Diameter Estimation of Eucalyptus spp. Plantations in Southern Brazil Using Global Ecosystem Dynamics Investigation Data and Support Vector Regression

Benjamin D. Miller

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Forestry

Randolph H. Wynne, Co-chair

Valerie Thomas, Co-chair

Stella Schons

May 10, 2022

Blacksburg, Virginia

Keywords: Remote Sensing, Eucalyptus, lidar, Machine Learning, forest plantations

Diameter Estimation of Eucalyptus spp. Plantations in Southern Brazil Using Global Ecosystem Dynamics Investigation Data and Support Vector Regression

Benjamin D. Miller

(ABSTRACT)

Forest plantations make up a large percentage of managed forest land globally. Assessing plantation productivity is vital from both commodity production and carbon management standpoints. Measuring the productivity of these areas is essential given their rapid growth and turnover. Transparent metrics to compare reported carbon storage with estimated values are required for internationally transferred mitigation outcomes under Article 6.2 of the Paris Agreement. Data from the Global Ecosystems Dynamics Investigation (GEDI) provide an excellent opportunity to measure plantation forests over large areas. We focused our efforts on Eucalyptus in southern Brazil and used data from an industrial partner to investigate plantation metrics (height, diameter, volume, stems per hectare, etc.) and to create a model of plantation diameter using Support Vector Regression (SVR). SVR enabled a robust model of tree diameter even given the heteroskedasticity and spatial autocorrelation present in the GEDI data, which deleteriously impacted attempts at linear modeling. We could predict tree diameter in these plantations to within 1 cm using space-borne lidar, with broad implications for using space-borne lidars to monitor carbon accretion in secondary forest plantation.

Diameter Estimation of Eucalyptus spp. Plantations in Southern Brazil Using Global Ecosystem Dynamics Investigation Data and Support Vector Regression

Benjamin D. Miller

(GENERAL AUDIENCE ABSTRACT)

Forest management practices have shifted in some cases to very crop-like forest plantings. These areas are functionally different from a 'natural' forest. Understanding the structure of these areas in a rapid and consistent manner is important to quantify the amount of carbon stored within these forests for international climate agreements such as the Paris Agreement. This effort focuses upon Eucalyptus forests in Southern Brazil. Using measurements from a lidar instrument (a lidar system fires a laser beam from space to the ground, recording the 'deflection' of the laser beam and the amount of time it takes to return to the sensor to measure features on the ground) we were able to measure the diameter of the trees to within a centimeter in these forests.

Dedication

To my parents, for instilling a deep love of the world in me along with a pinch of skepticism...

just in case...

Acknowledgments

I would first like to thank my advisors, Randy & Val. They have been a constant source of encouragement and support throughout this process and have put up with the multiple questions I pursued prior to arriving at this work (No matter how zany it seemed). I'd also like to thank them for the amount of patience they showed me while I was working out the details of managing a 'Big-Data' project for the first time. They have been a great support and are gold-standard mentors. Thank you.

I'd like to thank Les Fuller as well. I had a lot of questions regarding how to go about computing in the environment he manages and he was a great resource for the entirety of this work.

My dad has a story in which there are two sawyers working in a forest together. They begin the day walking into the woods in the morning before heading their respective way for the day's labor. The sawyers are working close enough to one another that they can hear each others saw cutting away through the forest. One of the sawyer hears the other stop periodically throughout the day, and is confused. At the end of the day, the sawyers are comparing how many trees they each cut down and it becomes clear that the one who had stopped cut down more trees. The lumberjack who cut all day asked the other how it was possible that they spent more time sawing than the other but cut down fewer trees. The other responded with 'well, I stopped to sharpen my saw'. Thank you to the many friends who have been there for guidance, adventures, and camaraderie these past few years. It has been a pleasure to sharpen my saw with you all during this time.

Thank you to my partner Sophie who has been a never-ending well of love and support.

Contents

List of Figures

List of Tables

Chapter 1

Estimating Tree Diameter from Waveform lidar

1.1 Introduction

Figure 1.1: An intensively managed Eucalyptus sp. forest in Uruguay. (Photo from Mashian 2018)

1.1.1 Background

Around 7% of the total forest area in the world is planted forest (Payn et al. 2015, Winjum and Schroeder 1996, FAO 2020). Accounting for the carbon cycling occurring in forest plantations is important to help improve the quantification of forest carbon stores. The frequent harvesting associated with forest plantations can affect the stability of soil carbon stores and reduce the amount of carbon stored in tree stems over time (Clark, Gholz, and Castro 2004; Cook, Binkley, and Jose Luiz Stape 2016).

Forest plantations are homogeneous in age and structure reducing the habitat and ecosystem services

that they provide (Bonan 2008). See Figure 1.1 for an example of stand structure. It is worth noting that while these areas provide fewer ecological services, they do permit intensive management to occur within a smaller land area than other forest management practices (Norfolk and Erdle 2005, Pirard, Dal Secco, and Warman 2016) and can help regenerate abandoned agricultural land (Campoe, José Luiz Stape, and Mendes 2010). Less intensive management methods often create more regular disturbance over a larger land area which can increase the overall impact of forest product harvesting (Buongiorno and Zhu 2014). Forest plantations focus the impact that forest management has on the landscape to a smaller fraction of land area. Additionally, these forest plantations create a better carbon sink than other land uses of converted primary forest such as agriculture (Hua et al. 2022), even though there is documented loss of soil carbon after repeated rotations (Cook, Binkley, and Jose Luiz Stape 2016), concern of effects on the water table, and restoration to native forests would be the better carbon and ecosystem decision (Hua et al. 2022).

Brazil is an area of rapid forest loss and conversion (FAO 2015, FAO 2020, Myers et al. 2000). The Atlantic forest region is under particular threat as 7.5 % of the native forest remains as of 2000 (Myers et al. 2000). However, the forested landscape is not entirely harvested without replanting trees, and eucalyptus plantations can create some but not all the ecological benefits of a secondary forest (Campoe, José Luiz Stape, and Mendes 2010). Eucalyptus plantations in Brazil are commonplace, wherein the same land area is used repeatedly for timber production and the amount of land area used as a planted forest is increasing (FAO 2015). These areas are primarily used for paper pulp and the short rotation age ($<$ 10 years (Cosenza et al. 2017)) creates rapid stand turnover.

Production amounts and holdings of these timber stores are often consolidated private entities and thus it is in the interest of the holders to maintain the privacy of their land holding data. There is a vested interest in privacy of these plantations as the volume of timber grown on a portion of land is a propriety advantage when choosing to harvest. This incentive for privacy creates difficulties in created landscape level assessments of forest volume. There are already ongoing efforts to map the

land area that eucalyptus plantations occupy (Harris, Goldman, and Gibbes 2021, Petersen et al. 2016) to better understand these areas prevalence on the landscape. Estimating the volume of wood produced in these areas can provide key information to estimate production in a given year. The information regarding production can be driven by the previously mentioned industrial interest, but also to provide information to governments and environmental organizations to better understand the effects and flux of these forests.

There are a variety of different approaches to estimating volume with existing models. Height and diameter are the primary drivers of these models with diameter being of particular importance (Burkhart and Tomé 2012). Volume as an estimated parameter from a direct physical measurement is difficult as it is a modeled parameter that is directly correlated to diameter and height. Specifically, estimation of eucalyptus form and volume is still an ongoing effort and still results in error of estimation (Boczniewicz, Mason, and Morgenroth 2022). As volume on its own can be estimated in a variety of ways it is prudent to try and use a estimation technique that can result in an easily verifiable field measurement such as estimating diameter. In a eucalyptus stand, height is closely related to diameter, so it becomes feasible to back-predict diameter directly, quantify the error of that estimate and then incorporate that error into a volume estimation. This would not be possible in a mixed-age forest as the canopy diversity would make initial estimates of volume directly from height only the better choice.

There is a long history of using lidar to measure forest parameters such as height (Fagan et al. 2018, and a great review paper Coops et al. 2021). Waveform lidar and forest parameters have previously enabled the estimation of forest heights, biomass, and basal area (Lefsky et al. 1999). Spaceborne waveform lidar in particular has been previously used to estimate forest parameters (Bye et al. 2017, Chen 2010, Neuenschwander and Pitts 2018) and provide a unique opportunity of measuring landscape level attributes as compared to more site specific work at the airborne lidar scale (Fagan et al. 2018, Coops et al. 2021).

To ask a landscape level questions, the implementation of an airborne waveform lidar system becomes logistically challenging and expensive. As such, spaceborne lidar becomes a good method to ask these questions. Previous spaceborne waveform lidars have not been designed around vegetative measurements, another study (Neuenschwander and Pitts 2018) use a lidar that was designed to measure ice and thus has limitations measuring forests. GEDI (Dubayah et al. 2020) has its measurements take place in the near infrared portion of the electromagnetic spectrum where vegetation has a high level of reflectance compared to other land covers. GEDI waveform lidar has previously been used to estimate canopy height and wood volume specifically in Eucalyptus plantations in Brazil (Fayad et al. 2021). These previous studies did not account for spatial auto-correlation.

This work creates an opportunity to use spaceborne lidar data to provide estimations of timber harvest for verification under Article 6.2 of the Paris Agreement ("Paris Agreement on Climate Change" 2015). "Paris Agreement on Climate Change" 2015 calls for independent verification of carbon stores (such as forests, and planted forests) of a country's claimed volumes. This verification requires accurate estimates of forest volumes, and the method outlined in this work creates an opportunity to provide that verification in addition to more information to landholders. There is also some utility in assessing silvo-pastoral lands which may not be directly quantified as a forest plantation in other contexts. Spaceborne lidar systems have been previously used to estimate the canopy height of plantations and volume (Potapov et al. 2021). These previous studies help improve the quantification of these areas on the landscape and better understand the volumes at a given time. What remains is the direct estimation of diameter from spaceborne lidar which is the focus of this work. The methods we use can help policy makers understand and quantify the volume of wood present on the landscape while weighing the ecological and economic trade-offs outlined in Hua et al. 2022.

1.1.2 Objectives

1. Create a robust model of diameter estimation using lidar data from the Global Ecosystem Dynamics Investigation (GEDI) dataset.

2. Using already produced boundaries (Petersen et al. 2016) of Eucalyptus plantation, produce a map of these areas diameter.

1.2 Methods

1.2.1 Study Area and Stand Data

The study area was defined by the approximate geographic bounds of data from our industrial partner. The boundaries of the data used can be found in Figure 1.2. The area is located in the São Paulo State of Brazil as a part of the Atlantic Forest biome. The specific geographic bounds of 20.2°S latitude,48.8°W longitude, 23.2°S latitude, and 46.1°W longitude were used (Figure 1.2). This area was selected as it is the geographic extent of our 'ground truth data'. Information regarding the ground truth data can be seen in Table Figure 1.4 and Table 1.1.

Stand Metric	Mean ($\bar{x}$)	Standard Deviation (σ)
Trees per Hectare	1177	212
Volume per Hectare (m^2/ha)	271	47
Stand Diameter (cm)	13.6	2.1
Stand Height (m)	20.63	4.14

Table 1.1: Summary Statistics of Ground Truth Data. All the data were pulled from the same geographic extent as the GEDI data used in this work.

Figure 1.2: Location of GEDI and stand inventory data (black box) used in this work.

1.2.2 GEDI Data Processing

The GEDI data (Dubayah et al. 2020) were downloaded from the Land Processes Distributed Active Archive Center (LP DAAC) operated by NASA. The level 2A (Version 2.0) product files were downloaded as well as the level 1B files (Version 2.0) to examine waveform characteristics of delineated stand features. Upon exploration, all data in these analyses used Algorithm 1 to convert data from waveform to relative height metrics. The data were then extracted from an archival file format using a tool called Gedi-Subsetter (Land Processes Distributed Active Archive Center (LP DAAC) 2021) from NASA that was modified to enable parallel processing. The total size of the files extracted from archival format was 1.44 terabytes. To reduce the amount of information that was not needed for this analysis, the GEDI data were then further trimmed down to a subset of the variables present in the dataset and then spatially joined to the data from our industrial partner (data from our partner were collected within a year of GEDI measurement thus reducing temporal lag), data from Hansen et al. 2013, and data from Petersen et al. 2016. The joined GEDI data were then filtered for quality using the built in quality flag in addition to being filtered for extreme height measurements by the GEDI instrument resulting in a final file size of 10 gigabytes. Correlations between the ground truth data and GEDI data were then explored.

1.2.3 Model Selection

We conducted some preliminary analyses that compared the stand data (Diameter, volume per hectare, trees per hectare, and stand height) using simple bi-variate fits. The obvious metric to measure using GEDI data is height, but there have already been studies investigating this metric (Potapov et al. 2021). We chose to leverage our detailed ground truth data to estimate a different forest parameter: diameter. The diameter of these areas are relatively homogeneous as the spacing methods and stand establishment timing are uniform and relatively homogeneous, respectively. This permits the trees to grow at a rate that are similar to that of their neighbors.

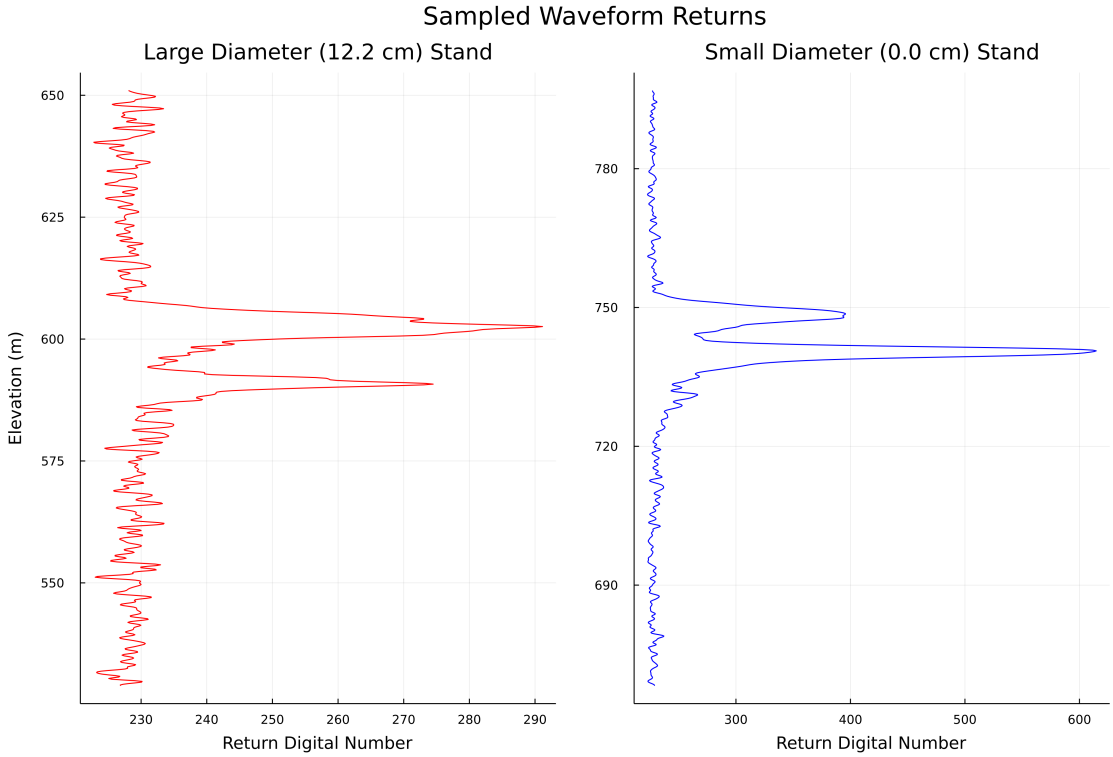

Figure 1.3: Sample GEDI Level 1B waveform returns from large and small diameter plantation stands. Stands sampled were intersected with ground truth data to ensure comparison accuracy.

1.2.4 Diameter Estimation

Modeling of the diameter of a plantation using GEDI data does present some problems as there is a spatial auto-correlation component present in the measurements ($p < .0001$). In addition, our preliminary analysis suggested there is heteroskedasticity present between the Relative Height 95 percent of waveform energy (RH 95) and the plantation diameter (Breusch-Pagan test of $13.1248, p < 0.0001$). As both of these criteria violate the assumptions of Ordinary Least Squares Regression, non-parametric modeling technique was employed. Support Vector Machine Regression (SVR) using a linear kernel was used to create a model to estimate plantation diameter (Vapnik 1998). The underlying goal of this method is to create a repeatable model that describes the diameter of the plantations using the GEDI data to ensure a more transparent relationship than other non-parametric modeling techniques.

The methods outlined in this work create a robust method of diameter estimation using machine learning to overcome the barriers presented above. There are distinct coefficients that are able to be extracted when using a linear SVR. These coefficients are then able to be used to compare the results of other modeling techniques (or similar models in different regions of the world) without having to directly share other trained 'black box' machine learning models while benefiting from machine learning algorithms. SVR was shown to be a more effective and accurate method of predicting stand level variables compared to other machine learning techniques and multiple linear regression (García-Gutiérrez et al. 2015).

Various relative height waveform return metrics were explored, but it was found that there was not a large difference between the heights at the return energy at 100%, 90%, or 80% in these stands so the 95% energy returns were what was selected for this analysis. A random sample of 80% of the measured heights (RH 95) from GEDI and the stand measured diameter were then fed into an SVR model. GEDI data that intersected with data from our partner numbered 21,438 quality

observations. Once the model was trained, the coefficients of the linear SVR model were extracted by asking the model to predict data on a regular interval (0-20). The model was then tested on a 20% of the data that was withheld from the training process (4,361 observations). The test data were then used to calculate a Root Mean Square Error (RMSE) of the model.

1.2.5 Mapping and diameter distribution

Following the creation of an SVR to predict stand diameter, we estimated the diameter of stands that were within the geographic bounds of our data but were not used as a part of model development (Petersen et al. 2016 were the stand boundaries used for this). The median estimated diameter of GEDI measurements within the bounds of the stand was used to determine the stand's diameter. The resulting predicted diameter were then mapped. The resulting predicted diameters were also binned into diameter groups of 0.2 cm and plotted.

1.2.6 Harvest Year Prediction

Using the estimated diameter and the measured height from GEDI we created estimates of growth trajectory for stands from our industrial partner. This process involved the determination of age, an estimate of height, height trajectory (rate of growth), diameter estimates from SVR, and diameter rate of growth. The year of disturbance from Hansen et al. 2013 to determine age was treated as year zero or stand reestablishment for the purposes of this exploration. The measured heights and the estimated diameters were extrapolated to a 'growth per year' estimate for each parameter. We also pulled our partner's estimate of harvest year out of the stand data to create at 'years until harvest' variable that would be our target estimate of stand diameter. We then used a random forest model (Breiman 2001) to try and predict the harvest year of a given stand using the variables described above.

1.3 Results

Comparing the stand data to its own variables as well as the relative height metric led to the conclusion that the diameter of the eucalypt stands had the strongest correlation to the GEDI data (Figure 1.4).

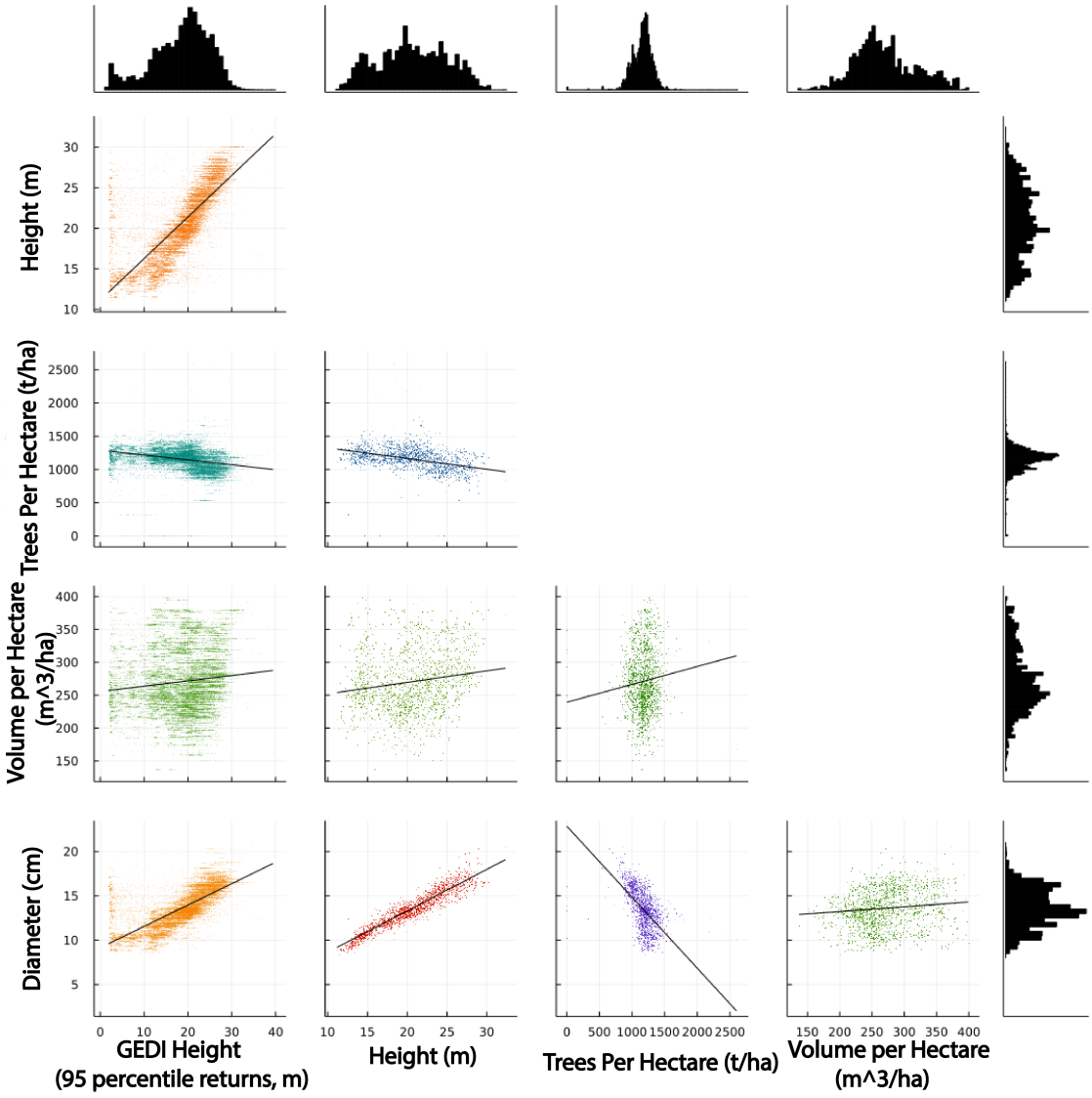

Figure 1.4: Correlations between level 2A GEDI RH 95 and various metrics of the ground truth data. $n = 21,438$. GEDI data were filtered for quality. Distributions of the various metrics and data can be found on the margins of the figure.

The created linear SVR model can be seen in Figure 1.5, the resulting slope equation of the hyperplane created by the Support Vector Machine can be seen in Equation 1.1. Figure 1.5 also demonstrates the fit of Equation 1.1 on both the training and test data.

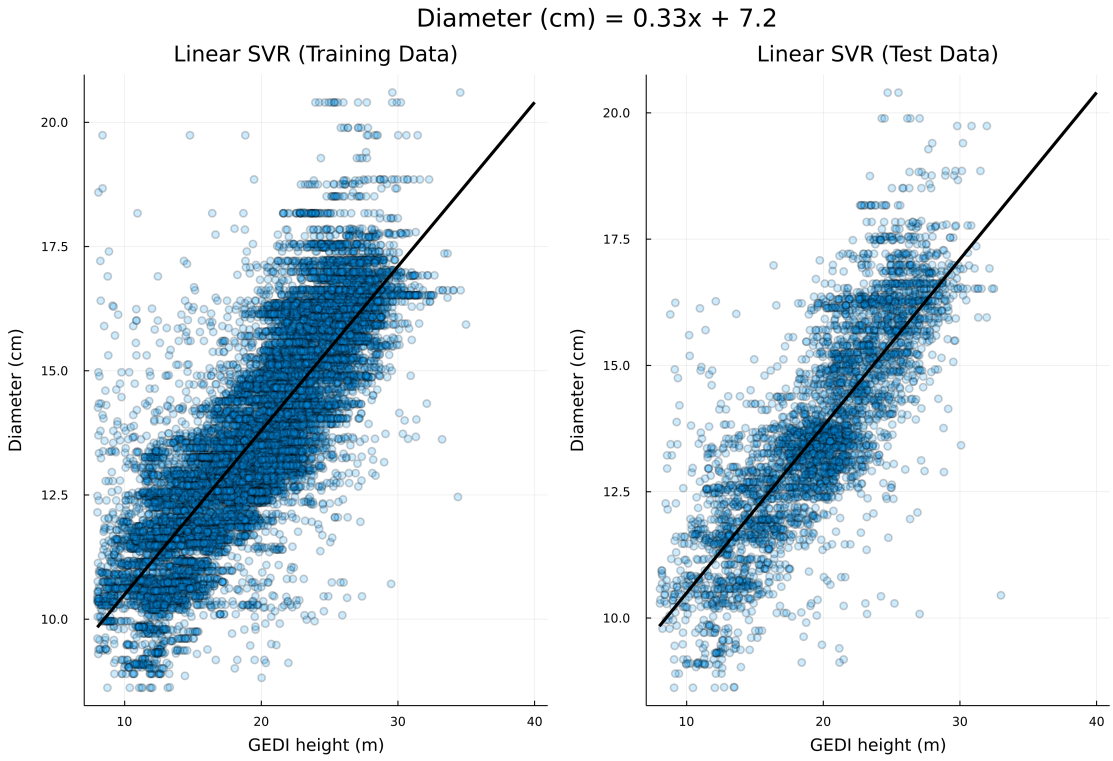

Figure 1.5: SVR performance performance on the training and testing data. Black line indicates the SVR model. ($n_{\text{train}} = 17,077, n_{\text{test}} = 4,361$). Training and test data were fully separated during model creation. Quality Level 2A GEDI RH 95 height data were used.

$$\text{Diameter (cm)} = 0.33(\text{rh}_{95}) + 7.18 \tag{1.1}$$

The resulting accuracy is assessed in Figure 1.6, which demonstrates the accuracy of the SVR at prediction diameter with an RMSE of 1cm.

The application of the SVR method to stands that did not contain 'ground truth' data can be seen in Figures 1.7 & 1.8.

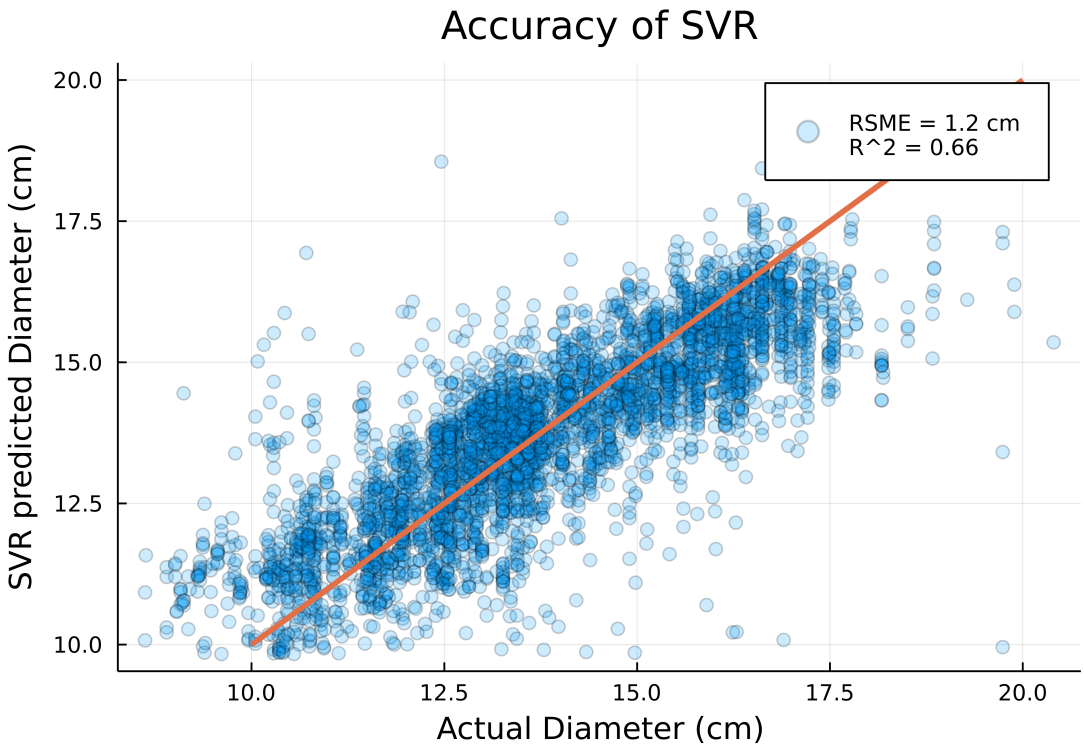

Figure 1.6: Predicted Versus Actual Diameter from the SVR model, with accuracy statistics. Line indicates a 1:1 line, ($n = 4,361$). RMSE was calculated using the Quality Level 2A GEDI RH 95 observations that were then predicted using the SVR model developed. These were then plotted with one following the calculation of the RMSE.

The random forest classification of harvest year was unsuccessful. The mean accuracy of the model after ten fold cross validation was 44%, so they were not pursued any further.

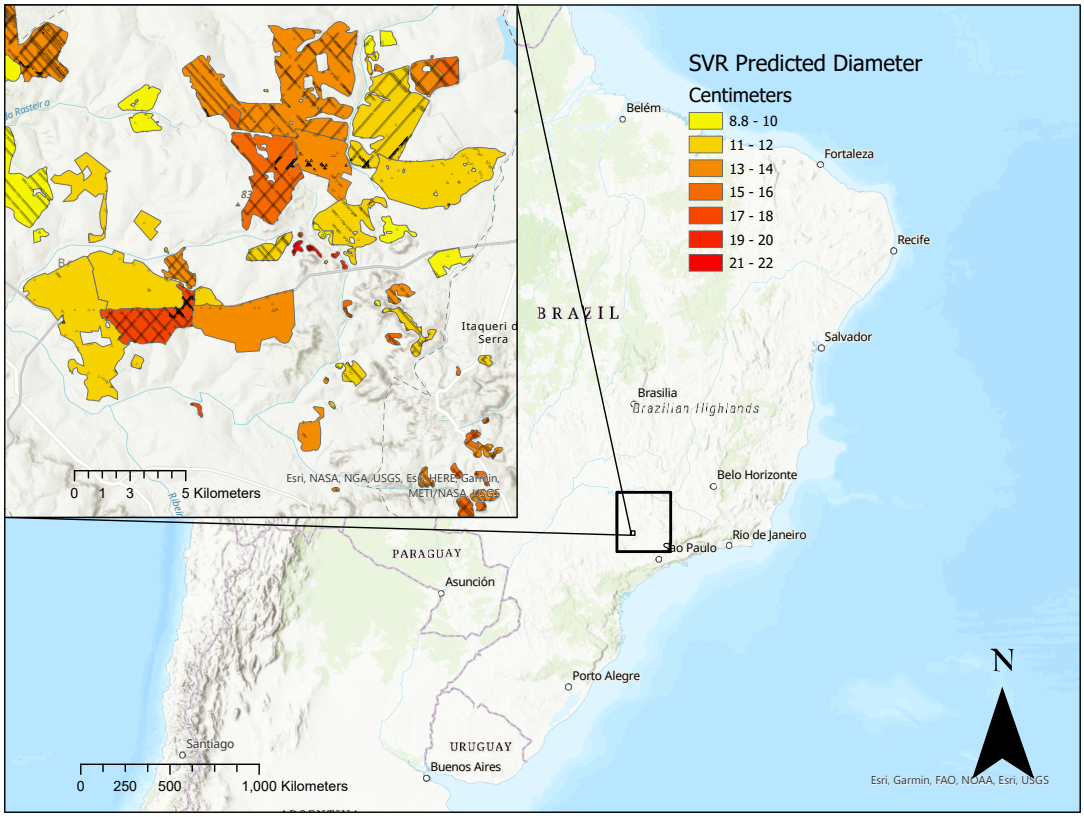

Figure 1.7: Map depicting applied SVR model to unmeasured plantations demonstrating the technique applied to the displayed GEDI measurements (displayed as small black 'x' marks). The quality Level 2A GEDI observations were plotted into the predefined stand boundaries from Petersen et al. 2016 and the median of the modeled diameters was selected to designate the colors of the plot.

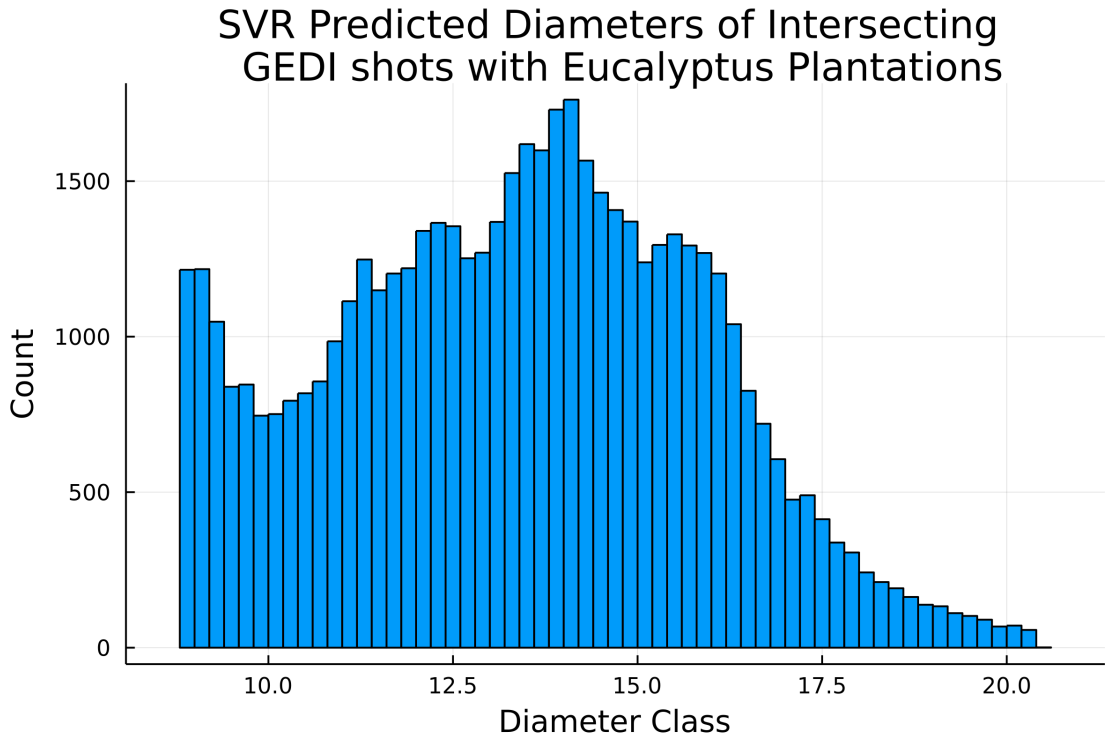

Figure 1.8: Distribution of intersecting GEDI shots with unclassified plantations. This is the overall distribution of the estimated diameters into small diameter classes. This will be useful to help understand the current holdings of Eucalyptus plantations in the study area.

1.4 Discussion

Using the SVR technique we are able to estimate the diameter of even aged stands of forest plantations using spaceborne lidar. The GEDI instrument proved to be an excellent source of forest structure data and performed admirably as the results of this work are accurate to within a centimeter using a waveform from space. This work also demonstrates that the sampling approach of the GEDI instruments' deployment is useful for estimating forest parameters and future instruments can focus on deploying a similar framework. It would be less crucial to have a higher spatial resolution and instead focus upon a higher temporal resolution to ensure that changes on the landscape can be more closely monitored.

Using this method creates an opportunity for a cost reduction in measuring these areas, both for industrial interests and entities such as government monitors. There is a potintial benefit to understanding how these areas are changing and having increased estimates of the total stocked area in the region could create an advantage to the land holders. Additionally, there is an opportunity to verify that the companies involved in the production of timber in these plantations are complying with their Environmental, Social and Governmental Standards (ESG) which could benefit the communities surrounding these forests as there is concern regarding the presence of these forests. If a company was interested in using this work it would be prudent to undergo a value of information analysis as there is a cost in computing time and effort to gain this information that may not actually be a net positive to a measuring campaign or a supply model.

This work had a number of challenges, the most pervasive of which was computing power. The data involved in this process were of a massive initial size (upwards of 15 Terabytes) and a full suite of data mining and computing techniques were needed to reduce data volume for analysis. Parallel processing was heavily utilized during the data mining processes as it rapidly increased the speed at which files could be processed and significantly reduced the overall processor time used.

Even using these techniques, the processor time used to get the data reduced to a use-able size for analysis was counted in weeks, not days or hours. While our hope for this work is for others to use our model and methods to monitor forests in this region of the world, the technological limitations should be discussed if attempting to apply the methods to a larger geographic region. This analysis was done without using a high powered computer (HPC) cluster, but it did require fast hardware, massive local storage, and computational awareness (i.e. ensuring that you are not attempting to load more data onto a singular processing thread than you have memory available). Additionally, when the data were reduced to their 'final analysis' format the computer memory needed to do any sort of data manipulation, transformation, model building, or graphing was non-negligible. Any larger geographic application of this technique would likely require a HPC cluster and still require weeks of processor time depending on the resources available.

Another challenge of this work was the GEDI data structure itself. When working on linking the waveform (level 1B) data to the summary data (level 2A) the naming convention between the unique shot identifier had changed; the change made it virtually impossible to link individual waveforms to their respective summary. A complete dissection of the underlying information in a unique shot ID would be required to link the 1B data to the summary data for a comparison of the summary 2A data. On our first attempt to link the data via the unique shot ID, there were no shots of either product that shared the same unique ID when a join was attempted by that modifier. Further investigation revealed that the aforementioned change in naming convention was not documented. A reading of the documentation did not lend itself to effective ways of how the names had been changed, or how to go about joining the two products. The only mention found to the naming convention change was in the documentation of a separate GEDI data product level. Future waveform lidar sensors should focus upon clarity and data point naming conventions between information levels to ensure data clarity and utility of offering multiple levels of product to ensure that there is a robust connection between the raw waveforms and created products.

The choice to only predict diameter with a model was deliberate. The model has a RSME of 1 centimeter using only the RH 95. Had we attempted to directly estimate forest stand volume, we would have had higher levels of error. Additionally, stand volume is a modelled parameter. There is a whole body of literature that is focused solely upon increasing the accuracy of forest volume estimates. Had we used our ground truth data to estimate volume, we would have been creating a model of volume that would itself be based upon a modeled estimate of volume. To ensure the widespread application of this technique it was determined that estimating an easily verifiable forest parameter outside of a height measurement would be more useful to end users. Diameter is also a crucial component of volume estimation (Burkhart and Tomé 2012) and by estimating diameter using the close relationship of height and diameter in these stands we are enabling end users to have access to a parameter that they can then use to estimate either timber volume or biomass using their preferred model.

This work will be helpful to those doing large scale regional assessments of timber volume or forest biomass of these even aged areas. The single time point of tree stem origin has allowed us to accurately create an estimate of forest diameter. This could be useful to a variety of end users. Those who are looking to understand the distribution of forest stand ages of other eucalyptus plantations around their own land holdings could use this technique to estimate the volume of timber and use heuristics to determine the effects on the value of their holdings. Projects seeking to understand the landscape level effects of these areas on land cover change can now better quantify the true impact of these areas on the available biomass or harvest-able timber. Carbon estimates can now be updated and compare the outcomes of different models as these areas will now have better available parameters for volume estimation. This work creates a clearly verifiable estimate of the average stand diameter of eucalyptus plantations from a spaceborne system. Verification of this work or the employment of the technique can be easily verified from an elementary field based audit without requiring a 'back modelling' of a more complex forest estimate such as volume.

Other methods of machine learning such as a Random Forest Regressor or a SVM with a non-linear kernel would likely outperform this model at predicting diameter with spaceborne lidar. However, there is a downside of using these models. They are not directly communicable and comparable outside of the specific black box that they were created in. However, our model accuracy is reduced at the higher end of the diameter predictions. This is the drawback of using a linear kernel for the SVR as it is unable to account for the reduced growth seen in these stands when the reach ages near the end of their rotation as the changes in diameter become less linear with changes in height. Reduced accuracy at ages 5-7 and beyond is not necessarily detrimental to using this model as the rotation ages of these areas is usually less than ten years.

The linear SVR is a way to communicate the direct results of this specific application in a meaningful and clear way. Linear SVR is a black box method but its results are significantly more transparent than other machine learning methods when used in this way. Our hope is that the data and methods used in this paper will permit governing bodies, land holders, and scientists to accurately measure eucalyptus plantations with a high level of clarity and accuracy while accounting for carbon estimates in this region of the world under Article 6.2 of the Paris Agreement ("Paris Agreement on Climate Change" 2015).

The results of the random forest prediction of harvest year were limited. The variables created and fed into the model failed to predict the harvest year with any discernible accuracy (less than 50%). We suspect that this has a lot to do with a number of factors. The first is that there is uncertainty present in the estimation of last disturbance (Hansen et al. 2013) in that there is often a lag between a disturbance and a logged change in reflectance. Second, is that there are a number of other variables that can influence when a stand is harvested and brought to market that we did not account for in this analysis. These variables could include timber prices, road conditions and transportation costs, or other economic predictors that will all play into the year that the stand is harvested. However, if these values are incorporated we would expect a higher accuracy of harvest timing and prediction

especially with the inclusion of a modeled result of stand volume.

1.5 Conclusions

It is possible to use spaceborne lidar to accurately estimate forest stand parameters; as demonstrated, diameter can be estimated to close to a centimeter of error. Given the spatial auto-correlation in the GEDI measurements, it is paramount to utilize non parametric techniques or account for the spatial auto-correlation when creating models of forest parameters. Predicting forest harvest in these areas can not be done using only stand parameters.

Bibliography

Boczniewicz, Daniel, Euan G. Mason, and Justin A. Morgenroth (2022). "Developing fully compatible taper and volume equations for all stem components of Eucalyptus globoidea Blakely trees in New Zealand". In: *New Zealand Journal of Forestry Science* 52. ISSN: 11795395. DOI: 10.33494/nzjfs522022x180x.

Bonan, Gordon B. (2008). "Forests and climate change: Forcings, feedbacks, and the climate benefits of forests". In: *Science* 320.5882, pp. 1444–1449. ISSN: 00368075. DOI: 10.1126/science.1155121.

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45, pp. 5–32.

Buongiorno, Joseph and Shushuai Zhu (2014). "Assessing the impact of planted forests on the global forest economy". In: *New Zealand Journal of Forestry Science* 44.Suppl 1, pp. 1–9. ISSN: 11795395. DOI: 10.1186/1179-5395-44-S1-S2.

Burkhart, Harold and Margarida Tomé (2012). *Modeling Forest Stands and Trees*. Springer US. ISBN: 9788578110796.

Bye, I. J. et al. (2017). "Estimating forest canopy parameters from satellite waveform LiDAR by inversion of the FLIGHT three-dimensional radiative transfer model". In: *Remote Sensing of Environment* 188, pp. 177–189. ISSN: 00344257. DOI: 10.1016/j.rse.2016.10.048. URL: http://dx.doi.org/10.1016/j.rse.2016.10.048.

Campoe, Otávio Camargo, José Luiz Stape, and João Carlos Teixeira Mendes (2010). "Can intensive management accelerate the restoration of Brazil's Atlantic forests?" In: *Forest Ecology and Management* 259.9, pp. 1808–1814. ISSN: 03781127. DOI: 10.1016/j.foreco.2009.06.026.

Chen, Qi (2010). "Retrieving vegetation height of forests and woodlands over mountainous areas in the Pacific Coast region using satellite laser altimetry". In: *Remote Sensing of Environment*

114.7, pp. 1610–1627. ISSN: 00344257. DOI: 10.1016/j.rse.2010.02.016. URL: http://dx.doi.org/10.1016/j.rse.2010.02.016.

Clark, Kenneth L, Henry L Gholz, and Mark S Castro (Aug. 2004). "Carbon dynamics along a chronosequence of slash pine plantations in north florida". In: *Ecological Applications* 14.4, pp. 1154–1171. DOI: 10.1890/02-5391.

Cook, Rachel L, Dan Binkley, and Jose Luiz Stape (Jan. 2016). "Eucalyptus plantation effects on soil carbon after 20years and three rotations in Brazil". In: *Forest Ecology and Management* 359, pp. 92–98. DOI: 10.1016/j.foreco.2015.09.035.

Coops, Nicholas C. et al. (2021). "Modelling lidar-derived estimates of forest attributes over space and time: A review of approaches and future trends". In: *Remote Sensing of Environment* 260.April, p. 112477. ISSN: 00344257. DOI: 10.1016/j.rse.2021.112477. URL: https://doi.org/10.1016/j.rse.2021.112477.

Cosenza, Diogo Nepomuceno et al. (Sept. 2017). "Site classification for eucalypt stands using artificial neural network based on environmental and management features". In: *CERNE* 23.3, pp. 310–320. DOI: 10.1590/01047760201723032352.

Dubayah, Ralph et al. (June 2020). "The Global Ecosystem Dynamics Investigation: High-resolution laser ranging of the Earth\textquoterights forests and topography". In: *Science of Remote Sensing* 1, p. 100002. ISSN: 26660172. DOI: 10.1016/j.srs.2020.100002. URL: https://linkinghub.elsevier.com/retrieve/pii/S2666017220300018.

Fagan, M. E. et al. (2018). "Mapping pine plantations in the southeastern U.S. using structural, spectral, and temporal remote sensing data". In: *Remote Sensing of Environment* 216.July, pp. 415–426. ISSN: 00344257. DOI: 10.1016/j.rse.2018.07.007. URL: https://doi.org/10.1016/j.rse.2018.07.007.

FAO (2015). *Global Forest Resources Assessment 2015 Desk reference*. ISBN: 9789251088265.

– (2020). *Global Forest Resources Assessment*. Tech. rep. Rome, pp. 1–164. DOI: 10.4324/9781315184487-1.

Fayad, Ibrahim et al. (June 2021). "Assessment of {GEDI\textquoterights} {LiDAR} Data for the Estimation of Canopy Heights and Wood Volume of Eucalyptus Plantations in Brazil". In: *{IEEE} Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14, pp. 7095–7110. URL: http://10.0.4.85/%7BJSTARS%7D.2021.3092836.

García-Gutiérrez, J. et al. (2015). "A comparison of machine learning regression techniques for LiDAR-derived estimation of forest variables". In: *Neurocomputing* 167, pp. 24–31. ISSN: 18728286. DOI: 10.1016/j.neucom.2014.09.091. URL: http://dx.doi.org/10.1016/j.neucom.2014.09.091.

Hansen, M. C. et al. (2013). "High-resolution global maps of 21st-century forest cover change". In: *Science* 342.6160, pp. 850–853. ISSN: 10959203. DOI: 10.1126/science.1244693.

Harris, N, E Goldman, and S Gibbes (2021). *Spatial Database of Planted Trees (SDPT) Version 1.0*. URL: http://www.globalforestwatch.org..

Hua, Fangyuan et al. (2022). "The biodiversity and ecosystem service contributions and trade-offs of forest restoration approaches". In: 4649, pp. 1–28. URL: https://www.science.org/doi/pdf/10.1126/science.abl4649?download=true.

Land Processes Distributed Active Archive Center (LP DAAC), Cole Krehbiel (2021). *GEDI Subsetter*. git.earthdata.nasa.gov. URL: https://git.earthdata.nasa.gov/projects/%7BLPDUR%7D/repos/gedi-subsetter/browse.

Lefsky, Michael A. et al. (1999). "Surface lidar remote sensing of basal area and biomass in deciduous forests of eastern Maryland, USA". In: *Remote Sensing of Environment* 67.1, pp. 83–98. ISSN: 00344257. DOI: 10.1016/S0034-4257(98)00071-6.

Mashian, Sean (2018). *Harvard's Natural Resources Empire (7 min) – Cornell Real Estate Review*. URL: https://blog.realestate.cornell.edu/2018/04/20/harvards-natural-resources-empire-7-min/.

Myers, Norman et al. (2000). "Biodiversity Hotspots for Conservation Priorities". In: *Nature* 403, pp. 853–858. ISSN: 21533660. DOI: 10.1080/21564574.1998.9650003.

Neuenschwander, Amy and Katherine Pitts (2018). "The ATL08 land and vegetation product for the ICESat-2 Mission". In: *Remote Sensing of Environment* 221, pp. 247–259. DOI: 10.1016/j.rse.2018.11.005. URL: https://doi.org/10.1016/j.rse.2018.11.005.

Norfolk, Christopher J. and Thom A. Erdle (2005). "Selecting intensive timber management zones as part of a forest land allocation strategy". In: *Forestry Chronicle* 81.2, pp. 245–255. ISSN: 00157546. DOI: 10.5558/tfc81245-2.

"Paris Agreement on Climate Change" (2015). In: *Unitied Nations Framework Convention on Climate Change*, pp. 1–16. DOI: 10.1201/9781351116589-2. URL: https://unfccc.int/files/meetings/paris_nov_2015/application/pdf/paris_agreement_english_.pdf.

Payn, Tim et al. (Sept. 2015). "Changes in planted forests and future global implications". In: *Forest Ecology and Management* 352, pp. 57–67. DOI: 10.1016/j.foreco.2015.06.021.

Petersen, Rachel et al. (2016). *Mapping tree plantations with mutlispectral imagery: preliminary results for seven tropical countries*. Tech. rep. January. World Resources Institute, pp. 185–205. URL: http://dx.doi.org/10.1016/j.apgeog.2012.06.014%5Cnwww.wri.org/publication/mapping-treeplantations%5Cnhttp://blog.globalforestwatch.org/2016/01/forest-loss-pushes-far-beyond-plantation-boundaries-in-south-america-africa/%5Cnhttp://dx.doi.org/10.1016/j.rse..

Pirard, Romain, Lise Dal Secco, and Russell Warman (2016). "Do timber plantations contribute to forest conservation?" In: *Environmental Science and Policy* 57, pp. 122–130. ISSN: 18736416. DOI: 10.1016/j.envsci.2015.12.010. URL: http://dx.doi.org/10.1016/j.envsci.2015.12.010.

Potapov, Peter et al. (2021). "Mapping global forest canopy height through integration of GEDI and Landsat data". In: *Remote Sensing of Environment* 253.August 2020, p. 112165. ISSN: 00344257. DOI: 10.1016/j.rse.2020.112165. URL: https://doi.org/10.1016/j.rse.2020.112165.

Vapnik, Vladimir (1998). “The Support Vector Method of Function Estimation”. In: *Nonlinear Modeling*. Springer, Boston, MA: Springer Science+Business Media. Chap. 3: The Sup, pp. 55–85. ISBN: 9780262100656. DOI: https://doi.org/10.1007/978-1-4615-5703-6. URL: https://doi.org/10.1007/978-1-4615-5703-6_.

Winjum, Jack K and Paul E Schroeder (Apr. 1996). “Forest plantations of the world: their extent, ecological attributes, and carbon storage”. In: *Agricultural and forest meteorology* 84, pp. 153–167.

references.bib

Appendices

Appendix A

GEDI Data Extraction

A.1 Overview and Description

This Section outlines the process I use to extract GEDI data from the archive, trim, tidy, and manipulate it to get it into a reduced and usable format for most applications. Not every action below is verbatum copy-pasteable. This document is intended to be used as a guide. There will be some differences depending upon the target study area and variables you need to extract. Related to needed variables, there are different levels of the GEDI products. This document will work for levels 1B, 2A, and 2B products. Please keep in mind that there are certain parts of this workflow that require modification depending on what levels you need. I will have what I created based on the needs of my work in the document and explain where to make changes if you need them.

This document also assumes that you have a basic understanding of computer terminal commands. The document uses the phrase 'terminal' and the phrase 'command line' interchangeably. They are OS based designations so please use the appropriate designation for the operating system you are using.

It is also suggested that you take good notes of the commands you are using so that it is easy to go back and change things as your workflow changes. The software Joplin is an excellent option for this. There is more than one way to skin a cat, and this document is one way, and you may find a better way.

Please contact me with any questions at benmiller@vt.edu

A.2 Data Download and Extraction

The GEDI data are stored at https://e4ftl01.cr.usgs.gov/GEDI/. There are two main ways to extract the data, one is to download and then mine the entirety of the archive and two is to download the explicit files that you will know you need and then extract the data. The principles and scripts used in the first method are also used in the second, so I will touch on both in the order that I find them applicable and logical.

A.2.1 Data Download

Both methods need to at one point or another download data from the website above. What I found as a useful tool was wget. There are other bulk download commands that are usable (such as curl) but I found this to be robust and the user base/manual was helpful. Here is what I used to download the entirety of the level 2B archive.

Steps

1. Bulk Data Download (This will download all the files you point it at. If you only want to download the needed files, skip step 5)

   (a) Download wget. If you are on a Mac or Linux machine this should be easy to do. If you are on a Windows computer I downloaded the software from https://eternallybored.org/misc/wget/ and chose the appropriate version and EXE option.

   (b) Place the downloaded EXE into your user folder on the C: drive (Windows). If using

Mac or Linux follow the install directions for this command and directions on how to use the command on those interfaces.

(c) To have fast acces to this system, you will need to have a NASA EarthData account. This will increase your data download speed as well as let NASA know who is using the data. You are welcome to skip this step if you wish to remain anonymous with your data download.

(d) Once you have created an EarthData Account, you will need to setup a wgetrc file. Here is how to go about that on a windows CMD prompt. This may be different on other operating systems. (read the manual https://www.gnu.org/software/wget/manual/wget.html if you have questions.)

```
type NUL >> .wgetrc
```

username and password will need to be changed below based on your credentials.

```
echo http-user=USER_NAME >> .wgetrc
    | echo http-password=
    YOURPASSWORDHERE >> .wgetrc
```

(e) Download the data (If you want to download a selection and not an **ENTIRE ARCHIVE** please skip this step and follow the directions outlined in the following section ("Data Finder"))

```
wget -r -nc -np -nd -A h5 --progress
    =bar:force:noscroll -P
    DRIVELETTER:/where/you/want/your/
    data "https://e4ftl01.cr.usgs.gov
    /GEDI/GEDI02_A.002/"
```

2. Data Finder (Most Efficient) This section will describe how to use the 'GEDI finder' tool that has a manual that can be found at https://lpdaac.usgs.gov/documents/591/GEDIFinder_

UserGuide_v1.0.pdf. This is a web browser too that can be used to find GEDI files that intersect a designated area. I have found this to be best practice as it permits the user to download all the data needed and no more. This saves both time and data storage resources.

(a) Edit the following text to meet your needs (i.e. change the product type(product=), and the bounding box(bbox=) coordinates for your area of study.) https://lpdaacsvc.cr.usgs.gov/services/gedifinder?product=GEDI01_B&bbox=[-20.2,-48.8,-23.2,-46.1]&output=html

(b) Copy the curated text in your browser into a simple text editor and save as a descriptively named text file.

(c) Use the following command to download the currated data to a designated loaction. This assumes you have already setup wget using the above section ("Bulk Data Download"). This will download the files that the GEDI Finder tool determined were within your AOI

```
1                                   wget -nc -A .h5 -r -np -i
                                        DRIVELETTER:/where/you/put/the/
                                        text/file/
                                        descriptively_named_text_file.txt
                                         -P DRIVELETTER:/where/you/want/
                                        your/data
```

These commands will run until completed. If there is an interruption they should restart from where they left off when you call them again. If they are not, please consult the wget manual to ensure the options are set up properly. Once complete the terminal will return to normal operation. To cease download close the window or use the appropriate halt keyboard command.

A.2.2 Data Subset

This section describes how to use the ”Gedi subsetter” script provided by NASA LPDAAC. The data as downloaded above are in .h5 format. This section will use an existing script to retrieve the files into a ’geoJSON’ format. The data as extracted by this process can be parsed into GIS suites such as ArcPro or QGIS using the existing tools these platforms have. I took these files and further extracted the values into a .csv file. This next step is in the following section, if your area is small or you are less progamatically inclined, you can likely stop after this section and use a GIS software to do you analysis. If you want to have a little more direct control over your data analsysis or have a large study area, I reccommend you use the next section of the pipeline.

The GEDI-subsetter tool can be found at https://git.earthdata.nasa.gov/projects/LPDUR/repos/gedi-subsetter/browse and I recommend you read through the instructions on this page so that the values that are being extracted by this tool are understood and clear. You can download the basic tool at the link above. Place the tool in your user folder on the C drive if on windows or your working directory on Mac or Linux. The basic tool is fine at doing what it needs to do. However, if you are processing a large amount of files I recommend that you use the modified version I have created. The main difference between the software I have created and the NASA version is speed and record keeping. The changes I made were to enable parallel processing so that multiple files can be processed at once, and the creation of a log of the files that have finished processing. The orginal nasa script goes through the files one by one in a directory and does not record what files have been processed requireing the double processing of files and increasing the number of files needed to be processed. My modified version can be found on vtrsstorage1 under the filename GEDI_Subsetter_BDM_edit_20210921.py

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
```

```
4  ----------------------------------------------------------------------------------------------

5  GEDI Spatial and Band/Layer Subsetting and Export to GeoJSON Script
6  Author: Cole Krehbiel
7  Last Updated: 04/13/2021
8  See README for additional information:
9  https://git.earthdata.nasa.gov/projects/LPDUR/repos/gedi-subsetter/browse/
10 ----------------------------------------------------------------------------------------------

11 """
12 # Import necessary libraries
13 import os
14 import h5py
15 import pandas as pd
16 from shapely.geometry import Polygon
17 import geopandas as gp
18 import argparse
19 import sys
20 import numpy as np
21 import multiprocess as mp
22
23 # -------------------------COMMAND LINE ARGUMENTS AND ERROR HANDLING
       ---------------------------- #
24 # Set up argument and error handling
25 parser = argparse.ArgumentParser(
26     description='Performs Spatial/Band Subsetting and Conversion to GeoJSON
          for GEDI L1-L2 files')
27 parser.add_argument('--dir', required=True,
28                     help='Local directory containing GEDI files to be
                            processed')
29 parser.add_argument('--beams', required=False, help='Specific beams to be
```

```
    included in the output GeoJSON (default is all beams) \
30                      BEAM0000,BEAM0001,BEAM0010,BEAM0011 are Coverage Beams.
                            BEAM0101,BEAM0110,BEAM1000,BEAM1011 are Full Power
                            Beams.')
31 parser.add_argument('--sds', required=False, help='Specific science datasets (
    SDS) to include in the output GeoJSON \
32                      (see README for a list of available SDS and a list of
                            default SDS returned for each product).')
33 parser.add_argument('--roi', required=True, help='Region of interest (ROI) to
    subset the GEDI orbit to in the output GeoJSON. \
34                      Valid inputs are a geojson or .shp file or bounding box
                            coordinates: ul_lat,ul_lon,lr_lat,lr_lon')
35 parser.add_argument('--cores', required=False, help='Number of requested cores
     for processing.')
36 args = parser.parse_args()
37
38 # ---------------------------------SET ARGUMENTS TO VARIABLES
    ------------------------------------ #
39 # Options include a GeoJSON or a list of bbox coordinates
40 ROI = args.roi
41
42 # Convert to Shapely polygon for geojson, .shp or bbox
43 if ROI.endswith('json') or ROI.endswith('.shp'):
44     try:
45         ROI = gp.GeoDataFrame.from_file(ROI)
46         ROI.crs = 'EPSG:4326'
47         if len(ROI) > 1:
48             print(
49                 'Multi-feature polygon detected. Only the first feature will
                        be used to subset the GEDI data.')
50         ROI = ROI.geometry[0]
```

```
51      except:
52          print('error: unable to read input geojson file or the file was not
                found')
53          sys.exit(2)
54  else:
55      ROI = ROI.replace("'", "")
56      ROI = ROI.split(',')
57      ROI = [float(r) for r in ROI]
58      try:
59          ROI = Polygon([(ROI[1], ROI[0]), (ROI[3], ROI[0]),
60                         (ROI[3], ROI[2]), (ROI[1], ROI[2])])
61          ROI.crs = 'EPSG:4326'
62      except:
63          print('error: unable to read input bounding box coordinates, the
                required format is: ul_lat,ul_lon,lr_lat,lr_lon')
64          sys.exit(2)
65
66  # Keep the exact input geometry for the final clip to ROI
67  finalClip = gp.GeoDataFrame([1], geometry=[ROI], crs='EPSG:4326')
68
69  # Format and set input/working directory from user-defined arg
70  if args.dir[-1] != '/' and args.dir[-1] != '\\':
71      inDir = args.dir.strip("'").strip('"') + os.sep
72  else:
73      inDir = args.dir
74
75  # Find input directory
76  try:
77      os.chdir(inDir)
78  except FileNotFoundError:
79      print('error: input directory (--dir) provided does not exist or was not
```

```
        found')
80      sys.exit(2)
81
82  # Define beam subset if provided or default to all beams
83  if args.beams is not None:
84      beamSubset = args.beams.split(',')
85  else:
86      beamSubset = ['BEAM0000', 'BEAM0001', 'BEAM0010', 'BEAM0011',
87                    'BEAM0101', 'BEAM0110', 'BEAM1000', 'BEAM1011']
88
89  # Define additional layers to subset if provided
90  if args.sds is not None:
91      layerSubset = args.sds.split(',')
92  else:
93      layerSubset = None
94
95  # ------------------------------------SET UP WORKSPACE
        ---------------------------------------- #
96  # Create and set output directory
97  outDir = os.path.normpath(
98      (os.path.split(inDir)[0] + os.sep + 'output')) + os.sep
99  if not os.path.exists(outDir):
100     os.makedirs(outDir)
101
102 # *****BDM EDITS 20210713******* Create and set up processed file list
103 procFiles = os.path.normpath(
104     (os.path.split(inDir)[0] + os.sep + 'processed.txt'))
105 if not os.path.exists(procFiles):
106     with open(procFiles, 'w') as processedFiles:
107         pass
108
```

```
109 # Create list of GEDI HDF-EOS5 files in the directory
110 gediFiles = [o for o in os.listdir() if o.endswith('.h5') and 'GEDI' in o]
111
112 # --------------------DEFINE PRESET BAND/LAYER SUBSETS
    ----------------------------------------- #
113 # Default layers to be subset and exported, see README for information on how
    to add additional layers
114 l1bSubset = ['/geolocation/latitude_bin0', '/geolocation/longitude_bin0', '/
    channel', '/shot_number',
115              '/rxwaveform', '/rx_sample_count', '/stale_return_flag', '/
                 tx_sample_count', '/txwaveform',
116              '/geolocation/degrade', '/geolocation/delta_time', '/geolocation/
                 digital_elevation_model',
117              '/geolocation/solar_elevation', '/geolocation/
                 local_beam_elevation', '/noise_mean_corrected',
118              '/geolocation/elevation_bin0', '/geolocation/elevation_lastbin',
                  '/geolocation/surface_type', '/geolocation/
                 digital_elevation_model_srtm']
119 l2aSubset = ['/lat_lowestmode', '/lon_lowestmode', '/channel', '/shot_number',
     '/degrade_flag', '/delta_time',
120              '/digital_elevation_model', '/elev_lowestmode', '/quality_flag',
                  '/rh', '/sensitivity', '/digital_elevation_model_srtm',
121              '/elevation_bias_flag', '/surface_flag', '/num_detectedmodes',
                  '/selected_algorithm', '/solar_elevation']
122 l2bSubset = ['/geolocation/lat_lowestmode', '/geolocation/lon_lowestmode', '/
    channel', '/geolocation/shot_number',
123              '/cover', '/cover_z', '/fhd_normal', '/pai', '/pai_z', '/rhov',
                   '/rhog',
124              '/pavd_z', '/l2a_quality_flag', '/l2b_quality_flag', '/rh100', '/
                 sensitivity',
125              '/stale_return_flag', '/surface_flag', '/geolocation/degrade_flag
```

```
                ', '/geolocation/solar_elevation',
126             '/geolocation/delta_time', '/geolocation/digital_elevation_model
                ', '/geolocation/elev_lowestmode']
127
128 # -------------------IMPORT GEDI FILES AS GEODATAFRAMES AND CLIP TO ROI
    -------------------------- #
129 # Defines as a function for parallel processing.
130 def t(g):
131 # Loop through each GEDI file and export as a point geojson
132     print(f"Processing file: {g}")
133     gedi = h5py.File(g, 'r')       # Open file
134     gediName = g.split('.h5')[0]  # Keep original filename
135     gedi_objs = []
136     gedi.visit(gedi_objs.append)  # Retrieve list of datasets
137
138     # *****BDM_EDIT 20210713****** Writes the file being processed to the end
        of a text document to
139     # allow easy transfer of already processed files out of the main database
        after the subset has
140     # been done
141     with open(procFiles, "a") as processed_file:
142         processed_file.write("\n" + inDir + gediName + ".*")
143
144     # Search for relevant SDS inside data file
145     gediSDS = [str(o) for o in gedi_objs if isinstance(gedi[o], h5py.Dataset)]
146
147     # Define subset of layers based on product
148     if 'GEDI01_B' in g:
149         sdsSubset = l1bSubset
150     elif 'GEDI02_A' in g:
151         sdsSubset = l2aSubset
```

```
152     else:
153         sdsSubset = l2bSubset
154 
155     # Append additional datasets if provided
156     if layerSubset is not None:
157         [sdsSubset.append(y) for y in layerSubset]
158 
159     # Subset to the selected datasets
160     gediSDS = [c for c in gediSDS if any(c.endswith(d) for d in sdsSubset)]
161 
162     # Get unique list of beams and subset to user-defined subset or default (
        all beams)
163     beams = []
164     for h in gediSDS:
165         beam = h.split('/', 1)[0]
166         if beam not in beams and beam in beamSubset:
167             beams.append(beam)
168 
169     gediDF = pd.DataFrame()  # Create empty dataframe to store GEDI datasets
170     del beam, gedi_objs, h
171 
172     # Loop through each beam and create a geodataframe with lat/lon for each
        shot, then clip to ROI
173     for b in beams:
174         beamSDS = [s for s in gediSDS if b in s]
175 
176         # Search for latitude, longitude, and shot number SDS
177         lat = [l for l in beamSDS if sdsSubset[0] in l][0]
178         lon = [l for l in beamSDS if sdsSubset[1] in l][0]
179         shot = f'{b}/shot_number'
180 
```

```
181         # Open latitude, longitude, and shot number SDS
182         shots = gedi[shot][()]
183         lats = gedi[lat][()]
184         lons = gedi[lon][()]
185 
186         # Append BEAM, shot number, latitude, longitude and an index to the
                GEDI dataframe
187         geoDF = pd.DataFrame({'BEAM': len(shots) * [b], shot.split('/', 1)
                [-1].replace('/', '_'): shots,
188                               'Latitude': lats, 'Longitude': lons, 'index': np.
                                   arange(0, len(shots), 1)})
189 
190         # Convert lat/lon coordinates to shapely points and append to
                geodataframe
191         geoDF = gp.GeoDataFrame(geoDF, geometry=gp.points_from_xy(
192             geoDF.Longitude, geoDF.Latitude))
193 
194         # Clip to only include points within the user-defined bounding box
195         geoDF = geoDF[geoDF['geometry'].within(ROI.envelope)]
196         gediDF = gediDF.append(geoDF)
197         del geoDF
198 
199     # Convert to geodataframe and add crs
200     gediDF = gp.GeoDataFrame(gediDF)
201     gediDF.crs = 'EPSG:4326'
202 
203     if gediDF.shape[0] == 0:
204         print(
205             f"No intersecting shots were found between {g} and the region of
                    interest submitted.")
206         return
```

```
207     del lats, lons, shots
208
209 # ------------------------------OPEN SDS AND APPEND TO GEODATAFRAME
    ---------------------------- #
210     beamsDF = pd.DataFrame()  # Create dataframe to store SDS
211     j = 0
212
213     # Loop through each beam and extract subset of defined SDS
214     for b in beams:
215         beamDF = pd.DataFrame()
216         beamSDS = [s for s in gediSDS if b in s and not any(
217             s.endswith(d) for d in sdsSubset[0:3])]
218         shot = f'{b}/shot_number'
219
220         try:
221             # set up indexes in order to retrieve SDS data only within the
                clipped subset from above
222             mindex = min(gediDF[gediDF['BEAM'] == b]['index'])
223             maxdex = max(gediDF[gediDF['BEAM'] == b]['index']) + 1
224             shots = gedi[shot][mindex:maxdex]
225         except ValueError:
226             # Probably don't need this print statement either. Makes things
                cluttered.
227             # print(f"No intersecting shots found for {b}")
228             continue
229         # Loop through and extract each SDS subset and add to DF
230         for s in beamSDS:
231             j += 1
232             sName = s.split('/', 1)[-1].replace('/', '_')
233
234             # Datasets with consistent structure as shots
```

```
235             if gedi[s].shape == gedi[shot].shape:
236                 beamDF[sName] = gedi[s][mindex:maxdex]  # Subset by index
237
238             # Datasets with a length of one
239             elif len(gedi[s][()]) == 1:
240                 # create array of same single value
241                 beamDF[sName] = [gedi[s][()][0]] * len(shots)
242
243             # Multidimensional datasets
244             elif len(gedi[s].shape) == 2 and 'surface_type' not in s:
245                 allData = gedi[s][()][mindex:maxdex]
246
247                 # For each additional dimension, create a new output column to
                        store those data
248                 for i in range(gedi[s].shape[1]):
249                     step = []
250                     for a in allData:
251                         step.append(a[i])
252                     beamDF[f"{sName}_{i}"] = step
253
254             # Waveforms
255             elif s.endswith('waveform') or s.endswith('pgap_theta_z'):
256                 waveform = []
257
258                 if s.endswith('waveform'):
259                     # Use sample_count and sample_start_index to identify the
                            location of each waveform
260                     start = gedi[f'{b}/{s.split("/")[-1][:2]}
                            _sample_start_index'][mindex:maxdex]
261                     count = gedi[f'{b}/{s.split("/")[-1][:2]}_sample_count'][
                            mindex:maxdex]
```

```
262
263                 # for pgap_theta_z, use rx sample start index and count to
                        subset
264                 else:
265                     # Use sample_count and sample_start_index to identify the
                            location of each waveform
266                     start = gedi[f'{b}/rx_sample_start_index'][mindex:maxdex]
267                     count = gedi[f'{b}/rx_sample_count'][mindex:maxdex]
268                 wave = gedi[s][()]
269
270                 # in the dataframe, each waveform will be stored as a list of
                        values
271                 for k in range(len(start)):
272                     singleWF = wave[int(start[k] - 1)
273                                     : int(start[k] - 1 + count[k])]
274                     waveform.append(','.join([str(q) for q in singleWF]))
275                 beamDF[sName] = waveform
276
277             # Surface type
278             elif s.endswith('surface_type'):
279                 surfaces = ['land', 'ocean', 'sea_ice',
280                             'land_ice', 'inland_water']
281                 allData = gedi[s][()]
282                 for i in range(gedi[s].shape[0]):
283                     beamDF[f'{surfaces[i]}'] = allData[i][mindex:maxdex]
284                 del allData
285             else:
286                 print(f"SDS: {s} not found")
287             # Don't need this print statement. Not useful information.
288             # print(f"Processing {j} of {len(beamSDS) * len(beams)}: {s}")
289
```

```
290         beamsDF = beamsDF.append(beamDF)
291     del beamDF, beamSDS, beams, gedi, gediSDS, shots, sdsSubset
292 
293     # Combine geolocation dataframe with SDS layer dataframe
294     outDF = pd.merge(gediDF, beamsDF, left_on='shot_number', right_on=[
295                     sn for sn in beamsDF.columns if sn.endswith('shot_number')
                            ][0])
296     outDF.index = outDF['index']
297     del gediDF, beamsDF
298 
299     # Subset the output DF to the actual boundary of the input ROI
300     outDF = gp.overlay(outDF, finalClip)
301     del outDF[0]
302 # --------------------------------EXPORT AS GEOJSON
        ---------------------------------------------- #
303     # Check for empty output dataframe
304     try:
305         # Export final geodataframe as Geojson
306         print(f"{g} is being written...")
307         outDF.to_file(f"{outDir}{g.replace('.h5', '.json')}", driver='GeoJSON
                ')
308         print(f"{g.replace('.h5', '.json')} saved at: {outDir}")
309     except ValueError:
310         print(f"{g} intersects the bounding box of the input ROI, but no shots
                intersect final clipped ROI.")
311 
312 #----------------- Starts  the above function as a parallel process
        .-------------------------------#
313 num_comp_cores = mp.cpu_count()
314 
315 if args.cores is not None:
```

```
316     requested_cores = int(args.cores)
317     if requested_cores > num_comp_cores:
318         print("Number of requested cores was greater than avaliable resources,
                defaulting to system max.")
319         requested_cores = num_comp_cores
320 else:
321     requested_cores = 1
322
323 if __name__ == "__main__":
324     p = mp.Pool(requested_cores)
325     p.map(t, gediFiles)
```

The script above automatically extracts different values based on the level of GEDI product that is specified in the file name at the time of download, so if you changed the file names that you downloaded and are having problems at this stage, that would be why. Below I will give an example of using each method.

Setup for both methods

This assumes you are on Windows but the same principles apply on other operating systems.

1. Download [anaconda](https://www.anaconda.com/)

2. Install anaconda

3. Update path to the following (CHANGE PATH TO WHERE YOU PUT ANACONDA):

```
1                   set PATH=%PATH%C:\Users\yourname\Anaconda3\Scripts;C:\
                        Users\yourname\Anaconda3\Library\bin;C:\Users\
                        yourname\Anaconda3\Library;
```

4. Run:

```
conda init cmd.exe
```

5. Restart the shell and repeat the path changes made in a step above if on a enterprise computer.

   If you are on a personal computer you likely only have to reopen the shell.

6. Run:

```
conda create -n gedi -c conda-forge --yes python=3.7 h5py
    shapely geopandas pandas multiprocess
```

7. Run:

```
conda activate gedi
```

8. Copy the subsetter script of your choice to either your working directory.

Basic NASA Method

In a terminal run the following if using a shapefile bounding box:

```
python GEDI_Subsetter.py --dir DRIVELETTER:/where/you/put/your
    /data --roi DRIVELETTER:/where/you/put/your/roi/file.shp
```

Or you can use a bounding box for example this geographic region:

```
python GEDI_Subsetter.py --dir DRIVELETTER:/where/you/put/your
    /data --roi '-20.2,-48.8,-23.2,-46.1'
```

This will subset the h5 data into a JSON file in a folder called "output" within your directory.

The significantly faster method (recommended)

You can set the number of cores you want to use after the '–cores' command or change the roi to a bounding box as above.

```
python GEDI_Subsetter_BDM_edit_20210921.py --dir DRIVELETTER:/
    where/you/put/your/data --roi DRIVELETTER:/where/you/put/
    your/roi/file.shp --cores 5
```

Moving processed files given an interruption

Often when processing these files there was some sort of error. I attempted to handle these by creating a log of the processed files that my modified version of the script creates. This can also help if there is a power loss or some other interruption of processing. The script as it stands does not recognize where it left off, so it starts from the top of the list of files it was given and these files do not always correspond to the order in which they appear in your file directory. Additionally, the files that are 'outputted' are sometimes a subset of the files you are having the script subset. This means that the script will discard files that do not intersect within the extent of the bounding geometry and skip them without creating an output file.

To work around this issue, I had the script append to a list of the files it has processed. The modified script appends each file subset (or discarded) to a file in the directory where you pointed the script called "processed.txt". The utility of this is that after a processing interruption you are able to have the script 'skip' the files it has already looked at by moving those files to a different folder. I typically created a folder of the processed files outside of the directory (i.e. not a subdirectory) containing the files that had already been processed. I then use the following command line command (windows, will be different for UNIX systems) to move the files that are on the list of processed files to the 'done' folder. This command does assume that you have navigated to the directory that

you are processing out of. If you are not in that directory simply add an appropriate path to the ‘processed.txt‘ file.

```
FOR /f "delims= " %F IN (processed.txt) DO MOVE "%F" "Z:\PATH\
    TO\WHERE\YOU\ARE\PUTTING\PROCESSED\FILES"
```

This will run through the list of files you have already processed. If you have a massive amount of files present in this directory it will save time to skip the files that you have already moved as the script will simply append the files it has processed to ‘processed.txt‘. If so, execute the following command using the number of files that are in the ’processed files directory’ for ”skip”. The alternative is delete the ‘processed.txt‘ file after every file moving instance. The modified version of the script will just create a new one.

```
FOR /f "delims= skip=12345" %F IN (processed.txt) DO MOVE "%F"
    "Z:\Data\GEDI\GEDI02_B.002\processed"
```

A.2.3 Data Extraction

The above method extracts files into a folder names ”output” that contains geojson files. These contain a lot of data and information. The information can all be extracted further from the geojson into the scripting language of your choice or using a GIS software product. I used a language to extract the data I needed into csv format and then intersected those data with existing geospatial products to perform my analysis. The extraction choices and the potential use cases are broad enough at this point that l that I will not go into more detail as they are different depending on the product level and the intended use.