

ALS Processing Tutorial - Part II

Fabian Jörg Fischer

2025-12-16

Table of contents

1	Introduction	2
1.1	Setup and Loading Packages	2
2	Sample Data: ALS Point Clouds from G-LiHT, NEON, 3DEP at Harvard Forest	2
2.1	Loading Pre-processed Rasters	3
2.2	Summary Statistics	5
2.3	Temporal Comparison	6
3	Guiding Question: How Does Forest Structure Change at Harvard Forest?	8
4	Approaches	9
4.1	Regrowth Analysis	9
4.2	One-Year Changes	14
4.3	Addressing Lateral Ingrowth	18
4.4	Disturbance Analysis	20
4.5	Tree-Based Changes	21
5	Summary and Discussion	21
6	Key Findings	21
7	Methodological Challenges	21
8	Best Practices	22
9	Research Questions	22
10	Exercises and Extensions	23

! Before You Begin: Data Requirements

This tutorial requires pre-processed ALS raster products. The data are available [here](#). Ensure you have:

- `products/dtms_8km2_12scans.tif`
- `products/dsms_8km2_12scans.tif`
- `products/chms_8km2_12scans.tif`

1 Introduction

This is a Notebook for a tutorial on ALS processing in R (Part II). In this part we analyze **forest structural changes** with ALS-derived rasters. The idea is for each group to experiment with different datasets.

1.1 Setup and Loading Packages

```
# Load libraries in this specific order to avoid conflicts
library(terra)      # for analyzing rasterized products
library(data.table) # fast data manipulation
library(viridis)    # plotting colors
library(ggplot2)    # plotting
library(patchwork)  # plotting layout
```

2 Sample Data: ALS Point Clouds from G-LiHT, NEON, 3DEP at Harvard Forest

As before, we will focus on Harvard Forest, but now the full 8 km², and all of them pre-processed into raster files. Note that we used a maximally robust CHM algorithm. That means it is calibrated to not be influenced by pulse density differences or scan angle differences, among other things.

Dataset Composition

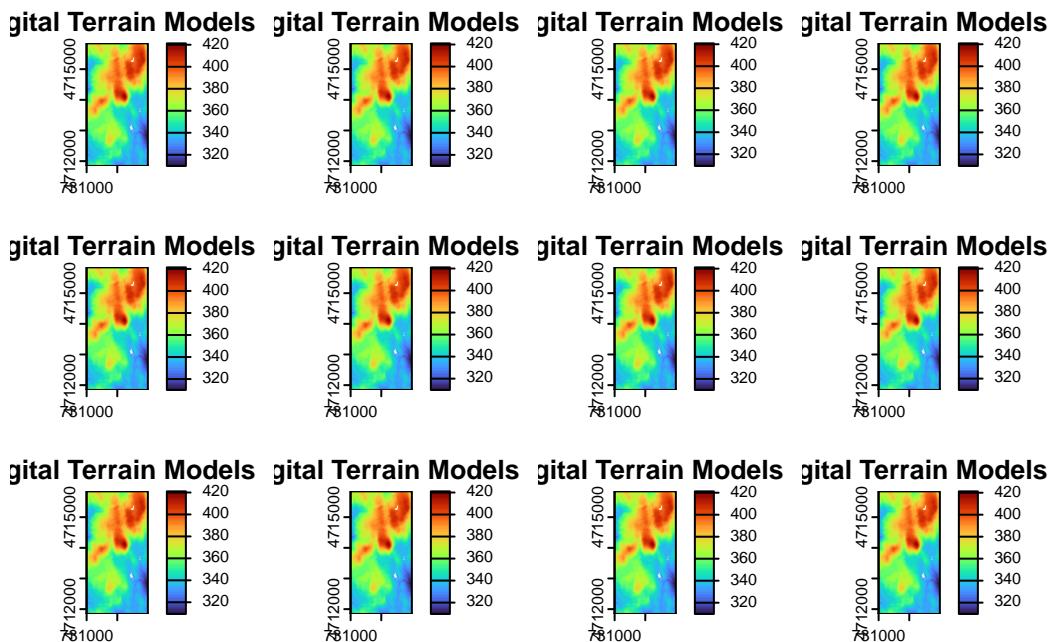
There are **12 scans** available:

- **1 scan** from 3DEP in winter 2013/2014
- **3 scans** from GLiHT (2012, 2017, 2021)
- **8 scans** from NEON

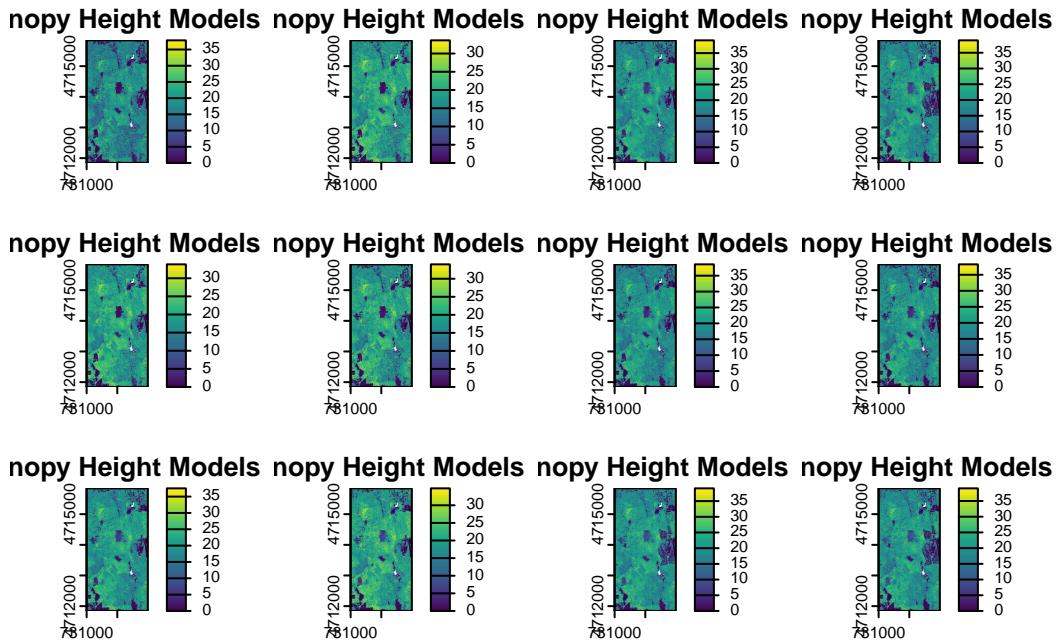
2.1 Loading Pre-processed Rasters

```
# read in rasters
dtms = rast("products/dtms_8km2_12scans.tif")
dsms = rast("products/dsms_8km2_12scans.tif")
chms = rast("products/chms_8km2_12scans.tif")

# plot DTMs
terra::plot(dtms, col = viridis(100, option = "turbo"), main = "Digital Terrain Models")
```



```
# plot CHMs
terra::plot(chms, col = viridis(100), main = "Canopy Height Models")
```

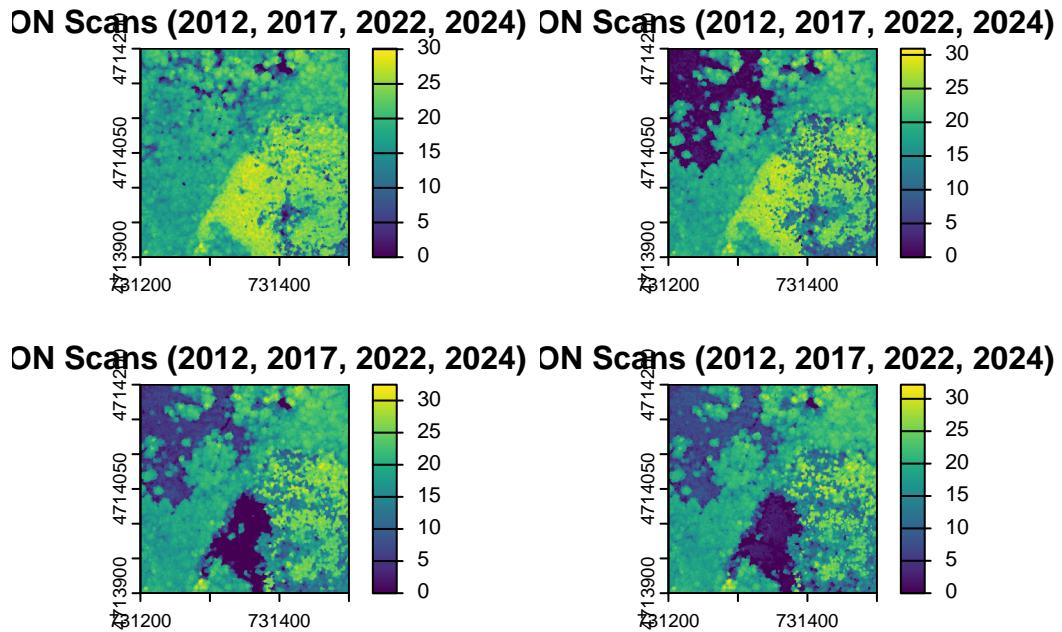


We can also zoom in to examine specific areas:

```
# zoom in
ext_full = ext(chms)
ext_reduced = ext(731200, 731500, 4713900, 4714200)
chms_reduced = crop(chms, ext_reduced)

# select only a few NEON scans
chms_NEON = subset(chms_reduced, c("NEON_2012", "NEON_2017", "NEON_2022", "NEON_2024"))

# plot these scans
terra::plot(chms_NEON, col = viridis(100), main = "NEON Scans (2012, 2017, 2022, 2024)")
```



2.2 Summary Statistics

We can calculate summary statistics for the DTMs:

```
# calculate summary statistics
# mean elevation from each scan
means_dtms = global(dtms, "mean", na.rm = TRUE)
print(means_dtms)
```

	mean
3DEP_2013.5	361.9564
GLiHT_2012	361.8035
GLiHT_2017	361.4712
GLiHT_2021	361.4877
NEON_2012	362.1649
NEON_2014	361.9043
NEON_2016	362.0542
NEON_2017	361.7781
NEON_2018	361.7946
NEON_2019	361.7137
NEON_2022	361.8275
NEON_2024	361.9200

We can also calculate summary statistics for the CHMs:

```
# mean canopy height from each scan
means_chms = global(chms, "mean", na.rm = TRUE)
print(means_chms)
```

	mean
3DEP_2013.5	16.15074
GLiHT_2012	17.54354
GLiHT_2017	18.31373
GLiHT_2021	17.89633
NEON_2012	17.69256
NEON_2014	17.08554
NEON_2016	17.74070
NEON_2017	17.68280
NEON_2018	18.20092
NEON_2019	18.32963
NEON_2022	18.36571
NEON_2024	18.11678

2.3 Temporal Comparison

We notice that there is some variation. Luckily (and this is rare), we have acquisitions from different institutions over similar time periods, so we can compare changes in time.

```
# calculate changes in time both for GLiHT and NEON canopy height estimates from 2012 to 2017
means_chms_dt = data.table(flight = rownames(means_chms), hmean = round(means_chms$mean, 2))
print(means_chms_dt[flight %like% "2017" | flight %like% "2012"])
```

	flight	hmean
1:	GLiHT_2012	17.54
2:	GLiHT_2017	18.31
3:	NEON_2012	17.69
4:	NEON_2017	17.68

```
deltaheight_GLiHT = as.numeric(means_chms_dt[flight == "GLiHT_2017", "hmean"] -
                                means_chms_dt[flight == "GLiHT_2012", "hmean"])
deltaheight_NEON = as.numeric(means_chms_dt[flight == "NEON_2017", "hmean"] -
                                means_chms_dt[flight == "NEON_2012", "hmean"])
```

```
cat("GLiHT height change 2012-2017:", deltaheight_GLiHT, "m\n")
```

GLiHT height change 2012-2017: 0.77 m

```
cat("NEON height change 2012-2017:", deltaheight_NEON, "m\n")
```

NEON height change 2012-2017: -0.01 m

⚠ Interesting Observation

Something is a bit odd. For example, if we look at GLiHT scans, then height has increased from 2012 to 2017, but using NEON, it has remained stable. Let's investigate further by examining the digital terrain models.

```
# calculate changes in time both for GLiHT and NEON from 2012 to 2017, now for the digital terrain models
```

```
means_dtms_dt = data.table(flight = rownames(means_dtms), hmean = round(means_dtms$mean, 2))
```

```
print(means_dtms_dt[flight %like% "2017" | flight %like% "2012"])
```

```
      flight  hmean  
      <char>  <num>  
1: GLiHT_2012 361.80  
2: GLiHT_2017 361.47  
3: NEON_2012 362.16  
4: NEON_2017 361.78
```

```
deltaheight_GLiHT_dtm = as.numeric(means_dtms_dt[flight == "GLiHT_2017", "hmean"] -  
                                     means_dtms_dt[flight == "GLiHT_2012", "hmean"])  
deltaheight_NEON_dtm = as.numeric(means_dtms_dt[flight == "NEON_2017", "hmean"] -  
                                     means_dtms_dt[flight == "NEON_2012", "hmean"])
```

```
cat("GLiHT DTM change 2012-2017:", deltaheight_GLiHT_dtm, "m\n")
```

GLiHT DTM change 2012-2017: -0.33 m

```
cat("NEON DTM change 2012-2017:", deltaheight_NEON_dtm, "m\n")
```

NEON DTM change 2012-2017: -0.38 m

DTM Accuracy Considerations

Both show a similar change in topography, so that does not explain the canopy height changes. But we can also see that, in practice, we have a change of ~30 cm in absolute elevation values in time (which may be an artefact), but also a ~30 cm difference for the scans in the same years. This is larger than the often claimed accuracy of 10 cm for ALS data.

Note that we could also directly calculate the difference between rasters:

```
# spatial plotting of height differences between two scans
hdiff_2012_2017 = chms$NEON_2017 - chms$NEON_2012
terra::plot(hdiff_2012_2017, col = viridis(100, option = "turbo"),
            main = "Height difference (m): NEON 2017 - NEON 2012")
```

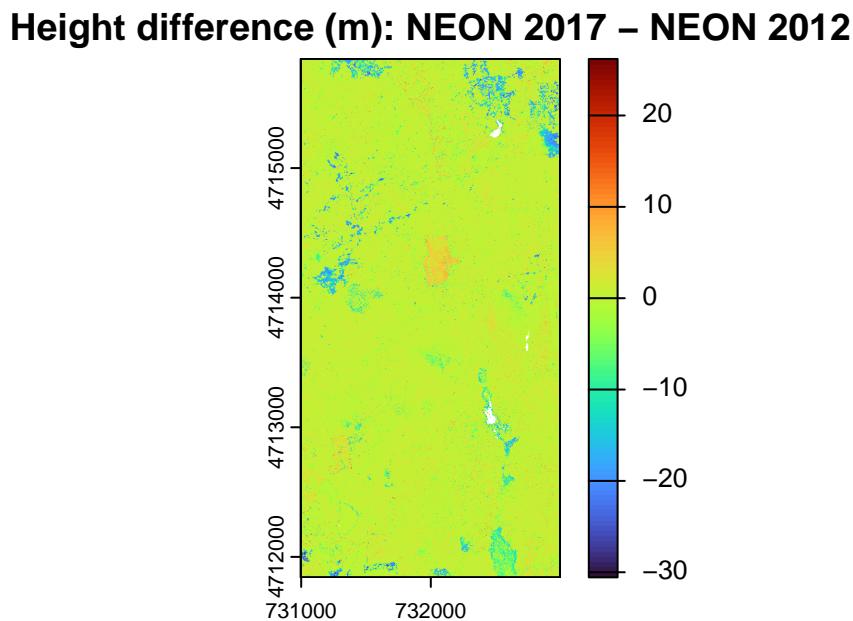


Figure 1: Spatial distribution of height differences between 2012 and 2017

3 Guiding Question: How Does Forest Structure Change at Harvard Forest?

Motivation:

- Relevance for disturbance monitoring
- Relevance for carbon monitoring

Research Questions:

1. What is the typical annual height growth rate of young trees at Harvard Forest, in meters per year?
2. Is Harvard Forest growing in height, balanced, or is it losing height?
3. What percentage of Harvard Forest experiences disturbance?
4. What are the typical noise levels between ALS acquisitions? (compare GLiHT and NEON)

4 Approaches

4.1 Regrowth Analysis

We define young forests and assess height growth.

```
# height growth with NEON
# define a threshold for what's considered disturbed
threshold_growing = 2

# define areas that were below threshold previously and are above it 5 years later
# and assign the 2012 canopy height to them
neon_2012regrowing = ifel(chms$NEON_2012 <= threshold_growing &
                           chms$NEON_2017 >= threshold_growing,
                           chms$NEON_2012, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$NEON_2017 - neon_2012regrowing
hdiff_regrowing_peryear = hdiff_regrowing / 5

# calculate statistics
cat("Mean annual growth:", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE)$mean, "m/year")
```

Mean annual growth: 0.9349838 m/year

```
cat("Max annual growth:", global(hdiff_regrowing_peryear, "max", na.rm = TRUE)$max, "m/year")
```

Max annual growth: 5.800006 m/year

```
# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = NEON_2017), alpha = 0.5, fill = "red", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency")
```

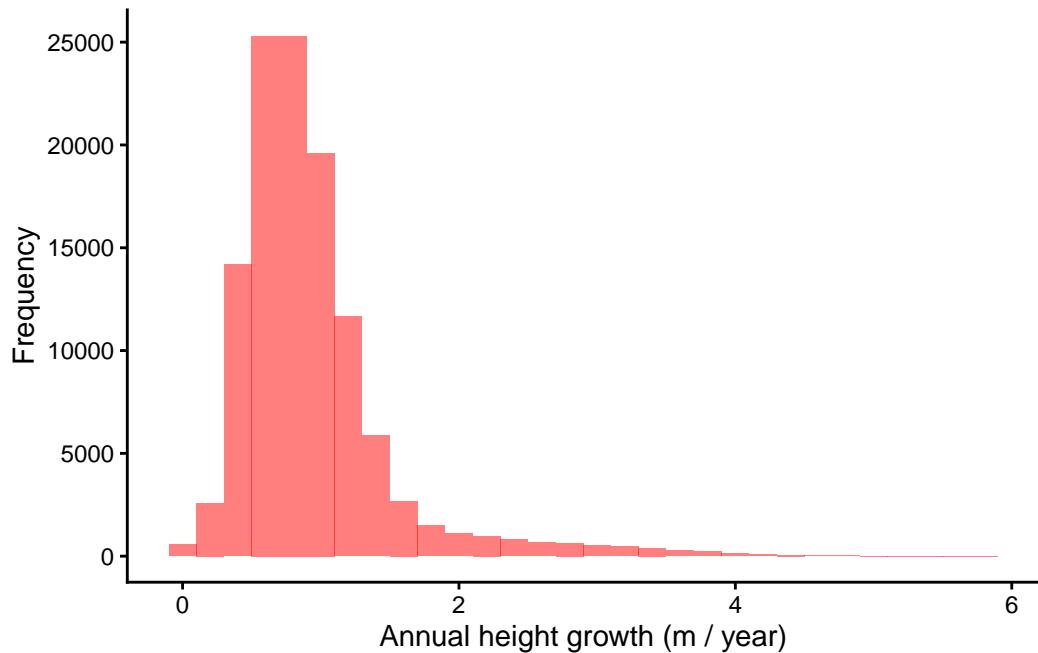


Figure 2: Annual height growth for areas initially below 2m

Now what if we take a different threshold?

```
# height growth with NEON
# define a threshold for what's considered disturbed
threshold_growing = 5

# define areas that were below threshold previously and are above it 5 years later
neon_2012regrowing = ifel(chms$NEON_2012 <= threshold_growing &
                           chms$NEON_2017 >= threshold_growing,
                           chms$NEON_2012, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$NEON_2017 - neon_2012regrowing
```

```

hdiff_regrowing_peryear = hdiff_regrowing / 5

# calculate statistics
cat("Mean annual growth (5m threshold):", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE))

Mean annual growth (5m threshold): 1.129185 m/year

cat("Max annual growth (5m threshold):", global(hdiff_regrowing_peryear, "max", na.rm = TRUE))

Max annual growth (5m threshold): 5.800006 m/year

# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = NEON_2017), alpha = 0.5, fill = "red", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency")

```

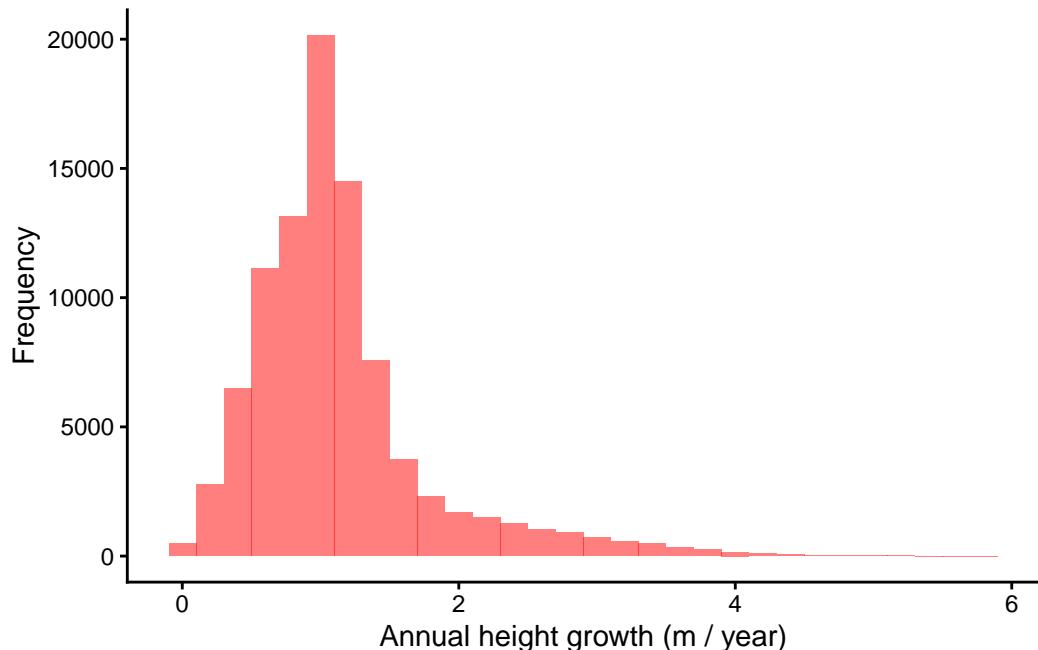


Figure 3: Annual height growth for areas initially below 5m

What if we calculate it with GLiHT?

```

# height growth with GLiHT
# define a threshold for what's considered disturbed
threshold_growing = 5

# define areas that were below threshold previously and are above it 5 years later
gliht_2012regrowing = ifel(chms$GLiHT_2012 <= threshold_growing &
                           chms$GLiHT_2017 >= threshold_growing,
                           chms$GLiHT_2012, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$GLiHT_2017 - gliht_2012regrowing
hdiff_regrowing_peryear = hdiff_regrowing / 5

# calculate statistics
cat("Mean annual growth (GLiHT):", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE)$mean)

```

Mean annual growth (GLiHT): 1.492952 m/year

```
cat("Max annual growth (GLiHT):", global(hdiff_regrowing_peryear, "max", na.rm = TRUE)$max,
```

Max annual growth (GLiHT): 6.152002 m/year

```

# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = GLiHT_2017), alpha = 0.5, fill = "red", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency")

```

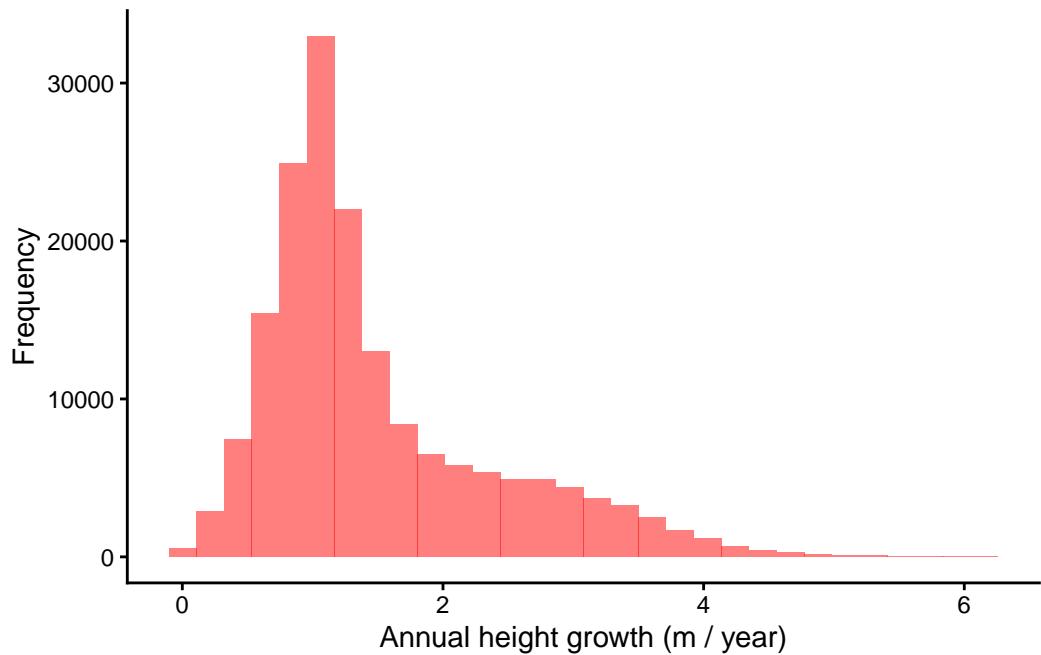


Figure 4: Annual height growth using GLiHT data

What if we calculate it with GLiHT, and over a longer time period?

```
# height growth with GLiHT over multiple time periods
threshold_growing = 5

# define areas that were below threshold in 2012
gliht_2012regrowing = ifel(chms$GLiHT_2012 <= threshold_growing &
                           chms$GLiHT_2017 >= threshold_growing,
                           chms$GLiHT_2012, NA)

# now calculate the regrowth with both 2017 and 2021 scans
hdiff_regrowing = subset(chms, c("GLiHT_2017", "GLiHT_2021")) - gliht_2012regrowing
hdiff_regrowing_peryear = hdiff_regrowing / c(5, 9)

# calculate statistics
cat("Mean annual growth 2012-2017:", global(hdiff_regrowing_peryear$GLiHT_2017, "mean", na.rm = TRUE))
```

Mean annual growth 2012-2017: 1.492952 m/year

```
cat("Mean annual growth 2012-2021:", global(hdiff_regrowing_peryear$GLiHT_2021, "mean", na.rm = TRUE))
```

```
Mean annual growth 2012-2021: 0.8637589 m/year
```

```
# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = GLiHT_2017), alpha = 0.5, fill = "red", bins = 30) +
  geom_histogram(aes(x = GLiHT_2021), alpha = 0.5, fill = "blue", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency") +
  labs(subtitle = "Red = 2012-2017 (5 years), Blue = 2012-2021 (9 years)")
```

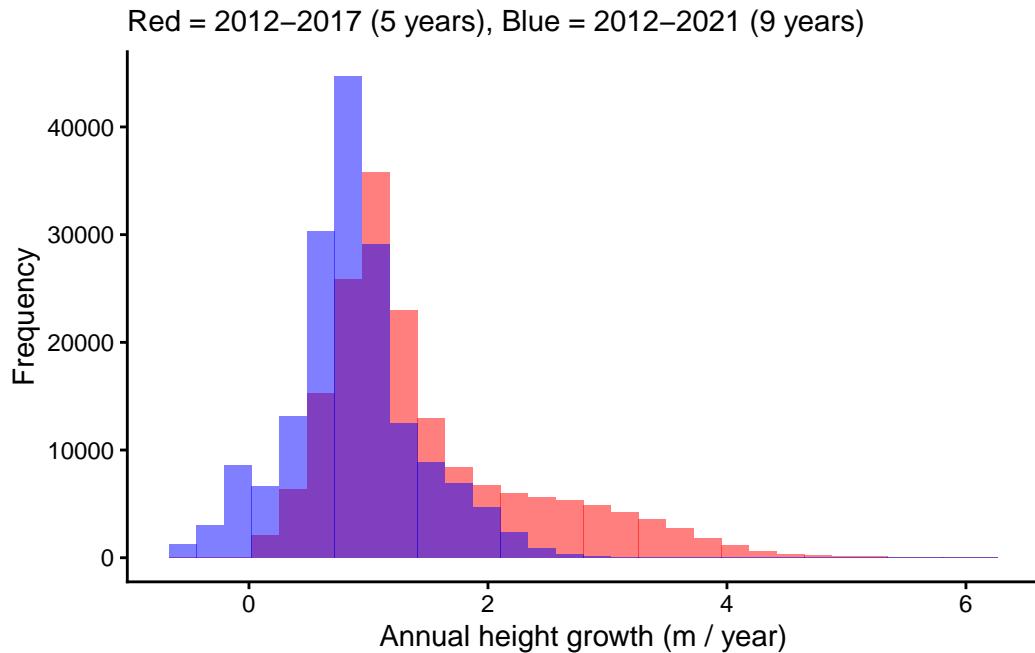


Figure 5: Comparison of growth rates over 5 and 9 years

4.2 One-Year Changes

What about analyzing one-year changes?

```

# height growth with NEON, but only for one year
threshold_growing = 2

# define areas that were below threshold previously and are above it 1 year later
neon_2016regrowing = ifel(chms$NEON_2016 <= threshold_growing &
                           chms$NEON_2017 >= threshold_growing,
                           chms$NEON_2016, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$NEON_2017 - neon_2016regrowing
hdiff_regrowing_peryear = hdiff_regrowing

# calculate statistics
cat("Mean annual growth 2016-2017:", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE)$m

```

Mean annual growth 2016-2017: 4.31011 m/year

```
cat("Max annual growth 2016-2017:", global(hdiff_regrowing_peryear, "max", na.rm = TRUE)$max
```

Max annual growth 2016-2017: 28.56 m/year

```

# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = NEON_2017), alpha = 0.5, fill = "red", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency")

```

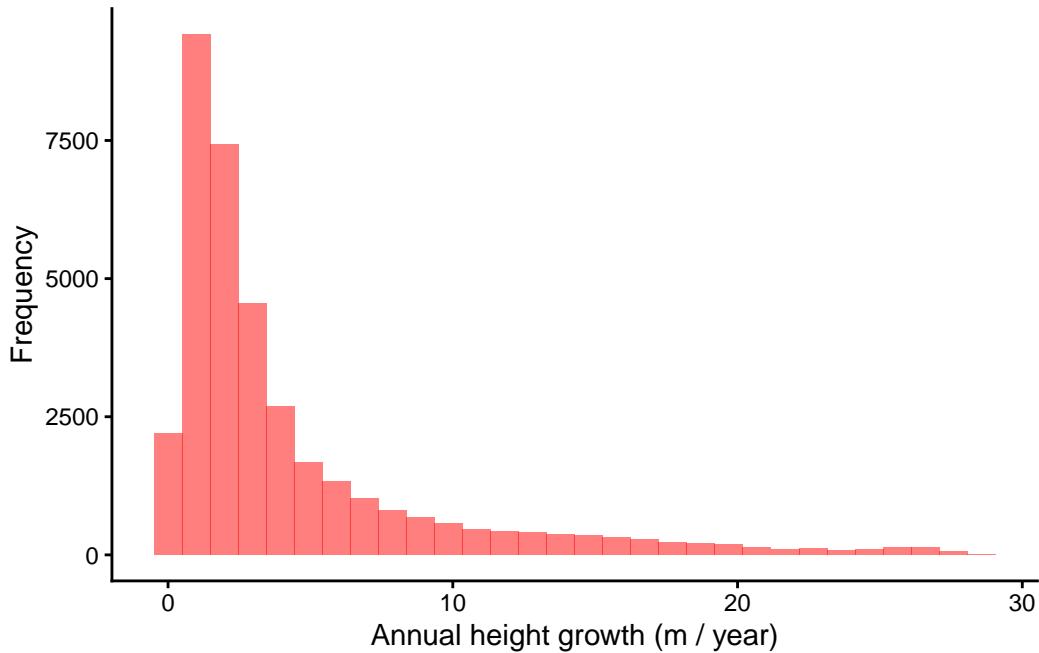


Figure 6: Annual height growth between 2016 and 2017

Now what about the following year?

```
# height growth with NEON
threshold_growing = 2

# define areas that were below threshold previously and are above it 1 year later
neon_2017regrowing = ifel(chms$NEON_2017 <= threshold_growing &
                           chms$NEON_2018 >= threshold_growing,
                           chms$NEON_2017, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$NEON_2018 - neon_2017regrowing
hdiff_regrowing_peryear = hdiff_regrowing

# calculate statistics
cat("Mean annual growth 2017-2018:", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE)$m
```

Mean annual growth 2017-2018: 6.049758 m/year

```
cat("Max annual growth 2017-2018:", global(hdiff_regrowing_peryear, "max", na.rm = TRUE)$max
```

Max annual growth 2017-2018: 32.54001 m/year

```
# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = NEON_2018), alpha = 0.5, fill = "red", bins = 30) +
  theme_classic() +
  xlab("Annual height growth (m / year)") +
  ylab("Frequency")
```

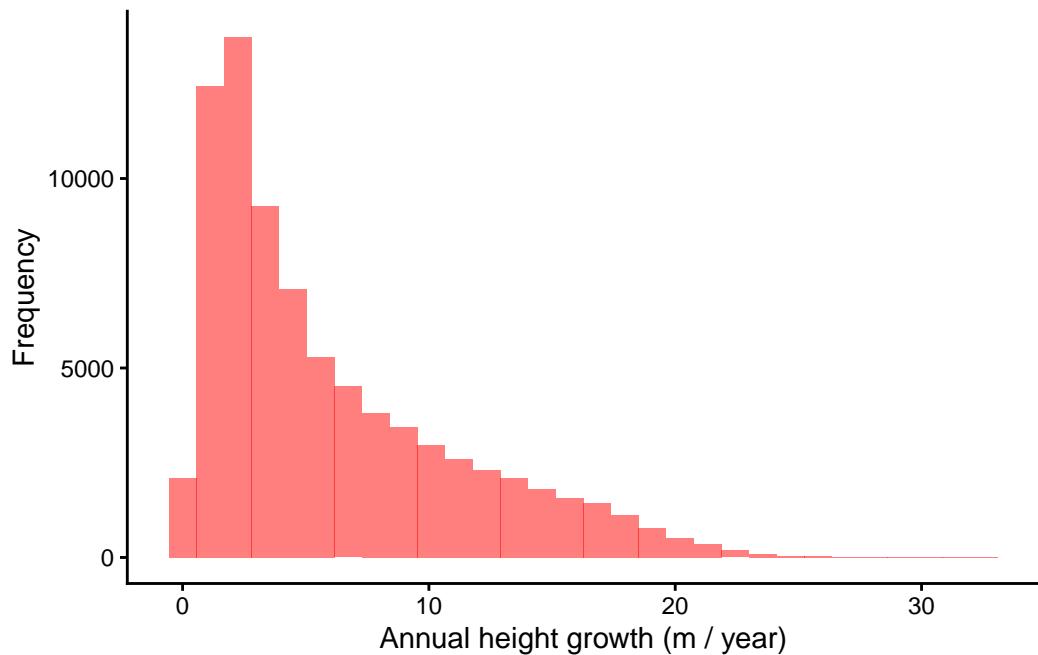


Figure 7: Annual height growth between 2017 and 2018

! Critical Questions

Is this true regrowth?

How could we better describe this?

Consider:

- Are we measuring actual tree growth or just detection artifacts?
- Could lateral crown expansion be confounding our measurements?

- What role do measurement uncertainties play in these estimates?

4.3 Addressing Lateral Ingrowth

One idea: find areas without lateral ingrowth by removing buffer areas around low vegetation.

```
# slightly adjusted procedure
# set neon_2017regrowing to 1 or NA to create a mask
threshold_growing = 5
neon_2017regrowing = ifel(chms$NEON_2017 <= threshold_growing &
                           chms$NEON_2018 >= threshold_growing,
                           1, NA)
terra::plot(neon_2017regrowing, col = "black", main = "Areas with regrowth (before filtering")
```

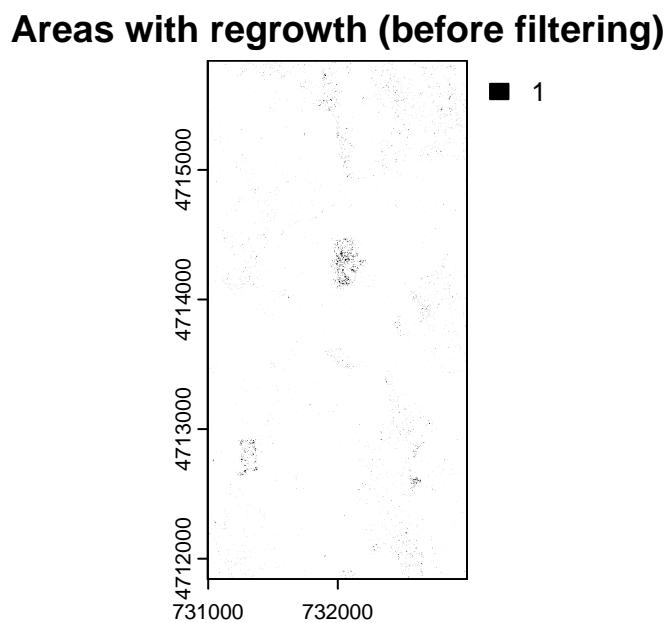


Figure 8: Height growth after removing edge effects

```
# now remove buffer areas around low areas, by using a focal filter
# that sets everything to NA at the borders
neon_2017regrowing = focal(neon_2017regrowing, w = 3, "mean", na.rm = FALSE) # 3 pixel window
terra::plot(neon_2017regrowing, col = "black", main = "Areas with regrowth (after filtering")
```

Areas with regrowth (after filtering)

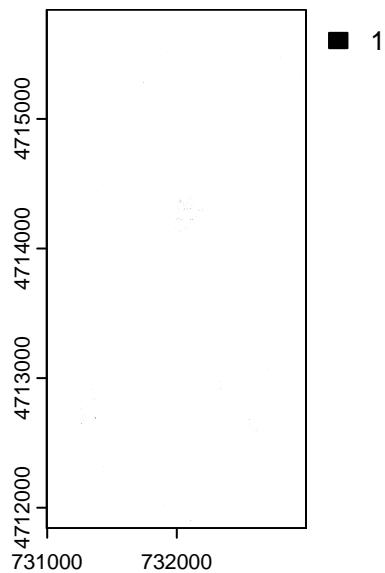


Figure 9: Height growth after removing edge effects

```
# fill in height
neon_2017regrowing = ifel(!is.na(neon_2017regrowing), chms$NEON_2017, NA)

# now calculate the regrowth with both layers
hdiff_regrowing = chms$NEON_2018 - neon_2017regrowing
hdiff_regrowing_peryear = hdiff_regrowing

# calculate statistics
cat("Mean annual growth (filtered):", global(hdiff_regrowing_peryear, "mean", na.rm = TRUE)$mean)

Mean annual growth (filtered): 3.812533 m/year

cat("Max annual growth (filtered):", global(hdiff_regrowing_peryear, "max", na.rm = TRUE)$max)

Max annual growth (filtered): 22.51999 m/year

# plot a histogram
ggplot(data = as.data.table(hdiff_regrowing_peryear)) +
  geom_histogram(aes(x = NEON_2018), alpha = 0.5, fill = "red", bins = 30) +
```

```

theme_classic() +
xlab("Annual height growth (m / year)") +
ylab("Frequency") +
labs(subtitle = "After removing edge effects")

```

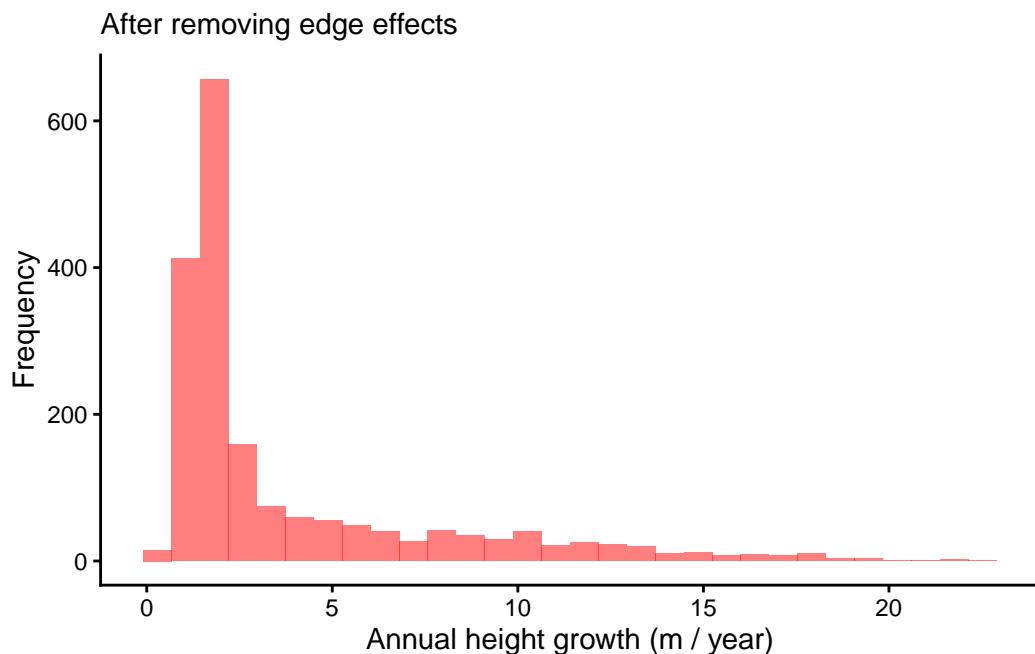


Figure 10: Height growth after removing edge effects

i Discussion Points

What are we missing? What could be the issue?

Think about:

- Edge detection artifacts
- Sensor noise and uncertainties
- Seasonal effects (leaf-on vs leaf-off)
- Co-registration errors between scans
- Natural variability in growth rates

4.4 Disturbance Analysis

To be developed in class discussion

4.5 Tree-Based Changes

To be developed in class discussion

5 Summary and Discussion

6 Key Findings

Growth Rate Estimates:

- Threshold selection significantly affects results
- One-year vs multi-year comparisons show different patterns
- GLiHT and NEON data produce different estimates
- Edge effects and lateral ingrowth confound measurements

Measurement Uncertainties:

- DTM differences between scans: ~30 cm
- Larger than commonly cited 10 cm accuracy
- Systematic biases between different sensors
- Temporal artifacts in elevation measurements

7 Methodological Challenges

Sources of Uncertainty:

- **Sensor differences:** GLiHT vs NEON vs 3DEP
- **Temporal mismatches:** Different acquisition dates and times
- **Phenological effects:** Leaf-on vs leaf-off conditions
- **Co-registration errors:** Alignment between scans
- **Threshold sensitivity:** Growth detection depends heavily on cutoff values

Confounding Factors:

- Lateral crown expansion vs vertical growth
- Edge effects at transition zones
- Detection vs actual growth
- Noise vs signal in one-year changes

8 Best Practices

For Growth Analysis:

- Use multiple time intervals to validate patterns
- Compare results from different sensors
- Account for edge effects and lateral expansion
- Consider measurement uncertainties in interpretation
- Validate with field data when possible

For Disturbance Detection:

- Define clear thresholds based on ecology and error
- Account for sensor noise levels
- Use spatial context (focal operations)
- Combine multiple temporal snapshots
- Ground-truth selected areas

9 Research Questions

Consider these questions for your analysis:

1. **What is a realistic annual growth rate for this forest type?**
 - Compare with field studies
 - Consider forest age and species composition
 - Account for measurement uncertainties
2. **How can we separate growth from measurement artifacts?**
 - Use multiple temporal baselines
 - Compare sensors in the same year
 - Apply spatial filters
3. **What level of change is detectable given sensor noise?**
 - Estimate minimum detectable change
 - Consider signal-to-noise ratios
 - Validate against independent data

10 Exercises and Extensions

💡 Try These Analyses

1. **Compare different time intervals:**
 - Calculate growth rates for 1, 5, and 10-year periods
 - How do estimates change with time interval?
2. **Test threshold sensitivity:**
 - Try thresholds from 1-10 m
 - Plot how results vary
 - What threshold seems most reasonable?
3. **Sensor comparison:**
 - Compare GLiHT, NEON, and 3DEP for the same time period
 - Quantify systematic differences
 - Which is most reliable?
4. **Spatial analysis:**
 - Map areas of growth vs decline
 - Identify disturbance hotspots
 - Relate to topography or other factors
5. **Validation:**
 - Compare with published growth rates for similar forests
 - Identify field plots for ground-truthing
 - Assess realism of your estimates

11 Conclusion

This tutorial demonstrates the challenges and opportunities of using multi-temporal ALS data to monitor forest structural changes. While ALS provides unprecedented spatial coverage and detail, careful attention to measurement uncertainties, sensor differences, and methodological artifacts is essential for robust conclusions.

Key Takeaways:

- ALS can detect forest height changes, but with significant uncertainties
- Sensor choice, threshold selection, and temporal baseline all strongly affect results
- Edge effects and lateral expansion confound vertical growth estimates

- Multiple lines of evidence and validation are essential
- Critical thinking about data quality is as important as analytical technique

Future Directions:

- Integration with other data sources (optical imagery, field plots)
 - Machine learning for change detection
 - Improved co-registration methods
 - Individual tree tracking across time
 - Uncertainty quantification frameworks
-

*This tutorial was designed to encourage critical thinking about ALS-based change detection.
The emphasis is on understanding limitations and uncertainties, not just applying methods.*

