

PLANETSERVER: A HYPERSPECTRAL ANALYSIS AND VISUALIZATION WEB CLIENT AND PYTHON API. R. Marco Figuera¹, B. Pham Huu¹, A. Halder¹, K. Yin¹, A. P. Rossi¹, ¹Jacobs University Bremen, Department of Physics and Earth Sciences, Campus Ring 1, 28795 Bremen, Germany. (r.marcofiguera@jacobs-university.de).

Introduction: Mineralogical analysis of planetary surfaces lacks of available open-source tools. In this abstract we present PlanetServer, a web client and Python API capable of analyze and visualize CRISM TRDR [1] hyperspectral imagery. PlanetServer provides the possibility to create RGB band math combinations as well as spectral analysis.

Service Description: PlanetServer comprises a server, a web client [2] and a Python API [3]. On the server side data are stored using the Array DataBase Management System (DBMS) Raster Data Manager (Rasdaman) [4]. Rasdaman is an Array DBMS offering features such as query languages, query optimization and parallelization on n-D arrays. Open Geospatial Consortium (OGC) standards such as the Web Coverage Processing Service (WCPS) [5], an SQL-like language that is capable to query information along an image cube, are implemented in the PetaScope component [6], a set of geospatial and geometry libraries, data access libraries and relational database access components. The web client uses NASA's World Wind [7] general-purpose 3D/4D client as a virtual globe to interactively analyze and visualize data. The Python client API provides the user the possibility to analyze the data within Python and embed the results in existing data analysis pipelines.

PlanetServer Python API: In order to characterize minerals from CRISM data, different CRISM summary products exist [8] and [9]. PlanetServer's Python API integrates all the CRISM products mentioned in [9] and performs different RGB band math combinations. In addition to the analyzed image, spectral profiles are also available in the API. The Python API has been constructed such that a user only needs to give five input arguments to the API: The coverage ID, the name of three CRISM summary products that are stored in each channel (Red, Green and Blue) and the name of a FITS table containing the relation between the band names and the corresponding wavelengths (Figure 1). The API combines the 3 summary products (predefined in the API) into a WCPS query¹ giving an image as output. The spectral analysis tool shows the reflectance versus wavelength and can be further compared to existing laboratory spectra such as the splib06a [10]. A set of Jupyter notebooks including all of the Planet-

Server capabilities are also available in the GitHub repository [3].

PlanetServer Web Client: PlanetServer's web client provides an easy and intuitive solution to visualize and analyze hyperspectral imagery from two planetary bodies with one dataset each: CRISM TRDR on Mars and M3 Level2 on the Moon (beta version). It contains a 3D/2D globe where all available cubes are deployed. The left panel contains the projections, base map and layers selector and the RGB combinator. In the right menu, the plot docks for single spectra retrieval and for spectral ratio calculations are located.

Discussion: PlanetServer web client and Python API has been used as an alternative to existing pipelines for hyperspectral analysis. The results obtained with PlanetServer correlate with previous studies [9], [11]. The creation and population of a library of WCPS queries with summary products allows reproducible analysis and workflow. All of the PlanetServer components (server engine, web client and Python API) are Open Source and suitable for customization.

Ongoing work: Currently, we are preprocessing M3 Moon data to be added to the database. A beta version with few coverages is already available. A subset of CRISM MRDR images is now available as a beta version. Future plans cover the inclusion of topographic data (MEX HRSC) and MEX OMEGA data.

Acknowledgments: EarthServer-2 has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 654367

References: [1] S. Murchie *et al.*, *JGR*. 2007. [2] R. Marco Figuera, "PlanetServer web client," Available: <http://doi.org/10.5281/zenodo.200371> 2016. [3] A. Halder and R. Marco Figuera, "PlanetServer Python API," Available: <http://doi.org/10.5281/zenodo.204667>. [4] P. Baumann *et al.*, *ACM SIGMOD Rec.*, Jun. 1998. [5] P. Baumann, *Geoinformatica*, Jul. 2010. [6] A. Aiordăchioaie *et al.*, *LNCS*, Jun. 2010. [7] D. G. Bell *et al.*, *IEEE Aerospace Conference Proceedings*, 2007. [8] S.M. Pelkey *et al.*, *JGR*. 2007. [9] C. E. Viviano-Beck *et al.*, *JGR. E Planets*, 2014. [10] R. N. Clark *et al.*, 2007. Available: <http://speclab.cr.usgs.gov/spectral.lib06>. [11] B. L. Ehlmann *et al.*, *JGR*. 2009.

¹ <https://tinyurl.com/n6jlar5>

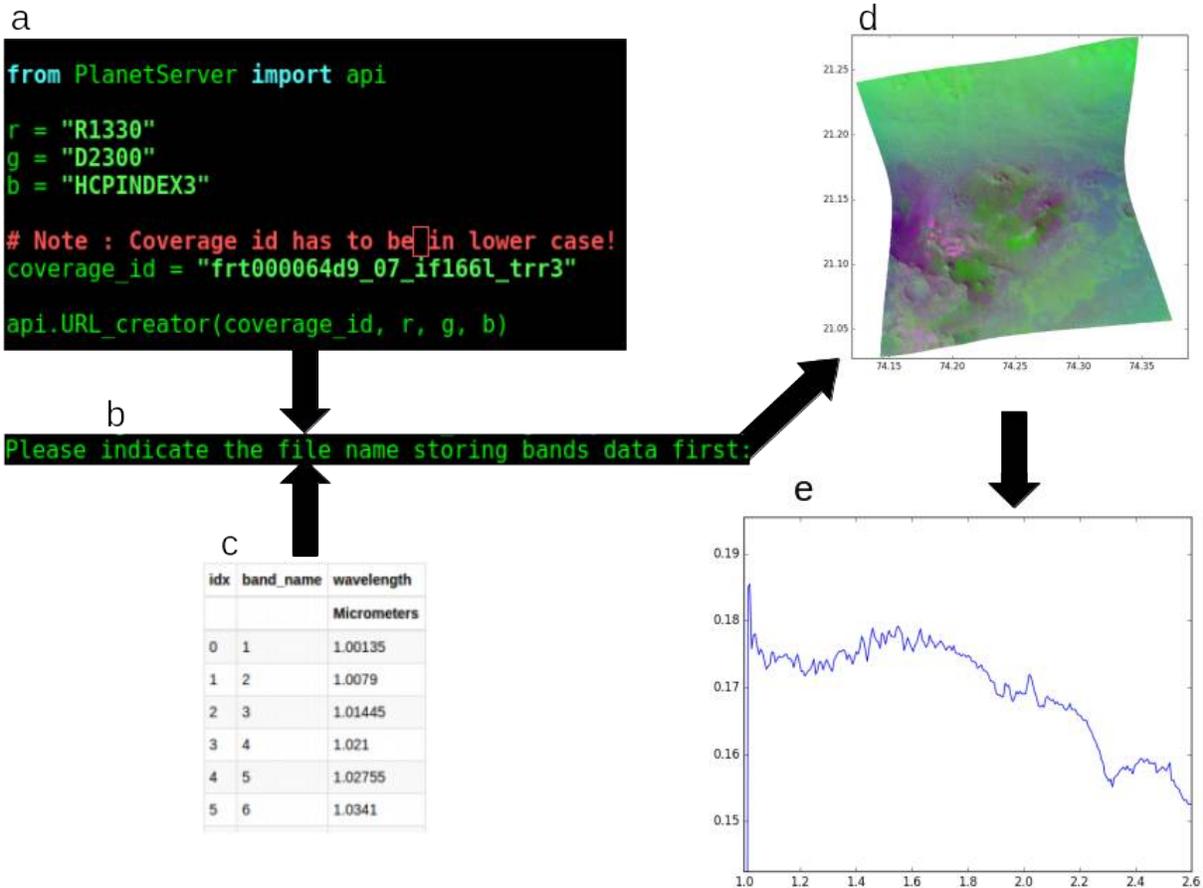


Figure 1: PlanetServer Python API workflow. a) The user inputs the three CRISM summary products in the corresponding RGB channels (R:LCPINDEX2, G:D2200 and B:HCPINDEX3) and the coverage ID. b) A FITS table (provided in the package) with the relation between band names and wavelengths (c) needs to be called in order to create correct WCPS queries. d) The API outputs an image containing the RGB combination. e) By clicking a location in the image (e.g: 21.19109°N, 74.24328°E), the API will show the spectral profile.