# Data visualization and processing with Jupyter Notebook in SeaDataCloud Virtual Research Environment

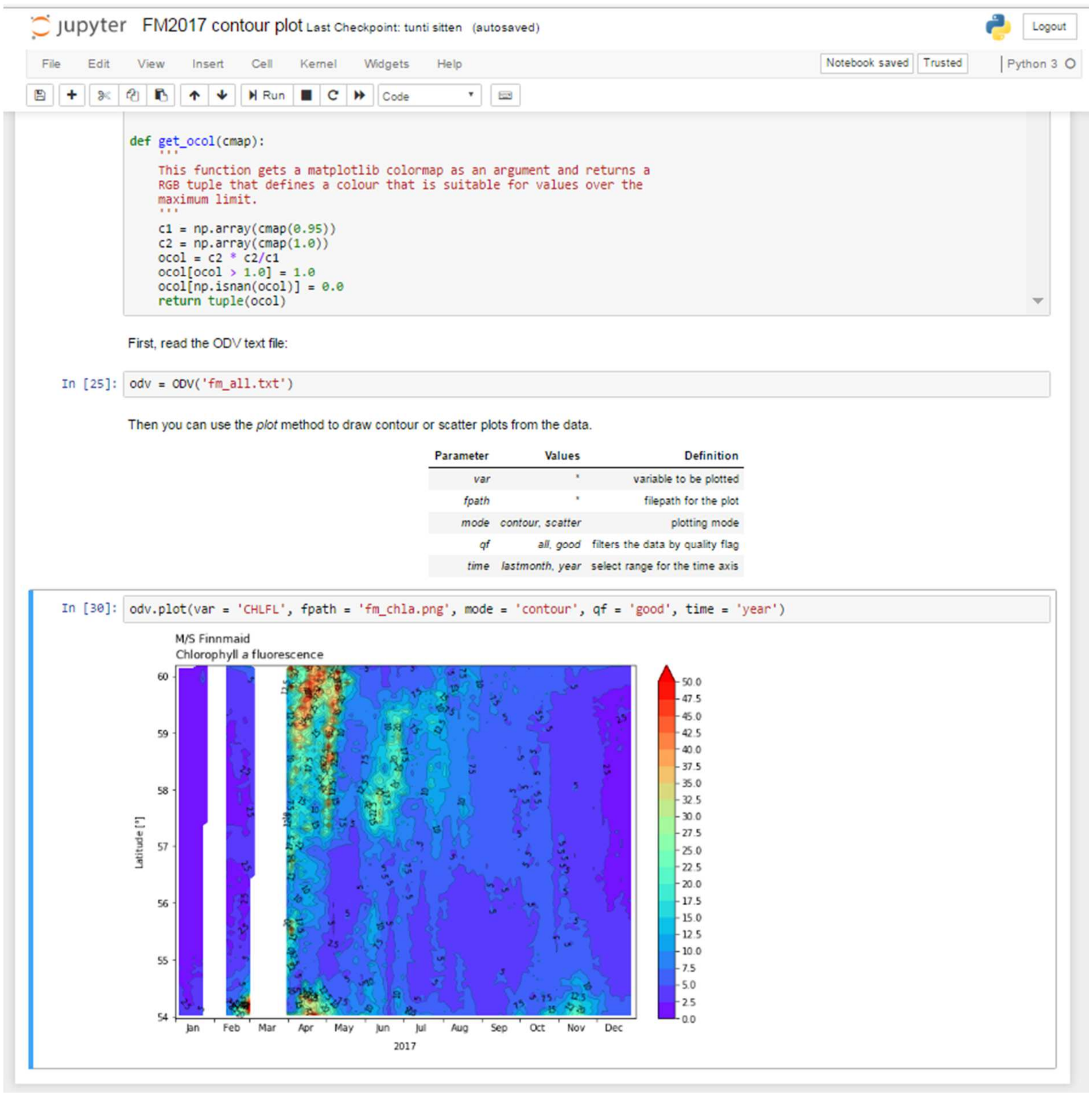
**Jani Ruohola,** Finnish Environment Institute SYKE (Finland), jani.ruohola@ymparisto.fi
**Sri Harsha Vathsavayi**, CSC IT Center for Science (Finland), sriharsha.vathsavayi@csc.fi
**Seppo Kaitala**, Finnish Environment Institute SYKE (Finland), seppo.kaitala@ymparisto.fi
**Christopher Ariyo**, CSC IT Center for Science (Finland), chris.ariyo@csc.fi

As reproducibility is an essential part of scientific research, the failure to reproduce experiments can lead to unreliable results and false scientific findings. Currently, irreproducible research is a problem across all domains of science[1]. We are facing a similar replication crisis in marine research as well. With the complex and rapidly changing nature of computer environments and software libraries, the computational reproducibility is getting significant attention from the domain scientists. In order to reproduce experiments, we need to capture the data, software environment and the whole workflow. This study demonstrates the use of notebooks and virtual research environments (VREs) to create reproducible experiments. We also demonstrate how Jupyter Notebooks coupled with user's personal data storage solutions (like EUDAT's B2DROP[2]) can open new possibilities for doing marine research.

An example workflow for reproducible research in SeaDataCloud VRE (SDC-VRE) is presented, which is a work-in-progress service and aims at integrating different tools, such as Jupyter Notebooks, B2DROP and webODV[3], to offer a common workplace for all marine researchers to facilitate the marine research within and without institutional boundaries. Data exists in the user's B2DROP account, which is integrated with the VRE. Data processing, analysis and visualization is conducted within Jupyter Notebook (Fig. 1) and the results can be distributed for example as netCDF files. The computational environment exists within the VRE in a separate Docker container, which offers OS-level virtualization and user-space isolation. Moreover, the data and notebooks can be securely shared with the research groups via B2DROP for collaboration and for reproducing the workflow.

The purpose of this workflow is to allow easily setting up and distributing different computational environments, without the so-called dependency hell. Docker containers allow the lightweight distribution of the environments and research data, while Jupyter Notebooks document the whole workflow, with rich-text annotations and figures. The versatility of Jupyter Notebooks also offer the possibility for clear and easily distributed tutorials of different computational methods. In the era of increasing importance of data-driven research, these tools are invaluable. However, they offer many possibilities also for more traditional hypothesis-driven experimental research, since the documentation of the computational part is as important as the documentation of the experimental methods.

[1] https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970
[2] https://eudat.eu/services/b2drop
[3] https://webodv.awi.de/



```
def get_ocol(cmap):
    '''
    This function gets a matplotlib colormap as an argument and returns a
    RGB tuple that defines a colour that is suitable for values over the
    maximum limit.
    '''
    c1 = np.array(cmap(0.95))
    c2 = np.array(cmap(1.0))
    ocol = c2 * c2/c1
    ocol[ocol > 1.0] = 1.0
    ocol[np.isnan(ocol)] = 0.0
    return tuple(ocol)
```

First, read the ODV text file:

```
In [25]: odv = ODV('fm_all.txt')
```

Then you can use the *plot* method to draw contour or scatter plots from the data.

| Parameter | Values | Definition |
|---:|---:|---:|
| *var* | * | variable to be plotted |
| *fpath* | * | filepath for the plot |
| *mode* | *contour, scatter* | plotting mode |
| *qf* | *all, good* | filters the data by quality flag |
| *time* | *lastmonth, year* | select range for the time axis |

```
In [30]: odv.plot(var = 'CHLFL', fpath = 'fm_chla.png', mode = 'contour', qf = 'good', time = 'year')
```

*Figure 1: Contour plot of chlorophyll a fluorescence for a FerryBox route interpolated and visualized with Jupyter running inside the SDC-VRE. Inline figures, Markdown and LaTeX support offer an efficient method for documentation of the workflow.*