THE PYTHON HYPERSPECTRAL ANALYSIS TOOL (PyHAT) AND LASER-INDUCED BREAKDOWN SPECTROSCOPY SPECTRAL DATABASE. R.B. Anderson[1], N. Finch[2], S. Clegg[3], T. Graff[4], I. Aneece[5], R.V. Morris[4] [1]U.S. Geological Survey, Astrogeology Science Center, Flagstaff, AZ (rbanderson@usgs.gov); [2]Arizona State University [3]Los Alamos National Laboratory, [4]NASA Johnson Space Center, [5]USGS Western Geographic Science Center

Introduction: The Python Hyperspectral Analysis Tool (PyHAT) [1] is an open-source library developed by the USGS Astrogeology Science Center for analyzing planetary spectral data using Python. PyHAT development to date has been supported by two NASA Planetary Data Archiving, Restoration, and Tools (PDART) grants. One is focused on the analysis of data from orbital imaging spectrometers such as the Moon Mineralogy Mapper (M3) or Compact Reconnaissance Imaging Spectrometer for Mars (CRISM)[2]. The second grant is currently in its final year and is the focus of this abstract. It involves the development of analysis capabilities for point spectra with a focus on Laser-Induced Breakdown Spectroscopy (LIBS) instruments such as ChemCam or SuperCam, as well as collecting and processing a suite of spectra of planetary analog materials on two LIBS instruments.

PyHAT Point Spectra Analysis: Point spectra are collected by instruments from a single location on a target. In particular, rover instruments commonly collect point spectra rather than hyperspectral images. ChemCam has been the most prolific point spectra instrument to date, with over 600,000 LIBS spectra collected. Its successor SuperCam on the Mars 2020 rover is expected to produce a similarly large suite of spectra. Working with LIBS spectra and deriving quantitative chemical abundances requires analysis methods beyond what is typically used for orbital VNIR spectra, and thus we have focused on those methods for our point spectra development. However, the methods described here are not unique to LIBS data, and we hope that by making the PyHAT library and graphical interface available to the planetary community, they can be applied to a wider variety of data sets.

We leverage existing Python libraries whenever possible. The PyHAT GUI uses PyQt5, and the data manipulation relies on numpy and Pandas. Many of the underlying algorithms used for point spectra analysis are from the scikit-learn machine learning library.

The PyHAT GUI is primarily designed to work with a simple CSV format, but also includes the capability to read ChemCam data in PDS format. The GUI includes a tool for looking up metadata from a separate CSV file, matching the metadata to the spectra based on the values in user-specified columns. It also includes data manipulation utilities such as removing spectra, combining two sets of spectra into one, and splitting one data set into many based on the unique values of a user-specified column. We use scikit-learn algorithms for Isolation Forest and Local Outlier Factor outlier removal. Data can also be resampled onto a new set of wavelengths, which is useful when working with spectra from multiple instruments.

A baseline removal interface allows users to choose from nine different continuum removal algorithms [3] to find one that best suits the data at hand. There is also the option to apply a "peak-binning" method that can reduce the size of spectra, speeding up calculations while improving calibrations that rely on faint emission lines [4].

Data can be masked, normalized, or multiplied by a vector. The GUI also includes a tool to replicate spectra with specified shifts in wavelength, which is useful for assessing the robustness of certain techniques to variations in wavelength calibration. The interface also allows users to divide data into multiple folds, stratified on a single metadata column, for use in cross validation.

In most cases, spectroscopic data have high dimensionality (number of spectral channels) and collinearity (the channels are highly correlated). PyHAT includes a dimensionality reduction tool that acts as a "wrapper" for a number of algorithms from scikit-learn and elsewhere [5], enabling users to transform data for visualization, clustering, or other analyses. K-means and spectral clustering are currently available, and we plan to add more algorithms in the near term.

The GUI also includes a set of tools for implementing and evaluating multivariate regression models of the type used to derive quantitative chemistry from ChemCam spectra [6]. These include a cross validation interface for identifying the optimum parameters for a regression, as well as the ability to train individual models and use them to predict results from new data. Individual "sub-models" can be trained on restricted composition ranges and then blended into a single model that can better handle diverse target types, as described in [7].

We have developed and implemented a "Local LASSO" method, inspired by [8], but taking advantage of the internal cross-validation and optimization capability of the scikit-learn LASSO algorithm. This method generates a new regression model for each unknown spectrum rather than attempting to use a single model to predict all spectra.

The GUI also includes the capability to plot columns or rows from the data with customizable colors, symbols, opacity, etc. There is also a pre-defined plot

function for generating score and loading plots after running dimensionality reduction methods such as principal component analysis (PCA) and independent component analysis (ICA).

Planetary Analog spectra: We collected spectra of a suite of pressed-powder planetary analog targets using the LIBS instrument at Johnson Space Center. We analyzed 133 targets made from the powders of 85 well-characterized rock samples from the JSC planetary analog collection. The targets were analyzed using three different laser energies to provide data useful for assessing the influence of laser energy density on LIBS spectra. A subset of 30 of the targets analyzed at JSC were also analyzed by the ChemCam engineering model at Los Alamos National Laboratory to provide a basis for comparison between instruments and evaluation of calibration transfer algorithms.

The laboratory data will be made available as CSV files in a format compatible with the PyHAT software at https://astrogeology.usgs.gov/data/pyhat-laser-induced-breakdown-spectroscopy-mars-analog-dataset.

Future work: We will continue to work to add new capabilities to PyHAT and improve the stability and usability of the point spectra analysis GUI. We will finish testing calibration transfer algorithms in PyHAT and provide access to those methods through the GUI so that users can transform data collected under different conditions or on different instruments to be compatible. We also plan to implement additional clustering methods from scikit-learn. Cross-validation and baseline removal are among the most time-consuming capabilities of PyHAT; we will work toward parallelizing these processes to take advantage of systems with multiple processor cores.We also plan to develop a detailed user guide, including example workflows and tutorials using the LIBS datasets described above.

Development of PyHAT will continue beyond the end of the current grant. A recently funded grant, detailed in [9] will add a variety of algorithms commonly used by the terrestrial remote sensing community to PyHAT, and enable ongoing improvements and user support. We hope that PyHAT will enable scientists to minimize time spent developing their own software and maximize the scientific results that can be extracted from increasingly large and complex spectral data sets.

References:[1] https://github.com/USGS-Astrogeology/PyHAT [2] Gaddis et al., This meeting. [3] Giguere, S., Carey, C.J., Boucher, T., et al. (2013) Proc. 5th IJCAI Workshop on Artificial Intelligence in Space [4] Clegg et al. (2018), 49th LPSC, #2576. [5] https://www.gbeckers.nl/pages/numpy_scripts/jadeR.py [6] Clegg et al., (2017) Spectrochim. Acta. B, 129, 64-85. [7] Anderson et al. (2017) Spectrochim. Acta B., 129, 49-57. [8] J.S. Shenk, et al. (1997) J. Near Infrared Spectrosc. 5, 223–232. [9] Aneece, et al., This meeting.

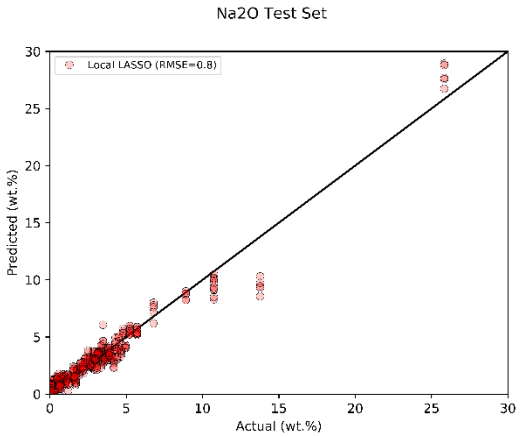

Figure 1: Example result of "Local LASSO" regression for quantifying Na_2O using the ChemCam database.

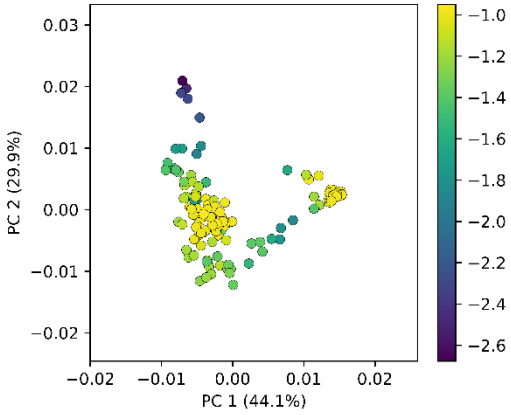

Figure 2: PCA scores plot of a subset of the JSC data set, with color corresponding to Local Outlier Factor (darker colors are more likely to be outliers).

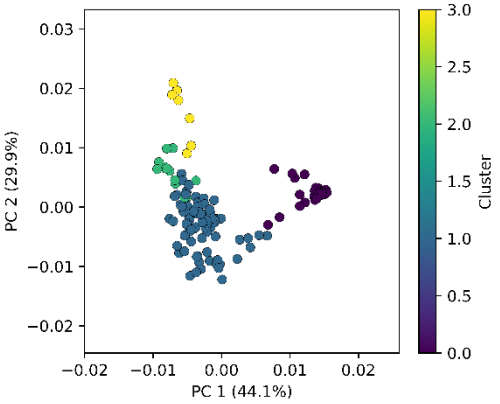

Figure 3: PCA scores plot with color corresponding to K-means cluster (k=4).