Using STOQS for Analysis and Visualization of Biological Oceanography Data
Michael P. McCann, Julio B. J. Harvey, Reiko Michisaki, Francisco Chavez, Robert C. Vrijenhoek
Monterey Bay Aquarium Research Institute, Moss Landing, CA

Abstract
Efficient analysis of growing types of oceanographic observations requires new approaches in data management. The Monterey Bay Aquarium Research Institute employs a two step harmonization process for diverse collections of in situ oceanographic measurements: 1) conversion of data to CF-NetCDF Discrete Sampling Geometry feature types and 2) loading data into Spatial Temporal Oceanographic Query System (STOQS) databases. STOQS includes a web-based graphical and command line user interface enabling data exploration across all dimensions of data collection, and command line interface for detailed analysis and visualization. The Python programming environment provides tools needed for direct access, graphical user interface enabling data exploration across all dimensions of data collection, and command line interface.

Efficient analysis of growing types of oceanographic observations requires new approaches in data management. The Monterey Bay Aquarium Research Institute employs a two step harmonization process for diverse collections of in situ oceanographic measurements: 1) conversion of data to CF-NetCDF Discrete Sampling Geometry feature types and 2) loading data into Spatial Temporal Oceanographic Query System (STOQS) databases. STOQS includes a web-based graphical and command line user interface enabling data exploration across all dimensions of data collection, and command line interface for detailed analysis and visualization. The Python programming environment provides tools needed for direct access, graphical user interface enabling data exploration across all dimensions of data collection, and command line interface.

About STOQS
- PostgreSQL/PostGIS database, Mapserver, and Python-Django running on a server and client-side software (jQuery, OpenLayers, Twitter Bootstrap) in a web browser.
- Faceted search capabilities allow a user to drill into data of interest. Selection can be constrained by spatial, temporal, depth, parameter, and platform.
- Users can write Python scripts to perform detailed analysis and visualization from the database. This poster presents a few such examples.

Data Exploration Workflow
1. STOQS User Interface
   a. Begin with an overview of all the data (Figure 2).
   b. Use a strided database for quick exploration. This campaign contains data from 27 platforms, 77 parameters, and 25 million measurements.
   c. Biologists want to see correlations between chlorophyll fluorescence and optical backscatter, and their spatial temporal distributions.
   d. We can explore each platform (Figure 3) and produce graphs of the regressions, location/track of the platform, and time-depth contours of the data.

2. Detailed Analyses and Visualization
   a. STOQS can help the user extract the data. Figure 4 shows the code used to extract the data for the platforms, times, parameters, and depths, currently selected.
   b. Using the code as a template (Figure 4) the user can create a program to step through the database to extract data to make a movie of the correlations over time and space such that we can gain more insight from the data.

3. Sharing and Contributing
   a. The STOQS code repository contains programs and software that you may be able to adapt to suit your specific needs, such as trajectory_biplots.py (Figure 5 & 6).
   b. The STOQS code repository (https://code.google.com/p/stoqs) contains instructions for cloning the software and setting up a local development environment.

Command line tool
After initial exploration a visualization tool is developed in Python. This program is checked into the STOQS contrib/analysis directory and is called trajectory_biplots.py.

The trajectory_biplots.py program creates a sequence of jpg images which can be stitched together to make a movie. A frame of which is shown at right in Figure 5. A cursor moves along the time line as the vehicle track lines and fluorescence/backscatter correlations update — replaying the data collected from 4 platforms during the campaign.

Separation of concerns and code reuse
Python provides object oriented programming features that encourage reusability of software development. The PlatformsBiPlot class in the trajectory_biplots.py script inherits methods from a abstract base class. This fixes the visualization tool developer from concerns over database interaction details.

Another use case is to visualize scatter plots between all parameters. A script can be written (crossproduct_biplots.py) inheriting methods from BiPlot, reusing the database access code originally developed for trajectory_biplots.py. This saves a great deal of time and effort.

Sample analysis data exploration
The stoqs_simz_aug2013 campaign produced 614,193 in situ electronic measurements and 972 ex situ sampled measurements from 4 platforms over a period of a week. The STOQS UI can be used to explore the data. Exploration of correlations between the 34 Parameters yields 1156 possible plots. We can use the crossproduct_biplots.py program to do the work for us. Figure 7 shows one page of output.

Conclusions between in situ (Measured) and ex situ (Sampled) measurements are automatically created, as these are simply different groupings of Parameters in the STOQS data model.

Summary
The STOQS platform solves a longstanding problem in the management and exploration of upper water column measurement and water sampled data. It provides an easy to use web-based interface for data exploration and a programming interface for more detailed and specialized analysis and visualization.

Acknowledgments
Acknowledgments are given to the numerous individuals involved in the deployment, recovery, operation and data handling for all of the platforms whose data are presented here. Development of STOQS is supported by the David and Lucile Packard Foundation at the Monterey Bay Aquarium Research Institute. It was inspired by the M2OQuat tool originally developed by Mike Godin (M. A. Godin and J. G. Bellingham, “The Metadata Oriented Query Assistant (MOQuA): A Web Tool for Finding Data in Heterogeneous, Multi-Platform Data Collections by Simultaneously Filtering on Multiple Metadata Hierarchies,”ACU 2005 Fall meeting. Abstract N2181-1176, San Francisco, CA, U.S.A., December 2005.)

STOQS is an open source software project built upon a framework of free and open source software and is available for anyone to use and extend. For more information please see the STOQS code repository at http://code.google.com/p/stoqs.