ARGO data visualization and delivery: for profiles and gridded products

Tyler Tucker, Donata Giglio, Megan Scanderbeg

University of Colorado, Boulder

October 20th 2019

Web Application Overview

- Argo data made available by GDACS is used by the community to make inferences.
- The traditional visualization and delivery tools can be improved to handle the nuances and complexity of this data set (RT, Delayed, BGC, Synthetic...).
- Argovis created to deliver and visualize profile data using modern web application architecture.
- Expanded into derived (gridded) products.
- Target audience: Students, Argo users (including BGC, polar, deep, and grid modelers), data managers, general public.
- Setup: All done on browser with an internet connection. Heavily tested on Firefox. Chrome works well too.

Argovis Module: Home

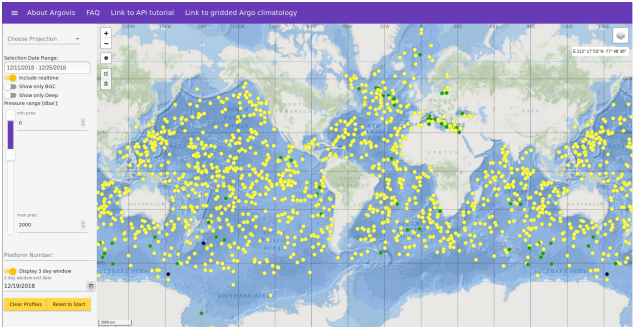

Figure: `https://argovis.colorado.edu`

- Angular web framework handles Front-end logic.
- Express.js handles back end logic/db routing, and API.
- MongoDB stores data.

Argovis Module: API

API call types	Input	Output	
Profile	Platform number and cycle number	lon, lat, position_qc POSITIONING_SYSTEM date, date_qc, date_added cycle_number, platform_number PLATFORM_TYPE DATA_MODE, DATA_CENTRE station_parameters, station_parameters_in_nc VERTICAL_SAMPLING_SCHEME DIRECTION, PI_NAME	Measurements: **TEMP** **PRES** **PSAL** WMO_INST_TYPE
Platform	Platform number		
Region	Time range, pressure range, set of latitude and longitude vertices to define shape		
Global metadata by month	Month and year		**PRES_max_for_TEMP** **PRES_max_for_PSAL** **PRES_min_for_TEMP** **PRES_min_for_PSAL**

Figure: Current API capabilities

- Meta data aggregated by month/year For example: `https://argovis.colorado.edu/selection/profiles/10/2018/`
- Can aggregate meta data by: lat-lon, DAC, etc.

Argovis Module: Profile Trajectory

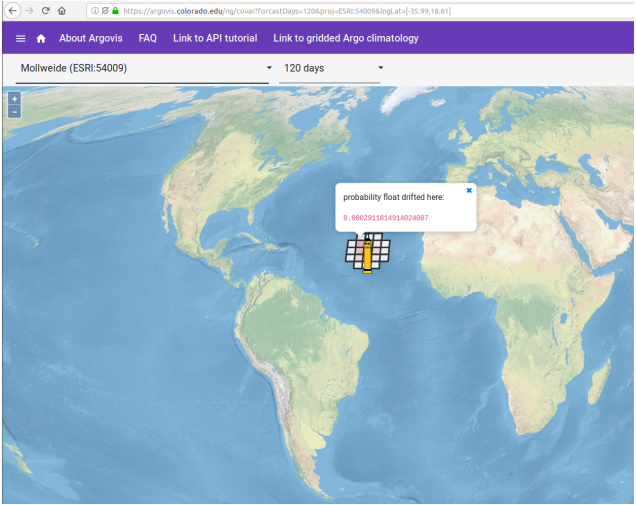

Figure: `https://argovis.colorado.edu/ng/covar`

- Trajectory estimates Paul Chamberlain's co variance estimations. (Paper is in review)
- Same archecteture, as Home module.
- Uses Open Layers Web map (Home uses Leaflet).

Argovis Module: Grid

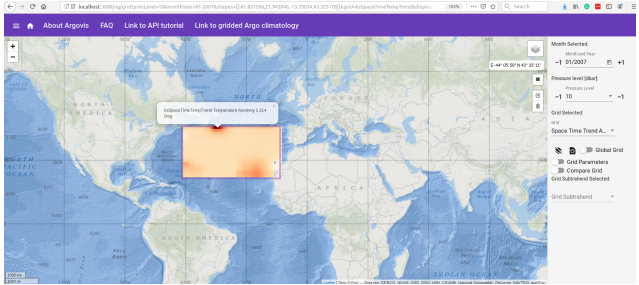

Figure: `https://argovis.colorado.edu/ng/grid`

- Displays local/global gridded products .
- Grid comparison tool takes difference between two grids.
- Model parameter display feature (in progress).
- Going to add additional grids in the coming months.

Opportunities

- Collaboration with other gridded products
- Possibility to co-locate gridded fields with Argo profiles
- Gridded display framework can be applied to other products (satellite track or other gridded products)
- Performance enhancements such as web assembly D3.js
- Improve or contribute to outreach
- Give user the option of changing colorscales, log-scales, etc.
- Extend API functions. (Make API calls in R, Matlab, etc)

Challenges

- How to compare grids of different resolution/pressure scales? Interpolation? Use a standardized grid?
- Address smart phone and tablet users
- How to display ultra fine grids and still perform the same?
- How to change map projection without loading page?
- End-to-end testing of front-end and back-end.

Additional materials

- Correspondence to developer: tyler.tucker@colorado.edu
- Request features or report bugs here on GitHub projects: `https://github.com/tylertucker202`
- Personal Blog: `https://www.itsonlyamodel.us`
- YouTube Tutorials: `https://www.youtube.com/watch?v=UOjSV4EIy00&list=PLJQE6R74BkyVV3PeTRdGh-W5llAY4XJ8v`

SWOT

STRENGTHS

1. Fast access
2. API
3. No need to download software
4. Uses FOSS libraries
5. Cheap to host

WEAKNESSES

1. Dependent on external libraries (leaflet)
2. Needs more testing
3. Angular front-end is still evolving

OPPORTUNITIES

1. Introduce other data sources (gliders, satalite products)
2. Deploy more tutorials and videos as the app expands
3. Include more gridded products

THREATS

1. One developer
2. Maintaining dependencies
3. Addressing GDAC changes