OMEGA-PY: A NEW ALL-IN-ONE PYTHON SOLUTION FOR OMEGA/MEX DATA.

A. Stcherbinine[1], Y. Langevin[2], J. Carter[2,3], M. Vincendon[2], Y. Leseigneur[4] and O. Barraud[5]
[1]IRAP, CNES, Université Toulouse III Paul Sabatier, CNRS, Toulouse, France (Aurelien.Stcherbinine@irap.omp.eu), [2]IAS, Université Paris-Saclay, CNRS, Orsay, France, [3]LAM, Université Aix-Marseille, CNRS, CNES, Marseille, France, [4]LATMOS/IPSL, UVSQ Université Paris-Saclay, CNRS, Sorbonne Université, Guyancourt, France, [5]German Aerospace Center (DLR), Institute of Planetary Research, Berlin, Germany.

Introduction: *OMEGA-Py* [1] is a Python 3 module dedicated to the scientific use of data provided by the Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité (OMEGA) instrument onboard the ESA Mars Express (MEx) orbiter [2]. It has been developed as an alternative to the IDL routines of the OMEGA legacy software provided by the instrument team for the past 20 years [3] and its validation by comparison with the results of the IDL routines has been endorsed by the OMEGA PI.

The module notably includes a Python re-implementation of the most recent release of the IDL OMEGA software (v10, *SOFT10*) [3], which performs the reading, calibration, and reduction of the level 1B data publicly available on the ESA PSA to produce level 2A data (calibrated reflectances cubes) that can be used for the scientific analysis. In addition, the module also include built-in functions to perform the atmospheric and thermal corrections of the data (using previously published methods), and graphic tools dedicated to the OMEGA hyperspectral data including interactive visualization of the cubes or generation of composite OMEGA maps.

Also, since the release of its version 3.0 in October 2023, *OMEGA-Py* is now distributed as part of the official OMEGA software ecosystem (`https://www.ias.u-psud.fr/omega/software.html`).

Why this module?: The accessibility of data returned by space missions is a crucial point to ensure the development of open science, and unfortunately the OMEGA dataset has acquired over the past years a reputation in the community for being challenging and requiring a significant amount of investment to use. With *OMEGA-Py* we aim to tackle this reputation by providing a free all-in-one toolbox to load, correct, analyze, and visualize the OMEGA data, along with a full online documentation with examples (available at `https://astcherbinine.github.io/omegapy`).

Plus, while the OMEGA dataset is publicly available with yearly releases, the legacy pipeline uses a proprietary software (IDL), which raises some concerns including: (1) the cost of an IDL license makes it not available to everyone; (2) as the community is moving to use mostly Python instead of IDL, this language requirement can limit the access to the OMEGA dataset, especially for the younger generation of scientists and students.

Data handling: *OMEGA-Py* includes a reimplementation of the `readomega.pro` routine of the IDL software, which reads and processes the L1B data that can be downloaded from the PSA to generate the L2A. A sensible improvement with *OMEGA-Py* is its implementation in Object Oriented Programming (OOP), which makes the simultaneous handling of several OMEGA observations easier. When loading an OMEGA observation, all the extracted data are stored within an `OMEGAdata` object. The loading process is as simple as `omega = OMEGAdata("0967_3")` to load the OMEGA observation cube 0967_3 for instance. Then, all the data (reflectance, geometry, wavelengths...) and metadata (L_s, MY, quality flag...) can be accessed as attributes of the newly created object.

In addition, the wavelengths are also automatically reordered in ascending order and the spectra are "cleaned" to remove the overlaps between the three channels (V, C, L) as well as the corrupted pixels previously identified with the `ic` array of the `readomega.pro` IDL routine. This allows the user to have a direct access to reflectance spectra that can be used for scientific analysis purposes.

Data correction: Using orbital remote sensing observations like OMEGA data for scientific analysis and research may require to apply some corrections to remove the atmospheric and/or thermal contributions that affect the spectra. Methods to correct the OMEGA observations have been described in several publications and extensively used over the past 20 years [e.g., 4, 5], but they have never been part of the distributed IDL software.

Atmospheric correction. The correction of the average atmospheric gas absorptions in the OMEGA spectra is performed using the widely used volcano-scan technique, which consists of scaling a typical atmospheric spectrum from the intensity of the 2 µm CO_2 atmospheric band [4].

Thermal correction. If the thermal emission of the planet in the OMEGA spectra can be neglected for wavelengths smaller than ~ 3 µm, it has to be taken into account when using larger wavelengths (i.e., data from the L-channel). The idea is to assess the temperature at the surface of the planet for each pixel, then the surface emission is modeled as a black body with a Planck's law to remove it from the measured spectra to have only the reflectance. Two methods have been implemented, with one relying on the availability of data from the C-channel [5] and an other that can be used with the L-channel only [6] (the C-channel stopped working in August 2010).

As the thermal correction of an OMEGA observation is resource and time-consuming, parallel processing using the `multiprocessing` module has been implemented for the first method.

Visualization: *OMEGA-Py* also comes with a set of visualization functions specifically developed for the OMEGA data.

Data projections. The user can choose to project the data on a longitude/latitude equatorial or polar grid according to the geometry of the observation, or to display an image without projection (axes in pixel #). When generating a projected map, it is possible to specify whether you want to use the geometry computed for the V-channel or for the C/L-channels (as the two optical blocks are not perfectly co-aligned).

Composite maps. It is also possible to easily generate composite maps from a series of OMEGA observations, either for the reflectance or a previously computed spectral criteria. See for instance Figure 1 that shows the intensity of the 1.9 µm band around the north polar cap.

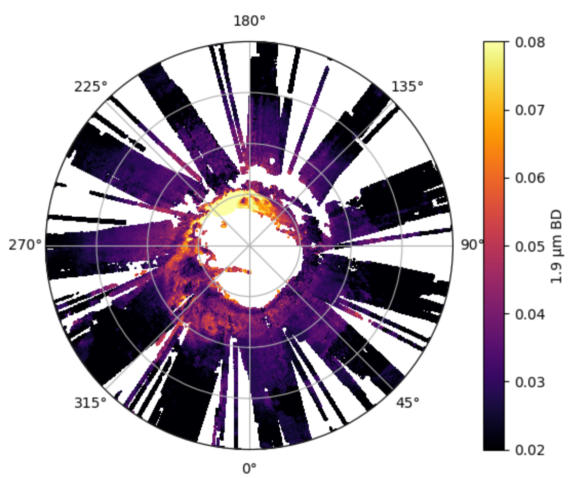

Figure 1 *– 1.9 µm band depth map in the North polar regions from OMEGA observations, excluding areas covered by water ice, made with* OMEGA-Py *[7].*

Interactive display. In addition, one of the very useful tool provided by *OMEGA-Py* is the interactive visualization of an OMEGA observation with the `show_omega_interactif_v2()` function. It provides a graphic interface to explore the spectra of the data cube, similar to what can be done with *ENVI* for IDL users. By clicking with the mouse on the OMEGA image (reflectance map or higher-level derived map), the user can display the spectra from the selected pixels on the map. This is a very useful way to explore the spectral and spatial diversity of an OMEGA observation.

Conclusion & Perspectives: With *OMEGA-Py* we provide a new and complete solution to use and analyze the data that are returned since 2004 by the OMEGA instrument. The Python implementation and the presence of built-in correction and visualization functions will help in making the huge and very complete OMEGA dataset (now rich of more than 20 years of observations) more accessible, especially for the younger generation of scientists and students, or for people who want to have a quick and easy access to some OMEGA data for comparison with other instruments, models or lab work. Plus, its very easy way to load OMEGA data and the functionalities to improve the processing time make the module perfectly suitable for batch processing or data mining.

The official validation of the module by the instrument team in 2023, as an alternative solution to the legacy IDL software, is an important step for the project. Following this release, an article dedicated to the *OMEGA-Py* module is currently under review for publication in the *Journal of Open Source Software*.

Since its first release in 2020, *OMEGA-Py* has been used in published studies and ongoing projects [7–10].

Acknowledgments: The OMEGA/MEx data are freely available on the ESA PSA at `https://archives.esac.esa.int/psa/#!Table%20View/OMEGA=instrument`.
The full documentation of the OMEGA-Py Python module is available at `https://astcherbinine.github.io/omegapy`.

We thank all the people who helped with testing and improving the module.

References: [1] Stcherbinine A., OMEGA-Py: Python tools for OMEGA data (2023), *Zenodo*. [2] Bibring J.-P. et al. (2004) *ESA Publication Division*, 1240:37–49. [3] OMEGA Team , SOFT 10 IDL routines (2004). [4] Langevin Y. et al. (2005) *Science*, 307(5715):1584–1586. [5] Jouglet D. et al. (2007) *JGR: Planets*, 112(E8). [6] Audouard J., Ph.D. thesis Université Paris-Sud XI (2014). [7] Stcherbinine A. et al. (2021) *Icarus*, 369:114627. [8] Leseigneur Y. and Vincendon M. (2023) *Icarus*, 392:115366. [9] Barraud O. et al. (2022) EPSC2022–847. [10] Brasil F. et al. (2023) *AAS/DPS* 55:213.06.