Dec 3

**Desk**

In GEE file Crop\_Vis\_v2, added ability to mask out CAR polygons based on percent area of polygon that’s planted (filter based on the year with the lowest planted area) and filter for minimum total area (e.g. take out tiny slivers of land). This may make the resulting estimates less likely to deviate very far from the center.

In R file Crop Timing Vis.R, added red lines to the ggridges plot representing the farm of interest.

**Meeting with Sally**

Look at phenology cameras at field level for validation.

It’s okay to use Landsat for land use only and ignore Landsat-MODIS fusion; work on writing first chapter next summer and finish all work by Dec of next year.

Dec 4

**Desk**

In GEE file Crop\_Vis\_v2, added option to filter out CAR polygons based on the year with the lowest planted area (i.e. filters out all years of that polygon) – need to uncomment something in neighboringPolys\_50km\_withArea, but it takes a long time to run; the plots run out of computation space; and the resulting histogram looks like they have chunks taken out in the middle. Instead, filter out only polygon-years with a low percentage planted.

Start putting together example visualizations at the polygon level using Crop Visualization Google doc.

Dec 6

**Desk**

Adjust Crop Timing Vis.R’s ggridges plot to include onset and put plant and harvest on the same line

Update Google Drive doc Crop Visualization according to Avery’s comments

Begin R Notebook, Crop Timing Vis Notebook.Rmd for final report on crop timing; added images for plots/maps from GEE, zipped all the necessary files (images and csvs from GEE) into the same folder called Crop Vis Notebook.

Dec 7

**Desk**

Planet Labs: looked at possibility of using it for validating planting, harvest date. Uploaded shp file of chosen CAR polygons and clicked through the images available at the polygon level.

Dec 8

**Desk**

Requested download of a sample image from Planet labs of a polygon in MT

Took screenshots of chosen polygons in MT and Matopiba over time in Planet labs to do initial validation in case can’t see details after free trial ends

Dec 9

**Desk**

Uploaded an example planet labs image from Rapid Eye to GEE asset planet\_labs\_test\_image

Dec 10 – 14

**AGU**

Mapbiomas: pastagem.org has pasture as polygons

Mapbiomas: people at AGU that I talked to were Claudinei Oliveira-Santos ([claudinei.santos@ipam.org.br](mailto:claudinei.santos@ipam.org.br)), Laerte Ferreira (gave the talk on mapbiomas), Leandro Parente (talked to him at his poster)

SAR: course by Franz J Meyer ([fjmeyer@alaksa.edu](mailto:fjmeyer@alaksa.edu), met him at his poster), class web site is <https://radar.community.uaf.edu/> and register at <https://ecampus.usf.edu/register/> and <http://learnsar.open.uaf.edu/> and they also have a SAR data processing that’s better than GEE at hyp3.asf.alaska.edu

Career: NASA, do development work with RS, NASA Applied Sciences Program

Career: IFPRI, added a girl from WeChat who works there and is interested in spatial analysis, international development, saw Claudia Ringler as part of Alternative Career panel

Career: telluslabs, talked to someone there who works for TellusLabs and he showed me the web app; they are hiring and are based in Boston, and work on forecasting agri productivity around the world using RS.

SAR: go to SERVIRglobal.net for SAR information. Look at SAR handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation. In it, there’s a paper by it’s by Franz Meyer called ‘Spaceborne Synthetic Aperture Radar – Principles, Data Access, and Basic Processing Techniques”.

Career: Applied Geosolutions

Talks: Vivianna Zales

Talks: by Beth Ziniti, I went to her talk on Dec 12 about classifying crops in US with random forest. She mentioned XAR which is a software to process SAR data. Email her to get in touch with the guy who does the software ([bziniti@appliedgeosolutions.com](mailto:bziniti@appliedgeosolutions.com))

GEE Workshop

* Classifiers
* Can classify by anything other than RGB bands as well (precip, elevation, etc)
* Constraints: integer classes, smaller numbers to keep confusion matrix small
* Examples -> Cloud Masking -> Sentinel 2 (to get cloud masking for Sentinel)
* 'CLOUDY\_PIXEL\_PERCENTAGE' means the total percent of pixels in image that are cloudy, not how cloudy individual pixels are
* var cirrusBitMask = 1 << 11; means get the bit 11
* To begin supervised classification, add a geometry, turn into Feature Collection, add property called class and set value to 0, 1, etc
* Control + space = autocomplete the arguments of a function
* Use sampleRegions() to extract the training data from RS images
* Then classify: common ones are CART, random forest, svm
* classifier.confusionMatrix().accuracy() : don’t use because it simply tells you how homogeneous your training data is. Instead, take out a random section of the training data and make it a validation dataset with the following:
* var random = training.randomColumn() // adds uniform, random distribution of numbers between 0 and 1
* var validation = random.filter(ee.Filter.lt('random', 0.3))
* training = random.filter(ee.Filter.gte('random', 0.3))
* Then,
* var classifier = ee.Classifier.randomForest(20).train(training, 'class', composite.bandNames())
* print(validation.classify(classifier).errorMatrix('class', 'classification')).accuracy()
* This method also gives deceivingly high accuracy because the training vs validation data was not independent of each other; polygons are homogenous to begin with
* If hit out of memory issue, it’s usually a tile issue. GEE uses 1000 x 1000 pixel tiles by default. To get around, set tileScale = 2 to 16. 16 will take longer (adds time for more space). Time max is 5 mins
* Classification script:
* goo.gl/tacWyQ
* Crop classification
* Look under my Tutorial Repo for GEE file GEE\_Classification
* By default, when sampling they don’t include lat and long because it takes memory; use geometries: true if want to put sampled points on map
* A tile error is a problem with the map
* Computations get cached, so if run code over and over again without changing anything, computation time should go faster
* To improve classification:
* Change number of trees (adds computation time and benefit flattens out)
* Add new information to classify with (best. Bring in phenology)
* More training points (but may add too much computation load)
* Fusing Sentinel 1 and Sentinel 2 for classification
* Sentinel 1: pick either ascending or descending because shadows change. Different areas have more ascending than descending data, etc. Descending is more common in brazil
* Sum, mean, etc reducers use weighted pixels (i.e. at boundary of polygons), can unweighted() to ignore
* Github repo about Python called GEE-Community
* Code Module: require() - code reuse!
* Global Surface Water: GEE app
* Collect Earth: data collection tool
* Next year: interface with SDGs/development/UN/scientists, data collection and integration (validation platform for anyone to use), User Summit (september, october)
* Earth Engine Announcements: email list to subscribe to

AGU oral session notes:

* Vivianna Zales: talk on Landsat crop mapping. They did manual training and made bagged classification tree model. They followed the good practices recommended by Olofsson, 2014. They have a paper coming out in PNAS, contact her at [vzalles@umd.edu](mailto:vzalles@umd.edu)
* Timeseries analysis: use Prophet package in R
* BFAST
* Meha Jain: talk on satellite data to calculate yield. Used Planet and SCYM.
* Idea: go to Planet Labs, pick up visual planting date for 50km, look at timeseries estimated planting date and see if time and space variability remains at CAR polygon level after aggregating pixels
* OpenFORIS SEPAL by FAO
* Collect Earth
* Mapbiomas: based on Landat, has 1985 to 2017 data (Collection 3)
* Sinop in Mato Grosso is a highly cropped municipality
* GeoGlam: projects crop production with remote sensing (early warning)
* NDVI exaggerates crop greenness towards harvest (compared to SIF), but closely follows SIF at start of season
* Planet labs images need additional calibration because CubeSat is cheap but not well corrected
* SAR for crop classification with machine learning: Applied Geosolutions does this
* Time mask to make sure each sample has enough points; also, to pick only homogenous pixels, use a 5x5 window and only use that pixel if surrounding pixels have the same crop
* XAR by Xiaodong Huang is a software to process SAR. Email Beth Ziniti for connection

Dec 16

**Desk**

Looked at screenshots of Planet imagery over Mato Grosso that were saved in Drive’s Crop Visualization folder, compared to my estimates of first crop planting/harvest and Jake’s soymap, found that 500m is not enough to get rid of natural vegetation; many years/fields didn’t have enough temporal resolution in Planet to get Planet-derived planting/harvest date, it’s easier to get ground harvest date than planting date because there’s a lot of missing data around planting, and Jake’s soymap tends to overestimate the amount of single cropped pixels. A lot of pixels that were classified as single cropped are actually double cropped (as it appears in Planet). Also very much need a way to get rid of center pivot pixels because they have a completely different crop cycle.

Planet validation in MT indicate a need to think about whether polygons are appropriate for aggregation, and if so, how to mask out pixels and whether to divide a polygon into separate pieces.

Multiple crop cycles might be represented in same polygon: center pivot; first crop harvesting/late stage when second crop is already greening up in other parts of the polygon; perhaps certain areas only have the ‘second crop’ and skip soy which will create a planting date estimate > 150.

Takeaway: Where Planet data is available, both estimated planting and harvest dates at the pixel scale match up with Planet images. There is a problem in certain places of (maybe) mistaking a second crop for the first crop; the pixel level estimates of planting capture the spatial pattern in planting. Re-do soymap or combine single and double cropped pixels. For plant date estimates that are around 150, check the timeseries.

Dec 17

**Avery Skpye**

* New climate and crop timing visualizations: download soy sourcing information by municipality for a company called Cofco from a web site called trase.earth. Need to turn the information (by municipality) into a shapefile and map the financial flow, trade volume at municipality level; see if $/ton is same across munis too (if not, also map $/ton)
* Make KDD, GDD, crop timing histogram plots by DOY in R; as Saleem for R code that does this; make these timeseries plots for MT, aggregated Cofco municipalities, Matopiba, Cerrado, my chosen polygons –wait for Dave to extract KDD and GDD at these points
* Re-run Jake’s land use map with mapbiomas3 instead of 2.3; Avery has it in his GEE script Soy Classification\_mb3
* Facet\_wrap: R package to look at

Dec 18

**Desk**

Under new GEE repo ‘AveryTasks’, started DOY\_timeseries\_KDD\_GDD to get KDD and GDD timeseries over specific regions, given images for KDD and GDD; and also started Cofco\_Municipalities to join municipality level soy sourcing data for the company Cofco.

In R code ‘Avery Work/Cofco\_Cleaning.R’, aggregated the raw data downloaded from trase.earth to muni-years, which is then input to GEE to create a spatial dataset.

Dec 19 - 21

**Desk**

* Looked at Matopiba CAR poly imagery from Planet Labs and did validation (compared survey reported dates, timeseries estimated dates, and Planet data). Purpose was to get initial idea of validation, areas to validate, issues to look for and avoid. Validation and images can be found in Google Drive folder called Crop Visualization.
* Updated java version to JDK 1.7 / 7 using the following link so I can run Alex’s code that extracts tables from pdfs. https://stackoverflow.com/questions/52524112/how-do-i-install-java-on-mac-osx-allowing-version-switching
* Avery sent me R notes for plotting spatial data (look at Section 12 and 13): <http://edrub.in/ARE212/notes.html> but this web site is also useful for regressions and causal inference stuff later!
* <https://kieranhealy.org/blog/archives/2018/12/09/canada-map/> is also useful for mapping in R
* <http://mazamascience.com/WorkingWithData/?p=1494> is also useful for mapping in R
* Mapping of Cofco munis is in R file Cofco\_mapping.R
* Worked on Alex’s code to extract reported crop progress data in R file r\_code\_agroserv\_cropdates\_v3.R

Dec 27

**Desk**

Worked on Avery’s task of overlaying DOY plots of KDD, GDD, and planting/harvest histograms. Extracted KDD, GDD from GEE file DOY\_timeseries\_KDD\_GDD and move into R file GDD\_KDD\_Timeseries.R to produce plots

Dec 28

**Desk**

Worked on Avery’s DOY plot task in R file GDD\_KDD\_Timeseries.R, added in histograms of planting/harvest date that were calculated in GEE and calculated DOY since Jan 1 of actual year to match with GDD and KDD dates, and plotted on top of KDD and GDD. Still need to get actual KDD, GDD and crop dates information; crop dates in particular may need to be exported as raw information (before reducing to histogram).

Looked up Jupyter Notebook, but have trouble integrating package versions – read that it’s better to use R Notebook within RStudio instead (also because it’s easier to track in GitHub because Jupyter produces .ipynb and Java stuff), so go with that in the future.