D10.8: DIVA online operational in VRE

C. Troupin, A. Barth & J.-M. Beckers (GHER - ULiège)

7th TTG meeting, Plouzané, France, 14-15 October, 2019

sdn-userdesk@seadatanet.org – www.seadatanet.org
Definitions

**DIVA:** software tool designed for spatial interpolation
- [https://github.com/gher-ulg/DIVA](https://github.com/gher-ulg/DIVA)

**DIVAnd:** n-dimensional version of DIVA
- [https://github.com/gher-ulg/divand.jl](https://github.com/gher-ulg/divand.jl)

**Julia:** high-level, high-performance dynamic programming language for numerical computing
- [https://julialang.org/](https://julialang.org/)

**Notebook:** documents that contain live code, equations, visualizations and narrative text

**Jupyter:** open-source web application to create and share notebook
- [http://jupyter.org/](http://jupyter.org/)

**Jupyterhub:** multi-user server for Jupyter notebooks
Context for DIVA in VRE
Installing DIVA was sometimes...painful

Back in 2007
Installing DIVA was sometimes...painful

or in 2013
What have we improved since then?

1. New mathematical formulation
2. Julia language
3. Only 2 (!!!) input files
4. Applications as Jupyter notebooks

Barth et al. 2014 instead of Fortran data & bathymetry all in one
What have we improved since then?

Congrats to the @JuliaLanguage team on their 1.0 release! We look forward to watching the @JuliaComputing team use it to smash the competition like so much bœuf à la Bourguignonne! [github.com/JuliaLang/juli ... #ProudInvestor

11:59 PM - 8 Aug 2018
What have we improved since then?

Founder Collective

Congrats to the @JuliaLanguage team on their 1.0 release! We look forward to watching the @JuliaComputing team use it to smash the competition like so much bœuf à la Bourguignonne! [github.com/JuliaLang/juli ... #ProudInvestor

11:59 PM - 8 Aug 2018
Jupyter notebooks as guidelines

In [27]:

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

Explenatory text

```python
In [27]:
figure("Adriatic-Additional-Data", figsize=(2,2))
ax = subplot(1,1,1)
ax[0].set_tick_params("both", labelsize=6)
ylim(39.0, 46.0);
xlim(11.5, 20.0);
contourf(bx, by, permutedims(Float64.(mask_edit[:,1,:]),[1,2]),
levels=[1e-5,0], cmap="binary");
plot(obslon, obslat, "bo", markersize=2, label="SeaDataNet")
plot(obslonnew[2][newpoints], obslatnew[2][newpoints], "go",
    markersize=2, label="Additional data\nfrom World Ocean Database")
legend(loc=3, fontsize=4)
```
Jupyter notebooks as guidelines

```
In [27]:
figure("Adriatic-Additional-Data", figsize=(2,2))
ax = subplot(1,1,1)
ax[0].tick_params("both", labelsize=6)
ylim(39.8, 46.0);
xlim(11.5, 20.0);
contourf(bx, by, permutedims(Float64.(maskEdit[:,1:1]),[2,1]),
levels=[-1e5,0], cmap="binary");
plot(obslon, obslat, "bo", markersize=.2, label="SeaDataNet")
plot(obslon, obslat, newpoints, obslat, newpoints, "go",
     markersize=.2, label="Additional data\nfrom World Ocean Database")
legend(loc=3, fontsize=4)
```

Explanatory text

Code fragment
Jupyter notebooks as guidelines

Explornatory text

Code fragment

Results or figure
Jupyter notebooks as guidelines

Kernel (language)

Explenatory text

Code fragment

Results or figure
What should we improve?
What should we improve?

+ Access to computing power
+ Data availability
+ Documentation
What should we improve?

+ Access to computing power
+ Data availability
+ Documentation

Virtual Research Environment!
Deployment in the VRE
Goals: seamless generation of DIVA products

- Data
- Tool
- Computing power
Providing users access to notebooks

2017 1st deployment @CINECA
2019 Deployment transferred to DKRZ for the first training workshop

Multiple copies of a Docker container are run (https://hub.docker.com/r/abarth/divand-jupyterhub)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language (V1.2.0)
- Julia packages: PyPlot, NCDatasets, DataStructures,
- DIVAnd.jl (V2.4.0, August 2019)
- DIVAnd notebooks (latest version)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language

DIVAnd.jl (V2.4.0, August 2019)
DIVAnd notebooks (latest version)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language
- Julia packages: PyPlot, NCDatasets, DataStructures, ...

(V1.2.0)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language
- Julia packages: PyPlot, NCDatasets, DataStructures, ...
- DIVAnd.jl

(V1.2.0)

(V2.4.0, August 2019)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language (V1.2.0)
- Julia packages: PyPlot, NCDatasets, DataStructures, ...
- DIVAnd.jl (V2.4.0, August 2019)
- DIVAnd notebooks (latest version)
What’s the recipe for the container?

- Libraries: netCDF, unzip, git, ...
- Julia language (V1.2.0)
- Julia packages: PyPlot, NCDatasets, DataStructures, ...
- DIVAnd.jl (V2.4.0, August 2019)
- DIVAnd notebooks (latest version)

**Note:** no need to edit the docker file when the DIVAnd code or the notebooks are modified

```julia
RUN julia --eval 'using Pkg;
pkg"add https://github.com/gher-ulg/DIVAnd.jl#master"'
```
Julia is becoming more famous!
Julia is becoming more famous!

Julia: come for the syntax, stay for the speed

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.

Jeffrey M. Perkel

https://www.nature.com/articles/d41586-019-02310-3
### DIVAnd file I/O

<table>
<thead>
<tr>
<th>Pre-processing</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Observations</strong></td>
<td>- SeaDataNet</td>
<td>Gridded fields (netCDF)</td>
</tr>
<tr>
<td></td>
<td>- WOD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- CMEMS</td>
<td>Metadata (XML)</td>
</tr>
<tr>
<td><strong>Bathymetry</strong></td>
<td>- EMODnet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EMODnet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- GEBCO</td>
<td></td>
</tr>
</tbody>
</table>

- **webODV**
- **Deltares VIZ**
DIVAnd file I/O

Protocol: web

(Web Distributed Authoring and Versioning)

WEBDAV_USERNAME
WEBDAV_PASSWORD
WEBDAV_URL

defined at user login
DIVAnd file I/O

get("nextcloud_file", "jupyterhub_file")

put("jupyterhub_file", "nextcloud_file")
Multi-user performance during the SDN training course
Oostende, 19–26 June, 2019

1. Introduction to Julia
2. Extraction of a regional bathymetry
3. Data import using ODV
4. Climatology generation
Almost all the users managed to create a small climatology

Despite the computation load, the VRE stayed alive
Thanks for your attention