**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Thomas Poncy

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

Valeria A. Thomas (Co-advisor)

Randolph H. Wynne (Co-advisor)

P. Corey Green

May 13, 2025

Blacksburg, VA

Keywords: remote sensing, forests, LiDAR, ICESat-2

Copyright 2025 William T. Poncy

**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Poncy

ABSTRACT

Accurate quantification of forest structure is necessary for forest inventory, growth prediction, and carbon stock estimates. Since late 2018, the ICESat-2 mission has employed a photon-counting LiDAR system to estimate along-track ground and canopy heights above the WGS 84 ellipsoid. This mission provides repeat coverage in 100 m data segments across the Southeastern United States, a region selected for its high timber productivity. In this research, we use multiple years of ICESat-2 ATL08 data to identify canopy height growth over time in a coastal region of North Carolina. ATL08 canopy height estimates demonstrate strong alignment (R2=0.88, RMSE=2.64 meters) with coincident airborne laser scanning. Equivalence tests reveal that locations sampled by ICESat-2 are homogenous within a margin of 2 meters. FIA remeasurement plots within our study area provide an expected net growth of 1.68 meters over a five-year timeframe, and a rate of change of 0.34 meters per year. Exchanging space for time reveals canopy height growth is detectable with five years of ATL08 data. However, the influence between land cover type and disturbance history on canopy height growth is uncertain, likely due to convolved nature of these factors and sampling pattern of ICESat-2 data collection. Ultimately, this research can serve as a proof-of-concept for using multiple years of spaceborne LiDAR to detect changes in forest structure. Future research should explore the detection of canopy height growth with spaceborne LiDAR in other regions of the globe, as solid results could demonstrate the potential of the ATL08 product in global forest structural monitoring in the face of a changing climate.

**Tracking Forest Height Growth Over Time with ICESat-2 ATL08**

William Poncy

GENERAL ABSTRACT

~~To accomplish this we 1) validate ATL08 canopy height measurements with coincident airborne lidar data across our study area. Next, we 2) assess the homogeneity of our study area forests to identify spatial biases in canopy height. After, we 3) establish expected rates of change in canopy height from FIA remeasurement plots. We then 4) exchange space for time, and use multiple years of ATL08 data to identify the emergence of canopy height change across our study area. Finally, we 5) remove observations from locations demonstrating spatial bias in canopy height, to assess how much canopy height change is driven by spatial bias. Exchanging space for time shows statistically significant trends in height change, namely growth, over time in the study area. Removing observations demonstrating spatial bias shows muffled trends in height change over time, likely due to a shorter timeframe and smaller sample size. Further research should explore the emergence of temporal trends in other regions of the globe and across ATL08 collection characteristics, as solid results could demonstrate the potential of the ATL08 product in tracking forest height change over time.~~

**Table of Contents**

List of figures…..……………………………………………………………………….…v

List of tables….…………………………………………………………………………...vi

1. Introduction…………………………………………………………………………....1
   1. Forests………………………………………………………………………….....1
   2. Forest Structure…………………………………………………………………...1
   3. Forest Structure with LiDAR……………………………………………………..2
   4. Airborne Laser Scanning……………………………………………………...….3
   5. Spaceborne LiDAR……………………………………………………………….4
   6. ICESat-2………………………………………………………………………….5
   7. Research Questions……………………………………………………………….7
2. Methods………………………………………………………………………………..7
   1. Study Area………………………………………………………………………..7
   2. Data Products……………………………………………………………………..9
      1. ALS………………………………………………………………………..9
      2. ICESat-2 ATL08…………………………………………………………..9
      3. National Land Cover Database…………………………………………..10
      4. Landscape Change Monitoring System………………………………….11
      5. Forest Inventory & Analysis Program…………………………………...12
   3. Data Processing…………………………………………………………………12
      1. ICESat-2 ATL08…………………………………………………………12
      2. Extracting ALS Data Within ATL08 Segments…………………………14
      3. Retrieving Study Area FIA Data…………………………………………18
   4. Statistical Methods Used………………………………………………………..20
      1. Linear Regression Model………………………………………………...20
      2. Equivalence Test…………………………………………………………21
      3. Wilcoxon Rank Sum Test……………….……………………………….21
      4. Mann Kendall Test……………………………………………………….22
      5. Bonferroni Correction……………………………………………………23
   5. Phase 1: Validating ATL08 Canopy Heights with ALS………………………...23
   6. Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection……………...24
      1. Sample Representation………………….………………………………..24
      2. Sample Location Equivalence……….……………………………...……25
   7. Phase 3: FIA-Derived Tree Height Growth…………………………………….26
   8. Phase 4: Assessing Canopy Height Growth with ATL08……………………....27
3. Results………………………………………………………………………………..28
   1. Phase 1: Validating ATL08 Canopy Heights with ALS………………………...28
   2. Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection ...…………...28
      1. Sample Representation……….…………………………………………..28
      2. Sample Location Equivalence……………………………………………29
   3. Phase 3: FIA- Derived Tree Height Growth…………………………………….30
   4. Phase 4: Assessing Canopy Height Growth with ATL08………….…...………32
4. Discussion……………………………………………………………………………37
5. Conclusion…………………………………………………………………………...43
6. Bibliography…………………………………………………………………………45
7. Appendix……………………………………………………………………………..69
   1. Python Script to Extract Canopy Height Metrics from ALS Data within Working Set of ATL08 Segments…………………………………………………………69
   2. Python Script to Extract Canopy Height Metrics from ATL08 Granules………78
   3. Python Script to Filter ATL08 Data and Extract Land Characteristics…………87
   4. Python Script to Create Polygons of ATL08 Segments………………………...96
   5. R Script to Access North Carolina FIADB and Conduct Analyses of Phase 3..103
   6. R Script to Conduct Analyses of Phase 1……………………………………...117
   7. Python Script to Generate Population of Forests Throughout Study Area…….121
   8. Python Script to Extract Canopy Height Metrics from ALS Data within Population of Study Area Forests……………………………………………...127
   9. R Script to Conduct Analyses of Phase 2……………………………………...136
   10. . R Script to Conduct Analyses of Phase 4………………………………………146
   11. R Script to Generate Level II Ecoregion Proportional Coverage from Annual ATL08 Samples………………………………………………………………...172

**List of Figures**

**Figure 1.** Study area in North Carolina……..………………………………………….....8

**Figure 2.** ATL08segments within study area after data quality filters………………….14

**Figure 3.** Workflow for extracting ALS data within ATL08 polygons…………………16

**Figure 4.** ALS data extracted within ATL08 segments…………………………………17

**Figure 5.** Overlapping collection scheme of ALS projects……...……………………...18

**Figure 6.** Counties of North Carolina within study area……...…………………………19

**Figure 7.** Spatial distribution of each year’s ATL08 segments across study area…..…..24

**Figure 8.** Population of forests throughout study area…………………………………..25

**Figure 9.** Correlation of canopy heights estimated by ATL08 and ALS………………..28

**Figure 10.** Representation of ATL08 annual samples..…………………………………29

**Figure 11.** Grid of sample location equivalence across bound values…………………..30

**Figure 12.** Distribution of FIA-derived tree height increment...……………………...…31

**Figure 13.** Distribution of FIA-derived net growth in tree height………………………32

**Figure 14.** Results of ATL08-derived net growth in canopy height.....….……………...33

**Figure 15.** Results of ATL08-derived incremental growth in canopy height……...……35

**Figure 16.** Results of ATL08-derived monotonic growth in canopy height…….………37

**Figure 17.** ATL08 canopy height estimation in disturbed vs. undisturbed stands………41

**Figure 18.** Proportion of ATL08 segments in EPA Level II Ecoregions by year….……42**List of Tables**

**Table 1.** Summary statistics of FIA-derived increment and net growth in tree height.....31

1. **Introduction**
   1. **Forests**

Supporting forest ecosystems is critical to combating climate change (Psistaki et al., 2024). Forests are the backbone of the land carbon sink (Pan et al., 2024), covering ~31% of the terrestrial land area and sequestering almost twice as much carbon as they emit (FAO and UNEP, 2020; Harris & Gibbs, 2021). Forests impact not only global carbon cycle dynamics but mediate local and microclimates through evapotranspiration and incident solar radiation. Furthermore, forests provide notable benefits to human health, lowering blood pressure and muscle tension while facilitating recovery from attention fatigue (Karjalainen et al., 2009).

However, disturbance regimes serve to alter forest structural traits (height, crown closure, etc.) thereby disrupting the age structure and spatial continuity of forests (Sturtevant & Fortin, 2021). These disturbances can result in stress-induced mortality, changes in species composition, and reduced tree regeneration capacity (Khaine & Woo, 2015). Disturbance regimes themselves are sensitive to climatic shifts, with warming temperatures likely to increase the frequency of fire, insect, and pathogen disturbances in coming years. Despite these rising pressures, current modeling techniques inadequately represent in the complex interplay between vegetation dynamics, climate pattens, and disturbance agents (Seidl et al., 2017). To effectively assess ecosystem resilience in the face of a changing disturbance landscape, dynamic ecosystem models require information on forest structural diversity over large scales (Mitchell et al., 2023).

* 1. **Forest Structure**

Forest structure is generally characterized by the horizontal and vertical distribution of its plant and organic components throughout space, described by variations in species and age classes, canopy layers, and diameter classes (Zenner & Hibbs, 2000). Forest structure emerges from the interplay of biophysical and ecological processes throughout time, and has regulatory effects on the microclimate and quality of ecosystem services (Hui et al., 2019). Variables describing forest structure, such as basal area and tree height heterogeneity, have demonstrated greater influence on forest productivity than species diversity (Bohn & Huth, 2017). Despite variability across ecoregions, forest structural complexity is positively correlated with ecosystem stability and productivity (X. Liu et al., 2024). However, uncertainties remain in the 3D structure of global forests, and by extension, the global carbon budget (Hall et al., 2011). As such, accurate quantification of global forest structure is necessary to model ecosystem processes, carbon allocation, and growth dynamics to enhance forest resilience in response to mounting disturbance pressures (Beland et al., 2019).

* 1. **Forest Structure with LiDAR**

Light detection and ranging (LiDAR) is an active remote sensing technique that uses laser pulses to calculate distance to objects and map out in high detail the environment surrounding the sensor (Harrap & Lato, 2010). Importantly, LiDAR data avoids the georeferencing of optical remote sensing-where the 3D world is projected onto an 2D image-and the inevitable geometric distortion (P. Dong & Chen, 2017). The waveform of LiDAR energy returned to the sensor can be recorded as peaks in the waveform curve (discrete return), arrival time of individual photons reflected from the laser pulse (photon counting), or the entire distribution (full waveform) (Podest, 2021). LiDAR sensors can be mounted to terrestrial, airborne, and spaceborne platforms, resulting in application across domains for efforts in landslide detection, autonomous vehicle navigation, and archaeological discovery (Jaboyedoff et al., 2012; Risbøl & Gustavsen, 2018; Roriz et al., 2022).

LiDAR has seen growing use in remote sensing of forests, primarily due to its rapid collection and precise quantification of 3D forest structure. Whereas passive remote sensing techniques cannot penetrate the canopy layer, LiDAR can describe the subcanopy vegetation and underlying topography of a forest ecosystem, which have strong influence on drought response and post-disturbance stability (Jarron et al., 2020; H. Zhao et al., 2022). LiDAR is also used for estimating aboveground biomass of forest ecosystems more precisely than optical sensors, an essential task for quantifying the global carbon budget and upholding international climate and emissions agreements (Neuenschwander et al., 2023; Zolkos et al., 2013). Commonly, LiDAR data is used to describe the structural complexity of forests by quantifying canopy cover, foliage height diversity, top rugosity, leaf area index, and leaf area density, which can then be used to infer ecosystem functions (Atkins et al., 2023). This study will focus on canopy height as the primary forest structural characteristic, as tree height is data is fundamental to forest management activities for estimating stem volume, describing stand health, and modeling forest biomass (F. Chen et al., 2023; Mielcarek et al., 2020).

* 1. **Airborne Laser Scanning**

For research at scales larger than forest stands, remotely sensed data is often combined with field inventories to generate a comprehensive view of forest structure (Lamping et al., 2021). Airborne laser scanning (ALS) is a form of LiDAR remote sensing that distributes emitted light along an aircraft’s flight path (Maltamo et al., 2014). ALS sensor technology can collect data at hundreds of points per square meter, leading to a growing adoption for quantifying vegetation structure by overcoming difficulties of traditional terrestrial sampling techniques in remote or topographically complex areas (Sumnall et al., 2022; Wilkes et al., 2015). Data from ALS is also used in forest inventory efforts, commonly with an area-based-approach or through individual tree detection (Xiang et al., 2024). The spatial coverage offered by airborne laser scanning is invaluable for forest research across ecological gradients, with programs like the U.S. Geological Survey’s (USGS) 3D Elevation Program and the National Ecological Observatory Network’s Airborne Observation Platform providing freely available LiDAR data for a range of forested environments.

While the accuracy of height estimations from LiDAR are dependent upon the forest species composition, airplane flight height, and scanner pulse density, LiDAR sensors may underestimate tree height due to the unlikely nature of laser pulses returning from the top of a tree (X. Yu et al., 2004; K. Zhao et al., 2018). While single-area unit ALS forest measurements may demonstrate bias, site-index and aboveground biomass estimations improve as spatial scale and plot size increase (Duncanson et al., 2020; Meyer et al., 2013; Noordermeer et al., 2018, 2020). Multiple studies have found that tree height can be estimated by ALS to within half a meter for pine species (Andersen et al., 2006; Roberts et al., 2005). ALS from the USGS 3D Elevation Program, which is used in this study, has demonstrated an error of 1-2 meters for estimating tree height and dominant height of forests (Oh et al., 2022; Ribas-Costa, Gastón, & Cook, 2024).

Ultimately forest height growth is itself heavily influenced by external factors, and estimating year-over-year height growth with ALS could be subject to non-trivial error (Guerra-Hernández et al., 2021; Socha et al., 2017). Identifying height growth with repeat ALS requires ample time between collections for the growth to exceed the noise associated in single-year height measurements. Yu et al. (2004) observed that plot-level growth could be ascertained at a scale of 10-15 centimeters in a 21-month timeframe. In a study of spruce, pine, and birch, Hyyppä et al. (2003) found a standard error of less than 5 centimeters in estimating height growth at the stand level. In a stand of temperate, mature red pine, Hopkinson et al. (2008) observed an approximate growth rate of 0.4 meters per year, and that the LiDAR-estimated growth error falls below an acceptable uncertainty value of 10% after 3 years.

Despite impressive performance in estimating forest height and growth, ALS data is not without its shortcomings, with collection costs prohibiting repeat or global coverage (Hancock et al., 2021; M. Liu et al., 2019). Moreover, differences in ALS acquisition parameters, namely point density, leads to variability in forest structural estimation (LaRue et al., 2022)

* 1. **Spaceborne LiDAR**

Estimates of forest aboveground biomass and density are critical to understanding the impacts of land use change on the global carbon cycle (Dubayah et al., 2022). To overcome limitations present in ALS, such as inconsistent collections and lack of wall-to-wall coverage, spaceborne LiDAR missions have been commissioned facilitating regional-to-global scale ecosystem insights (Coops et al., 2021). These missions exchange repeated measurements of small, specific areas for broad spatial coverage, sampling earth’s surface during continued orbit throughout time. When harmonized with other data sources (optical imagery, ALS), spaceborne LiDAR data enabled creation of create gridded biomass products at national and regional scales, filling gaps of field-based or ALS campaigns (F. Chen et al., 2023; Dubayah et al., 2022; Neuenschwander et al., 2024)

NASA’s Ice, Cloud and Land Elevation Satellite (ICESat) mission collected global waveform LiDAR data from 2003 to 2009 with a laser altimeter system, demonstrating strong performance in elevation retrieval with relative accuracy and precision of ~2 and ~14 centimeters (Gong et al., 2011; Shuman et al., 2006). While the ICESat mission’s main objectives were to quantify changes in ice sheet elevation (Markus et al., 2017), the derived data products saw use in efforts measuring land topography, vegetation canopy heights, and atmospheric composition (Schutz et al., 2005). More recently, NASA launched the Global Ecosystem Dynamics Investigation (GEDI) mission in late 2018 to directly assessing land surface carbon balance and biodiversity using waveform LiDAR (Dubayah et al., 2020). Building on the success of the first ICESat mission, NASA also launched the ICESat-2 mission in September 2018 to continue measurements of ice sheet elevation through improvements in beam design, along-track sampling rate, and footprint diameter (Markus et al., 2017).

* 1. **ICESat-2**

The ICESat-2 satellite employs the Advanced Topographic Laser Altimeter System (ATLAS), a photon-counting LiDAR instrument, to sample earth’s surface (Carabajal & Boy, 2020). ATLAS operates at a 532 nm (green) laser wavelength, optimized for maximal photon detection with current technology (Neumann et al., 2019a). Prior to exiting the satellite, ATLAS splits the laser into six beams arranged into three beam pairs, with each beam pair containing a strong and weak beam with an energy transmission ratio of 4:1 (Neumann et al., 2019a)

For greater coverage of Earth’s surface, beams within a pair are separated by 90 meters and beam pairs are separated by ~3 kilometers (Markus et al., 2017). ATLAS’s reduced laser power requirement allows a smaller payload aboard the ICESat-2 bus (Sun et al., 2020). With a spacecraft velocity of ~7 km/s and a laser frequency of 10 kHz, the ATLAS instrument achieves an along-track sampling interval of 70 centimeters. This generates strong overlap between shots to determine terrain slope along and across the orbital track. In contrast to the GEDI mission, which collects data only within latitudes ±51.6°, ICESat-2 provides near-global coverage between 88° N and S (Markus et al., 2017; Pronk et al., 2024). Moreover, ICESat-2’s orbit altitude of ~500 km and 91-day repeat cycle facilitate analyses of seasonal variation for its coverage areas (Wang et al., 2024).

Ultimately, the consistency of surface measurements by the ATLAS instrument is influenced by atmospheric conditions and land cover attributes. Over highly reflective surfaces like land ice, up to 10 signal photons may be returned per laser pulse. Over vegetated areas with lower surface reflectance, the energy and detection ratio of ATLAS may drop to ~2.5:1, with only 0-4 signal photons returned per laser pulse (Neuenschwander et al., 2023, 2024).

Though primarily designed for ice sheet monitoring, products for ICESat-2 data have been developed for characterizing ocean elevation, inland water height, and terrestrial vegetation. This research will employ the ICESat-2 L3A Land and Vegetation Height (ATL08) Version 6 data product, which estimates ground and canopy elevations in 100-meter segments.

Commonly, assessing the performance of ATL08 segments in estimating canopy height requires comparison against coincident ALS. While studies commonly report an underestimation of canopy heights by ATL08 segments, the magnitude of error varies with forest conditions. Comparison of ATL08 canopy height estimations to ALS data has shown mean biases ranging from -1.71 meters across diverse biomes, with strong performance in conifer forests, to 3.05 meters in boreal forests (A. Liu et al., 2021; Malambo & Popescu, 2021; Neuenschwander et al., 2020).

ATL08 segments collected at nighttime with the strong beam of the satellite consistently yield the lowest height estimation errors due to reduced atmospheric interference and greater energy output. Use of weak beam data is generally discouraged for canopy height estimation (F. Chen et al., 2023; Guerra-Hernández et al., 2022; A. Liu et al., 2021; Neuenschwander et al., 2020; Rai et al., 2024; J. Yu et al., 2022). Moreover, canopy height estimation errors are minimized in a range of 40-80% canopy cover. Sparse vegetation has an inherently low probability of generating returns to adequately describe canopy height, while in dense canopy cover the algorithm for the ATL08 data product struggles to interpolate the underlying terrain, introducing error into the resulting canopy height estimation (Neuenschwander et al., 2020).

Ultimately, the accuracy of ATL08 forest height estimation is a function of many factors, including acquisition characteristics (e.g., beam strength, time of collection), terrain variability, forest density, geolocation accuracy, atmospheric noise, local disturbance history, and temporal coincidence to reference data (Fernandez-Diaz et al., 2022; Malambo & Popescu, 2021; Neuenschwander et al., 2023; Rai et al., 2024).

These challenges notwithstanding, ICESat-2’s near-global coverage and high geolocation accuracy (<5m) offer a unique advantage over GEDI to track forest dynamics over time (Luthcke et al., 2021; Neuenschwander & Magruder, 2019). The question remains as to whether spaceborne LiDAR can fully describe changes in vegetation structure when and where ALS data is not available (B. Li et al., 2022; Malambo & Popescu, 2021). By leveraging its strengths and effectively handling the limitations in its data quality, this study aims to showcase ICESat-2’s potential for forest structural monitoring in the context of a shifting climate.

* 1. **Research Questions**

Our research aims to use multiple years of spaceborne LiDAR data to monitor forest height growth. To that end, we aim to answer the following research questions:

1. To what extent can ICESat-2 ATL08 detect forest growth over time?
2. How is this ability influenced by forest cover type and disturbance history?
3. **Methods**

Our approach for answering the research questions is organized into four sequential phases-validation, addressing bias, FIA reference, and growth detection-each described in subsections below.

* 1. **Study Area**

Hurricane Florence, a slow moving category one hurricane, made landfall in coastal North Carolina on September 14th, 2018 (Callaghan, 2020). Florence brought record-breaking levels of rain, surpassing 30 inches of rain in some regions of North Carolina, resulting in 53 fatalities and an estimated $16-44 billion in damages to public and private infrastructure (Griffin et al., 2020; Paul et al., 2019). In response, the USGS 3D Elevation Program (3DEP), under the Additional Supplemental Appropriations for Disaster Relief Act of 2019 (H.R. 2157), collected high-resolution ALS data to quantify the extent of hurricane damage and support infrastructure recovery efforts (Columbia Environmental Research Center, 2019). The study area of this research is defined as the collective boundaries of the 3DEP Hurricane Florence ALS projects (Figure 1). The study area comprises ~5.1 million total hectares in the Eastern region of North Carolina.

A map of canada with different colored squares

AI-generated content may be incorrect.

*Figure 1: Boundaries of USGS 3D Elevation Program ALS projects collected in response to Hurricane Florence. Projects were flown in late 2019 and early 2020.*

This study area was selected for several factors. While repeat coverage by ICESat-2 is unlikely for the stand-or-plantation-scale, the chosen study area is large enough to ensure repeat coverage even after data quality filters. Moreover, the collection of data to validate ATL08 canopy height estimations can be financially cumbersome. However, the chosen study area contains extensive ALS data across ecological gradients.

Topography has noticeable effects on ATL08 canopy height estimations (A. Liu et al., 2021; Malambo & Popescu, 2021). The study area demonstrates low terrain variability, with an elevation standard deviation of 27.3 meters and an average slope of 0.78 degrees. As of 2023, the land cover of the study area is predominantly Woody Wetlands (29%), Cultivated Crops (26%), and Evergreen Forest (19%) as determined from the National Land Cover Database. Finally, the chosen study area falls within the P*inus taeda* (loblolly pine) historical natural range. As such, the regular tree spacing, consistent tree heights, and fast tree growth observed by plantations in the study region should facilitate accurate canopy height estimation by ATL08 segments (Baker & Langdon, 1990).

* 1. **Data Products**

This research utilizes ICESat-2 ATL08 canopy height data to assess forest height growth over time. Land cover data from the Annual National Landcover Database was used to identify forested regions within the study area. The disturbance history of these forests was derived from the U.S. Forest Service’s Landscape Change Monitoring System. ALS data from the USGS 3DEP was used as reference data to validate ATL08 canopy height estimates and assess spatial biases in ICESat-2 coverage locations. Finally, data from the U.S. Forest Service’s Forest Inventory and Analysis program provided local, repeat measurement plot surveys for determining expected canopy height growth over time within our study area.

* + 1. **ALS**

ALS data from the Hurricane Florence region is contained in 11 projects of the USGS 3DEP flown between December 7th, 2019, and February 28th, 2020. All projects utilized Reigl VQ1560i or Reigl VQ1560ii instruments to collect data, meeting the Quality Level 1 requirements specified by the 3DEP LiDAR Base Specification. This guarantees that reported heights have an average error (RMSE) within 10 centimeters, and an average point density of 8 points per square meter. In consideration of the strong vertical accuracy and high point density of this ALS data, we determined it to be adequate in serving as reference for ATL08 canopy height estimations. USGS 3DEP LiDAR Point Clouds was accessed on May 17, 2025, from https://registry.opendata.aws/usgs-lidar. A custom Python script (Appendix 7.1) was created in Python 3.12.5 (Rossum & Drake, 2010) to stream the 3DEP data from the Entwine Point Tiles format in during data processing.

* + 1. **ICESat-2 ATL08**

Photon measurements from the ATLAS instrument are aggregated into several data products. The Global Geolocated Photon Data (ATL03) product records heights above the WGS84 ellipsoid for photons detected by ATLAS (Neumann et al., 2023). From the geolocated photon data, the ICESat-2 L3A Land and Vegetation Height (ATL08) data product is derived through a Differential, Regressive, and Gaussian Adaptive Nearest Neighbor (DRAGANN) method. This process filters out background noise and estimates the land surface and vegetation height, labeling individual photons as noise, ground, canopy, or top of canopy (Malambo & Popescu, 2024; Neuenschwander et al., 2020; Neuenschwander & Pitts, 2019).

The ATL08 product is used in this research as it reports several canopy parameters in 20-meter and 100-meter segments, including the mean, minimum, maximum, and median of relative and absolute height of canopy photons. Metrics are also reported for relative and absolute canopy heights at the following percentiles: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95. ATL08’s primary canopy height metric, *h\_canopy*, uses a segment’s 98th percentile canopy relative heights to represent the top of canopy, as maximum canopy height may include background noise (Neuenschwander & Pitts, 2019).

For this study, all ATL08 granules collected between October 14th, 2018, and December 24th, 2024 within the study area were downloaded from the NASA Earthdata Search (<https://search.earthdata.nasa.gov/search>), yielding 831 ATL08 granules for further analysis in the HDF5 file format. These granules were processed in a custom Python script (see Appendix 7.2 for code) to extract metrics related to the vegetation, terrain, and spacecraft operation of each 100-meter segment

* + 1. **National Land Cover Database**

To aid research on land surface characteristics within human-environmental systems, the USGS Resources Observation and Science Center developed the National Land Cover Database (NLCD). This suite of data products used Landsat imagery to provide information on land cover and change processes from 2001 to 2021. NLCD products have been used widely in remote sensing research, with applications in land management, risk mitigation, and environmental planning (Homer et al., 2020). In 2024, the USGS released the Annual National Land Cover Database Collection 1 Science Products, improving legacy NLCD datasets by providing annual land cover classification and land change assessments for each year from 1985 to 2023 (U.S. Geological Survey, 2024). These products are created through a framework that combines continuous change detection, geospatial deep learning, and probability-based post-processing.

Within the collection, the Land Cover product categorizes the land surface into broad natural and artificial surface cover types for each mapping year, employing a modified Anderson Level II classification system to identify 16 land cover classes. While ATL08 data includes segment land cover information in the *segment\_landcover* attribute-derived from the Copernicus Land Cover (ANC18) data product at a 100-meter resolution-the NLCD Land Cover dataset was chosen for this research for its higher spatial resolution and alignment with historical disturbance data provided by the U.S. Forest Service. The NLCD data used in this research was accessed from the Multi-Resolution Land Characteristics Consortium data archive (https://www.mrlc.gov/data). Land cover for the contiguous United States was downloaded for the most recent year available, 2023, to provide the most up-to-date representation of forest cover within the study area. This data is provided as a single-band raster in the TIFF file format, with a 30-meter cell spatial resolution

* + 1. **Landscape Change Monitoring System**

The Landscape Change Monitoring System (LCMS) is a suite of remotely sensed-data products developed by the U.S. Forest Service to monitor landscape dynamics across the United States (Housman et al., 2021). LCMS provides three annual products using an ensemble of predictor models to characterize ecosystem processes over four decades: change, land cover, and land use.

LCMS relies on input imagery from Landsat 4, 5, 7, and 8 (Collection 1 Tier 1) and Sentinel-2a and -2b (level 1C top of atmosphere reflectance data). From these inputs, clouds and cloud shadows are masked with the CFmask, cloudScore, 2Cloudless, and Temporal Dark Outlier Mask algorithms (Chastain et al., 2019; Foga et al., 2017; Housman et al., 2021). The processed outputs are assembled into an annual time series and segmented using the LandTrendr and CCDC methods (Kennedy et al., 2010, 2018; Zhu & Woodcock, 2014). The segmented data is incorporated with terrain data in a random forest model to generate predictions for land change, land cover, and land use.

The land change product is focused on vegetation cover, with subproducts categorizing change processes into fast loss, slow loss, and gain. Fast loss events refer to an abrupt disturbance to vegetation cover (fire, harvesting, mechanical disturbance), slow loss events indicate gradual structural or spectral decline, and gain represents vegetation recovery. A summary product is provided for each of these change processes at a 30-meter spatial resolution, with pixel values identifying the most recent year of occurrence of the given change between 1985 and 2023. For this research, the Fast Loss change summary product was downloaded for the Conterminous United States from the Landscape Change Monitoring System data archive (https://apps.fs.usda.gov/lcms-viewer/home.html#download).

* + 1. **Forest Inventory & Analysis Program**

The Forest Inventory & Analysis (FIA) program is an effort by the U.S. Forest Service’s Research and Development Branch aimed at monitoring the distribution and health of forest resources. With data collected in all 50 states, U.S. territories, and Freely Associated States (Republic of Marshall Islands, Federated States of Micronesia, Republic of Palau), FIA provides data on forest resources domestically and abroad. A core tenet of the FIA is the Nationwide Forest Inventory (NFI), a nationally distributed network of forest inventory plots that receive repeat measurements every 5-10 years. Depending on the plot, the NFI provides data on land use, soil, down woody material, understory vegetation, and individual tree height measurements. Though precise FIA plot locations are fuzzed for privacy concerns (U.S. Department of Agriculture Forest Service, 2024), the county of each plot record is provided with Census Bureau FIPS codes, a level of spatial-specificity sufficient for this research

* 1. **Data Processing**

All spatial data sources, with exception of FIA data, were first projected from their native coordinate systems to the NAD 1983 North Carolina State Plane projection system (EPSG:32119). This projection system maintained a horizontal distance unit of meters- necessary for comparing ATL08 and ALS canopy height estimations-while minimizing geometric distortion across the extent of the study area in North Carolina.

* + 1. **ICESat-2 ATL08**

Using LiDAR to identify statistically meaningful growth in canopy height over time requires growth significant enough to surpass the margin of error associated with single-point-in-time canopy height estimations. In the case of ICESat-2, this is especially challenging given the issues of atmospheric noise sensitivity, probability of canopy height underestimation by the ATL08 algorithm, and sampling scheme of the satellite. As such, careful handling of all potential sources of error in ATL08 data is necessary to optimize its application for given analyses (Feng et al., 2023; B. Li et al., 2022; W. Li et al., 2020; A. Liu et al., 2021; Rai et al., 2024; Wang et al., 2024; Xi et al., 2022).

ATL08 data was processed in a custom Python script to filter observations and intersect with land cover characteristics (see Appendix 7.3 for code). First, ATL08 segments with invalid (3.402e+38) canopy height values were removed. To minimize biases introduced by low vegetation or high noise, canopy heights were then filtered to values between 2 meters and a global maximum (Cao et al., 2016; Rai et al., 2024). Following the approach of Malambo & Popescu (2024), a global maximum for canopy heights was calculated as *kP, k* is a constant factor (1.2) and *P* is the 98th percentile of all ATL08 canopy heights in our dataset. For our study area, the calculated global maximum was 39.42 meters. This was effective in removing erroneously high canopy heights estimated by the ATL08 algorithm. Additional filters selected only segments collected by strong beams, at nighttime, within the months of May to September to minimize the effects of solar noise and vegetation phenology on canopy height estimation (Neuenschwander et al., 2020). As done by Malambo & Popescu (2024), segments with a multiple scattering warning flag greater than zero or a cloud confidence flag greater than one were removed to mitigate possible atmospheric interference.

After these preliminary filters on data quality, ATL08 segments were geolocated with the GeoPandas library from the *latitude* and *longitude* attributes, identifying center coordinates of signal photons for each segment (Jordahl et al., 2020; Neuenschwander et al., 2023). These points were intersected with annual NLCD land cover to determine the land cover of each segment’s centroid for the year of 2023. Using the most recent year of NLCD available ensures that land cover classification reflects the latest conditions within the ICESat-2 mission lifetime. Forested segments were identified by selecting only NLCD values of 41 (Deciduous Forest), 42 (Evergreen Forest), 43 (Mixed Forest), or 90 (Woody Wetlands). The centroids of these forest segments were then intersected with the annual LCMS Fast Loss change product from the year 2023 to provide insight into the disturbance history of these segments. For segments with a disturbance event reported by the Fast Loss product, the years since the last disturbance was calculated as the difference between the year of ICESat-2 collection and the most recent disturbance year. For this research, segments with disturbances occurring *after* the year of ICESat-2 collection were removed, as ATL08 canopy height values may no longer reflect the forest conditions present. Additionally, segments with less than 3 years since the last disturbance event were removed to provide at least 3 years of forest regrowth and recovery.

A map of the state of maine

AI-generated content may be incorrect.

*Figure 2: ATL08 segments collected in study area after data quality filters. Segments colored by year of collection to demonstrate ICESat-2 sampling pattern.*

* + 1. **Extracting ALS Data Within ATL08 Segments**

For this research it is necessary to extract 2020 ALS data within each individual ATL08 segment, regardless of the segment’s collection year. To generate the rectangular segments of the ATL08 product, a polygon of 100 x 11 meters was created around the centroid coordinates of each segment and rotated to align with the ICESat-2 track inclination using a custom python script (see Appendix 7.4; Rai et al., 2024). These polygons were then reprojected to the Web Mercator projection system, which the ALS data uses in its native storage format (Hobu, Inc., 2025). These polygons serve as cropping geometries for the ALS data.

ALS data was processed in a custom pipeline (Figure 3) using the *PDAL* library (Butler et al., 2024; see Appendix 7.1 for code). First, ALS data was streamed from any 3DEP project intersecting a given polygon using the polygon’s coordinate bounds. The Simple Morphological Filter (SMRF) was used to interpolate the ground surface as it minimized Type I errors in ground classification, which could’ve resulted in inaccurate canopy height estimation (Pingel et al., 2013). Heights above the interpolated ground surface were then calculated using a nearest neighbor approach. This normalized point cloud was cropped to the geometry of the respective polygon to match the polygon’s along-track rotation (Figure 4). A diagram of a diagram

AI-generated content may be incorrect.

*Figure 3: Workflow of python script used to extract ALS data collected within ATL08 polygons.*

A green field with blue lines

AI-generated content may be incorrect.

*Figure 4: ALS data extracted within ATL08 segments over a plantation area. ALS data contains only non-ground returns with reasonable vegetation height as described below.*

For each ATL08 segment, the normalized ALS point cloud was converted to a Pandas DataFrame to calculate metrics for return heights above ground (The pandas development team, 2024). We selected only non-ground returns with heights above ground between 2 and the previously determined global maximum of 39.42 meters to match the filters applied to ATL08 data. These returns were deemed as *vegetation* returns, from which several metrics were calculated for the return heights including minimum, maximum, mean, median, and the following percentiles: 90, 95, 98, 99. Finally, the ratio of vegetation returns to total returns provided an approximation of vegetation density (Neuenschwander et al., 2020). These ALS metrics were then joined to the corresponding ATL08 segment using a unique identifier.

Due to the overlapping collection scheme of ALS 3DEP projects, many ATL08 segments intersected with multiple ALS datasets, resulting in duplicate segments with vegetation metrics calculated for more than one ALS project (Figure 5).

A screen shot of a computer screen

AI-generated content may be incorrect.

*Figure 5: Track of ATL08 segments (white polygons) passing through overlapping area (green) of separate ALS projects (light blue, orange).*

In the instance duplicate segments, only the segment associated with the ALS project providing the highest number of vegetation returns was retained. Additionally, the original 3DEP ALS project boundaries are generalized. The initial crop of ATL08 segments to the ALS project boundaries resulted in several segments with zero ALS returns within their bounds. These were also removed from analyses, resulting in a final working set of53,905ATL08 segments. In summary, our working set contains 6 years of ICESat-2 ATL08 segments (2019-2024), from which vegetation height metrics were also calculated from 2020 airborne laser scanning collected within each segment’s geometry.

* + 1. **Retrieving Study Area FIA Data**

A copy of the North Carolina Forest Inventory and Analysis database (FIADB) was created in PostgreSQL using a custom repository that provides scripts for manipulating FIA data in a local environment (radt0005, 2025/2025). North Carolina FIA remeasurement plots were accessed using a custom R script (see Appendix 7.5 for code). This script selected only those in the evaluation group 372023, corresponding to plots in North Carolina (state code 37) remeasured during 2023. Using data from the most recent FIA remeasurement interval of 2023 provides the most up-to-date insight into the expected patterns of forest growth within the study area. These plots were further subset to only those from the 40 counties within the study area using county code attributes (Figure 6).

A map of the united states

AI-generated content may be incorrect.

*Figure 6: Counties of North Carolina within study area.*

Unique plot sequence numbers were extracted from FIA plots within the study area counties. For these sequence numbers, the plot, condition, and tree tables were extracted from the North Carolina FIADB. These tables were then joined into a single R dataframe to provide a detailed table of *current* measurements.

To obtain previous FIA measurements, the current record of a remeasurement plot was matched to the record of the previous inventory using the previous plot sequence number. For these previous plot sequence numbers, the same plot, condition, and tree data records were gathered from the FIADB. These tables were also joined into a single R dataframe to provide a detailed table of *historical* measurements.

Finally, the current and historical measurements were joined by matching state, survey unit, county, plot, subplot and species-specific individual tree records. This provided a single R dataframe of individual tree growth data on remeasured FIA plots to establish baseline rates of canopy height growth.

* 1. **Statistical Methods Used**

General descriptions for the statistical methods used in this research are provided below. For further information, readers should refer to the references cited herein. Unless otherwise stated, all statistical tests used a significance (alpha) level of 0.05. Analyses were conducted in R Statistical Software (v4.3.2; R Core Team, 2023) and JMP (*JMP Student Edition*, 1989).

* + 1. **Linear Regression Model**

Linear regression is a statistical method that aims to represent the relationship between a dependent and explanatory variable by fitting a linear equation to the observations (esri, 2025). Linear regression employs an equation following the form:

Where *Y* is the dependent variable, *X* is the explanatory variable, *b* is the slope of the line, and *a* is the intercept (*Y* when *x* = 0). Linear regression is often done using a “least squares” approach, where the fitted line attempts to minimize the sum of the squared residuals (Department of Statistics, 1997). However, least squares approaches can be highly sensitive to outliers (C. Yu & Yao, 2017), a non-trivial issue in the context of canopy height estimation using LiDAR which may generate errant returns. The least squares estimation of a regression slope is ultimately a *weighted average* of pairwise slopes, and an extreme slope value between a data point and an outlier will influence the slope estimate (Goldstein-Greenwood, 2023). As such, this research will employ a method of “robust regression”, the Theil-Senregression (Sen, 1968; Theil, 1992). The Theil-Sen regression calculates the slope between a predictor and response variable as the *median* of slopes between each pair of points in the data set, providing a correlation estimate that is more insensitive to outliers.

* + 1. **Equivalence Test**

An insignificant p-value of a null-hypothesis significance test (NHST) simply indicates the absence of evidence of a difference between treatments, which is not the same as stating equivalence between treatments (Altman & Bland, 1995). Failure in rejecting the null hypothesis of an NHST does not automatically make it true, merely that there is insufficient evidence to support the alternate (Wachs, 2015). Put simply, absence of evidence is not evidence of absence.

An equivalence test is a subtype of interval hypothesis testing, which tests against the null hypothesis that differences between group averages are larger than a margin of equivalence (Shtaynberger & Bar, 2023). Equivalence tests are used to show that group averages are equivalent within a margin that is *practically* important, as it is never truly possible to show that an effect size (Δ) is zero (Lakens, 2022). As such, when sharing results of an equivalence test it is common practice to report the equivalence margin the data is tested against.

This research will employ a two one-sided tests (TOST) procedure for equivalence testing. In TOST, the smallest effect size of interest is specified with an upper (ΔU) and lower (-ΔL) bound of equivalence. TOST utilize two simultaneous null hypotheses (Lakens, 2017), following the formulas:

With this setup, we can reject the presence of meaningful effects only if *both* tests yield p-values below 0.05 (Lakens, 2022), and consider the groups practically equivalent within the margin provided (JMP Support, 2024).

* + 1. **Wilcoxon Rank-Sum Test**

A t-test is a parametric test that evaluates the means of one or two populations. A one sample t-test evaluates if a population’s mean differs from a known value, a two-sample t-test evaluates if two populations differ significantly from each other, and a paired t-test evaluates significant difference of paired measurements (JMP Statistical Discovery, 2025). Given that this research aims to identify growth in forest height between two points in time with sampled measurements, a two-sample t-test is appropriate. However, this research will employ a non-parametric alternative to the two-sample t-test, the Wilcoxon Rank Sum test, which tests purely on the order (ranks) of observations from the two samples. The Wilcoxon statistic, W, is a sum of the ranks from one of the samples (Wild & Seber, 2000).

* + 1. **Mann-Kendall Test**

The Mann-Kendall test is a non-parametric test that assesses whether a time series has a monotonic upward or downward trend, without needing data to be normally distributed or linear (Kendall & Gibbons, 1990; Mann, 1945; Meals et al., 2011). The test itself is rank-based, unaffected by presence of extreme values in the dataset (Ringard et al., 2019). This makes it especially useful for time series of biological, chemical, or financial data, as it can identify whether trends of perceived growth or decline are statistically significant despite fluctuations. The test statistic S is calculated with the equation:

This equation determines the sign of the difference between each observation (*yj*) and the previous observation (*yi*), returing a -1, 0, or 1, if the differenceis negative, zero, or positive, respectively (EarthSoft, Inc., 2024). This S value provides the number of increasing occurrences in the dataset, with larger, positive S values indicative of an upward trend over time. The Z value provides a more practical look at the strength of the trend, calculated with:

where S – 1 is used if S > 0, or S – 1 is used if S < 0. Larger and more positive Z values provide greater confidence in rejecting the null hypothesis of no trend. Finally, Kendall’s Tau coefficient indicates the correlation of observations and their order in time (S. Chen et al., 2022), with values ranging from -1 (negative correlation) to 1 (positive correlation).

* + 1. **Bonferroni Correction**

In a single hypothesis test with a significance level of 0.05, the probability of a Type I error (false positive) is 5%. However, when running multiple significance tests (a family) the Type I error rate increases considerably. In our case, we are running 15 pairwise equivalence tests at each equivalence bound, increasing the family-wise error rate to 54% (Frost, 2023). The Bonferroni correction controls the likelihood of false positives by dividing the significance level by the number of tests performed (Amplitude, Inc., 2025). This correction determines a new significance level for our multiple hypothesis tests to be evaluated against.

* 1. **Phase 1: Validating ATL08 Canopy Heights with ALS**

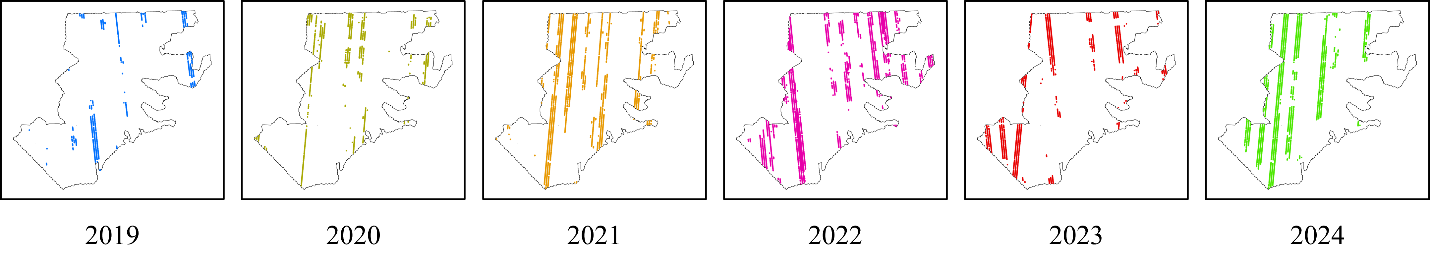
Prior to identifying growth over time, it is necessary to establish a strong relationship between ATL08 canopy heights and those from reference data at a single point in time. Doing so will allow us to reliably use ATL08 canopy height estimations throughout the remainder of this research, taking their reported estimations at face value. To accomplish this, Phase 1 uses linear regression to compare ATL08 canopy height estimations against ALS-derived canopy heights within the bounds of each ATL08 segment (see Section 2.3.2).

To align with the timing of ALS data, ATL08 segments in the study area were restricted to only those acquired during the summer of 2020, resulting in 5,026 segments. Though the ALS data was collected in the winter months of late 2019 to early 2020, this was deemed to provide an adequate temporal coincidence with the ATL08 data. These segments will serve as a *validation set* to assess the accuracy of the ATL08 product in canopy height estimation.

For the validation set of segments, the correlation between ATL08 and ALS canopy height metrics was assessed with a Theil-Sen regression model using the *RobustLinearReg* package in R (Hurtado, 2020; see Appedix 7.6 for code). For the most direct comparison, we compared the 98th percentile ATL08 canopy height (*h\_canopy)* variableto the 98th percentile height of ALS vegetation returns. The ATL08 height was used as the response (Y) variable, while ALS height was used as the predictor (X) variable.

* 1. **Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection**

Using ICESat-2 or GEDI data to track growth in forest height over time is challenged by the transect sampling patterns of spaceborne LiDAR (see Figure 2; Mulverhill et al., 2022). ICESat-2 intentionally employs an off-nadir pointing at mid-latitudes to fill gaps between tracks and obtain dense coverage of terrestrial vegetation data, prohibiting repeat measurement of individual forest stands (Markus et al., 2017; Neuenschwander et al., 2023; Neumann et al., 2019b). As a result, each year of ATL08 segment data is a distinct sample of the study area (Figure 7). With this in mind, we refer to each year of ATL08 segments as ‘samples’ for this phase of the approach.



*Figure 7: Separate plots showing spatial distribution of each year’s ATL08 segments across the study area, unique colors chosen for each year of segments. Each year of ATL08 segments do not align, but shift spatially to provide greater surface coverage.*

Since spatial differences in forest height may be misinterpreted as change over time, we must address two possible sources of biases. First, we verify that each annual sample represents the overall distribution of forests throughout the entire study area. However, this will only provide confidence in any individual sample. To use the samples together to track growth, we also verify that the footprints of each sample are equivalent with each other. By adequately addressing the bias that may exist within ICESat-2 data collection, we can move across space and throughout time to identify canopy height growth, with confidence that trends emerge *even though* the samples are distinct, and not *because* the samples are distinct.

* + 1. **Sample Representation**

To generate the population of forests within our study area, we use NLCD land cover data to randomly select 50,000 pixels from all forest cover types in a custom Python script (Figure 8; see Appendix 7.7 for code).

A screenshot of a computer generated map

AI-generated content may be incorrect.

*Figure 8: Layout showing a) all forested pixels throughout the study area, colored by forest cover type, and b) a random selection of 50,000 forest pixels. Selected pixels have an ATL08 segment created at their center coordinates, resulting in the red speckled effect.*

For each selected pixel, we create an ATL08 segment at its center and extract ALS canopy heights within the segment (see Appendix 7.8 for code). These combined extractions comprise the population of forests across the entire study area.

To establish individual samples as representative, we compare the ALS canopy heights within the footprints of each sample against the population (see Appendix 7.9 for code). If each sample’s distribution of ALS canopy heights falls within the overall population distribution, we can proceed with confidence that ICESat-2 collection yields representative samples of our study area.

* + 1. **Sample Location Equivalence**

Next, we establish the samples as equivalent by comparing the distribution of ALS canopy heights in their footprints against *each other,* rather than against population. For this, we perform equivalence testing with a bootstrapped TOST technique from the *TOSTER* package in R (Caldwell, 2022; Lakens, 2017), which was robust to possible violations of the standard t-test (Caldwell, 2025; see Appendix 7.9 for code).

In this case, the equivalence margin of our TOST corresponds to tolerable difference (in meters) in canopy height between the footprints of samples. These tests were performed at equivalence bound values of 2, 1.5, 1, 0.5, 0.25, and 0.1 meters, spanning a range of interpretations for “equivalence” in canopy height. The equivalence tests in this phase employ a Bonferroni-corrected significance level of α = *0.05 / 15* or *0.0033*.

Ultimately, these tests aim not to demonstrate perfect equivalence among samples, but rather to identify samples with noticeable divergence from the rest. Samples with poor performance in equivalence testing are removed temporal analyses, under the assumption of spatial differences in forest conditions that would confound growth over time.

* 1. **Phase 3: FIA-Derived Tree Height Growth**

Phase 3 determines expected tree height growth within the study area using FIA remeasurement plots. For this, we use *actual* height measurements from FIA plots rather than *total* height measurements. Actual height measurements use the canopy surface sampled by LiDAR data, while total height measurements may involve subjective estimation for trees with missing tops (U.S. Department of Agriculture Forest Service, 2024). Moreover, we use measurements only from live trees by removing dead trees using the FIA tree status code.

Importantly, FIA measurements are collected at the individual-tree level. For this phase we aggregate tree up to the plot-level to align with the spatial scale of ATL08 segments. From each FIA plot, we calculate the median values of annual increment and net growth in tree height.

Increment represents the annual, year-over-year growth for individual trees calculated as:

Net growth scales this annual increment to match the timeframe of our ATL08 segments, as FIA plots have remeasurements periods ranging from 5-10 years (U.S. Department of Agriculture Forest Service, 2023). It is calculated as:

Records with negative net growth were removed, possibly resulting from errors in the FIA sampling protocol or damage from disturbance events. Distributions of increment and net growth are summarized into minimum, 1st quartile, mean, median, 3rd quartile, and maximum values, and visualized with histograms to provide insight into typical annual and cumulative forest growth patterns in the study area (see Appendix 7.5). These metrics from FIA remeasurement plots serve not as targets to hit when tracking growth with ATL08 segments, but rather a gut-check for additional context.

* 1. **Phase 4: Assessing Canopy Height Growth with ATL08**

Phase 4 uses multiple statistical approaches to identify canopy height growth in ATL08 segment data. Growth trends are stratified across the following three factors:

1. **Forest cover type**: Deciduous Forest, Evergreen Forest, Mixed Forest, Woody Wetlands
2. **Disturbance presence:** Yes, No
3. **Time since disturbance:** <10 Years, 10-20 Years, 20-30 Years, 30+ Years

First, we use a Wilcoxon Rank Sum test to determine net growth in canopy height comparing the first and last years of ATL08 segments. This one-sided test uses an alternative hypothesis that canopy heights in the last year are significantly *greater* than those in the first year, indicative of growth over time. For the Wilcoxon tests we report the W statistic and the difference in mean canopy height between the first and last year of segment data.

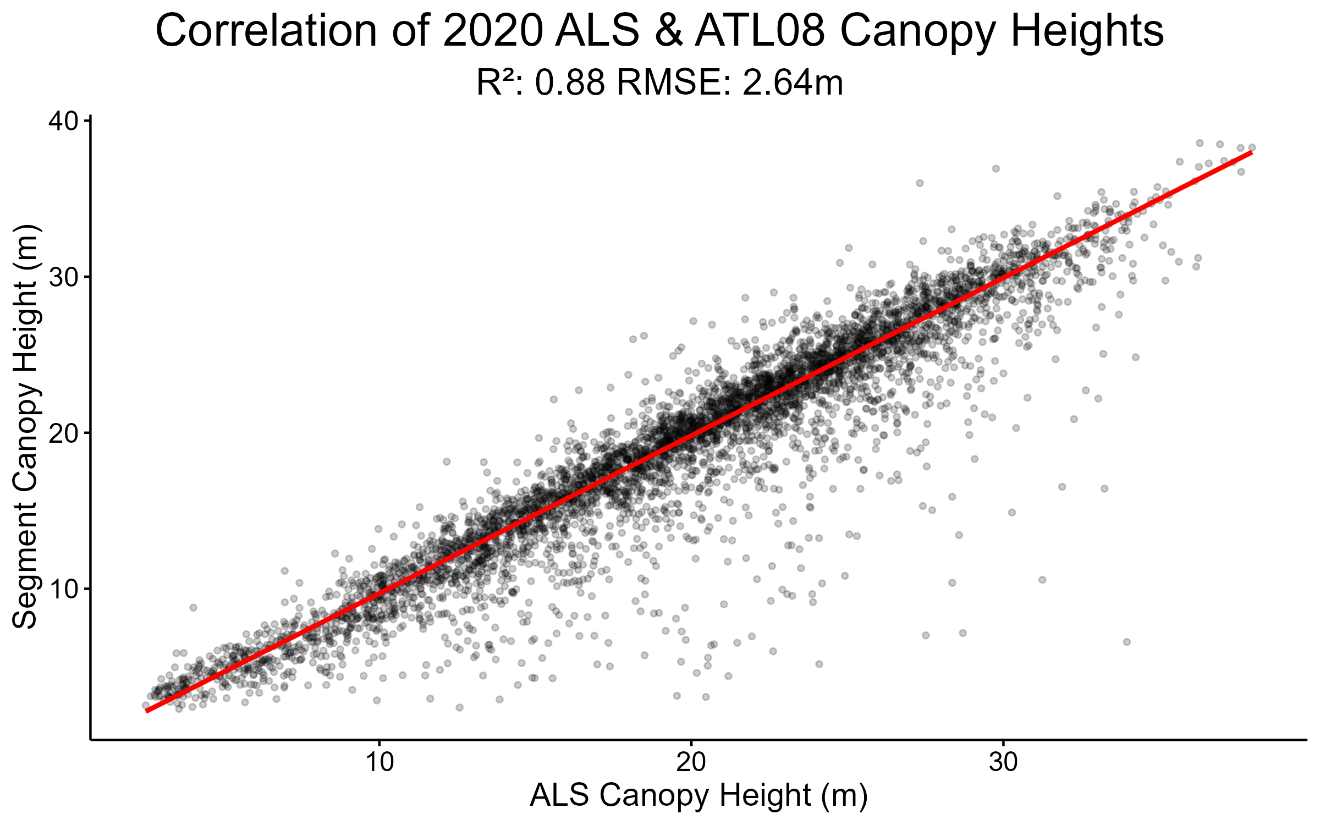
Next, we employ a Theil-Sen regression to estimate incremental growth in canopy height over time modeling the response of ATL08 canopy heights (Y) to the year of collection (X). From this model we report the slope, approximating the rate of growth from year-to-year, and the intercept, representing the baseline canopy height at the start of the segment timeframe.

Finally, we use a one-sided Mann-Kendall (MK) test was used to identify monotonic growth in canopy height over time. The MK test is also one-sided, using an alternative hypothesis that later values are significantly *greater* than earlier values, suggesting growth over time. From the MK test results we report the Z and Tau statistics.

Due to the large sample size in our ATL08 segment dataset, all statistical tests in Phase 4 were run with a bootstrapping technique (see Appendix 7.10 for code). Each test was run for 10,000 iterations on an independent sample of 500 observations taken with replacement. This accommodates the unequal sample sizes between years of ATL08 data, and ensures that any ATL08 segment is an independent pull from the dataset. Additionally, using 10,000 iterations stabilized the variability inherent in the bootstrapping technique. Statistics reported in Phase 4 are the median values of the 10,000 iterations.

1. **Results**
   1. **Phase 1: Validating ATL08 Canopy Heights with ALS**

ATL08 segment canopy height estimations demonstrate strong correlation with reference ALS data at a single point in time (Figure 8). With an R2 of 0.88 and an RMSE of 2.64 meters, ATL08 segments are adequately estimating canopy heights within the study area. The greater dispersion of points below the regression (red) line is indicative of an overall underestimation of canopy height by ATL08 in comparison to ALS data.

**

*Figure 9: Correlation of canopy heights estimated by ALS (X) and ATL08 (Y), with each point in scatterplot corresponding to an individual ATL08 segment. Canopy heights are reported in meters above ground. Red line indicates trend line from a Theil-Sen regression.*

* 1. **Phase 2: Addressing Sampling Bias of ICESat-2 Data Collection**
     1. **Sample Representation**

When comparing ALS canopy heights in the footprints of each sample against the full population of ALS canopy heights across the study area, the samples appear representative of the broader forest population (Figure 10). Each sample’s canopy height distribution falls within the bounds of the overall population distribution, with good overlap between samples. This suggests that ICESat-2 samples sufficiently capture the range of canopy heights across the study area, despite the spatial shift between samples.

A graph of a sound wave

AI-generated content may be incorrect.

*Figure 10: Plot showing distributions of ALS canopy heights for each ICESat-2 sample year (colored lines), overlaid on full population distribution of ALS canopy heights across the study area (light gray histogram). Unique colors chosen for distributions of each sample.*

* + 1. **Sample Location Equivalence**

Bootstrapped TOST show that all sample footprints are practically equivalent within a margin of two meters in canopy height (Figure 13). Below this equivalence bound, not all samples demonstrate equivalence.

A group of blue and yellow squares

AI-generated content may be incorrect.

*Figure 11: Grid of plots showing results from equivalence tests between sample distributions of ALS canopy heights. Each subplot represents a unique equivalence bound (in meters). Individual cells indicate the result of a TOST conducted between Sample 1 (X) and Sample 2 (Y). Blue cells indicate practical equivalence within the bound specified in the subplot title. Yellow cells indicate non-equivalence, as at least one of the null hypotheses in the TOST was not rejected.*

At more strict equivalence margins such as 0.1 or 0.25 meters, none of the samples demonstrate equivalence in canopy height. Based on the results in Figure 11, the 2019 sample demonstrates the lowest equivalence with other samples. This suggests that although this sample may represent the overall population, it contains spatial differences in canopy height with the other samples which could bias trends over time.

With this in mind, we remove the 2019 sample from further analyses and all remaining samples (2020-2024) now demonstrate equivalence at a margin of 1 meter in canopy height. This minimizes the sampling bias present in our ATL08 segments, but slightly limits the timeframe for tracking canopy height growth to only 5 years of data.

* 1. **Phase 3: FIA-Derived Tree Height Growth**

Table 1 shows plot-level summary statistics for annual increment and net growth in tree height from FIA remeasurement plots.

*Table 1: Summary statistics for annual height increment and five-year net growth measurements from FIA remeasurement plots. Units reported as original FIA measurements were in feet.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Units** | **Min.** | **1st Qu.** | **Median** | **Mean** | **3rd Qu.** | **Max.** |
| Increment | Meters per Year | 0 | 0.20 | 0.34 | 0.41 | 0.53 | 1.76 |
| Net Growth | Meters | 0 | 1.03 | 1.68 | 2.07 | 2.63 | 8.82 |

Figure 12 shows the distribution of actual height increment from remeasurement plots. The median value of 0.34 meters per year indicates the expected plot-level tree height growth across the range of forest conditions in the study area.

A graph of blue bars

AI-generated content may be incorrect.

*Figure 12: Distribution of height increment values from FIA remeasurement plots. Red line indicates median value.*

Figure 13 shows the distribution of five-year net growth in tree height at the plot level. This distribution is centered at 1.68 meters with a strong right skew, indicating a modest growth for most plots with some experiencing up to 8 meters of growth in a five-year period.

A graph of a graph

AI-generated content may be incorrect.

*Figure 13: Distribution of five-year net growth in tree height from FIA remeasurement plots. Red line indicates median value.*

* 1. **Phase 4: Assessing Canopy Height Growth with ATL08**

Wilcoxon Rank Sum tests show **s**tatistically significant net growth in canopy height between 2020 and 2024 for all factor levels except forests with 20-30 years since disturbance (Figure 13). The W statistic-indicating summed ranks of canopy heights in the last year of data-is highest in Woody Wetlands (W = 15,119). W values are comparable between disturbed (W = 147,364) and undisturbed forests (W = 146,018). For forests with a disturbance history, W values are highest in the <10 years group (W = 154,953), and decrease moving away from disturbance in time.

The mean differences in canopy height between the last and first year of data follow a similar pattern. Of the forest cover types it is highest in Woody Wetlands, with a mean difference of 2.59 meters in five years. Mean differences are nearly equivalent in disturbed (2.15 m) and undisturbed forests (2.13 m). Mean difference decreases moving away from disturbance in time until the 30+ years group, which shows the largest mean difference of any factor level at 3.33 meters.

A set of colorful bars

AI-generated content may be incorrect.

*Figure 14: Bar graphs showing results of canopy height net growth over time from Wilcoxon Rank Sum tests, grouped by factor and level. Separate plots show a) the W statistic, and b) mean difference in canopy height between the first and last years of data. Bars with value labels indicate statistically significant trends (p < 0.05).*

Theil-Sen regression models reveal statistically significant incremental canopy height growth for only four of ten factor levels (Figure 14). The highest estimated growth rate occurred in Woody Wetlands (0.44 m/year), followed by Evergreen Forest (0.42 m/year), Mixed Forest (0.39 m/year), and finally Deciduous Forest (0.31 m/year).

Disturbed forests show a higher growth rate (0.49 m/year) compared to undisturbed forests (0.36 m/year). For forests with a disturbance history, growth rates did not increase in tandem with time since disturbance. The highest rate was observed in forests disturbed <10 years ago (0.64 m/year), followed by lower values in the 10-20 years (0.33 m/year) and 20-30 years (0.14 m/year) groups. Notably, the 30+ years group showed a rebound in growth with a rate of 0.55 m/year.

Trends for baseline canopy height differed from growth rates. Baseline canopy height is highest in Deciduous Forests (24.81 m), followed by Mixed Forest (24.17 m), Woody Wetlands (20.46 m), and is lowest in Evergreen Forests (19.91 m). Baseline canopy height was higher in undisturbed forests (23.58 m) than disturbed forests (18.71). For forests with a history of disturbance, baseline canopy height generally increased with time since disturbance.

A graph of different colored bars

AI-generated content may be incorrect.

*Figure 15: Bar graphs showing results of incremental growth in canopy height over time from Theil-Sen regressions, grouped by factor and level. Separate plots show a) the regression slope value, and b) regression intercept values. Bars with value labels indicate statistically significant trends (p < 0.05).*

Finally, MK tests show statistically significant monotonic growth in canopy height over time for six of ten factor levels (Figure 15). No significant trends were found Woody Wetlands, undisturbed forests, and forests disturbed 10-20 or 20-30 years ago. The Z value is highest in Mixed Forests (Z = 2.51) of all forest cover types, while Deciduous (Z = 1.87) and Evergreen Forests (1.83) yield comparable values.

The Z value is higher in disturbed than undisturbed forests, which showed no significant trend. For forests with a disturbance history, Z values were high in areas disturbed less than 10 years ago (Z = 2.62), and decreased with time since disturbance. However, forests disturbed 30+ years ago showed the strongest trend of all factor levels (Z = 3.51).

Trends for Kendall’s Tau statistic mirror those of the Z statistic. Tau values ranged from 0.05 for Woody Wetlands, indicating the lowest agreement between canopy height growth and time, to 0.08 in Mixed Forests. Disturbed forests have a slightly higher Tau (0.06) than undisturbed forests (0.05). Tau values are highest in forests with <10 or 30+ years since disturbance, compared to forests with more moderate time since disturbance.

A graph of different colored squares

AI-generated content may be incorrect.

*Figure 16: Bar graphs showing results of monotonic growth in canopy height over time from Mann-Kendall tests, grouped by factor and level. Separate plots show a) Z statistic, and b) Tau statistic. Bars with value labels indicate statistically significant trends (p < 0.05).*

1. **Discussion**

The ability of ICESat-2 to reliably detect growth in forest height throughout time is vital for broad scale forest monitoring, overcoming the temporal limitations of repeat ALS data acquisition and spatial limitations of field inventories. However, the performance of ATL08 canopy height estimation can vary across multitude of topographic, atmospheric, and ecological conditions. As such, any study utilizing ATL08 canopy height estimations must compare them against reference data sources.

We first demonstrated that ATL08 sufficiently estimates canopy heights within our study area (R2 = 0.88, RMSE = 2.64) using coincident high-resolution ALS data. The minor underestimation of canopy height against reference ALS is to be expected from the literature (Neuenschwander et al., 2020; Rai et al., 2024; Q. Yu et al., 2024). As posited when specifying the study area, the strong performance of height estimation by ATL08 was likely due to the consistent tree height and regular tree spacing found in forest plantations within our study area. From the results of Phase 1, ATL08 canopy height estimations in the study area at a single point in time were deemed adequate for identifying trends throughout time.

Next we assessed the sampling bias present ATL08 data by treating each year of ATL08 segments as an individual sample of the landscape. Comparing 2020 ALS in the footprints of each sample against the population of forests indicate that ATL08 segment locations provide a representative sample of forest conditions throughout the study area. Comparing the samples against each other, all samples demonstrate equivalence within a margin of 2 meters in canopy height, showing a considerable level of agreement across a study area of over 5 million hectares. When tightening the equivalence margin to 1.5 meters, the 2019 sample did not demonstrate equivalence with the samples from 2021, 2023, or 2024, indicating some spatial difference in canopy height between the locations of those samples. Still, equivalence shown in 75% of groups within a 1.5-meter margin speaks to the forest cover homogeneity across the study area. After removing the 2019 sample, all samples demonstrate equivalence to each other within an even tighter 1-meter margin. Ultimately, the equivalence of ALS groups demonstrated in Phase 2 was deemed sufficient for our purposes, especially considering the RMSE of 2.64 meters in Phase 1, as it is unlikely that ATL08 measurements would be heavily influenced by spatial biases in canopy height falling within this margin of error.

Phase 3 established reference tree growth within our study area from FIA remeasurement plots. The distribution of height increment (Figure 14) is heavily concentrated in values below 0.5 meters per year, which may reflect maturing forests across the study area, as forest height growth and productivity slows as stands mature (Bennett, 1963; Tang et al., 2014). For loblolly pines vastly present throughout the southeastern United States (and therefore the study area), crown closure may occur within 15-20 years of age (Baker & Langdon, 1990; Bradley & Kush, 2019), leading to a slowed growth rate. While time since disturbance is not a direct proxy for stand age, ATL08 segments were collected primarily in forests with 10-20 (n = 9,476) and 20-30 (n = 7,999) years since disturbance, followed by forests with <10 years (n = 7,451) and 30+ years (n = 3,671) years since disturbance.

Dead trees were removed during FIA data processing to focus solely on living-tree growth. One can argue for retaining all standing trees to more holistically represent the forest structure surveyed by LiDAR, as ATL08 segments do not report tree status in canopy height estimations. However, including dead trees in FIA remeasurement analyses would have lowered values of both increment and net growth. Dead trees were removed from FIA remeasurement plots to select the best data subset to demonstrate forest height growth, just as the heavy filtering of ATL08 data aimed to achieve.

Phase 4 used multiple statistical approaches to quantify net, incremental, and monotonic canopy height growth. In revisiting the research questions, the results of Phase 4 show that ATL08 segments can detect canopy height growth over a five-year period. The trends shown are specifically indicative of growth over time as the regression slope coefficients, Wilcoxon test mean differences, and Mann-Kendall Tau statistics are all positive. Net growth showed the most utility, with significant results present in nine of ten factor levels. Incremental and monotonic growth trends were likely subjected to the noise associated with ATL08 canopy height estimations, with significant results present in only four and six factor levels, respectively.

The role of forest cover type in determining ATL08-derived growth generally followed expectations. The highest rates of growth-indicated by regression slopes-were found in plantation-like conditions (Evergreen Forest and Woody Wetlands). Mixed Forests showed strong agreement with reference FIA data, with a canopy height growth rate of 0.39 m/year matching closely with the FIA-derived annual increment of 0.34 m/year. Similarly, the mean difference of 1.83 meters in Mixed Forests aligned with the FIA-derived five-year net growth of 1.68 meters. This consistency was expected, as Mixed Forests provide a reasonable parallel to the breadth of forest conditions surveyed by the FIA program. Limiting FIA remeasurement plots only to plantation-like conditions would have resulted in greater values for both increment and net growth, and closer alignment with ATL08-derived trends of Evergreen Forests or Woody Wetlands.

The role of disturbance history had a more nuanced impact on the ATL08-derived growth, as disturbance histories are likely convolved within specific forest cover types. Forests that experienced disturbance <10 years ago demonstrated significant growth across tests, with notably high values for the mean difference (2.95 m) and growth rate (0.64 m/year). This suggests that the ATL08 segments are picking up a spike in forest growth immediately following disturbance events, such as heavy thins on forest plantations (Dickens & Moorhead, 2015). However, forests with 30+ years since disturbance also demonstrated significant growth across tests, yielding the highest mean difference (3.33 m), Z statistic (Z = 3.51), and Tau value (Tau = 0.11) of all factor levels. This went against the expected patterns of forest growth and recovery from disturbance, and necessitates further research. These results could be an artefact of the LCMS data product used to derive years since disturbance, as the comparable Global Forest Change product only extends back to the year 2000 in identifying disturbances (Hansen et al., 2013). Similarly, the divisions we selected to aggregate years since disturbance may be too broad, with variability present in the 30+ years group yielding the trends observed. However, aggregating years since disturbance in more granular periods would reduce the sample size of ATL08 segments falling within each.

Ultimately, the disturbance history of forests will challenge future research estimating canopy height growth with ATL08 segments. Primarily, the sampling rate of the ATLAS instrument will lead to an amplified underestimation of canopy height in sparser stands (Figure 15), which have a lower probability of generating returns throughout the canopy profile. In comparison, a denser canopy with a more homogenous top surface (present in undisturbed forests, or plantations with ample recovery from disturbance) will yield a more consistent canopy height.

A screenshot of a video game

AI-generated content may be incorrect.

*Figure 17: Diagram showing canopy surface (black line) in a disturbed (left) and undisturbed (right) forest stand which have the same top-of-canopy height. Due to the canopy openness, photons returned to the ATLAS sensor (red dots) are more likely to underestimate true 98th percentile canopy height in a disturbed forest.*

It should be noted the undeniable role of the study area in influencing the results, which was specifically chosen for its gentle topographic relief and known presence of forest plantations. In mountainous regions of greater terrain variability or tropical regions of greater canopy complexity, the ATL08 algorithm will struggle to identify ground photons (J. Dong et al., 2021; Fernandez-Diaz et al., 2022), likely failing to yield sensible results for canopy height growth over time. Moreover, our study area is located within the loblolly pine range-a region of exceptional forest productivity (Ribas-Costa et al., 2024). This region undoubtedly provides canopy height growth greater than that to be expected in other U.S. forests, where growth patterns over five years may not reach a level detectable in ATL08 segment data. As such, we must note that researchers should strongly consider the dominant topography, forest conditions, and forest productivity of their study area if attempting to identify canopy height growth with ATL08 data.

Researchers must also consider the spatial scale of their study area. Ideally, a study area should be large enough to provide an ample set of ATL08 segments after data quality filters, but small enough to maintain continuity in forest conditions. In line with the idea of ecological consistency, the Environmental Protection Agency has outlined *Ecoregions* of the United States with similar geology, vegetation, climate, wildlife, etc. These Ecoregions are available at multiple scales of granularity, from the generalized Level I Ecoregions to the detailed Level IV Ecoregions. Figure 18 shows the proportion of ATL08 segments from each annual sample in the Level II Ecoregions found within our study area (see Appendix 7.11 for code).

A screenshot of a video game

AI-generated content may be incorrect.

*Figure 18: Stacked bar plot showing the proportion ATL08 segments from each annual sample across EPA Level II Ecoregions. Percent values are relative to each annual sample’s total number of segments.*

To effectively track growth over time, all annual samples should ideally share similar proportional coverage, regardless of whether the distribution between Ecoregions is 50:50, 75:25, 90:10, etc. In other words, the consistency in relative representation is more important than the balance between the Ecoregions themselves. The Phase 2 results showed that the 2019 sample had the lowest canopy height equivalence with other samples, which led to its removal from further analyses. As shown in Figure 18, the 2019 sample also demonstrates the lowest similarity in its distribution across Level II Ecoregions with the other samples. When considering spatial scales best suited for forest monitoring with ATL08 data, this presents EPA Ecoregions as a possible recommendation to maintain a tight margin of canopy height equivalence across segment footprints.

Testing equivalence between treatments requires careful definition of a margin of “practical” equivalence-a range within treatment results are close enough to be considered the same in practice. In this research, this margin was the difference in canopy height between the footprints of annual samples we would tolerate when tracking growth over time. Ultimately, trends emerging from analyses of multitemporal ATL08 data will always be influenced by *some* spatial differences in forest height, purely due to the nature of ICESat-2’s sampling pattern. In absence of a widely accepted standard, researchers may derive their own definition of canopy height equivalence to balance the duration over which they’re assessing growth, the amount and spatial coverage of ATL08 data available, and the noise of ATL08 canopy height estimates within their study area. Since canopy height is emergent from multiple ecological factors and canopy height observations are limited by the precision of the measurement instrument, we found it appropriate to test equivalence across a range of equivalence bound values.

Other studies have used a flexible percentage-based approach for paired measurements of tree height, deeming a region within +25% of the slope and intercept means as practically equivalent (Corrao et al., 2022; Falkowski et al., 2008). While this research does not employ paired measurements for equivalence testing, a percentage-based approach could accommodate the range of forest cover types sampled by ICESat-2’s global coverage. A recent study by Guo et al. (2025) found equivalence for tree height measurements between instruments at a tolerance above 1.5m in trees with a “pyramidal and well-defined crown”-similar to that of pine plantations in our study area. While their study focused on measurement errors arising from field crew experience, an argument could be made that the numerous sources of error in ATL08 data present similar inconsistencies, despite our attempts to control them in this study.

1. **Conclusion**

Land management decisions at a local scale affect biodiversity and climatic processes, and contribute collectively to global climate change (Griffith et al., 2003). Understanding regional differences in environmental resource use and degradation is therefore critical to ensure effective land management that is locale-specific (Omernik & Griffith, 2014). In consideration of spatial biases in canopy height presented in the discussion, future research could stratify ATL08-derived trends by EPA Ecoregions.

The results of Phase 4 show that ATL08 segments have the potential to identify forest height growth throughout time. However, this ability will be reliant upon several factors, not only those influencing ATL08 canopy height estimations at a single point in time (Malambo & Popescu, 2021; Rai et al., 2024), but their amplified effects throughout time. This study applied several quality filters to ATL08 data that were non-trivial in their reduction of the sample size, controlling for beam strength, time of collection, and seasonality to select the best subset of ATL08 data. While strong beam, nighttime data collected in leaf on conditions have historically provided the highest accuracy in static canopy height estimation, future studies should maintain all ATL08 segments within a study area to test the emergence of temporal trends *across* factors of ATL08 data and environmental conditions. This would extend the work of Rai et. al (2024), A. Liu et al. (2021), Malambo & Popescu (2021), Neuenschwander et al. (2020), and others across multiple timesteps.

Spatial biases may have arisen in Phase 2 from post-Hurricane Florence recovery of the forests in our study area, as the occurrence of Hurricane Florence (September 2018) aligns closely with the beginning of ICESat-2 data collection (December 2018). Further work is required to investigate the spatial stratification of the segments using a Hurricane Florence flood extent product (NC OneMap, 2020), which could determine if Hurricane Florence inundation is impacting the canopy height values reported by the 2020 ALS.

Ultimately, this research aims to understand structural changes of vegetation throughout time using modern active remote sensing missions. In understanding these dynamics at the regional scale, this research can aid governing agencies and local stakeholders with spatially explicit management strategies. Such strategies can protect vegetation and natural resources against site-specific threats, supporting conservation efforts and land-use planning in the face of a changing climate.

1. **Bibliography**

Altman, D. G., & Bland, J. M. (1995). Statistics notes: Absence of evidence is not evidence of absence. *BMJ*, *311*(7003), 485–485. https://doi.org/10.1136/bmj.311.7003.485

Amplitude, Inc. (2025). *Bonferroni Correction Explained: Managing Multiple Testing in Statistics*. https://amplitude.com/explore/experiment/what-is-bonferroni-correction

Andersen, H.-E., Reutebuch, S. E., & McGaughey, R. J. (2006). A rigorous assessment of tree height measurements obtained using airborne lidar and conventional field methods. *Canadian Journal of Remote Sensing*, *32*(5), 355–366. https://doi.org/10.5589/m06-030

Atkins, J. W., Costanza, J., Dahlin, K. M., Dannenberg, M. P., Elmore, A. J., Fitzpatrick, M. C., Hakkenberg, C. R., Hardiman, B. S., Kamoske, A., LaRue, E. A., Silva, C. A., Stovall, A. E. L., & Tielens, E. K. (2023). Scale dependency of lidar‐derived forest structural diversity. *Methods in Ecology and Evolution*, *14*(2), 708–723. https://doi.org/10.1111/2041-210X.14040

Baker, J. B., & Langdon, G. O. (1990). Pinus Taeda L. Loblolly Pine. In *Silvics of North America* (Vol. 1, pp. 497–512). U.S. Dept. of Agriculture, Forest Service.

Beland, M., Parker, G., Sparrow, B., Harding, D., Chasmer, L., Phinn, S., Antonarakis, A., & Strahler, A. (2019). On promoting the use of lidar systems in forest ecosystem research. *Forest Ecology and Management*, *450*, 117484. https://doi.org/10.1016/j.foreco.2019.117484

Bennett, F. (1963). *Gorwth and Yield of Slash Pine Plantations* (No. SE-1). U.S. Department of Agriculture, Forest Service Southern Forest Experiment Station.

Bohn, F. J., & Huth, A. (2017). The importance of forest structure to biodiversity–productivity relationships. *Royal Society Open Science*, *4*(1), 160521. https://doi.org/10.1098/rsos.160521

Bradley, S., & Kush, J. (2019, January 30). Thinning Loblolly Pine Stands To Benefit Wildlife & Timber Production. *Alabama Cooperative Extension System*. https://www.aces.edu/blog/topics/forestry/thinning-loblolly-pine-stands-to-benefit-wildlife-timber-production/

Butler, H., Andrew Bell, Michael P. Gerlek, chambbj, Pete Gadomski, Connor Manning, Mateusz Łoskot, Bas Couwenberg, Norman Barker, Paul Ramsey, Julia Dark, Nicolas Chaulet, Even Rouault, Kyle Mann, Claire Foster, Guilhem Villemin, Michael Rosen, Grigory, Ogi Moore, … Aaron Yonas. (2024). *PDAL/PDAL: 2.7.1* (Version 2.7.1) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.10884408

Caldwell, A. R. (2022). *Exploring Equivalence Testing with the Updated TOSTER R Package*. PsyArXiv. https://doi.org/10.31234/osf.io/ty8de

Caldwell, A. R. (2025). *Robust TOST Procedures* [Vigenette]. https://cran.r-project.org/web/packages/TOSTER/vignettes/robustTOST.html

Callaghan, J. (2020). Extreme rainfall and flooding from Hurricane Florence. *Tropical Cyclone Research and Review*, *9*(3), 172–177. https://doi.org/10.1016/j.tcrr.2020.07.002

Cao, L., Coops, N. C., Innes, J. L., Sheppard, S. R. J., Fu, L., Ruan, H., & She, G. (2016). Estimation of forest biomass dynamics in subtropical forests using multi-temporal airborne LiDAR data. *Remote Sensing of Environment*, *178*, 158–171. https://doi.org/10.1016/j.rse.2016.03.012

Carabajal, C. C., & Boy, J.-P. (2020). ICESAT-2 ALTIMETRY AS GEODETIC CONTROL. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B3-2020*, 1299–1306. https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1299-2020

Chastain, R., Housman, I., Goldstein, J., Finco, M., & Tenneson, K. (2019). Empirical cross sensor comparison of Sentinel-2A and 2B MSI, Landsat-8 OLI, and Landsat-7 ETM+ top of atmosphere spectral characteristics over the conterminous United States. *Remote Sensing of Environment*, *221*, 274–285. https://doi.org/10.1016/j.rse.2018.11.012

Chen, F., Zhang, X., Wang, L., Du, B., Dang, S., & Wang, L. (2023). Systematic Evaluation of Multi-Resolution ICESat-2 Canopy Height Data: A Case Study of the Taranaki Region. *Remote Sensing*, *15*(24), 5686. https://doi.org/10.3390/rs15245686

Chen, S., Ghadami, A., & Epureanu, B. I. (2022). Practical guide to using Kendall’s *τ* in the context of forecasting critical transitions. *Royal Society Open Science*, *9*(7), 211346. https://doi.org/10.1098/rsos.211346

Columbia Environmental Research Center. (2019). *2019 Disaster Relief Act: USGS recovery activities* (USGS Numbered Series Nos. 2019–3066; Fact Sheet, p. 4). U.S. Geological Survey.

Coops, N. C., Tompalski, P., Goodbody, T. R. H., Queinnec, M., Luther, J. E., Bolton, D. K., White, J. C., Wulder, M. A., Van Lier, O. R., & Hermosilla, T. (2021). Modelling lidar-derived estimates of forest attributes over space and time: A review of approaches and future trends. *Remote Sensing of Environment*, *260*, 112477. https://doi.org/10.1016/j.rse.2021.112477

Corrao, M. V., Sparks, A. M., & Smith, A. M. S. (2022). A Conventional Cruise and Felled-Tree Validation of Individual Tree Diameter, Height and Volume Derived from Airborne Laser Scanning Data of a Loblolly Pine (P. taeda) Stand in Eastern Texas. *Remote Sensing*, *14*(11), 2567. https://doi.org/10.3390/rs14112567

Department of Statistics. (1997). *Linear Regression* (No. 101; Course List for 1997-98). Yale University. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

Dickens, E. D., & Moorhead, D. J. (2015). A Guide to Thinning Pine Plantations. *Georgia Forest Productivity*, *0010*. https://bugwoodcloud.org/bugwood/productivity/pdfs/Guide\_thinning\_pine\_plantations.pdf

Dong, J., Ni, W., Zhang, Z., Sun, G., 中国科学院空天信息创新研究院 遥感科学国家重点实验室, 北京 100094 Aerospace Information Research Institute, Chinese Academy of Sciences, State Key Laboratory of Remote Sensing Science, Beijing 100094, China, 中国科学院大学 资源与环境学院, 北京 100049 School of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China, & 美国马里兰大学地理系 马里兰州College Park, 美国 20742 Department of Geographical Sciences, University of Maryland, College Park, Maryland, USA, 20742. (2021). Performance of ICESat-2 ATL08 product on the estimation of forest height by referencing to small footprint LiDAR data. *National Remote Sensing Bulletin*, *25*(6), 1294–1307. https://doi.org/10.11834/jrs.20219449

Dong, P., & Chen, Q. (2017). *LiDAR Remote Sensing and Applications* (1st ed.). CRC Press. https://doi.org/10.4324/9781351233354

Dubayah, R., Armston, J., Healey, S. P., Bruening, J. M., Patterson, P. L., Kellner, J. R., Duncanson, L., Saarela, S., Ståhl, G., Yang, Z., Tang, H., Blair, J. B., Fatoyinbo, L., Goetz, S., Hancock, S., Hansen, M., Hofton, M., Hurtt, G., & Luthcke, S. (2022). GEDI launches a new era of biomass inference from space. *Environmental Research Letters*, *17*(9), 095001. https://doi.org/10.1088/1748-9326/ac8694

Dubayah, R., Blair, J. B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P. L., Qi, W., & Silva, C. (2020). The Global Ecosystem Dynamics Investigation: High-resolution laser ranging of the Earth’s forests and topography. *Science of Remote Sensing*, *1*, 100002. https://doi.org/10.1016/j.srs.2020.100002

Duncanson, L., Neuenschwander, A., Hancock, S., Thomas, N., Fatoyinbo, T., Simard, M., Silva, C. A., Armston, J., Luthcke, S. B., Hofton, M., Kellner, J. R., & Dubayah, R. (2020). Biomass estimation from simulated GEDI, ICESat-2 and NISAR across environmental gradients in Sonoma County, California. *Remote Sensing of Environment*, *242*, 111779. https://doi.org/10.1016/j.rse.2020.111779

EarthSoft, Inc. (2024, July 8). *Mann-Kendall Tests*. EQuIS 7.24.3. https://help.earthsoft.com/lib\_mann-kendall.htm

esri. (2025, April 18). How Linear regression algorithm works. *Feature and Tabular Analysis Toolset Concepts*. https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-linear-regression-works.htm

Falkowski, M. J., Smith, A. M. S., Gessler, P. E., Hudak, A. T., Vierling, L. A., & Evans, J. S. (2008). The influence of conifer forest canopy cover on the accuracy of two individual tree measurement algorithms using lidar data. *Canadian Journal of Remote Sensing*, *34*(sup2), S338–S350. https://doi.org/10.5589/m08-055

FAO and UNEP. (2020). *The State of the World’s Forests 2020*. FAO and UNEP. https://doi.org/10.4060/ca8642en

Feng, T., Duncanson, L., Montesano, P., Hancock, S., Minor, D., Guenther, E., & Neuenschwander, A. (2023). A systematic evaluation of multi-resolution ICESat-2 ATL08 terrain and canopy heights in boreal forests. *Remote Sensing of Environment*, *291*, 113570. https://doi.org/10.1016/j.rse.2023.113570

Fernandez-Diaz, J. C., Velikova, M., & Glennie, C. L. (2022). Validation of ICESat-2 ATL08 Terrain and Canopy Height Retrievals in Tropical Mesoamerican Forests. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 2956–2970. https://doi.org/10.1109/JSTARS.2022.3163208

Foga, S., Scaramuzza, P. L., Guo, S., Zhu, Z., Dilley, R. D., Beckmann, T., Schmidt, G. L., Dwyer, J. L., Joseph Hughes, M., & Laue, B. (2017). Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sensing of Environment*, *194*, 379–390. https://doi.org/10.1016/j.rse.2017.03.026

Frost, J. (2023, July 17). What is the Bonferroni Correction and How to Use It. *Hypothesis Testing*. https://statisticsbyjim.com/hypothesis-testing/bonferroni-correction/

Goldstein-Greenwood, J. (2023, April 28). Theil-Sen Regression: Programming and Understanding an Outlier-Resistant Alternative to Least Squares. *UVA Library StatLab*. https://library.virginia.edu/data/articles/theil-sen-regression-programming-and-understanding-an-outlier-resistant-alternative-to-least-squares

Gong, P., Li, Z., Huang, H., Sun, G., & Wang, L. (2011). ICESat GLAS Data for Urban Environment Monitoring. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(3), 1158–1172. https://doi.org/10.1109/TGRS.2010.2070514

Griffin, M., Malsick, M., Mizzell, H., & Moore, L. (2020). Historic Rainfall and Record-Breaking Flooding from Hurricane Florence in the Pee Dee Watershed. *The Journal of South Carolina Water Resources*, *6*, 28–35. https://doi.org/10.34068/JSCWR.06.03

Griffith, J. A., Stehman, S. V., & Loveland, T. R. (2003). Landscape Trends in Mid-Atlantic and Southeastern United States Ecoregions. *Environmental Management*, *32*(5), 572–588. https://doi.org/10.1007/s00267-003-0078-2

Guerra-Hernández, J., Arellano-Pérez, S., González-Ferreiro, E., Pascual, A., Sandoval Altelarrea, V., Ruiz-González, A. D., & Álvarez-González, J. G. (2021). Developing a site index model for P. Pinaster stands in NW Spain by combining bi-temporal ALS data and environmental data. *Forest Ecology and Management*, *481*, 118690. https://doi.org/10.1016/j.foreco.2020.118690

Guerra-Hernández, J., Narine, L. L., Pascual, A., Gonzalez-Ferreiro, E., Botequim, B., Malambo, L., Neuenschwander, A., Popescu, S. C., & Godinho, S. (2022). Aboveground biomass mapping by integrating ICESat-2, SENTINEL-1, SENTINEL-2, ALOS2/PALSAR2, and topographic information in Mediterranean forests. *GIScience & Remote Sensing*, *59*(1), 1509–1533. https://doi.org/10.1080/15481603.2022.2115599

Hall, F. G., Bergen, K., Blair, J. B., Dubayah, R., Houghton, R., Hurtt, G., Kellndorfer, J., Lefsky, M., Ranson, J., Saatchi, S., Shugart, H. H., & Wickland, D. (2011). Characterizing 3D vegetation structure from space: Mission requirements. *Remote Sensing of Environment*, *115*(11), 2753–2775. https://doi.org/10.1016/j.rse.2011.01.024

Hancock, S., McGrath, C., Lowe, C., Davenport, I., & Woodhouse, I. (2021). Requirements for a global lidar system: Spaceborne lidar with wall-to-wall coverage. *Royal Society Open Science*, *8*(12), 211166. https://doi.org/10.1098/rsos.211166

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S. J., Loveland, T. R., Kommareddy, A., Egorov, A., Chini, L., Justice, C. O., & Townshend, J. R. G. (2013). High-Resolution Global Maps of 21st-Century Forest Cover Change. *Science*, *342*(6160), 850–853. https://doi.org/10.1126/science.1244693

Harrap, R., & Lato, M. (2010, June 1). *An Overview of LIDAR: collection to applications*. Norwegian Geotechnical Institute.

Harris, N., & Gibbs, D. (2021). *Forests Absorb Twice As Much Carbon As They Emit Each Year*. https://www.wri.org/insights/forests-absorb-twice-much-carbon-they-emit-each-year

Hobu, Inc. (2025). *USGS 3DEP LiDAR Point Clouds* [S3 Bucket; Entwine Point Tiles]. https://registry.opendata.aws/usgs-lidar

Homer, C., Dewitz, J., Jin, S., Xian, G., Costello, C., Danielson, P., Gass, L., Funk, M., Wickham, J., Stehman, S., Auch, R., & Riitters, K. (2020). Conterminous United States land cover change patterns 2001–2016 from the 2016 National Land Cover Database. *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*, 184–199. https://doi.org/10.1016/j.isprsjprs.2020.02.019

Housman, I., Campbell, L., Heyer, J., Goetz, W., Finco, M., Megown, K., & Pugh, N. (2021). US Forest Service landscape change monitoring system methods. *USDA General Technology and Applications Center, Salt Lake City, UT*.

Hui, G., Zhang, G., Zhao, Z., & Yang, A. (2019). Methods of Forest Structure Research: A Review. *Current Forestry Reports*, *5*(3), 142–154. https://doi.org/10.1007/s40725-019-00090-7

Hurtado, S. (2020). *RobustLinearReg: Robust Linear Regressions* (p. 1.2.0) [Dataset]. https://doi.org/10.32614/CRAN.package.RobustLinearReg

Jaboyedoff, M., Oppikofer, T., Abellán, A., Derron, M.-H., Loye, A., Metzger, R., & Pedrazzini, A. (2012). Use of LIDAR in landslide investigations: A review. *Natural Hazards*, *61*(1), 5–28. https://doi.org/10.1007/s11069-010-9634-2

Jarron, L. R., Coops, N. C., MacKenzie, W. H., Tompalski, P., & Dykstra, P. (2020). Detection of sub-canopy forest structure using airborne LiDAR. *Remote Sensing of Environment*, *244*, 111770. https://doi.org/10.1016/j.rse.2020.111770

JMP Statistical Discovery. (2025). The t-test. *Statistics Knowledge Portal*. https://www.jmp.com/en/statistics-knowledge-portal/t-test

*JMP Student Edition* (Version 18.2.0). (1989). [Computer software]. SAS Institute Inc.

JMP Support. (2024). *Equivalence Test Reports*. JMP Statistical Discovery LLC. https://www.jmp.com/support/help/en/18.1/index.shtml#page/jmp/equivalence-test-reports.shtml

Jordahl, K., Bossche, J. V. D., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., Maxalbert, Bilogur, A., … Leblanc, F. (2020). *geopandas/geopandas: V0.8.1* (Version v0.8.1) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3946761

Karjalainen, E., Sarjala, T., & Raitio, H. (2009). Promoting human health through forests: Overview and major challenges. *Environmental Health and Preventive Medicine*, *15*(1), 1–8. https://doi.org/10.1007/s12199-008-0069-2

Kendall, M. G., & Gibbons, J. D. (1990). *Rank correlation methods* (5th ed). E. Arnold ; Oxford University Press.

Kennedy, R. E., Yang, Z., & Cohen, W. B. (2010). Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr — Temporal segmentation algorithms. *Remote Sensing of Environment*, *114*(12), 2897–2910. https://doi.org/10.1016/j.rse.2010.07.008

Kennedy, R. E., Yang, Z., Gorelick, N., Braaten, J., Cavalcante, L., Cohen, W. B., & Healey, S. (2018). Implementation of the LandTrendr Algorithm on Google Earth Engine. *Remote Sensing*, *10*(5), 691. https://doi.org/10.3390/rs10050691

Khaine, I., & Woo, S. Y. (2015). An overview of interrelationship between climate change and forests. *Forest Science and Technology*, *11*(1), 11–18. https://doi.org/10.1080/21580103.2014.932718

Lakens, D. (2017). Equivalence Tests: A Practical Primer for *t* Tests, Correlations, and Meta-Analyses. *Social Psychological and Personality Science*, *8*(4), 355–362. https://doi.org/10.1177/1948550617697177

Lakens, D. (2022). *Improving Your Statistical Inferences*. https://doi.org/10.5281/ZENODO.6409077

Lamping, J. E., Zald, H. S. J., Madurapperuma, B. D., & Graham, J. (2021). Comparison of Low-Cost Commercial Unpiloted Digital Aerial Photogrammetry to Airborne Laser Scanning across Multiple Forest Types in California, USA. *Remote Sensing*, *13*(21), 4292. https://doi.org/10.3390/rs13214292

LaRue, E. A., Fahey, R., Fuson, T. L., Foster, J. R., Matthes, J. H., Krause, K., & Hardiman, B. S. (2022). Evaluating the sensitivity of forest structural diversity characterization to LiDAR point density. *Ecosphere*, *13*(9), e4209. https://doi.org/10.1002/ecs2.4209

Li, B., Zhao, T., Su, X., Fan, G., Zhang, W., Deng, Z., & Yu, Y. (2022). Correction of Terrain Effects on Forest Canopy Height Estimation Using ICESat-2 and High Spatial Resolution Images. *Remote Sensing*, *14*(18), 4453. https://doi.org/10.3390/rs14184453

Li, W., Niu, Z., Shang, R., Qin, Y., Wang, L., & Chen, H. (2020). High-resolution mapping of forest canopy height using machine learning by coupling ICESat-2 LiDAR with Sentinel-1, Sentinel-2 and Landsat-8 data. *International Journal of Applied Earth Observation and Geoinformation*, *92*, 102163. https://doi.org/10.1016/j.jag.2020.102163

Liu, A., Cheng, X., & Chen, Z. (2021). Performance evaluation of GEDI and ICESat-2 laser altimeter data for terrain and canopy height retrievals. *Remote Sensing of Environment*, *264*, 112571. https://doi.org/10.1016/j.rse.2021.112571

Liu, M., Popescu, S., & Malambo, L. (2019). Feasibility of Burned Area Mapping Based on ICESAT−2 Photon Counting Data. *Remote Sensing*, *12*(1), 24. https://doi.org/10.3390/rs12010024

Liu, X., Feng, Y., Hu, T., Luo, Y., Zhao, X., Wu, J., Maeda, E. E., Ju, W., Liu, L., Guo, Q., & Su, Y. (2024). Enhancing ecosystem productivity and stability with increasing canopy structural complexity in global forests. *Science Advances*, *10*(20), eadl1947. https://doi.org/10.1126/sciadv.adl1947

Luthcke, S. B., Thomas, T. C., Pennington, T. A., Rebold, T. W., Nicholas, J. B., Rowlands, D. D., Gardner, A. S., & Bae, S. (2021). ICESat‐2 Pointing Calibration and Geolocation Performance. *Earth and Space Science*, *8*(3), e2020EA001494. https://doi.org/10.1029/2020EA001494

Malambo, L., & Popescu, S. (2024). Mapping vegetation canopy height across the contiguous United States using ICESat-2 and ancillary datasets. *Remote Sensing of Environment*, *309*, 114226. https://doi.org/10.1016/j.rse.2024.114226

Malambo, L., & Popescu, S. C. (2021). Assessing the agreement of ICESat-2 terrain and canopy height with airborne lidar over US ecozones. *Remote Sensing of Environment*, *266*, 112711. https://doi.org/10.1016/j.rse.2021.112711

Maltamo, M., Næsset, E., & Vauhkonen, J. (Eds.). (2014). *Forestry Applications of Airborne Laser Scanning: Concepts and Case Studies* (Vol. 27). Springer Netherlands. https://doi.org/10.1007/978-94-017-8663-8

Mann, H. B. (1945). Nonparametric Tests Against Trend. *Econometrica*, *13*(3), 245. https://doi.org/10.2307/1907187

Markus, T., Neumann, T., Martino, A., Abdalati, W., Brunt, K., Csatho, B., Farrell, S., Fricker, H., Gardner, A., Harding, D., Jasinski, M., Kwok, R., Magruder, L., Lubin, D., Luthcke, S., Morison, J., Nelson, R., Neuenschwander, A., Palm, S., … Zwally, J. (2017). The Ice, Cloud, and land Elevation Satellite-2 (ICESat-2): Science requirements, concept, and implementation. *Remote Sensing of Environment*, *190*, 260–273. https://doi.org/10.1016/j.rse.2016.12.029

Meals, D., Spooner, J., Dressing, S., & Harcum, J. (2011). *Statistical Analysis for Monotonic Trends* (No. 6; Tech Notes, p. 23). U.S. Environmental Protection Agency. https://www.epa.gov/sites/default/files/2016-05/documents/tech\_notes\_6\_dec2013\_trend.pdf

Meyer, V., Saatchi, S. S., Chave, J., Dalling, J. W., Bohlman, S., Fricker, G. A., Robinson, C., Neumann, M., & Hubbell, S. (2013). Detecting tropical forest biomass dynamics from repeated airborne lidar measurements. *Biogeosciences*, *10*(8), 5421–5438. https://doi.org/10.5194/bg-10-5421-2013

Mielcarek, M., Kamińska, A., & Stereńczak, K. (2020). Digital Aerial Photogrammetry (DAP) and Airborne Laser Scanning (ALS) as Sources of Information about Tree Height: Comparisons of the Accuracy of Remote Sensing Methods for Tree Height Estimation. *Remote Sensing*, *12*(11), 1808. https://doi.org/10.3390/rs12111808

Mitchell, J. C., Kashian, D. M., Chen, X., Cousins, S., Flaspohler, D., Gruner, D. S., Johnson, J. S., Surasinghe, T. D., Zambrano, J., & Buma, B. (2023). Forest ecosystem properties emerge from interactions of structure and disturbance. *Frontiers in Ecology and the Environment*, *21*(1), 14–23. https://doi.org/10.1002/fee.2589

Mulverhill, C., Coops, N. C., Hermosilla, T., White, J. C., & Wulder, M. A. (2022). Evaluating ICESat-2 for monitoring, modeling, and update of large area forest canopy height products. *Remote Sensing of Environment*, *271*, 112919. https://doi.org/10.1016/j.rse.2022.112919

NC OneMap. (2020). *Hurricane Florence Flood Extent Across the Piedmont and Coastal Plain of North Carolina* [Imagery Dataset]. https://services.nconemap.gov/secure/rest/services/Florence\_Flood\_Extent/ImageServer

Neuenschwander, A., Duncanson, L., Montesano, P., Minor, D., Guenther, E., Hancock, S., Wulder, M. A., White, J. C., Purslow, M., Thomas, N., Mandel, A., Feng, T., Armston, J., Kellner, J. R., Andersen, H. E., Boschetti, L., Fekety, P., Hudak, A., Pisek, J., … Stereńczak, K. (2024). Towards global spaceborne lidar biomass: Developing and applying boreal forest biomass models for ICESat-2 laser altimetry data. *Science of Remote Sensing*, *10*, 100150. https://doi.org/10.1016/j.srs.2024.100150

Neuenschwander, A., Guenther, E., White, J. C., Duncanson, L., & Montesano, P. (2020). Validation of ICESat-2 terrain and canopy heights in boreal forests. *Remote Sensing of Environment*, *251*, 112110. https://doi.org/10.1016/j.rse.2020.112110

Neuenschwander, A., & Magruder, L. A. (2019). Canopy and Terrain Height Retrievals with ICESat-2: A First Look. *Remote Sensing*, *11*(14), 1721. https://doi.org/10.3390/rs11141721

Neuenschwander, A., & Pitts, K. (2019). The ATL08 land and vegetation product for the ICESat-2 Mission. *Remote Sensing of Environment*, *221*, 247–259. https://doi.org/10.1016/j.rse.2018.11.005

Neuenschwander, A., Pitts, K., Jelley, B. J., Robbins, J., Markel, J., Popescu, S., Nelson, R., Harding, D., Pederson, Klotz, B., & Sheridan, R. (2023). *Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project Algorithm Theoretical Basis Document (ATBD) for Land—Vegetation Along-Track Products (ATL08), version 6*. https://doi.org/10.5067/8ANPSL1NN7YS

Neumann, T. A., Brenner, A., Hancock, D., Robins, J., Saba, J., Harbeck, K., Gibbons, A., Lee, J., Luthcke, S., & Rebold, T. (2023). *Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project Algorithm Theoretical Basis Document (ATBD) for Global Geolocated Photons ATL03, version 6*. https://doi.org/10.5067/GA5KCLJT7LOT

Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., … Thomas, T. C. (2019a). The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System. *Remote Sensing of Environment*, *233*, 111325. https://doi.org/10.1016/j.rse.2019.111325

Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., … Thomas, T. C. (2019b). The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System. *Remote Sensing of Environment*, *233*, 111325. https://doi.org/10.1016/j.rse.2019.111325

Noordermeer, L., Bollandsås, O. M., Gobakken, T., & Næsset, E. (2018). Direct and indirect site index determination for Norway spruce and Scots pine using bitemporal airborne laser scanner data. *Forest Ecology and Management*, *428*, 104–114. https://doi.org/10.1016/j.foreco.2018.06.041

Noordermeer, L., Gobakken, T., Næsset, E., & Bollandsås, O. M. (2020). Predicting and mapping site index in operational forest inventories using bitemporal airborne laser scanner data. *Forest Ecology and Management*, *457*, 117768. https://doi.org/10.1016/j.foreco.2019.117768

Oh, S., Jung, J., Shao, G., Shao, G., Gallion, J., & Fei, S. (2022). High-Resolution Canopy Height Model Generation and Validation Using USGS 3DEP LiDAR Data in Indiana, USA. *Remote Sensing*, *14*(4), 935. https://doi.org/10.3390/rs14040935

Omernik, J. M., & Griffith, G. E. (2014). Ecoregions of the Conterminous United States: Evolution of a Hierarchical Spatial Framework. *Environmental Management*, *54*(6), 1249–1266. https://doi.org/10.1007/s00267-014-0364-1

Pan, Y., Birdsey, R. A., Phillips, O. L., Houghton, R. A., Fang, J., Kauppi, P. E., Keith, H., Kurz, W. A., Ito, A., Lewis, S. L., Nabuurs, G.-J., Shvidenko, A., Hashimoto, S., Lerink, B., Schepaschenko, D., Castanho, A., & Murdiyarso, D. (2024). The enduring world forest carbon sink. *Nature*, *631*(8021), 563–569. https://doi.org/10.1038/s41586-024-07602-x

Paul, S., Ghebreyesus, D., & Sharif, H. O. (2019). Brief Communication: Analysis of the Fatalities and Socio-Economic Impacts Caused by Hurricane Florence. *Geosciences*, *9*(2), 58. https://doi.org/10.3390/geosciences9020058

Pingel, T. J., Clarke, K. C., & McBride, W. A. (2013). An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *77*, 21–30. https://doi.org/10.1016/j.isprsjprs.2012.12.002

Podest, E. (2021, March 16). *The Fundamentals of LiDAR*. ARSET - Use of Solar Induced Fluorescence and LIDAR to Assess Vegetation Change and Vulnerability, Jet Propulsion Laboratory, California Institute of Technology.

Pronk, M., Eleveld, M., & Ledoux, H. (2024). Assessing Vertical Accuracy and Spatial Coverage of ICESat-2 and GEDI Spaceborne Lidar for Creating Global Terrain Models. *Remote Sensing*, *16*(13), 2259. https://doi.org/10.3390/rs16132259

Psistaki, K., Tsantopoulos, G., & Paschalidou, A. K. (2024). An Overview of the Role of Forests in Climate Change Mitigation. *Sustainability*, *16*(14), Article 14. https://doi.org/10.3390/su16146089

R Core Team. (2023). *R: A Language and Environment for Statistical Computing* [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org

radt0005. (2025). *FIADB\_Direct* [R]. https://github.com/radt0005/FIADB\_Direct (Original work published 2025)

Rai, N., Ma, Q., Poudel, K. P., Himes, A., & Meng, Q. (2024). Evaluating the Uncertainties in Forest Canopy Height Measurements Using ICESat-2 Data. *Journal of Remote Sensing*, *4*, 0160. https://doi.org/10.34133/remotesensing.0160

Ribas-Costa, V. A., Gastón, A., Bloszies, S. A., Henderson, J. D., Trlica, A., Carter, D. R., Rubilar, R., Albaugh, T. J., & Cook, R. L. (2024). Nature vs. nurture: Drivers of site productivity in loblolly pine (Pinus taeda L.) forests in the southeastern US. *Forest Ecology and Management*, *572*, 122334. https://doi.org/10.1016/j.foreco.2024.122334

Ribas-Costa, V. A., Gastón, A., & Cook, R. L. (2024). Modeling dominant height with USGS 3DEP LiDAR to determine site index in even-aged loblolly pine ( *Pinus taeda* L.) plantations in the southeastern US. *Forestry: An International Journal of Forest Research*, cpae034. https://doi.org/10.1093/forestry/cpae034

Ringard, J., Chiriaco, M., Bastin, S., & Habets, F. (2019). Recent trends in climate variability at the local scale using 40 years of observations: The case of the Paris region of France. *Atmospheric Chemistry and Physics*, *19*(20), 13129–13155. https://doi.org/10.5194/acp-19-13129-2019

Risbøl, O., & Gustavsen, L. (2018). LiDAR from drones employed for mapping archaeology – Potential, benefits and challenges. *Archaeological Prospection*, *25*(4), 329–338. https://doi.org/10.1002/arp.1712

Roberts, S. D., Dean, T. J., Evans, D. L., McCombs, J. W., Harrington, R. L., & Glass, P. A. (2005). Estimating individual tree leaf area in loblolly pine plantations using LiDAR-derived measurements of height and crown dimensions. *Forest Ecology and Management*, *213*(1–3), 54–70. https://doi.org/10.1016/j.foreco.2005.03.025

Roriz, R., Cabral, J., & Gomes, T. (2022). Automotive LiDAR Technology: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6282–6297. https://doi.org/10.1109/TITS.2021.3086804

Rossum, G. van, & Drake, F. L. (2010). *The Python language reference* (Release 3.0.1 [Repr.]). Python Software Foundation.

Schutz, B. E., Zwally, H. J., Shuman, C. A., Hancock, D., & DiMarzio, J. P. (2005). Overview of the ICESat Mission. *Geophysical Research Letters*, *32*(21), 2005GL024009. https://doi.org/10.1029/2005GL024009

Seidl, R., Thom, D., Kautz, M., Martin-Benito, D., Peltoniemi, M., Vacchiano, G., Wild, J., Ascoli, D., Petr, M., Honkaniemi, J., Lexer, M. J., Trotsiuk, V., Mairota, P., Svoboda, M., Fabrika, M., Nagel, T. A., & Reyer, C. P. O. (2017). Forest disturbances under climate change. *Nature Climate Change*, *7*(6), 395–402. https://doi.org/10.1038/nclimate3303

Sen, P. K. (1968). Estimates of the Regression Coefficient Based on Kendall’s Tau. *Journal of the American Statistical Association*, *63*(324), 1379–1389. https://doi.org/10.1080/01621459.1968.10480934

Shtaynberger, J., & Bar, H. (2023). *Equivalence Testing*. Cornell Statistical Consulting Unit. https://cscu.cornell.edu/wp-content/uploads/equiv.pdf

Shuman, C. A., Zwally, H. J., Schutz, B. E., Brenner, A. C., DiMarzio, J. P., Suchdeo, V. P., & Fricker, H. A. (2006). ICESat Antarctic elevation data: Preliminary precision and accuracy assessment. *Geophysical Research Letters*, *33*(7), 2005GL025227. https://doi.org/10.1029/2005GL025227

Socha, J., Pierzchalski, M., Bałazy, R., & Ciesielski, M. (2017). Modelling top height growth and site index using repeated laser scanning data. *Forest Ecology and Management*, *406*, 307–317. https://doi.org/10.1016/j.foreco.2017.09.039

Sturtevant, B. R., & Fortin, M.-J. (2021). Understanding and Modeling Forest Disturbance Interactions at the Landscape Level. *Frontiers in Ecology and Evolution*, *9*, 653647. https://doi.org/10.3389/fevo.2021.653647

Sumnall, M. J., Albaugh, T. J., Carter, D. R., Cook, R. L., Hession, W. C., Campoe, O. C., Rubilar, R. A., Wynne, R. H., & Thomas, V. A. (2022). Effect of varied unmanned aerial vehicle laser scanning pulse density on accurately quantifying forest structure. *International Journal of Remote Sensing*, *43*(2), 721–750. https://doi.org/10.1080/01431161.2021.2023229

Sun, T., Qi, J., & Huang, H. (2020). Discovering forest height changes based on spaceborne lidar data of ICESat-1 in 2005 and ICESat-2 in 2019: A case study in the Beijing-Tianjin-Hebei region of China. *Forest Ecosystems*, *7*(1), 53. https://doi.org/10.1186/s40663-020-00265-w

Tang, J., Luyssaert, S., Richardson, A. D., Kutsch, W., & Janssens, I. A. (2014). Steeper declines in forest photosynthesis than respiration explain age-driven decreases in forest growth. *Proceedings of the National Academy of Sciences*, *111*(24), 8856–8860. https://doi.org/10.1073/pnas.1320761111

The pandas development team. (2024). *pandas-dev/pandas: Pandas* (Version v2.2.3) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3509134

Theil, H. (1992). A Rank-Invariant Method of Linear and Polynomial Regression Analysis. In B. Raj & J. Koerts (Eds.), *Henri Theil’s Contributions to Economics and Econometrics* (Vol. 23, pp. 345–381). Springer Netherlands. https://doi.org/10.1007/978-94-011-2546-8\_20

U.S. Department of Agriculture Forest Service. (2023, January 13). *Forest Inventory and Analysis*. Research. https://research.fs.usda.gov/programs/fia

U.S. Department of Agriculture Forest Service. (2024). *The Forest Inventory and Analysis Database User Guide (NFI): Vol. Database Description*.

U.S. Geological Survey. (2024). *Annual National Land Cover Database (NLCD)* (Science Product User Guide No. LSDS-2103; p. 43). https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/media/files/LSDS-2103%20Annual%20National%20Land%20Cover%20Database%20%28NLCD%29%20Collection%201%20Science%20Product%20User%20Guide%20-v1.0%202024\_10\_15%201.pdf

Wachs, S. (2015). *What Is Equivalence Testing & When Should We Use It?* Integral Concepts, Inc. https://www.integral-concepts.com/wp-content/media/What-is-Equivalence-Testing-and-When-Should-We-Use-It.pdf

Wang, X., Liang, X., Gong, W., Häkli, P., & Wang, Y. (2024). Accuracy fluctuations of ICESat-2 height measurements in time series. *International Journal of Applied Earth Observation and Geoinformation*, *135*, 104234. https://doi.org/10.1016/j.jag.2024.104234

Wild, C. J., & Seber, G. A. F. (2000). The Wilcoxon Test. In *Chance encounters: A first course in data analysis and inference* (p. 10). Wiley.

Wilkes, P., Jones, S. D., Suarez, L., Haywood, A., Woodgate, W., Soto-Berelov, M., Mellor, A., & Skidmore, A. K. (2015). Understanding the Effects of ALS Pulse Density for Metric Retrieval across Diverse Forest Types. *Photogrammetric Engineering & Remote Sensing*, *81*(8), 625–635. https://doi.org/10.14358/PERS.81.8.625

Xi, Z., Xu, H., Xing, Y., Gong, W., Chen, G., & Yang, S. (2022). Forest Canopy Height Mapping by Synergizing ICESat-2, Sentinel-1, Sentinel-2 and Topographic Information Based on Machine Learning Methods. *Remote Sensing*, *14*(2), 364. https://doi.org/10.3390/rs14020364

Xiang, B., Wielgosz, M., Kontogianni, T., Peters, T., Puliti, S., Astrup, R., & Schindler, K. (2024). Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. *Remote Sensing of Environment*, *305*, 114078. https://doi.org/10.1016/j.rse.2024.114078

Yu, C., & Yao, W. (2017). Robust linear regression: A review and comparison. *Communications in Statistics - Simulation and Computation*, *46*(8), 6261–6282. https://doi.org/10.1080/03610918.2016.1202271

Yu, J., Nie, S., Liu, W., Zhu, X., Lu, D., Wu, W., & Sun, Y. (2022). Accuracy Assessment of ICESat-2 Ground Elevation and Canopy Height Estimates in Mangroves. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5. https://doi.org/10.1109/LGRS.2021.3107440

Yu, Q., Ryan, M. G., Ji, W., Prihodko, L., Anchang, J. Y., Kahiu, N., Nazir, A., Dai, J., & Hanan, N. P. (2024). Assessing canopy height measurements from ICESat-2 and GEDI orbiting LiDAR across six different biomes with G-LiHT LiDAR. *Environmental Research: Ecology*, *3*(2), 025001. https://doi.org/10.1088/2752-664X/ad39f2

Yu, X., Hyyppä, J., Kaartinen, H., & Maltamo, M. (2004). Automatic detection of harvested trees and determination of forest growth using airborne laser scanning. *Remote Sensing of Environment*, *90*(4), 451–462. https://doi.org/10.1016/j.rse.2004.02.001

Zenner, E. K., & Hibbs, D. E. (2000). A new method for modeling the heterogeneity of forest structure. *Forest Ecology and Management*, *129*(1–3), 75–87. https://doi.org/10.1016/S0378-1127(99)00140-1

Zhao, H., Wu, J., Wang, A., Guan, D., & Liu, Y. (2022). Microtopography mediates the climate–growth relationship and growth resilience to drought of Pinus tabulaeformis plantation in the hilly site. *Frontiers in Plant Science*, *13*, 1060011. https://doi.org/10.3389/fpls.2022.1060011

Zhao, K., Suarez, J. C., Garcia, M., Hu, T., Wang, C., & Londo, A. (2018). Utility of multitemporal lidar for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux. *Remote Sensing of Environment*, *204*, 883–897. https://doi.org/10.1016/j.rse.2017.09.007

Zhu, Z., & Woodcock, C. E. (2014). Continuous change detection and classification of land cover using all available Landsat data. *Remote Sensing of Environment*, *144*, 152–171. https://doi.org/10.1016/j.rse.2014.01.011

Zolkos, S. G., Goetz, S. J., & Dubayah, R. (2013). A meta-analysis of terrestrial aboveground biomass estimation using lidar remote sensing. *Remote Sensing of Environment*, *128*, 289–298. https://doi.org/10.1016/j.rse.2012.10.017

1. **Appendix**
   1. **Python Script to Extract Canopy Height Metrics from ALS Data within Working Set of ATL08 Segments**

# THIS SCRIPT GATHERS USGS 3DEP LIDAR DATA IN PARALLEL

# USE IT FOR COMPARING SPACEBORNE AND AIRBORNE CANOPY HEIGHT OBSERVATIONS

import pandas as pd

import pdal

import geopandas as gpd

import json

import numpy as np

import copy

import os

from concurrent.futures import ProcessPoolExecutor

from pathlib import Path

pipelineTemplate = {

    "pipeline" : [

        # READS DATA FROM EPT

        {

            "type": "readers.ept",

            "filename":"" , # BLANK FILENAME TO ACCOMMODATE SPECIFIC 3DEP PROJECT

            "bounds": "", # READING WTIHIN BOUNDS OF SPECIFIC POLYGON

        },

        # removed the elm filter

        # removed the outliers filter

        # HANDLES SMRF ERROR WITH NO RETURNS

        {

            "type": "filters.assign",

            "value": [

              "ReturnNumber = 1 WHERE ReturnNumber < 1",

              "NumberOfReturns = 1 WHERE NumberOfReturns < 1"

            ]

        },

        # REPROJECT TO OUR STUDY AREA CRS: NAD83 NC STATE PLANE

        # {

        #     "type": "filters.reprojection",

        #     "out\_srs": "EPSG:32119",

        #     "tag": "reprojectUTM"

        # },

        # SMRF FILTER TO IDENTIFY GROUND PER PINGEL ET. AL (2013)

        {

          "type":"filters.smrf",

          "ignore":"Classification[7:7]",

          "slope":0.2,

          "window":16,

          "threshold":0.45,

          "scalar":1.2

        },

        # CUTOFF FOR SMRF VS CSF FILTER ROUTES

        {

            "type":"filters.hag\_nn"

        },

        # CROPPING TO BOUNDS OF SPECIFIC POLYGON

        {

            "type":"filters.crop",

            "polygon":""

        },

        # UNCOMMENT IF YOU WANT TO WRITE CROPPED POINT CLOUD FOR TESTING

        # {

        #     "type":"writers.las",

        #     "filename":""

        # }

    ]

}

def calculateMetrics(pointCloudArray, outputUniqueID):

    # Empty object to hold vegetation metrics:

    geosegmentStats = None

    # Getting the normalized heights out as numpy array

    allReturns = pd.DataFrame(pointCloudArray[0])

    # number of total points

    totalPoints = len(allReturns)

    # This is the case where it falls outside the EPT returns, it'll have no returns regardless of the heights

    if totalPoints < 1:

        print(f'NO RETURNS AT ALL FOR THIS UID {outputUniqueID}, FALLS OUTSIDE OF EPT COLLECTION')

        return None

    # These two queries to only vegetation points in the height range of interest between 2 meters and global maxima (39.432 meters)

    vegReturns = allReturns.query('Classification != 2') # removes ground to get only vegetation returns

    validReturns = vegReturns.query('HeightAboveGround > 2 and HeightAboveGround < 39.42389758300781') # valid height range removes shrubs & high noise (birds/powerlines)

    heights = validReturns['HeightAboveGround'] # Now we can just grab the vegetation heights to do statistics

    totalVegPoints = len(heights)

    # This is the case where there is only ground and no valid vegetation (powerlines)

    # This is still interesting as it shows ICESat has spatial inaccuracies

    if totalVegPoints < 1:

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # this will use the number of GROUND returns

            'total\_veg\_points': 0, # this will use the number of GROUND returns

            'veg\_proportion': 0,

            'max': 0,

            'min': 0,

            'mean': 0,

            'median': 0,

            'percentile\_90':0,

            'percentile\_95':0,

            'percentile\_98':0,

            'percentile\_99':0

        }

    # This is the case where there actually is vegetation

    elif totalPoints >= 1:

        # Calculate height metrics

        max\_height = np.max(heights)

        min\_height = np.min(heights)

        mean\_height = np.mean(heights)

        median\_height = np.median(heights)

        # Height percentiles

        p90,p95, p98, p99 = np.percentile(heights,[90,95,98,99])

        # Have to convert to a pandas series to run these stats

        pdSeries = pd.Series(heights)

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # this will use the total number returns

            'total\_veg\_points': totalVegPoints, # this will use the number of VEGETATION returns

            'veg\_proportion': totalVegPoints / totalPoints,

            'max': max\_height,

            'min': min\_height,

            'mean': mean\_height,

            'median': median\_height,

            'percentile\_90':p90,

            'percentile\_95':p95,

            'percentile\_98':p98,

            'percentile\_99':p99

        }

    metricsDf = pd.DataFrame([geosegmentStats])

    return metricsDf

def executePipeline(dataframeRow):

    uid = dataframeRow['segmentUID']

    eptAsset = dataframeRow['asset']

    projectID = dataframeRow['projectID']

    outputID = dataframeRow['outputID']

    # creating copy of pipeline template for specific sample

    pipelineConfig = copy.deepcopy(pipelineTemplate)

    # geometry wkt in original crs as its already in a column

    wkt = dataframeRow.geometry.wkt

    # geometry bounds in format: ([xmin, xmax], [ymin, ymax], [zmin, zmax])

    boundsString = dataframeRow['bounds']

    # boundsString = f"([{bounds[0]}, {bounds[2]}], [{bounds[1]}, {bounds[3]}])"

    # adjusting pipeline specific to the row

    pipelineConfig['pipeline'][0]['bounds'], pipelineConfig['pipeline'][0]['filename'], pipelineConfig['pipeline'][4]['polygon'] = boundsString, eptAsset, wkt

    # uncomment this if you want to write .las file for testing

    # pipelineConfig['pipeline'][5]['filename'] = f"D:\\IceSat\\ETD\\segments\\pointCloudTesting\\{outputID}.las"

    # formats custom pipeline into JSON

    pipelineJSON = json.dumps(pipelineConfig)

    try:

        pipeline = pdal.Pipeline(pipelineJSON) # converts JSON into PDAL pipeline object

        pipeline.execute() # executes pipelins

        PCarray = pipeline.arrays # gets array of return data

        metrics = calculateMetrics(PCarray,outputID) # feeds array of return data into function for calculating vegetation metrics

        return metrics # finally returns vegetation metrics array to main function

    except Exception as e:

        return None

def main():

    # Reading in ATL08 polygons

    polygons = gpd.read\_parquet(r"D:\IceSat\ETD\segments\workingSetPolygonsWITHOUTALSMetrics.parquet")

    # Reprojecting polygons to web mercator (epsg 3857) as thats what ALS data is stored in

    print('\nCreating bounds in epsg 3857')

    polygons3857 = polygons.to\_crs(epsg=3857)

    bounds = polygons3857.geometry.bounds

    # print(bounds)

    # Creating column for bounds of each polygon, serve as boundary filters when streaming ALS returns from server

    polygons3857['bounds'] = bounds.apply(

        lambda row: (

        f"([{row['minx']}, {row['maxx']}], "

        f"[{row['miny']}, {row['maxy']}])"

    ), axis=1)

    rows = [row for \_, row in polygons3857.iterrows()]

    print(f"\nProcessing EPT data for {len(polygons)} polygons...")

    # processing rows in parallel

    with ProcessPoolExecutor(max\_workers=15) as executor:

        results = list(executor.map(executePipeline, rows))

    # creating list of results from pipeline processing

    results = [df for df in results if df is not None]

    # stacking these resulting dataframes

    combinedMetrics = pd.concat(results, axis=0).reset\_index(drop=True)

    print("Merging with polygons")

    # merging polygongs with ALS vegetation metrics

    merged = polygons3857.merge(combinedMetrics, on="outputID", how='left')

    # dropping polygons with no vegetation returns

    mergedDroppedNAs = merged.dropna(subset=['total\_veg\_points'])

    # dropping multi-project polygons by selecting project which yielded max number of ALS vegetation returns

    mergedMaxVegPoints = mergedDroppedNAs.loc[mergedDroppedNAs.groupby('segmentUID')['total\_veg\_points'].idxmax()].reset\_index(drop=True)

    # reprojecting back to NC state plane

    reprojectedMaxVegPoints = mergedMaxVegPoints.to\_crs(32119)

    # Writing to file

    print("Writing to file")

    reprojectedMaxVegPoints.to\_parquet(r"D:\IceSat\ETD\segments\workingSetPolygonsWITHALSMetrics.parquet")

    reprojectedMaxVegPoints.to\_file(r"D:\IceSat\ETD\segments\workingSetPolygonsWITHALSMetrics.GeoJSON")

    print("Done. Results written to workingSetPolygonsWITHALSMetrics.parquet")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

* 1. **Python Script to Extract Canopy Height Metrics from ATL08 Granules**

import pandas as pd

import geopandas as gpd

from scipy import stats

import numpy as np

import os

from concurrent.futures import ProcessPoolExecutor, as\_completed

from pathlib import Path

import glob

import h5py as h5

# Change this to directory of raw .h5 files.

rawGranuleDirectory = r"D:\IceSat\ETD\granules\\*.h5"

# Change this to the directory of extracted ICESat-2 granules. Ouputs will be in .parquet format

outputDirectory = r"D:\IceSat\ETD\extracted"

# Here we define a project-wide CRS for all our ICESat-2 segments, as well as any geometry we want to clip our segments to

# For our purposes, we're using NC state plane

projectCRS = 32119

# Commonly, we need to clip ICESat-2 data to only the segments collected within a focus area. Provide a path for a shapefile to read in

# We're using the bounds of the 2020 usgs 3dep hurricane florence als projects

clippingGeometry = gpd.read\_file(r"D:\IceSat\ETD\mapping\ALSdata\dissolvedProjects.parquet").to\_crs(projectCRS)

def extractFileData(filePath):

    try:

        with h5.File(filePath, 'r') as file:

            filename = file.filename # grabbing the file name from the H5 file object

            outputDataframePath = filename.replace('granules', 'extracted').replace('h5', 'parquet') # file extension for the extacted parquet

            # print('input:', filename, 'output:', outputDataframePath)

            validBeams = getBeams(file)

            allBeamDataframes = []

            for beam in validBeams:

                # Reading in ICESat-2 data from key groups to reduce file opening/closing

                land\_segments = file[beam + '/land\_segments']

                orbit\_info = file['orbit\_info']

                ancillary\_data = file['/ancillary\_data']

                rowCount = len(land\_segments['delta\_time'][:])

                data = {

                # |---------------------- COLLECTION DATA ---------------|

                    'delta\_time': land\_segments['delta\_time'][:],

                    'filename': [filename] \* rowCount,

                    'sc\_orient': [orbit\_info['sc\_orient'][0]] \* rowCount,

                    'beam': [beam] \* rowCount,

                    'segment\_landcover' : land\_segments['segment\_landcover'][:],

                    'rgt' : land\_segments['rgt'][:],

                    'start\_cycle' : [ancillary\_data['start\_cycle'][0]] \* rowCount,

                    'end\_cycle' : [ancillary\_data['end\_cycle'][0]] \* rowCount,

                    'segment\_id\_beg' : land\_segments['segment\_id\_beg'][:],

                    'segment\_id\_end' : land\_segments['segment\_id\_end'][:],

                    'lat' : land\_segments['latitude'][:],

                    'lon' : land\_segments['longitude'][:],

                # |---------------------- CANOPY DATA ---------------|

                    'h\_canopy':land\_segments['canopy/h\_canopy'][:],

                    'canopy\_openness':land\_segments['canopy/canopy\_openness'][:],

                    'h\_mean\_canopy':land\_segments['canopy/h\_mean\_canopy'][:],

                    'h\_median\_canopy':land\_segments['canopy/h\_median\_canopy'][:],

                    'h\_dif\_canopy':land\_segments['canopy/h\_dif\_canopy'][:],

                    'h\_min\_canopy':land\_segments['canopy/h\_min\_canopy'][:],

                    'h\_max\_canopy':land\_segments['canopy/h\_max\_canopy'][:],

                    'toc\_roughness':land\_segments['canopy/toc\_roughness'][:],

                    'h\_canopy\_quad':land\_segments['canopy/h\_canopy\_quad'][:],

                    'n\_ca\_photons':land\_segments['canopy/n\_ca\_photons'][:],

                    'n\_toc\_photons':land\_segments['canopy/n\_toc\_photons'][:],

                    'centroid\_height':land\_segments['canopy/centroid\_height'][:],

                    'h\_canopy\_uncertainty':land\_segments['canopy/h\_canopy\_uncertainty'][:],

                    'can\_noise':land\_segments['canopy/can\_noise'][:],

                    'dem\_h':land\_segments['dem\_h'][:],

                    'h\_te\_best\_fit':land\_segments['terrain/h\_te\_best\_fit'][:],

                    'h\_te\_interp':land\_segments['terrain/h\_te\_interp'][:],

                    'h\_te\_mean':land\_segments['terrain/h\_te\_mean'][:],

                    'h\_te\_median':land\_segments['terrain/h\_te\_median'][:],

                    'h\_te\_max':land\_segments['terrain/h\_te\_max'][:],

                    'h\_te\_min':land\_segments['terrain/h\_te\_min'][:],

                    'n\_te\_photons':land\_segments['terrain/n\_te\_photons'][:],

                    'terrain\_slope':land\_segments['terrain/terrain\_slope'][:],

                    # |---------------------- QUALITY FLAGS ---------------|

                    'brightness\_flag':land\_segments['brightness\_flag'][:],

                    'cloud\_flag':land\_segments['cloud\_flag\_atm'][:],

                    'cloud\_fold\_flag':land\_segments['cloud\_fold\_flag'][:],

                    'dem\_flag':land\_segments['dem\_flag'][:],

                    'dem\_removal\_flag':land\_segments['dem\_removal\_flag'][:],

                    'layer\_flag':land\_segments['layer\_flag'][:],

                    'msw\_flag':land\_segments['msw\_flag'][:],

                    'night\_flag':land\_segments['night\_flag'][:],

                    'ph\_removal\_flag':land\_segments['ph\_removal\_flag'][:],

                    'psf\_flag':land\_segments['psf\_flag'][:],

                    'sat\_flag':land\_segments['sat\_flag'][:],

                    'segment\_watermask':land\_segments['segment\_watermask'][:],

                    'snr':land\_segments['snr'][:],

                    'terrain\_flag':land\_segments['terrain\_flg'][:],

                    'urban\_flag':land\_segments['urban\_flag'][:],

                    # |---------------------- GEOSEGMENT DATA ---------------|

                    # LATITUDE COORDS

                    'lat\_0\_20':land\_segments['latitude\_20m/'][:, 0],

                    'lat\_20\_40':land\_segments['latitude\_20m/'][:, 1],

                    'lat\_40\_60':land\_segments['latitude\_20m/'][:, 2],

                    'lat\_60\_80':land\_segments['latitude\_20m/'][:, 3],

                    'lat\_80\_100':land\_segments['latitude\_20m/'][:, 4],

                    # LONGITUDE COORDS

                    'long\_0\_20':land\_segments['longitude\_20m/'][:, 0],

                    'long\_20\_40':land\_segments['longitude\_20m/'][:, 1],

                    'long\_40\_60':land\_segments['longitude\_20m/'][:, 2],

                    'long\_60\_80':land\_segments['longitude\_20m/'][:, 3],

                    'long\_80\_100':land\_segments['longitude\_20m/'][:, 4],

                    # CANOPY HEIGHTS

                    'h\_canopy\_0\_20':land\_segments['canopy/h\_canopy\_20m/'][:, 0],

                    'h\_canopy\_20\_40':land\_segments['canopy/h\_canopy\_20m/'][:, 1],

                    'h\_canopy\_40\_60':land\_segments['canopy/h\_canopy\_20m/'][:, 2],

                    'h\_canopy\_60\_80':land\_segments['canopy/h\_canopy\_20m/'][:, 3],

                    'h\_canopy\_80\_100':land\_segments['canopy/h\_canopy\_20m/'][:, 4],

                    # TERRAIN ESTIMATES

                    'h\_te\_best\_fit\_0\_20':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 0],

                    'h\_te\_best\_fit\_20\_40':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 1],

                    'h\_te\_best\_fit\_40\_60':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 2],

                    'h\_te\_best\_fit\_60\_80':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 3],

                    'h\_te\_best\_fit\_80\_100':land\_segments['terrain/h\_te\_best\_fit\_20m'][:, 4],

                    # |---------------------- CANOPY HEIGHT METRICS ---------------|

                    'canopy\_h\_metrics\_10':land\_segments['canopy/canopy\_h\_metrics/'][:,0],

                    'canopy\_h\_metrics\_15':land\_segments['canopy/canopy\_h\_metrics/'][:,1],

                    'canopy\_h\_metrics\_20':land\_segments['canopy/canopy\_h\_metrics/'][:,2],

                    'canopy\_h\_metrics\_25':land\_segments['canopy/canopy\_h\_metrics/'][:,3],

                    'canopy\_h\_metrics\_30':land\_segments['canopy/canopy\_h\_metrics/'][:,4],

                    'canopy\_h\_metrics\_35':land\_segments['canopy/canopy\_h\_metrics/'][:,5],

                    'canopy\_h\_metrics\_40':land\_segments['canopy/canopy\_h\_metrics/'][:,6],

                    'canopy\_h\_metrics\_45':land\_segments['canopy/canopy\_h\_metrics/'][:,7],

                    'canopy\_h\_metrics\_50':land\_segments['canopy/canopy\_h\_metrics/'][:,8],

                    'canopy\_h\_metrics\_55':land\_segments['canopy/canopy\_h\_metrics/'][:,9],

                    'canopy\_h\_metrics\_60':land\_segments['canopy/canopy\_h\_metrics/'][:,10],

                    'canopy\_h\_metrics\_65':land\_segments['canopy/canopy\_h\_metrics/'][:,11],

                    'canopy\_h\_metrics\_70':land\_segments['canopy/canopy\_h\_metrics/'][:,12],

                    'canopy\_h\_metrics\_75':land\_segments['canopy/canopy\_h\_metrics/'][:,13],

                    'canopy\_h\_metrics\_80':land\_segments['canopy/canopy\_h\_metrics/'][:,14],

                    'canopy\_h\_metrics\_85':land\_segments['canopy/canopy\_h\_metrics/'][:,15],

                    'canopy\_h\_metrics\_90':land\_segments['canopy/canopy\_h\_metrics/'][:,16],

                    'canopy\_h\_metrics\_95':land\_segments['canopy/canopy\_h\_metrics/'][:,17]

                }

                # Converting dictionary to dataframe to hold granule data

                beamDataframe = pd.DataFrame(data)

                # Describing spacecraft orientation

                beamDataframe['orientation'] = beamDataframe['sc\_orient'].map({0: 'backward', 1: 'forward', 2: 'transition'})

                # Gathering datetime information based on delta time, uses seconds since 2018/01/01 00:00:00

                beamDataframe['datetime'] = pd.to\_datetime(beamDataframe['delta\_time'], unit='s', origin='2018-01-01')

                beamDataframe['year'] = beamDataframe['datetime'].dt.year

                beamDataframe['month'] = beamDataframe['datetime'].dt.month

                # Determining beam strength of collection

                beamDataframe['beam\_strength'] = beamDataframe.apply(determineBeamStrength, axis=1)

                # Assigning a unique ID to the segment describing its collection scheme

                beamDataframe['segmentUID'] = beamDataframe[['start\_cycle', 'rgt', 'segment\_id\_beg', 'beam', 'year']].astype(str).agg('\_'.join, axis=1) # creating unique ID for each 100m segment

                # Append current beam dataframe to array holding all beam dataframes

                allBeamDataframes.append(beamDataframe)

            # Stacking all beam dataframes into single dataframe

            finalDataframe = pd.concat(allBeamDataframes, ignore\_index=True)

            # Here we convert our beam dataframe into a geodataframe to allow for clipping

            fileGeodataframe = gpd.GeoDataFrame(

                finalDataframe, geometry=gpd.points\_from\_xy(finalDataframe.lon, finalDataframe.lat), crs="EPSG:4326"

            ).to\_crs(projectCRS)

            # Clipping to our geometry of interest

            clippedGeodataframe = fileGeodataframe.clip(clippingGeometry).reset\_index(drop=True)

            # Skips writing a file for files that have no ICESat-2 segments after clipping to our geometry

            if clippedGeodataframe.empty:

                return (f'empty dataframe after clip for {filename}')

            # If its not empty, we write to parquet

            clippedGeodataframe.to\_parquet(outputDataframePath) # writing our clipped geodataframe to parquet file

            return (f'Successful extraction of {outputDataframePath} to dataframe of length {len(clippedGeodataframe)}')

    # Error handling if the file could not be opened

    except IOError as errorMessage:

        return f'for {filePath} the following error was received: {errorMessage}'

# Function to determine the strength of the beam based on spacecraft orientation

def determineBeamStrength(row):

    if row['orientation'] == 'forward' and row['beam'] in ['gt1r', 'gt2r', 'gt3r']:

        return 'strong'

    elif row['orientation'] == 'backward' and row['beam'] in ['gt1l', 'gt2l', 'gt3l']:

        return 'strong'

    else:

        return 'weak'

# Functiont to gather beams for a file that actually have land\_segments data

def getBeams(fileObject):

    allBeams = [beam for beam in list(fileObject.keys()) if 'gt' in beam]

    landSegmentBeams = []

    for beam in allBeams:

        if '{}/land\_segments/'.format(beam) in fileObject:

            landSegmentBeams.append(beam)

    return landSegmentBeams

# Function to extract data from ICESat-2 h5 files in parallel

def processGranulesInParallel(filePaths):

    results = []

    with ProcessPoolExecutor(max\_workers=14) as executor:

        future\_to\_file = {executor.submit(extractFileData, file): file for file in filePaths}

        for future in as\_completed(future\_to\_file):

            file = future\_to\_file[future]

            try:

                result = future.result()

                results.append(result)

            except Exception as exc:

                results.append(f"Failure: Exception while processing {file} - {str(exc)}")

    return results

# Main function

def main():

    rawGranules = glob.glob(rawGranuleDirectory)

    print(f'\n|--------------------- There are {len(rawGranules)} granules to process, beginning extraction -----------------|')

    results = processGranulesInParallel(rawGranules)

    # results = list(map(extractFileData, rawGranules))

    for result in results:

        print(result)

    print(f'\n|--------------------- Extraction complete -----------------|')

# Entry point check

if \_\_name\_\_ == "\_\_main\_\_":

    main()

* 1. **Python Script to Filter ATL08 Data and Extract Land Characteristics**

#!/usr/bin/env python

# coding: utf-8

# In[ ]:

from osgeo import gdal

import numpy as np

import geopandas as gpd

import pandas as pd

import rasterio

import os

import dask.dataframe as dd

from tqdm.notebook import trange, tqdm

import glob

pd.set\_option('display.max\_columns', None)

import copy

from collections import Counter

projectCRS = 32119

rasters = []

# # 1. Filtering

# Reading in extracted parquets from all granules within study area

# In[ ]:

extractedGranules = glob.glob("D:/IceSat/ETD/extracted/\*.parquet")

extractedGdfs = []

for granulePath in tqdm(extractedGranules):

    gdf = gpd.read\_parquet(granulePath)

    extractedGdfs.append(gdf)

print(len(extractedGdfs))

extracted = pd.concat(extractedGdfs, ignore\_index=True)

extracted

# In[ ]:

extracted.h\_canopy.max()

# Calculating global cutoff folllowing approach of Popescu & Malambo (2024)

# In[ ]:

valid = extracted.query('h\_canopy < 1000').reset\_index(drop=True) # First removing noise canopy values

globalCutoff = valid['h\_canopy'].quantile(.98) \* 1.2

globalCutoff

# Then applying all filters

# In[ ]:

# Assembles query conditions, which can be commented, removed, added.

# Query column names must match column naming convention in extracted files.

queryConditions = [

    f"h\_canopy > 2 and h\_canopy < {globalCutoff}", # canopy height filter between 2 and 39.432 meters

    "night\_flag == 1", # nighttime collection

    "month >= 5 and month <= 9", # leaf-on collection (May - September)

    "beam\_strength == 'strong'", # strong beam

    "cloud\_flag <= 1", # cloud confidence flag

    "msw\_flag <= 0" # multiple scattering warning

]

# Filters our dataframe based on provided query conditions

filtered = extracted.query(" and ".join(queryConditions)).reset\_index(drop=True)

filtered['yearCentered'] = filtered['year'] - filtered['year'].mean()

filtered

# # 2. Intersecting with land characteristic

# Reading in our florence study area

# In[ ]:

florence = gpd.read\_parquet(r"D:\IceSat\ETD\mapping\ALSdata\dissolvedProjects.parquet")

florence

# ## First intersecting with Level 4 EPA ecoregions

# Reading in ecoregion shapefile

# In[ ]:

l4eco = gpd.read\_file(r"D:\IceSat\ETD\mapping\shapefiles\us\_eco\_l4\_state\_boundaries\Level4Ecoregions.shp")

l4eco

# Intersecting all segment centroids in the study area with the EPA level 2 ecoregions

# In[ ]:

florenceCentroidsWithEco = gpd.sjoin(filtered, l4eco[['geometry','NA\_L2NAME', 'NA\_L3NAME', 'US\_L4NAME']], how='left', predicate='within').drop('index\_right', axis=1)

florenceCentroidsWithEco.rename(columns={'NA\_L2NAME': 'Level2Ecoregion', 'NA\_L3NAME': 'Level3Ecoregion', 'US\_L4NAME': 'Level4Ecoregion'}, inplace=True)

florenceCentroidsWithEco

# ## Function to intersect raster with segments

# In[ ]:

def intersectWithRaster(dataframe, rasterPath, columnName, adjustmentValue):

    print(f'intersecting with {rasterPath}')

    with rasterio.open(rasterPath) as raster:

        if dataframe.crs != raster.crs:

            print('CRS mismatch')

            return None

        else:

            coordinates = [(x, y) for x, y in zip(dataframe["geometry"].x, dataframe["geometry"].y)]

            print('sampling coords')

            dataframe[f"{columnName}"] = [val[0] if val else None for val in raster.sample(coordinates)]

            if adjustmentValue:

                print('adjusting raster values by', adjustmentValue)

                dataframe[f"{columnName}"] = dataframe[f"{columnName}"].apply(lambda x: x + adjustmentValue if pd.notna(x) else x)

            dataframe[f"{columnName}"] = dataframe[f"{columnName}"].astype("Int64")

    print('done')

    return dataframe

# ## First, intersecting with most recent fast loss

# In[ ]:

florenceWithFastLoss = intersectWithRaster(florenceCentroidsWithEco,

                                     rasterPath = r"D:\IceSat\ETD\mapping\imagery\LCMSLandChange\FastLossExtracted.tif",

                                     columnName = 'FastLossYear',

                                     adjustmentValue = False)

florenceWithFastLoss

# In[ ]:

florenceWithFastLoss['FastLossYear'].value\_counts()

# Fast Loss years now grouped by year of segment collection

# Assigning undisturbed cells, removing disturbances with <3 years recovery

# In[ ]:

def groupFastLossYear(row):

    if row['FastLossYear'] == 65535: # values for NA in disturbance history

        return 'Undisturbed'

    elif row['FastLossYear'] > row['year']: # segments with disturbance events after collection year no longer reflect forest conditions

        return 'After Collection'

    else:

        age = row['year'] - row['FastLossYear'] # calculates years since disturbance

        if age <= 3:

            return '0-3 Years'

        elif age <= 10:

            return '<10 Years'

        elif age <= 20:

            return '10-20 Years'

        elif age <= 30:

            return '20-30 Years'

        else:

            return '30+ Years'

florenceWithFastLossAgeGroup = florenceWithFastLoss.assign(FastLossAgeGroup=florenceWithFastLoss.apply(groupFastLossYear, axis=1))

florenceWithFastLossAgeGroup['FastLossAgeGroup'].value\_counts()

# Removing segments with < 3 years of recovery, or disturbance after collection

# In[ ]:

florenceWithFastLossAgeGroup = florenceWithFastLossAgeGroup.query("FastLossAgeGroup != 'After Collection' and FastLossAgeGroup != '0-3 Years'").reset\_index(drop=True)

florenceWithFastLossAgeGroup['FastLossAgeGroup'].value\_counts()

# Then adding a binary yes/no on whether forests have disturbance histories

# In[ ]:

florenceWithFastLossAgeGroup['DisturbancePresence'] = np.where(

    florenceWithFastLossAgeGroup['FastLossAgeGroup'] == 'Undisturbed',

    'No',

    'Yes'

)

florenceWithFastLossAgeGroup['DisturbancePresence'].value\_counts()

# ## Assigning NLCD landcover

# In[ ]:

florenceWithNLCD = intersectWithRaster(florenceWithFastLossAgeGroup,

                                      rasterPath = r"D:\IceSat\ETD\mapping\imagery\NLCDLandCover\LandCoverExtracted.tif",

                                      columnName = 'NLCD\_2023',

                                      adjustmentValue = False)

florenceWithNLCD['NLCD\_2023\_Label'] = florenceWithNLCD['NLCD\_2023'].map({41: 'Deciduous Forest', 42: 'Evergreen Forest', 43: 'Mixed Forest', 90: 'Woody Wetlands'})

florenceWithNLCD

# ## Assigning a Hurricane Florence flood inundation mask

# In[ ]:

florenceWithFlooding = intersectWithRaster(florenceWithNLCD,

                                           rasterPath = r"D:\IceSat\ETD\mapping\imagery\FlorenceFloodExtent\FlorenceFloodExtentMasked.tif",

                                           columnName = 'FloodBinary',

                                           adjustmentValue = False)

# Recoding to a more verbose column

florenceWithFlooding['FloodImpact'] = florenceWithFlooding['FloodBinary'].apply(

    lambda x: 'Unaffected' if pd.isna(x) else 'Affected' if x == 1 else None

)

florenceWithFlooding['FloodImpact'].value\_counts()

# # Finally filtering to only forested cells

# In[ ]:

forestCentroids = florenceWithFlooding[florenceWithFlooding["NLCD\_2023"].isin([41, 42, 43, 90])].reset\_index(drop=True)

print(len(forestCentroids))

forestCentroids

# # Writing out the forest centroids within the study area

# In[ ]:

forestCentroids.to\_parquet(r"D:\IceSat\ETD\segments\workingSetCentroids.parquet")

forestCentroids.to\_file(r"D:\IceSat\ETD\segments\workingSetCentroids.GeoJSON")

* 1. **Python Script to Create Polygons of ATL08 Segments**

#!/usr/bin/env python

# coding: utf-8

import numpy as np

import geopandas as gpd

import pandas as pd

import os

import dask.dataframe as dd

pd.set\_option('display.max\_columns', None)

import shapely

projectCRS = 32119

rasters = []

# using the satellite track inclination to determine angle of resulting atl08 polygon

def determineAngleFromKML(dataframe, allGroundTrackKMLs, eastCoastEnvelope, projectCRS):

    # sorting by collection order of ICESat-2 data

    dataframe = dataframe.sort\_values('datetime')

    cycleNumber = dataframe['start\_cycle'].iloc[0]

    rgtNumber = str(dataframe['rgt'].iloc[0]).zfill(4)

    # Determining relevant kml file to read from

    matchingFile = next((file for file in allGroundTrackKMLs if f'RGT\_{rgtNumber}\_cycle{cycleNumber}' in os.path.basename(file)), None)

    # reading KML file

    kml = gpd.read\_file(matchingFile, driver='libKML')

    # # Creating a dataframe for our kml geometry and renaming columns

    kmlDf = pd.DataFrame(kml.geometry[0].coords)

    kmlDf.columns = ['longitude', 'latitude', 'z']

    # Converting dataframe to a geodataframe and clipping to the east coast,

    # this is crucial for correctly determining satellite inclination passing through study area

    kmlGdf = gpd.GeoDataFrame(

        kmlDf, geometry=gpd.points\_from\_xy(kmlDf.longitude, kmlDf.latitude), crs="EPSG:4326"

    ).to\_crs(projectCRS).clip(eastCoastEnvelope)

    kmlGdf = kmlGdf.to\_crs(32119)

    # Getting first row, i.e., the first moment the track passes through the east coast

    # print('LENGTH:',len(kmlGdf))

    firstRow = kmlGdf.iloc[0]

    # Getting last row, i.e., the last moment the track passes through the east coast

    lastRow = kmlGdf.iloc[-1]

    # VERTICAL DIRECTION

    # moving south to north

    if firstRow['latitude'] < lastRow['latitude']:

        # print('Moving SOUTH to NORTH, bottom to top')

        #HORIZONTAL DIRECTION

        # the earliest segment is more WESTWARD than the last, platform moving SW to NE

        if firstRow['longitude'] < lastRow['longitude']:

            # return ('rotation should be +2')

            return -2

        # HORIZONTAL DIRECTION

        # ealiest segment is more EASTWARD than the last, platform is moving SE to NW

        elif firstRow['longitude'] > lastRow['longitude']:

            # primt('rotation should be -2')

            return 2

    # VERTICAL DIRECTION

    # moving north to south

    elif firstRow['latitude'] > lastRow['latitude']:

        # print('Moving NORTH to SOUTH, top to bottom\n')

        # HORIZONTAL DIRECTION

        # earliest segment is more WESTWARD than last, platform is moving NW to SE

        if firstRow['longitude'] < lastRow['longitude']:

            return 2

        # HORIZONTAL DIRECTION

        # earliest segment is more EASTWARD than last, platform is moving NE to SW

        elif firstRow['longitude'] > lastRow['longitude']:

            return -2

def createSegments(geodataframe, cycleRGT, allGroundTrackKMLs, eastCoastEnvelope, projectCRS):

    angle = determineAngleFromKML(geodataframe, allGroundTrackKMLs, eastCoastEnvelope, projectCRS)

    geodataframe['geometry'] = geodataframe.apply(lambda row: shapely.box(row['geometry'].x - 5.5, row['geometry'].y - 50, row['geometry'].x + 5.5, row['geometry'].y + 50), axis=1)

    # now we assign the previously determined rotaion angle to a column value

    geodataframe['rotation\_angle'] = angle

    # finally, rotating each row's polygon geometry by the corresponding value in the rotation\_angle column

    geodataframe['geometry'] = geodataframe.apply(lambda row: shapely.affinity.rotate(row['geometry'], row['rotation\_angle'], origin=(row['geometry'].centroid.x, row['geometry'].centroid.y)), axis=1)

    return geodataframe

def createPolygons(dataframe, geosegmentFlag, allGroundTrackKMLs, eastCoastEnvelope, projectCRS):

    # Making a dictionary to hold a dataframe for each unique combination of ICESat-2 cycle and reference ground track, there's 143 unique combinations of these variables

    geosegmentDict = {}

    dataframesWithPolygons = []

    for (startCycle, rgt), cycleRGTDataframe in dataframe.groupby(['start\_cycle', 'rgt']):

        geosegmentDict[(startCycle, rgt)] = cycleRGTDataframe

        print('\nCreating segments for cycle:', startCycle, 'rgt:', rgt)

        cycleRGT = f"{startCycle},{rgt}"

        dfWithPolygons = createSegments(cycleRGTDataframe, cycleRGT, allGroundTrackKMLs, eastCoastEnvelope, projectCRS)

        dataframesWithPolygons.append(dfWithPolygons)

    # # Stacking them vertically after into one final dataframe

    finalDataframe = pd.concat(dataframesWithPolygons, ignore\_index=True)

    # uncomment chunk below if want to use 20m geosegments rather than 100m segments

    # will need to create function for it

    # if geosegmentFlag:

    #     # pivoting to geosegments

    #     finalDataframe = pivotToGeosegments(finalDataframe)

    return finalDataframe

def main():

    # Reading in our EPT projects to join with centroids of atl08 segments

    undissolvedProjects = gpd.read\_parquet(r"D:\IceSat\ETD\mapping\ALSdata\undissolvedProjects.parquet")

    undissolvedProjects = undissolvedProjects.rename(columns={'id':'projectID'})

    # Reading in centroids of atl08 segments

    workingSet = gpd.read\_parquet(r"D:\IceSat\ETD\segments\workingSetCentroids.parquet")

    # rough envlope of east coast to trim the ICESat-2 tracks

    eastCoastEnvelope = gpd.read\_file(r"D:\IceSat\ETD\mapping\shapefiles\eastCoastEnvelope\roughEastCoastEnvelope.shp")

    if eastCoastEnvelope.crs != projectCRS:

        print('reprojecting envelope')

        eastCoastEnvelope = eastCoastEnvelope.to\_crs(projectCRS)

    # collecting KMLs of ICESat-2 tracks into array

    allGroundTrackKMLs = []

    parentDir = r"D:\IceSat\ReferenceGroundTrackKMLs"

    for root, dirs, files in os.walk(parentDir):

        for file in files:

            if file.endswith(".kml"):

                allGroundTrackKMLs.append(os.path.join(root, file))

    # Converting centroids to polygons

    print('Converting centroids to polygons')

    polygons = createPolygons(workingSet, geosegmentFlag=False,

                          allGroundTrackKMLs=allGroundTrackKMLs,

                          eastCoastEnvelope=eastCoastEnvelope,

                          projectCRS=projectCRS)

    # Joining atl08 polygons to ept projects

    print('Joining with EPT projects')

    polygonsWithEPT = polygons.sjoin(undissolvedProjects[['geometry']], how='left', predicate='intersects')

    polygonsFull = polygonsWithEPT.merge(undissolvedProjects[['asset', 'projectID']], left\_on='index\_right', right\_index=True, how='left').reset\_index(drop=True)

    # Generating output ID for extracting ALS data within polygons as some polygons intersect with >1 project

    # In that case, this specifies which project is being specified for a given segment

    print('Creating unique IDs')

    polygonsFull['outputID'] = polygonsFull['segmentUID'] + polygonsFull['projectID']

    # Writing to file

    print('Writing to file')

    polygonsFull.to\_parquet(r"D:\IceSat\ETD\segments\workingSetPolygonsWITHOUTALSMetrics.parquet")

    polygonsFull.to\_file(r"D:\IceSat\ETD\segments\workingSetPolygonsWITHOUTALSMetrics.GeoJSON")

    print('All done')

if \_\_name\_\_ == '\_\_main\_\_':

    main()

* 1. **R Script to Access North Carolina FIADB and Conduct Analyses of Phase 3**

---

title: "Phase3\_Reference\_Growth"

format: html

editor: visual

---

## Imports

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(dplyr)

library(RobustLinearReg)

library(ggplot2)

library(glue)

data\_path = file.path("C:/Users/poncy/Desktop/Masters/Research/FIA/FIADB\_Direct/Working")

source(file.path(file.path(data\_path,'FIADB\_Direct\_source\_all.R')))

```

## Global Variables

```{r}

# Light plotting theme

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

# Global minima of 2 meters

globalMinima <- 2

# Global maxima of 39.432 meters

globalMaxima <- 39.42389758300781

```

## Current Plot Measurements

```{r}

NC\_plots\_AGB <- PLOT\_obs(EVAL\_GRP = 372023, ATTRIBUTE\_NBR = 10,GRP\_BY\_ATTRIB = c('unitcd','countycd')) |>

filter(COUNTYCD %in% c(13, 15, 17, 19, 29, 31, 41, 47, 49, 53, 55, 61, 65, 69, 73, 79, 83, 91, 95,101, 103, 107, 117, 127, 129, 131, 133, 137, 139, 141, 143, 147, 155, 163, 177, 181, 185, 187, 191, 195)

) # here i'm generating plot-level estimates from all the plots in evaluation group 372023, corresponding to North Carolina plots visited in 2023. Then im filtering to just those within the counties of my study area

# unique plot codes for EVAL\_GRP 372023 within study area

PLT\_CN <- unique(NC\_plots\_AGB$PLT\_CN) # 1483 unique plot sequence numbers

# this grabs plots from the PLOT table with sequnce numbers matching our NC\_plots\_AGB

plot <- GET\_record(TABLE\_NAME = 'plot',VAR\_NAME = 'cn',VAR\_VALUES = PLT\_CN) |>

rename(PLT\_CN = CN)

# this grabs conditions from the CONDITION table with sequnce numbers matching our NC\_plots\_AGB

cond <- GET\_record(TABLE\_NAME = 'cond',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PLT\_CN)

# unique plot conditions, as there are multiple plot sequences in cond table

PLT\_CN\_ALL <- unique(cond$PLT\_CN)

# this only gets plots which match the unique sequence numbers

plot <- plot |> filter(PLT\_CN %in% PLT\_CN\_ALL)

# i think this gets individual tree records

tree <- GET\_record(TABLE\_NAME = 'tree',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PLT\_CN\_ALL) |>

rename(TRE\_CN = CN)

### Removing dates

plot <- plot[,-grep("\_DATE",names(plot))]

cond <- cond[,-grep("\_DATE",names(cond))]

tree <- tree[,-grep("\_DATE",names(tree))]

### Joining plot, cond, tree

plot\_cond <- left\_join(plot,cond)

plot\_cond\_tree <- left\_join(plot\_cond,tree)

# test <- anti\_join(tree,plot\_cond\_tree) # Trees from plots with partial non-SYP fortypecd

### Creating current df

current\_tree <- plot\_cond\_tree |>

dplyr::select(PLT\_CN,PREV\_PLT\_CN,STATECD,UNITCD,COUNTYCD,MEASYEAR,REMPER,CONDID,COND\_STATUS\_CD,FORTYPCD,STDORGCD,

STDORGSP,DSTRBCD1,DSTRBYR1, TRTCD1,TRE\_CN,PREV\_TRE\_CN,SUBP,TREE,STATUSCD,SPCD,DIA,HT,ACTUALHT,CCLCD,

DAMLOC1,DAMTYP1,DAMSEV1) |>

rename(PLT\_CN2 = PLT\_CN,

MEASYEAR2 = MEASYEAR,

CONDID2 = CONDID,

COND\_STATUS\_CD2 = COND\_STATUS\_CD,

FORTYPCD2 = FORTYPCD,

STDORGCD2 = STDORGCD,

STDORGSP2 = STDORGSP,

DSTRBCD12 = DSTRBCD1,

DSTRBYR12 = DSTRBYR1,

TRTCD12 = TRTCD1,

TRE\_CN2 = TRE\_CN,

STATUSCD2 = STATUSCD,

DIA2 = DIA, HT2 = HT,

ACTUALHT2 = ACTUALHT,

CCLCD2 = CCLCD,

DAMLOC12 = DAMLOC1,

DAMTYP12 = DAMTYP1,

DAMSEV12 = DAMSEV1)

```

## Previous Plot Measurements

```{r}

# grabs the previous sequence numbers for each plot

PREV\_PLT\_CN <- unique(plot$PREV\_PLT\_CN) # 1485 plots

# then gets the plot records for those previous plot records

prev\_plot <- GET\_record('plot','cn',PREV\_PLT\_CN) |>

rename(PLT\_CN = CN)

# then grabs the conditions for the previous plot records

prev\_cond <- GET\_record(TABLE\_NAME = 'cond',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PREV\_PLT\_CN) |>

filter(FORTYPCD %in% c(141, 142, 161, 162, 166, 171, 402, 403, 404, 406, 407, 409, 501, 503, 504, 505, 506, 507, 508, 511, 512, 514, 515, 516, 517, 519, 520, 962, 995, 601, 602, 605, 606, 607, 608, 609, 701, 702, 703, 704, 705, 706, 708, 709))

# finally, joining tree data for previous plot records

prev\_tree <- GET\_record(TABLE\_NAME = 'tree',VAR\_NAME = 'plt\_cn',VAR\_VALUES = PREV\_PLT\_CN) |>

rename(TRE\_CN = CN)

# drop \_DATE variables from previous plot records

prev\_plot <- prev\_plot[,-grep("\_DATE",names(prev\_plot))]

prev\_cond <- prev\_cond[,-grep("\_DATE",names(prev\_cond))]

prev\_tree <- prev\_tree[,-grep("\_DATE",names(prev\_tree))]

### Joining previous plot, cond, tree

# joinging condiiton and tree data from previous plot records

prev\_plot\_cond <- left\_join(prev\_plot,prev\_cond)

prev\_plot\_cond\_tree <- left\_join(prev\_plot\_cond,prev\_tree)

### Creating previous df

# establishing a df of previous tree measurments

previous\_tree <- prev\_plot\_cond\_tree |>

dplyr::select(PLT\_CN,STATECD,UNITCD,COUNTYCD,MEASYEAR,CONDID,COND\_STATUS\_CD,FORTYPCD,STDORGCD,

STDORGSP,DSTRBCD1,DSTRBYR1, TRTCD1,TRE\_CN,SUBP,TREE,STATUSCD,SPCD,DIA,HT,ACTUALHT,CCLCD,

DAMLOC1,DAMTYP1,DAMSEV1)

```

## Joining Remeasurements

```{r}

# The number of years to match our ICESat-2 ATL08 segments timeframe

timeframeLength <- 5

## 4: Joining current & previous measurments, join that is species-specific

fullDf <- left\_join(current\_tree,previous\_tree,

by=join\_by(STATECD,UNITCD,COUNTYCD,SUBP,TREE,SPCD,

PREV\_PLT\_CN == PLT\_CN,PREV\_TRE\_CN == TRE\_CN)) |>

mutate(DIA\_INC = (DIA2-DIA)/REMPER,

DIA\_CHNG = DIA2-DIA,

# Calculating change measurements

HT\_CHNG = HT2-HT, # net change in height (feet)

HT\_INC = (HT2-HT)/REMPER, # height increment (feet)

!!paste0("NET\_CHNG\_", timeframeLength, "YR") := (HT2-HT)/REMPER \* timeframeLength, # scaling rate to net change over the length of our timeframe (feet)

ACTUALHT\_CHNG = ACTUALHT2-ACTUALHT, # net change in actual height (feet)

ACTUALHT\_INC = (ACTUALHT2-ACTUALHT)/REMPER, # actual height increment (feet)

!!paste0("NET\_ACTUALCHNG\_", timeframeLength, "YR") := (ACTUALHT2-ACTUALHT)/REMPER \* timeframeLength, # scaling rate to net actual change over the length of our timeframe (feet)

# Getting measurements in meters

HT\_m = HT \* .3048,

HT2\_m = HT2 \* .3048,

ACTUALHT\_m = ACTUALHT \* .3048,

ACTUALHT2\_m = ACTUALHT2 \* .3048,

# Calculating change measurements in meters

HT\_CHNGm = (HT2\_m-HT\_m), # net change in height (meters)

HT\_INCm = (HT2\_m-HT\_m)/REMPER, # height increment (meters)

!!paste0("NET\_CHNG\_", timeframeLength, "YR\_m") := (HT2\_m-HT\_m)/REMPER \* timeframeLength, # scaling rate to net change over the length of our timeframe (meters)

ACTUALHT\_CHNGm = ACTUALHT2\_m-ACTUALHT\_m, # net change in actual height (meters)

ACTUALHT\_INCm = (ACTUALHT2\_m-ACTUALHT\_m)/REMPER, # actual height increment (meters)

!!paste0("NET\_ACTUALCHNG\_", timeframeLength, "YR\_m") := (ACTUALHT2\_m-ACTUALHT\_m)/REMPER \* timeframeLength, # scaling rate to net actual change over the length of our timeframe (meters)

# i think this piece accounts for dia\_inc only in planted stands, as natural are multiplied by 0

DIAINC\_STDORG = DIA\_INC\*STDORGCD) |>

filter(STATUSCD <= 1 & STATUSCD2 <= 1) # STATUS CODE to keep alive trees

```

## Adding Characteristics

```{r}

## -------------------------- Mapping forest type ----------------------

fortypMapping <- c(

# Evergreens

"141" = "Evergreen Forest", "142" = "Evergreen Forest", "161" = "Evergreen Forest",

"162" = "Evergreen Forest", "166" = "Evergreen Forest", "171" = "Evergreen Forest",

"606" = "Evergreen Forest",

# then deciduous

"501" = "Deciduous Forest", "503" = "Deciduous Forest", "504" = "Deciduous Forest",

"505" = "Deciduous Forest", "506" = "Deciduous Forest", "507" = "Deciduous Forest",

"508" = "Deciduous Forest", "511" = "Deciduous Forest", "512" = "Deciduous Forest",

"514" = "Deciduous Forest", "515" = "Deciduous Forest", "516" = "Deciduous Forest",

"517" = "Deciduous Forest", "519" = "Deciduous Forest", "520" = "Deciduous Forest",

"701" = "Deciduous Forest", "702" = "Deciduous Forest", "703" = "Deciduous Forest",

"704" = "Deciduous Forest", "705" = "Deciduous Forest", "706" = "Deciduous Forest",

"708" = "Deciduous Forest", "709" = "Deciduous Forest", "962" = "Deciduous Forest",

# then mixed

"402" = "Mixed Forest", "403" = "Mixed Forest", "404" = "Mixed Forest", "406" = "Mixed Forest",

"407" = "Mixed Forest", "409" = "Mixed Forest",

# then woody wetlands

"601" = "Woody Wetlands", "602" = "Woody Wetlands", "605" = "Woody Wetlands",

"607" = "Woody Wetlands", "608" = "Woody Wetlands", "609" = "Woody Wetlands",

# i'll leave unknown just as-is

"999" = "Unknown"

)

# then applying the mapping

fullDf <- fullDf |>

mutate(NLCD\_2023\_Label = recode(as.character(FORTYPCD), !!!fortypMapping)

)

## -------------- Mapping disturbance presence ----------------

# ADDING DISTURBANCE PRESENCE

fullDf$DisturbancePresence <- ifelse(fullDf$DSTRBCD12 == 0, 'No', 'Yes')

# ADDING YEARS SINCE DISTURBANCE

fullDf$YearsSinceDisturbance <- fullDf$MEASYEAR2 - fullDf$DSTRBYR12

# here marking continuous disturbances (9999 for disturbance year) and were negative in prev calculation

fullDf$YearsSinceDisturbance <- ifelse(fullDf$YearsSinceDisturbance < 0, 'Continuous', fullDf$YearsSinceDisturbance)

```

## Filtering

This entails filters for

- only positive height change in heights (both total and actual)

- current and previous heights total and actual heights between 2 (global minima) and 432 meters (global maxima)

- remeasurement period of \~6 years

```{r}

filtered <- fullDf |>

filter(!is.na(HT\_CHNGm) & HT\_CHNGm >= 0) |> # removing NA or negative changes in total height

filter(!is.na(ACTUALHT\_CHNGm) & ACTUALHT\_CHNGm >= 0) |> # removing NA or negative changes in actual height

filter(HT\_m > globalMinima & HT\_m < globalMaxima) |> # filtering previous total heights to within our min and max values

filter(HT2\_m > globalMinima & HT2\_m < globalMaxima) |> # filtering current total heights to within our min and max values

filter(ACTUALHT\_m > globalMinima & ACTUALHT\_m < globalMaxima) |> # filtering previous actual heights to within our min and max values

filter(ACTUALHT2\_m > globalMinima & ACTUALHT2\_m < globalMaxima) |> # filtering current actual heights to within our min and max values

filter(REMPER > (timeframeLength - 0.1) & REMPER < (timeframeLength + 1.1)) # here going to match the length of our ICESat-2 ATL08 segment timeframe

```

## Plot-Level Summary

```{r}

# Increment in actual height

medianHeightIncrements <- filtered |>

group\_by(PLT\_CN2) |> # grouping by plot sequence numbers at revisit

summarise("Height Increment" = median(ACTUALHT\_INCm, na.rm = TRUE)) # getting plot-level median of increment in actual tree height

# Net change in actual height

medianHeightChange <- filtered |>

group\_by(PLT\_CN2) |> # aggregating up by plot sequence numbers at revisit

summarise("5 Year Net Change" = median(NET\_ACTUALCHNG\_5YR\_m, na.rm = TRUE)) # getting plot-level median of 5-year net change in actual tree height

# Here we're merging net change and increment into a single Df

plotLevelChanges <- merge(medianHeightChange, medianHeightIncrements, by="PLT\_CN2")

cat("\nPlot-level summary of actual height increment\n")

summary(plotLevelChanges$`Height Increment`)

cat("\nPlot-level summary of net change in actual height\n")

summary(plotLevelChanges$`5 Year Net Change`)

```

## Plot-Level Graphing

## Height Increment

```{r}

# increment in actual height

actualHTINCPlot <- ggplot(plotLevelChanges, aes(x = `Height Increment`)) +

geom\_histogram(binwidth = 0.05, fill = "#3689D2", color = "black") +

# geom\_density(fill='lightblue', alpha=0.3)+

geom\_vline(xintercept=median(plotLevelChanges$`Height Increment`), color='red', lwd=.75)+

annotate("text", x = .6, y = 34, size=5, face='bold', color='black', label = glue("Median: {round(median(plotLevelChanges$`Height Increment`), 2)

}"))+

labs(title = "Remeasurement Plot Yearly Height Increment",

x = "Height Increment (meters per year)",

y = "Frequency")+

theme\_minimal() +

customTheme

actualHTINCPlot

```

Saving

```{r}

ggsave("D:/IceSat/ETD/figures//Phase3\_increment\_plot.png", plot = actualHTINCPlot + customTheme, bg = "transparent", width = 8, height = 5, units = "in", dpi = 300)

```

### Net change

```{r}

actualHTNetChangePlot <- ggplot(plotLevelChanges, aes(x = `5 Year Net Change`)) +

geom\_histogram(binwidth = 0.3,fill = "#8A7D63", color = "black") +

# geom\_density(fill='lightblue', alpha=0.3)+

geom\_vline(xintercept=median(plotLevelChanges$`5 Year Net Change`), color='red', lwd=.75)+

annotate("text", x = 2.95, y = 42, size=5, face='bold', color='black', label = glue("Median: {round(median(plotLevelChanges$`5 Year Net Change`), 2)

}"))+

labs(title = "Remeasurement Plot Five-Year Net Change",

x = "Net Change in Height (m)",

y = "Density")+

theme\_minimal() +

customTheme

actualHTNetChangePlot

```

Saving

```{r}

ggsave("D:/IceSat/ETD/figures/Phase3\_netchange\_plot.png", plot = actualHTNetChangePlot + customTheme, bg = "transparent", width = 8, height = 5, units = "in", dpi = 300)

```

* 1. **R Script to Conduct Analyses of Phase 1**

---

title: "Phase1\_Validation"

format: html

editor: visual

---

## Libraries

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(RobustLinearReg)

library(caret)

library(mgcv)

library(arrow)

library(dplyr)

library(cowplot)

library(broom)

library(glue)

library(trend)

library(TOSTER)

```

## Global Variables

### Plot Theme

```{r}

# Light plotting theme

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

```

## Reading Data

Our validation set is ICESat-2 ATL08 segments with:

- \> 3 years recovery

- disturbance before the year of collection

Further, we only use segments collected in 2020 for temporal alignment with ALS data

```{r}

validationSet <- read\_parquet("D:/IceSat/ETD/segments/workingSetPolygonsWITHALSMetrics.parquet") |>

filter(year == 2020)

```

## Regression

Determining correlation of canopy heights estimated by ATL08 segments against coincident ALS

```{r}

validationTSRModel <- theil\_sen\_regression(h\_canopy ~ percentile\_98, data=validationSet)

validationTSRPredictions <- predict(validationTSRModel)

validationTSRrSquared <- summary(validationTSRModel)$r.squared

validationTSRRMSE <- sqrt(mean(validationTSRModel$residuals^2))

summary(validationTSRModel)

```

## Plotting

```{r}

validationPlot <- ggplot(validationSet, aes(x = percentile\_98, y = h\_canopy)) +

geom\_point(color = "black", size = 1, alpha = 0.2) +

geom\_line(aes(y = validationTSRPredictions), color = 'red', linewidth = 1) +

labs(

title = "Correlation of 2020 ALS & ATL08 Canopy Heights",

subtitle = glue("R²: {round(validationTSRrSquared, 2)} RMSE: {round(validationTSRRMSE, 2)}m"),

x = "ALS Canopy Height (m)",

y = "Segment Canopy Height (m)",

# colour = "Forest Cover Type"

) +

customTheme +

guides(color = guide\_legend(override.aes = list(size = 10, color = "white")))

validationPlot

```

## Saving

```{r}

ggsave(

"D:/IceSat/ETD/figures/Phase1\_validation\_plot.png",

plot = validationPlot,

bg = "transparent",

width = 8, height = 5, units = "in", dpi = 300

)

```

* 1. **Python Script to Generate Population of Forests Throughout Study Area**

#!/usr/bin/env python

# coding: utf-8

# # Generating population of forests in study area

# ## Libraries

# In[18]:

from osgeo import gdal

import numpy as np

import geopandas as gpd

import pandas as pd

import rasterio

import os

from shapely.geometry import box

from shapely.geometry import Point

import random

import copy

projectCRS = 32119

# ## First creating ATL08 segments for forest population

# Reading raster of land cover types within study area, and extracting forest cells of NLCD code 41, 42, 43, or 90

# In[2]:

rasterPath = r"D:\IceSat\ETD\mapping\imagery\NLCDLandCover\LandCoverExtracted.tif"

with rasterio.open(rasterPath) as src:

    raster = src.read(1)

    transform = src.transform

    crs = src.crs

forestValues = [41, 42, 43, 90]

forestMask = np.isin(raster, forestValues)

forestIndices = np.column\_stack(np.where(forestMask))

forestIndices

# Randomly sampling 50,000 forest cells

# In[3]:

sampledIndices = random.sample(range(len(forestIndices)), 50000)

sampledCoords = forestIndices[sampledIndices]

sampledCoords

# Getting rows and cols from the sample

# In[4]:

sampledRows = sampledCoords[:, 0]

sampledCols = sampledCoords[:, 1]

# Getting the forest cover types at the sampled indices

# In[6]:

cellValues = raster[sampledRows, sampledCols]

cellValues

# Creating a geodataframe to hold the points of the subsetted forest coords

# In[7]:

# Getting the coordinates and dropping a point at the center of the cells

xs, ys = rasterio.transform.xy(transform, sampledRows, sampledCols)

geometries = [Point(x, y) for x, y in zip(xs, ys)]

# Creating geodataframe of these cells

gdf = gpd.GeoDataFrame({

    'NLCD\_2023': cellValues,

    'geometry': geometries

}, crs=crs)

# Mapping forest cover typ[es

gdf['NLCD\_2023\_Label'] = gdf['NLCD\_2023'].map({

    41: 'Deciduous Forest',

    42: 'Evergreen Forest',

    43: 'Mixed Forest',

    90: 'Woody Wetlands'

})

gdf

# Seeing distribution of forest cover types

# In[8]:

gdf['NLCD\_2023\_Label'].value\_counts()

# Creating ATL08 polygon at the points of each selected forest cell

# In[9]:

# Buffering is also an option instead of polygons

# gdf.geometry = gdfWithProjects.geometry.buffer(5.5)

# gdf

def make\_rectangle(point):

    x, y = point.x, point.y

    return box(x - 5.5, y - 50, x + 5.5, y + 50)

gdf['geometry'] = gdf.geometry.apply(make\_rectangle)

gdf

gdf.head(100).explore(tiles='CartoDB dark\_matter')

# ## Then intersecting these polygons with EPT projects

# In[12]:

undissolvedProjects = gpd.read\_parquet(r"D:\IceSat\ETD\mapping\ALSdata\undissolvedProjects.parquet")

undissolvedProjects = undissolvedProjects.rename(columns={'id':'projectID'})

undissolvedProjects

# Intersecting with EPT projects

# In[24]:

gdfWithProjects = copy.deepcopy(gdf) # Creating a copy of our gdf which has EPT projects

# Assigning uniqueID BEFORE intersecting with projects, that way we can see segments which intersect >1 project

gdfWithProjects['uniqueID'] = gdfWithProjects.apply(

    lambda row: f"{row.name}\_{row['NLCD\_2023']}",

    axis=1)

# Then we intersect with projects

gdfWithProjects = gdfWithProjects.sjoin(undissolvedProjects[['geometry']], how='left', predicate='intersects')

# Merging with the ALS projects to get the other attributes of EPT projects

gdfWithProjects = gdfWithProjects.merge(undissolvedProjects[['asset', 'projectID']], left\_on='index\_right', right\_index=True, how='left').drop('index\_right',axis=1).reset\_index(drop=True)

# Then creating a unique output ID, which can handle segments which intersect > 1 project by incorporating project name

gdfWithProjects['outputID'] = gdfWithProjects.apply(

    lambda row: f"{row.name}\_{row['NLCD\_2023']}\_{row['projectID']}",

    axis=1

)

gdfWithProjects

# This allows us to check some segments intersecting with multiple projects

# In[27]:

gdfWithProjects['uniqueID'].value\_counts().head()

# Writing to file

# In[28]:

gdfWithProjects.to\_parquet(r"D:/IceSat/ETD/segments/populationPolygonsWITHOUTALSMetrics.parquet")

gdfWithProjects.to\_file(r"D:/IceSat/ETD/segments/populationPolygonsWITHOUTALSMetrics.GeoJSON")

* 1. **Python Script to Extract Canopy Height Metrics from ALS Data within Population of Study Area Forests**

# THIS SCRIPT GATHERS USGS 3DEP LIDAR DATA IN PARALLEL

# USE IT FOR COMPARING SPACEBORNE AND AIRBORNE CANOPY HEIGHT OBSERVATIONS

import pandas as pd

import pdal

import geopandas as gpd

import json

from scipy import stats

import numpy as np

import copy

import os

from concurrent.futures import ProcessPoolExecutor

from pathlib import Path

import glob

pipelineTemplate = {

    "pipeline" : [

        # READS DATA FROM EPT

        {

            "type": "readers.ept",

            "filename":"" , # BLANK FILENAME TO ACCOMMODATE SPECIFIC 3DEP PROJECT

            "bounds": "", # READING WTIHIN BOUNDS OF SPECIFIC POLYGON

        },

        # removed the elm filter

        # removed the outliers filter

        # HANDLES SMRF ERROR WITH NO RETURNS

        {

            "type": "filters.assign",

            "value": [

              "ReturnNumber = 1 WHERE ReturnNumber < 1",

              "NumberOfReturns = 1 WHERE NumberOfReturns < 1"

            ]

        },

        # REPROJECT TO OUR STUDY AREA CRS: NAD83 NC STATE PLANE

        # {

        #     "type": "filters.reprojection",

        #     "out\_srs": "EPSG:32119",

        #     "tag": "reprojectUTM"

        # },

        # SMRF FILTER TO IDENTIFY GROUND PER PINGEL ET. AL (2013)

        {

          "type":"filters.smrf",

          "ignore":"Classification[7:7]",

          "slope":0.2,

          "window":16,

          "threshold":0.45,

          "scalar":1.2

        },

        # NEAREST NEIGHBOR METHOD TO CALCULATE HEIGT ABOVE GROUND FOR EACH RETURN

        {

            "type":"filters.hag\_nn"

        },

        # CROPPING TO BOUNDS OF SPECIFIC POLYGON

        {

            "type":"filters.crop",

            "polygon":""

        },

        # UNCOMMENT IF YOU WANT TO WRITE CROPPED POINT CLOUD FOR TESTING

        # {

        #     "type":"writers.las",

        #     "filename":""

        # }

    ]

}

def calculateMetrics(pointCloudArray, outputUniqueID):

    # Empty object to hold vegetation metrics:

    geosegmentStats = None

    # Getting the normalized heights out as numpy array

    allReturns = pd.DataFrame(pointCloudArray[0])

    # number of total points

    totalPoints = len(allReturns)

    # This is the case where it falls outside the EPT returns, it'll have no returns regardless of the heights

    if totalPoints < 1:

        print(f'NO RETURNS AT ALL FOR THIS UID {outputUniqueID}, FALLS OUTSIDE OF EPT COLLECTION')

        return None

    # These two queries to only vegetation points in the height range of interest between 2 meters and global maxima (39.432 meters)

    vegReturns = allReturns.query('Classification != 2') # removes ground to get only vegetation returns

    validReturns = vegReturns.query('HeightAboveGround > 2 and HeightAboveGround < 39.42389758300781') # valid height range removes shrubs & high noise (birds/powerlines)

    heights = validReturns['HeightAboveGround'] # Now we can just grab the vegetation heights to do statistics

    totalVegPoints = len(heights)

    # This is the case where there is only ground and no valid vegetation (powerlines)

    # This is still interesting as it shows ICESat has spatial inaccuracies

    if totalVegPoints < 1:

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # total number of returns for a polygon

            'total\_veg\_points': 0, # number of vegetation returns for a polygon

            'veg\_proportion': 0,

            'max': 0,

            'min': 0,

            'mean': 0,

            'median': 0,

            'percentile\_90':0,

            'percentile\_95':0,

            'percentile\_98':0,

            'percentile\_99':0

        }

    # This is the case where there actually is vegetation

    elif totalPoints >= 1:

        # Calculate height metrics

        max\_height = np.max(heights)

        min\_height = np.min(heights)

        mean\_height = np.mean(heights)

        median\_height = np.median(heights)

        # Height percentiles

        p90,p95, p98, p99 = np.percentile(heights,[90,95,98,99])

        # Output the results as a dictionary, makes it easy to reformat later

        geosegmentStats = {

            'outputID' : outputUniqueID,

            'total\_points': totalPoints, # total number of returns for a polygon

            'total\_veg\_points': totalVegPoints, # numhber of vegetation returns

            'veg\_proportion': totalVegPoints / totalPoints, # approximation of vegetation density

            'max': max\_height,

            'min': min\_height,

            'mean': mean\_height,

            'median': median\_height,

            'percentile\_90':p90,

            'percentile\_95':p95,

            'percentile\_98':p98,

            'percentile\_99':p99

        }

    metricsDf = pd.DataFrame([geosegmentStats])

    return metricsDf

def executePipeline(dataframeRow):

    uid = dataframeRow['uniqueID']

    eptAsset = dataframeRow['asset']

    projectID = dataframeRow['projectID']

    outputID = dataframeRow['outputID']

    # creating copy of pipeline template for specific sample

    pipelineConfig = copy.deepcopy(pipelineTemplate)

    # geometry wkt in original crs as its already in a column

    wkt = dataframeRow.geometry.wkt

    # geometry bounds in format: ([xmin, xmax], [ymin, ymax], [zmin, zmax])

    boundsString = dataframeRow['bounds']

    # boundsString = f"([{bounds[0]}, {bounds[2]}], [{bounds[1]}, {bounds[3]}])"

    # adjusting pipeline specific to the row

    pipelineConfig['pipeline'][0]['bounds'], pipelineConfig['pipeline'][0]['filename'], pipelineConfig['pipeline'][4]['polygon'] = boundsString, eptAsset, wkt

    # uncomment this if you want to write .las file for testing

    # pipelineConfig['pipeline'][6]['filename'] = f"D:\\IceSat\\Final\\segments\\pointCloudTesting\\{outputID}.las"

    # formats custom pipeline into JSON

    pipelineJSON = json.dumps(pipelineConfig)

    try:

        pipeline = pdal.Pipeline(pipelineJSON) # converts JSON into PDAL pipeline object

        pipeline.execute() # executes pipelins

        PCarray = pipeline.arrays # gets array of return data

        metrics = calculateMetrics(PCarray,outputID) # feeds array of return data into function for calculating vegetation metrics

        # finally returns vegetation metrics array to main function

        return metrics

    except Exception as e:

        return None

def main():

    # reading in polygons

    polygons = gpd.read\_parquet(r"D:\IceSat\ETD\segments\populationPolygonsWITHOUTALSMetrics.parquet")

    # Reprojecting polygons to web mercator (epsg 3857) as thats what ALS data is stored in

    print('\nCreating bounds in epsg 3857')

    polygons3857 = polygons.to\_crs(epsg=3857)

    bounds = polygons3857.geometry.bounds

    # Creating column for bounds of each polygon, serve as boundary filters when streaming ALS returns from server

    polygons3857['bounds'] = bounds.apply(

        lambda row: (

        f"([{row['minx']}, {row['maxx']}], "

        f"[{row['miny']}, {row['maxy']}])"

    ), axis=1)

    rows = [row for \_, row in polygons3857.iterrows()]

    print(f"\nProcessing {len(polygons)} polygons...")

    # Process rows in parallel

    with ProcessPoolExecutor(max\_workers=os.cpu\_count()-2) as executor:

        results = list(executor.map(executePipeline, rows))

    # creating list of results from pipeline processing

    results = [df for df in results if df is not None]

    # stacking these resulting dataframes

    combinedMetrics = pd.concat(results, axis=0).reset\_index(drop=True)

    print(combinedMetrics)

    # merging polygongs with ALS vegetation metrics

    merged = polygons3857.merge(combinedMetrics, on="outputID", how='left')

    # dropping polygons with no vegetation returns

    mergedDroppedNAs = merged.dropna(subset=['total\_veg\_points'])

    # dropping multi-project polygons by selecting project which yielded max number of ALS vegetation returns

    mergedMaxVegPoints = mergedDroppedNAs.loc[mergedDroppedNAs.groupby('uniqueID')['total\_veg\_points'].idxmax()].reset\_index(drop=True) # dropping multi-project polygons by selecting project with max returns

    # reprojecting back to NC state plane

    mergedMaxVegPoints = mergedMaxVegPoints.to\_crs(32119)

    # Writing to file

    mergedMaxVegPoints.to\_parquet(r"D:\IceSat\ETD\segments\populationPolygonsWITHALSMetrics.parquet")

    mergedMaxVegPoints.to\_file(r"D:\IceSat\ETD\segments\populationPolygonsWITHALSMetrics.GeoJSON")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

* 1. **R Script to Conduct Analyses of Phase 2**

---

title: "Phase2\_Addressing\_Sampling\_Bias"

format: html

editor: visual

---

## Libraries

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(RobustLinearReg)

library(caret)

library(mgcv)

library(arrow)

library(dplyr)

library(cowplot)

library(broom)

library(glue)

library(trend)

library(TOSTER)

```

## Global Variables

### Plot Theme

```{r}

# Light plotting theme

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

```

## Reading Data

```{r}

workingSet <- read\_parquet("D:/IceSat/ETD/segments/workingSetPolygonsWITHALSMetrics.parquet")

population <- read\_parquet("D:/IceSat/ETD/segments/populationPolygonsWITHALSMetrics.parquet")

```

## 1. Sample Representation

```{r}

commonBinWidth <- 1

fullALSHist <- ggplot() +

# population histogram faintly in the background

geom\_histogram(data = population, aes(x = percentile\_98),

binwidth = commonBinWidth,

fill = "gray90",

color = "gray60",

size = 0.5,

alpha = .1,

inherit.aes= FALSE,

position = "identity"

) +

# then each year has its own hisotgram

geom\_histogram(data = workingSet, aes(x = percentile\_98, color = as.factor(year), fill=as.factor(year)),

binwidth = commonBinWidth,

size = 1,

position = "identity",

alpha = 0.1

) +

# Then adding labels

labs(

title = "Distribution of ALS Canopy Heights by Sample Year",

x = "ALS Canopy Height (m)",

y = "Count",

fill = "Year",

color = "Year"

) +

theme\_minimal() +

customTheme

fullALSHist

```

### Saving

```{r}

ggsave(

"D:/IceSat/ETD/figures/Phase2\_Sample\_Distributions.png",

plot = fullALSHist,

bg = "transparent",

width = 8, height = 5, units = "in", dpi = 300

)

```

## 2. Sample Location Equivalence

### Bootstrapped TOST

Here we run a bootstrapped TOST for each unique pair of samples across a number of equivalence bounds

```{r}

unique\_years <- sort(unique(workingSet$year))

allPairResults <- list()

equivBounds <- c(2.5, 2, 1.5, 1, 0.5, 0.25, 0.1)

# The outer loop and inner loops combine to get the unique pairs of years

# Outer loop

for (i in 1:(length(unique\_years)-1)) {

# Inner loop

for (j in (i+1):length(unique\_years)) {

# Assigning years

year1 <- unique\_years[i]

year2 <- unique\_years[j]

# Print after defining year1 and year2

print(glue("\n\nyear 1: {year1}, year 2: {year2}"))

# Subsetting our working set to only year1 or year2

currentPairSample <- workingSet |>

filter(year %in% c(year1, year2))

print(glue("length of current df {nrow(currentPairSample)}"))

# Finally, the innermost loop of our equivalence bounds, the years and sample won't change across eqb values

currentPairResults <- list()

for(currentBound in equivBounds){

print(glue("Testing {year1} against {year2} with an EQB of: {currentBound}\n"))

# Running TOST with the current year pair and current EQB

currentBoundTOST <- boot\_t\_TOST(formula = percentile\_98 ~ as.factor(year),

data = currentPairSample,

paired = FALSE,

var.equal = FALSE,

eqb = currentBound,

eqbound\_type="raw"

)

currentBoundBootResults <- data.frame(

yearA = year1,

yearB = year2,

equivalenceBound = currentBound,

pairLength = nrow(currentPairSample),

lower\_p\_value = currentBoundTOST$TOST$p.value[2],

upper\_p\_value = currentBoundTOST$TOST$p.value[3],

result = currentBoundTOST$decision$combined,

lower\_t\_value = currentBoundTOST$TOST$t[2],

upper\_t\_value = currentBoundTOST$TOST$t[3],

lower\_raw\_ci = currentBoundTOST$effsize$lower.ci[1],

upper\_raw\_ci = currentBoundTOST$effsize$upper.ci[1]

)

# appending results of current EQB to full list for current year pair

currentPairResults[[length(currentPairResults) + 1]] <- currentBoundBootResults

}

# stacking all EQB results for current year pair

combinedCurrentPairResults <- bind\_rows(currentPairResults)

# then adding all EQB results of current year pair to list of all year pairs

allPairResults[[length(allPairResults) + 1]] <- combinedCurrentPairResults

}

}

# Combine all results into a dataframe

finalTOSTResults <- bind\_rows(allPairResults)

# View the final dataframe

finalTOSTResults

```

### Bonferroni Correction

Here we convert encode the results of equivalence tests to more explicit format, and use the bonferroni correction

```{r}

# calculating bonferroni corrected p-value

bonferroniPValue <- 0.05 / 15

finalTOSTGrid <- finalTOSTResults |>

mutate(Decision = if\_else(lower\_p\_value < bonferroniPValue & upper\_p\_value < bonferroniPValue, "Yes", "No")) |>

select(yearA, yearB, Decision, equivalenceBound)

finalTOSTGrid

```

### Equivalence Grid

Then we compile the results into a grid format to enable pairwise comparisons

```{r}

allEQBGrid <- finalTOSTGrid |>

filter(equivalenceBound != 2.5) |>

ggplot(aes(yearA, yearB, fill= as.factor(Decision))) +

geom\_tile(width = 0.95, height = 0.95, size = 0.2) +

facet\_wrap(~equivalenceBound, labeller = labeller(equivalenceBound = function(x) paste0(x, " meters")))+

# theme\_minimal()+

scale\_fill\_manual(values = c("Yes" = "#025196", "No" = "#FDB338")) +

labs(title = glue("Sample Location Equivalence by Bound"),

# subtitle = glue("Bonferroni Corrected p-value: {round(bonferroniPValue, 4)}"),

x = "Sample 1",

y = "Sample 2",

fill = "Equivalent?") +

customTheme+

theme(

axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1),

# panel.grid.major = element\_line(color = "grey90", size = 0.3), # faint gridlines

panel.grid.minor = element\_line(color = "grey90", size = 0.3), # optional: hide minor gridlines

)

allEQBGrid

```

### Saving

```{r}

ggsave(

"D:/IceSat/ETD/figures/Phase2\_EQB\_grid.png",

plot = allEQBGrid,

bg = "transparent",

width = 8, height = 5, units = "in", dpi = 300

)

```

## Testing

```{r}

currentPairSample <- workingSet |>

filter(year %in% c(2021, 2024))

rm(currentPairSample)

rm(currentBoundTOST)

currentBoundTOST <- boot\_t\_TOST(formula = percentile\_98 ~ as.factor(year),

data = currentPairSample,

paired = FALSE,

var.equal = FALSE,

eqb = .1,

eqbound\_type="raw"

)

print(currentBoundTOST)

```

* 1. **R Script to Conduct Analyses of Phase 4**

---

title: "Phase4\_ATL08\_Growth"

format: html

editor: visual

---

## Libraries

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(RobustLinearReg)

library(caret)

library(mgcv)

library(arrow)

library(dplyr)

library(cowplot)

library(broom)

library(glue)

library(trend)

library(TOSTER)

library(readxl)

```

## Global Variables

```{r}

# Light plotting theme

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

axis.text = element\_text(size=12, color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

# Number of iterations for bootstrapping

numberOfIterations <- 10000

```

## Reading Data

Our working set is ICESat-2 ATL08 segments with:

- \> 3 years recovery

- disturbance before the year of collection

Further, we remove segments collected in 2019 which demonstrated spatial differences in forest height from equivalence testing

```{r}

workingSet <- read\_parquet("D:/IceSat/ETD/segments/workingSetPolygonsWITHALSMetrics.parquet") |>

filter(!(year == 2019))

```

## Net Growth

### Wilcoxon Rank Test Function

```{r}

wilcoxByFactor <- function(dataframe, factor, level, heightsColumn, firstYear, lastYear){

# Pulling heights from our first year of data

firstYearHeights <- dataframe |>

filter(year == firstYear) |>

filter(.data[[factor]] == level) |>

pull(.data[[heightsColumn]])

# Pulling heights from our last year of data

lastYearHeights <- dataframe |>

filter(year == lastYear) |>

filter(.data[[factor]] == level) |>

pull(.data[[heightsColumn]])

allWilcoxTestResults <- data.frame(

W\_statistic = numeric(),

p\_value = numeric(),

mean\_diff = numeric(),

parameter = numeric()

)

targetN <- min(500, min(length(firstYearHeights), length(lastYearHeights)))

# 10000 iterations of 500 samples seems fine

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

firstYearSample <- sample(firstYearHeights, targetN, replace=TRUE)

lastYearSample <- sample(lastYearHeights, targetN, replace=TRUE)

wilcoxTest <- wilcox.test(lastYearSample, firstYearSample, alternative='greater')

# storing results into df

allWilcoxTestResults <- rbind(allWilcoxTestResults, data.frame(

W\_statistic = wilcoxTest$statistic,

p\_value = wilcoxTest$p.value,

mean\_diff = mean(lastYearSample) - mean(firstYearSample)

))

}

# then aggregating results for more meaninful stats

aggregatedWilcoxResults <- allWilcoxTestResults %>%

summarize(

sampleSize = targetN,

startYear = firstYear,

endYear = lastYear,

firstYearSamples = length(firstYearHeights),

lastYearSamples = length(lastYearHeights),

median\_W\_statistic = median(W\_statistic, na.rm = TRUE),

median\_p\_value = median(p\_value, na.rm = TRUE),

median\_mean\_diff = median(mean\_diff, na.rm = TRUE)

)

return(aggregatedWilcoxResults)

}

samp2020 <- workingSet |>

filter(year == 2020) |>

pull(h\_canopy)

samp2024 <- workingSet |>

filter(year == 2024) |>

pull(h\_canopy)

testWilc <- wilcox.test(samp2024, samp2020, alternative='greater')

testWilc

```

### Results

#### Forest cover type

##### Deciduous Forest

```{r}

wilcoxByFactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy", 2020, 2024)

```

##### Evergreen Forest

```{r}

wilcoxByFactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy", 2020, 2024)

```

##### Mixed Forest

```{r}

wilcoxByFactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy", 2020, 2024)

```

##### Woody Wetlands

```{r}

wilcoxByFactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy", 2020, 2024)

```

#### Disturbance Presence

##### Yes

```{r}

wilcoxByFactor(workingSet, "DisturbancePresence", "Yes", "h\_canopy", 2020, 2024)

```

##### No

```{r}

wilcoxByFactor(workingSet, "DisturbancePresence", "No", "h\_canopy", 2020, 2024)

```

#### Years Since Disturbance

##### \<10 Years

```{r}

wilcoxByFactor(workingSet, "FastLossAgeGroup", "<10 Years", "h\_canopy", 2020, 2024)

```

##### 10-20 Years

```{r}

wilcoxByFactor(workingSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy", 2020, 2024)

```

##### 20-30 Years

```{r}

wilcoxByFactor(workingSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy", 2020, 2024)

```

##### 30+ Years

```{r}

wilcoxByFactor(workingSet, "FastLossAgeGroup", "30+ Years", "h\_canopy", 2020, 2024)

```

### Graphing

#### Assembling Results

```{r}

wilcoxResultsDf <- read\_xlsx("Results.xlsx", sheet = "Wilcoxon")

wilcoxResultsDf$Factor <- factor(wilcoxResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

# Define custom colors by Factor\_Level

custom\_colors <- c(

# forest cover

"Deciduous Forest" = "#a1d99b",

"Evergreen Forest" = "#74c476",

"Mixed Forest" = "#31a354",

"Woody Wetlands" = "#006d2c",

# disturbance binary (reds)

"Yes" = "#fb6a4a",

"No" = "#cb181d",

# years since

"<10 Years" = "#9ecae1",

"10-20 Years" = "#6baed6",

"20-30 Years" = "#3182bd",

"30+ Years" = "#08519c"

)

# converting names to factor to set order

wilcoxResultsDf$Factor\_Level <- factor(wilcoxResultsDf$Factor\_Level, levels = names(custom\_colors))

# then shifting to the order we want

wilcoxResultsDf$Factor <- factor(wilcoxResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

```

#### W statistic

```{r}

wilcoxonWPlot <- ggplot(wilcoxResultsDf, aes(x = Factor, y = W\_Statistic, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "W Statistic by Factor and Level", y = "W", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

) # we'll comment this as its only needed for one plot

wilcoxonWPlot

ggsave("D:/IceSat/ETD/figures/Phase4\_Wilcoxon\_W\_barplot.png", plot = wilcoxonWPlot, bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300)

```

#### Mean Difference

```{r}

wilcoxonMeanDiffsPlot <- ggplot(wilcoxResultsDf, aes(x = Factor, y = Mean\_Difference, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "Mean Difference by Factor and Level", y = "Mean Difference (meters)", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"))

ggsave("D:/IceSat/ETD/figures/Phase4\_Wilcoxon\_MeanDiffs\_barplot.png", plot = wilcoxonMeanDiffsPlot, bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300)

wilcoxonMeanDiffsPlot

```

## Incremental Growth

### Theil Sen Regression Function

```{r}

tsrByfactor <- function(dataframe, factor, factorLevel, heightsColumn){

print(glue("Running theil-sen regression for {factor}: {factorLevel}"))

currentLevelDataframe <- dataframe |>

arrange(.data[[factor]]) |>

filter(.data[[factor]] == factorLevel)

allTSRResults = list()

targetN <- min(500, nrow(currentLevelDataframe))

for(currentIteration in 1:numberOfIterations){

if (currentIteration %% 1000 == 0) {

print(glue("Current iteration: {currentIteration}"))

}

currentIterationSample <- sample\_n(currentLevelDataframe, targetN, replace=TRUE)

tsrModel <- theil\_sen\_regression(reformulate("yearCentered", response = heightsColumn), data = currentIterationSample)

modelIterationSummary <- tidy(tsrModel)

modelIterationSummary$iteration <- currentIteration

modelIterationSummary$sampleSize <- targetN

modelIterationSummary$factor <- factor

modelIterationSummary$factorLevel <- factorLevel

allTSRResults[[currentIteration]] <- modelIterationSummary

}

allTSRResultsDf <- do.call(rbind, allTSRResults)

# Aggregate results

aggregatedTSRResults <- allTSRResultsDf %>%

group\_by(term) %>%

summarize(

factor = unique(factor),

factorLevel = unique(factorLevel),

median\_estimate = median(estimate, na.rm = TRUE),

median\_p\_value = median(p.value, na.rm = TRUE)

)

return(aggregatedTSRResults)

}

```

### Results

#### Forest Cover Type

##### Deciduous Forest

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest", "h\_canopy")

```

##### Evergreen Forest

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest", "h\_canopy")

```

##### Mixed Forest

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest", "h\_canopy")

```

##### Woody Wetlands

```{r}

tsrByfactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands", "h\_canopy")

```

#### Disturbance Presence

##### Yes

```{r}

tsrByfactor(workingSet, "DisturbancePresence", "Yes", "h\_canopy")

```

##### No

```{r}

tsrByfactor(workingSet, "DisturbancePresence", "No", "h\_canopy")

```

#### Years Since Disturbance

##### \<10 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "<10 Years", "h\_canopy")

```

##### 10-20 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "10-20 Years", "h\_canopy")

```

##### 20-30 Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "20-30 Years", "h\_canopy")

```

##### 30+ Years

```{r}

tsrByfactor(workingSet, "FastLossAgeGroup", "30+ Years", "h\_canopy")

```

### Graphing

#### Assembling Results

```{r}

# Here i put the results of each chunk above into an excel sheet to then create a dataframe more cleanly

tsrResultsDf <- read\_xlsx("Results.xlsx", sheet = "TSR")

tsrResultsDf$Factor <- factor(tsrResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

# Define custom colors by Factor\_Level

custom\_colors <- c(

# forest cover

"Deciduous Forest" = "#a1d99b",

"Evergreen Forest" = "#74c476",

"Mixed Forest" = "#31a354",

"Woody Wetlands" = "#006d2c",

# disturbance binary (reds)

"Yes" = "#fb6a4a",

"No" = "#cb181d",

# years since

"<10 Years" = "#9ecae1",

"10-20 Years" = "#6baed6",

"20-30 Years" = "#3182bd",

"30+ Years" = "#08519c"

)

# converting names to factor to set order

tsrResultsDf$Factor\_Level <- factor(tsrResultsDf$Factor\_Level, levels = names(custom\_colors))

# then shifting to the order we want

tsrResultsDf$Factor <- factor(tsrResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

```

#### Slope Values

```{r}

# heres the plot

tsrSlopePlot <- ggplot(tsrResultsDf, aes(x = Factor, y = Slope, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "Growth Rate by Factor and Level", y = "Rate (meters per year)", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"))

ggsave("D:/IceSat/ETD/figures/Phase4\_TSR\_Slope\_barplot.png", plot = tsrSlopePlot, bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300)

tsrSlopePlot

```

#### Intercept Plot

```{r}

# heres the plot

tsrInterceptPlot <- ggplot(tsrResultsDf, aes(x = Factor, y = Intercept, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "Baseline Canopy Height by Factor and Level", y = "Baseline (meters)", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"))

ggsave("D:/IceSat/ETD/figures/Phase4\_TSR\_Intercept\_barplot.png", plot = tsrInterceptPlot, bg = "transparent", width = 10, height = 5, units = "in", dpi = 300)

tsrInterceptPlot

```

## Monotonic Growth

### Mann-Kendall Test Function

```{r}

mkTestByFactor <- function(dataframe, factor, level, heightsColumn, numberOfIterations = 10000) {

# Filter and sort data chronologically for the given level of the factor

filteredData <- dataframe |>

filter(.data[[factor]] == level) |>

arrange(datetime)

allMKResults <- data.frame(

sampleSize = numeric(),

p\_value = numeric(),

z\_statistic = numeric(),

S = numeric(),

tau = numeric()

)

targetN <- min(500, nrow(filteredData))

for (currentIteration in 1:numberOfIterations) {

if (currentIteration %% 1000 == 0) {

print(glue::glue("Current iteration: {currentIteration}"))

}

# Sample row indices WITH REPLACEMENT

sampledIndices <- sample(1:nrow(filteredData), size = targetN, replace = TRUE)

# Create a new bootstrap sample of rows and sort chronologically

bootstrapSample <- filteredData[sampledIndices, ] |>

arrange(datetime)

# Pull canopy height values in time order

heightSeries <- bootstrapSample[[heightsColumn]]

# Perform Mann-Kendall test

mk <- trend::mk.test(heightSeries, alternative = "greater")

# Store the results

allMKResults <- rbind(allMKResults, data.frame(

sampleSize = targetN,

p\_value = as.numeric(mk$p.value),

z\_statistic = as.numeric(mk$statistic),

S = as.numeric(mk$estimates["S"]),

tau = as.numeric(mk$estimates["tau"])

))

}

aggregatedResults <- allMKResults %>%

dplyr::summarize(

sampleSize = targetN,

median\_p\_value = median(p\_value, na.rm = TRUE),

median\_z\_statistic = median(z\_statistic, na.rm = TRUE),

median\_s = median(S, na.rm = TRUE),

median\_tau = median(tau, na.rm = TRUE)

)

return(aggregatedResults)

}

```

### Results

#### Forest Cover Type

##### Deciduous Forest

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Deciduous Forest","h\_canopy")

```

##### Evergreen Forest

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Evergreen Forest","h\_canopy")

```

##### Mixed Forest

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Mixed Forest","h\_canopy")

```

##### Woody Wetlands

```{r}

mkTestByFactor(workingSet, "NLCD\_2023\_Label", "Woody Wetlands","h\_canopy")

```

#### Disturbance Presence

##### Yes

```{r}

mkTestByFactor(workingSet, "DisturbancePresence", "Yes","h\_canopy")

```

##### No

```{r}

mkTestByFactor(workingSet, "DisturbancePresence", "No","h\_canopy")

```

#### Years Since Disturbance

##### \<10 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "<10 Years","h\_canopy")

```

##### 10-20 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "10-20 Years","h\_canopy")

```

##### 20-30 Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "20-30 Years","h\_canopy")

```

##### 30+ Years

```{r}

mkTestByFactor(workingSet, "FastLossAgeGroup", "30+ Years","h\_canopy")

```

### Graphing

#### Assembling Results

```{r}

# Here i put the results of each chunk above into an excel sheet to then create a dataframe more cleanly

mkResultsDf <- read\_xlsx("Results.xlsx", sheet = "MK")

mkResultsDf$Factor <- factor(mkResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

# Define custom colors by Factor\_Level

custom\_colors <- c(

# forest cover

"Deciduous Forest" = "#a1d99b",

"Evergreen Forest" = "#74c476",

"Mixed Forest" = "#31a354",

"Woody Wetlands" = "#006d2c",

# disturbance binary (reds)

"Yes" = "#fb6a4a",

"No" = "#cb181d",

# years since

"<10 Years" = "#9ecae1",

"10-20 Years" = "#6baed6",

"20-30 Years" = "#3182bd",

"30+ Years" = "#08519c"

)

# converting names to factor to set order

mkResultsDf$Factor\_Level <- factor(mkResultsDf$Factor\_Level, levels = names(custom\_colors))

# then shifting to the order we want

mkResultsDf$Factor <- factor(mkResultsDf$Factor, levels = c(

"Forest Cover Type",

"Disturbance Presence",

"Years Since Disturbance"

))

```

#### Z statistic

```{r}

# heres the plot

mkZPlot <- ggplot(mkResultsDf, aes(x = Factor, y = Z, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "Z Statistic by Factor and Level", y = "Z", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"))

ggsave("D:/IceSat/ETD/figures/Phase4\_MK\_Z\_barplot.png", plot = mkZPlot, bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300)

mkZPlot

```

#### Tau

```{r}

mkTauPlot <- ggplot(mkResultsDf, aes(x = Factor, y = Tau, fill = Factor\_Level)) +

geom\_bar(stat = "identity", position = position\_dodge(width = 0.99)) +

scale\_fill\_manual(values = custom\_colors) +

labs(title = "Tau Statistic by Factor and Level", y = "Tau", x=NULL, fill = "Factor Level") +

theme\_minimal() +

customTheme+

theme( axis.text = element\_text(size=10, color = "black"),

# axis.text.x = element\_text(angle = 25, hjust = 1, vjust = 1),

axis.title = element\_text(size=14),

legend.title = element\_text(size = 12, color = "black"),

legend.text = element\_text(size = 10, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"))

ggsave("D:/IceSat/ETD/figures/Phase4\_MK\_Tau\_barplot.png", plot = mkTauPlot, bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300)

mkTauPlot

```

* 1. **R Script to Generate Level II Ecoregion Proportional Coverage from Annual ATL08 Samples**

---

title: "Discussion\_L2Ecoregion\_Coverage"

format: html

editor: visual

---

## Libraries

```{r setup, warning=FALSE, message=FALSE}

library(tidyverse)

library(arrow)

library(dplyr)

library(broom)

library(glue)

```

## Global Variables

```{r}

customTheme <- theme(

plot.title.position = "plot", # can also be "panel"

plot.title = element\_text(hjust = 0.5, size = 20, color = "black"),

plot.subtitle = element\_text(hjust = 0.5, size = 16, color = "black"),

axis.title = element\_text(size=14, color = "black"),

axis.text = element\_text(size=12, color = "black"),

axis.line = element\_line(linewidth = 0.5, color = "black"),

axis.ticks = element\_line(color = "black"),

# axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1), # we'll comment this as its only needed for one plot

legend.title = element\_text(size = 14, color = "black"),

legend.text = element\_text(size = 12, color = "black"),

legend.key.size = unit(.5, "cm"),

legend.key.height = unit(0.5, "cm"),

panel.background = element\_rect(fill = NA, color = NA),

plot.background = element\_rect(fill = NA, color = NA),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

legend.background = element\_rect(fill = NA, color = NA),

legend.box.background = element\_rect(fill = NA, color = NA),

strip.text = element\_text(size = 14, color = "black"),

strip.background = element\_rect(fill = NA, color = NA),

)

```

## Reading Data

```{r}

workingSet <- read\_parquet("D:/IceSat/ETD/segments/workingSetPolygonsWITHALSMetrics.parquet")

```

You can add options to executable code like this

```{r}

result\_list <- list()

# Loop through unique years

for (group in unique(workingSet$year)) {

# Subset data for the current year

subset\_df <- workingSet[workingSet$year == group, ]

# Get value counts for Level2Ecoregion

counts <- as.data.frame(table(subset\_df$Level2Ecoregion))

# Rename columns for clarity

colnames(counts) <- c("Level2Ecoregion", "Count")

# Add year column

counts$Category <- group

# Store in list

result\_list[[as.character(group)]] <- counts

}

# Combining into single df

final\_Ecoregion\_df <- bind\_rows(result\_list)

# Arranging readability

final\_Ecoregion\_df <- final\_Ecoregion\_df |> arrange(Category, Level2Ecoregion)

final\_Ecoregion\_df <- final\_Ecoregion\_df |>

mutate(Level2Ecoregion = recode(Level2Ecoregion, 'MISSISSIPPI ALLUVIAL AND SOUTHEAST USA COASTAL PLAINS' = "Mississippi Alluvial & Southeast USA Coastal Plains", 'SOUTHEASTERN USA PLAINS' = "Southern USA Plains")) |>

group\_by(Category) |>

mutate(Proportion = Count / sum(Count) \* 100)

```

## Graphing

```{r}

l2ProportionPlot <- ggplot(final\_Ecoregion\_df, aes(fill = Level2Ecoregion, y = Count, x = as.factor(Category))) +

geom\_bar(position = "fill", stat = "identity") +

geom\_text(

aes(label = sprintf("%.1f%%", Proportion / 100 \* 100)),

position = position\_fill(vjust = 0.5),

size = 4,

face = 'bold',

color = 'white'

) +

scale\_fill\_manual(

values = c(

"Mississippi Alluvial & Southeast USA Coastal Plains" = "#002673",

"Southern USA Plains" = "#267300"

),

labels = function(x) str\_wrap(x, width = 20)

) +

labs(

title = "Ecoregion Proportion Coverage by Sample Year",

x = "Sample Year",

y = "Coverage Proportion (Percent)",

fill = "Level II Ecoregion"

) +

scale\_y\_continuous(labels = scales::percent) +

theme\_minimal() +

customTheme

l2ProportionPlot

```

## Saving

```{r}

ggsave(

"D:/IceSat/ETD/figures/Discussion\_Ecoregion\_Proportion\_plot.png",

plot = l2ProportionPlot,

bg = "transparent",

width = 10, height = 5, units = "in", dpi = 300

)

```