Question & Answer Session, Part 2

Please type your questions in the Question Box. We will try our best to answer all your questions. If we don't, feel free to Zach Bengtsson (bengtsson@baeri.org), Amber McCullum (amberjean.mccullum@nasa.gov), or Juan Torres-Pérez (juan.l.torres-perez@nasa.gov).

Question 1: How do we know of all the in-built functions in GEE?
Answer 1: You can use the "Docs" tab in the GEE JavaScript API. This tab allows you to click through or search all of the functions available in GEE. You may also wish to reference the API guides found here generated by the GEE developers: https://developers.google.com/earth-engine/guides

Question 2: For binary classifications, should I provide training samples for both classes?
Answer 2: Yes, we would recommend providing training data for any classes you would like to identify. This improves the algorithm's ability to differentiate between classes, making your classifications more accurate.

Question 3: Would you need to use polygons for training data? Or what sort of format is needed?
Answer 3: You can either use polygons or points for your training data. But these geometries must have known land cover types. We used point data for the exercise to better differentiate between coniferous and deciduous tree pixels.

Question 4: For supervised classification if we zoom in to look at the pixels, many would have an overlapping colour or some mixed colour, how do we ensure that each pixel gets classified correctly into the classes ?
Answer 4: In today's session, each pixel is definitively assigned to a specific class. If you are referring to speckling or pixels of different classes being grouped together in a way that appears to have too much variation for that segment of the landscape, there are methods to smooth out classifications and reduce this speckling. One option is to use the image.unmix() function, which completes spectral unmixing. You can find more about this function here: https://developers.google.com/earth-engine/apidocs/ee-image-unmix

Question 5: If you have a large set of files, is it possible to upload to GEE without having to do each one individually?
Answer 5: I believe it depends on the type of file and how your data is organized within a file structure. We typically recommend uploading distinct geometries and image collections separately, just because this allows you to share them individually and include them in your script on a case by case basis.

Question 6: With Landsat dataset, how to set the date range in GEE to compute a yearly time series Land use/land cover mapping?
Answer 6: Hopefully we addressed some of this question in the code activity, but you can filter date range for the landsat dataset according to your specifications. For example, you could create images that average the highest quality pixels for all summer months of each year from 2015 to 2020.

Question 7: Can you explain a little bit more about this idea of a "black box". How are the algorithms used unknown to us?
Answer 7: Apologies if my description in the presentation was unclear! We know what the algorithm is and what it is doing, as we saw in the slides describing how the random forest algorithm works, but we aren't able to observe the process as it happens. Which means that we observe its performance by examining the accuracy (and thereby success) of its outputs. Hopefully that clears this up a little!

Question 8: Which classifier is better to identify a single crop? For the crop type mapping, can we follow the same code and same methodology given in this webinar, or should we consider including different data? What are the key points for crop type mapping?
Answer 8: There isn't a perfect answer for this question. It depends on what data you have available for training and what statistical approach works best for you. In GEE, you could attempt this classification in a few chosen algorithms, like RandomForest, CART, or SVM, assess the accuracy of each, and then choose the one with the best performance. If you want to classify crop types, you'll need to gather training data for each type of crop. Each crop type would become a separate land class. This can be difficult though because some crops look very similar to one another in satellite imagery. You might want to look into the use of phenology and pre-existing land use data to differentiate between crops.

Question 9: How do we know which type of classification is best for a particular area? When would you use the different types of classifiers: Random Forest vs. CART vs.

SVM and Native Bayes? Any reference documents we could further explore on this topic? Is there a paper that assess the fit for different classification models?
Answer 9: Each type of classifier has its own pros and cons, so it's definitely on a case-by-case basis to know which one to select. This article here looks at the classifiers in GEE and how they classify multi-temporal satellite imagery for crop mapping, which you may find useful (also useful for Q8):
https://www.frontiersin.org/articles/10.3389/feart.2017.00017/full

Question 10: With the Landsat dataset, how do we set the date range in GEE to compute a yearly time series Land use/land cover mapping?
Answer 10: Refer to Q6

Question 11: What are your suggestions to solve the memory limit error when running ensemble ML algorithms such ee.Classifier.smileGradientTreeBoost over extensive land areas?
Answer 11: Working around scaling errors can be difficult, but one way is to export the imagery you're using for the algorithm and import it as an Earth Engine Asset. This ensures processing is occurring on a single image, rather than a GEE composite or mosaic type (which often require more processing power). You can also increase the scale of the ML algorithm, but keep in mind how that may affect your results.You could ask for more memory from Earth Engine, or you could split your imagery into sections that process without a memory limit error and mosaic them back together afterward.

Question 12: Can we tune the models on GEE?
Answer 12: There are many options within GEE classifiers that give you the option to control and change model parameters! Things like scale, number of decision trees, variables per split, and more can be specified within your code. Make sure you're looking at the documentation available for each of the classifiers to see how you can tune them:
https://developers.google.com/earth-engine/apidocs/ee-classifier-smilerandomforest

Question 13: How do we calculate the optimal number of training points required for an optimal classification?
Answer 13: We typically function using the principle that the more training data you can provide, the better. There is not a single guiding principle for the number of training points you should have. This is what makes accuracy assessment especially valuable. If your overall accuracy ends up lower than 80% using testing data that the algorithm has not seen before, you'll likely want to include more training data. Having somewhere

around at least 100 points for testing in your accuracy assessment is probably a good minimum standard.

Question 14: So what happens to those pixels that weren't supplied in the training data? Are they all classed into a separate class or left unclassified? Is there an algorithm that can be used to predict the spectral range beyond the training data?
Answer 14: Those pixels were withheld so we can use them later in our accuracy assessment. There are some algorithms that can specify beyond the spectral range, but there is a chance data can be misclassified. Take a look at all of the classifiers available in GEE here using this guide:
https://developers.google.com/earth-engine/apidocs/ee-classifier-smilerandomforest

Question 15: In the Random Forest classification, if it completely uses the training data, then how do we test its accuracy? Do we provide another image of a nearby area? For accuracy assessment, what if we do not have the reference data?
Answer 15: If possible, ground truthing can be a good way to test the accuracy of the classification. Additionally, you can split your training data into two groups for training and testing. It is up to you to decide how many points you want to include in the testing data, but typically 10% or 20% of the total amount of data is suggested. This means you could withhold 20% of your training data for testing and use the remaining 80% to train RandomForest. You can also use other land cover maps for reference data, but this might not be as accurate as you need if there are differences in your classifications. Toward the end of this example the GEE developers use the MODIS land cover product for reference data:
https://developers.google.com/earth-engine/guides/classification

Question 16: Can you pull in data APIs or Web Services? Or does all of your data need to come from GEE's available data library or uploaded directly?
Answer 16: Data needs to be in Google Earth Engine, whether through the data catalog or uploading personal data as assets.

Question 17: What do you mean by: for the Random Forest algorithm, training data must cover the entire spectral range? How can we ensure that the training data covers the entire range during image classification?
Answer 17: When we say “spectral range”, we were referring to the coverage of the thresholds of bands and brightness. Basically the range of pixel values for each band used in a classification. If you have knowledge of this range in your own data, you can ensure training points at the min and max are used to train the classifier.

Question 18: Is there any option within GEE to evaluate the sample set, to have an idea of what samples should be kept and which ones should be discarded to improve the classification results?
Answer 18: One way to diminish the influence of outliers is to have more reference data. The more training data you provide the algorithm, the less influence outliers will have on classification. You can also plot your training data to identify likely outliers and exclude them.

Question 19: How are producer accuracy and user accuracy different?
Answer 19: Producer accuracy is a measure of how often real features on the ground are correctly shown on the classified map, or the probability that a certain land cover of an area on the ground is classified as such. User accuracy is the probability that a pixel labeled as a certain land-cover class in the map is really this class, so this is a measure of the analyst's accuracy in correctly defining training data.

Question 20: If I want to do a vegetation classification, but I only have the GPS location of 3 species, which are the ones I want to identify in all the images, is this method the best for doing that?
And the second question is, if once trained with my data, can I use the same classifier with images of other years or locations?
Answer 20: If you are interested in identifying three species of vegetation, you may run into issues. Unless these three species exist in larger patches, the 30m Landsat pixel resolution may not be enough to identify species spatially. Species differentiation is also challenging with vegetation. Hyperspectral imagery can help with this. You'll also want to make sure you have many training points. Having more training points helps build greater accuracy. It also helps to have reference data for other land types as well, so the classifier can differentiate your target classes from other classes on the ground. The method presented in this training might not be best for your needs. With reference to species differentiation, you might want to take a look at our hyperspectral imagery training:
https://appliedsciences.nasa.gov/join-mission/training/english/arset-hyperspectral-data-land-and-coastal-systems

Question 21: Is there a way to program a k-fold training validation or other types?
Answer 21: This is beyond the scope of this training, and I'm not sure about k-fold training validation. But you can take a look at the functions available in GEE. For example, here is reference to the producer's accuracy function:

https://developers.google.com/earth-engine/apidocs/ee-confusionmatrix-producersaccuracy

Question 22: How do I get the shapefile and images in my GEE? Is there a particular projection needed?
Answer 22: Under the “Assets” tab, you can select the red “New” button and select what data type you want to upload from there. If what you are looking for is in the data catalog, you can draw from that directly. In this exercise, we are calling in the public asset (shapefile) and imagery (from the GEE data catolog) with code in the code editor.

Question 23: I have a different validation overall accuracy and validation kappa than what Britnay got, why is this? Should we run it multiple times and take the mean?
Answer 23: With cloud computing, there is a little bit of variation based on factors such as time running the function and the different selection of random points for testing between runs of the function. This is normal. You could ensure that your accuracy assessment uses the same points at all times, or you could take an average like you mentioned. The variation usually doesn’t create drastic changes in accuracy percentages.

Question 24: Does the impervious surface layer from USGS/NLCD cover all parts of the world? How do you change the code to classify in a country such as Colombia?
Answer 24: The USGS NLCD covers all 50 states and Puerto Rico. We showcased NLCD to show you how you can include additional datasets available in GEE within your maps. To classify urban cover types anywhere, collect reference training data for urban areas and include them in your training data with the other land cover type training points. This will train the classifier to also classify urban areas.

Question 25: I have uploaded shapefiles in my asset for classification, but I am getting errors while adding the property. So can you tell me how to add a property to feature collection?
Answer 25: Sometimes it can take a bit for assets to be reflected on your account, so you might need to wait a bit after upload. You might want to email us for specific clarification on this, but a good place to start might be the GEE developer page on feature collections:
https://developers.google.com/earth-engine/guides/feature_collections

Question 26: Is there any option within GEE to evaluate the sample set, to have an idea of what samples should be kept and which ones should be discarded to improve the classification results?
Answer 26: Refer to Q18

Question 27: What would be the best method to include very high resolution imagery (e.g. UAV data) into this Landsat based classification process?
Answer 27: You could use the UAV imagery to create training or testing points to train a classifier using Landsat imagery or test the accuracy of a Landsat classification. Basically, since UAV data has such high resolution, you might be able to visually determine land class from these images, creating a wide array of training and testing points for a Landsat classification.

Question 28: Why is there overlap in the variables cloudShadowBitMask and cloudsBitMask? (1 << 3, 1 << 5)
Answer 28: The values 3 and 5 refer to the assigned pixel_qa value of each pixel. “Pixel_qa” stands for Pixel Quality Assessment, where pixels go through Landsat’s CFMask Algorithm where each pixel is assigned a value based on their contents. Values 3 and 5 are for pixels identified as cloud and cloud shadow. By masking them out, we get cloud and cloud shadow free imagery. If you want to learn about the pixel_qa values and the CFMask Algorithm, they’re explained in the Landsat 8 Collection 1 Land Surface Reflectance Code Product Guide here: https://www.usgs.gov/media/files/landsat-8-collection-1-land-surface-reflectance-code-product-guide

Question 29: How do you handle shadows in high resolution imagery like NAIP? Is it possible to complete a land cover classification without masking out clouds given that no clouds are used in the training data?
Answer 29: Just because you don't include something as a classification doesn't mean it wouldn’t be classified anyways. Clouds would likely be classified as one of your other classifications. In our example, we used the QA band in the Landsat 8 surface reflectance imagery dataset to exclude all pixels labeled as cloudy. With NAIP imagery, you’d want to mask the clouds by removing cloudy pixels or by classifying clouds and removing them later.

Question 30: Not clear why divided by 10000 in the mask...Could you explain it?
Answer 30: For Landsat 8 Surface Reflectance data, the reflectance fraction is scaled by 10,000 so that the data can be distributed as Int16 type to reduce file size. We

unscale these data by dividing by 10000. You can learn more about it in the Landsat 8 Collection 1 Land Surface Reflectance Code Product Guide here: https://www.usgs.gov/media/files/landsat-8-collection-1-land-surface-reflectance-code-product-guide

Question 31: I see you use /* in the GEE script. // is used to comment out a chunk of the code. What is the purpose of using /* and */? What is the command for "Comment/Uncomment" on a linux machine?
Answer 31: The use of the “/*” and “*/” is just another way to comment out code. I’m not sure for Linux, but I would guess that maybe “Ctrl + /” would work (the same as for Windows).

Question 32: I find the gee docs functions description pretty bare bones and not that helpful. Is there another source that provides examples and a more comprehensive description of how to use the functions?
Answer 32: Yes! Please refer to the GEE guides link provided in the slides or here: https://developers.google.com/earth-engine/guides

Question 33: How to save this code in my GEE account?
Answer 33: To save code in your GEE account, click the save button at the top of the code editor, and then select a file path. It will save to your owner repository or you can specify a subfolder. You can view all of your scripts in the scripts tab.

Question 34: Is it possible to add more counties or satellite paths to cover more area in the land classification? Can you please show how to change the code for multiple counties please.
Answer 34: Yes this is possible! You would just need to use a different shapefile to filter your chosen image collection. The county we used in our example uses 2 landsat tiles for full coverage. So if you wanted to, for example, map land cover over the entire state of Maine, you would upload a new shapefile for the whole state as an asset, and then use that geometry in your .filterBounds() functions and .clip functions.

Question 35: Can we upload external .HDF files in GEE as assets and process such as subsetting, time series creation, etc.? How can I add an asset into my tab (e.g., an image collection)?
Answer 35: I believe image uploads to GEE must be in GeoTIFF or TFRecord format. So you cannot do this with HDF files. You might want to explore common methods of converting HDF data to GeoTIFF files. Once you do this, you can upload your chosen

files to GEE as assets, call them into your script, and manipulate your data similar to how we did in our activity.

Question 36: What is the advantage of importing variables as opposed as declaring them within the JS code? When is it better to import assets?
Answer 36: There is not necessarily an advantage one way or the other. In GEE, even if you use JavaScript code to import a variable (geometry, feature, table, etc.), the code editor will give you the option to include it in the import section of the code editor, thus taking it out of the code. This allows the imports to load as soon as you open your code.

Question 37: Why was it necessary to include summer date ranges for the years 2019 and 2020? Asking this in the context of land cover changes that may have taken place during that span of time. Will this not hamper the classification?
Answer 37: We composited this way to obtain a full image with no clouds or cloud shadow. Unless an abrupt event prompts change in land cover, cover types don't usually change drastically from year to year. We filtered for summer months because this is when vegetation in temperate regions, like Maine, is most well represented (due to green up). You can filter however you see fit for your own ecoregion or time series needs.

Question 38: I cannot see any assets in my Assets tab. Do we need to import any assets to follow the training? Do we have access to these assets so that we can also run this code?
Answer 38: Brittany hosts the assets necessary for this training on her account. She made them public, so we are able to use them freely by calling them in with code and/or including them in the imports section of the code editor. You do not need to upload these assets individually.

Question 39: Do we need to add individual bands or stacked images in GEE?
Answer 39: Once you import a dataset, like Landsat 8 Surface Reflectance, all band information and the entire image collection is called into the code. We can filter bands and images from this point depending on our specific needs.

Question 40: var CCmaine says "Table" but she said it is a shapefile. Does "Table" always mean shapefile?

Answer 40: Table doesn't always mean shapefile. GEE stores external data, like shapefiles and CSV files, as tables. A table could also refer to another geometry type or user uploaded data.

Question 41: Is this function for cloud masking similar to the code script for cloud masking Sentinel imagery? Does it work for Sentinel-2 imagery as well?
Answer 41: This type of cloud masking will only work for surface reflectance data that includes quality assurance bands. Sentinel-2 surface reflectance data available on GEE includes similar information, so you would be able to use a similar technique. However, the bands and bits for masking vary from this Landsat method. Take a look at the bands in this Sentinel-2 surface reflectance product to see how you need to adjust bitmasking for a Sentinel dataset:
https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2_SR

Question 42: The bitwise operator means that you transform values from 3 to 1 in this case: 1 << 3? Why is there overlap in the variables cloudShadowBitMask and cloudsBitMask? (1 << 3, 1 << 5)
Answer 42: Refer to Q28

Question 43: When dividing the samples into training and testing sets, is there a way to set the split ratio by category, in order to assure that all categories are included in both sets in unbalanced datasets? Are each one of the training points just one pixel or more, per training point? How do you increase the number of training points?
Answer 43: You could split training data before merging all of them into a single dataset. That would essentially allow you to take a certain percentage from each class, and then merge this testing data into a single dataset. In our case, each training point represents a single pixel. To increase training points, you'll need to obtain more reference data, either from the field or by looking at very high resolution imagery to determine known cover type data.

Question 44: As far as I know, the `print` and `getInfo` functions move the computation from server side to client side. Is there a way to inspect GEE objects and display their size, for instance, keeping the computation on the server side?
Answer 44: I believe all functions, including those you mentioned, are processed on the server side. GEE is completely cloud based, so everything you run in the code editor is processed via Google servers.

Question 45: Can you please explain the part in the beginning of the GEE code? How do you import to GEE your points for training and validation from your own machine as this is not clear to me? Can you please show what this point file looks like? Is it a .shp or .csv? What are the attributes? Could you explain what is the origin/data source of the training and validation points used in the RandomForest model?
Answer 45: In our case, we uploaded training and testing data as assets. We determined reference points using high resolution imagery, noted cover type and lat/long in an Excel table, saved as a CSV, and then uploaded CSV files as assets in GEE to call into our script. In this case the attributes are just cover type and lat/long.

Question 46: Other than Kappa statistics, what other statistics could you use?
Answer 46: GEE has a variety of functions you might find useful, one that stands out for accuracy assessment is the producer's accuracy function: https://developers.google.com/earth-engine/apidocs/ee-confusionmatrix-producersaccuracy