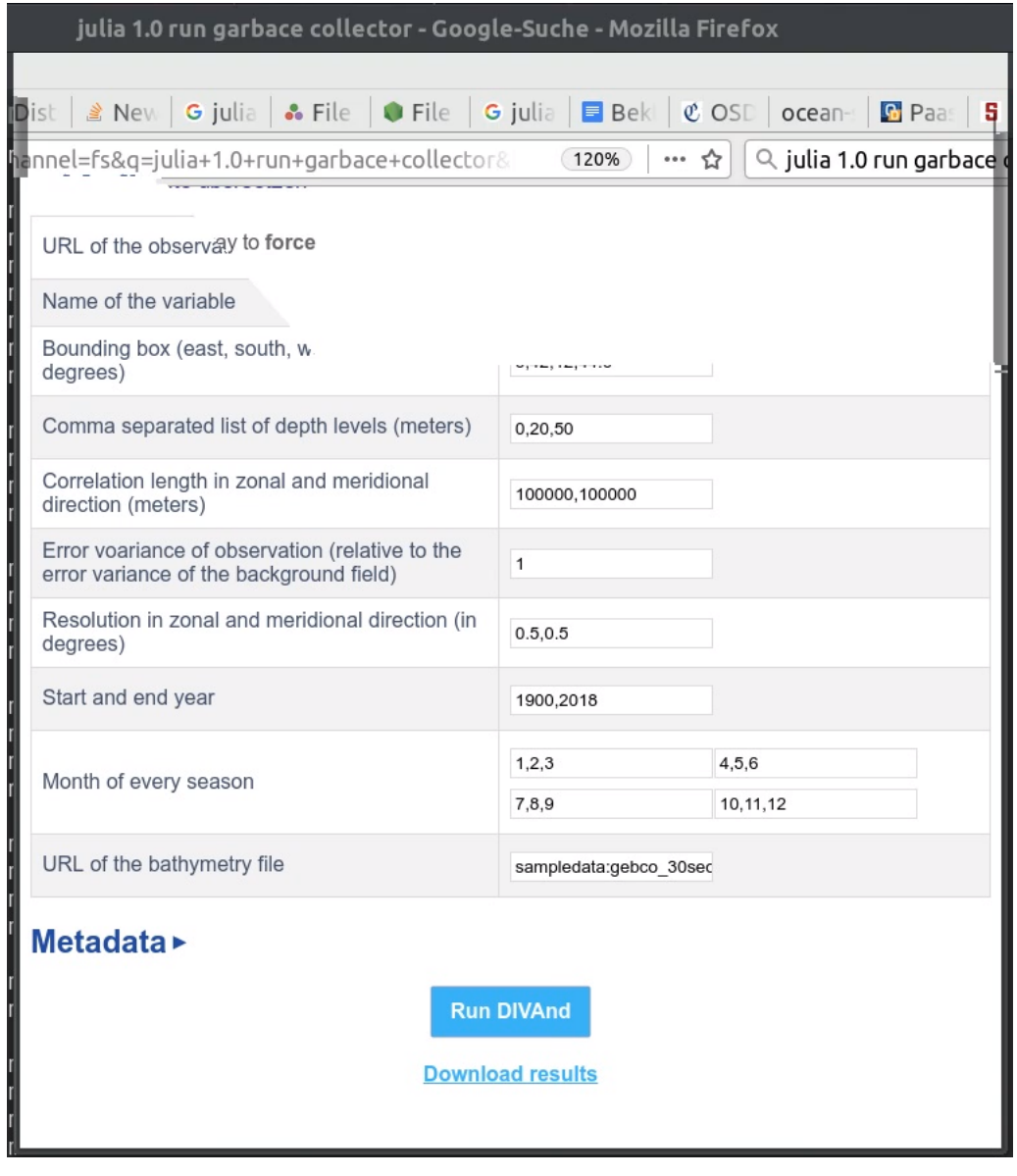
1. Ingest data from webODV (netCDF)
2. Set the analysis parameters
3. Apply DIVAnd interpolation
4. Export the results in a new netCDF
5. Visualise using Deltares toolbox


I WAS TOLD
THERE WOULD BE AN API

Implementation

1. Julia using HTTP and JSON modules
2. Deployment as a Docker container



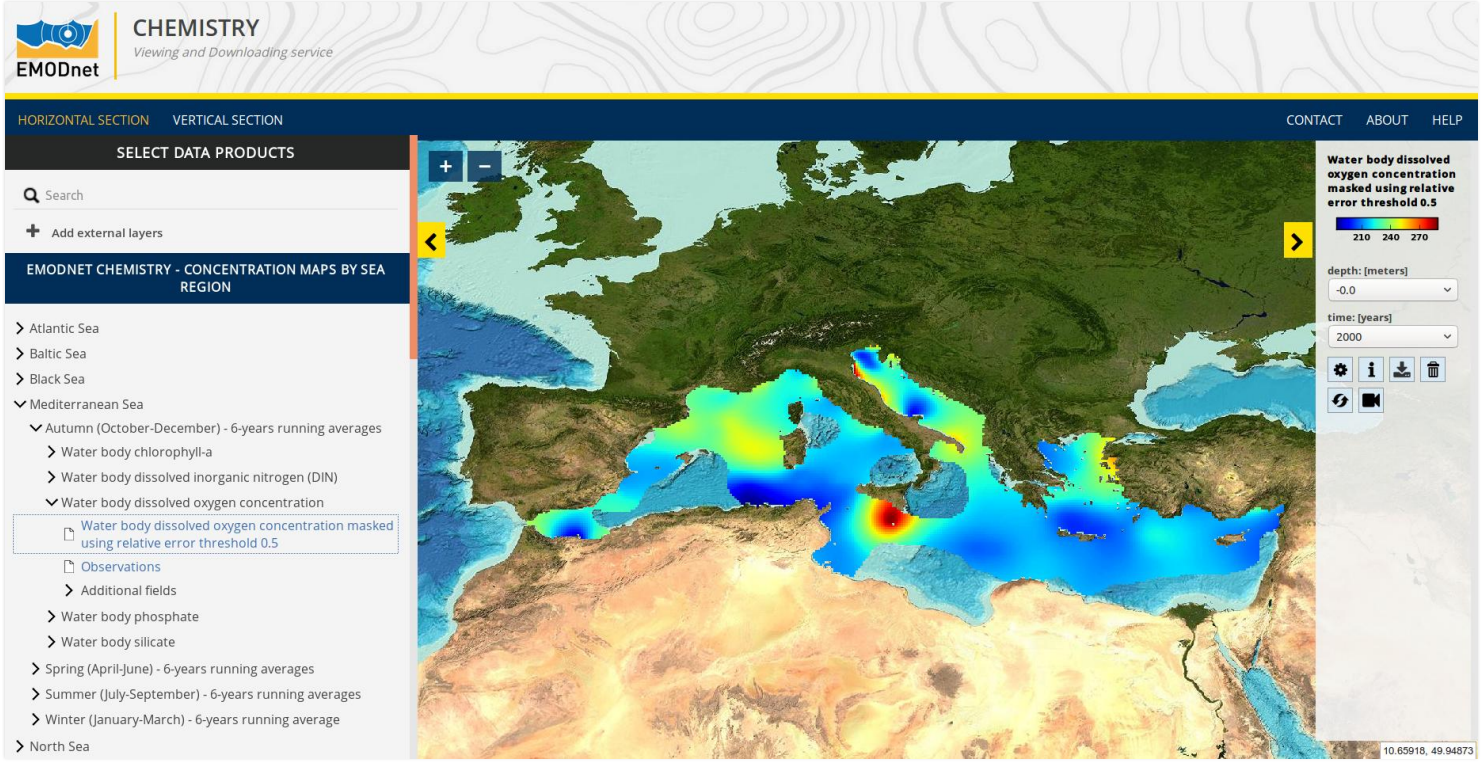

julia 1.0 run garbace collector - Google-Suche - Mozilla Firefox
120%
julia 1.0 run garbace
URL of the observ
ay to force
Name of the variable
Bounding box (east, south, w
degrees)
Comma separated list of depth levels (meters)
0,20,50
Correlation length in zonal and meridional direction (meters)
100000,100000
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

Applications

SeaDataCloud climatologies

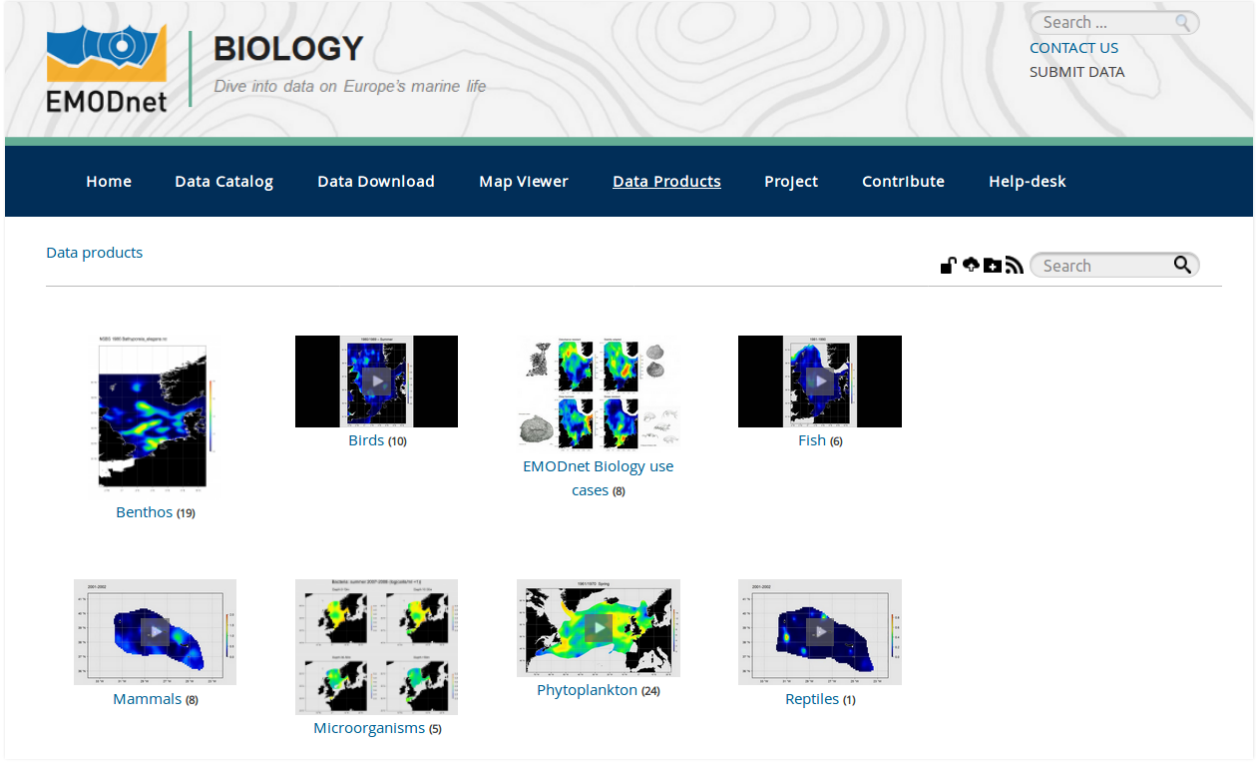
https://www.seadatanet.org/Products/Climatologies

EMODnet Chemistry gridded fields



http://www.emodnet-chemistry.eu/products

EMODnet Biology products

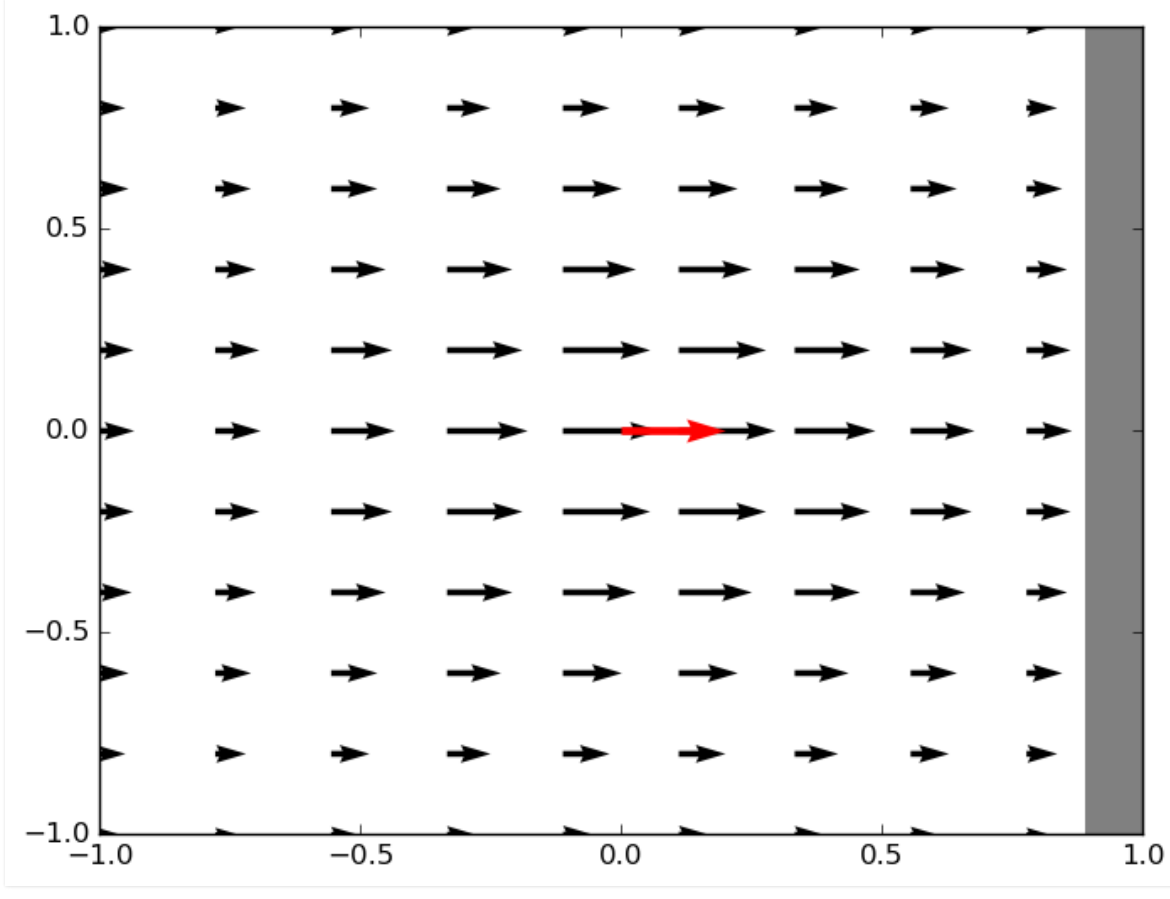


http://www.emodnet-biology.eu/data-products

#Innovations

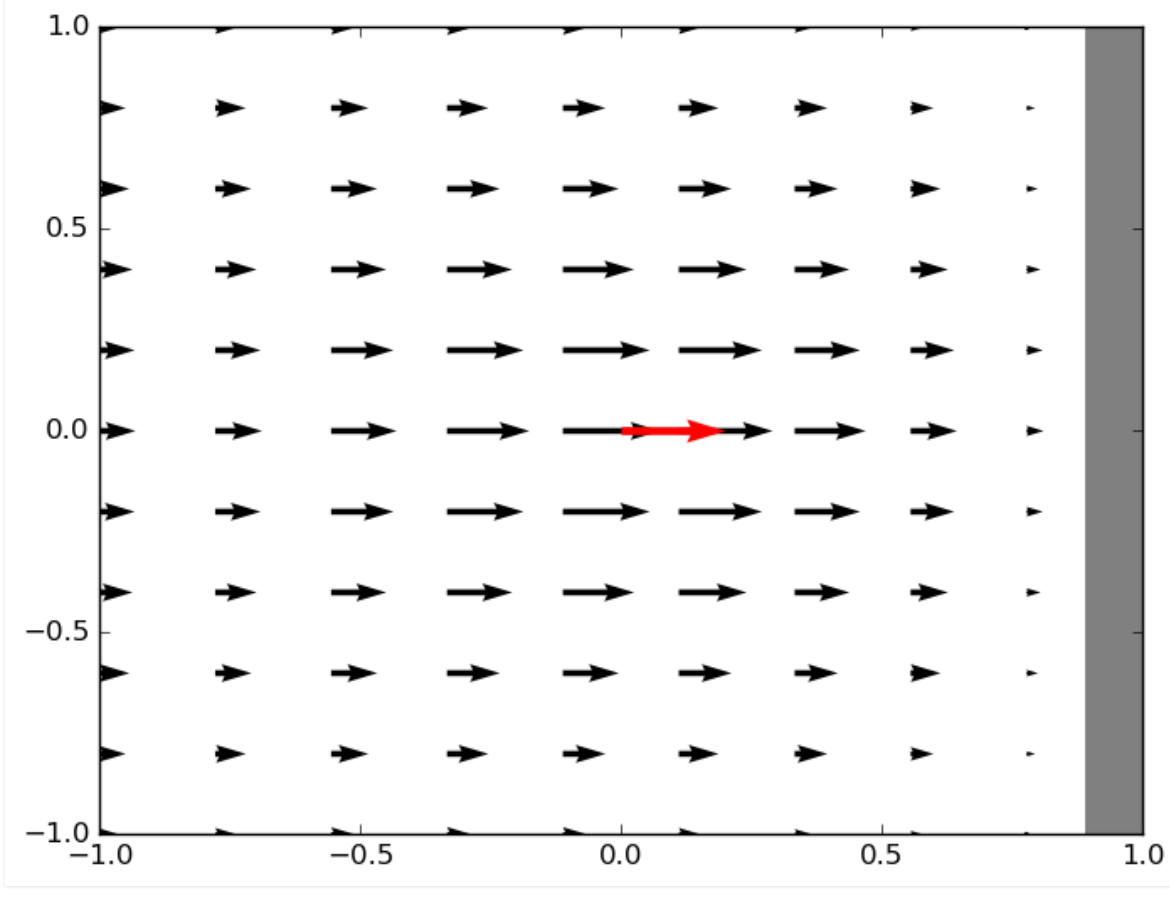
High-frequency radar interpolation

Synthetic velocity field, red arrow = measurement



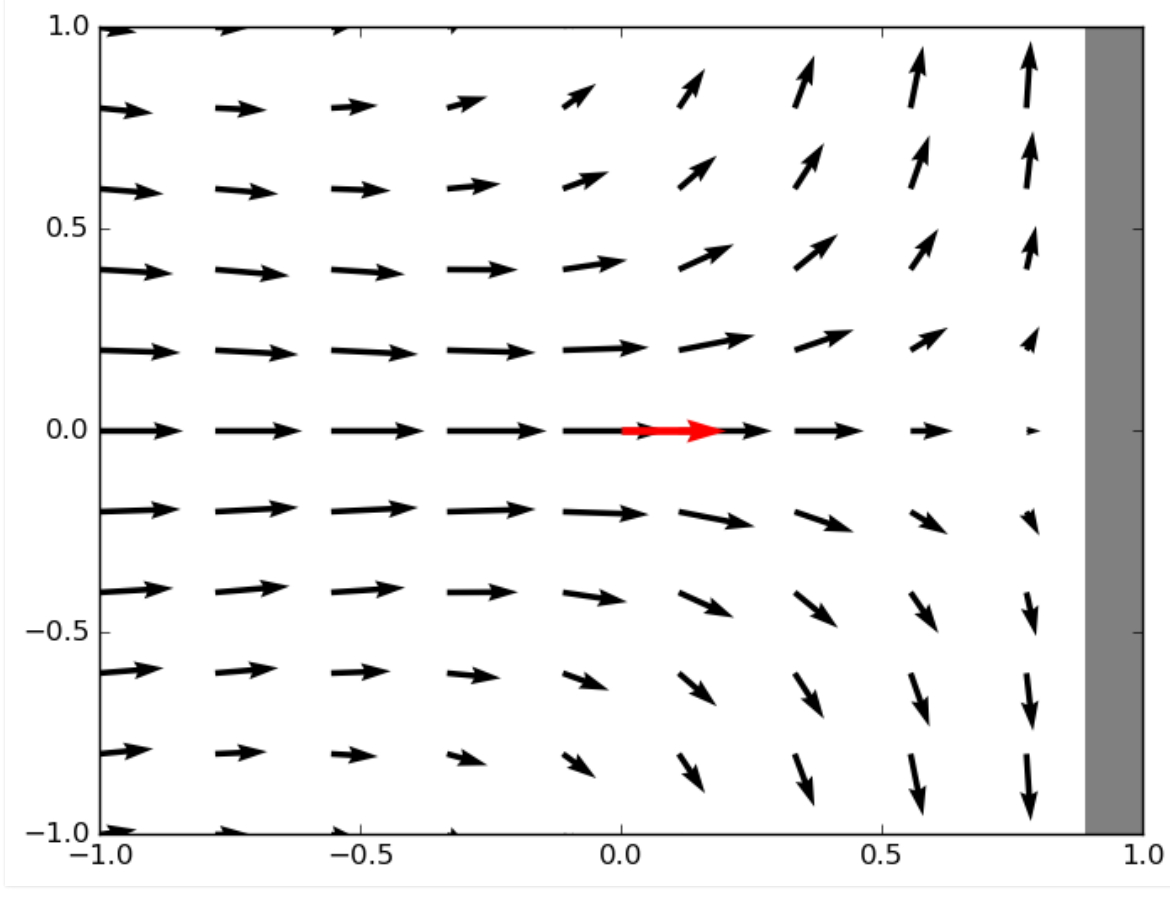
High-frequency radar interpolation

Adding the influence of the coast



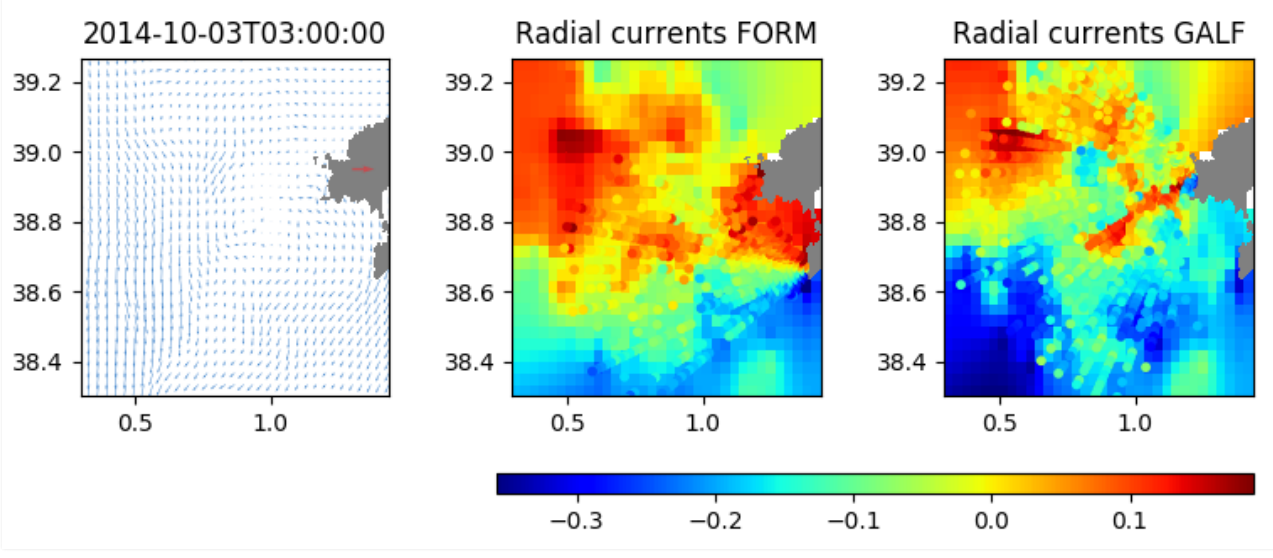
High-frequency radar interpolation

Low horizontal divergence of currents



High-frequency radar interpolation

Including Coriolis force and geostrophically balanced mean flow



Test areas: Ibiza Channel, Gulf of Trieste

Neural network

Neural network

From univariate to multivariate...

Principle:
Use other co-variables to improve the interpolation
Use neural network to derive the relationships between the variables

Application: zooplankton count in the Baltic Sea

Covariables:

- Dissolved oxygen → EMODnet Chemistry
- Salinity → SeaDataCloud
- Temperature → SeaDataCloud
- Chlorophyll concentration → MODIS-Aqua from NASA
- Bathymetry → EMODnet Bathymetry, GEBCO
- Distance from coast → GSFC, NASA

Application: zooplankton count in the Baltic Sea

Conclusions

1. `DIVA` is a software tool written in Fortran
2. `DIVAnd` is a software tool written in Julia
3. Both are designed for the spatial interpolation of data
4. We are open and willing to improve and adapt the code for different data types

Thanks for your attention

(and use `DIVA{nd}` many times)