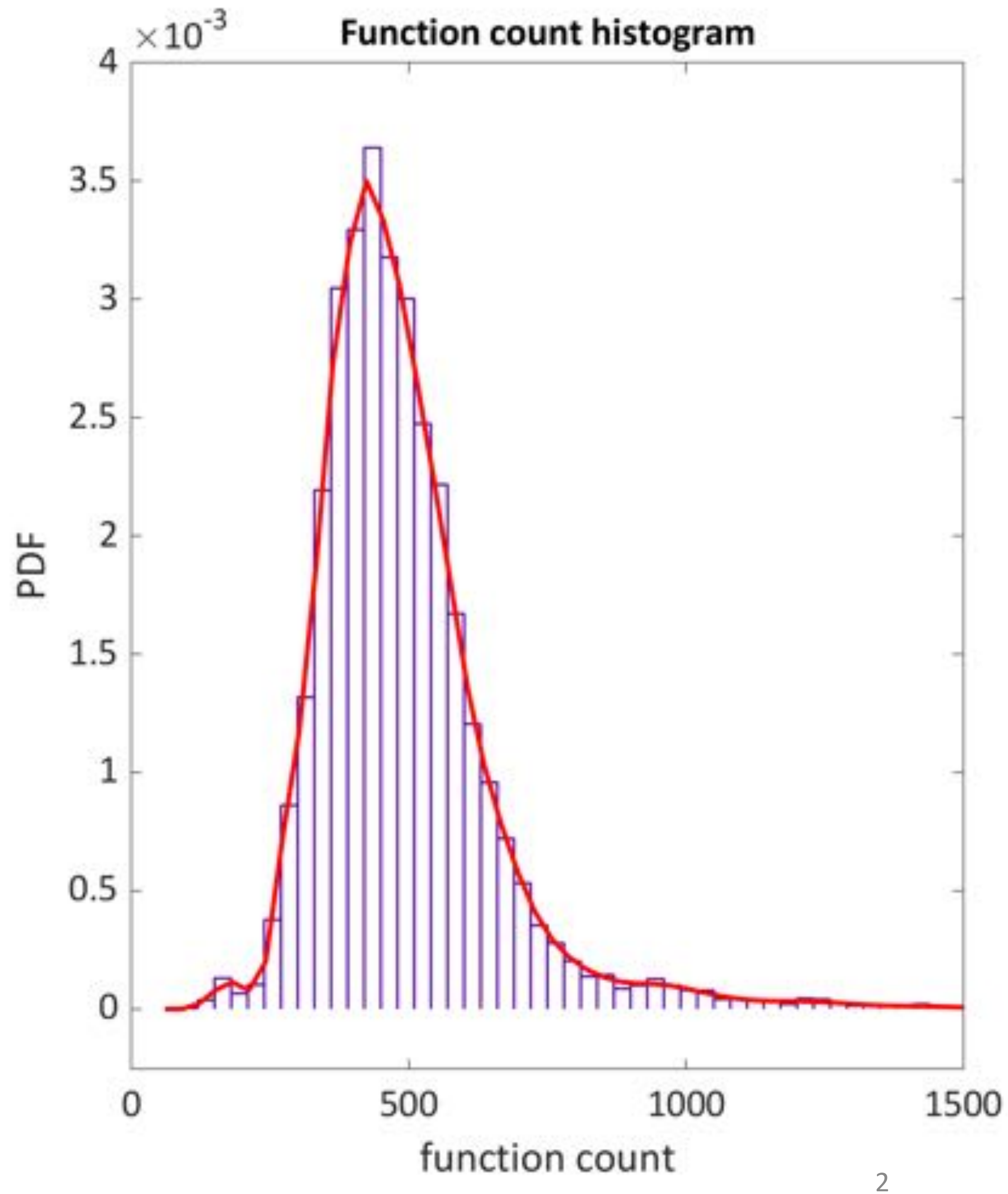
Strategies to Employ Machine Learning to Scale Up Algorithms in Imaging Spectroscopy

CHIME/SBG meeting
2022 Oct 19-21

Jeff Dozier@ucsb.edu

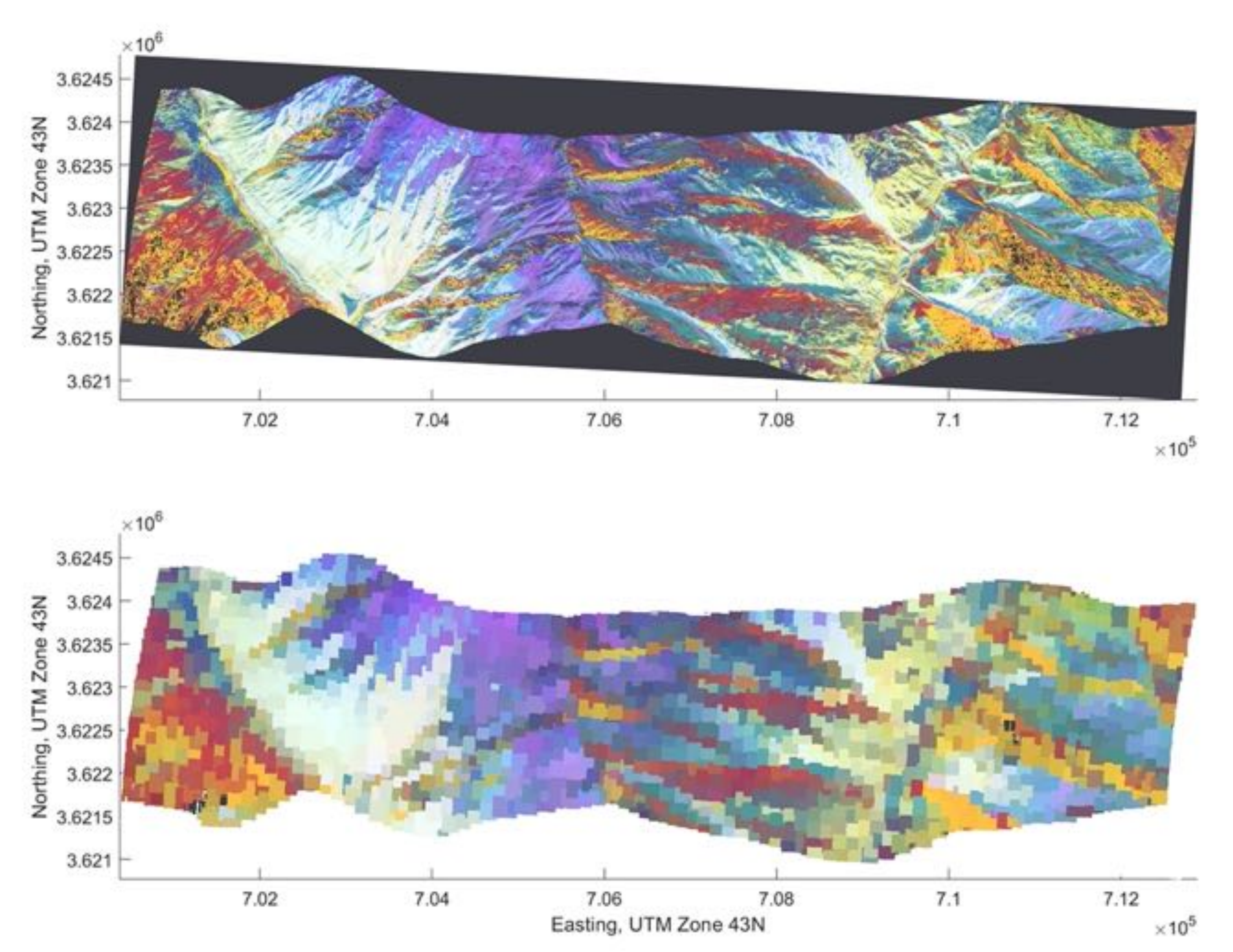
Typical computation

- Forward radiative transfer model that calculates spectrum = f(x), a set of biogeophysical properties
- Assume initial set of properties x_0
- Iteratively, compare f(x) with measured spectrum
- Adjust x, typically along a gradient f’(x), which requires multiple calculations of the model function
- Until f(x) matches measured spectrum based on some minimization criterion (Euclidean norm of residuals, spectral angle, . . ., custom)
- Lots of function evaluations, example →



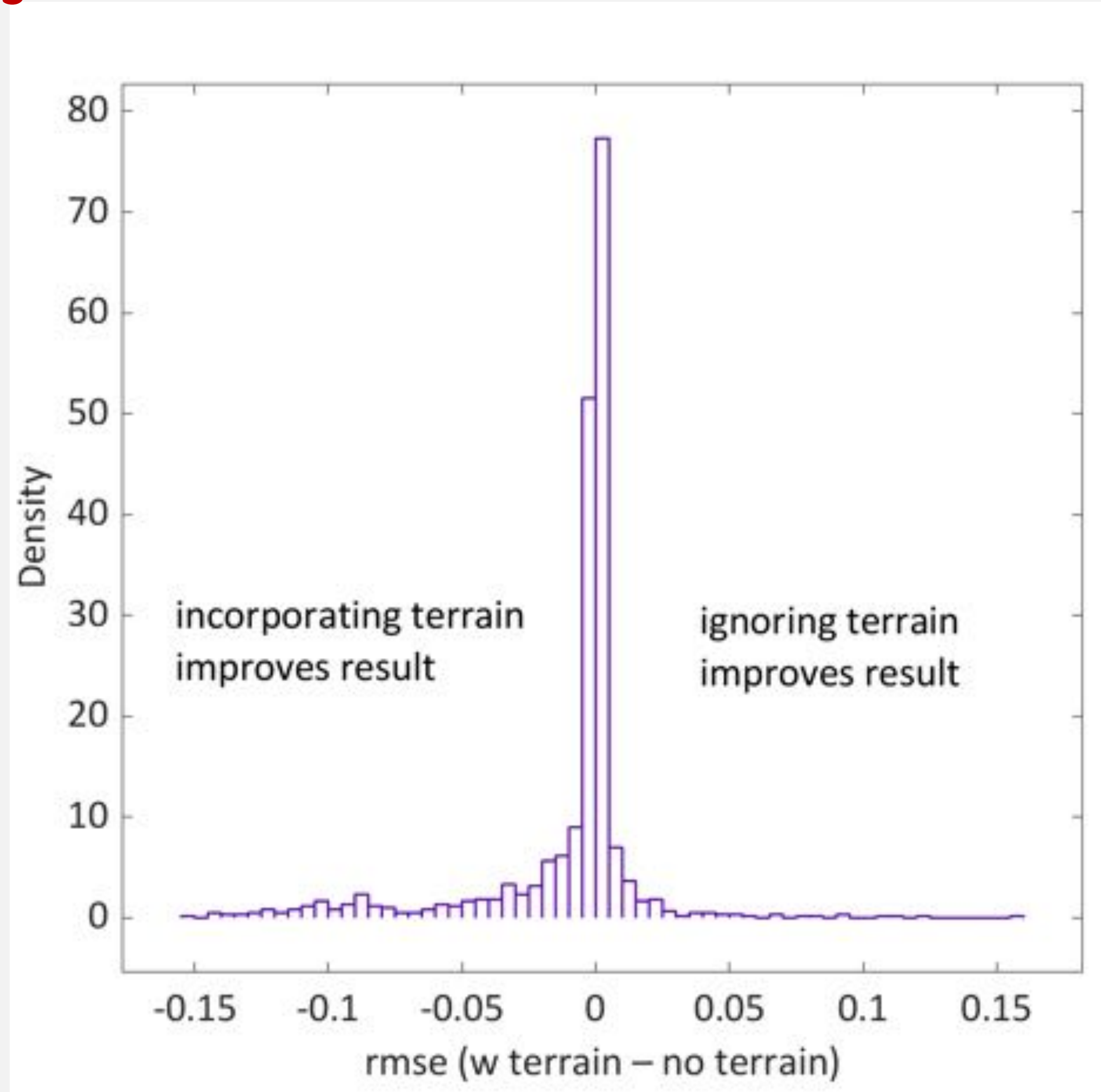
Train on a sample of pixels

- Superpixels to sample the image (0.1% sample)
    - Or use “uniquetol” to identify similar pixels
- For each sample element, invert reflectance model to estimate properties of interest
- Use those to train a machine learning model
- Improve over time with subsequent images



Retrieve superpixels' snow properties

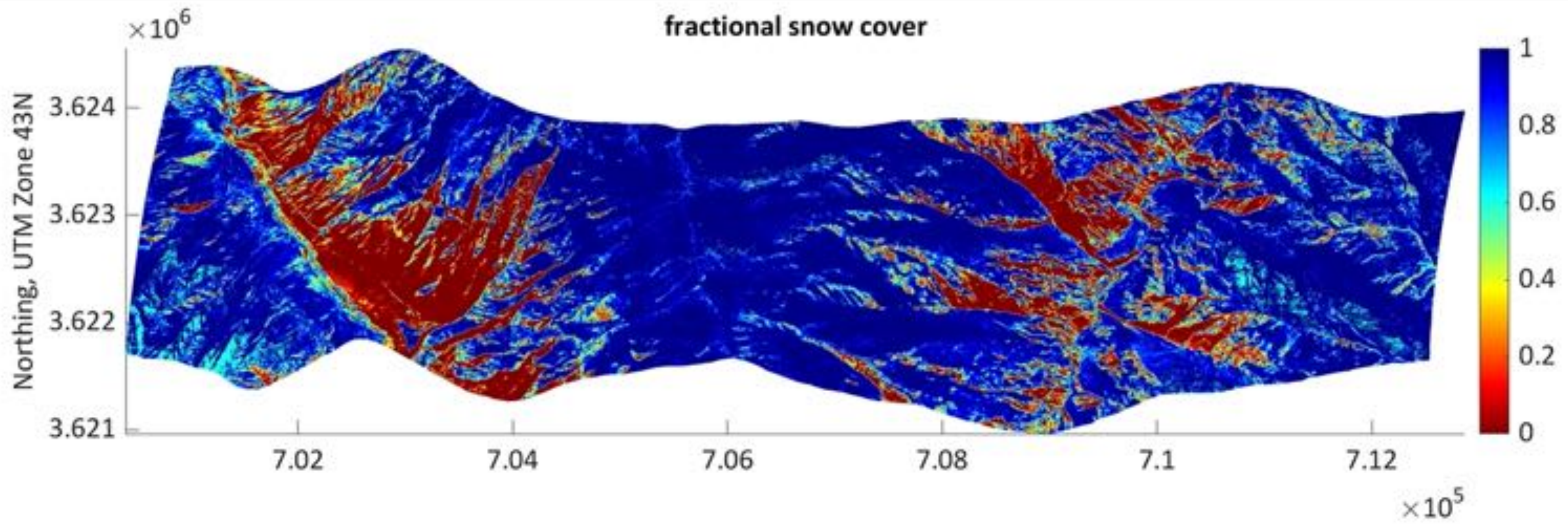
- Eliminate superpixels where scaled 1030-nm absorption absent, hence no snow
- Set aside pixels with $\cos\theta_S < \cos 80°$ (weak reflectance in wavelengths sensitive to grain size)
- Find snow properties in each superpixel that minimize difference with measurement, ***with and without considering terrain***
- Assess based on 4 comparisons with measurement: RMS error < 0.07, spectral angle < 10°, $R^2 > 0.9$, grain size reasonable
- Caveat: validation mostly through simulation



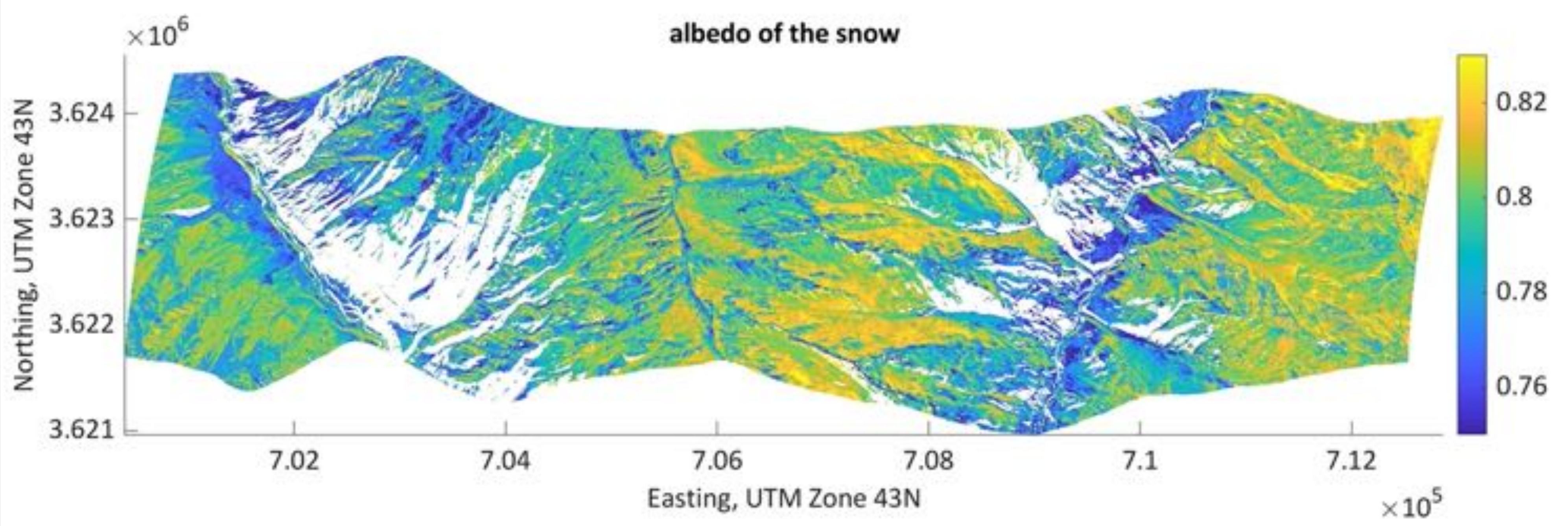
From superpixels (or sample pixels) to the whole image

- With the superpixels or sample pixels, consider topographic properties (elevation, illumination, view factor) and reflectance values (pixel albedo, scaled absorption, weighted spectrum) to develop statistical relationships (machine learning) with properties of interest (e.g., f_{SCA}, snow albedo, grain size, LAP fraction)
- Apply those relationships to the whole image
  - Generally, I find that the selection of training data makes as much of a difference as the choice of the machine learning method
  - (My friends in Computer Science don't like to hear this)
  - For the example, I used Gaussian Process Regression

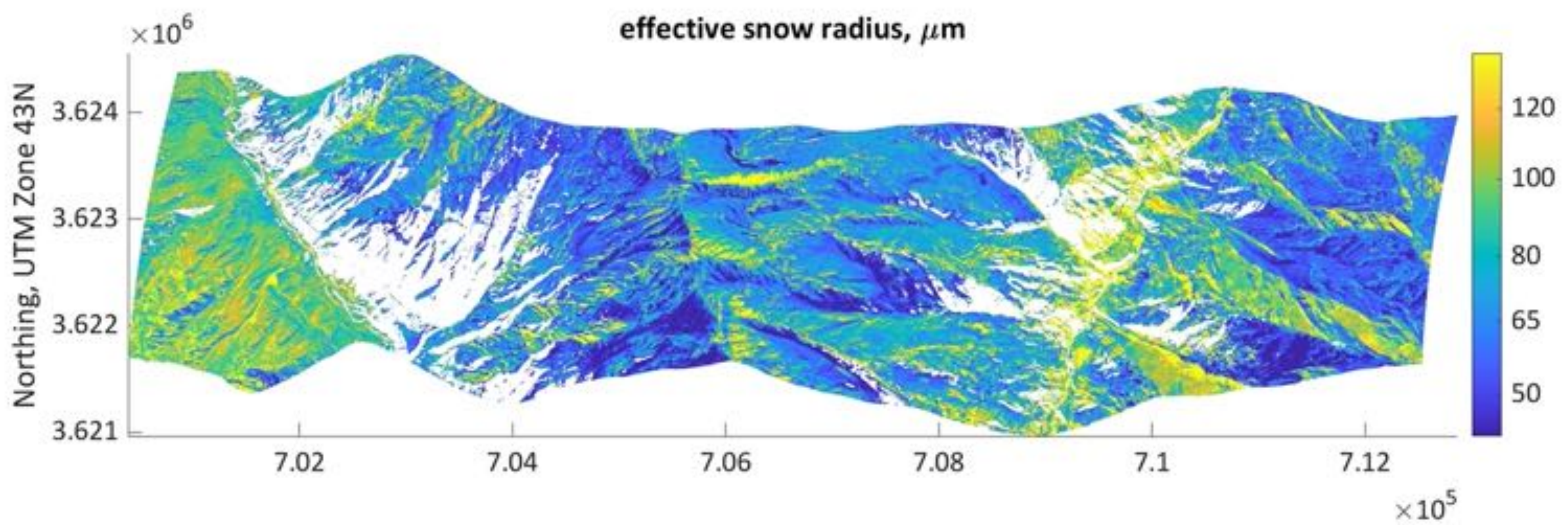
f_{SCA} and snow albedo, Indian Himalaya, Feb 17 2016

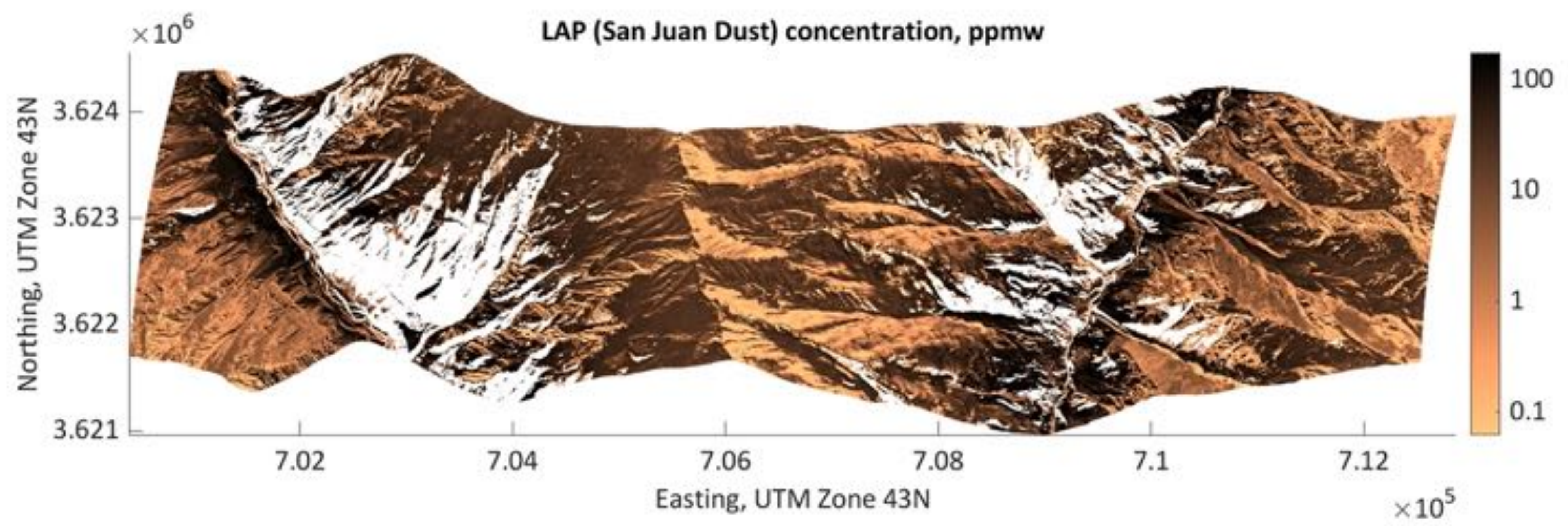


(albedo range display from 5^{th} to 95^{th} percentile)

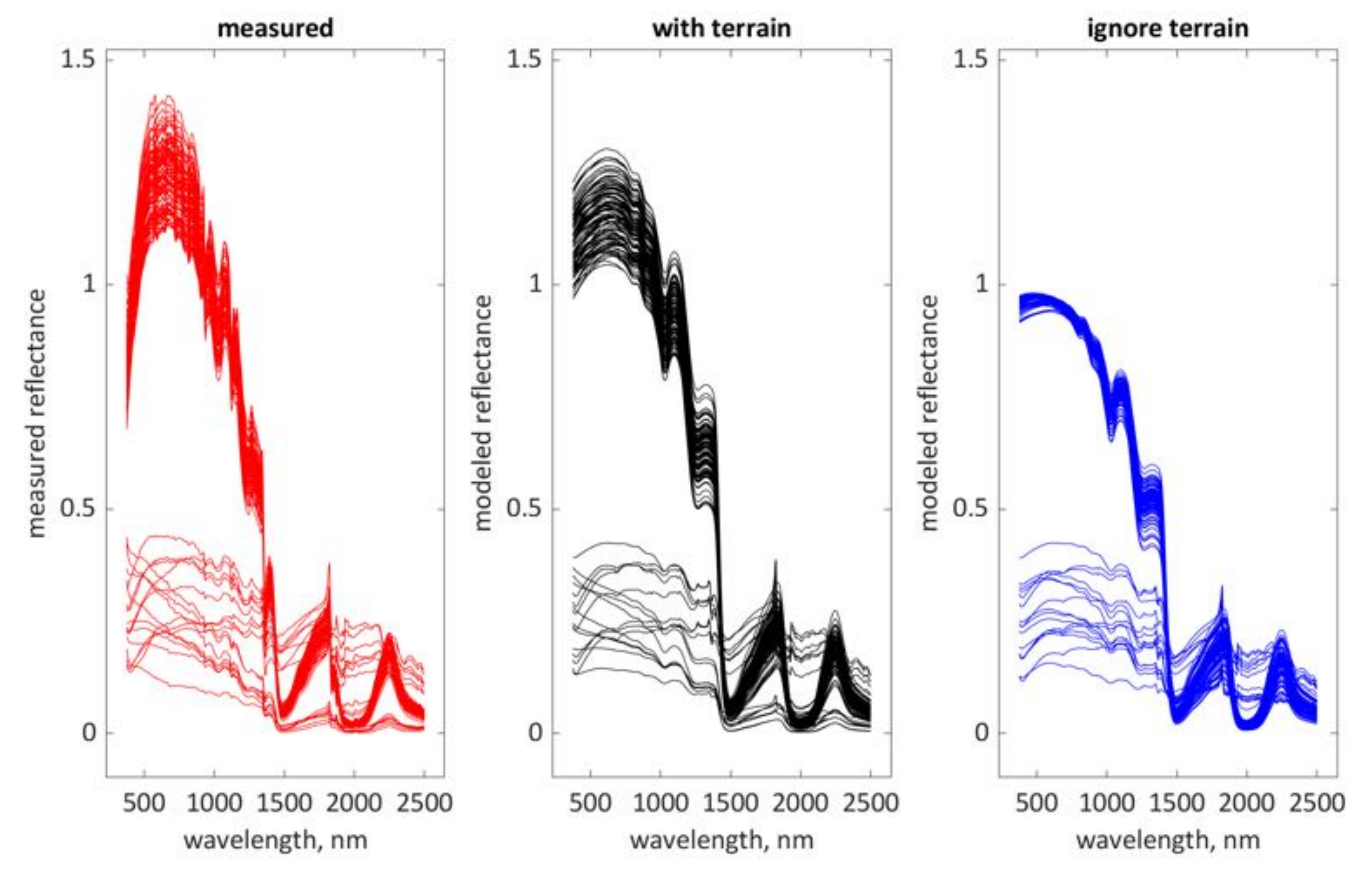


Snow grain size (µm) and LAP concentration (ppmw), Feb 17 2016

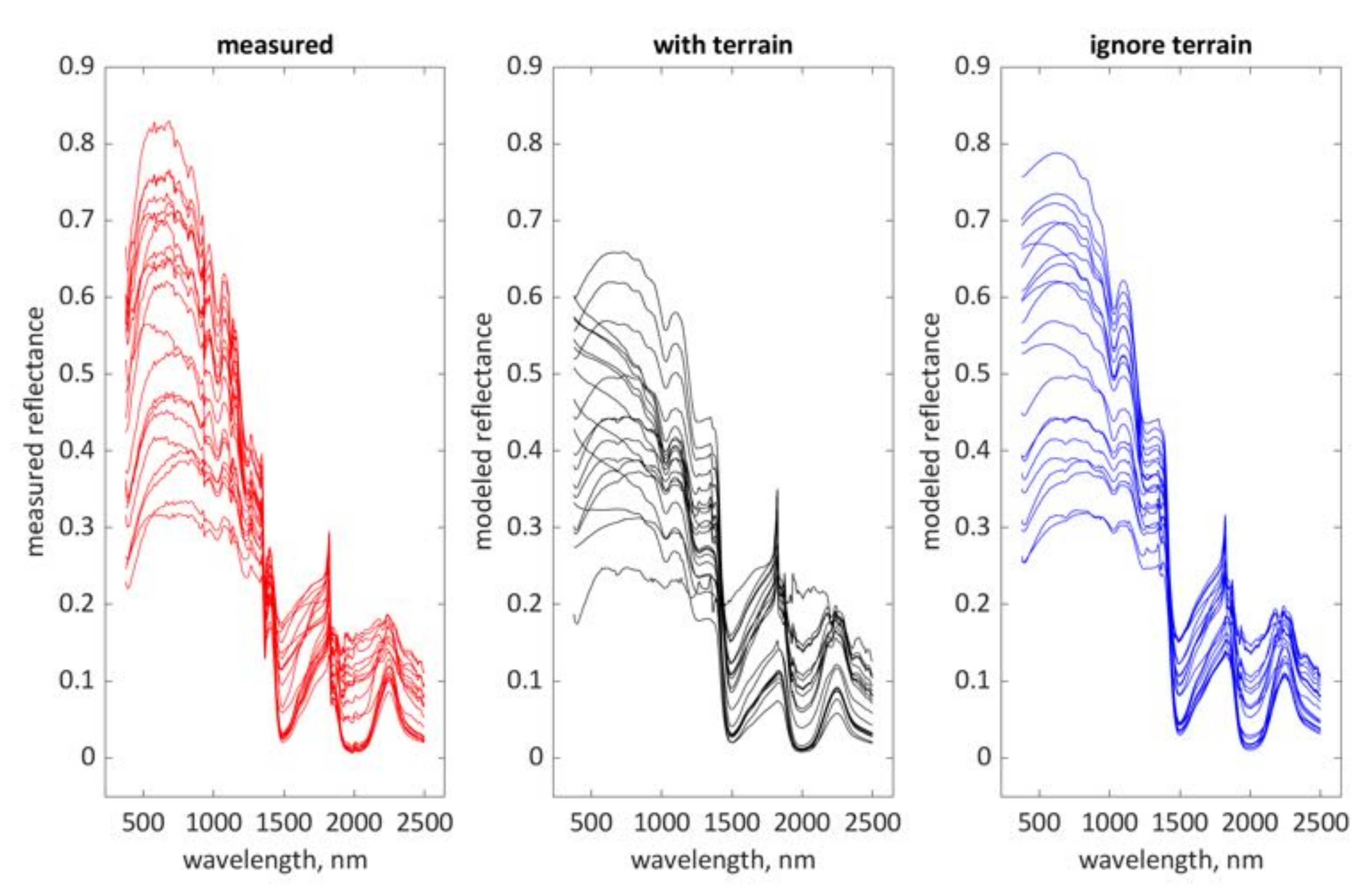




Better
with
terrain



Best ignore terrain

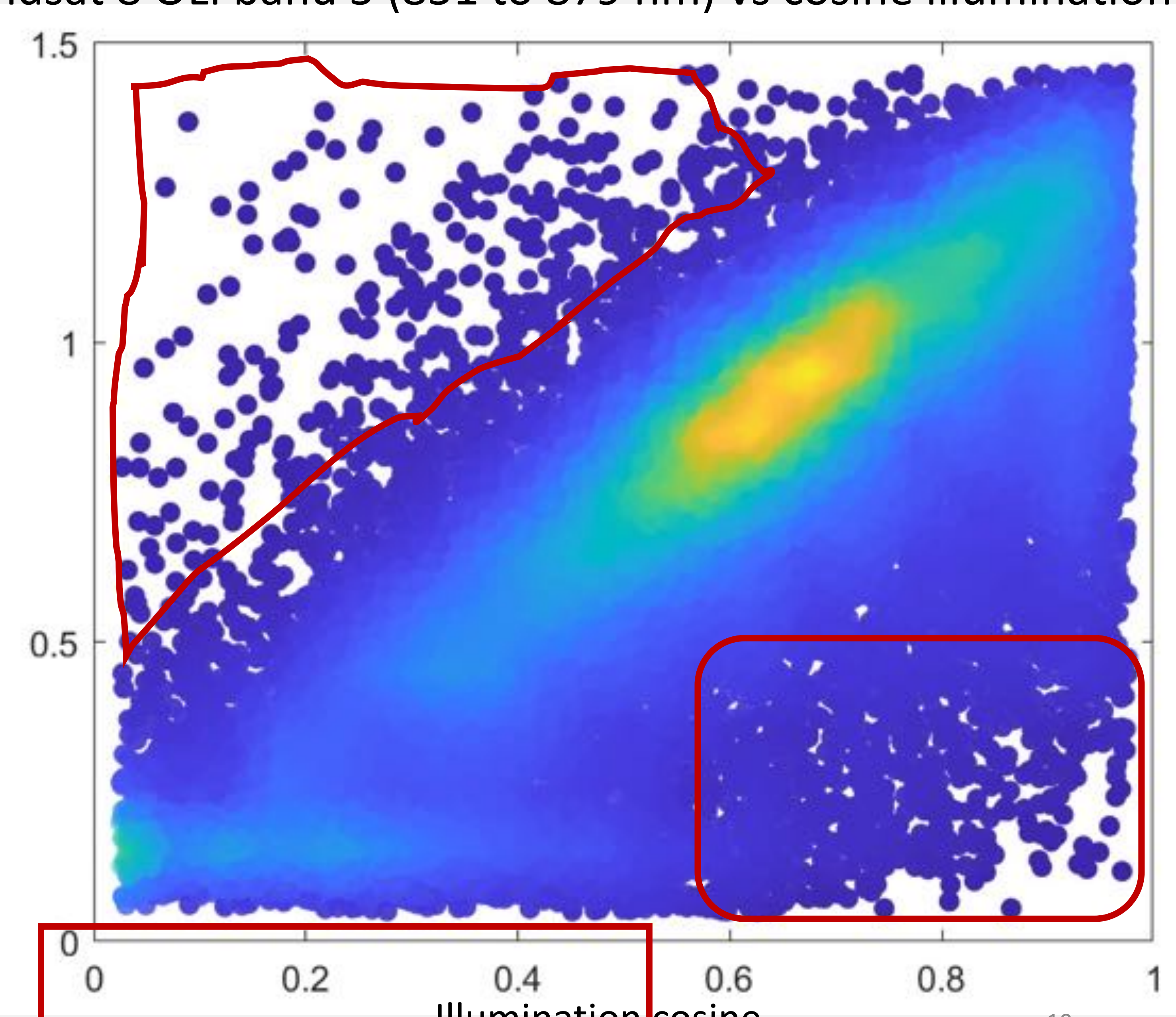


Why does ignoring some physics sometimes lead to better results?

Because we understand the physics but maybe not the geometry and its errors

Another machine learning problem: identify pixels with “unlikely” reflectances

Landsat 8 OLI band 5 (851 to 879 nm) vs cosine illumination



Illumination cosine

Conclusions

- Modeling the effect of the terrain—illumination, sky view factor, and multiple re-reflection in topographic hollows—helps
  - but only if illumination angles and view factors are about correct
  - and many are badly incorrect, especially in globally available DEMs
  - so we must be careful about how we consider the terrain
  - and we must deal with roughness (sub-pixel topography)
- Superpixels enable sampling of terrain to provide training sets for machine learning, avoid computational intensity over entire images
  - “uniquetol” is another sampling option, but superpixels also smooth noise so they’re a good option