Breakout Workshop: Scientific computing and the Open Source software revolution: Opportunities for ocean colour science

Joaquín E. Chaves (NASA GSFC/SSAI), Erdem M. Karaköylü (NASA GSFC/SAIC), Joel P. Scott (NASA GSFC/SAIC)
<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:00–2:05</td>
<td>Intro, motivation and goals (co-chair: <strong>J. Chaves</strong>, NASA GSFC/SSAI): Open Source software revolution: Resources and opportunities for ocean color science research, data distribution, education and mentoring</td>
</tr>
<tr>
<td>2:05–2:25</td>
<td>Python for multi-year GOCI ocean color products analysis: sharing the advantages and issues (<strong>Dr. Myung-Sook Park</strong>, KIOST)</td>
</tr>
<tr>
<td>2:25–2:30</td>
<td>QA and interactive discussion.</td>
</tr>
<tr>
<td>2:50–2:55</td>
<td>QA and interactive discussion.</td>
</tr>
<tr>
<td>2:55–3:00</td>
<td>Coffee break</td>
</tr>
<tr>
<td>3:00–3:20</td>
<td>Use of Python and SeaDAS for Ocean Color Processing, Analysis and Visualization (<strong>Bruce Monger</strong>, Cornell University)</td>
</tr>
<tr>
<td>3:25–3:45</td>
<td>The importance of open science principles for expanding and diversifying the user base of marine satellite data (<strong>Dr. Hayley Evers-King</strong>, Plymouth Marine Lab)</td>
</tr>
<tr>
<td>3:45–3:50</td>
<td>QA and interactive discussion.</td>
</tr>
<tr>
<td>3:50–4:10</td>
<td>Probabilistic Programming Workflow in Python (co-chair <strong>E. Karaköylü</strong>, NASA GSFC/SAIC)</td>
</tr>
<tr>
<td>4:10–4:15</td>
<td>QA and interactive discussion.</td>
</tr>
<tr>
<td>4:15–4:30</td>
<td>Summary and group discussion (Co-chairs, others)</td>
</tr>
</tbody>
</table>
Agenda:

Key questions:

- What recent shifts among Ocean Color community have there been towards open source tools?
- How and why is the community using these tools? □
- How are teaching, mentoring, and research deliverables being changed by open source?
- Are these tools promoting reproducibility, code preservation, accessibility, diversity & inclusion?

Breakout Workshop: Scientific computing and the Open Source software revolution:
What is Open Source software?

“Open source software is software with source code that anyone can inspect, modify, and enhance”
Some early history:

- Historically (1950–60s) software was free, profits were on hardware.
- AT&T’s commercialization of UNIX ensues the OS movement:
  - Richard Stallman, then at MIT is credited as the founder of the open source software movement in the early 1970s.
- GNU, BSD & MIT licenses for UNIX–like systems
- Linux
...and today:

OS/server platforms:

Computing:

Database tools:
The Incredible Growth of Python

by David Robinson on September 6, 2017

Breakout Workshop: Scientific computing and the Open Source software revolution:

https://stackoverflow.blog/2017/09/06/incredible-growth-python/
Python popularity in astronomy:

- IDL
- Matlab
- Python

Breakout Workshop: Scientific computing and the Open Source software revolution:

Breakout Workshop: Scientific computing and the Open Source software revolution:

https://pypi.org/
Open Source and diversity in scientific computing:

NumFOCUS Diversity Statement

NumFOCUS welcomes and encourages participation in our community by people of all backgrounds and identities. We are committed to promoting and sustaining a culture that values mutual respect, tolerance, and learning, and we work together as a community to help each other live out these values.

We have created this diversity statement because we believe that a diverse community is stronger, more vibrant, and produces better software and better science. A diverse community where people treat each other with respect has more potential contributors, more sources for ideas, and fewer shared assumptions that might hinder development or research.

Although we have phrased the formal diversity statement generically to make it all-inclusive, we recognize that there are specific identities that are impacted by systemic discrimination and marginalization. We welcome all people to participate in the NumFOCUS community regardless of their identity or background.

https://numfocus.org/code-of-conduct
...Let’s hear it from the speakers
Back up slides for discussion at the end...
Repositories as research deliverables and complement to traditional papers:

https://joss.theoj.org/
NUMFOCUS
Fiscally Sponsored Projects

Julia, pandas, Jupyter, NumPy, Matplotlib, JUMP, Stan, IPython, SymPy, Bokeh, PyTables, PyMC3, nteract, FEniCS Project, yt, EconARK, QuantEcon, openJournals, sunpy, Astropy, xarray, OpenSci, CONDA-FORGE, Cogenera

Open Code = Better Science
How to visualize data in Python

Welcome to PyViz! PyViz is a coordinated effort to make data visualization in Python easier to use, easier to learn, and more powerful.

Focusing on interactive plotting in web browsers, PyViz provides:

- High-level tools that make it easier to apply Python plotting libraries to your data.
- A comprehensive tutorial showing how to use the available tools together to do a wide range of different tasks.
- A Conda metadata package "pyviz" that makes it simple to install matching versions of libraries that work well together.
- Sample datasets to work with.

Core high-level libraries

Panel, hvPlot, HoloViews, GeoViews

These PyViz-maintained packages each make great starting points — Panel for making apps and dashboards for your plots from any supported plotting library, hvPlot to quickly generate interactive plots from your data, HoloViews to help you make all of your data instantly visualizable, and GeoViews to extend HoloViews for geographic data.

Supported viz libraries

Bokeh, matplotlib, Datashader

PyViz tools provide extensive support for Bokeh's interactive plotting, Matplotlib's publication-quality output, and Datashader's rendering of even the largest datasets.

plotly, seaborn, Altair, Vega, Folium, GraphViz, pillow

Objects from nearly every other plotting library can be used with Panel, including specific support for all those listed here plus anything that can generate HTML, PNG, or SVG. HoloViews also supports PyViz for 3D visualizations.

http://pyviz.org/
EarthML

Machine learning and visualization in Python for Earth science

Python offers a wide variety of open-source libraries covering a huge range of functionality, but it can be difficult to work out which libraries are suitable for which tasks. The EarthML project helps to:

- Demonstrate how to use Python tools for machine learning and analysis in the earth sciences
- Identify libraries suitable for working with earth science data
- Make improvements to these libraries as needed to help improve earth-science workflows

EarthML contains no code of its own, only tutorials and examples showing how to use packages like:

- **Data libraries:**
  - Integridy: Easily loading data from various sources.
  - Skimage: Processing grids and (n-dimensional) data structures.
  - Pandas: Processing columnar (tabular) data structures.
  - GeoPandas: Geospatial operations on tabular data.

- **Visualization libraries:**
  - VizPlot: Simple data-centric API for plotting, building on:
    - Bokeh: Interactive browser-based plotting.
    - Cartopy: Easy construction of geospatial plots.
    - GeoViz: GeoJSON visualization with earth-specific projections.
    - DataDots: Rendering large datasets into images for display in browsers.
    - Panel: Dashboards, apps, and widgets for any library’s plots.

- **Other tools:**
  - Jupyter: Reproducible notebooks (source code for all examples on this site).
  - Cartopy: Geographic coordinate reference systems.
  - GeoPandas: Geospatial operations at scale.

- **ML tools:** (representative only – use any you like)
  - Scikit-learn: General ML for numeric data.

The EarthML Tutorial offers a general-purpose overview of the concepts and tools involved, and the Tutorials section shows examples of how these tools may be used to perform machine learning and related tasks in the Earth sciences, such as:

- Carbon Flux
- Heat and Thermo
- Water Cycle
- Classification
- Clustering

Please feel free to report issues or contribute code.

Installation

Step 1: Install a Miniconda (or Anaconda) environment

Any Linux or Mac OS X computer with a modern web browser (preferably Google Chrome) should be suitable. Windows machines can also run most of the functionality, but the EarthML team is currently investigating some Windows-specific issues. WSGI of Flask is required for some of the examples, but most will run fine in WSGI.

http://earthml.pyviz.org/index.html
Exponential growth in Machine Learning and Artificial Intelligence has been driven by open source:
Earth Observation Open Science and Innovation

Editors (view affiliations)
Pierre-Philippe Mathieu, Christoph Aubrecht

Open Access | Book
Citations 154 | Mentions 178 | Readers 109k | Downloads

Part of the ISSI Scientific Report Series book series (ISSI, volume 15)

Download book PDF | Download book EPUB

Machine Learning Applications for Earth Observation

Abstract

Machine learning has found many applications in remote sensing. These applications range from retrieval algorithms to bias correction, from code acceleration to detection of disease in crops, from classification of pelagic habitats to rock type classification. As a broad subfield of artificial intelligence, machine learning is concerned with algorithms and techniques that allow computers to “learn” by example. The major focus of machine learning is to extract information from data automatically by computational and statistical methods. Over the last decade there has been considerable progress in developing a machine learning methodology for a variety of Earth Science applications involving trace gases, retrievals, aerosol products, land surface products, vegetation indices, and most recently, ocean applications. In this chapter, we will review some examples of how machine learning has already been useful for remote sensing and some likely future applications.
Open Source in education, mentoring, and knowledge distribution:

Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Language of choice

The Notebook has support for over 40 programming languages, including Python, R, Julia, and Julia.

Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Interactive output

Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.

Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore the same data with pandas, scikit-learn, ggplot2, TensorFlow.
Repositories as research deliverables and complement to traditional papers: