Hills Forest ReMapping

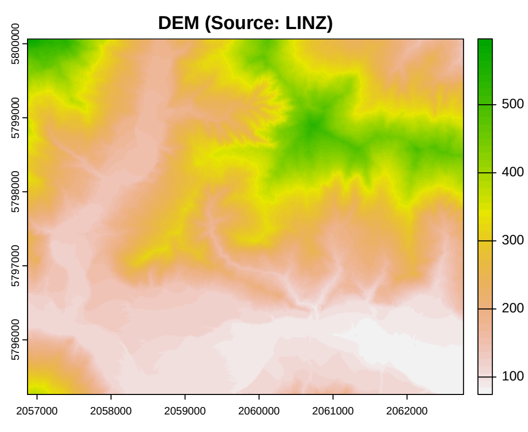
Summit-GIS

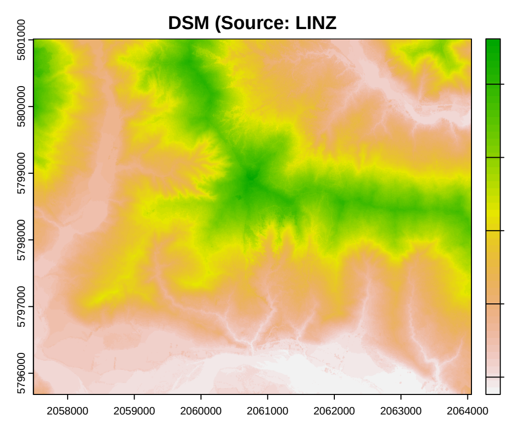
17/08/2023

Contents

Action:

Some notes on the data processing steps and mapping outputs completed in the remapping exercise of the Hills Forest Area. To map changes occurring between original stands records and newly LiDAR-derived attributes, two variables were derived: 'Stocking Density (stem/ha)' & 'Age Class'. Age class was drawn approximately from a canopy height model, which was computed using the following functions. Using DEM and DSM datasets sourced from the LINZ website, a stem map was derived by applying a variable window algorithm and a default taper function. With the stem map feature, two rasters were calculated showing a canopy height model and stocking density per hectare. This allowed some rough comparisons with values at StandID level using Establishment year and Stocking in stand records provided. Scripts, chunk outputs, and markdown reports were stored at the repository 'hills_forest_stocking': https://github.com/seamusrobertmurphy/hills_forest_stocking





1. Inputs: LiDAR Projection

No custom projection found with original LiDAR files. Instead everything was re-assigned or reporojected to EPSG2193/NZGD2000. Chunk below allow for bulk loading and merging of LiDAR tiles into raster mosaic. Sinks were filled using inverse watershed algorithm by Wang & Liu (2020) with following settings.

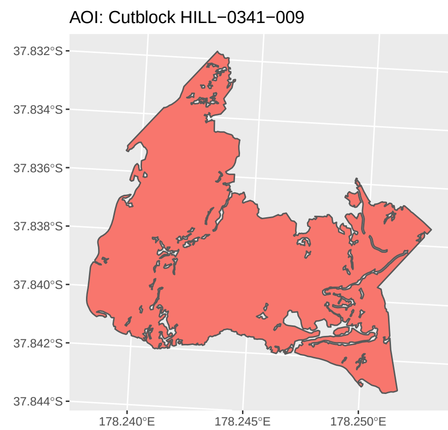
```
# Merge chunks
filez_dem = list.files("~/Desktop/Summit_Forestry/dem", full.names = T, all.files = FALSE, pattern = '.t
filez_dsm = list.files("~/Desktop/Summit_Forestry/dsm", full.names = T, all.files = FALSE, pattern = '.t
dem_raster_list <- lapply(filez_dem, raster)
dsm_raster_list <- lapply(filez_dsm, raster)
dem_raster = do.call(merge, c(dem_raster_list, tolerance = 1))
dsm_raster = do.call(merge, c(dsm_raster_list, tolerance = 1))
writeRaster(dem_raster, filename = "~/Desktop/Summit_Forestry/dem/dem_raster.tif", overwrite=TRUE)
writeRaster(dsm_raster, filename = "~/Desktop/Summit_Forestry/dsm/dsm_raster.tif", overwrite=TRUE)

# Rasterize and fill sinks
fill_sinks(dem = "dem_mosaic.tif", out = "dem_fill.tif", size = 1, overwrite = TRUE)
fill_sinks(dsm = "dsm_mosaic.tif", out = "dsm_fill.tif", size = 1, overwrite = TRUE)
writeRaster(dem_fill, filename = "~/Desktop/Summit_Forestry/dem/dem_filled.tif", overwrite=TRUE)
writeRaster(dsm_fill, filename = "~/Desktop/Summit_Forestry/dsm/dsm_filled.tif", overwrite=TRUE)

# Assign EPGS2193/NZGD2000
dem_rast = terra::rast(dem_filled)
dsm_rast = terra::rast(dsm_filled)
terra::crs(dem_rast) =  "epsg:2193"
terra::crs(dsm_rast) =  "epsg:2193"
terra::plot(dem_rast, main='DEM (Source: LINZ)')
terra::plot(dsm_rast, main='DSM (Source: LINZ')
```

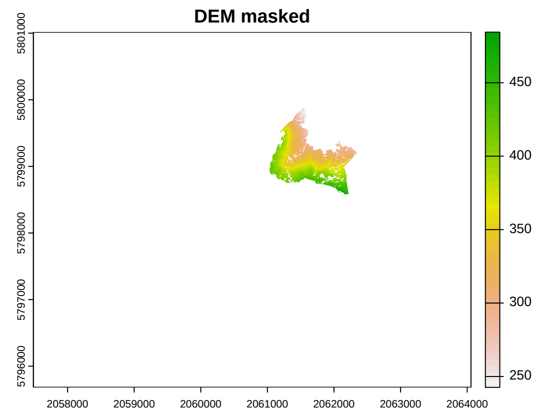
2. Inputs: AOI & Masking layers

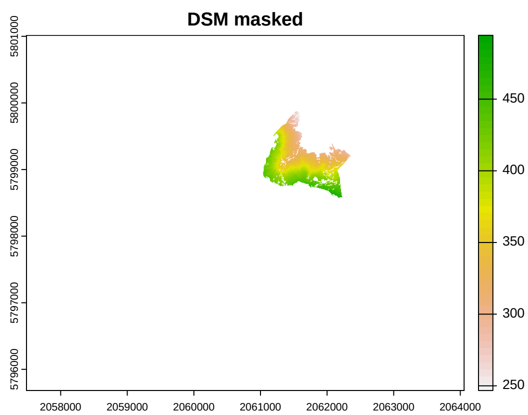
Further work needed here on filtering out non-stand areas such as roads and biodiversity features. However, 1.5m floor threshold is later inputted to the stem detection algorithm, which cancels out a lot of non-canopy areas automatically. Final 10m and 20m resolution rasters show some of this feature filtering. AOI filter here was used for visualization purposes only, showing results for single cutblock HILL-0341-009.

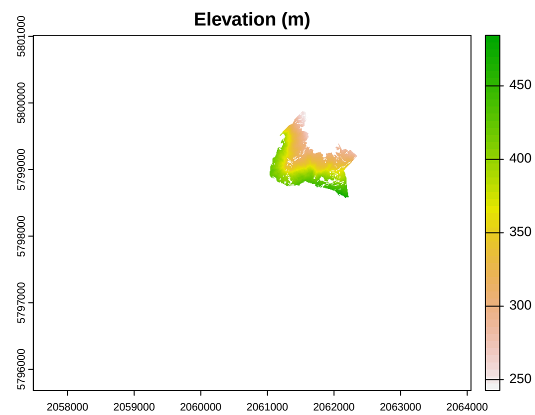
```
# Derive mask from single cutblock shapefile: HILL-0341-009
mask_sf = sf::read_sf("~/Desktop/Summit_Forestry/stands/HILL-0341-009.shp")
#mask_sf = sf::read_sf("~/Desktop/Summit_Forestry/stands/hills_forest_cutblocks.shp")
mask_rast = rasterize(vect(mask_sf), dem_rast, touches = TRUE)
terra::crs(mask_rast) =  "epsg:2193"
ggplot(mask_sf) + geom_sf(aes(fill = 'red'), show.legend = FALSE) +
  ggtitle("AOI: Cutblock HILL-0341-009")

# Align & apply mask
mask_rast = terra::resample(mask_rast, dsm_rast, method="near")
dem_rast = terra::resample(dem_rast, dsm_rast, method="near")
dem_masked = mask(dem_rast, mask_rast, inverse=FALSE)
dsm_masked = mask(dsm_rast, mask_rast, inverse=FALSE)
plot(dem_masked, main="DEM masked")
plot(dsm_masked, main = "DSM masked")
```

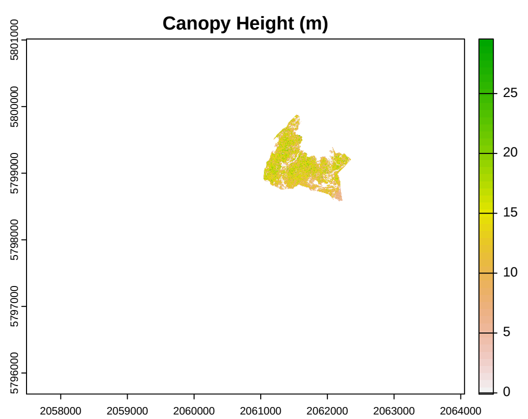
```
# Derive CHM & save all rasters
elev = dem_masked
chm = dsm_masked - dem_masked
elev_raster = raster::raster(elev)
chm_raster = raster::raster(chm)
writeRaster(elev_raster, filename = "~/Desktop/Summit_Forestry/dem/elev_raster.tif", overwrite=TRUE)
writeRaster(chm_raster, filename = "~/Desktop/Summit_Forestry/dsm/chm_raster.tif", overwrite=TRUE)
plot(elev, main="Elevation (m)")
plot(chm, main="Canopy Height (m)")
```







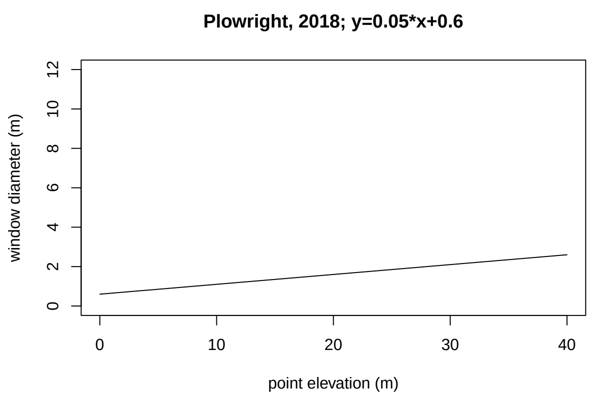




3. Inputs: Variable Window Function

We used Plowright's variable window function here as a temporary fix only. This ratio was developed from in the Kootenays looking at mixed-species and lodgepole pine stands here: https://cran.r-project.org/web/packages/ForestTools/vignettes/treetop_analysis.html. This needs replaceing with taper curves from local Radiata pine stands.

```
wf_plowright<-function(x){
  a=0.05
  b=0.6
  y<-a*x+b
  return(y)}
heights <- seq(0,40,0.5)
window_plowright <- wf_plowright(heights)
plot(heights, window_plowright, type = "l", ylim = c(0,12), xlab="point elevation (m)", ylab="window di
```



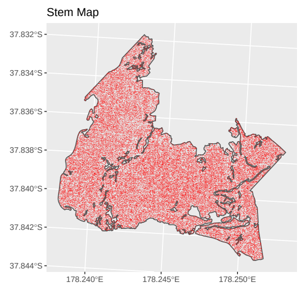
4. Outputs: 95% Canopy Height & Stem Count Layers

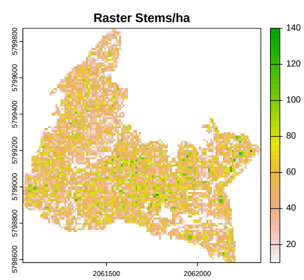
The ForestTools package was used to apply the variable window function, fitted with custom taper above and minimum floor threshold of 1.5 metres above ground. This produces point shapefile of tree tops that can be read in as sp dataframe. With more time available, it is recommended to validate tree top results either using the resulting TreeID attribute or by running a crown segmentation function. To capture estimates at the 95th percentile of tree height, a custom function 'custFuns' is derived below and inputted into the 'sp_summarise' command. Outputs are in raster format and resolution is set with grid units factored from current crs. Whatever the tree count at this resolution must be recaculated to the level of 100m sq before aggregating rasters. Yet, I've found the sp_summarize sometimes differs and is be more accurate over wider extents, not sure why.

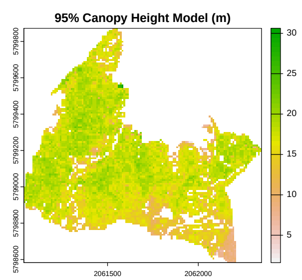
```
kernel <- matrix(1,3,3)
chm_raster = focal(chm, w = kernel, fun = median, na.rm = TRUE) %>% raster()
ttops_1.5mfloor_plowright = ForestTools::vwf(chm_raster, wf_plowright, 1.5)
writeOGR(ttops_1.5mfloor_plowright, "~/Desktop/Summit_Forestry/stands", "treetops_hills_009.shp", drive

quant95 <- function(x, ...)
  quantile(x, c(0.95), na.rm = TRUE)
custFuns <- list(quant95, max)
names(custFuns) <- c("95thQuantile", "Max")
```

```r
# Derive stems and canopy rasters and assign resolution (per hectare)
ttops_1.5mfloor_height <- ForestTools::sp_summarise(ttops_1.5mfloor_plowright, grid = 10, variables = "
stem_count_raster = ttops_1.5mfloor_height[["TreeCount"]]
chm_95height_raster = ttops_1.5mfloor_height[["height95thQuantile"]]
raster::writeRaster(chm_95height_raster, filename = "~/Desktop/Summit_Forestry/stands/chm_95height_10m_
raster::writeRaster(stem_count_raster, filename = "~/Desktop/Summit_Forestry/stands/stem_count_10m_allS

mypalette<-brewer.pal(8,"Greens")
stem_count_rast = terra::rast(stem_count_raster)
# Aggregate to stems per 100m^2 from 10m resolution
stem_count_ha = 10*stem_count_rast
raster::writeRaster(stem_count_ha, filename = "~/Desktop/Summit_Forestry/stands/stem_count_10m_ha.tif",
chm_95height_rast = terra::rast(chm_95height_raster)
stem_count_ha_sf = st_as_sf(ttops_1.5mfloor_plowright)
ggplot() +
  geom_sf(data=mask_sf) +
  geom_sf(data=stem_count_ha_sf, cex = 0.2, pch="+", col = 'red', lwd=1, alpha=1, show.legend = "point"
  ggtitle("Stem Map") +
  coord_sf(default_crs = sf::st_crs(2193))
#{plot(chm_95height_rast, col = mypalette, alpha=0.6, main="Stem Map over 95% CHM")
#plot(st_geometry(stem_count_ha_sf["treeID"]), cex = 0.2, pch="+", col = 'red', lwd=1, alpha=1, add=TRU
plot(stem_count_rast, main="Raster Stems/ha (10m Res)")
plot(chm_95height_rast, main="95% Canopy Height Model (m)")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/home/seamus/Desktop/Summit_Forestry/stands", layer: "treetops_hills_009.shp"
## with 34585 features
## It has 3 fields
```







5. Outputs: Site Maps

For comparison, four sets of maps were produced in pdf format and added to document below. First two maps were drawn for overview and access purposes. Maps 3 and 4 include original and LiDAR-derived data showing stocking density, stand age and canopy height.

1. Site Access Map
2. Site Recon Map
3. Stocking Density Maps
4. Age Class & Canopy Height Maps

All raster files were stored at 10m, 20m, 50m, and 100m resolutions and zipped in attachment with this report, along with spatial point dataframe of treetop locations in ESRI shapefile format.