Generating ocean climatologies from in situ observations

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**
**DIVAnd**

n-dimensional generalisation of DIVA

https://github.com/gher-ulg/DIVAnd.jl


**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:

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

A. Barth¹*, J.-M. Beckers¹, C. Troupin², A. Alvera-Azcárate¹, and L. Vandenbulcke³,⁴

¹GHER, University of Liège, Liège, Belgium
²IMEDEA, Esportes, Illes Balears, Spain
³seamod.ro/Jailoo srl, Sat Valeni, Com. Salatruce, Jud. Arges, Romania
⁴CIIMAR, 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.

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

Create a simple plot of the analysis

```python
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
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
Multiple dispatch
Math-friendly syntax
Unicode support: \( \pi, \eta, \int \in \alpha \)

```julia
julia> 🐠 = 1.
julia> 🐢 = 2.
julia> N = 🐠 + 🐢
3.0
```
Faster

Just-in-time (JIT) compiled
Parallelism

```latex
function fib(n::Int)
    f=Vector{Int}(undef, n+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)
```
Metaprogramming:
Julia programs can read, analyse, generate other Julia programs

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

```python
@pyimport numpy.random as nr
nr.rand(3,4)
```
Learning a new and evolving language
Transition from 0.6 to 1.0

Julia 1.0 Released
10 Aug 2018 | Andrew Claster

London, UK – Julia 1.0 was released today during JuliaCon 2018.

Today’s Julia 1.0 release is the most important Julia milestone since Julia was introduced in February 2012.

Julia 1.0 is the first complete, reliable, stable and forward-compatible Julia release. More information about Julia 1.0 is available here.
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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL of the observatory to force</td>
<td></td>
</tr>
<tr>
<td>Name of the variable</td>
<td></td>
</tr>
<tr>
<td>Bounding box (east, south, west, degrees)</td>
<td></td>
</tr>
<tr>
<td>Comma separated list of depth levels (meters)</td>
<td>0.2, 0.5</td>
</tr>
<tr>
<td>Correlation length in zonal and meridional</td>
<td>100000, 100000</td>
</tr>
<tr>
<td>direction (meters)</td>
<td></td>
</tr>
<tr>
<td>Error variance of observation (relative to</td>
<td>1</td>
</tr>
<tr>
<td>the error variance of the background field)</td>
<td></td>
</tr>
<tr>
<td>Resolution in zonal and meridional direction</td>
<td>0.5, 0.5</td>
</tr>
<tr>
<td>(in degrees)</td>
<td></td>
</tr>
<tr>
<td>Start and end year</td>
<td>1990, 2018</td>
</tr>
<tr>
<td>Month of every season</td>
<td>1, 2, 3, 4, 5, 6</td>
</tr>
<tr>
<td>Month of every season</td>
<td>7, 8, 9, 10, 11, 12</td>
</tr>
<tr>
<td>URL of the bathymetry file</td>
<td>sampledata.gebco_30sec</td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
</tr>
<tr>
<td>Run DIVAnd</td>
<td></td>
</tr>
<tr>
<td>Download results</td>
<td></td>
</tr>
</tbody>
</table>
Applications
SeaDataCloud climatologies

CLIMATOLOGIES

SeaDataNet gridded climatologies are based on the aggregated datasets v1.1. The preparation of the products has also improved the quality, the consistency and the overall coherence of the data made available by SeaDataNet. They have been computed using DIVA software.

https://www.seadatanet.org/Products/Climatologies
EMODnet Chemistry gridded fields

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

When you're fundraising, it's AI
When you're hiring, it's ML
When you're implementing, it's linear regression
When you're debugging, it's printf()
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

Covariates:

- **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)