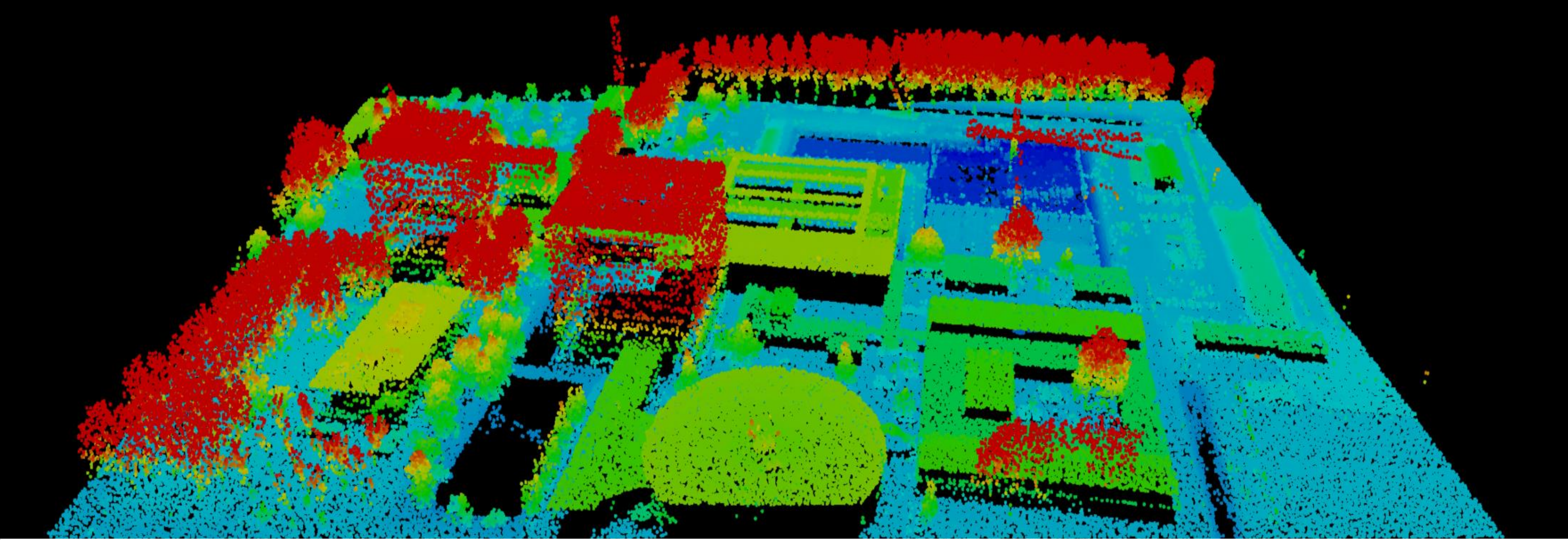


Processing and handling 3D point clouds using open-source software solutions

Zsófia Koma

IBED

Universiteit van Amsterdam

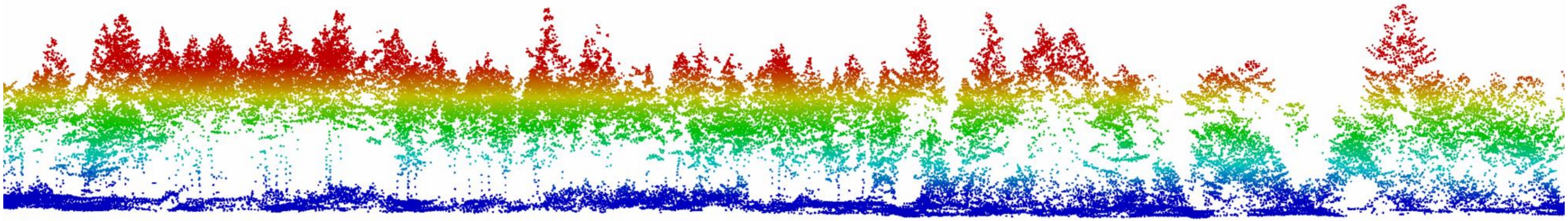
SustainScapes

Aarhus University

Background

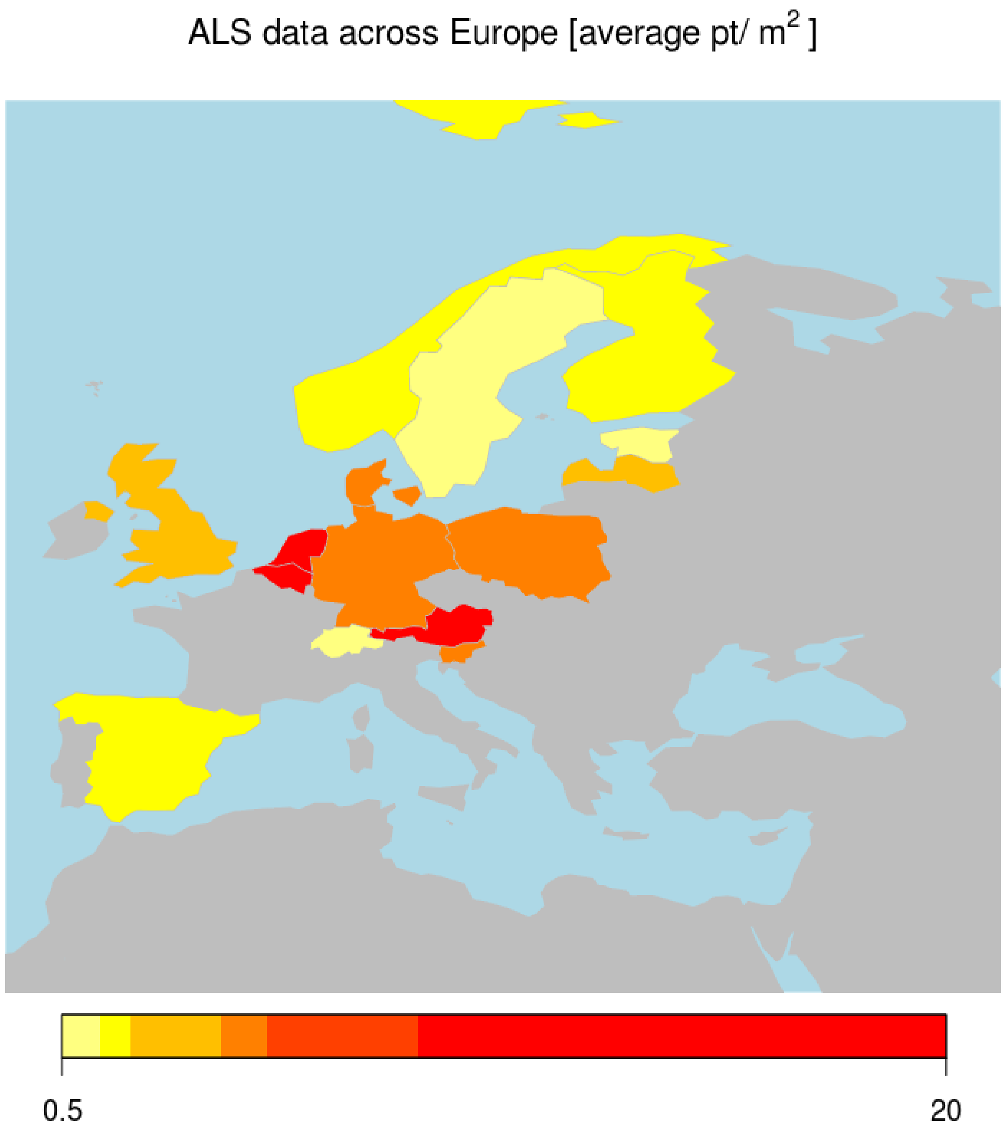
- Light Detection and Ranging (LiDAR) provides high resolution 3D point cloud dataset

- This type of dataset is useful for extracting high resolution information about terrain, buildings, vegetation structure



Background

- Availability of freely accessable ALS datasets is growing – country-wide ALS
- Important task to process, handle these country-wide ALS datasets



Software solutions

- Not long ago the community lacked open-source software solutions for process and handle 3D point clouds

Software solutions

- Now, the software landscape has been rapidly changed

Software solutions

- Now, the software landscape has been rapidly changed

LAStools

opals

TerraScan

pdal

GRASS GIS

QGIS

FUSION
Providing fast, efficient, and flexible access to LIDAR, IFSAR and terrain datasets
Robert J. McGaughey
Pacific Northwest Research Station

CloudCompare V2

lidR

LASERCHICKEN

build passing | code quality | coverage 86% | DOI 10.5281/zenodo.4035231 | docs passing

laspy/laspy
Laspy is a pythonic interface for reading/modifying/creating .LAS LIDAR files matching specification 1.0-1.4.
28 Contributors | 214 Used by | 257 Stars | 88 Forks

Introduction to LiDR

lidR

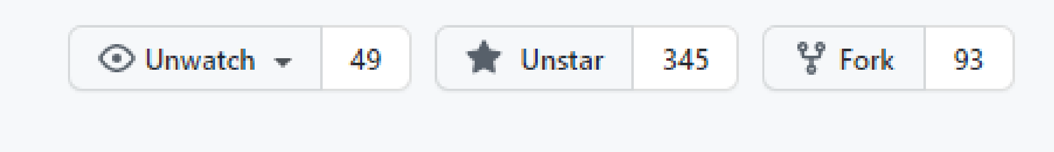
Licence GPL-3 | R-CMD-check failing | codecov unknown

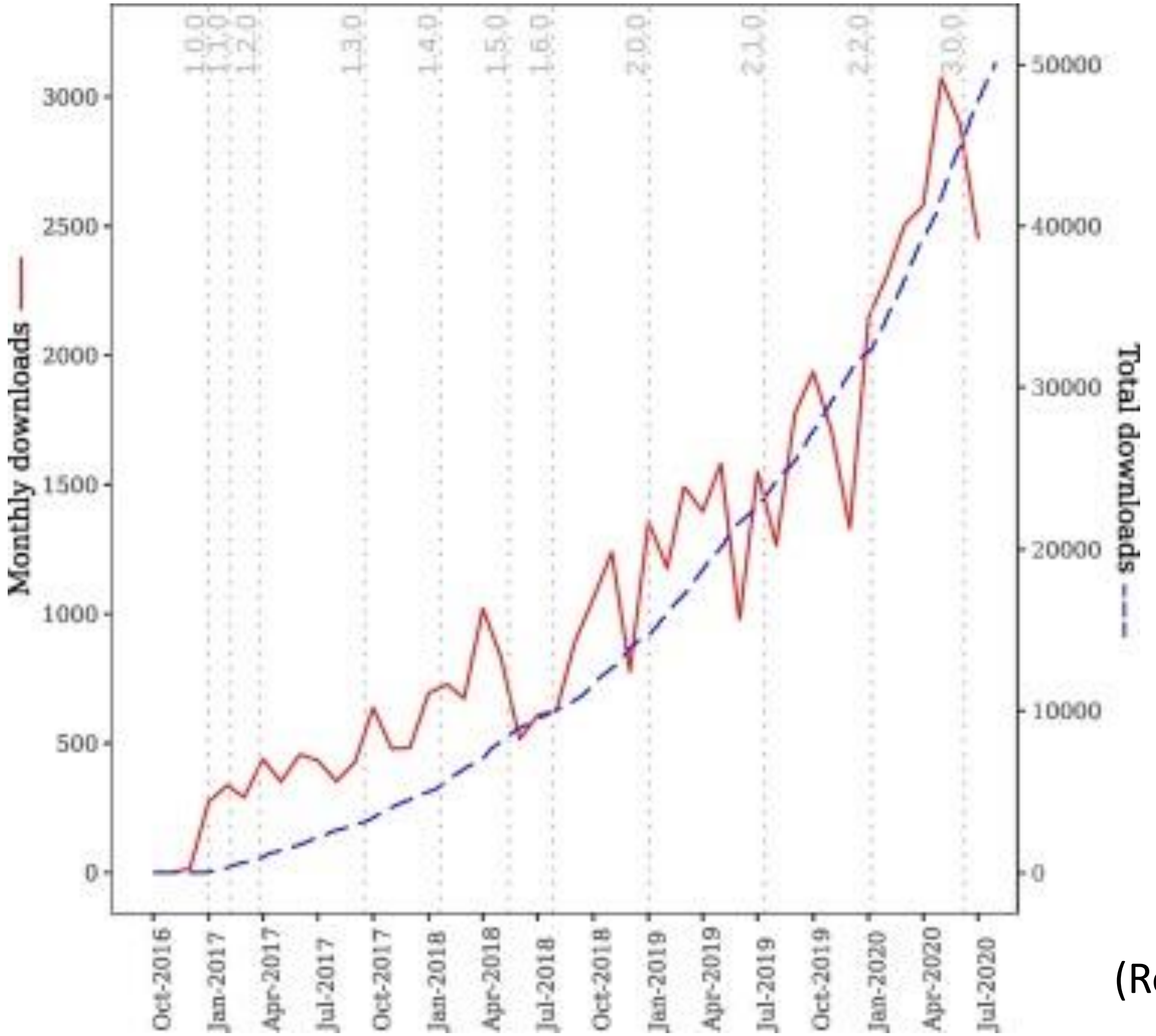
R package for Airborne LiDAR Data Manipulation and Visualization for Forestry Applications

The lidR package provides functions to read and write `.las` and `.laz` files, plot point clouds, compute metrics using an area-based approach, compute digital canopy models, thin LiDAR data, manage a collection of LAS/LAZ files, automatically extract ground inventories, process a collection of tiles using multicore processing, segment individual trees, classify points from geographic data, and provides other tools to manipulate LiDAR data in a research and development context.

📖 Read the book to get started with the lidR package. See changelogs on NEW.md

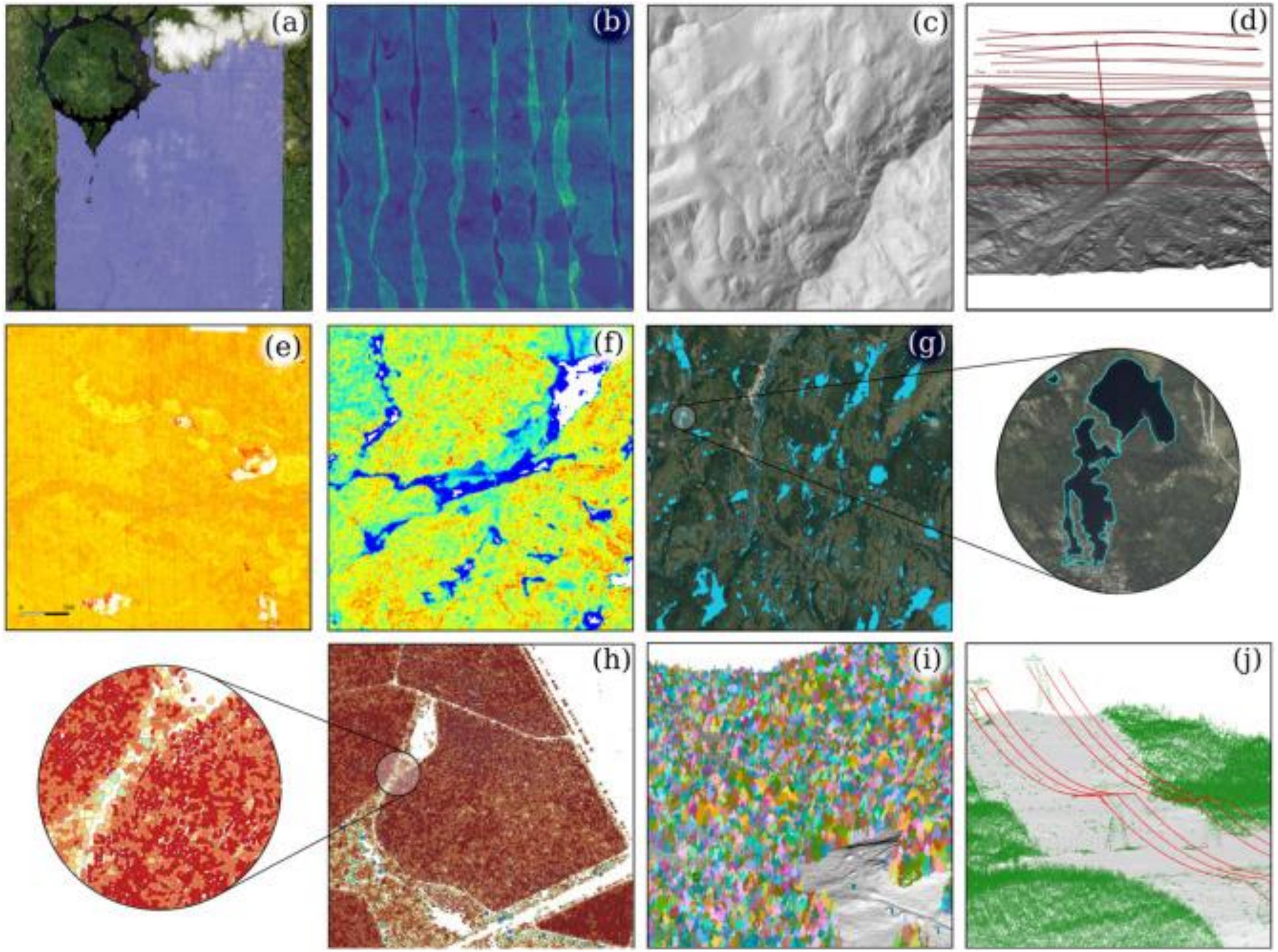
Introduction to LiDR





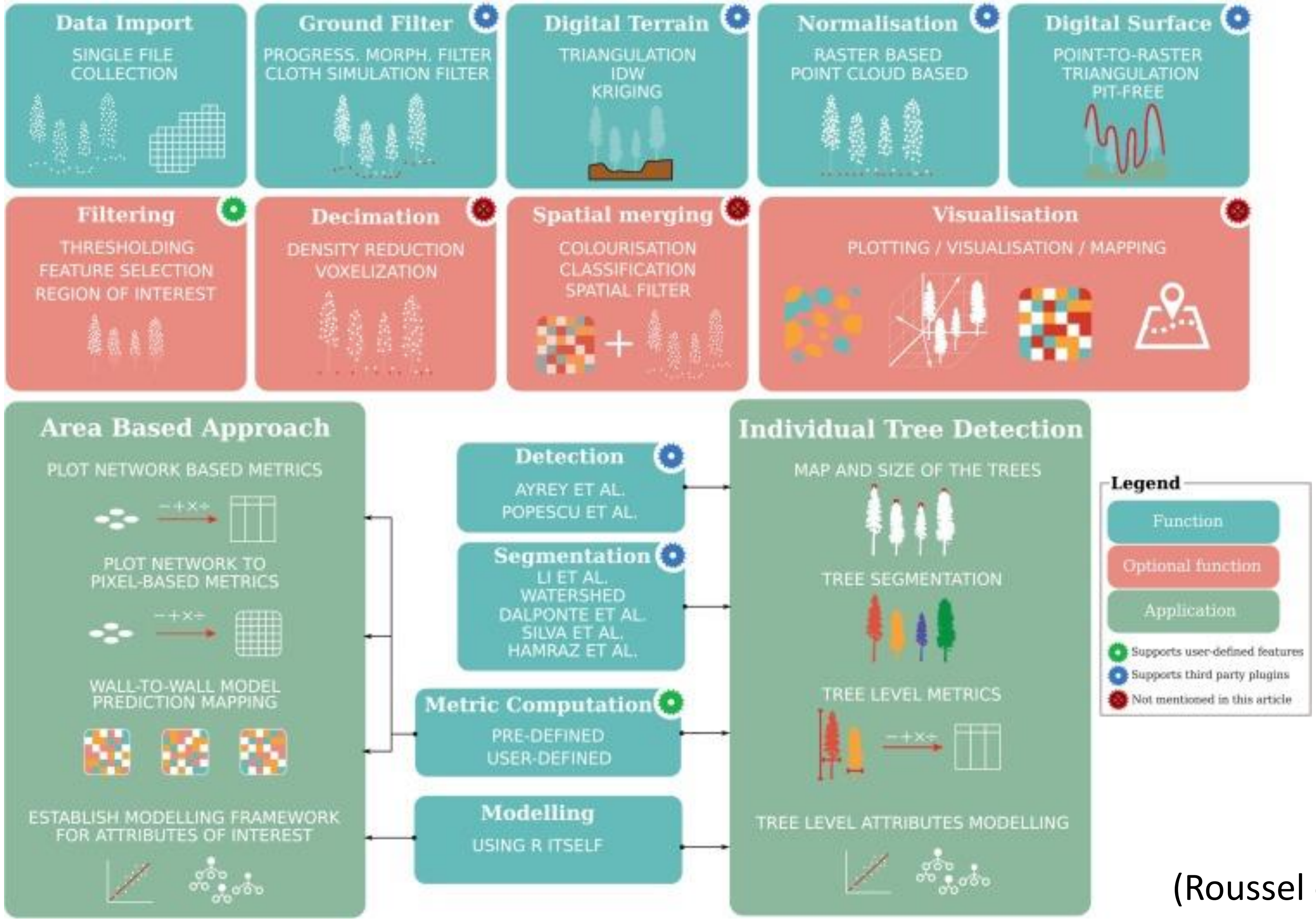
(Roussel et al., 2020)

Introduction to LiDR – what this software do?



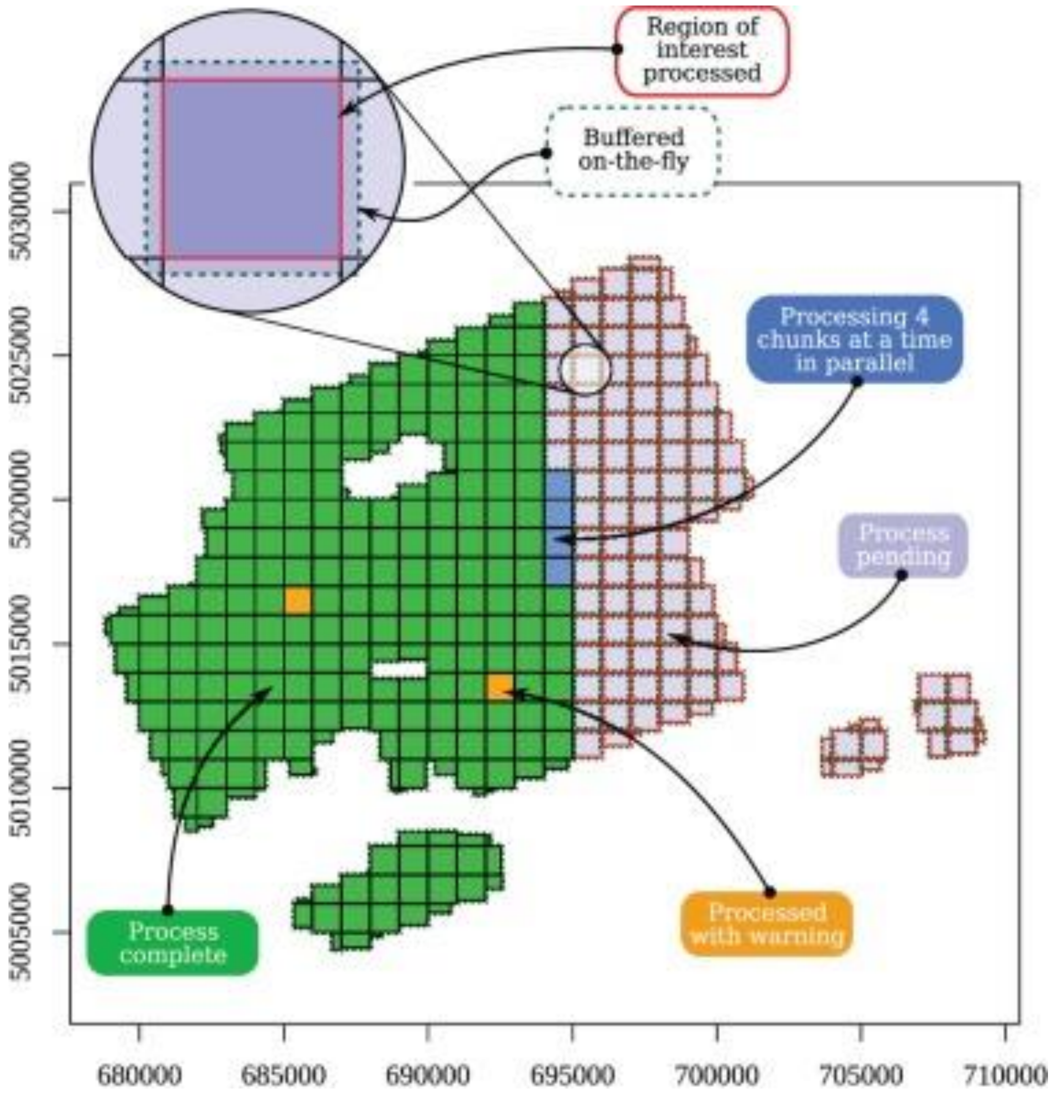
(Roussel et al., 2020)

Introduction to LiDR – what this software do?



(Roussel et al., 2020)

Introduction to LiDR – what this software do?



(Roussel et al., 2020)

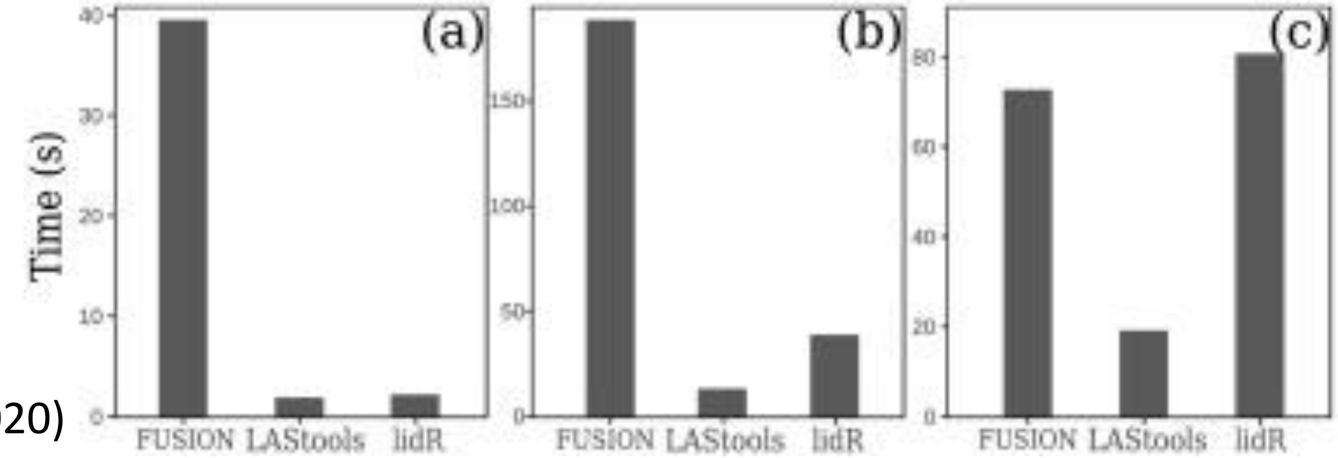
Introduction to LiDR – what this software do?

```
# Import required R packages
library("lidR")
library("rgdal")

# Set working directory
workingdirectory="D:/Koma/Paper1/Revision/input/process/" ## set this directory where your input las files are located
#workingdirectory="D:/Koma/Paper1/ALS/"
setwd(workingdirectory)

cores=18
chunksize=2000
buffer=1
resolution=1

rasterOptions(maxmemory = 200000000000)

library(future)
plan(multisession, workers = 4L)
set_lidr_threads(4L)

# Create catalog
ctg <- catalog(workingdirectory)

# Normalize with point neighborhood

opt_chunk_buffer(ctg) <- buffer
opt_chunk_size(ctg) <- chunksize
opt_cores(ctg) <- cores
opt_output_files(ctg) <- paste(workingdirectory,"normalized_neibased/{XLEFT}_{YBOTTOM}_gr_norm",sep="")

normalized_ctg=lasnormalize(ctg,knnidw(k=20,p=2))

# Generate DTM

opt_chunk_size(ctg) <- chunksize
opt_output_files(ctg)=""

dtm = grid_terrain(ctg, algorithm =knnidw(k=20,p=2), res=1, keep_lowest = TRUE)
crs(dtm) <- "+proj=sterea +lat_0=52.15616055555555 +lon_0=5.38763888888889 +k=0.9999079 +x_0=155000 +y_0=463000 +ellps=bessel +units=m +no_de
writeRaster(dtm, "dtm_1.tif",overwrite=TRUE)
```

Introduction to LiDR – what this software don't do?

- It is not the fastest software compared to LASTools however it is fully open-source
- Main application area is in forestry – no building modelling
- Predominantly only for processing airborne laser scanning data – processing terrestrial laser scanning data probably should be carried out in other software

(Roussel et al., 2020)



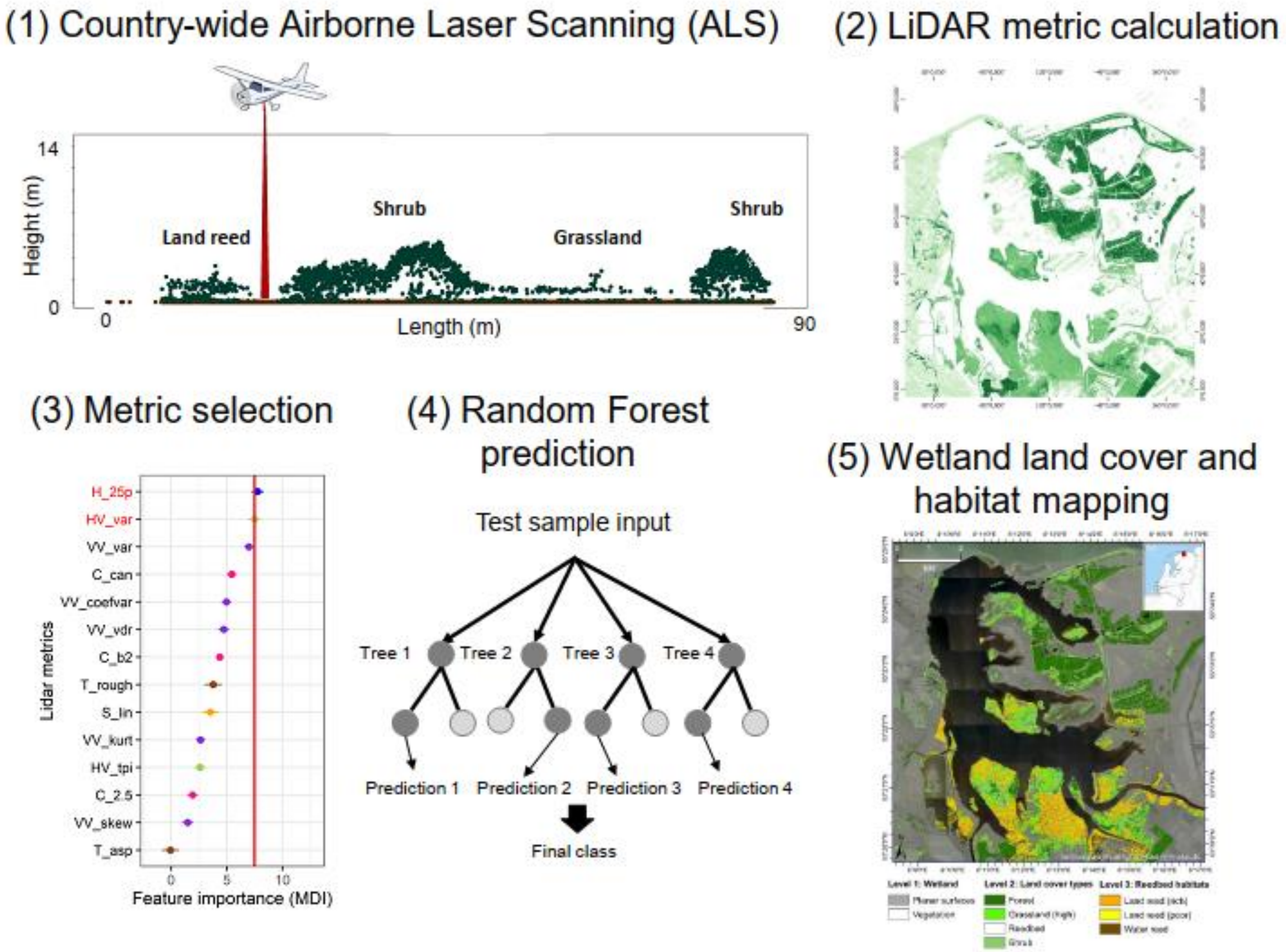
Application example I. : classifying wetlands

Remote Sensing in Ecology and Conservation — Open Access — ZSL LET'S WORK FOR WILDLIFE

Original Research | Open Access

Classifying wetland-related land cover types and habitats using fine-scale lidar metrics derived from country-wide Airborne Laser Scanning

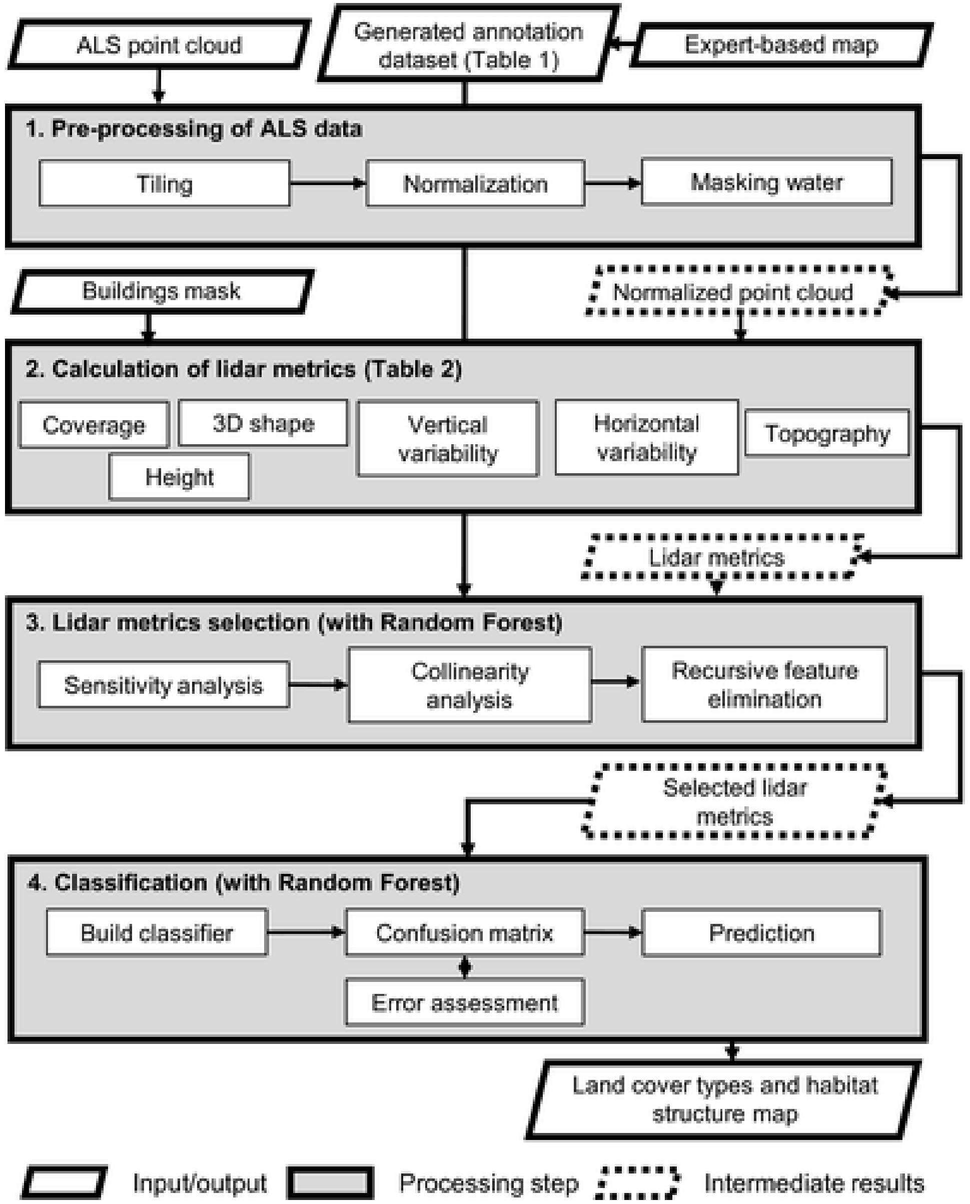
Zsófia Koma, Arie C. Seijmonsbergen, W. Daniel Kissling



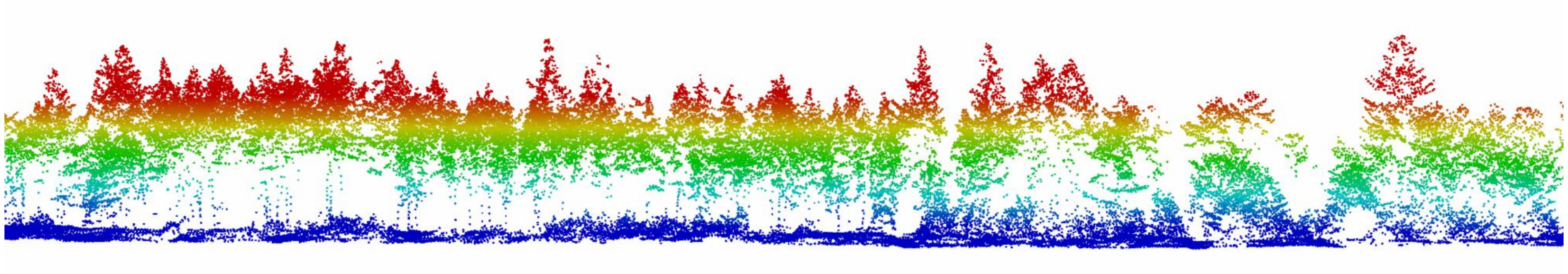
GitHub

https://github.com/eEcoLiDAR/PhDPaper1_Classifying_wetland_habitats

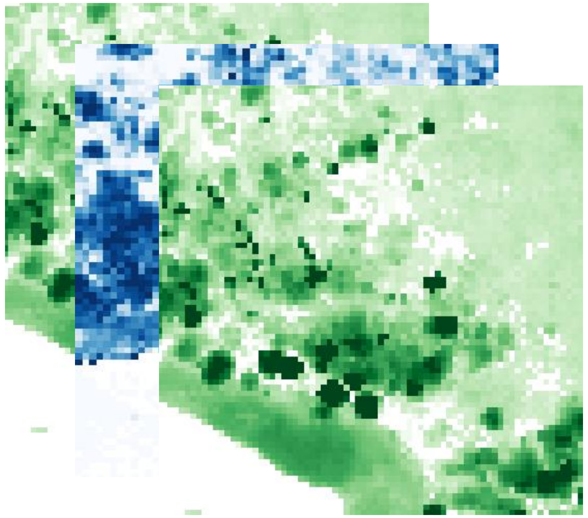
Classifying wetlands



Classifying wetlands – calculation of LiDAR metrics



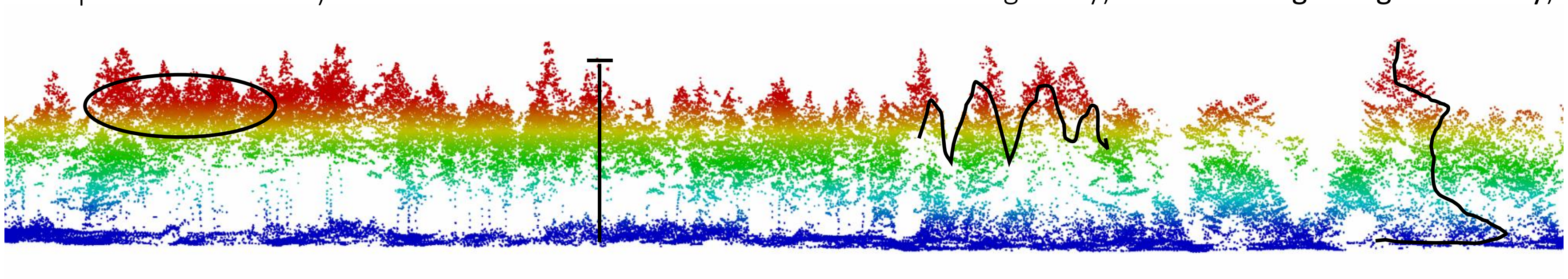
Result: multiple raster layers



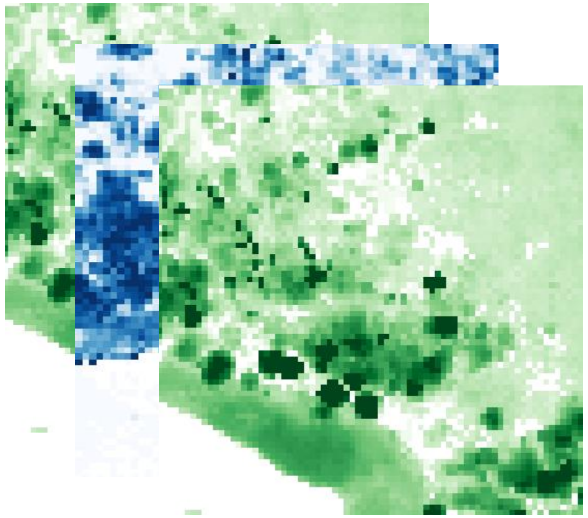
Bakx, T.R.M., **Koma, Z.**, Seijmonsbergen, A.C., Kissling, W.D., 2019. Use and categorization of light detection and ranging vegetation metrics in avian diversity and species distribution research. Divers. Distrib. 25, 1045–1059. https://doi.org/10.1111/ddi.12915

Classifying wetlands – calculation of LiDAR metrics

Vegetation cover	***Vegetation height***	***Vegetation complexity***	
Cover (e.g. canopy cover, penetration ratios)	Height (e.g. mean, maximum)	Horizontal heterogeneity (e.g. canopy height heterogeneity)	Vertical variability (e.g. Shannon evenness, Foliage Height Diversity)



Result: multiple raster layers



Bakx, T.R.M., **Koma, Z.**, Seijmonsbergen, A.C., Kissling, W.D., 2019. Use and categorization of light detection and ranging vegetation metrics in avian diversity and species distribution research. Divers. Distrib. 25, 1045–1059. https://doi.org/10.1111/ddi.12915

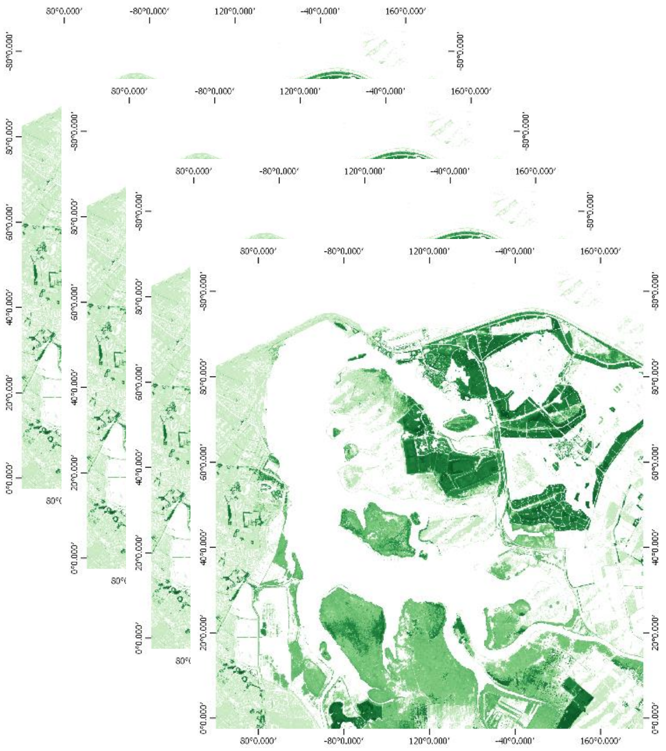
Classifying wetlands – calculation of LiDAR metrics

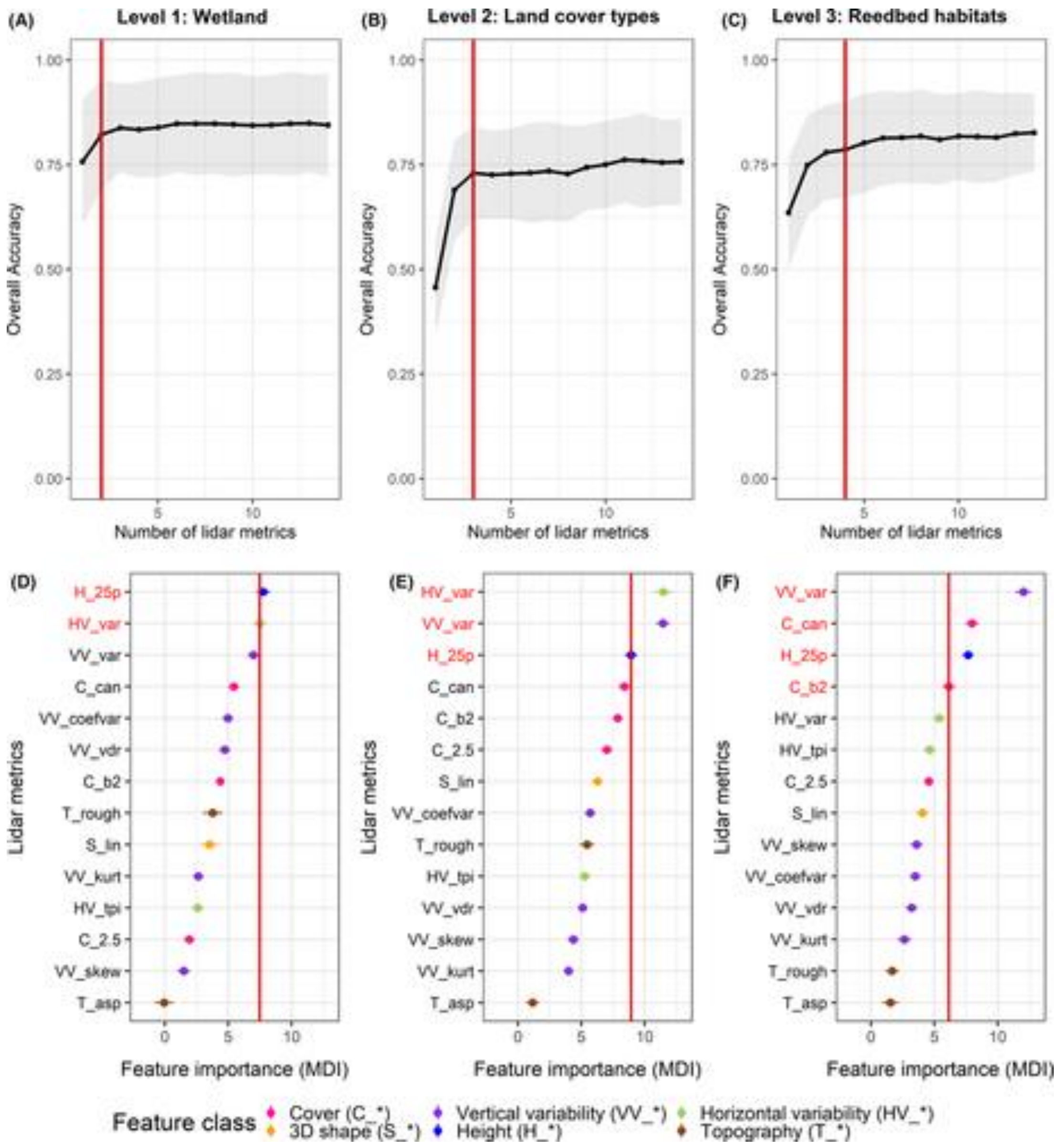
```r
# Import required R packages
library("lidR")
library("rgdal")
source("D:/Koma/GitHub/PhDPaper1_Classifying_wetland_habitats/Function_LiDARMetricsCalc.R") #set where the Function*.R file located
#source("D:/GitHub/eEcoLiDAR/myPhD_escience_analysis//Paper1_inR_v2/Function_LiDARMetricsCalc.R")

# Set working directory
workingdirectory="D:/Koma/Paper1/Revision/input/process/" ## set this directory where your input las files are located
#workingdirectory="D:/Koma/Paper1/ALS/"
setwd(workingdirectory)

cores=18
chunksize=2000
buffer=1
resolution=10

rasterOptions(maxmemory = 200000000000)

### Ground run

ground_ctg <- catalog(workingdirectory)

opt_chunk_buffer(ground_ctg) <- buffer
opt_chunk_size(ground_ctg) <- chunksize
opt_cores(ground_ctg) <- cores

library(future)
plan(multisession, workers = 6L)
set_lidr_threads(6L)

# Calculate metrics

covermetrics = grid_metrics(ground_ctg,  CoverageMetrics(Z,Classification), res = resolution)
#plot(covermetrics)
writeRaster(covermetrics,paste("covermetrics_gr_",resolution,"m.grd",sep=""),overwrite=TRUE)

shapemetrics = grid_metrics(ground_ctg,  EigenMetrics(X,Y,Z), res = resolution)
#plot(shapemetrics)
writeRaster(shapemetrics,paste("shapemetrics_gr_",resolution,"m.grd",sep=""),overwrite=TRUE)

vertdistr_metrics = grid_metrics(ground_ctg, VertDistr_Metrics(Z),res=resolution)
#plot(vertdistr_metrics)
writeRaster(vertdistr_metrics,paste("vertdistr_metrics_gr_",resolution,"m.grd",sep=""),overwrite=TRUE)

height_metrics = grid_metrics(ground_ctg, HeightMetrics(Z),res=resolution)
#plot(height_metrics)
writeRaster(height_metrics,paste("height_metrics_gr_",resolution,"m.grd",sep=""),overwrite=TRUE)

proj4string(height_metrics)<- CRS("+proj=sterea +lat_0=52.15616055555555 +lon_0=5.38763888888889 +k=0.9999079 +x_0=155000 +y_0=463000 +ellps=bessel +units=m +no_defs")

horizontal_metrics = HorizontalMetrics(height_metrics$zmax)
#plot(horizontal_metrics)
writeRaster(horizontal_metrics,paste("horizontal_metrics_gr_",resolution,"m.grd",sep=""),overwrite=TRUE)
```

Classifying wetlands – calculation of LiDAR metrics

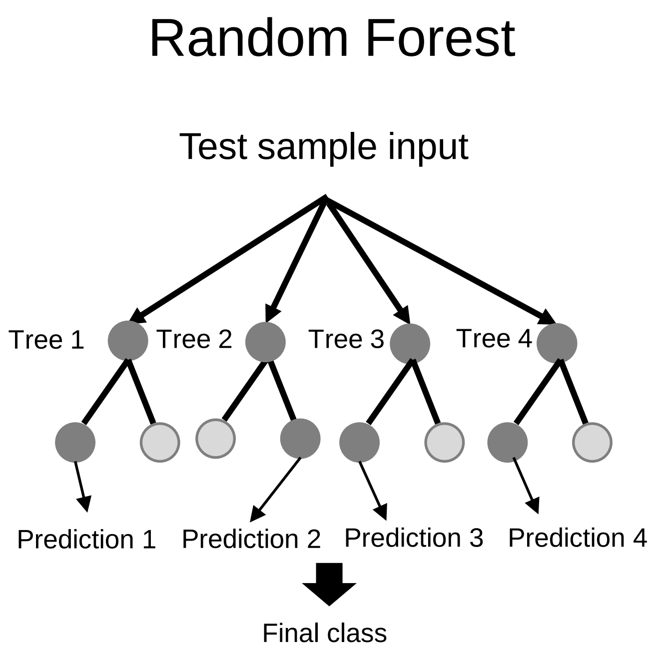
```
VertDistr_Metrics = function(z)
{
  library("e1071")

  p=proportion(z, by = 1)
  p_whnull=p[p>0]

  vertdistr_metrics = list(
    zstd = sd(z),
    zvar = var(z),
    zskew = skewness(z),
    zkurto = kurtosis(z),
    canrelrat = (mean(z)-min(z))/max(z)-min(z),
    vertdenrat = (max(z)-median(z))/max(z),
    simpson = 1/sum(sqrt(p)),
    shannon = -sum(p_whnull*log(p_whnull))
  )
  return(vertdistr_metrics)
}

HeightMetrics = function(z)
{

  heightmetrics = list(
    zmax = max(z),
    zmean = mean(z),
    zmedian = median(z),
    z025quantile = quantile(z, 0.25),
    z075quantile = quantile(z, 0.75),
    z090quantile = quantile(z, 0.90),
    zcoeffvar = sd(z)/mean(z),
    zmin = min(z)
  )
  return(heightmetrics)
}
```

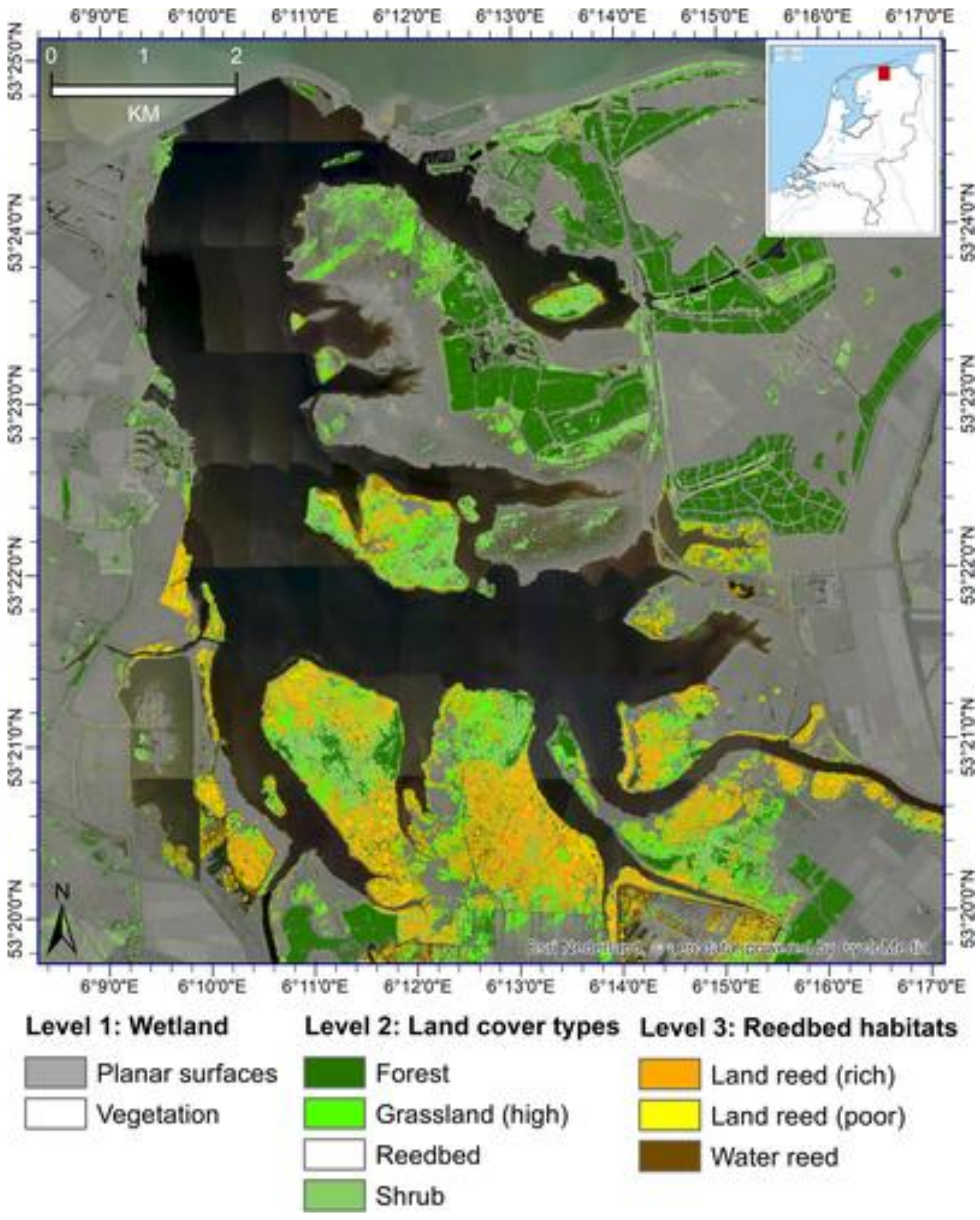
Classifying wetlands – LiDAR metrics selection



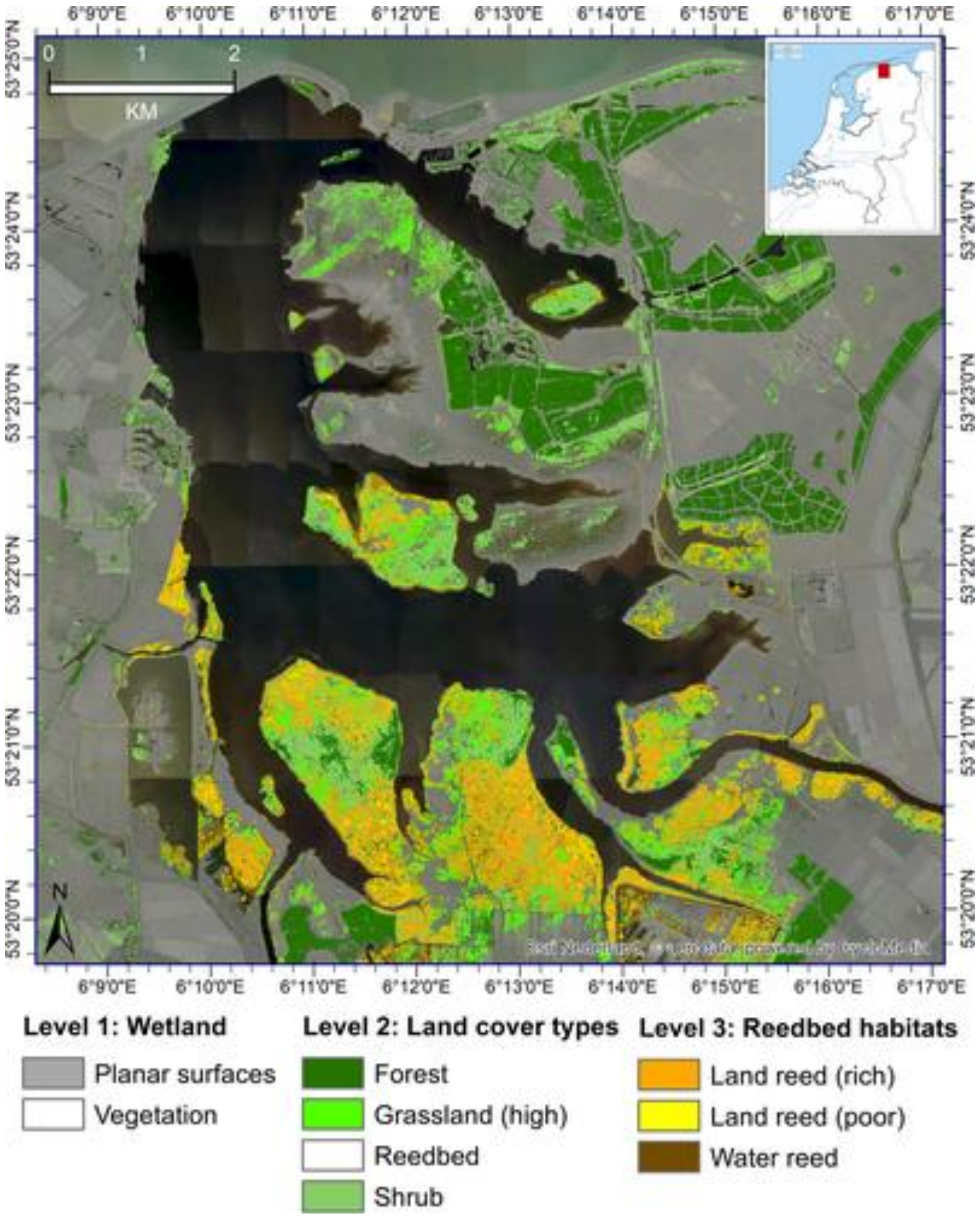


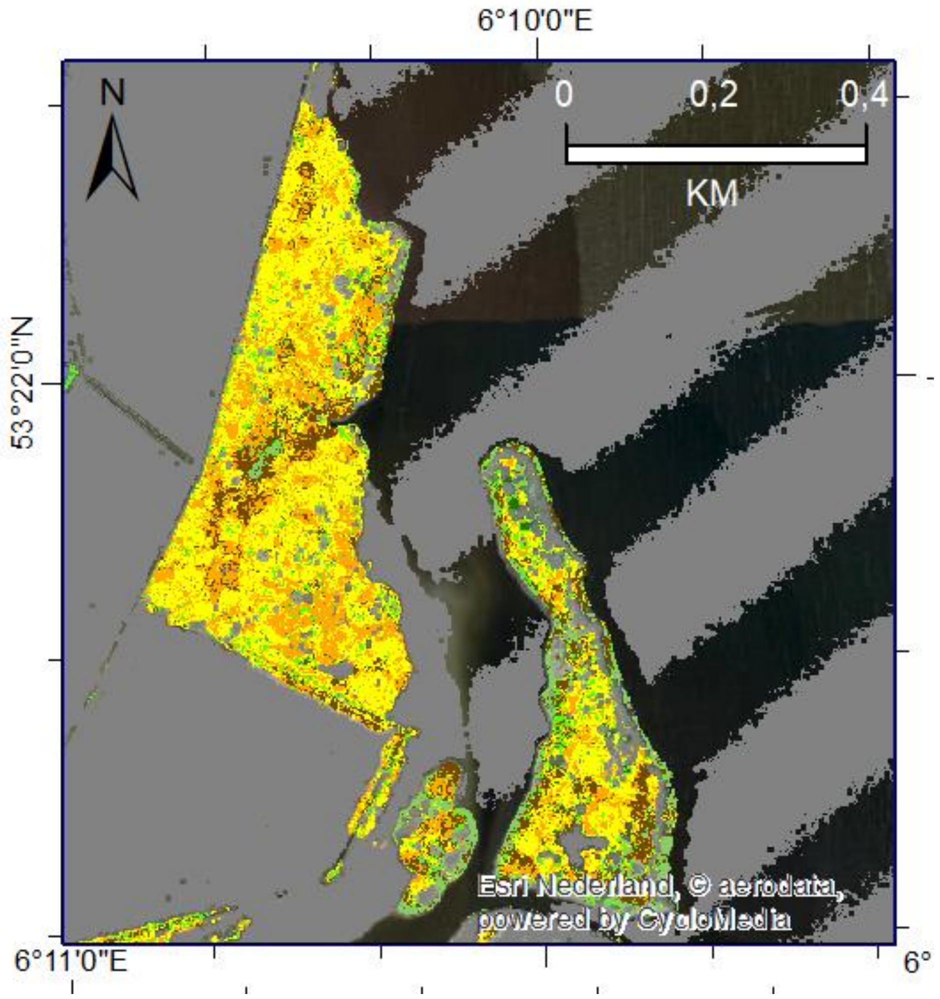
Classifying wetlands – Results





Classifying wetlands – Results





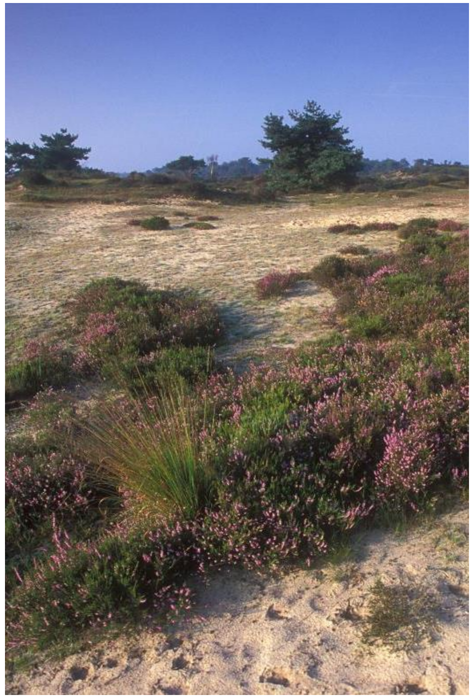
Application example II. : habitat preference of butterflies

Diversity and Distributions
Open Access
A Journal of Conservation Biogeography

RESEARCH ARTICLE | Open Access

Identifying fine-scale habitat preferences of threatened butterflies using airborne laser scanning

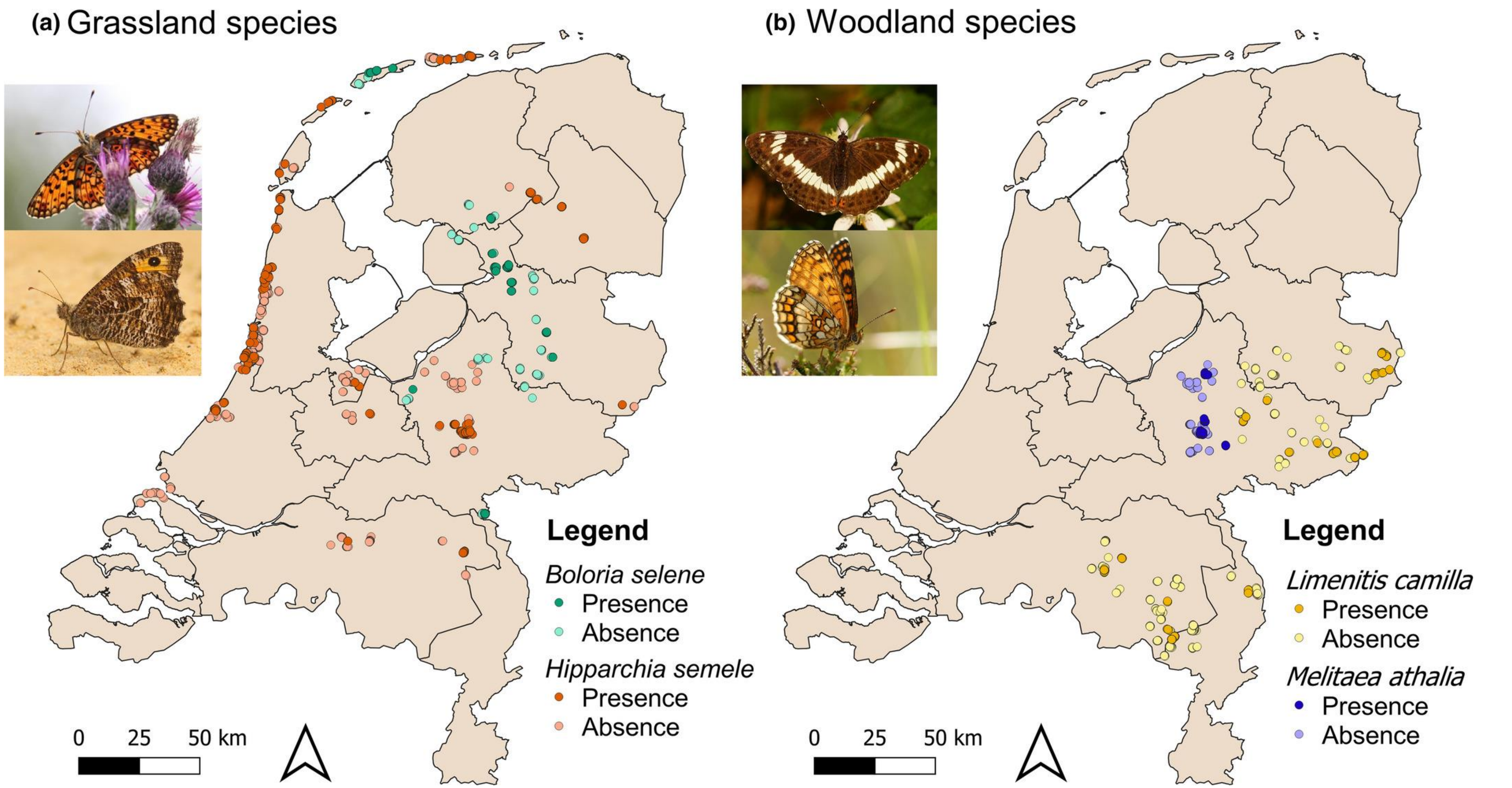
Jan Peter Reinier de Vries, Zsófia Koma, Michiel F. WallisDeVries, W. Daniel Kissling



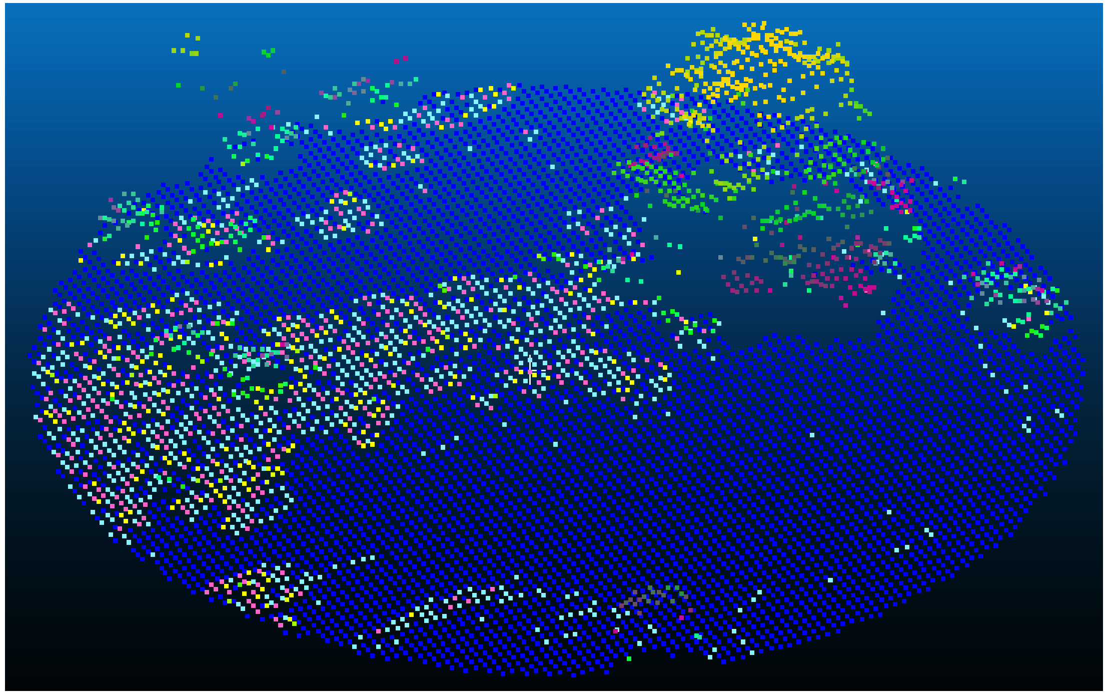
GitHub

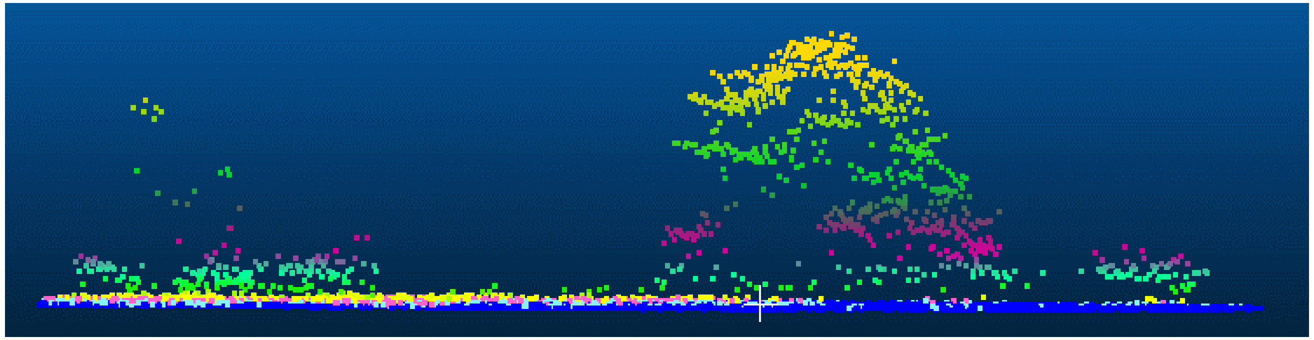
https://github.com/eEcoLiDAR/lidarButterfly

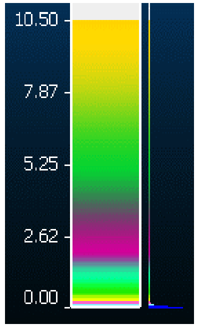
Application example II. : habitat preference of butterflies



Application example II. : extract the point cloud across the country



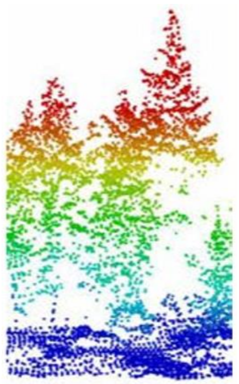




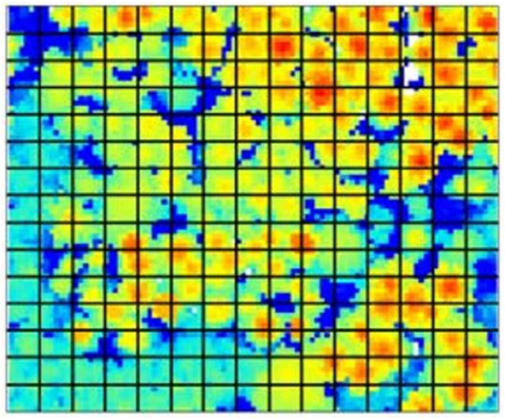
Application example II. : habitat preference of butterflies

Airborne Laser Scanning

Point cloud

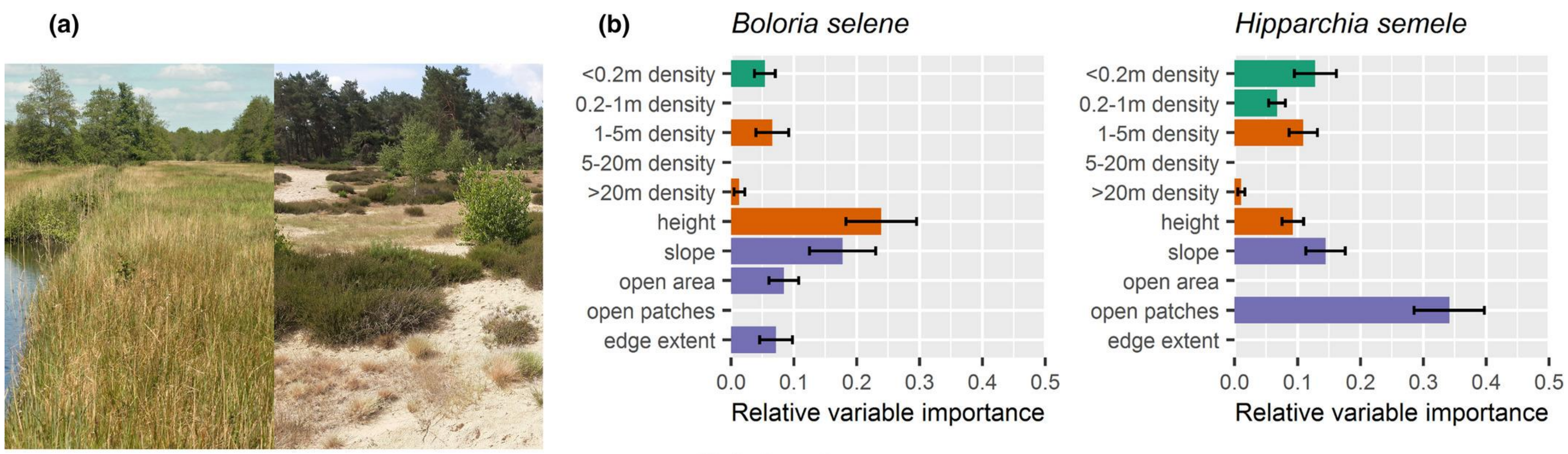


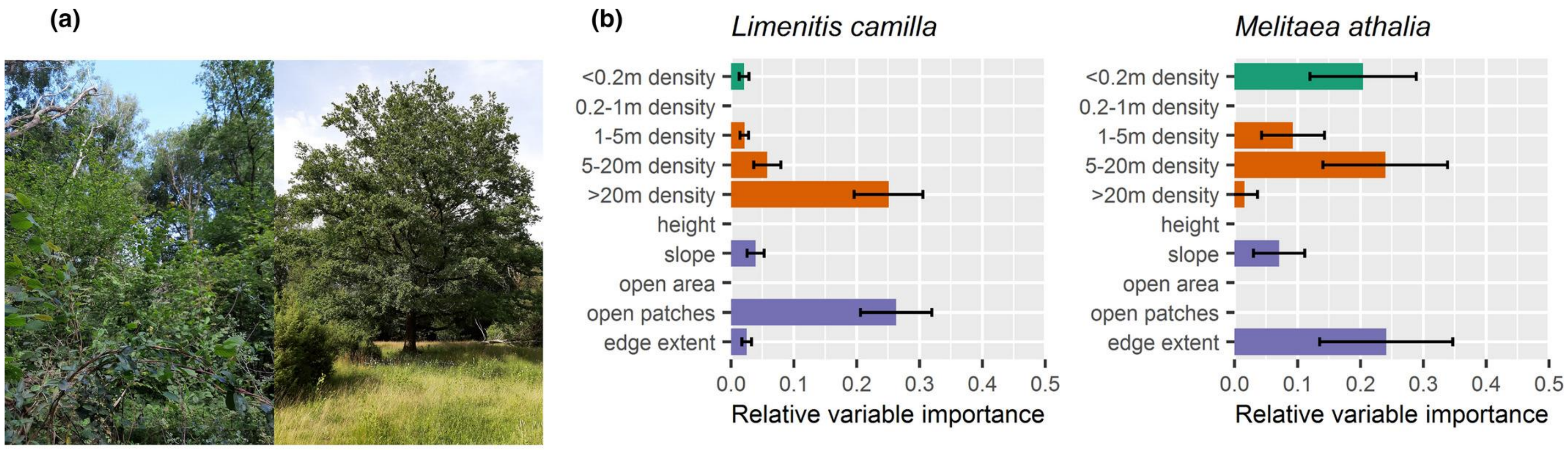
LiDAR metrics



Class	Type	Feature	Relevance
Vertical vegetation structure	**Low vegetation**	Vegetation density per height layer [-]:	Vegetation elements:
Vertical vegetation structure	**Low vegetation**	• <0.2 m	Low herbs/grasses
Vertical vegetation structure	**Low vegetation**	• >0.2 and <1 m	Tall herbs
Vertical vegetation structure	**High vegetation**	• >1 and <5 m	Shrubs
Vertical vegetation structure	**High vegetation**	• >5 and <20 m	Trees
Vertical vegetation structure	**High vegetation**	• >20 m	High canopy
Vertical vegetation structure	**High vegetation**	90th height percentile [m]	Highest vegetation
Horizontal hetero-geneity	**High vegetation**	Vegetation height roughness [m]: • Total vegetation	Structural diversity: Total (mainly high)
Horizontal hetero-geneity	**Low vegetation**	• Low vegetation	Low (<1 m)
Horizontal (landscape) structure, per 100 m radius	**Terrain**	Mean slope [degree]	Hillyness
Horizontal (landscape) structure, per 100 m radius	**Open landscape elements**	Area of low vegetation [-]	Open area extent
Horizontal (landscape) structure, per 100 m radius	**Open landscape elements**	Number of patches of low vegetation [-]	Fragmentation
Horizontal (landscape) structure, per 100 m radius	**Open landscape elements**	Edges of low vegetation [-]	Edges extent

Application example II. : habitat preference of butterflies





Conclusion

- The software landscape of handling and processing LiDAR datasets has been changed – more and more open-source software tools are available

- This overall makes LiDAR data more accessible for different applications

- The standardized processing, handling country-.wide ALS datasets still remain a challenge

Conclusion

- The software landscape of handling and processing LiDAR datasets is rapidly changing – more and more open-source software tools are available
- This overall makes LiDAR data more accessible for different applications
- The standardized processing, handling country-.wide ALS datasets still remain a challenge

Thank you for the attention!