OneArgoPy

Creating a Python Library for Accessing, Analyzing, and Plotting Argo Float Data

Savannah Stephenson
Texas A&M University Corpus Christi
Oceanic and Atmospheric Research(OAR)/Pacific Marine Environmental Laboratory(PMEL)
Hartmut Frenzel



Outline

- Background
- Objectives
- Methodology
- Results
- Summary
- Next Steps
- Acknowledgments
- Questions

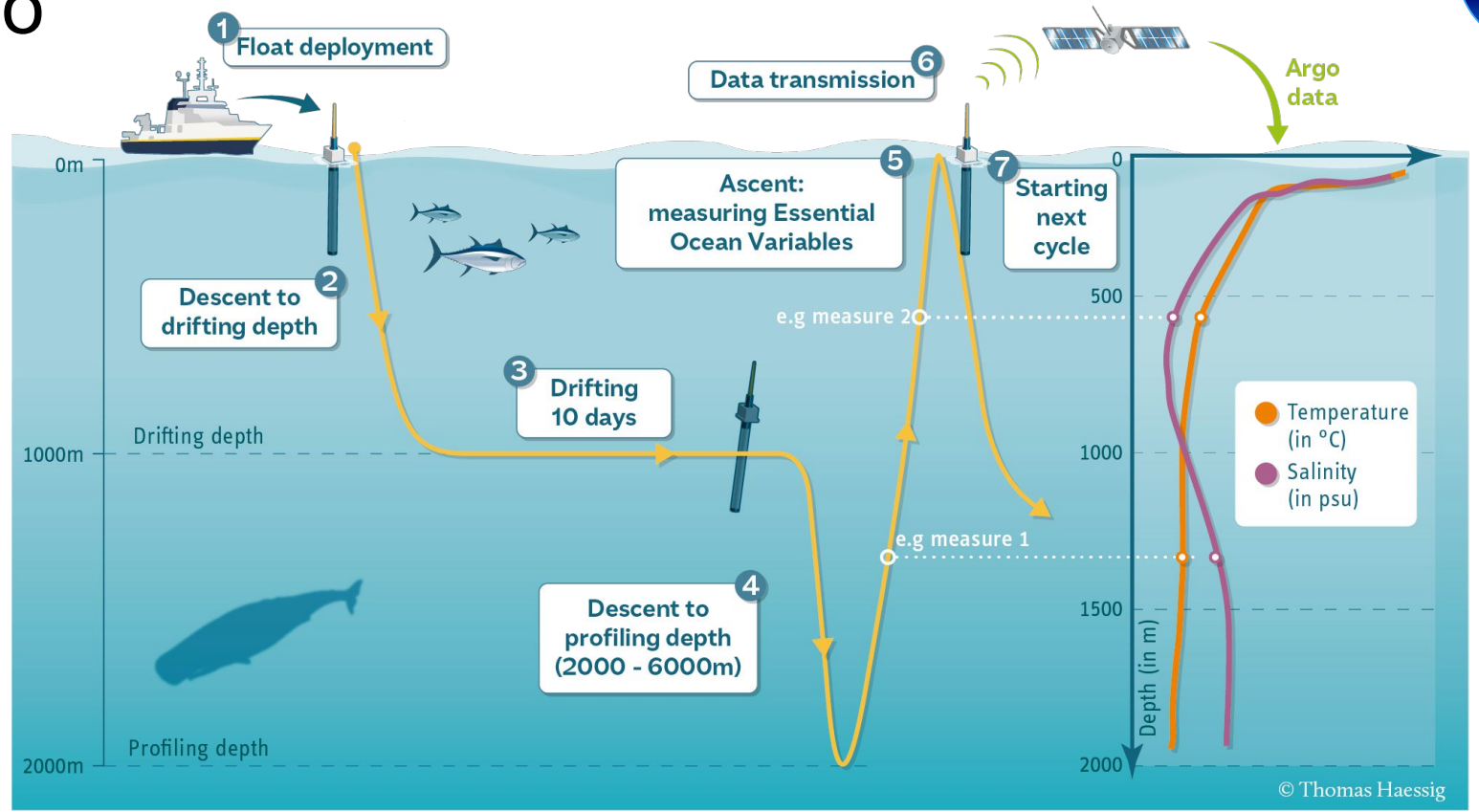
Background

Argo

- International program
- Measures water properties
- Collected with robotic instruments



Argo

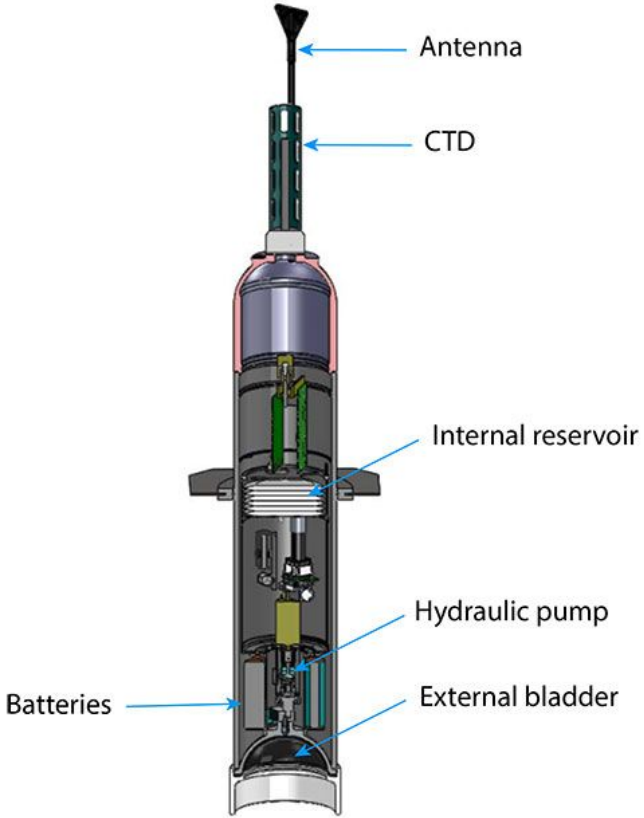

1 Float deployment
2 Descent to drifting depth
3 Drifting 10 days
4 Descent to profiling depth (2000 - 6000m)
5 Ascent: measuring Essential Ocean Variables
6 Data transmission
7 Starting next cycle
Argo data
0m
Drifting depth
1000m
Profiling depth
2000m
e.g measure 2
e.g measure 1
0
500
1000
1500
2000
Depth (in m)
Temperature (in °C)
Salinity (in psu)
© Thomas Haessig

Argo

Core Floats

Biogeochemical (BGC) Floats

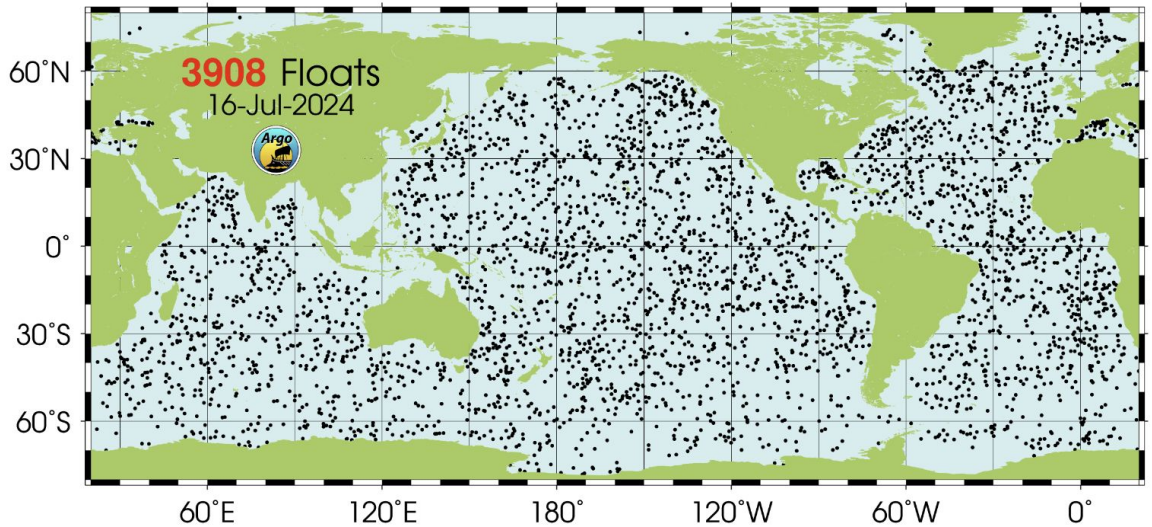
Deep Floats



COURTESY OF MICHAEL MCCLUNE AT SCRIPPS INSTITUTION OF OCEANOGRAPHY

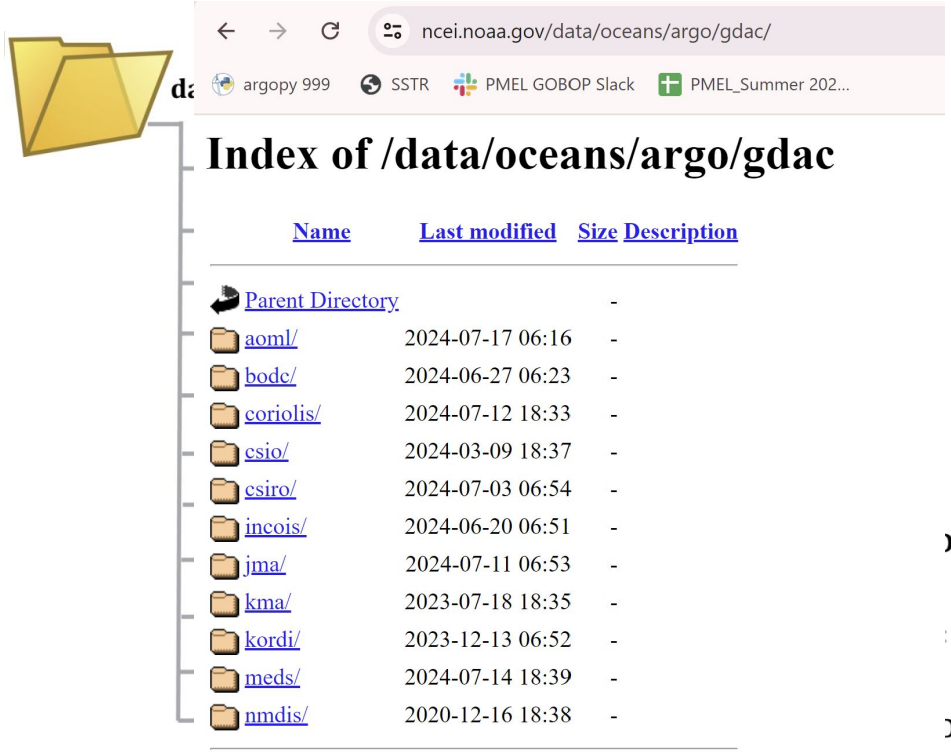
Argo

- Argo data are FREE to anyone wishing to use it.
- The data can be obtained from two global data servers, one in France and one in the USA.



https://argo.ucsd.edu/about/status/

Global Data Assembly Center (GDAC)



- GDAC file organization by Data Assembly Center (DAC) and WMO IDs makes it hard to find float data by region or by time period.
- Difficult to access and work with data

Existing Tools: OneArgo-Mat

Pros:

- Designed to be maximally efficient
- Up-to-date data
- Downloads data (GDAC)

NOAA-PMEL / **OneArgo-Mat**

Cons:

- Matlab Licensing
  - especially outside of universities



Existing Tools: argopy

Pros:

- Python module
- Wide functionality
- Created by argo data users

Cons:

- ERDDAP server
- Alpha stage after 8 years
- Internet connection required

Objectives

Python Libraries

Module

Package

Library

```
pip install datetime
```

```
conda install numpy
```

```
from pathlib import Path
import requests
from datetime import datetime, timedelta
import shutil
import gzip
import numpy as np
import matplotlib.path as mpltPath
```

oneargopy

Python Library

GDAC

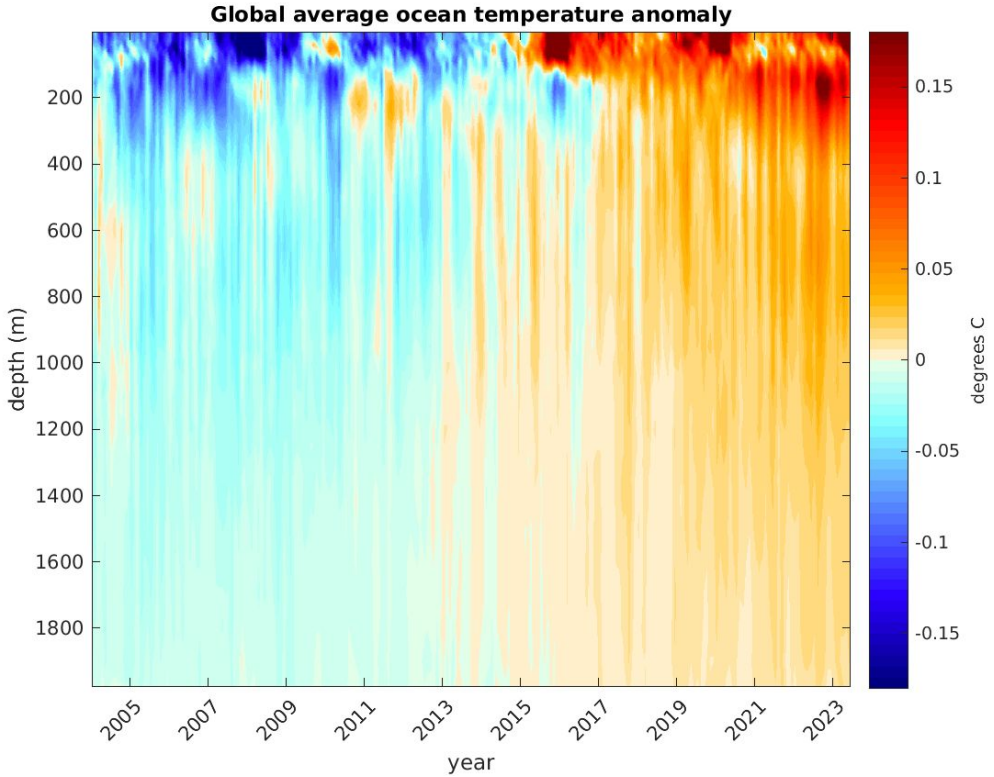


Pros without Cons

```
pip install oneargopy
```

```
conda install -c conda-forge oneargopy
```

Relevance



https://argo.ucsd.edu/about/status/

Project Interest

Coding a Library

Graphing

Bonus:

Most Organized Interview

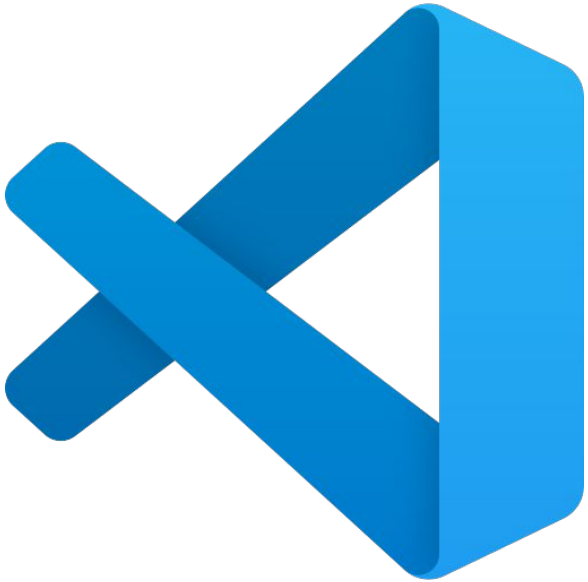
Dataframes

Pandas

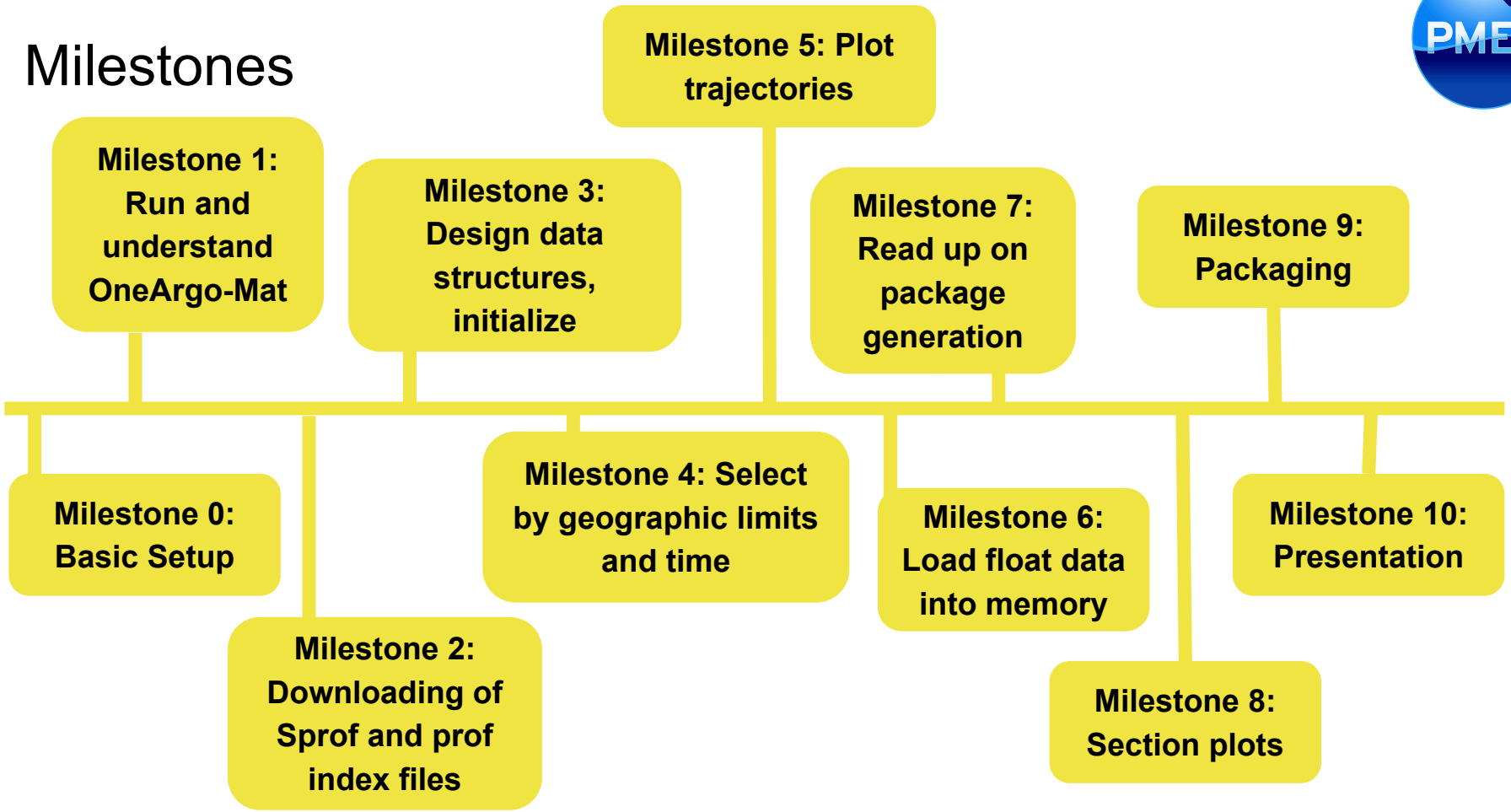
Methodology

Development Tools





Milestones


Milestone 5: Plot trajectories
Milestone 1: Run and understand OneArgo-Mat
Milestone 3: Design data structures, initialize
Milestone 7: Read up on package generation
Milestone 9: Packaging
Milestone 0: Basic Setup
Milestone 2: Downloading of Sprof and prof index files
Milestone 4: Select by geographic limits and time
Milestone 6: Load float data into memory
Milestone 8: Section plots
Milestone 10: Presentation

Results

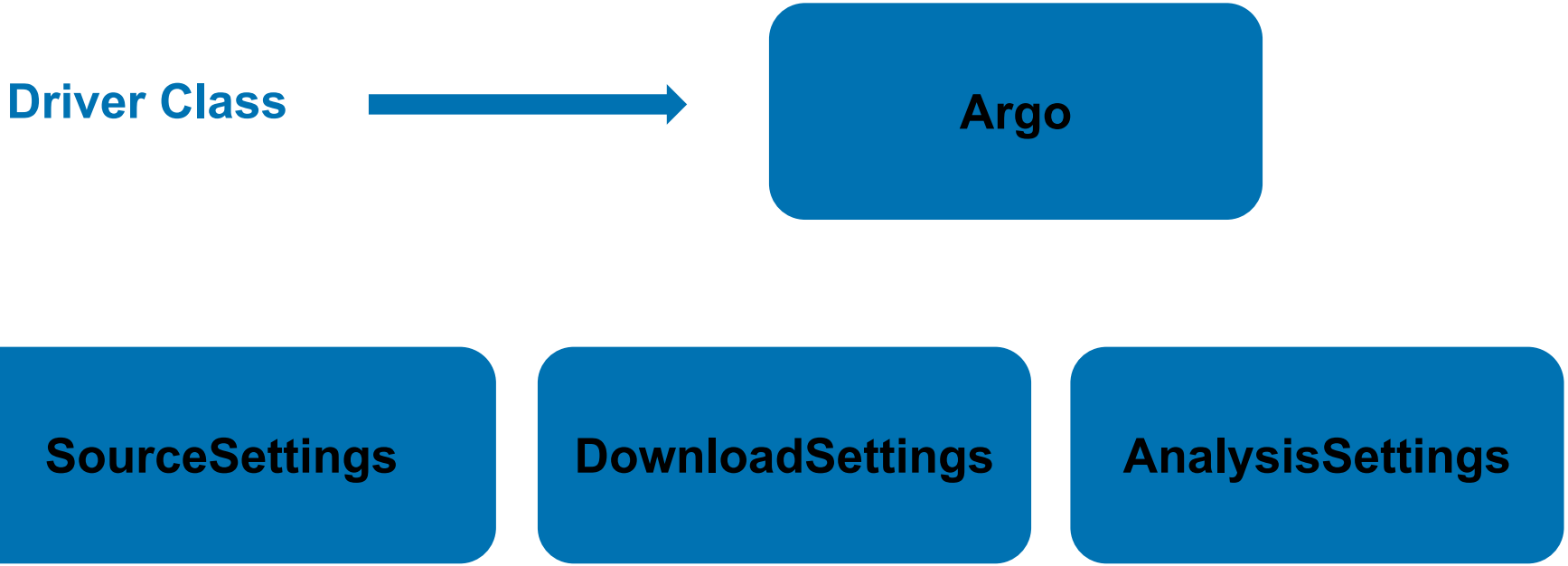
Pip and Conda Packaging

```
pip install oneargopy
```

```
conda install -c
conda-forge oneargopy
```

Instructions on Packaging Written and Saved for Future Versions

Classes in the oneargopy Library:



Public Argo Functions:

Argo() constructor

select_profiles()

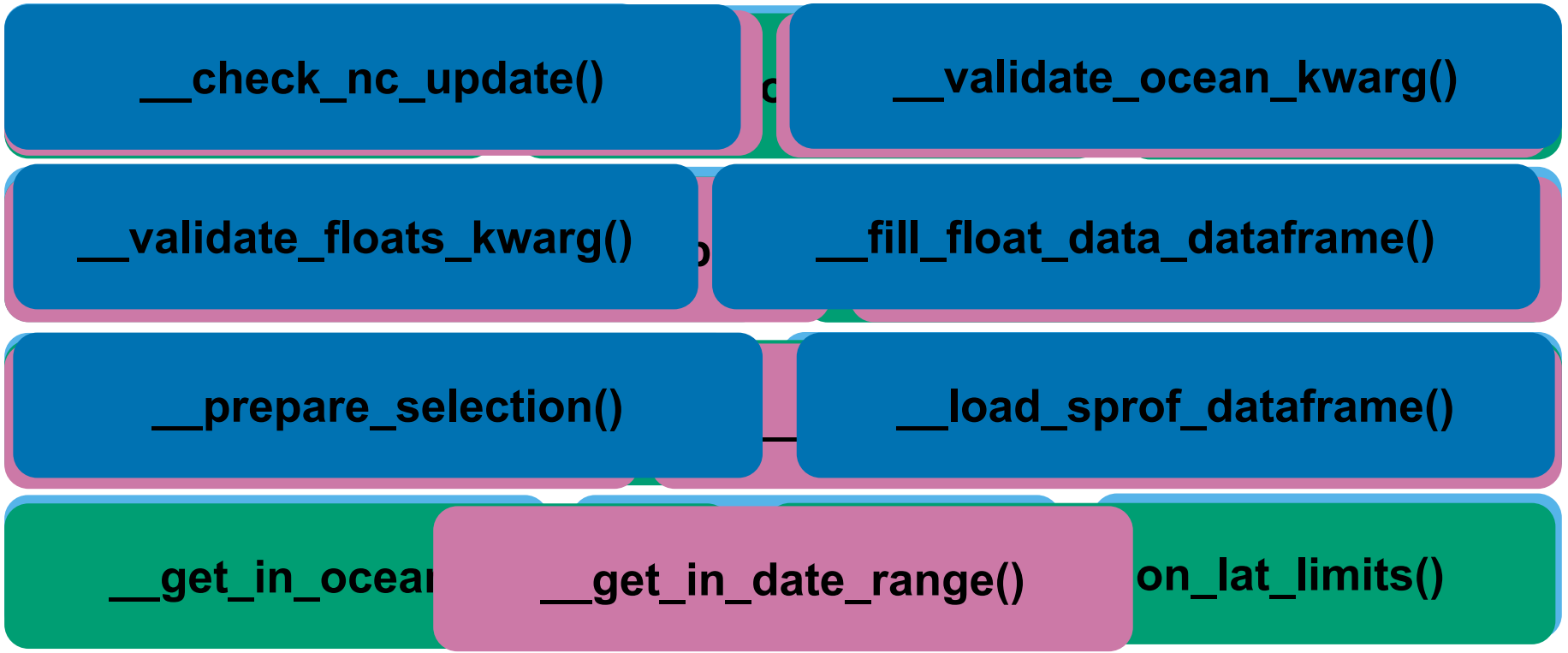
trajectories()

load_float_data()

sections()

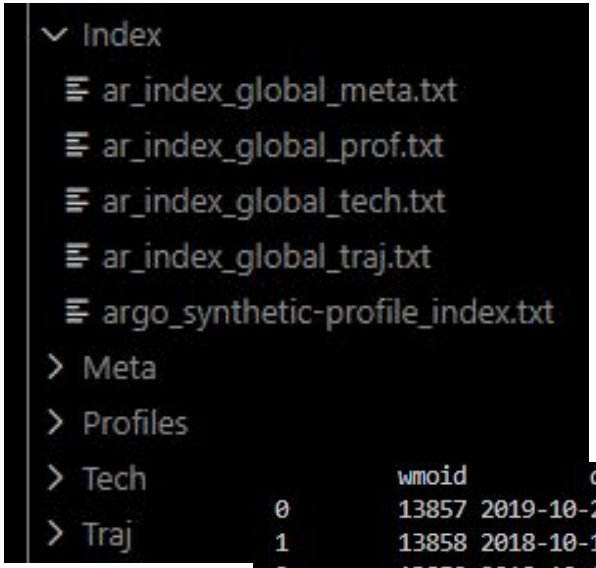


Private Argo Functions:



Argo()

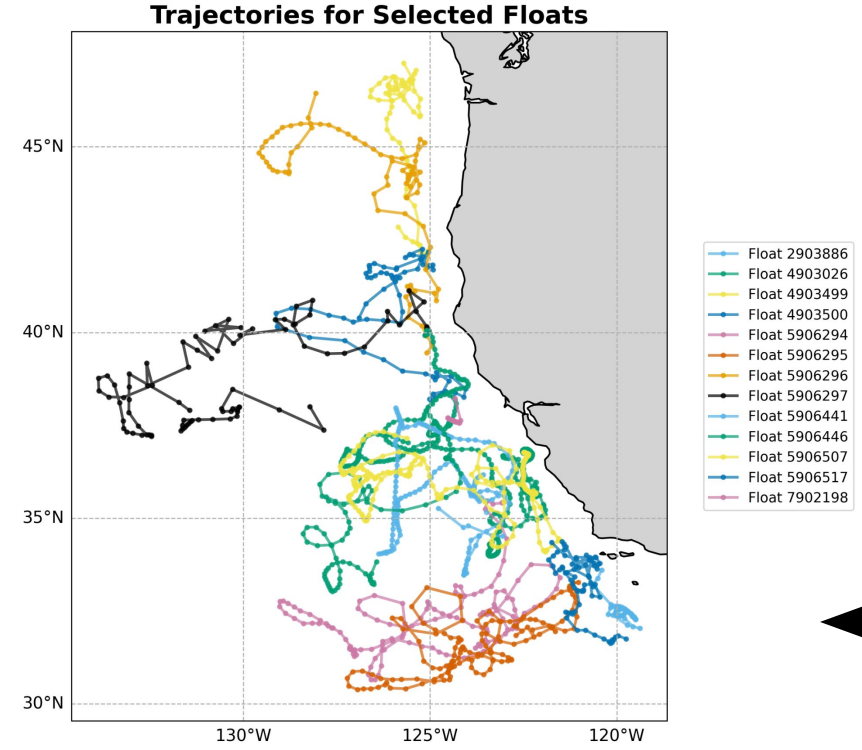
```
from Argo import Argo
argo = Argo()
```



	wmoid	date_update	is_bgc
0	13857	2019-10-24 21:15:17	False
1	13858	2018-10-11 18:06:09	False
2	13859	2018-10-11 18:09:43	False
3	15819	2019-10-24 21:15:20	False
4	15820	2019-11-18 22:15:25	False
...	...	...	...
2251	7902196	2024-07-12 11:15:14	True
2252	7902197	2024-07-15 08:13:59	True
2253	7902198	2024-07-03 22:13:57	True
16271	7902201	2024-07-07 10:00:40	False
2254	7902203	2024-07-14 14:14:47	True

[18527 rows x 3 columns]

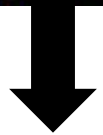
trajectories()




```
from Argo import Argo
argo = Argo()
profiles = argo.select_profiles(lon_lim=[-127,-115],
                                lat_lim=[32.5,45.5],
                                start_date='2021-09-01',
                                type='bgc')
argo.trajectories(list(profiles))
```

load_float_data()

```
from Argo import Argo
argo = Argo()
profiles = argo.select_profiles(lon_lin=[-127, -115], lat_lim=[32.5, 45.5], start_date='2023-09-01', type='bgc')
float_data_dataframe = argo.load_float_data(profiles,    riables='DOXY')
```

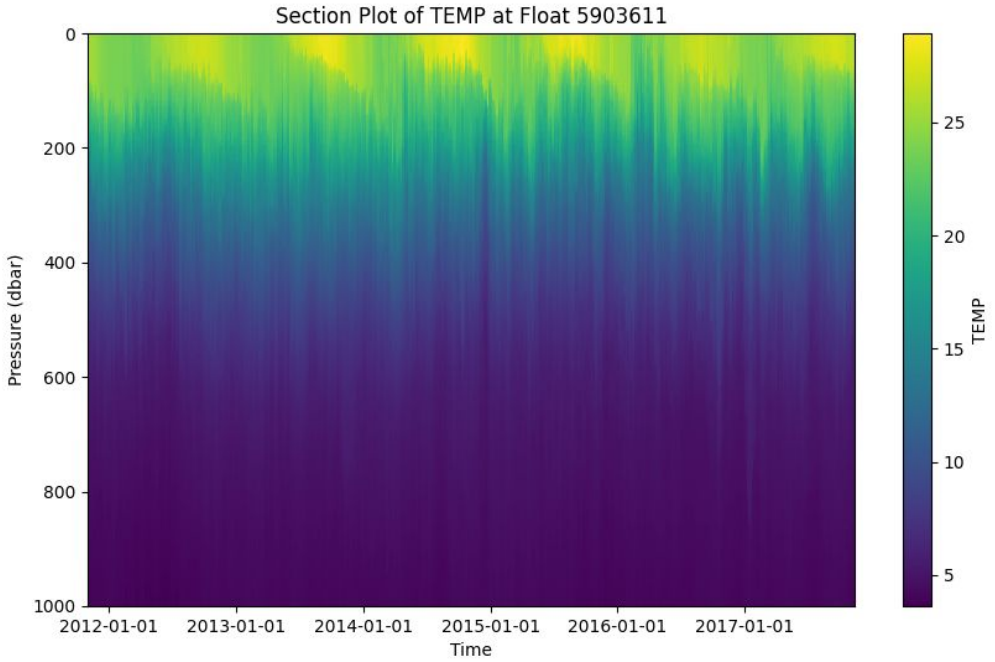


	WMOID	CYCLE_NUMBER	DIRECTION	DATE	DATE_QC	...	DOXY	DOXY_QC	DOXY_ADJUSTED	DOXY_ADJUSTED_QC	DOXY_ADJUSTED_ERROR
0	1902646	1	A	2023-10-30 10:42:47.001134	1	...	NaN	0.0	NaN	0.0	NaN
1	1902646	1	A	2023-10-30 10:42:47.001134	1	...	NaN	0.0	NaN	0.0	NaN
2	1902646	1	A	2023-10-30 10:42:47.001134	1	...	205.278458	3.0	213.900146	1.0	2.048767
3	1902646	1	A	2023-10-30 10:42:47.001134	1	...	NaN	0.0	NaN	0.0	NaN
4	1902646	1	A	2023-10-30 10:42:47.001134	1	...	NaN	0.0	NaN	0.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
1131415	7900875	24	A	2024-07-16 08:00:01.000000	1	...	116.637901	3.0	117.466034	1.0	10.909296
1131416	7900875	24	A	2024-07-16 08:00:01.000000	1	...	116.474602	3.0	117.301575	1.0	10.904677
1131417	7900875	24	A	2024-07-16 08:00:01.000000	1	...	116.485298	3.0	117.312347	1.0	10.899818
1131418	7900875	24	A	2024-07-16 08:00:01.000000	1	...	116.497101	3.0	117.324234	1.0	10.893674
1131419	7900875	24	A	2024-07-16 08:00:01.000000	1	...	116.448799	3.0	117.275589	1.0	10.888726

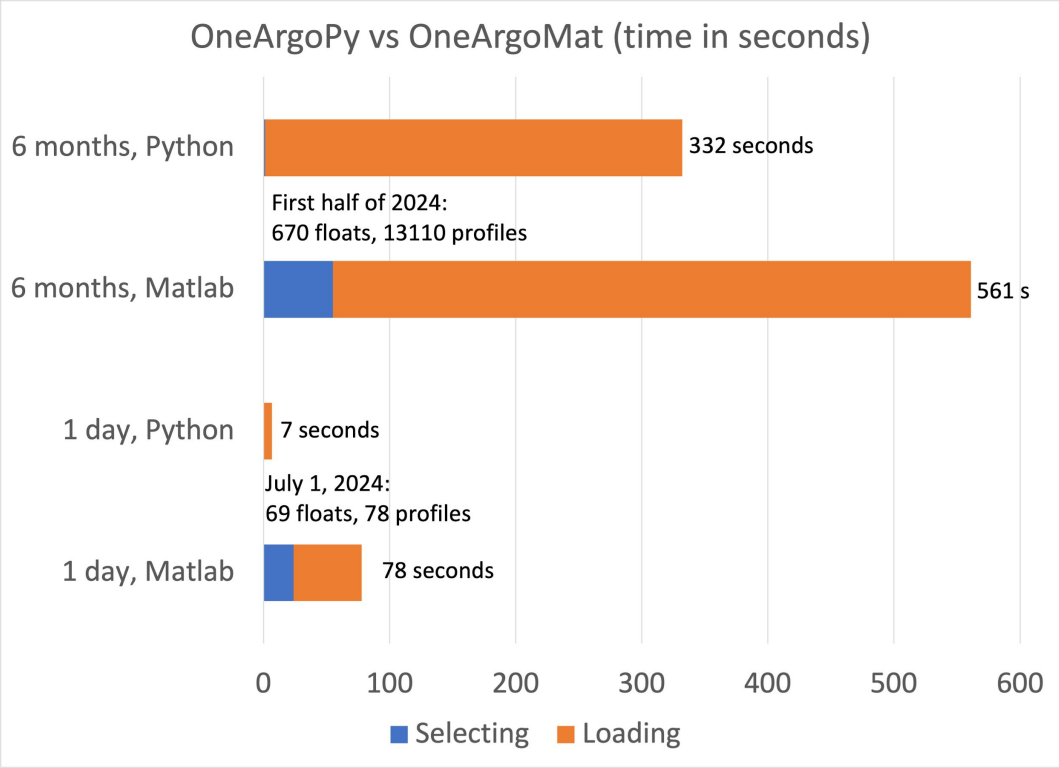
[1131420 rows x 18 columns]

sections()

```
from Argo import Argo
argo = Argo()
data = argo.load_float_data([5903611],
                            variables='TEMP')
argo.sections(data, 'TEMP')
```



Compared to Matlab Toolbox



Summary

Summary

- Made and packaged the first version of oneargopy.
  - Uncomplicated function calls
  - Well documented functions
  - Flexibility of function inputs

Github Repository



Next Steps

Long Term: Library Expansion

Expand Filtering Options of select_profiles()

More Public Functions

Optimization

Acknowledgements

Acknowledgements

Thank you to my funding sponsor the EPP/MSI Undergraduate Scholarship Program.

Thank you to Hartmut Frenzel for guiding me throughout the project and having patience with me when I got stuck on things.

Thank you to the GO-BOP team for being welcoming and kind in meetings.

Questions?

