

OMEGA-Py: Python Tools for OMEGA Data – v2.3

Aurélien Stcherbinine,

John Carter, Yves Langevin, Mathieu Vincendon, Yann Leseigneur,
Océane Barraud, Jean-Pierre Bibring

6th Planetary Data Workshop – Flagstaff, 26 June 2023

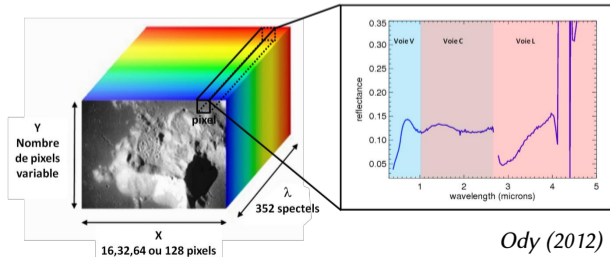


Department of Astronomy and
Planetary Science



The OMEGA instrument & data

- ▶ *Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité*
- ▶ Visible – near-infrared imaging spectrometer onboard the ESA *Mars Express* (MEx) orbiter
- ▶ Operating since 2004 (recently extended until 2026)
- ▶ Covers the 0.35 – 5.1 μm spectral range over 352 spectels
- ▶ 3 channels: V / C / L
- ▶ Spatial resolution: 300 m to 2-5 km



Ody (2012)

What is OMEGA-Py?

- ▶ **Python 3** module



- ▶ Available on GitHub at: <https://github.com/AStcherbinine/omegapy>

- ▶ And on PyPI: <https://pypi.org/project/omegapy>

- ▶ DOI:10.5281/zenodo.7818829

- ▶ Current version: **2.3**

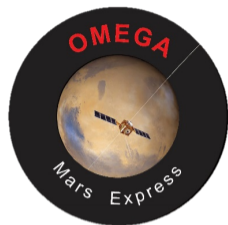
- ▶ Specifically dedicated to the scientific analysis of OMEGA data:

- ▶ All-in-one toolbox
- ▶ Developed as an alternative to the historical *SOFT 10* IDL routines of the official OMEGA software
- ▶ Easier handling of several OMEGA observations using OOP
- ▶ Provide easier access to OMEGA data to a new generation of scientists



OMEGA-Py features

- ▶ Handling of OMEGA/MEx data
- ▶ Loading and correction of *.QUB* and *.NAV* binary files to obtain L1B
 - ▶ *Python re-implementation of the SOFT 10 IDL routines*
- ▶ Interactive display of hyperspectral images
- ▶ Thermal & Atmospheric corrections pre-implemented
- ▶ Generation of composite maps



Module structure

omegapy

- ▶ `omega_data`: data importation, handling, corrections functions
- ▶ `omega_plots`: all the display functions
- ▶ `useful_functions`: additional useful functions (e.g., searching wavelength index in array)

Recommended importation

```
1 import omegapy.omega_data as od
2 import omegapy.omega_plots as op
3 import omegapy.useful_functions as uf
```

Data importation

- ▶ Read binaries to produce L1B data similarly to *SOFT 10* `readomega.pro`

New features:

- ▶ Clever search for observations
- ▶ Spectral correction: re-ordering wavelength + removing overlaps
- ▶ "No-verbose" option
- ▶ Possibility to skip Vis or L channels corrections to fasten importation

```
In [2]: omega = OMEGAdata('0967')
7 files found :
1 : /data2/opt/geomeg/data/product/ORB0967_0.QUB
2 : /data2/opt/geomeg/data/product/ORB0967_1.QUB
3 : /data2/opt/geomeg/data/product/ORB0967_2.QUB
4 : /data2/opt/geomeg/data/product/ORB0967_3.QUB
5 : /data2/opt/geomeg/data/product/ORB0967_4.QUB
6 : /data2/opt/geomeg/data/product/ORB0967_5.QUB
7 : /data2/opt/geomeg/data/product/ORB0967_6.QUB

Enter the corresponding number to select one filename :
>>> 4

Computing OMEGA observation ORB0967_3
core:          128      352      596  cbyte:      2
suffix:         1         7         0  sbyte:      4
      0 or less IR:   173073
      negative pixels VIS:   12402
      anomalous pixels VIS:    100
      saturated pixels VIS:    100
      spikes VIS:      3385

Computing data extraction and correction... [done]

In [4]: omega = OMEGAdata('0967_3', disp=False)
```

Data handling

```
In [3]: omega
Out[3]:

OMEGA/MEx observation ORB0967_3 - (v2)

Ls = 103.5° - MY 27

Cube quality : 3
Thermal correction : False
Atmospheric correction : False

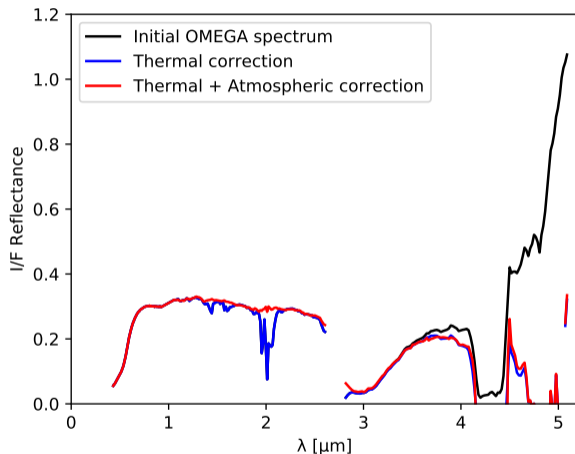
Corrupted 128 pixels cube
```

- ▶ Object Oriented Programming
 - ▶ Easy handling of multiple OMEGA observations
 - ▶ Saving/Loading of OMEGAdata objects
- ▶ All informations stored as attributes of the OMEGAdata class:
 - ▶ `omega.name`: observation ID
 - ▶ `omega.lam`: wavelength array
 - ▶ `omega.cube_rf`: I/F.cos(i) data cube $[X, Y, \lambda]$
 - ▶ `omega.l_s`: Solar Longitude (L_s)
 - ▶ `omega.my`: Martian Year
 - ▶ `omega.lat`: Latitude array $[X, Y]$
 - ▶ `omega.lon`: Longitude array $[X, Y]$
 - ▶ *etc*
 - ▶ Getters for whole headers if needed

OMEGA data corrections

Pre-implemented data corrections:

- ▶ Thermal correction
 - ▶ Required to use L-channel ($\lambda > 2.7 \mu\text{m}$)
 - ▶ 2 methods available (with / without C-channel)
- ▶ Atmospheric correction
 - ▶ Volcano-scan technique – Scaling an empirical spectrum of the atmosphere using the CO_2 $2 \mu\text{m}$ band
 - ▶ 2 methods available
- ▶ Simultaneous thermal & atmospheric corrections (*recommended for using the L-channel*)



Applying corrections to OMEGA data

```
1 import omegapy.omega_data as od
2
3 omega = od.OMEGAdata('0967_3') # Loading cube 0967_3
4
5 # Atmospheric correction only
6 omega_corr_atm = od.corr_atm(omega)
7
8 # Thermal correction only
9 omega_corr_therm = od.corr_therm(omega, npool=10)
10
11 # Both thermal & atmospheric corrections
12 omega_corr_therm_atm = od.corr_therm_atm(omega, npool=10)
```

`npool` → Number of parallel processes used to compute thermal correction

Checking applied corrections

```
1 omega_corr.therm_corr # boolean
2
3 omega_corr.atm_corr   # boolean
```

```
In [3]: omega
Out[3]:

OMEGA/MEX observation ORB0967_3 - (v2)

Ls = 103.5° - MY 27

Cube quality : 3
Thermal correction : False
Atmospheric correction : False

Corrupted 128 pixels cube
```

No corrections applied

```
In [6]: omega_corr
Out[6]:

OMEGA/MEX observation ORB0967_3 - (v2)

Ls = 103.5° - MY 27

Cube quality : 3
Thermal correction : True
Atmospheric correction : True

Corrupted 128 pixels cube
```

Both corrections applied

Saving & Loading OMEGAdata

Single observation

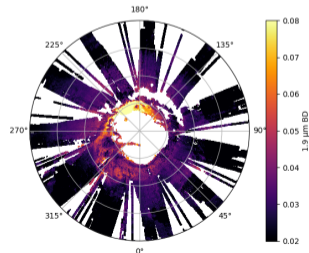
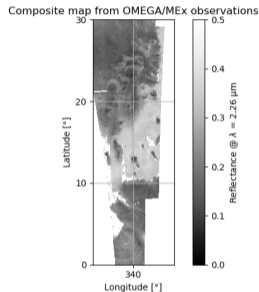
```
1 od.autosave_omega(omega) # Saving one OMEGAdata object
2 # Apply both corrections then save the data
3 od.corr_save_omega2('0979_3', npool=10)
4 # Load previously saved OMEGAdata object (with selected corrections)
5 omega_corr = od.autoload_omega('0979_3', therm_corr=True, atm_corr=True)
```

Multiple observations

```
1 od.corr_save_omega2_list(cubesID_list, npool=10)
2 # cubesID_list, e.g., ['0967_3', '0979_3', '0979_4', ...]
3 omega_list = od.load_omega_list2(cubesID_list, therm_corr=True,
  ↪ atm_corr=True)
```

Visualizing the OMEGA data

- ▶ Non-projected (X / Y grid)
- ▶ Projected (lon / lat grid)
- ▶ Polar projection
- ▶ Interactive display of a cube
- ▶ Composite maps from multiple cubes
- ▶ Reflectance (at a given λ) or computed high-level map
- ▶ Masks to hide flagged pixels

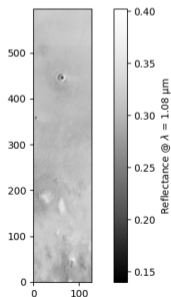


Data projection

Non-projected

```
1 op.show_omega(omega, lam=1.085)
```

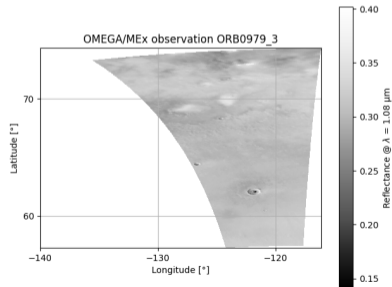
OMEGA/MEx observation ORB0979_3



Projected

```
1 op.show_omega_v2(omega,  
↪ lam=1.085)
```

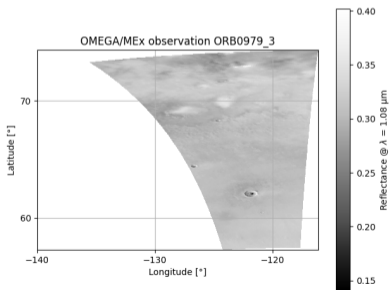
OMEGA/MEx observation ORB0979_3



Replacing the reflectance by previously computed data maps

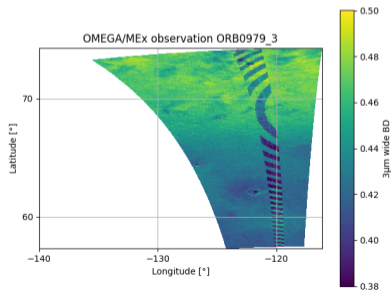
Surface reflectance

```
1 op.show_omega_v2(omega,  
  ↪ lam=1.085)
```



Band depth

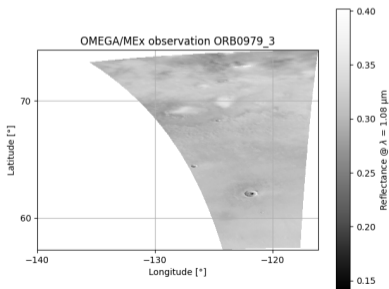
```
1 op.show_data_v2(omega,  
  ↪ data=nibd30, vmin=0.38,  
  ↪ vmax=0.5)
```



Data masks

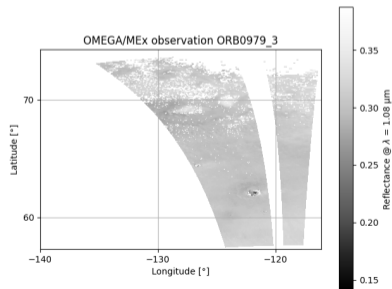
Without mask

```
1 op.show_omega_v2(omega,  
↪ lam=1.085)
```



With mask (corrupted pixels + ice)

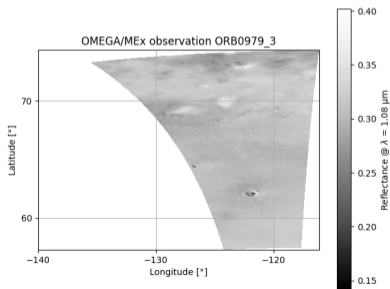
```
1 op.show_omega_v2(omega,  
↪ lam=1.085, mask=data_mask)
```



Polar viewing

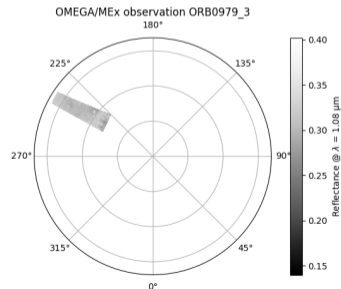
Equatorial projection

```
1 op.show_omega_v2(omega,  
↪ lam=1.085)
```



Polar projection

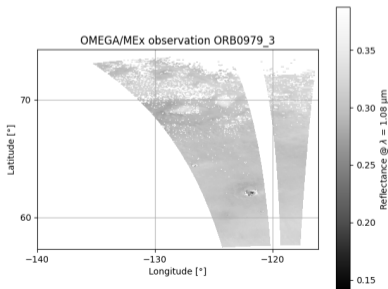
```
1 op.show_omega_v2(omega,  
↪ lam=1.085, polar=True)
```



Polar viewing – With masks

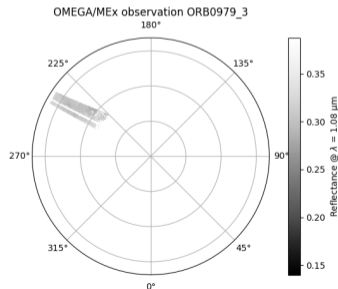
Equatorial projection

```
1 op.show_omega_v2(omega,  
↳ lam=1.085, mask=data_mask)
```



Polar projection

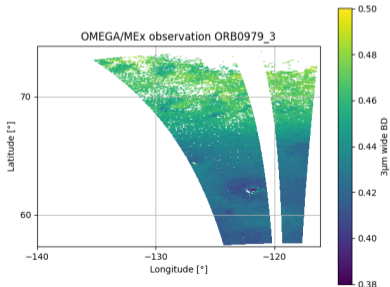
```
1 op.show_omega_v2(omega,  
↳ lam=1.085, polar=True,  
↳ mask=data_mask)
```



Polar viewing – With masks & Data maps

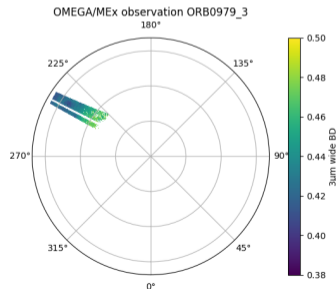
Equatorial projection

```
1 op.show_data_v2(omega,  
  ↪ data=nibd30, mask=data_mask)
```



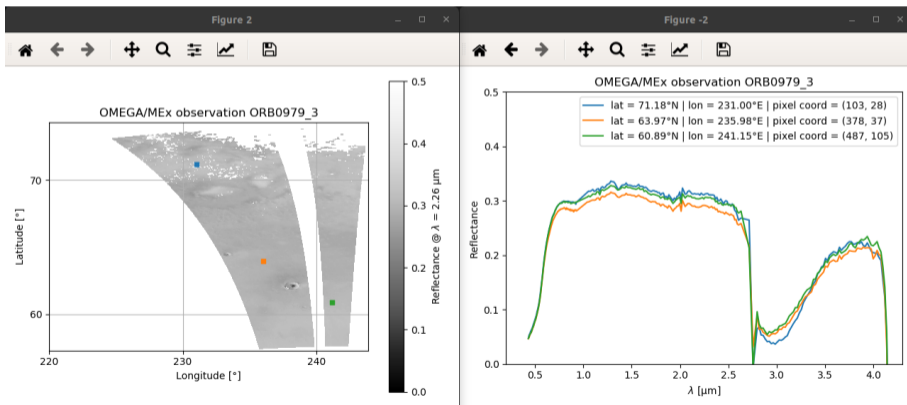
Polar projection

```
1 op.show_data_v2(omega,  
  ↪ data=nibd30, polar=True,  
  ↪ mask=data_mask)
```



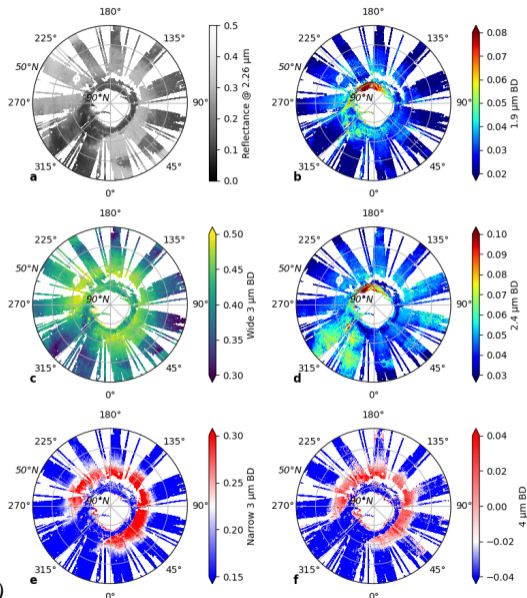
Interactive display

```
1 op.show_omega_interactif_v2(omega_corr_therm_atm, lam=1.085, mask=mask,  
  ↪ vmin=0, vmax=0.5) # Possible to use data map instead of reflectance
```



Composite maps

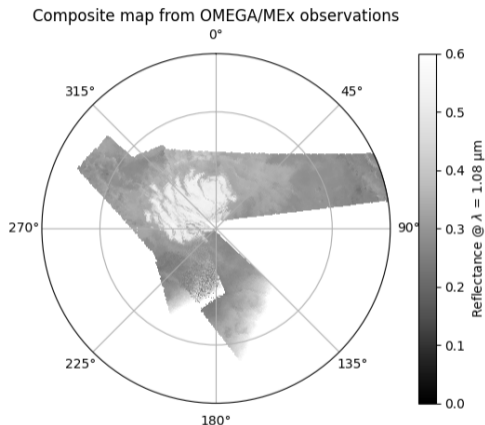
- ▶ Generation of composite maps from multiple OMEGA observations with `op.show_omega_list_v2`
- ▶ Each individual map is projected on a new user-defined lon/lat grid
- ▶ Average data if overlapping
- ▶ Possibility to export the gridded data



Stcherbinine et al. (2021)

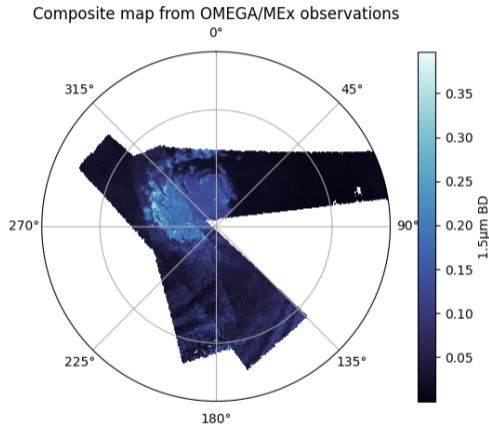
Composite maps – Reflectance

```
1 op.show_omega_list_v2(  
2   [omega41, omega61, omega103],  
3   # Grid parameters  
4   lat_min=-90, lat_max=-75,  
5   lon_min=0, lon_max=360,  
6   pas_lon=0.1, pas_lat=0.1,  
7   # Polar viewing  
8   polar=True,  
9   # Colorscale min/max  
10  vmin=0, vmax=0.6,  
11  lam=1.085, # Reflectance wavelength  
12  cmap='Greys_r', # Colormap  
13 )
```



Composite maps – Data & Masks

```
1 op.show_omega_list_v2(  
2     [omega41, omega61, omega103],  
3     # Grid parameters  
4     lat_min=-90, lat_max=-75,  
5     lon_min=0, lon_max=360,  
6     pas_lon=0.1, pas_lat=0.1,  
7     # Polar viewing  
8     polar=True,  
9     cmap=cmo.ice, # Colormap  
10    data_list = [bd15_41, bd15_61,  
11                 ↪ bd15_103],  
12    mask_list = [mask41, mask15,  
13                 ↪ mask103],  
14    cb_title = r'1.5$\mu$m BD',  
15    )
```



Saving & Loading composite maps

Saving gridded map

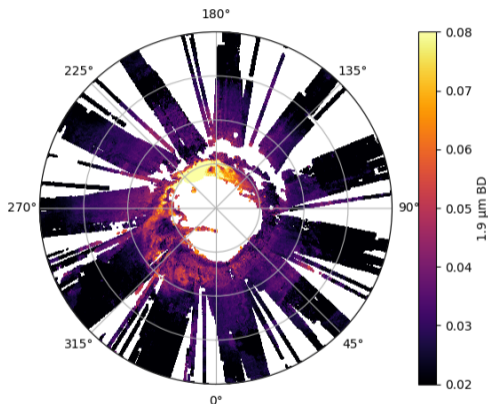
```
1 op.save_map_omega_list(  
2     omega_list,  
3     lat_min=50, lat_max=90,  
4     lon_min=0, lon_max=360,  
5     pas_lat=0.1, pas_lon=0.1,  
6     data_list=bd_19_list,  
7     data_desc='1.9 $\mu$m BD',  
8     mask_list=mask_ice_list,  
9     ext='BD_1.9microns_mask4C',  
10    sub_folder='mask4C',  
11    folder='../data',  
12    )
```

Loading data

```
1 data_bd19, mask, grid_lat,  
  ↪ grid_lon, mask_obs, infos =  
  ↪ op.load_map_omega_list(  
2     '../data/mask4C/'  
3     'res_show_omega_list_v2'  
4     '__lat50-90_pas0.1_'  
5     'lon000-360_pas0.1_'  
6     'BD_1.9microns_mask4C.pkl')
```

Displaying previously saved map

```
1 op.show_omega_list_v2_man(  
2   data_bd19,  
3   grid_lat,  
4   grid_lon,  
5   infos,  
6   cmap='inferno',  
7   vmin=0.02,  
8   vmax=0.08,  
9   title='',  
10  polar=True,  
11  cb_title='auto',  
12  )
```



Summary

- ▶ New tool to handle, display and analyze OMEGA/MEx data
- ▶ Python alternative to the historical IDL software
- ▶ Freely available on GitHub and PyPi
- ▶ Easy way to access OMEGA data:
 - ▶ Reading of binary files
 - ▶ Pre-implemented thermal & atmospheric corrections
 - ▶ Interactive display & Generation of composite maps
- ▶ Already used in several studies



The future of OMEGA-Py

- ▶ Official validation is ongoing → Will be released as **version 3**
- ▶ Will probably come with a better naming of the functions
- ▶ Improvements of the customization of the plotting functions
- ▶ *Ongoing development → Feedback and suggestions are welcome!*



@ aurelien.stcherbinine@nau.edu

🌐 aurelien.stcherbinine.net



@Astrorelien

