GSOC 2022 Project Proposal at IOOS

Making ocean biodiversity data easily accessible with python (pyobis revamp)

Ayush Anand
National Institute of Technology Durgapur
India

Bachelor of Technology in Computer Science and Engineering

kingwarrior880@gmail.com

aa.21u10425@btech.nitdgp.ac.in

Apr 12, 2022

Summary of the Proposal

Pyobis is an interesting python package that helps users fetch data from OBIS API which holds a great amount of ocean open-data, with ease. This project is intended to update the existing pyobis python package to use the new OBIS API v3 and ensure the package is used for product generation in the future.
The pyobis package is really powerful and can fetch huge records of marine species, particularly in the indo-pacific ocean region, and regions near the US, UK, and Australian coastline. It is interesting to note that OBIS also holds data for species even dating back to around 1078 AD, which makes pyobis even more essential to be maintained.

Mentors

Mentors (and GitHub handles**)**:
Tylar Murray (@7yl4r), Filipe Fernandes (@ocefpaf), Mathew Biddle (@MathewBiddle)

Table of Contents

1 Introduction and Motivation

Why me?

I have previous experience in building and managing REST APIs, Data & ML pipelines, and Python programming. I love playing with data and bringing out interesting visualizations and hidden insights using libraries like Pandas, Matplotlib, and Seaborn. You can visit my Kaggle profile: https://kaggle.com/theayushanand (I am a Kaggle Notebooks Expert) and have a look over my previous projects.

I also have prior experience working with marine/underwater data, with the project **FasterRCNN model on Underwater Image Data from the Great Barrier Reef** which involved pre-processing underwater images and identifying COTS in the Coral Reef using deep learning. I have also successfully contributed to the pyobis package to integrate existing modules with the new v3 API (PR #8, Status: merged). **Above this, I have a deep desire to work for marine biodiversity and IOOS has brought me the opportunity to contribute to marine ecosystem conservation, which has made me so excited for this project.** Even my final high school project for CS was "A study on the effects of climate change on Agriculture", which involved grabbing data from AgroSphere (a NASA project), cleaning it, bringing out insights, and visualizing. I am naturally inclined toward Climate science projects.

I am also excited about the new things I'll learn every day, which will make my journey much more beautiful. Even after the GSoC period ends, I'll still be contributing to the pyobis package because it aligns perfectly with my passion. This is because for me open source is different from GSoC, I had in the past contributed to AOSSIE (although only to a little extent, since I was in my high school at that time - this was not GSoC). Open Source is a community where we learn, and help others learn and IOOS is giving me such a beautiful opportunity. Additionally, my passion and the project's objective are absolutely the same - that of marine conservation, moments like these come very rarely. So, it's a natural choice for me to pick this project.

2 Goals and Objectives (deliverables)

The following objectives are enlisted with the project allocation.

- Update pyobis to use the new API.
- Make mof response more efficient.
- Create tutorial usage of each module.
- Clarify listing of GBIF instead of OBIS in package description ref.
- Increase test coverage.
- Use CI push to PyPI.
- Merge in URL changes done on the forks.
- Create a Jupyter notebook with a demo to analyze/visualize data grabbed using this package.

Extended deliverables

- Initiate a Contributor-guidelines.MD for prospective contributors which the current repository is lacking The data OBIS hosts is invaluable, and so increasing the reach to potential contributors will make it more powerful.
- Developing an umbrella module inside the pyobis package that can be used directly by the users to visualize data through existing modules. For eg., we can build a top-level module, say pyobis.visualize which ingests JSON data from say pyobis.occurrence and visualizes it based on some parameters like scientificname, basisOfRecords, etc.

  Eg. Syntax:
  >>> import pyobis.visualize as viz
  >>> viz.occ_mapPlot(scientificname=value, orthographic=1, **kwargs)) *# returns occurrence records plotted on world map, implementation here.*

3 Methods (deliverables)

The goals and objectives for the project can be worked upon as:

➔ **Modifying API calls in the pyobis package**: We will need to modify the obis_baseurl at pyobis/pyobis/obisutils.py as well as modify GET requests in all functions of modules. Some packages are missing in the existing pyobis package which is present in the new v3 API, we will also need to create modules for them.

pyobis.module_name	This is the reference table for below.
OBIS_API/resource	module_name.function() #corresponding function

Changes I have already done (existing PR #8 and #11):

pyobis.checklist

checklist/	checklist.list(scientificname, **params, **kwargs)
checklist/redlist	checklist.redlist(scientificname,**params, **kwargs)
checklist/newest	checklist.newest(scientificname, **param, **kwargs)

pyobis.dataset ***It has been renamed from pyobis.resource***

dataset/	dataset.search(scientificname,**params, **kwargs)
dataset/{id}/	dataset.get(id, **kwargs). It requires only ID

All other functions existing previously in the *resources* module have been deleted, and it is renamed to the *dataset* module.

pyobis.nodes

node/{id}	nodes.search(id, **kwargs). It requires only Node UUID
node/{id}/activities	nodes.activities(id, **kwargs). It requires only Node UUID

pyobis.groups has become obsolete and hence proposed to be deleted.

*Resources in the new API for which new functions/modules need to be created (**Proposing changes**):*

pyobis.occurrences as occ

occurrences/grid/{precision}	occ.grid(precision =None, geojson=False, **kwargs)

If GeoJSON=1, then GeoJSON response, if 0 then KML response.

occurrences/points	occ.getpoints(scientificname,...,**kwargs)
occurrences/point/{x}/{y}/{z}	occ.point(x,y,z,scientificname,...,**kwargs)
occurrences/tile/{x}/{y}/{z}	occ.tile(x,y,z,scientificname,mvt=1...,**kwargs)

If mvt=1, return response as MVT, if mvt =0, return response as GeoJSON

occurrences/centroid	occ.centroid(scientificname,...,**kwargs)

pyobis.taxa

taxon/{scientificname}	taxa.search(scientificname, **kwargs). It requires only scientificname
taxon/{id}	taxa.taxon(id, **kwargs). It requires only TaxonID
taxon/annotations	taxa.taxon(scientificname, **kwargs). It requires only scientific name

Refer: **pyobis/Function_Changes.md at gsoc2022-working-aa · ayushanand18/pyobis (github.com)** See already pushed PR #8 (successfully merged) and #11 (pending review).

➔ **Make mof response more efficient**: Will need to present each mof (MeasurementOrFacts) record per occurrence as pandas DataFrame object for ease of use. Since each id is unique to each occurrence, we can use it club mof records. For this, I will make a function in the occurrence module to return the dataframes.

See implementation: [Getting "null" mof value through /occurrence resource in the new OBIS API v3 · Issue #9 · iobis/pyobis (github.com)]

Suggested Code:

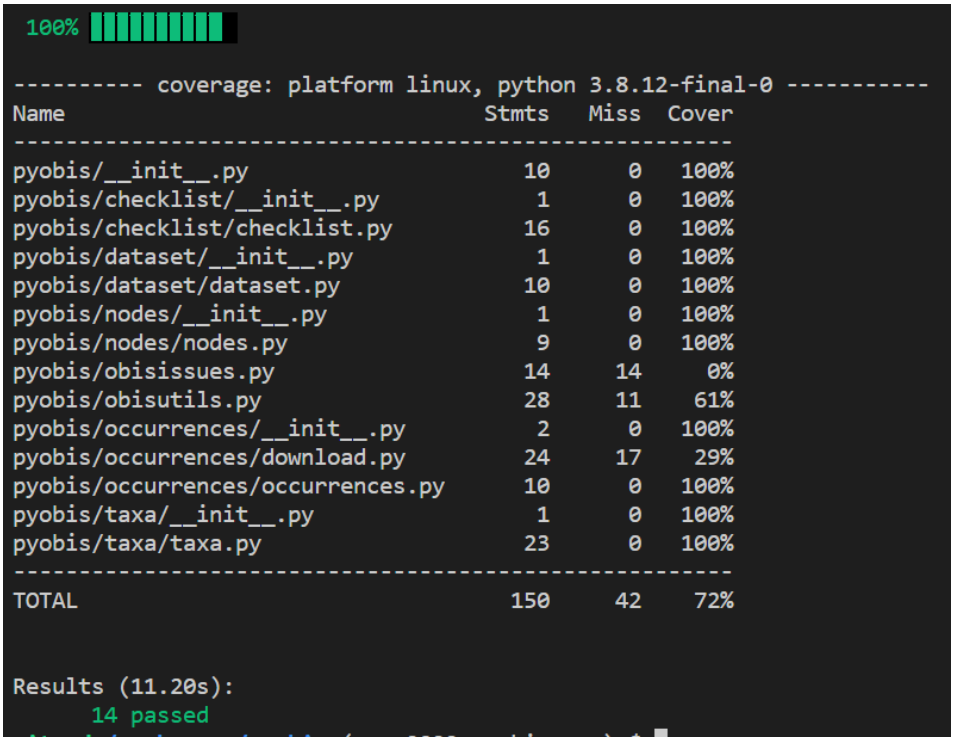
```
import requests
import pandas as pd
from pyobis import occurrence
response  = occurrence.search(scientificNames, **params)
def returnMOF(data):
      a = pd.json_normalize(data, "mof", ["scientificName",
      "eventDate", "id"])
      ids = a.id.unique()
      return [a[a['id']==ids[i]] for i in range(len(ids))]
# I will implement this function as parameter in the occurrence.search
as: occurrence.search(scientificname,..,mof=True, **kwargs)
```

- ➔ **Create tutorial usage of each module**: Adding examples of each module in a separate Jupyter Notebook with a try-it-out button to run on the Google Colab. Also, define the type of input data and the nature of expected output from the function.
  See Examples here: [pyobis/EXAMPLES.md at gsoc2022-working-aa · ayushanand18/pyobis (github.com)]
  Expand current documentation to include all module functions and create a comprehensive guide for researchers willing to use this tool, gaining inspiration from the `robis` package manual [here].
- ➔ **Clarify listing of GBIF instead of OBIS in package description ref.**:
  Will need to package description to "Python Client for OBIS".
- ➔ **Increase test coverage.**: Will increase test coverage for modules that do not already have 100% test coverage.

```
---------- coverage: platform linux, python 3.8.12-final-0 -----------
Name                                    Stmts   Miss  Cover
-----------------------------------------------------------
pyobis/__init__.py                         11      0   100%
pyobis/checklist/__init__.py                1      0   100%
pyobis/checklist/checklist.py               6      0   100%
pyobis/groups/__init__.py                   1      0   100%
pyobis/groups/groups.py                     5      1    80%
pyobis/nodes/__init__.py                    1      0   100%
pyobis/nodes/nodes.py                       5      0   100%
pyobis/obisissues.py                       14     14     0%
pyobis/obisutils.py                        28     11    61%
pyobis/occurrences/__init__.py              2      0   100%
pyobis/occurrences/download.py             24     17    29%
pyobis/occurrences/occurrences.py          10      3    70%
pyobis/resources/__init__.py                1      0   100%
pyobis/resources/resources.py              15      8    47%
pyobis/taxa/__init__.py                     1      0   100%
pyobis/taxa/taxa.py                        18      3    83%
-----------------------------------------------------------
TOTAL                                     143     57    60%
```

I aim for at least 95% (read 100%) test coverage in these highlighted modules. For this, I will write tests cases for each function in all modules. As well as increase the percentage of test cases that pass successfully.

During my experiment on the taxa module, I could successfully increase test coverage of the taxa module to 100% from 83%, and 72% coverage overall. I was also able to get all test cases passed. My experiment can be found here.



➔ **Use CI push to PyPI.**: Will update the parent package on PyPI (using GitHub Actions) once all the above modifications are approved by the mentors.

Steps:

a. Will write a .push-to-pypi.yml file in .github/workflows, and update the configuration. Although this will require mentors to set up the PyPI token and password as environment secrets.

➔ **Merge in URL changes done on the forks.**: Will initiate PR to merge all changes done so far.

➔ **Create a Jupyter notebook with a demo to analyze/visualize data grabbed using this package**: I will create a Jupyter notebook (preferably run on Google Colab) and visualize different batches of data using both Matplotlib (pyplot) and Plotly (geo_scatter plot). Will use pandas DataFrame to analyze results fetched from the package. I will also use ipywidgets for interactive viewing of data so that the users can view specific visualization without modifying the code, and above all, they look cool. Additionally if time permits, I will try to implement this paper: Chaikin, S., Dubiner, S., & Belmaker, J. (2021). Cold-water species deepen to escape warm water temperatures. *Global Ecology and Biogeography*, 31, 75– 88.

https://doi.org/10.1111/geb.13414, this is an amazing paper that investigates the shift of marine animals towards deeper waters using OBIS data.

See initial working (with some example visualizations): https://colab.research.google.com/drive/1EkeICC0tGZ3qOfX29bfMOXbejjjNwzyp#scrollTo=52px5OAO3viq

Proposed Visualizations:

Occurrence module - barchart for comparing basisOfRecords for each species (HumanObservation and Occurrence), plotting occurrence records on the world map;

Statistics resource - bar plot including the number of taxa, datasets, species, records in the yearrange, year-wise number of records from /statistics/years, pie chart for /statistics/composition;

4 Tentative Timeline & Working Hours

**All times are Indian Standard Time (UTC +5:30) unless stated otherwise.

Phase	Time	Tasks
Community Bonding	May 20 - June 12	Getting familiar with the team and mentors. Understanding the project, responsibilities, and code better, and connecting with other team members also (new people bring in new insights and sometimes a better approach to existing problems). *Other workloads:* mid-semester exams spread during the two months, so I will not be able to commit more than 3 hours in a day. *Working hours:* 16:00 - 19:00 hours IST.
Phase 1	June 13 - June 21	1: **Modifying API calls in the pyobis package** *I have already tested and successfully modified the existing modules, and created a new `dataset` module. I will also implement the proposed function changes. [See: PR #8 and PR #11*, the method here*]* 2: **Make mof response more efficient** *Will need to present each mof (MeasurementOrFacts) record per occurrence as pandas DataFrame object for ease of use. Will need to create a new function in the occurrence module to return dataframes.* *[See methods for suggested implementation]* *Other workloads:* semester studies, so can put in

		up to 5 hours in a day. *Working hours:* 15:00 - 21:00 hours IST.
	June 21 - July 3	3: **Create example usage of each module** *Will need to create example usage of each module, with input parameters explained and sample output along with the Syntax, both as docstring as well as a try-it-out Jupyter Notebook on Google Colab.* *Other workloads:* end semester exams, so I can put up to 3 hours in a day (I will compensate for this by working more during the summer break). *Working hours:* 15:30 - 18:30 hours IST.
	July 9 - July 18	4: **Clarify listing of GBIF instead of OBIS in package description ref.** *We will need to update the ./setup.py file.* 5: **Increase test coverage** *Will aim to expand test coverage to at least 95% in all modules by writing unit tests for all functions in a module.* 6: **Merge in URL changes done on the forks.** *Initiate a PR for all changes done until then.* *Other workloads:* No other workloads as summer breaks can put around 7-8 hours a day. *Working hours:* 10:00 - 20:00 hours IST. (Have added some buffer days to understand and iterate upon mentors' feedback.)
Phase 2	July 25 - Aug 10	7: **Create a Jupyter notebook with a demo to analyze/visualize data grabbed using this package.** *Will be using Matplotlib and Plotly to visualize the data obtained from API requests. Will ingest the JSON data as a pandas DataFrame and then create plots on Google Colab. Methods discussed in the previous section.* [See initial work] 8: **Use CI push to PyPI.** *Will first wait for approvals from mentors before making a CI push to check if work has been done the same way as they wanted. I will use GitHub Actions for deploying to PyPI.*

		Other workloads: semester studies, so can put in upto 3 hours in a day. *Working hours:* 16:00 - 19:30 hours IST.
	Aug 10 - Sept 4	9: **Extended deliverables** *1 Documenting each and every task I did during the period, my reasoning behind that and how it can be improved, the impact of my contribution to the Open Source community as well as ocean data researchers, challenges faced and solutions arrived at, what mistakes I committed and what others can learn from it - everything possibly in the form of blogs. And create a final project report.* *2 Will proceed towards completing extended deliverables upon mentors' ascent.* *3 If time permits, I'll also try to implement the paper I mentioned in the methods section.* *4 Continue to find opportunities to contribute to IOOS.* *Other workloads:* semester studies plus some planned research work, so I can put in up to 2.5 hours in a day. *Working hours:* 16:00 - 19:30 hours IST.
	Sept 5	**Final Evaluation** Even after the GSoC period ends, I will still work towards improving the pyobis package. If extended deliverables stand incomplete, I will work on them post-GSoC.

5 Challenges ahead

There are many challenges that I could encounter during the course of my GSoC contribution, including

a. **mof response**: While trying out the new v3 API, some fields in the mof records are null in the /occurrence resource. This missing data might be uneasy to work with during visualizations.

I tried doing a portion of each objective for the project, and I was able to reciprocate my thoughts into code. Therefore, I do not think I will face any major challenges during the course of the project.

6 Pre-GSoC IOOS Contributions

IOBIS/pyobis

March 2022

Initiated a PR after making changes to the pyobis/groups, nodes, occurrences, checklist, taxa modules to integrate them with OBIS v3 API. [Status: Successfully merged to master] [Fix] Integrated existing pyobis modules with OBIS v3 API · Pull Request #8 · iobis/pyobis (github.com)

April 2022

Initiated another PR after creating dataset module, and deleted pyobis/groups and resources. [Status: Pending Review] [Fix] Created dataset module and other module changes. · Pull Request #11 · iobis/pyobis (github.com). At the time of writing this proposal, I am working to enhance Unit Tests coverage.

7 FAQs

Q Why you? This project aims to improve a python package made for open data, we can find tons of people to work on it. Why should we choose only you?

Ans. True, pyobis is an interesting and powerful package to gather open ocean data through the OBIS API and there will be many people who are ready to contribute to this project. But, I am sure the passion I hold for marine biodiversity research and conservation can be found only in a few of them. I am committed to helping step up efforts to conserve the ocean ecosystem, make existing ocean data systems more powerful and harness the power of data into realizing our climate vision. Even my final high school project for CS was "A study on the effects of climate change on Agriculture", which involved grabbing data from AgroSphere (a NASA project), cleaning it, bringing out insights, and visualizing. It can be found here.

To become a contributor to a full-fledged package is something that I have been unable to do till now. This will be my first major open-source project and I plan on working hard to make it successful.

Q Mentors are very busy people who are volunteering their time. How will you communicate your progress or thoughts to them without burdening them?

Ans. I have shared the tentative timeline above and added some buffer days to account for mentor unavailability. I'll push separate commits for each objective accomplished to my fork (on a dev branch) and ask for a review from mentors over the changes done. After successful review, if they suggest changes I'll adopt them. Before each evaluation (twice during the period), I'll initiate PR to the parent repo.

I'll also separately mail them twice a week about project progress, and my plan of thought, and action. It is important to take their feedback before attempting any objectives so that the work proceeds smoothly without any hindrance. All my methods will be transparent and will be strictly according to the timeline.

Q Why do you want to participate in GSoC? You could have contributed otherwise too?
Ans. There are multiple reasons for me to participate in GSoC. My seniors who have previously cleared GSoC, have described it as a surreal experience. Plus, when I contributed to AOSSIE during high school it was a beautiful experience (although outside GSoC), and I want to not just relive but magnify that amazing experience. It's an exciting learning opportunity not just on the software development side but also on the community side. I'll get to interact with so many great brains, and this experience is worth having.
The other reason is working on such an impactful project and big organization like IOOS, which many governments are funding, one that has a great objective that aligns perfectly with mine - that of marine conservation, moments like these come very rarely. So, it's a natural choice for me to pick this project and this platform.

Q Do you consider yourself fit for the project? Why did you choose this project?
Ans. I believe I'm the perfect fit for this project. I'm very interested in being part of this project because of the fact that my passion and this project's primary objective are absolutely the same! This project has the prospect of impacting the tons of people who are researchers, analysts, or developers using pyobis or OBIS API for analyzing ocean data. This aspect is invigorating and it excites me, beyond all measures, to work on this project.
I am also reading climate science, especially ocean conservation research papers, and ideating on creating an open-data dashboard, and I am sure learning from this project will help me a lot.
I am well-versed with the tech stack to be used in this project and am proficient with them - Python, Pandas, Unit Tests, and Git. If the timeline is followed, the project can be completed without any issues. I have already tried doing a part of every objective for this project, and I could successfully do that. Above all, I am an Open Source advocate, and almost all tools I use every day are Open Source!

8 Open Source Contributions

AOSSIE (Australian Open Source Software Innovation and Education)
during *Google Code-In 2019*
Contributed to Open Source Projects under Australian Open Source Software Innovation and Education (AOSSIE) Organization, such as

- **Carbon Assist Function:** A Google Assistant Chatbot that helps users calculate the climate impact of their lifestyle by estimating carbon emissions produced equivalent to their daily life use of Electrical Appliances.

Successfully did many pull requests for improving the application (project on GitLab). https://gitlab.com/aossie/CarbonAssistant-Function/-/merge_requests/96
Also improved UI for other projects, including AOSSIE Agora-web, etc.

`Volunteer Experience`

Interact Club Coordinator (Rotary International sponsored clubs)
Handled shared responsibility for managing Rotary Sponsored Interact Club programs such as raising awareness against Polio Eradication and Donation drives to local Orphanages in high school.

`Other Experience`

Technical Content Writer - Indian Society for Technical Education (ISTE) NIT Durgapur Branch
I frequently write about big and emerging technologies like Big Data and artificial intelligence in medicine.

9 References

[1] pyobis 0.1.0 - PyPI, https://pypi.org/project/pyobis/, Accessed on 15th March 2022.
[2] Making Ocean biodiversity data easily accessible with python (pyobis revamp) - Issue #15 - ioos/GSoC, https://github.com/ioos/gsoc/issues/15, Accessed on 14th March 2022.
[3] Getting "null" mof value through /occurrence resource in the new OBIS API v3 · Issue #9 · iobis/pyobis (github.com), https://github.com/iobis/pyobis/issues/9, Accessed between 23 - 26 March 2022.

Ayush Anand

GitHub: https://github.com/ayushanand18
Website: https://ayushanand18.github.io
LinkedIn: linkedin.com/in/theayushanand
Email: kingwarrior880@gmail.com
Phone: (+91) 8076085724
Location: New Delhi, India [UTC +5:30]

SUMMARY

I love leveraging big data, and web technologies to build apps that address social challenges that matter the most. I am dedicated to solving daily-life problems using machine learning, and in the past have created an app that can diagnose chronic kidney using blood reports. I have also worked towards serving the community through Rotary International supported Interact Club at my school.

EDUCATION

12.2021 - present **Bachelor of Technology** at National Institute of Technology Durgapur, India
Computer Science and Engineering
Expected graduation in May of 2025

4.2020 -5.2020 **Deep Learning Nanodegree** at Udacity

3.2011 - 7.2021 **High School** at JKG International School, Ghaziabad, India
Grade: 98.8%

EXPERIENCE

11.2019 - 1.2020 **Contributor** Australian Open Source Software Innovation and Education (AOSSIE)

SKILLS AND QUALIFICATIONS

Programming Languages

Advanced skills:	PyTorch, Python, NodeJS, REST APIs (using ExpressJS)
Basic skills:	MySQL, TensorFlow, ReactJS, Microsoft Azure

Languages

Native:	Hindi
Advanced:	English

PROJECTS

Jan 2022 **GAN based Lip Movement synthesis** [https://github.com/ayushanand18/aud2lips]

- PyTorch based conditional Generative Adversarial Network model to synthesize lip movement. The model takes input as a facial image and spectrogram of the speech and generates a facial image with corresponding lip movement.
- Currently in development.

Nov - Dec 2021 **FasterRCNN model on Underwater Image Data from the Great Barrier Reef** [https://github.com/ayushanand18/FasterRCNN-Great-Barrier-Reef]

- Built a PyTorch-based FasterRCNN model to accurately identify starfish in real-time, trained on underwater videos of coral reefs.

Dec 2021 **High Energy Physics particle tracking and detection on CERN CMS detectors data**. [https://github.com/ayushanand18/cern-cms-detect-exp]

- Built a PyTorch-based Linear Sequential model on CERN high energy physics particle tracking dataset publicly available. The trained model yielded an accuracy of 98.88%

May 2020 - Oct 2020 **RespirCov | Blockchain-powered interactive ventilator and hospital beds tracker** [https://github.com/ayushanand18/covid-19]

- RespirCov is a blockchain-powered application to provide a live, interactive dashboard for monitoring ventilator and hospital beds availability in an area. It shows hospital-wise numbers for vacant and occupied ventilators and hospital beds enabling the doctors, healthcare activists, patients, and other stakeholders to check the status in real-time.