The Journal of Open Source Software

# Abil: A Python package for the interpolation of aquatic biogeochemical datasets

**Joost de Vries** [1], **Nicola A. Wiseman** [1], **and Levi John Wolf** [1]

**1** School of Geographical Sciences, University of Bristol, BS8 1HB, UK

**DOI:** 10.21105/joss.08755

**Software**

- Review
- Repository
- Archive

**Editor:** Chris Vernon

**Reviewers:**

- @ruiying-ocean
- @althonos

**Submitted:** 23 May 2025
**Published:** 14 October 2025

## Summary

Our oceans play a critical role in regulating the Earth's climate and sustaining local economies through fisheries and tourism (Dyck & Sumaila, 2010; Moreno & Amelung, 2009). However, the vast size of the ocean means that observations are inherently sparse, posing significant challenges to contextualizing these observations on a global scale (Hauck et al., 2023). Ensemble-based machine learning approaches offer an exciting avenue to address this challenge. However, the complexity of implementing these algorithms, combined with the need for extensive pre-processing and post-processing, highlights the necessity for efficient, reproducible numerical tools. Here we provide a Python package for training, predicting, and post-processing a machine learning ensemble to facilitate the global interpolation of sparse observational datasets, such as those from oceanographic cruises.

## Statement of Need

`Abil` is a Python package to interpolate sparse observations using ensemble-based machine learning algorithms. Oceanographic data is sparse in terms of spatial and temporal distribution due to the nature of collection during oceanographic cruises and requires a more informed approach than traditional gap-filling interpolation. Previous studies have utilized machine-learning methods as a solution to this problem (Luo et al., 2014; Weber et al., 2019; Yang et al., 2020), but often underlying code was not written with wide user adoption in mind. To improve reproducibility within the community, `Abil` was developed as an open-source, user-friendly interface through which machine learning methods can be more easily implemented.

## Model Design

The API for `Abil` was designed to provide a user-friendly interface to fast implementations of `scikit-learn` (Pedregosa et al., 2011) and `XGBoost` (Chen & Guestrin, 2016) ensemble-based machine learning algorithms. The user interface centers around three Python classes (optimization, prediction and post-processing) and three ensemble-based machine learning algorithms: random forests, bagged KNN and bagged XGBoost. `Abil` uses a user-defined YAML (Evans et al., 2021) model configuration for model setup, which contains model specifications such as the model features to include, hyper-parameter values, and the number of cross-folds to be used. By containing all model specifications inside a single and easy to read YAML, each model run is highly traceable and reproducible.

In addition, `Abil` includes essential tools such as pre-implemented pipelines which include environmental feature scaling – a step which is required for algorithms such as nearest neighbor algorithms, optional predictor log transformation – a step desirable in cases where high-value outliers can skew predictions, and zero-stratified cross validation for predictors where absences are more common than occurrences ("zero-inflation", which is common for ocean biogeochemical observations) (Nolan et al., 2022). For zero-inflated models we also provide zero-inflated regressor support, through the implementation of a 2-phase model pipeline which includes a classifier step to predict presence/absence before a regressor is applied in samples where presence is inferred.

## Uncertainty Estimation

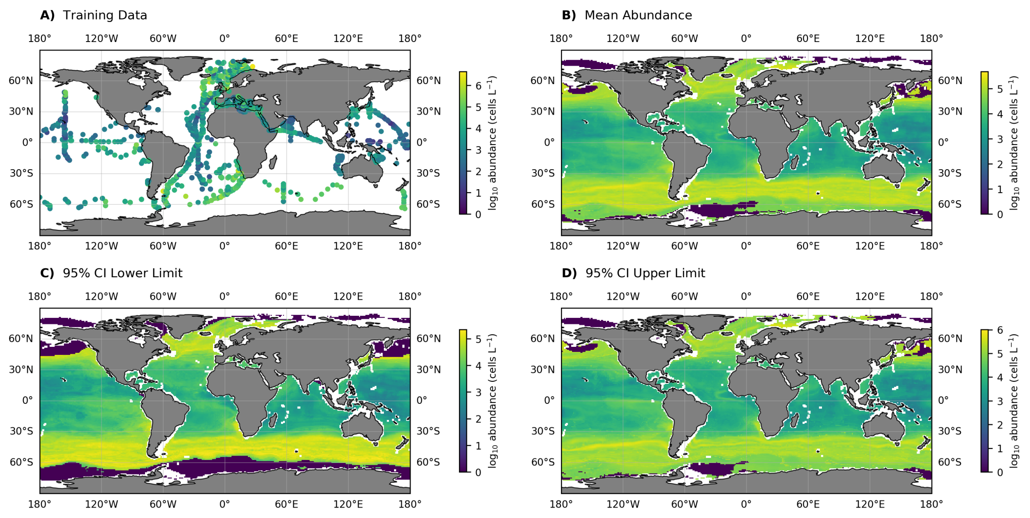

**Figure 1:** `Abil` 2-phase monthly mean abundance prediction of the coccolithophore species *Gephyrocapsa huxleyi* in the top 5 meters of the ocean. A) training data which was extracted from the CASCADE database (de Vries et al., 2024); B) Predicted mean abundance; C) Upper 95th percentile confidence intervals; D) Lower 95th percentile confidence intervals.

To estimate model prediction uncertainty, `Abil` leverages predictions of the ensemble-based machine learning members (i.e. decision trees for random forests (Breiman, 2001), and bags for bagged-KNN and bagged-XGBoost). To infer prediction quantiles, predictions are made for each ensemble member, which are then combined to estimate the 95th percentiles using loss-weighted quantiles. To reduce RAM requirements, this step is implemented using chunking, and `joblib loky` multiprocessing (The joblib developers, 2025).

## Post-processing Capabilities

To aid in model analysis `Abil` supports comprehensive post-processing functionalities. The base functionality of the `post` class merges the parameters and performance metrics for the individual models and ensemble for ease of readability. The base `post` class also creates a copy of the model_config YAML file within the ModelOutput directory to ensure documentation for reproducibility. There are additional post-processing functionalities for a variety of use cases. For species distribution modelling, `post` contains `estimate_carbon` (estimates carbon content for each target based on value from targets file), `def_groups` (sums data for groups defined in a specified dictionary), `cwm` (calculates the community weighted mean for a specified target), `diversity` (calculates the Shannon diversity index (Shannon, 1948)) and `total` (sums all targets, useful for calculating total species abundances). Additionally, the `process_resampled_runs` function calculates the mean, standard deviation, and 2.5 and 97.5 percentiles of all the targets, which is useful when running multiple targets that have been resampled from the same original data. The `integration` class is used to calculate the total integrated value for the given targets and contains options for specifying the latitudinal and longitudinal resolution (for determining the volume of each grid cell), the magnitude conversion (such as umol to Tmol), the molar mass (for converting mols to grams), as well as whether or not the target is a rate measurement or not (as this impacts whether the integrated total is a mean over time or the sum over time). The `integration` class will also automatically include the mean, standard deviation and percentiles generated from `process_resampled_runs` if they are present within the dataframe. The post-processing functionality also includes the function, `estimate_applicability`, which estimates the area of applicability of the data domain using a method similar to Meyer & Pebesma (2021).

Beyond `scikit-learn`, `Abil` leverages libraries like `xarray` (Hoyer & Hamman, 2017), `pandas` (McKinney, 2010; The pandas development team, 2025), and `Numpy` (Harris et al., 2020) for efficient data manipulation, as well as `scikit-bio` (Rideout et al., 2025) for biodiversity metrics.

## Parallel Processing and Continuous Integration

The package is optimized for parallel processing through the use of `joblib` `loky` multiprocessing (The joblib developers, 2025) and provides vignettes of high-performance computing scripts such that it can be easily ported to large scale parallel programming contexts. `Abil` is thus particularly suited to modeling the distribution of species, genes, and transcripts, as well as biogeochemical processes such as organic carbon and calcite production. Unlike many existing tools focused on predicting species occurrence (e.g., `elapid` (Anderson, 2023) and `biomod2` (Thuiller et al., 2009)), `Abil` specializes in regression challenges, enabling the prediction of abundances and rates. This focus on regression complements existing packages and fills a critical gap in the application of statistical models to ocean ecology and biogeochemistry.

`Abil` employs automated testing and continuous integration (CI) through the Python unittest framework and GitHub CI workflows. The test suite validates results against known model outputs, ensuring both correctness and reproducibility of probabilistic model results.

By combining a user-friendly interface, parallel processing capabilities, and a specific focus on regression problems, `Abil` facilitates novel scientific explorations of sparse oceanic datasets. Its versatility and computational efficiency enable researchers to address complex challenges in ocean biogeochemistry and ecology with greater ease and accuracy.

## Documentation

`Abil` documentation can be found through GitHub. The documentation includes instruction for installing the model locally and running on a High Performance Computing (HPC) system. The model process consists of tuning the model (class: `ModelTuner`), predicting the model (class: `ModelPredictor`), and postprocessing the model (class: `AbilPostProcessor`). The documentation includes usage examples for running these model steps.

## Examples

| Example | Code location |
|---|---|
| Global distribution of *Gephyrocapsa huxleyi* (Figure 1) | `paper/figure_1.py` |
| Coccolithophore abundance in the Bermuda Atlantic Time Series | `examples/regressor.py` |
| Southern Ocean distribution of *Gephyrocapsa huxleyi* | `examples/2-phase.py` |

## Research projects using the package

Abil has been used to predict the global carbon stocks of the most commonly occurring calcifying plankton ('coccolithophores') (de Vries et al., 2025). In this application the package was used to predict the abundance of 58 species with latitude, longitude, depth and month at 1-degree resolution, on a high-performance computing system using 2-phase regressors. Post processing functionalities were then used to convert abundances to cellular organic and inorganic carbon, and to accurately estimate globally integrated stocks accounting for latitudinal differences in grid sizes.

## Acknowledgements

This work was supported by funding from the UK Research and Innovation Natural Environment Research Council (CoccoTrait, NE/X001261/1; NERC GW4+ DTP studentship, NE/L002434/1). We gratefully acknowledge Fanny Monteiro for leading the acquisition of the CoccoTrait grant funding that supported this research.

## References

Anderson, C. B. (2023). Elapid: Species distribution modeling tools for Python. *Journal of Open Source Software*, *8*(84), 4930. https://doi.org/10.21105/joss.04930

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

de Vries, J., Monteiro, F. M., Poulton, A. J., Wiseman, N. A., & Wolf, L. J. (2025). A diverse community constitutes global coccolithophore calcium carbonate stocks. *bioRxiv*. https://doi.org/10.1101/2025.09.11.675535

de Vries, J., Poulton, A. J., Young, J. R., Monteiro, F. M., Sheward, R. M., Johnson, R., Hagino, K., Ziveri, P., & Wolf, L. J. (2024). CASCADE: Dataset of extant coccolithophore size, carbon content and global distribution. *Scientific Data*, *11*(1), 920. https://doi.org/10.1038/s41597-024-03724-z

Dyck, A. J., & Sumaila, U. R. (2010). Economic impact of ocean fish populations in the global fishery. *Journal of Bioeconomics*, 227–243. https://doi.org/10.1007/s10818-010-9088-3

Evans, C., Ben-Kiki, O., & Net, I. döt. (2021). *YAML ain't markup language (YAML™) version 1.2*. https://yaml.org/spec/1.2.2/

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hauck, J., Nissen, C., Landschützer, P., Rödenbeck, C., Bushinsky, S., & Olsen, A. (2023). Sparse observations induce large biases in estimates of the global ocean CO2 sink: an ocean model subsampling experiment. *Philosophical Transactions of the Royal Society A*, *381*. https://doi.org/10.1098/rsta.2022.0063

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Luo, Y.-W., Lima, I. D., Karl, D. M., Deutsch, C. A., & Doney, S. C. (2014). Data-based assessment of environmental controls on global marine nitrogen fixation. *Biogeosciences*, *11*(3), 691–708. https://doi.org/10.5194/bg-11-691-2014

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Meyer, H., & Pebesma, E. (2021). Predicting into unknown space? Estimating the area of applicability of spatial prediction models. *Methods in Ecology and Evolution*, *12*(9), 1620–1633. https://doi.org/10.1111/2041-210X.13650

Moreno, A., & Amelung, B. (2009). Climate Change and Coastal & Marine Tourism: Review and Analysis. *Journal of Coastal Research*, 1140–1144. http://www.jstor.org/stable/25737965

Nolan, V., Gilbert, F., & Reader, T. (2022). Solving sampling bias problems in presence–absence or presence-only species data using zero-inflated models. *Journal of Biogeography*, *49*(1), 215–232. https://doi.org/10.1111/jbi.14268

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.48550/arXiv.1201.0490

Rideout, J. R., Caporaso, G., Bolyen, E., McDonald, D., Baeza, Y. V., Alastuey, J. C., Pitman, A., Morton, J., Zhu, Q., Navas, J., Gorlick, K., Debelius, J., Xu, Z., Aton, M., llcooljohn, Shorenstein, J., Luce, L., Treuren, W. V., Chase, J., … Murray, Dr. K. D. (2025). *Scikit-bio/scikit-bio: Scikit-bio 0.6.3* (Version 0.6.2). Zenodo. https://doi.org/10.5281/zenodo.593387

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

The joblib developers. (2025). *joblib* (latest). https://doi.org/10.5281/zenodo.14915601

The pandas development team. (2025). *Pandas-dev/pandas: pandas* (Version 2.2.2). Zenodo. https://doi.org/10.5281/zenodo.10957263

Thuiller, W., Lafourcade, B., Engler, R., & Araújo, M. B. (2009). BIOMOD–a platform for ensemble forecasting of species distributions. *Ecography*, *32*(3), 369–373. https://doi.org/10.1111/j.1600-0587.2008.05742.x

Weber, T., Wiseman, N. A., & Kock, A. (2019). Global ocean methane emissions dominated by shallow coastal waters. *Nature Communications*, *10*(1), 4584. https://doi.org/10.1038/s41467-019-12541-7

Yang, S., Chang, B. X., Warner, M. J., Weber, T. S., Bourbonnais, A. M., Santoro, A. E., Kock, A., Sonnerup, R. E., Bullister, J. L., Wilson, S. T., & Bianchi, D. (2020). Global reconstruction reduces the uncertainty of oceanic nitrous oxide emissions and reveals a vigorous seasonal cycle. *Proceedings of the National Academy of Sciences*, *117*(22), 11954–11960. https://doi.org/10.1073/pnas.1921914117