Biofloat: Python Software for Bio-Argo Float Data

Mike McCann, Carole Sakamoto, Josh Plant, and Ken Johnson
mccann@mbari.org, saca@mbari.org, jplant@mbari.org, johnson@mbari.org
Monterey Bay Aquarium Research Institute

The Problem

Analyzing Bio-Argo profiling data from archive centers can be an onerous task. One must scan catalog lists for data assembly centers, then scan those sites for the NetCDF files and then read the data from many individual files before being able to perform any analysis or visualization. These difficulties hamper usefulness of the archive and limits access to only those familiar with NetCDF files and the ARGO data architecture.

A Solution

The Python programming language is gaining popularity as a tool for analyzing large collections of data. The Scientific Python ecosystem (consisting of packages such as Numpy, Pandas, SciPy, Matplotlib, Xarray, and Jupyter) provides similar capabilites to commercial products such as Matlab, yet is free and open source. To solve the problem of using data from the archive we developed the Python module "biofloat" that handles all the details of scanning catalogs, finding the files to load and then creating local caches of data. The biofloat module is free and open source and is available on GitHub at **https://github.com/biofloat/biofloat**. It can be installed from the Python Package Index (PyPI) with the command "pip install biofloat". The biofloat module provides access to Bio-Argo data using simple Python method calls and delivers it in easy-to-use Pandas dataframes. The GitHub site has several Jupyter Notebooks which demonstrate using biofloat for various analyses and visulizations.

In [7]:
```
msdf = sdf.groupby(['wmo', 'year', 'month']).mean()

from biofloat.utils import o2sat, convert_to_ml1
msdf['o2sat'] = 100 * (msdf.DOXY_ADJUSTED / o2sat(msdf.PSAL_ADJUSTED, msdf.TEMP_ADJUSTED))
msdf.head(10)
```

Out[7]:

			TEMP_ADJUSTED	PSAL_ADJUSTED	DOXY_ADJUSTED	lon	lat	o2sat
wmo	year	month						
1900650	2006	6	28.360667	35.855661	205.710002	-34.276000	3.892000	105.919919
		7	28.212667	35.563439	207.386667	-33.333750	4.579750	106.336035
		8	28.060556	35.031504	208.171110	-31.027000	4.595000	106.122748
		9	28.284222	34.886179	207.644444	-30.321667	5.486667	106.136009
		10	28.738000	34.431447	207.946665	-28.316667	6.555333	106.754944
		11	28.449334	35.208669	207.763334	-26.357667	6.368000	106.692811
		12	27.748222	35.490718	209.246665	-25.575333	6.333000	106.446434
	2007	1	27.297334	35.559545	209.997779	-26.867000	6.146000	106.101382
		2	26.830889	35.678819	211.642220	-28.320000	6.723333	106.206474
		3	26.726000	35.816099	212.120000	-28.167333	7.562667	106.357337

Fig. 1 – Jupyter Notebook showing processing of Bio-Argo data.

Oxygen Calibration Use Case

Takeshita *et al*. (2013) present a climatology based quality control procedure for dissolved oxygen calibration utilizing the World Ocean Atlas 2009 (WOA09). We implement a similar procedure in Python and fully document it in the Jupyter Notebook calibrate_all_oxygen_floats.ipynb, which is available on the GitHub site. Figure 1 demonstrates listing the monthly binned surface float measurements in preparation for the WOA09 lookup. Figure 2 shows surface oxygen saturation values for a single float compared with climatological values for the location and times of the float's trajectory. The ratio of these values is a gain factor that may be used to calibrate the float's oxygen profile. Figure 3 shows the calculated oxygen calibration gain factors for 287 floats in the Bio-Argo archive. Biofloat is free to use for these kind of analyses, and source code contributions are welcomed to have it meet additional needs of the Bio-Argo user community.

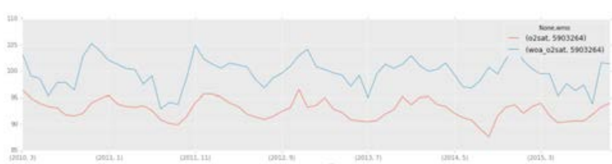

Fig. 2 – Over 5 years of surface oxygen saturation measured by Bio-Argo float 5903264 (blue) compared with oxygen saturation from the World Ocean Atas (red). The biofloat Python software enabled easy generation of this plot in a Jupyter Notebook.

Presented at International Conference on Marine Data and Information Systems – Gdańsk, Poland October 11-13, 2016

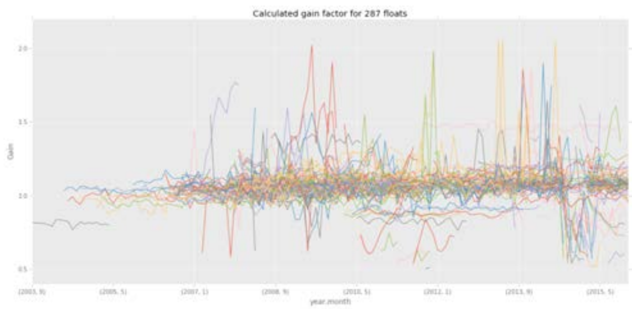

Fig. 3 – Over 10 years of oxygen calibration gain values for 287 Bio-Argo floats (plot created in Jupyter Notebook using the biofloat Python module).

Plot the distribution of gains from this minimally qc'ed data — comparible to Figure 4 in Takeshita et al. (2013).

In [5]:
```
%pylab inline
import pylab as plt
plt.rcParams['figure.figsize'] = (18.0, 4.0)
plt.style.use('ggplot')
ax = qdf.groupby('wmo').mean().gain.hist(bins=100)
ax.set_xlabel('Gain')
ax.set_ylabel('Count')
floats = qdf.index.get_level_values('wmo').unique()
ax.set_title(('Distribution of WOA calibrated gains from {} floats').format(len(floats)))
```

```
Populating the interactive namespace from numpy and matplotlib
```

Out[5]:
```
<matplotlib.text.Text at 0x7ffc8a9812d0>
```

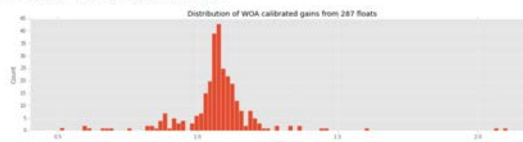

Fig. 4 – Histogram of over 10 years of oxygen calibration gain values for 287 Bio-Argo floats along with the Python code that created it in the Jupyter Notebook that is available on https://github.com/biofloat/biofloat.

In [13]:
```
from mpl_toolkits.basemap import Basemap
m = Basemap(llcrnrlon=15, llcrnrlat=-90, urcrnrlon=390, urcrnrlat=90, projection='cyl')
m.fillcontinents(color='0.8')
dflm = df1.groupby(level=['wmo','lon','lat']).mean()
for wmo, c in zip(wmo_list, colors):
    try:
        lons = dflm.xs(wmo, level='wmo').index.get_level_values('lon')
        lats = dflm.xs(wmo, level='wmo').index.get_level_values('lat')
        try:
            m.scatter(lons, lats, latlon=True, color=c)
        except IndexError:
            # Some floats have too few points
            pass
        lon, lat = lons[0], lats[0]
        if lon < 0:
            lon += 360
        plt.text(lon, lat, wmo)
    except KeyError:
        pass
```

Fig. 5 – Map of the locations of the last 4 profiles of each float with DOXY_ADJUSTED oxygen data.

Looking Ahead

This poster introduces an open source Python module that simplifies the tasks required for working with archived Bio-Argo float data. The code is hosted on GitHub and serves as a community resource for anyone wishing to analyze and visualize Bio-Argo float data using Python. We welcome any ideas or code contributions to help this software meet the needs of the Bio-Argo community.

References

Takeshita, Y., Martz, T.R., Johnson, K.S., Plant, J.N., Gilbert, D., Riser, S.C., Neill, C., and Tilbrook, B. 2013 A Climatology-Based Quality Control Procedure for Profiling Float Oxygen Data. Journal of Geophysical Research-Oceans, 118, 5640-5650. doi: 10.1002/jgrc.20399

Acknowledgments

Development of biofloat is supported by the David and Lucile Packard Foundation at the Monterey Bay Aquarium Research Institute.