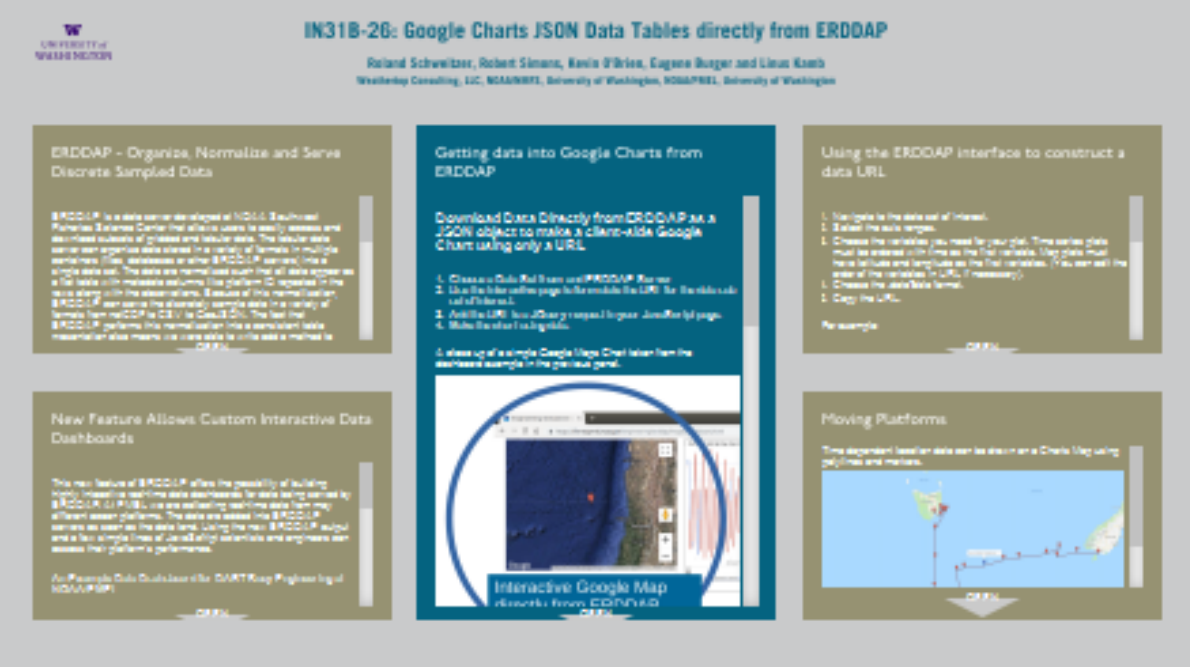
IN31B-26: Google Charts JSON Data Tables directly from ERDDAP



Roland Schweitzer, Robert Simons, Kevin O'Brien, Eugene Burger and Linus Kamb

Weathertop Consulting, LLC, NOAA/NMFS, University of Washington, NOAA/PMEL, University of Washington

PRESENTED AT:

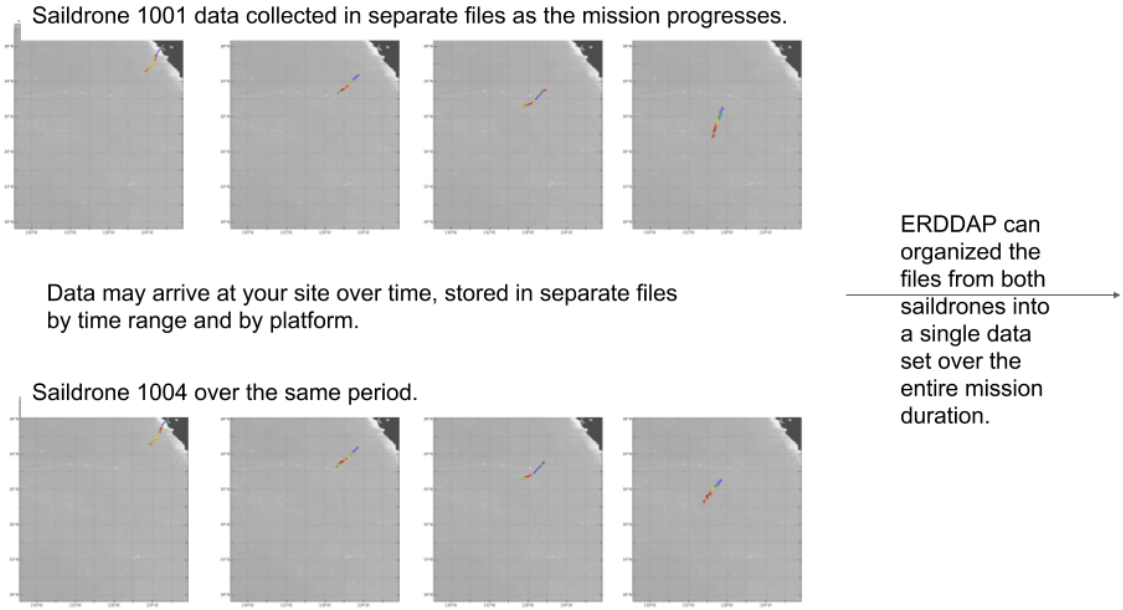
ERDDAP - ORGANIZE, NORMALIZE AND SERVE DISCRETE SAMPLED DATA

ERDDAP is a data server developed at NOAA Southwest Fisheries Science Center that allows users to easily access and download subsets of gridded and tabular data. The tabular data server can organize data stored in a variety of formats in multiple containers (files, databases or other ERDDAP servers) into a single data set. The data are normalized such that all data appear as a flat table with metadata columns like platform ID repeated in the rows along with the observations. Beause of this normailization, ERDDAP can serve the discretely sample data in a variety of formats from netCDF to CSV to GeoJSON. The fact that ERDDAP performs this normalization into a consistent table presentation also means we were able to write add a method to ERDDAP to write the tablular data as a Google Charts DataTable JSON object. This allows JavaScript clients using Google Charts to import ERDDAP data directly from a URL.

ERDDAP Home Page (http://coastwatch.pfeg.noaa.gov/erddap/index.html)

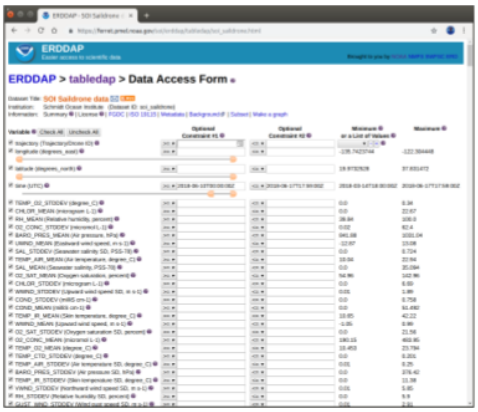
Beside being able to serve data read from a huge variety of file formats and soruces and being able to serve the resulting collection in may formats, ERDDAP can aggregate collections of files into a single data set.

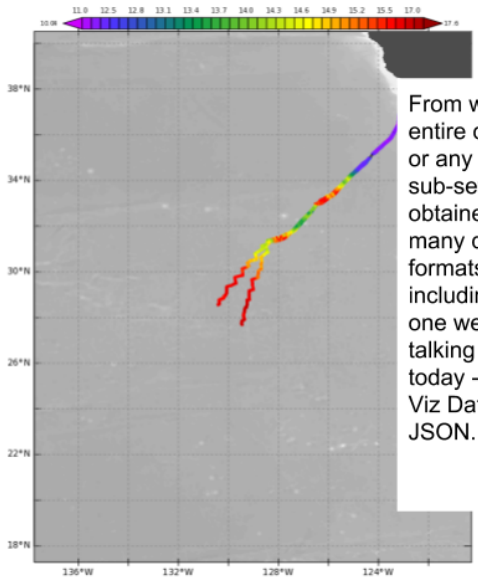
During a recent saildrone mission we collected daily data files from two saildrones.



These files were combined in realtime on the ERDDAP server to produce a single data set serving data over all times and both drones.

Both saildrones from the individual time and drone files aggregated into a single data set.





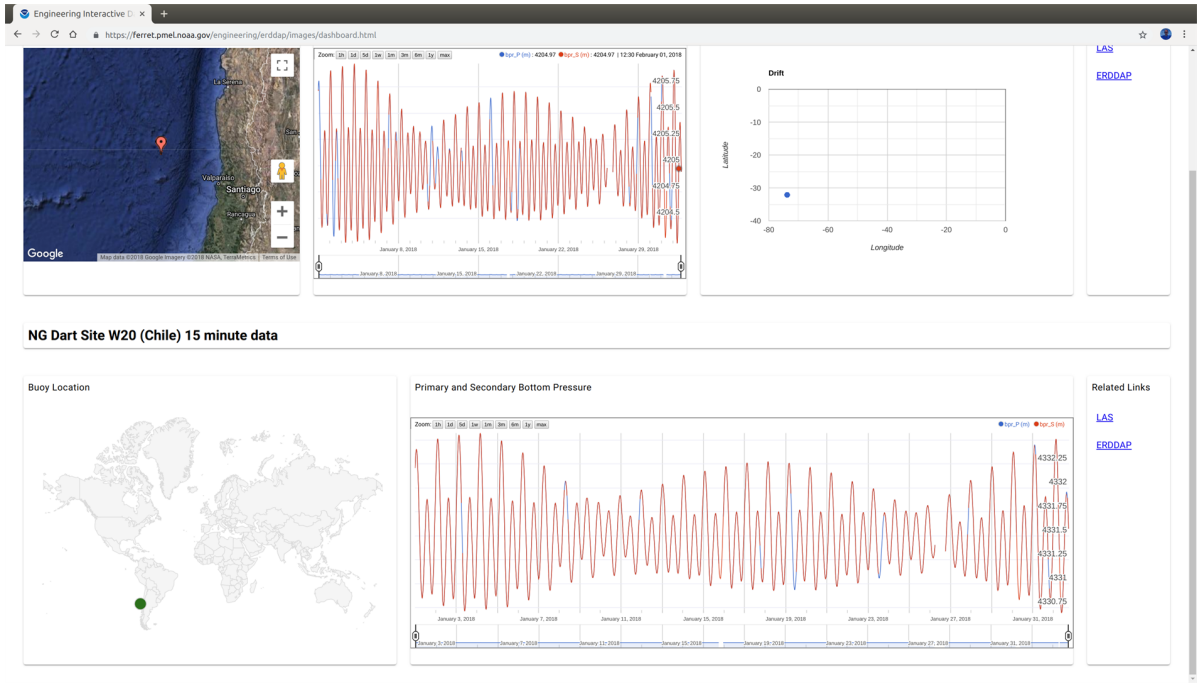
From which the entire data set or any desired sub-set can be obtained in many different formats including the one we're talking about today - Google Viz DataTable JSON.

We have added the Google DataTable JSON format to the long list of possible data types available form an ERDDAP server. The rest of this presentaiton explores the use of this new capability.

NEW FEATURE ALLOWS CUSTOM INTERACTIVE DATA DASHBOARDS

This new feature of ERDDAP offers the possbility of building highly inteactive real-time data dashboards for data being served by ERDDAP. At PMEL we are collecting real-time data from may different ocean platforms. The data are added into ERDDAP servers as soon as the data land. Using the new ERDDAP output and a few simple lines of JavaScfript scientists and engineers can assess their platform's performance.

An Example Data Dsahsboard for DART Buoy Engineering at NOAA/PMEL



A Live Link to this Example (http://ferret.pmel.noaa.gov/engineering/erddap/images/dashboard.html)

Because the charts are drawn on the client they are interactive.

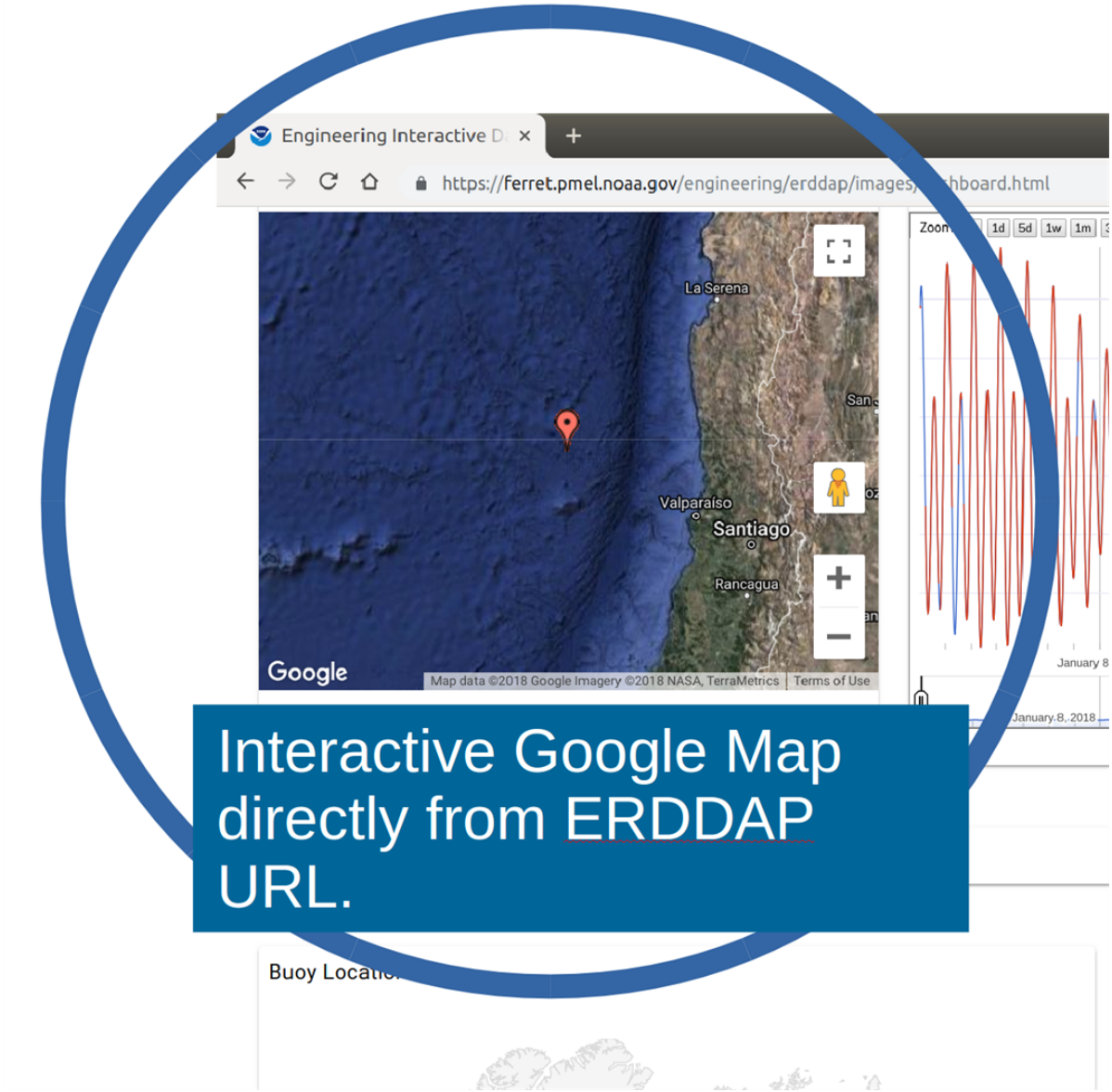
[VIDEO] https://www.youtube.com/embed/xEYjJ6atihs?feature=oembed&fs=1&modestbranding=1&rel=0&showinfo=0

GETTING DATA INTO GOOGLE CHARTS FROM ERDDAP

Download Data Directly from ERDDAP as a JSON object to make a client-side Google Chart using only a URL

1. **Choose a Data Set from and ERDDAP Server**
2. **Use the interactive page to formulate the URL for the data sub-set of interest.**
3. **Add the URL to a JQuery request in your JavaScript page.**
4. **Make the chart using data.**

A close up of a simple Google Maps Chart taken from the dashboard example in the previous panel.



For example here's how to make a simple Google Maps Chart to show the location of a Dart Buoy.

This map is constructed with just 7 lines of JavaScript code. The first line is loading the Google Maps chart type and setting your Google Maps key (http://developers.google.com/chart/interactive/docs/gallery/map). The data download is done with a JQuery call in one line as show below:

```
var locationData = $.ajax({
                   url: "ERDDAP URL",
                   dataType: "json",
                   async: false
                   }).responseText;
```

You can construct the "ERDDAP URL"*** in the interactive ERDDAP Web interface.

After downloading the JSON data to be plotted you can construct the JavaScript object which is the input to all Google Charts -- a Google Charts DataTable object.

```
var locdata =
    new google.visualization.DataTable(locationData);
```

After setting a few map options for how the data will be displayed:

```
var moptions = {
                zoomLevel: 6,
                displayMode: 'markers',
                height: 370
               };
```

The next line of code tells the Google Charts library to create the chart and where to put the chart on you page.

```
var mapchart =
    new google.visualization.Map(
           document.getElementById('map_div')
        );
```

The last line of code, renders the chart using the data you downloaded from ERDDAP.

```
mapchart.draw(locdata, moptions);
```

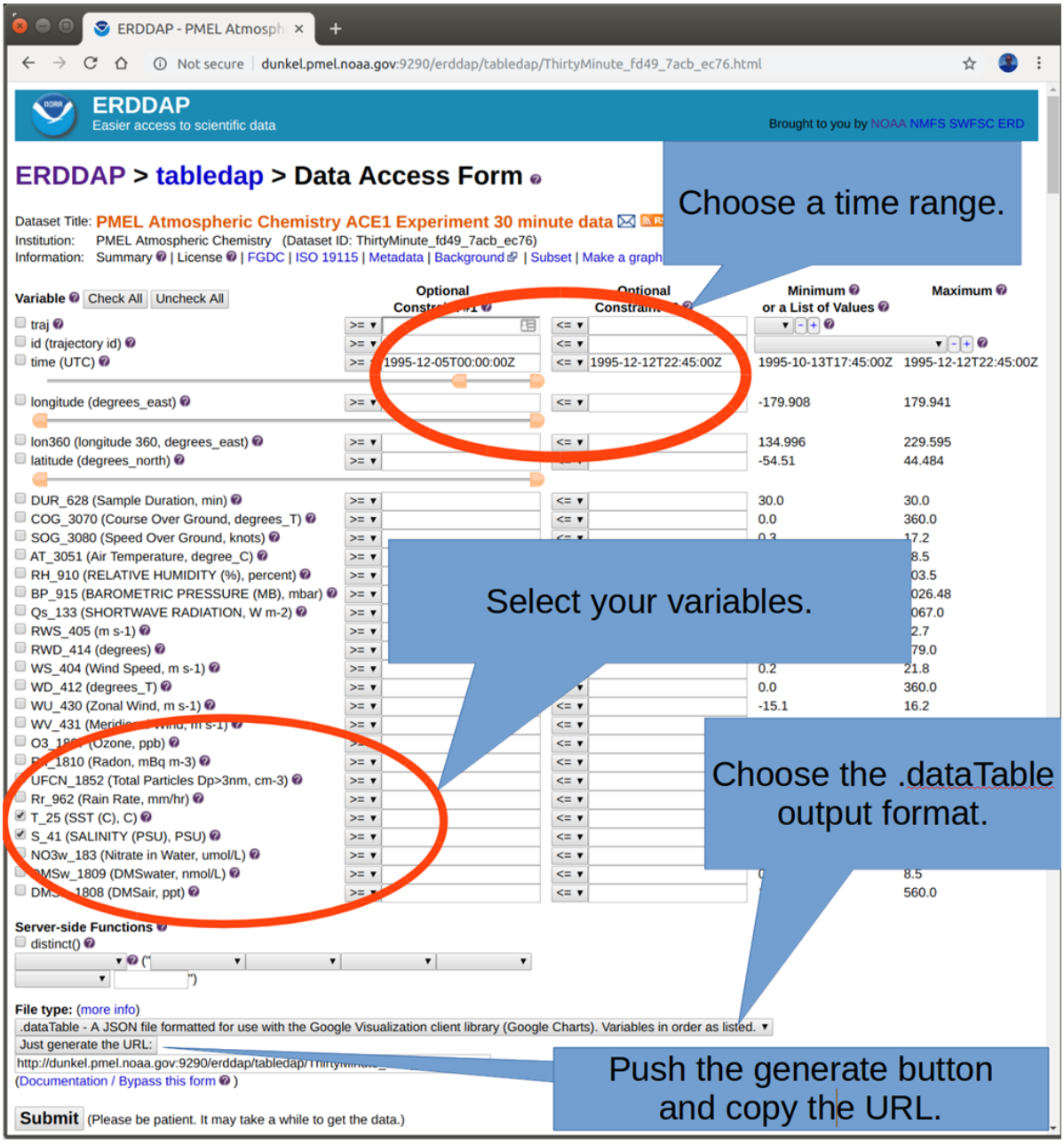
***Here is the full URL for the data download for this chart. Don't worry that it looks complicated and is encoded. ERDDAP takes care of all that for you in the user interface used to consttuct the URL. See the next panel for instructions on how to consturct the URL.

https://ferret.pmel.noaa.gov/engineering/erddap/tabledap/15min_w20_fdd7_a060.dataTable?latitude%2Clongitude&time%3E=2018-01-01T00%3A00%3A00Z&time%3C=2018-02-01T18%3A00%3A00Z (https://ferret.pmel.noaa.gov/engineering/erddap/tabledap/15min_w20_fdd7_a060.dataTable?latitude%2Clongitude&time%3E=2018-01-01T00%3A00%3A00Z&time%3C=2018-02-01T18%3A00%3A00Z)

USING THE ERDDAP INTERFACE TO CONSTRUCT A DATA URL

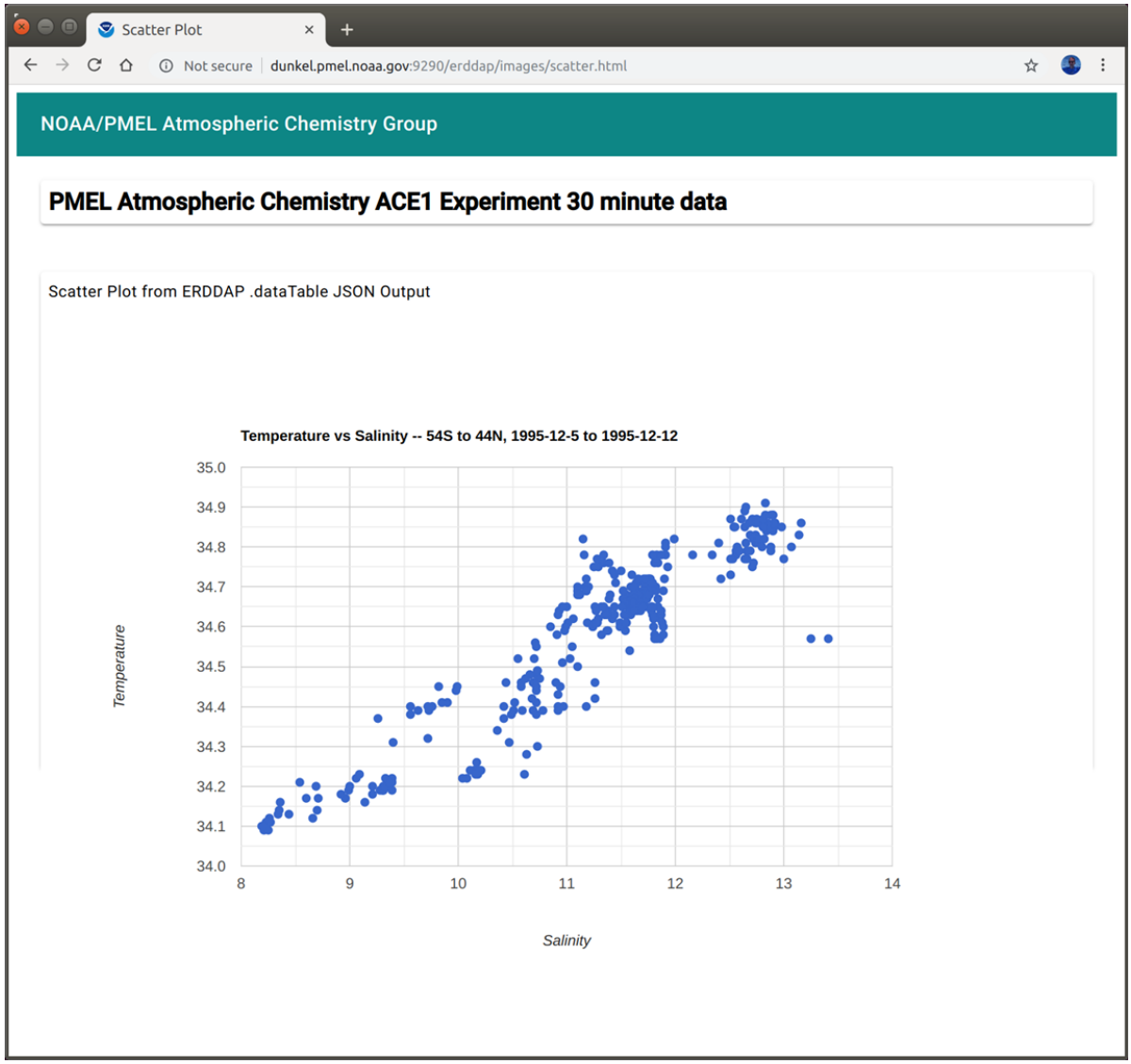
1. Navigate to the data set of interest.
2. Select the axis ranges.
3. Choose the variables you need for your plot. Time series plots must be ordered with time as the first variable. Map plots must have latitude and longitude as the first variables. (You can edit the order of the variables in URL if necessary).
4. Choose the .dataTable format.
5. Copy the URL.

For example:



Following the steps above produces the following URL:

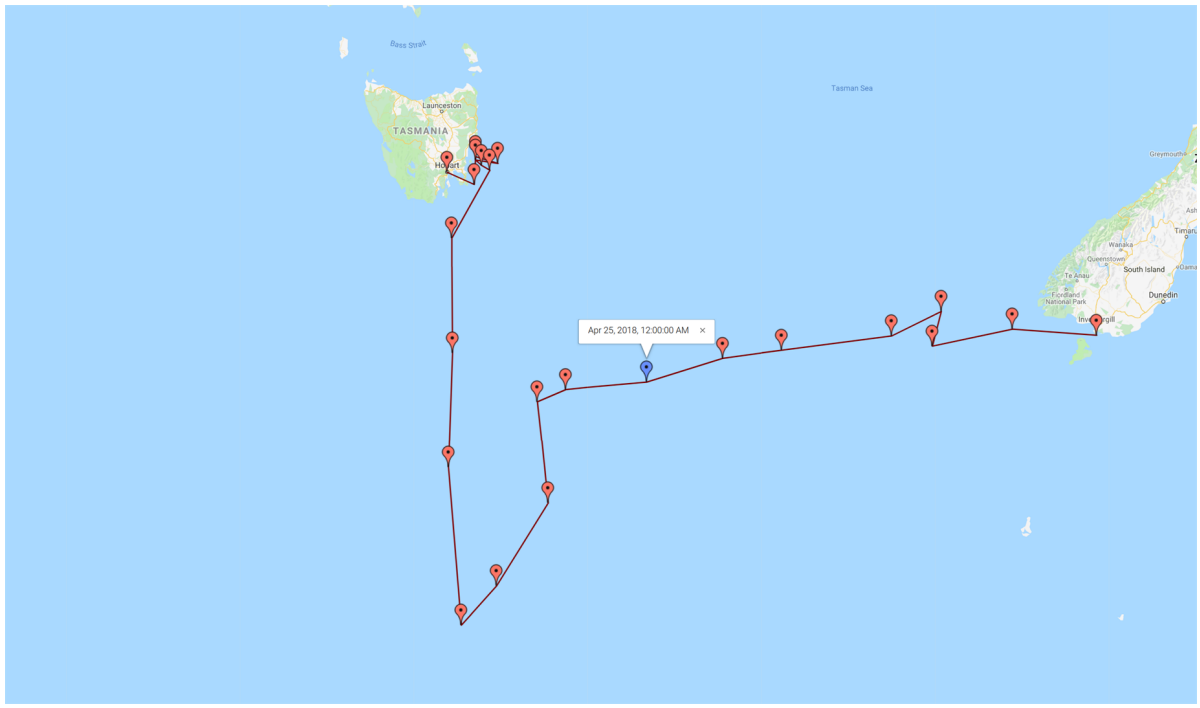
http://dunkel.pmel.noaa.gov:9290/erddap/tabledap/ThirtyMinute_fd49_7acb_ec76.dataTable?T_25%2CS_41&time%3E=1995-12-05T00%3A00%3A00Z&time%3C=1995-12-12T22%3A45%3A00Z
(http://dunkel.pmel.noaa.gov:9290/erddap/tabledap/ThirtyMinute_fd49_7acb_ec76.dataTable?T_25%2CS_41&time%3E=1995-12-05T00%3A00%3A00Z&time%3C=1995-12-12T22%3A45%3A00Z)



which can be pasted directly into a line of JavaScript code to download the data and produce a scatter plot like show below.

MOVING PLATFORMS

Time dependent location data can be drawn on a Charts Map using polylines and markers.



AUTHOR INFORMATION

Roland Schweitzer

ABSTRACT

ERDDAP is a data server developed at NOAA Southwest Fisheries Science Center that allows users to easily access and download subsets of gridded and tabular data. At the Scientific Data Integration Group at NOAA's Pacific Marine Environmental Laboratory we have focused on ERDDAP as the primary mechanism by which we serve all of our tabular data. Besides offering data access in a variety of useful formats, ERDDAP takes care of normalizing the data structure so that all tabular data sets have a simple row and column structure. Additionally, time values are normalized to have a consistent units across all data sets on the server.

The simplified tabular nature of the discrete data in ERDDAP lends itself nicely to use with the Google Visualization API (Google Charts). Charts is an API which renders nearly all of the graphics in the browser client. The basis of all charts is the Google Charts DataTable which is exactly analogous to a TableDap table in ERDDAP. Charts are interactive, visually pleasing, free and easy to use and so connecting ERDDAP data to Google Charts is a natural fit.

Even though ERDDAP has many output formats including several different JSON renderings, none fit exactly into the specified Charts data types. Initially, as a result, it was necessary to create custom code (either in the client or on a server between the client and the ERDDAP server) to manipulate the ERDDAP data stream to match the Charts API requirements. However, since Charts has a well-documented specification for a JSON representation of the required data type and because ERDDAP is an open source project, it was possible to extend ERDDAP to support the Charts DataTable as a new data type.

Using this new data type, it's possible to instantiate a Charts DataTable object directly from the ERDDAP URL using a simple Ajax call in the client which downloads the JSON DataTable representation. Because the newly added data type is now native to ERDDAP, it is possible to use default constraint syntax and ERDDAP RESTful services (using the ERDDAP user interface if desired) to build the URL for the exact data subset which is to be visualized by the desired google chart.

In this presentation, we'll explain what we've done to add the JSON data type to ERDDAP and provide many examples of client-side rendering of earth science and engineering data using Google Charts directly from ERDDAP URLs.