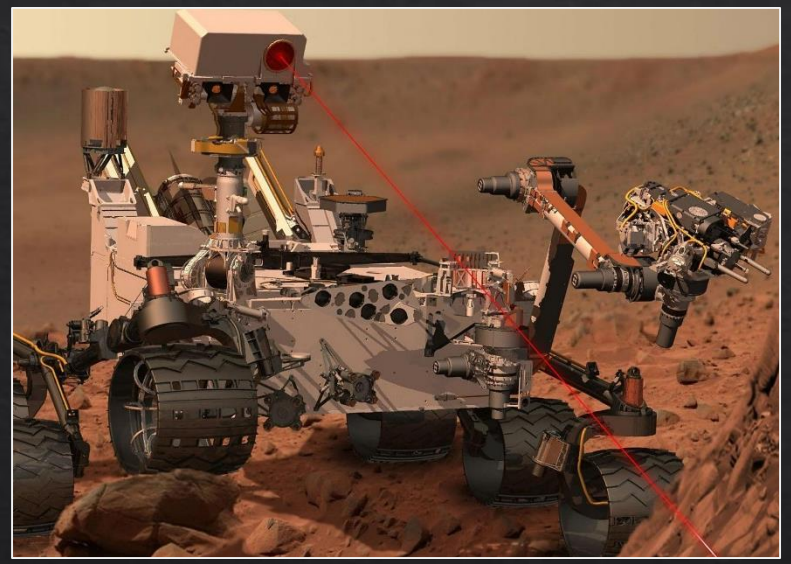
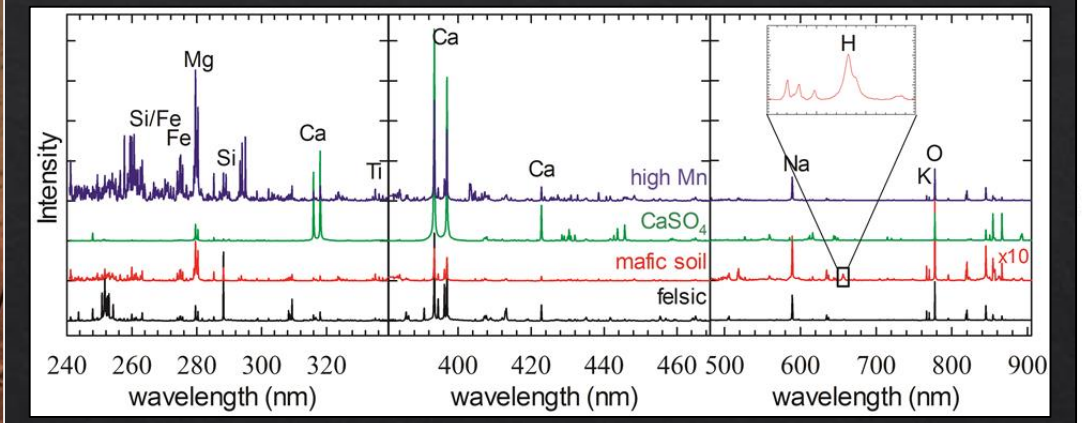
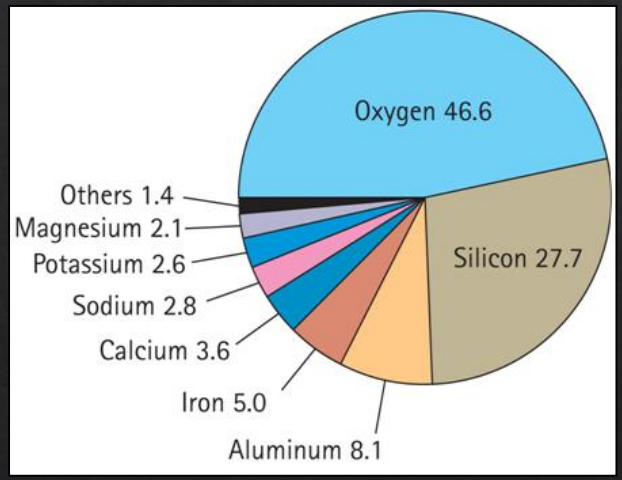
The Python Spectral Analysis Tool (PySAT): Powerful, Flexible, and Easy Preprocessing and Machine Learning with Point Spectral Data







Ryan Anderson[1], Nicholas Finch[1], Sam Clegg[2], Trevor Graff[3], Dick Morris[3], Jay Laura[1]

[1]USGS Astrogeology, [2]Los Alamos, [3]NASA JSC

What is PySAT?

- A Python library for spectral analysis
- Two parallel PySAT projects led by USGS
  - Gaddis, Laura et al.: orbital spectrometers (e.g. M3, CRISM, etc.)
  - Anderson et al.: point spectrometers (e.g. ChemCam, SuperCam, etc.)
- Shared back-end:
  - https://github.com/USGS-Astrogeology/PySAT
- Point Spectra GUI:
  - https://github.com/USGS-Astrogeology/PySAT_Point_Spectra_GUI

Why is PySAT Necessary?

- Interpreting spectral data is vital, but difficult.
  - Even for instrument team members, it is hard to test new processing and analysis methods.
- Scientists end up writing their own code.
  - Often reinvent the wheel
  - Often end up using simple analysis methods out of necessity
- Commercial options are often expensive, proprietary, inflexible.
- We want to remove these barriers.

PySAT Point Spectra GUI

- Python-based tool for preprocessing and analyzing point spectra
  - Free
  - Open-source
  - Powerful
  - Flexible
  - User friendly

- We want to enable planetary scientists to process and analyze point spectra without specialized programming expertise.
  - PyQT5-based GUI
  - Leverage scikit-learn to provide machine learning methods

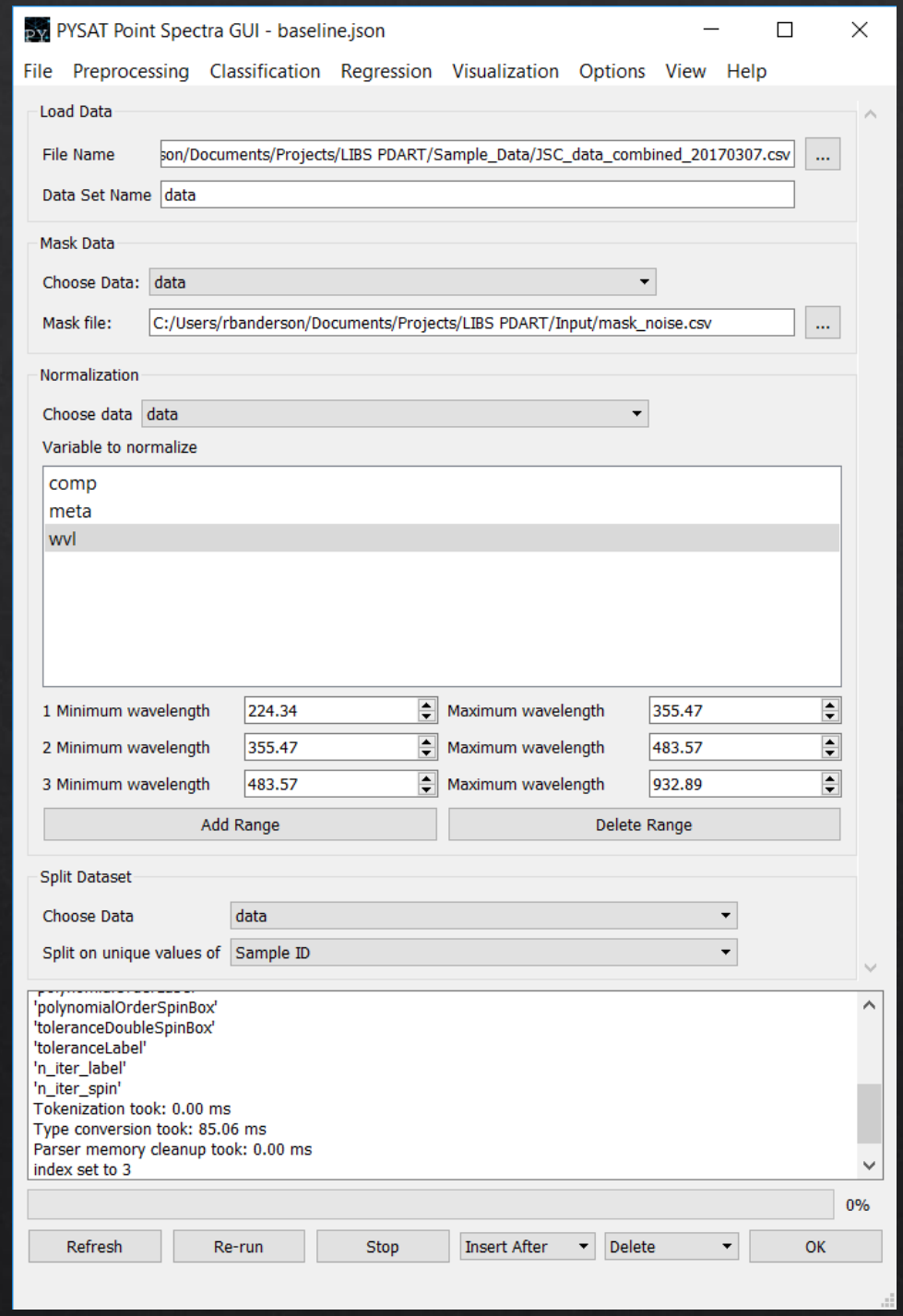
Data Format

- .csv files with dual-level column labels (read into a multi-indexed Pandas data frame)
  - The tool can read ChemCam "CCS" data on the PDS into this format.
- Standard top-level labels are:
  - "meta" = metadata
  - "comp" = compositional data
  - "wvl" = spectral data
- Others are added to record additional information (e.g. PCA scores, regression predictions, etc.)

meta	meta	meta	comp	comp	comp	wvl	wvl	wvl	wvl
Sample	XRF #	Sample ID	SiO2	TiO2	Al2O3	224.336	224.391	224.446	224.501
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	-23.17	-23.17	-23.17	3.83
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	11.17	11.17	11.17	-23.83
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	7.67	7.67	7.67	-7.33
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	7.94	7.94	7.94	-16.06
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	23.5	23.5	23.5	14.5
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	2.78	2.78	2.78	-2.22
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	6	6	6	8
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	19.11	19.11	19.11	10.11
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	-11.56	-11.56	-11.56	10.44
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	-3.61	-3.61	-3.61	11.39
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	-5.39	-5.39	-5.39	23.61
CP-5, Basalt / AREF019	JSC-0270	JSC1416	54.06	2.27	13.17	-6.28	-6.28	-6.28	23.72

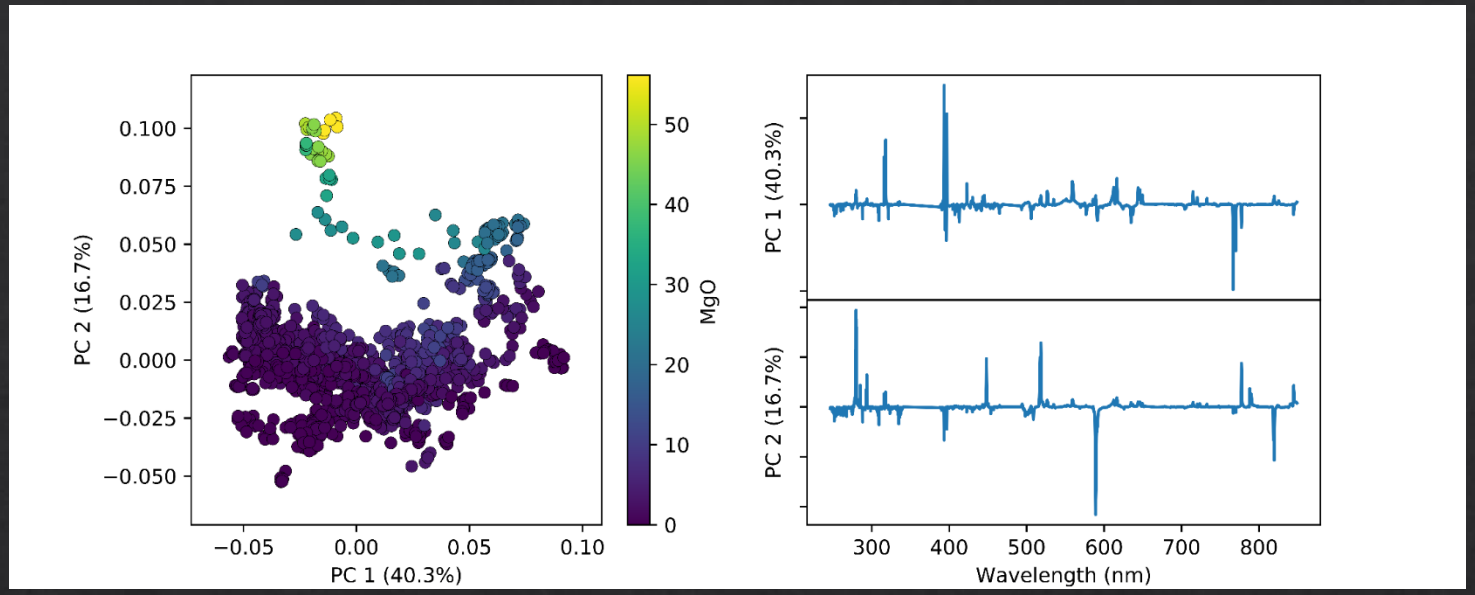
Interface and "Workflows"

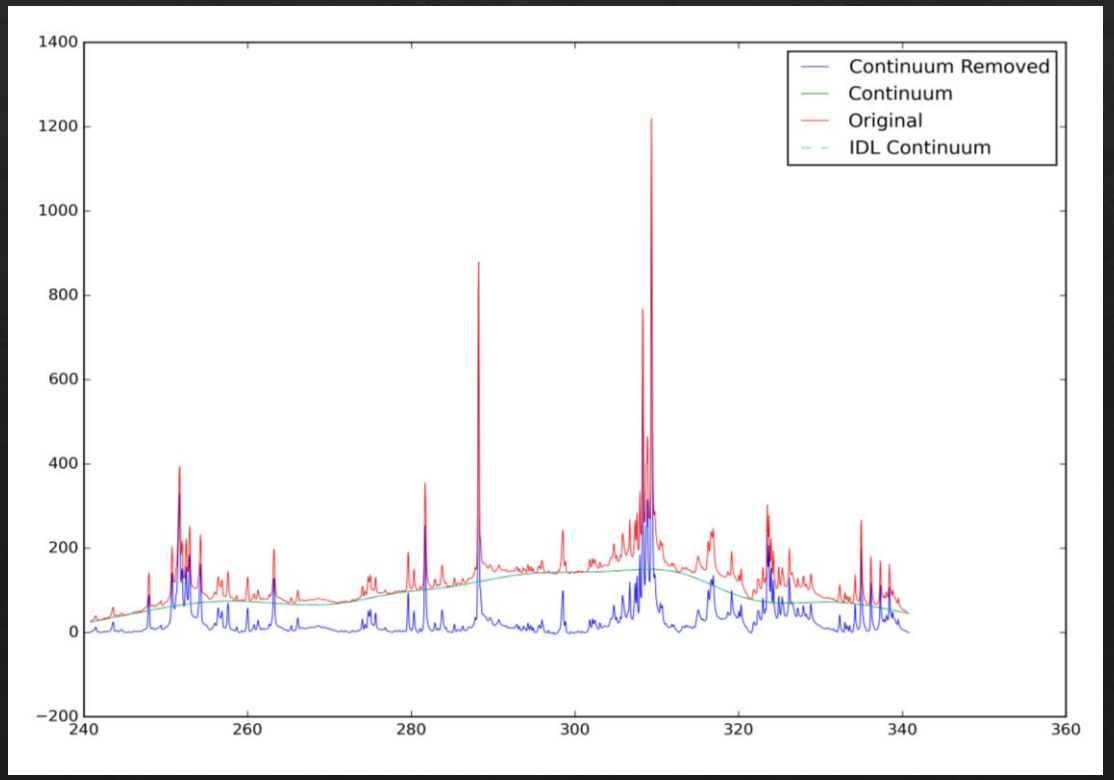
- The PySAT GUI is modular for maximum flexibility:
  - User can specify different steps in whatever order desired (within reason)
- We call these steps "modules"
- We call a series of steps a "workflow"
- Modules can be inserted and deleted
- Information can be passed from one module to the next without having to run the workflow
- Workflows can be saved and restored using .json



Data Transformation

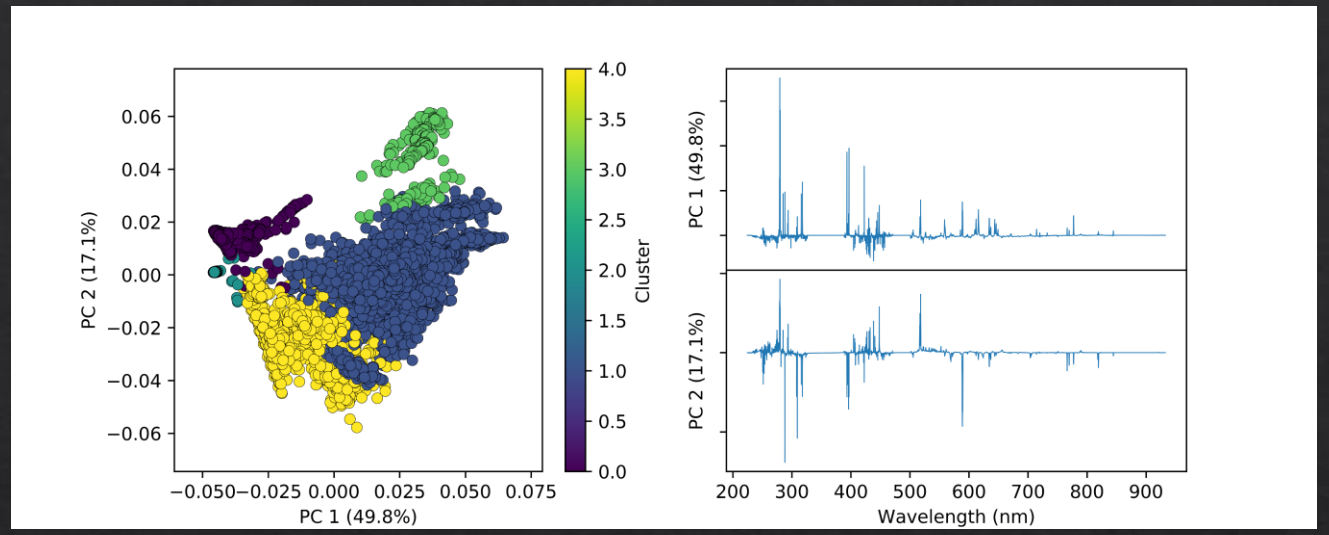
- Interpolate data onto new wavelengths
- Masking
- Normalization
- Derivative
- Multiply by vector
- Peak Area Binning
- Baseline removal (9 algorithms available)
- Dimensionality reduction
  - PCA, ICA, t-SNE, LLE

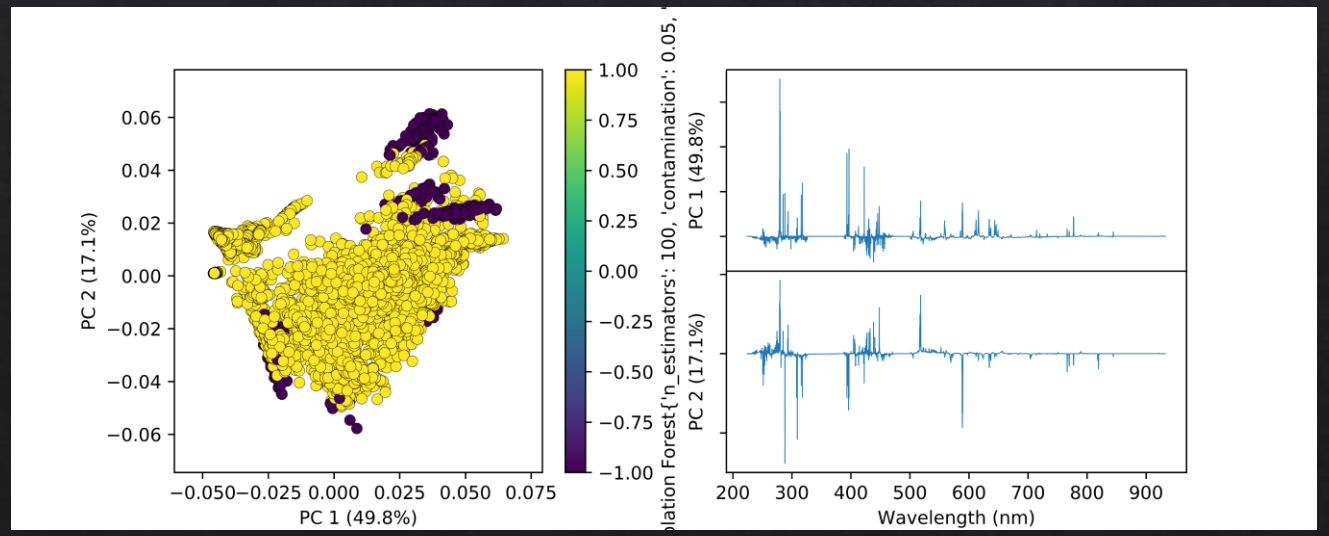




Data Manipulation and Grouping

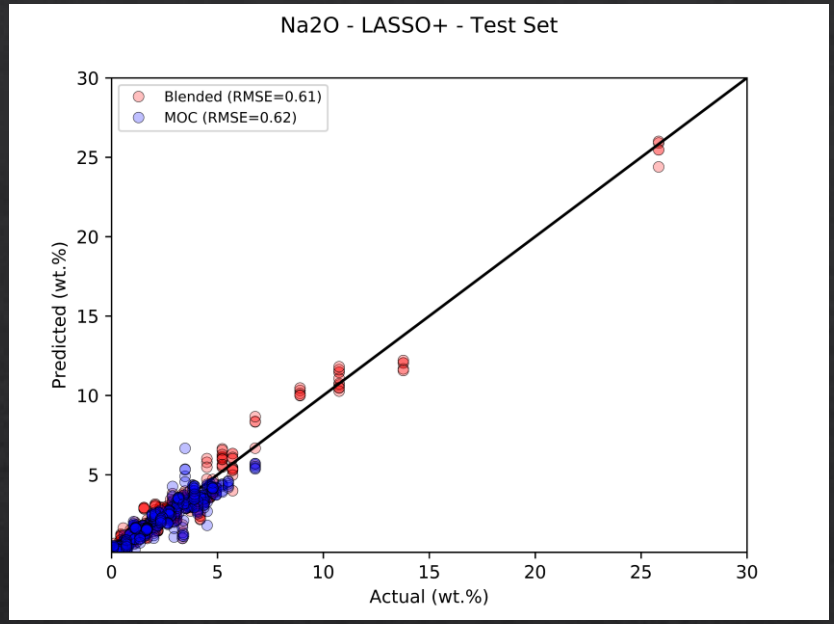
- Data Manipulation
  - Remove rows
  - Split data
  - Combine data sets
- Outlier removal
  - Isolation Forest
  - Local Outlier Factor
- Clustering
  - K-Means
  - Spectral
  - More to come
- Stratified Folds

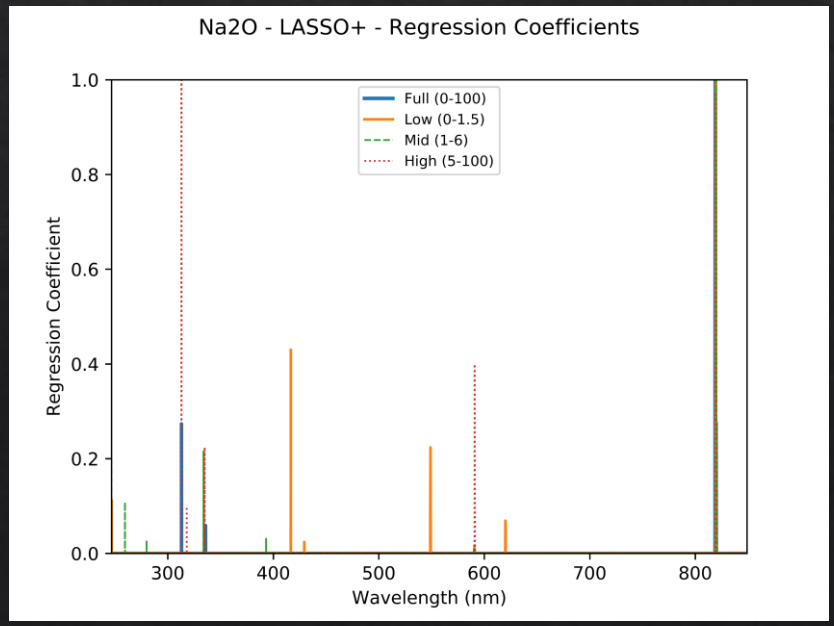




Regression

- K-fold cross validation over any parameters
- Multivariate regression:
  - Ordinary Least Squares (OLS)
  - Partial Least Squares (PLS)
  - Gaussian Process (GP)
  - Support Vector Machines (SVM)
  - Bayesian Ridge Regression (BRR)
  - Lasso
  - Elastic net
  - Orthogonal Matching Pursuit (OMP)
  - Least Angle Regression (LARS)
  - Automatic Relevance Determination (ARD)
- Blend “sub-models”
- Save/restore trained models





Conclusion

- PySAT point spectra tool is a powerful and flexible spectral processing and regression software
  - Let scientists spend their time analyzing data rather than writing code.
- Future work
  - Calibration transfer
  - Additional clustering and classification
  - Additional regression
    - Ensemble methods (e.g. bagging, boosting, etc.)
    - Local methods
  - Better plotting
  - Debugging and testing
  - Documentation
- Although designed with LIBS in mind, it is flexible enough for most spectral work (and even suitable for some non-spectral data!)
  - Interested, but not sure how to use PySAT for your work? **Let's talk!**
- Is PySAT missing some capability you would use? **Let me know!**

PySAT Point Spectra Tool

Back end:
https://github.com/USGS-Astrogeology/PySAT

GUI:
https://github.com/USGS-Astrogeology/PySAT_Point_Spectra_GUI

Contact me:

Ryan Anderson
rbanderson@usgs.gov