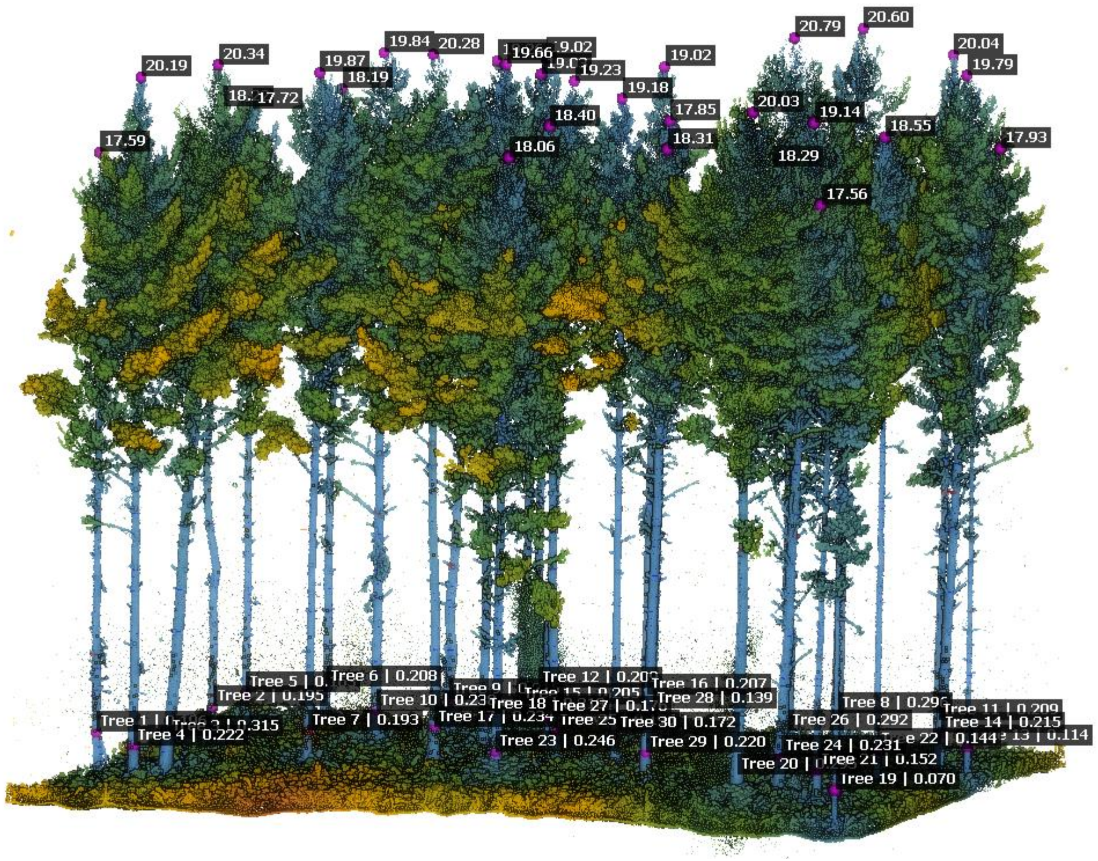
3DFin: *a free and user-friendly* software *for* forest inventory *from* terrestrial point clouds



Carlos Cabo → *Universidad de Oviedo, Swansea University*
Diego Laíño → *IMIB, CSIC*
Covadonga Prendes → *CETEMAS*
Celestino Ordóñez → *Universidad de Oviedo*
Cristina Santín → *IMIB-CSIC, Swansea University*

Point cloud types

Technologies →

Point of view ↓

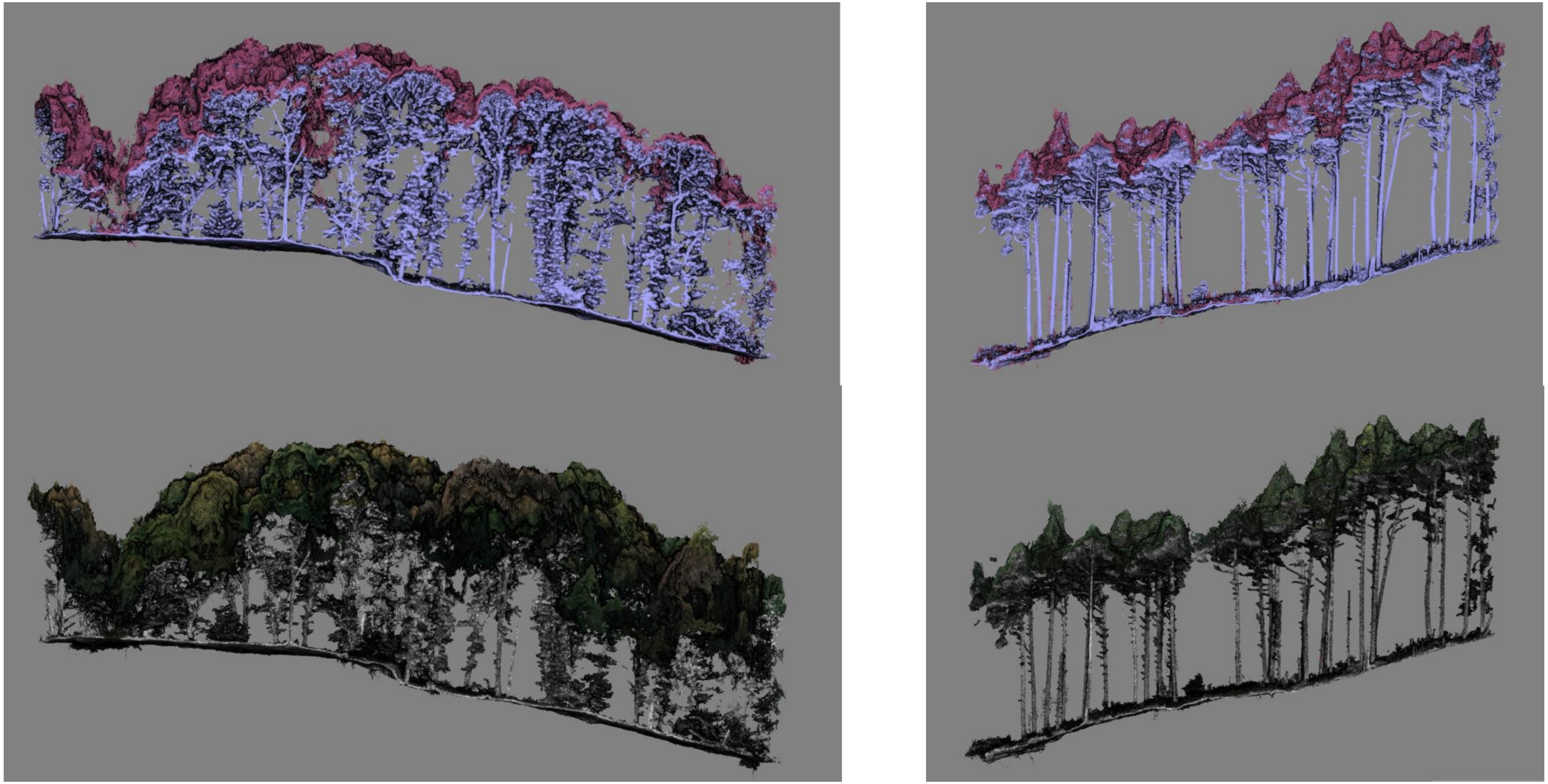
	Laser Scanning	Photogrammetry
Terrestrial	Static (**TLS**) Mobile (**WLS** / HHLS)	Terrestrial **SfM**
Aerial	**ALS**	Aerial **SfM**

Multi-sensor comparison

pointclouds.wsl.ch

Swiss NFI @ a Glance – CRS –
Data acquisition – Information Extraction –
Make CRS operational?

Do we have to choose?



And, all that, for what?

Precision forest inventory

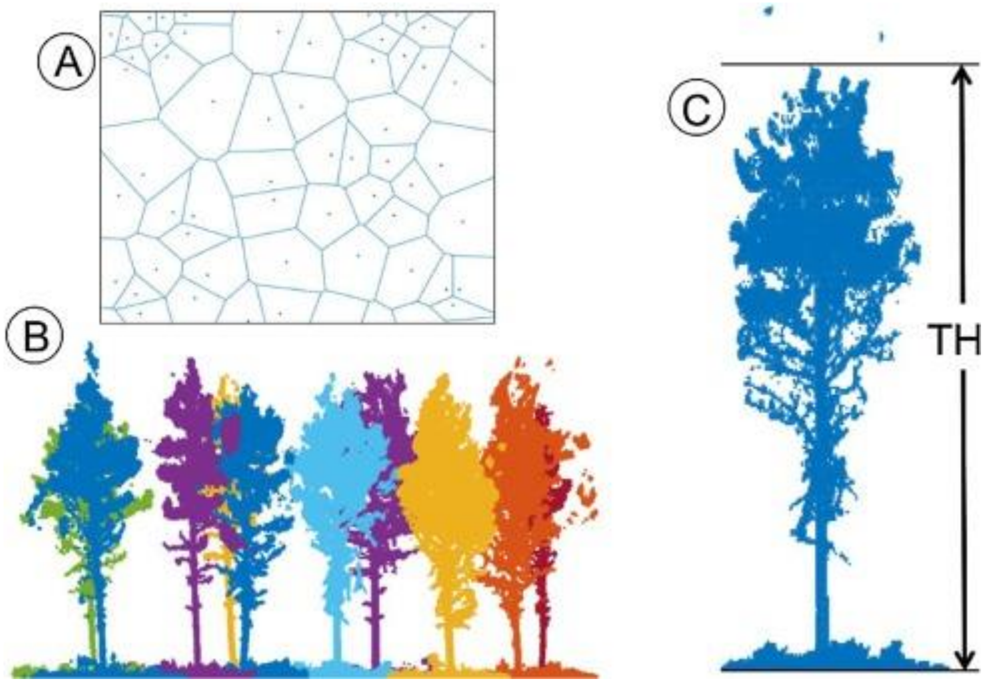
→ Analogy with traditional inventories

Forest ecology

→ Semantic 3D models of the vegetation structure

Our algorithms + implementations

Cabo, C., Ordóñez, C., López-Sánchez, C. A., & Armesto, J. (**2018**). Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning. *International journal of applied earth observation and geoinformation*, *69*, 164-174.




```
for each Tree
    labelTree → Revise(false)
    Grouped → false
    fitCircle&Check(PointsBH) → (Center, R, Revise(true/false))
end for

function *fitCircle&Check(PointsBH)    *[Recursive]
    fitCircle(PointsBH) → (Center, R)
    createCircle(Center, R / 2) → InnerCircle
    if [there are PointsBH inside InnerCircle] or not [MinR < R < MaxR]
        if Grouped → false
            groupNeighborPoints → Grouped(true)
            *fitCircle&Check(LargestGroup)
        else
            labelTree → Revise(true)
        end if
    end if
end function

//{Variables.fields   Standard operators   Functions and other operators   Results}//
```

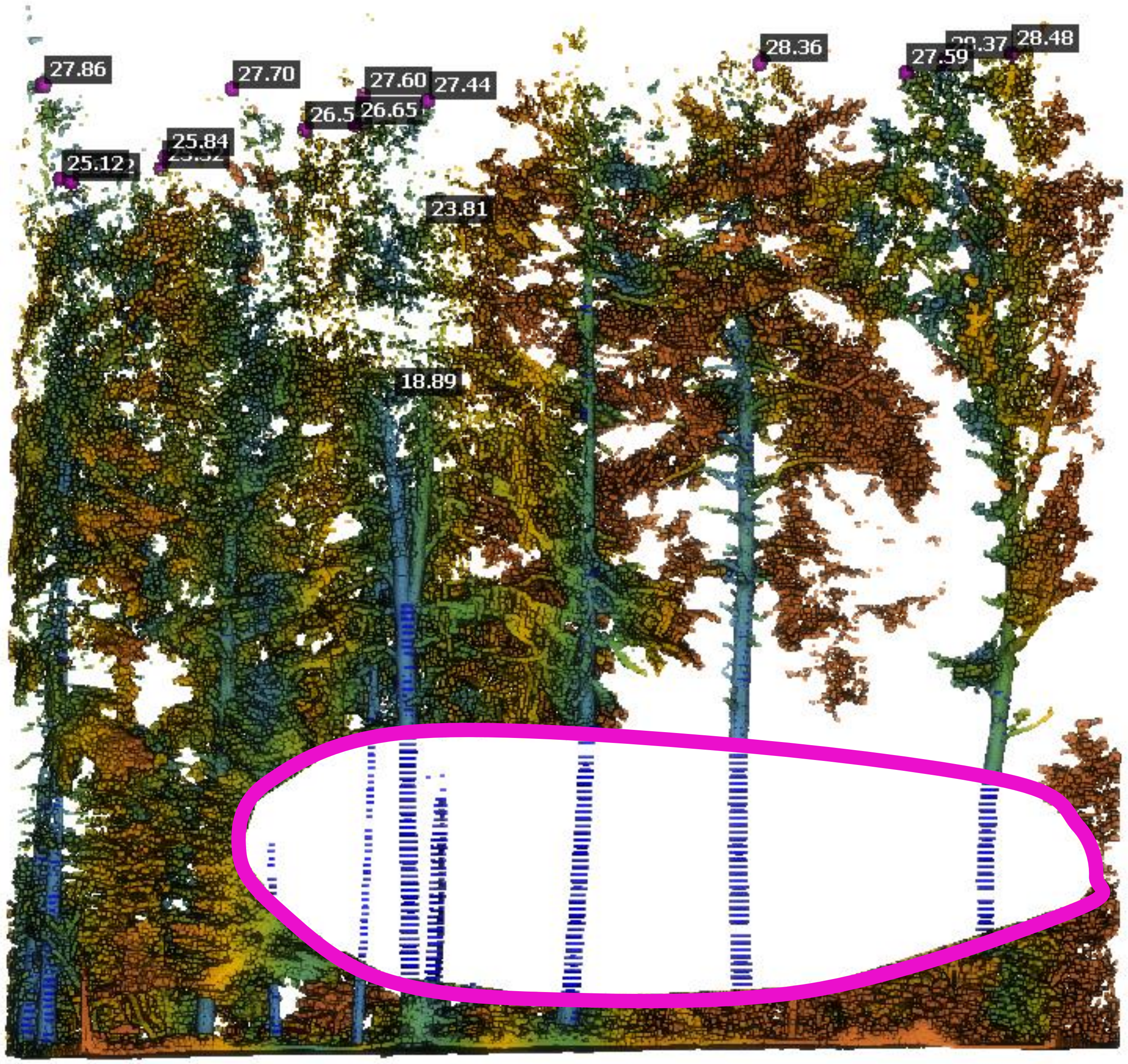
Our algorithms + implementations

Inputs:
Terrestrial point cloud
+ UAV data (optional)

Outputs:
tree location
DBH
tree height
diameters along stem

Robust to:
understorey
branches
abrupt topography
point cloud type

Also:
Easy to use
Fast
Free and opensource
Well documented



3DFin

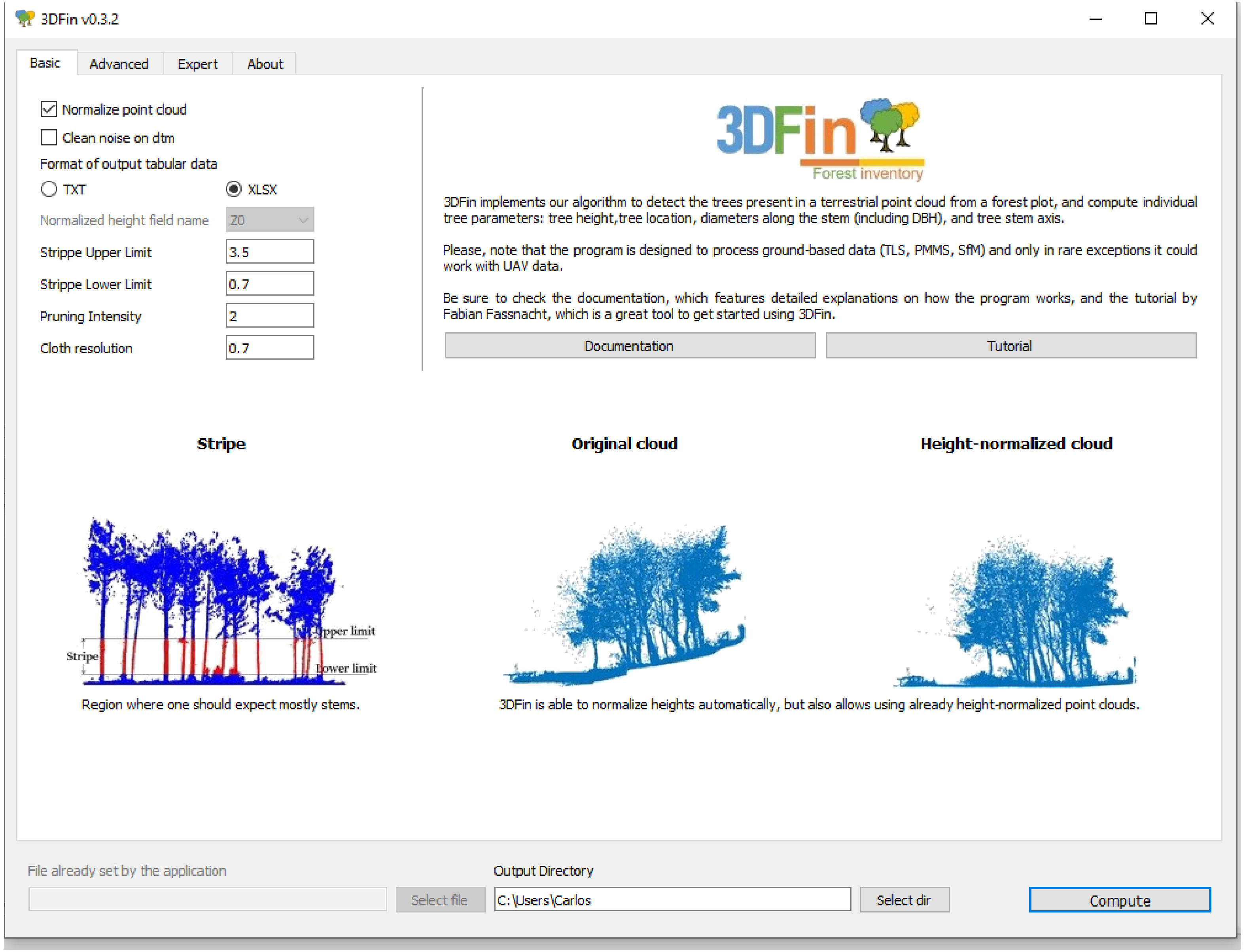
Forest inventory

Public implementations:

3DFin Forest inventory

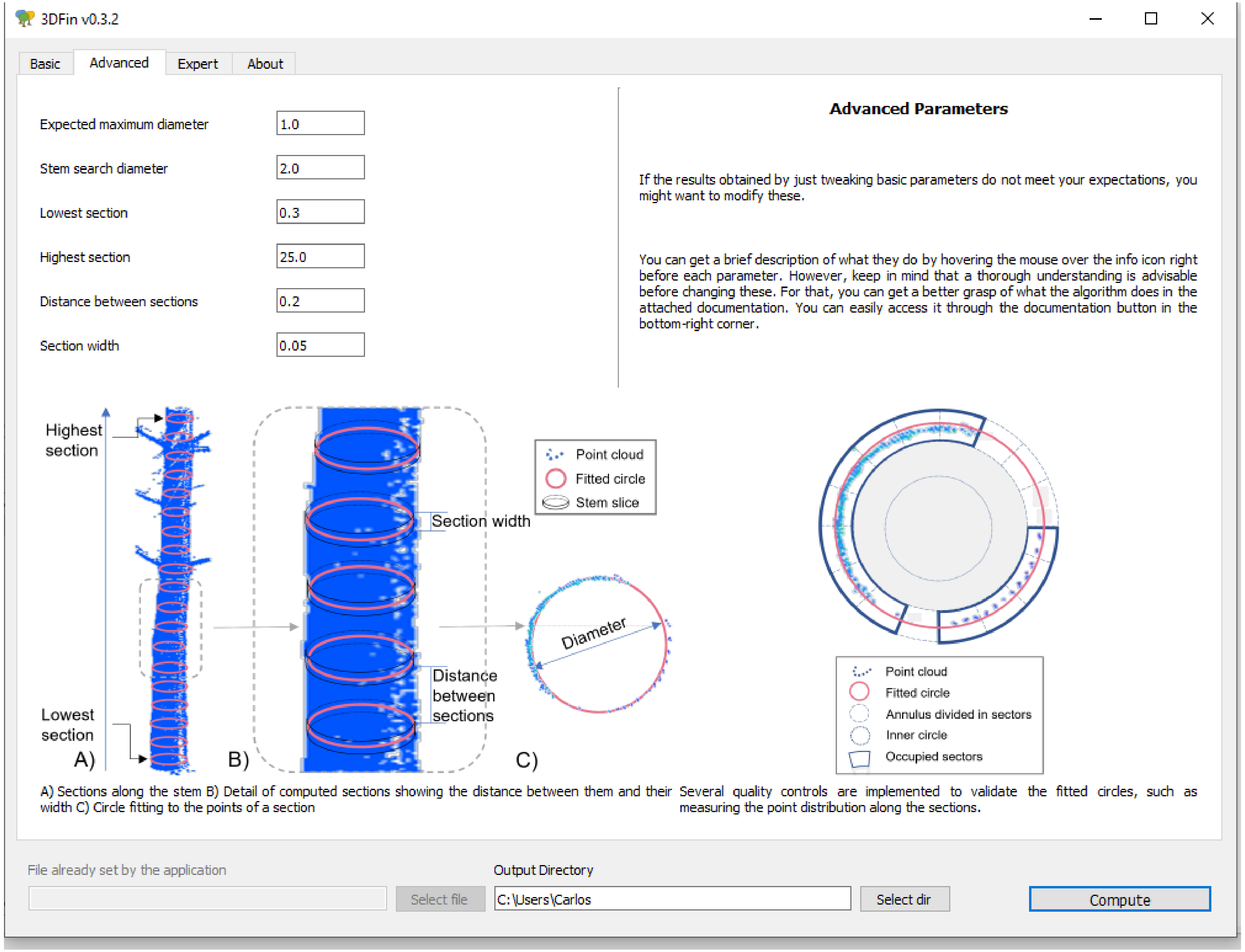
- P Python Package
- S Standalone
- C Plugin CloudCompare
- Q Plugin QGIS

Interface



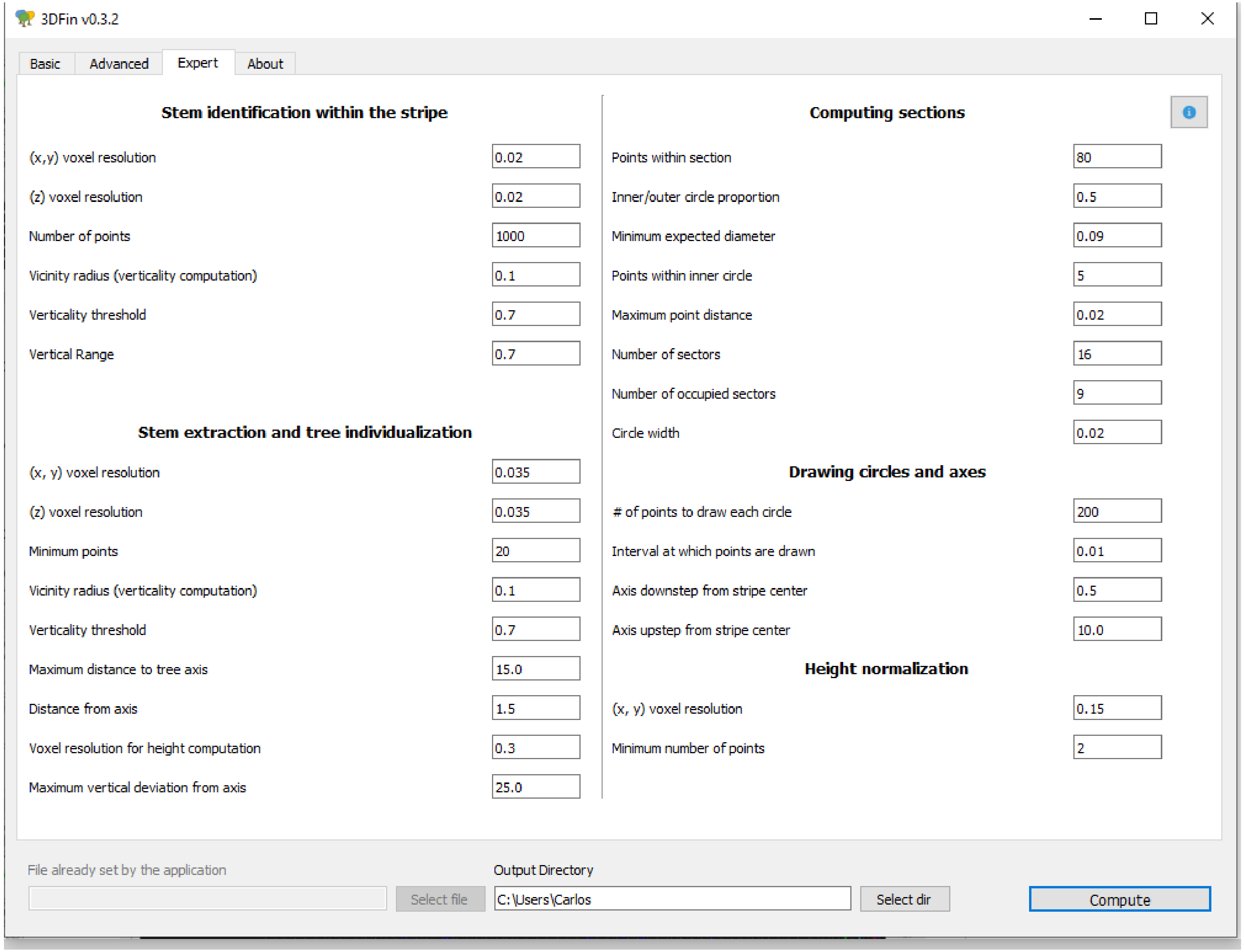
3DFin v0.3.2
Basic
Advanced
Expert
About
Normalize point cloud
Clean noise on dtm
Format of output tabular data
TXT
XLSX
Normalized height field name
Z0
Stripe Upper Limit
3.5
Stripe Lower Limit
0.7
Pruning Intensity
2
Cloth resolution
0.7
3DFin
Forest inventory
3DFin implements our algorithm to detect the trees present in a terrestrial point cloud from a forest plot, and compute individual tree parameters: tree height, tree location, diameters along the stem (including DBH), and tree stem axis.
Please, note that the program is designed to process ground-based data (TLS, PMMS, SfM) and only in rare exceptions it could work with UAV data.
Be sure to check the documentation, which features detailed explanations on how the program works, and the tutorial by Fabian Fassnacht, which is a great tool to get started using 3DFin.
Documentation
Tutorial
Stripe
Original cloud
Height-normalized cloud
Upper limit
Stripe
Lower limit
Region where one should expect mostly stems.
3DFin is able to normalize heights automatically, but also allows using already height-normalized point clouds.
File already set by the application
Select file
Output Directory
C:\Users\Carlos
Select dir
Compute

Interface

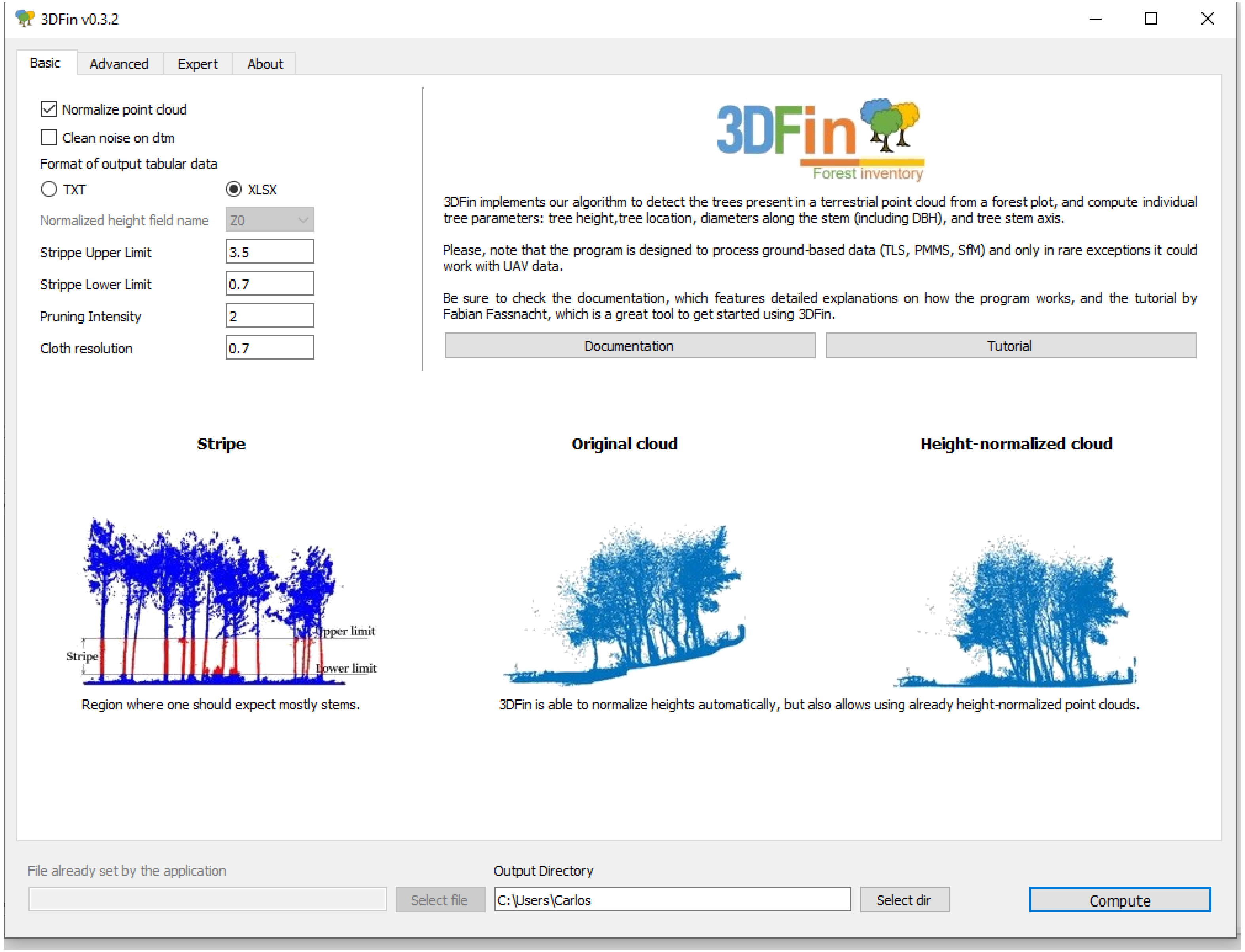


3DFin v0.3.2
Basic
Advanced
Expert
About
Expected maximum diameter
1.0
Stem search diameter
2.0
Lowest section
0.3
Highest section
25.0
Distance between sections
0.2
Section width
0.05
Advanced Parameters
If the results obtained by just tweaking basic parameters do not meet your expectations, you might want to modify these.
You can get a brief description of what they do by hovering the mouse over the info icon right before each parameter. However, keep in mind that a thorough understanding is advisable before changing these. For that, you can get a better grasp of what the algorithm does in the attached documentation. You can easily access it through the documentation button in the bottom-right corner.
Highest section
Lowest section
A)
B)
C)
Section width
Distance between sections
Diameter
Point cloud
Fitted circle
Stem slice
Point cloud
Fitted circle
Annulus divided in sectors
Inner circle
Occupied sectors
A) Sections along the stem B) Detail of computed sections showing the distance between them and their width C) Circle fitting to the points of a section
Several quality controls are implemented to validate the fitted circles, such as measuring the point distribution along the sections.
File already set by the application
Select file
Output Directory
C:\Users\Carlos
Select dir
Compute

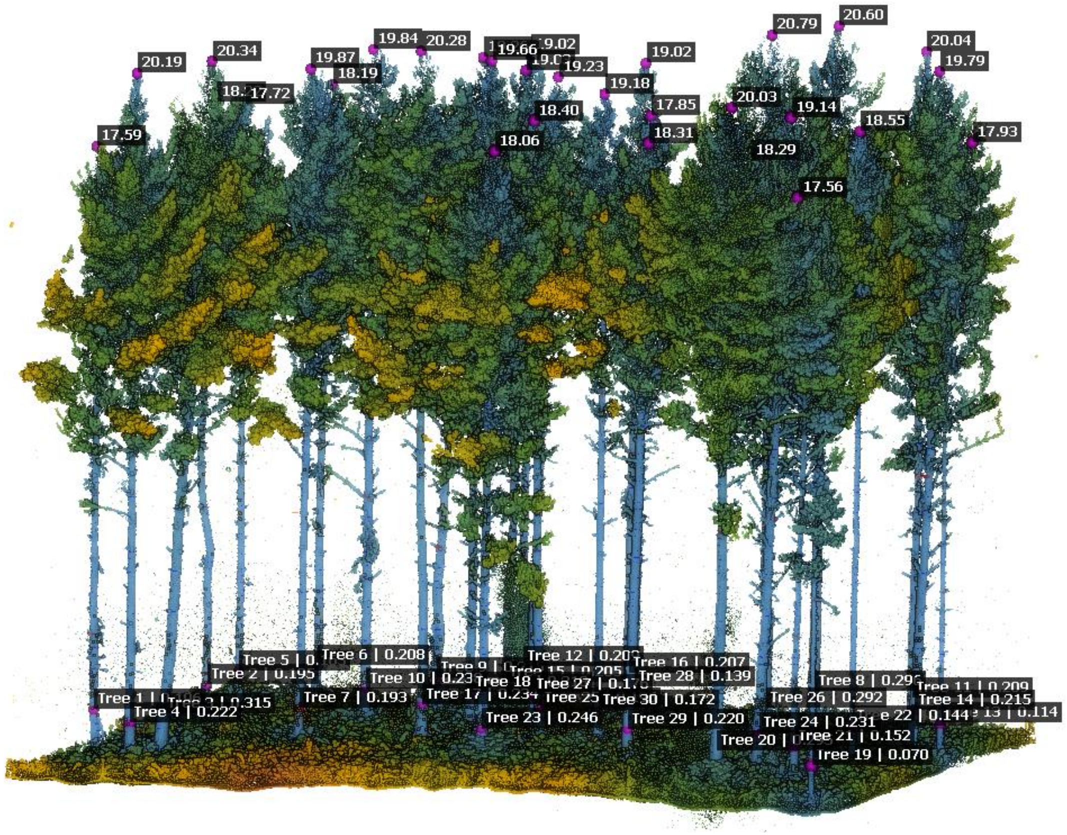
Interface

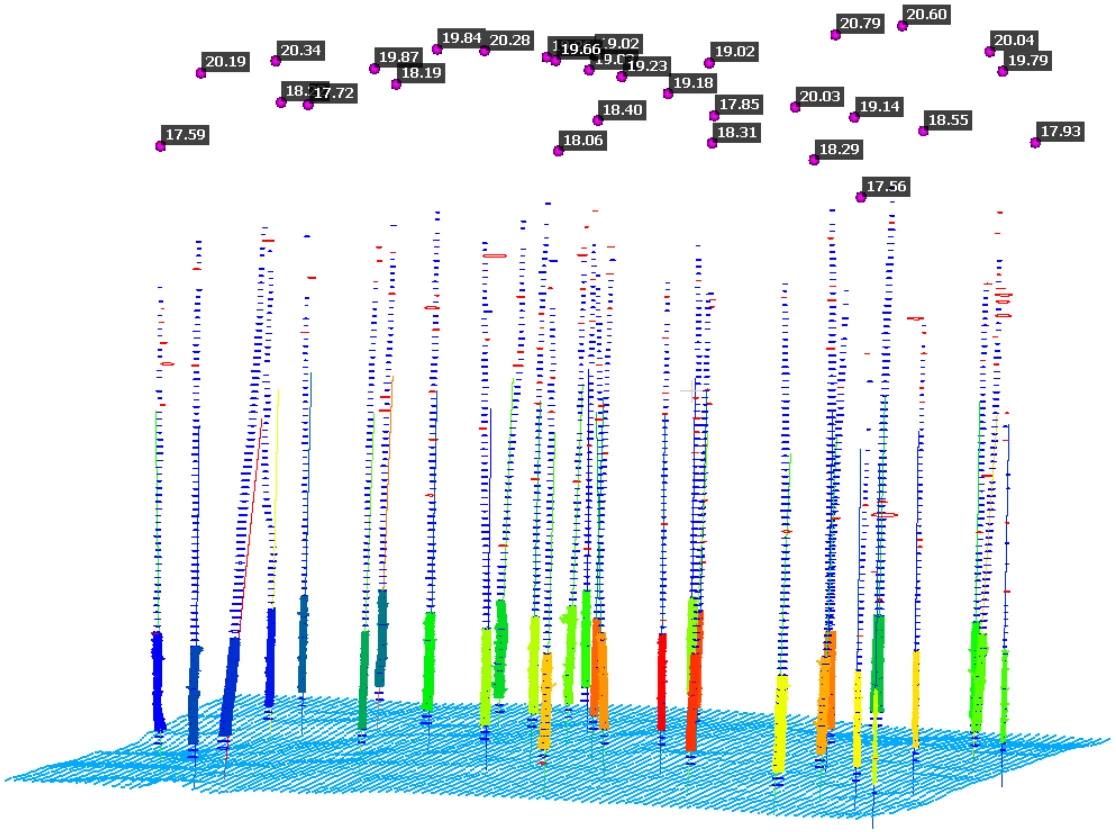

3DFin v0.3.2
Basic
Advanced
Expert
About
Stem identification within the stripe
(x,y) voxel resolution
0.02
(z) voxel resolution
0.02
Number of points
1000
Vicinity radius (verticality computation)
0.1
Verticality threshold
0.7
Vertical Range
0.7
Stem extraction and tree individualization
(x, y) voxel resolution
0.035
(z) voxel resolution
0.035
Minimum points
20
Vicinity radius (verticality computation)
0.1
Verticality threshold
0.7
Maximum distance to tree axis
15.0
Distance from axis
1.5
Voxel resolution for height computation
0.3
Maximum vertical deviation from axis
25.0
Computing sections
Points within section
80
Inner/outer circle proportion
0.5
Minimum expected diameter
0.09
Points within inner circle
5
Maximum point distance
0.02
Number of sectors
16
Number of occupied sectors
9
Circle width
0.02
Drawing circles and axes
# of points to draw each circle
200
Interval at which points are drawn
0.01
Axis downstep from stripe center
0.5
Axis upstep from stripe center
10.0
Height normalization
(x, y) voxel resolution
0.15
Minimum number of points
2
File already set by the application
Select file
Output Directory
C:\Users\Carlos
Select dir
Compute

Interface

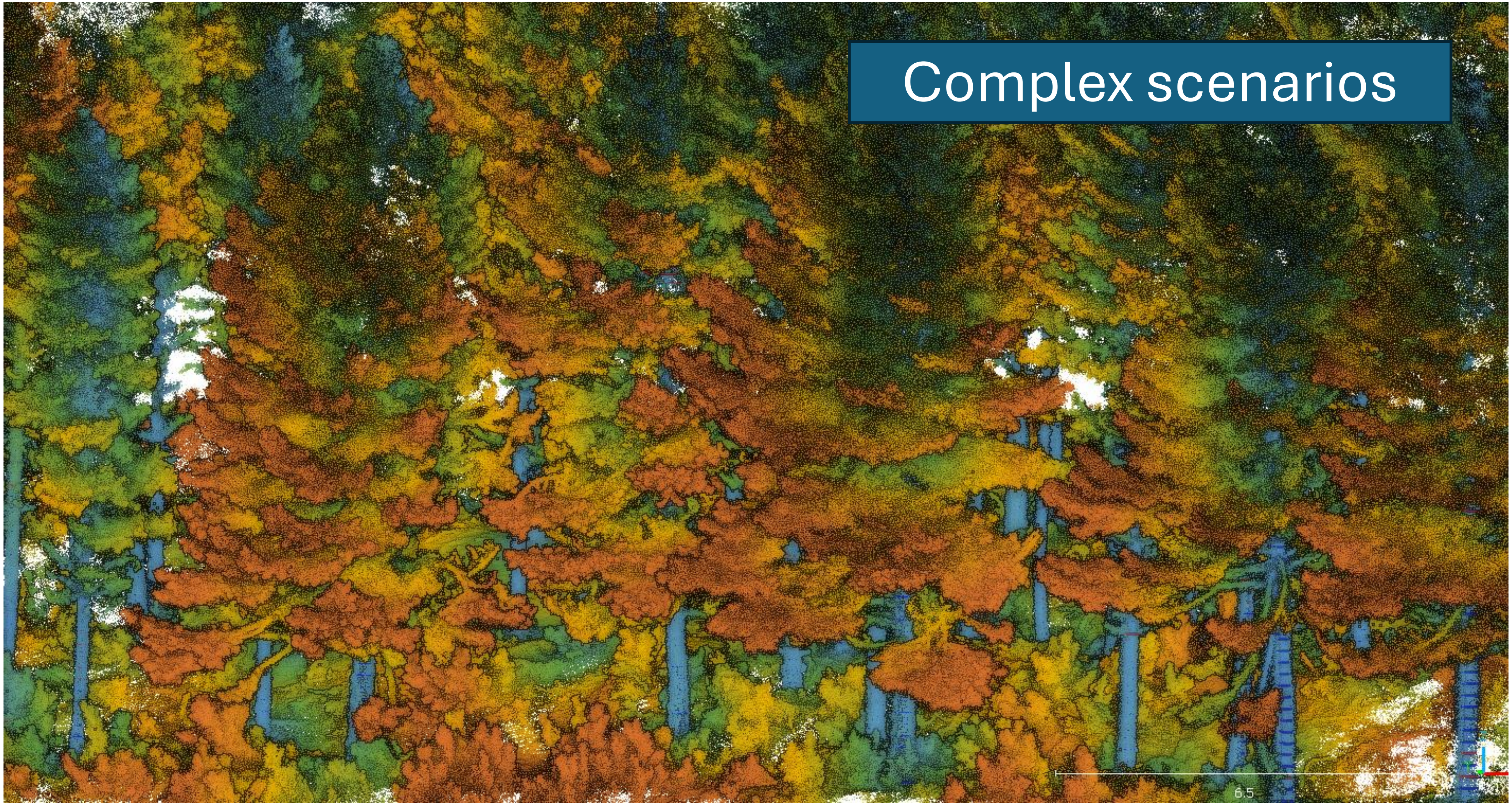

3DFin v0.3.2
Basic
Advanced
Expert
About
Normalize point cloud
Clean noise on dtm
Format of output tabular data
TXT
XLSX
Normalized height field name
Z0
Stripe Upper Limit
3.5
Stripe Lower Limit
0.7
Pruning Intensity
2
Cloth resolution
0.7
3DFin
Forest inventory
3DFin implements our algorithm to detect the trees present in a terrestrial point cloud from a forest plot, and compute individual tree parameters: tree height, tree location, diameters along the stem (including DBH), and tree stem axis.
Please, note that the program is designed to process ground-based data (TLS, PMMS, SfM) and only in rare exceptions it could work with UAV data.
Be sure to check the documentation, which features detailed explanations on how the program works, and the tutorial by Fabian Fassnacht, which is a great tool to get started using 3DFin.
Documentation
Tutorial
Stripe
Original cloud
Height-normalized cloud
Upper limit
Stripe
Lower limit
Region where one should expect mostly stems.
3DFin is able to normalize heights automatically, but also allows using already height-normalized point clouds.
File already set by the application
Select file
Output Directory
C:\Users\Carlos
Select dir
Compute

Example of graphic outputs



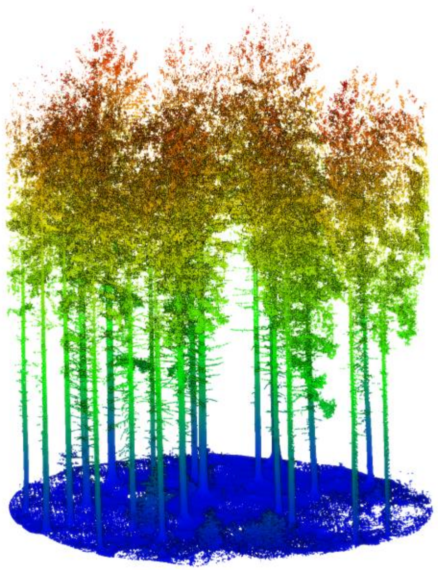


Complex scenarios

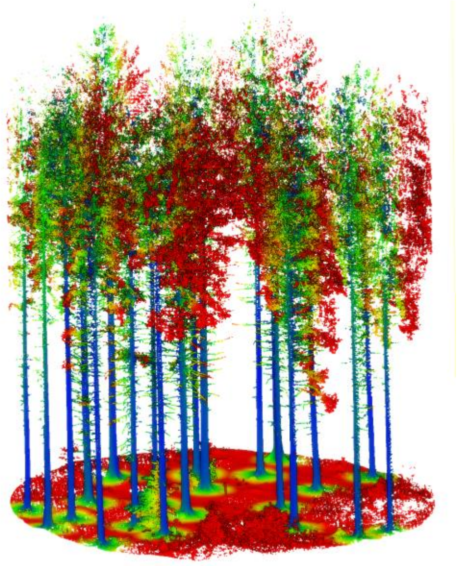


Outputs

‘Enriched’ point clouds

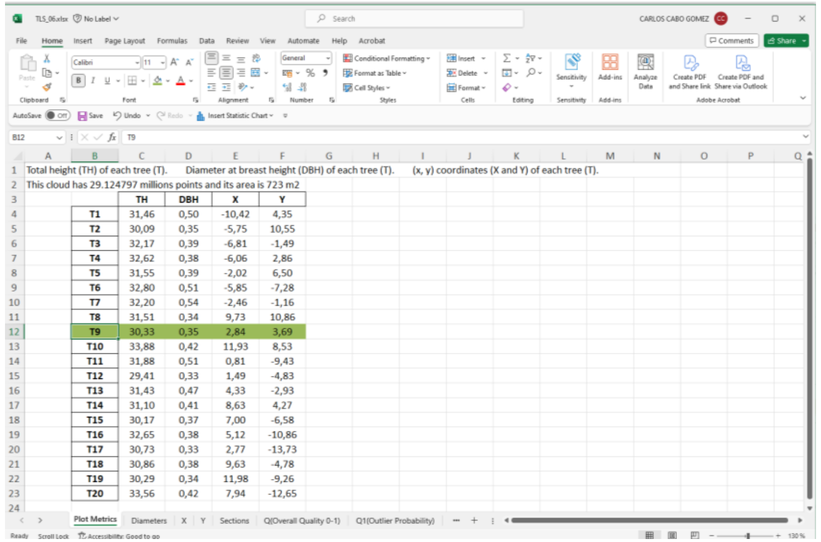


Normalized Point cloud



Distance to axes

Tabular data



Total height (TH) of each tree (T). Diameter at breast height (DBH) of each tree (T). (x, y) coordinates (X and Y) of each tree (T).

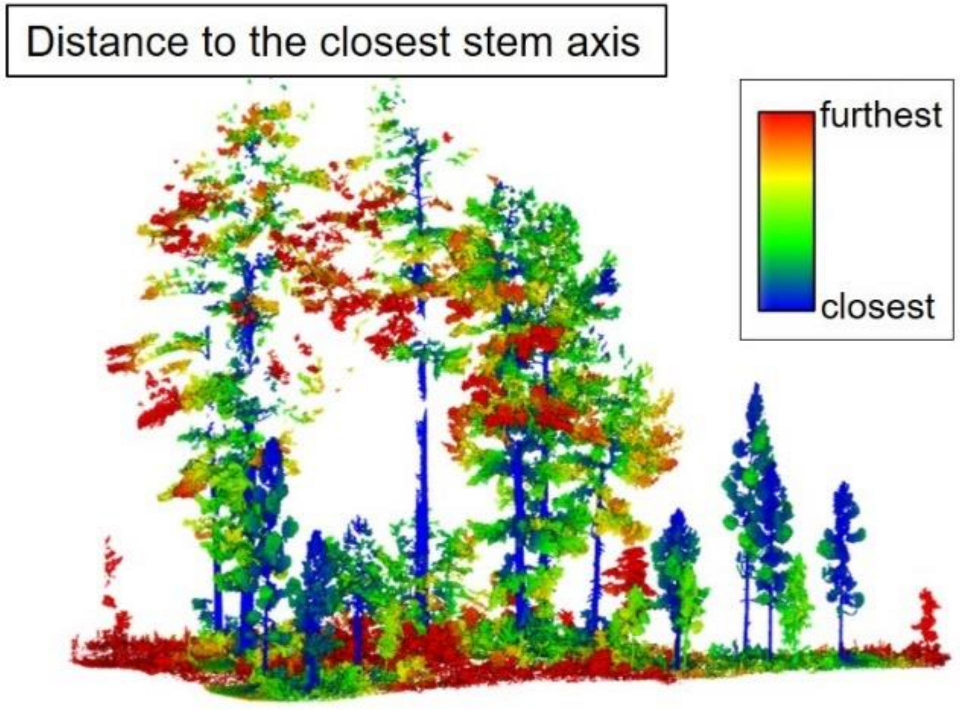
This cloud has 29.124797 millions points and its area is 723 m2

	TH	DBH	X	Y
T1	31,46	0,50	-10,42	4,35
T2	30,09	0,35	-5,75	10,55
T3	32,17	0,39	-6,81	-1,49
T4	32,62	0,38	-6,06	2,86
T5	31,55	0,39	-2,02	6,50
T6	32,80	0,51	-5,85	-7,28
T7	32,20	0,54	-2,46	-1,16
T8	31,51	0,34	9,73	10,86
T9	30,33	0,35	2,84	3,69
T10	33,88	0,42	11,93	8,53
T11	31,88	0,51	0,81	-9,43
T12	29,41	0,33	1,49	-4,83
T13	31,43	0,47	4,33	-2,93
T14	31,10	0,41	8,63	4,27
T15	30,17	0,37	7,00	-6,58
T16	32,65	0,38	5,12	-10,86
T17	30,73	0,33	2,77	-13,73
T18	30,86	0,38	9,63	-4,78
T19	30,29	0,34	11,98	-9,26
T20	33,56	0,42	7,94	-12,65

Plot Metrics | Diameters | X | Y | Sections | Q(Overall Quality 0-1) | Q1(Outlier Probability)

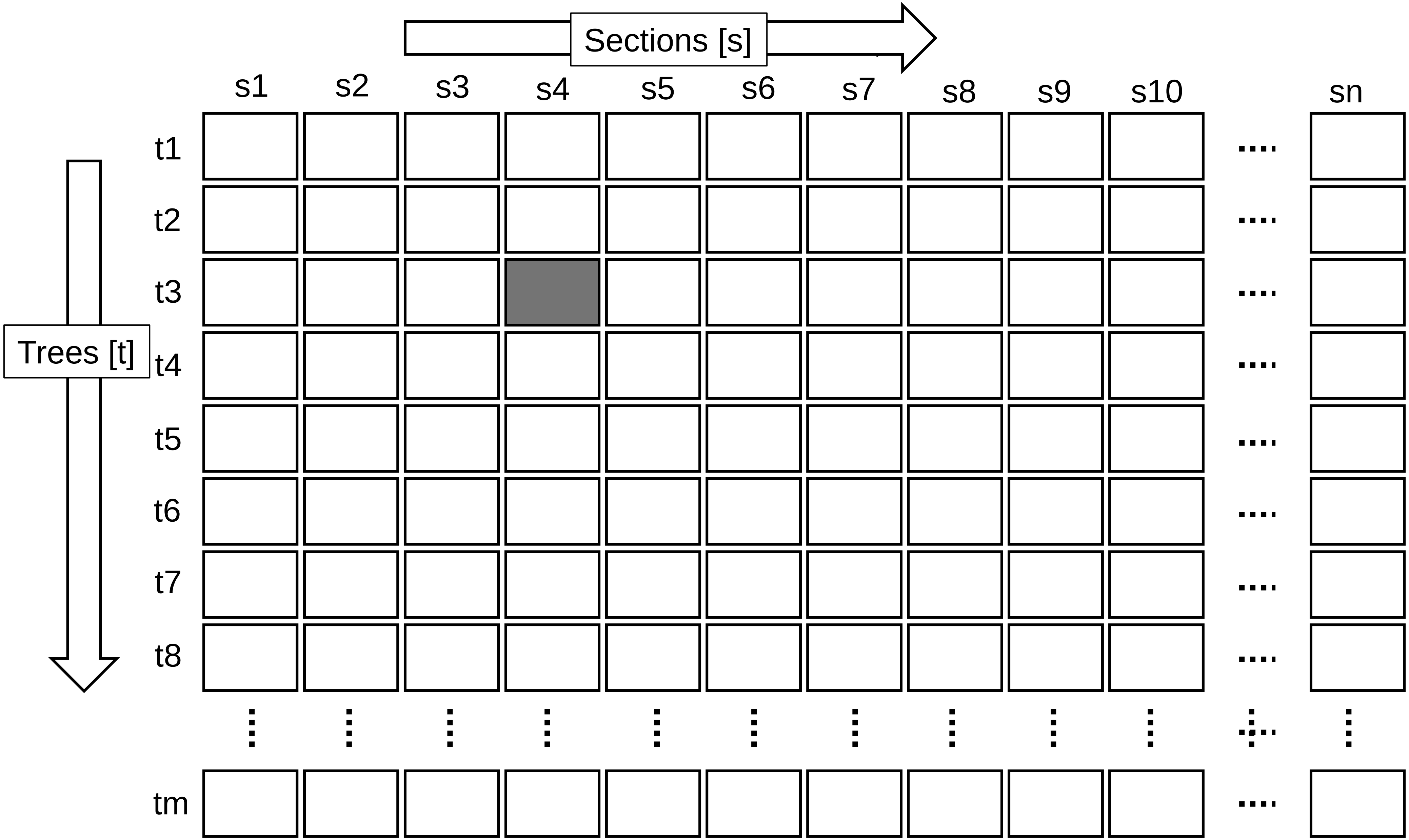
Point cloud outputs

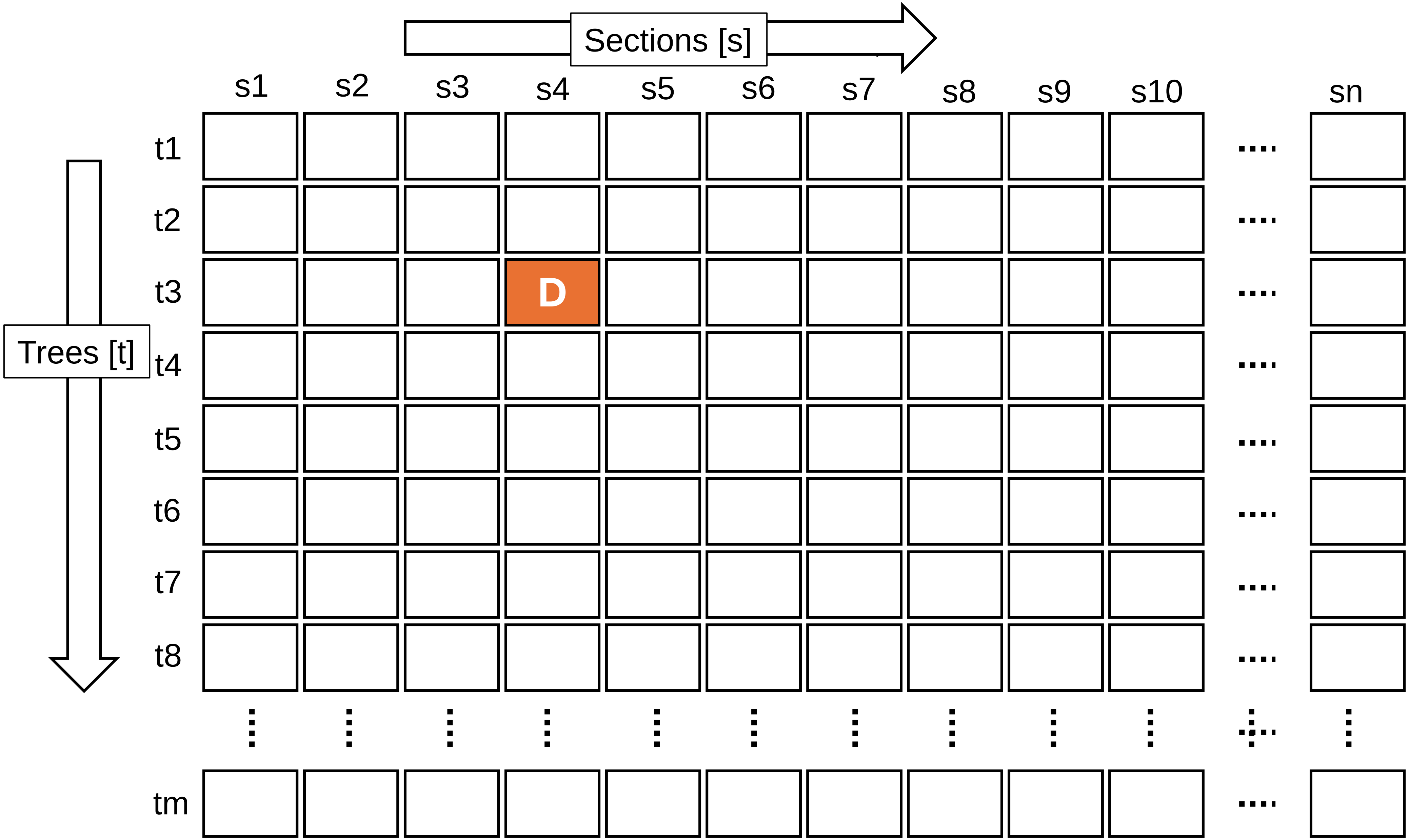
‘Enriched’ point cloud
-aditional data-



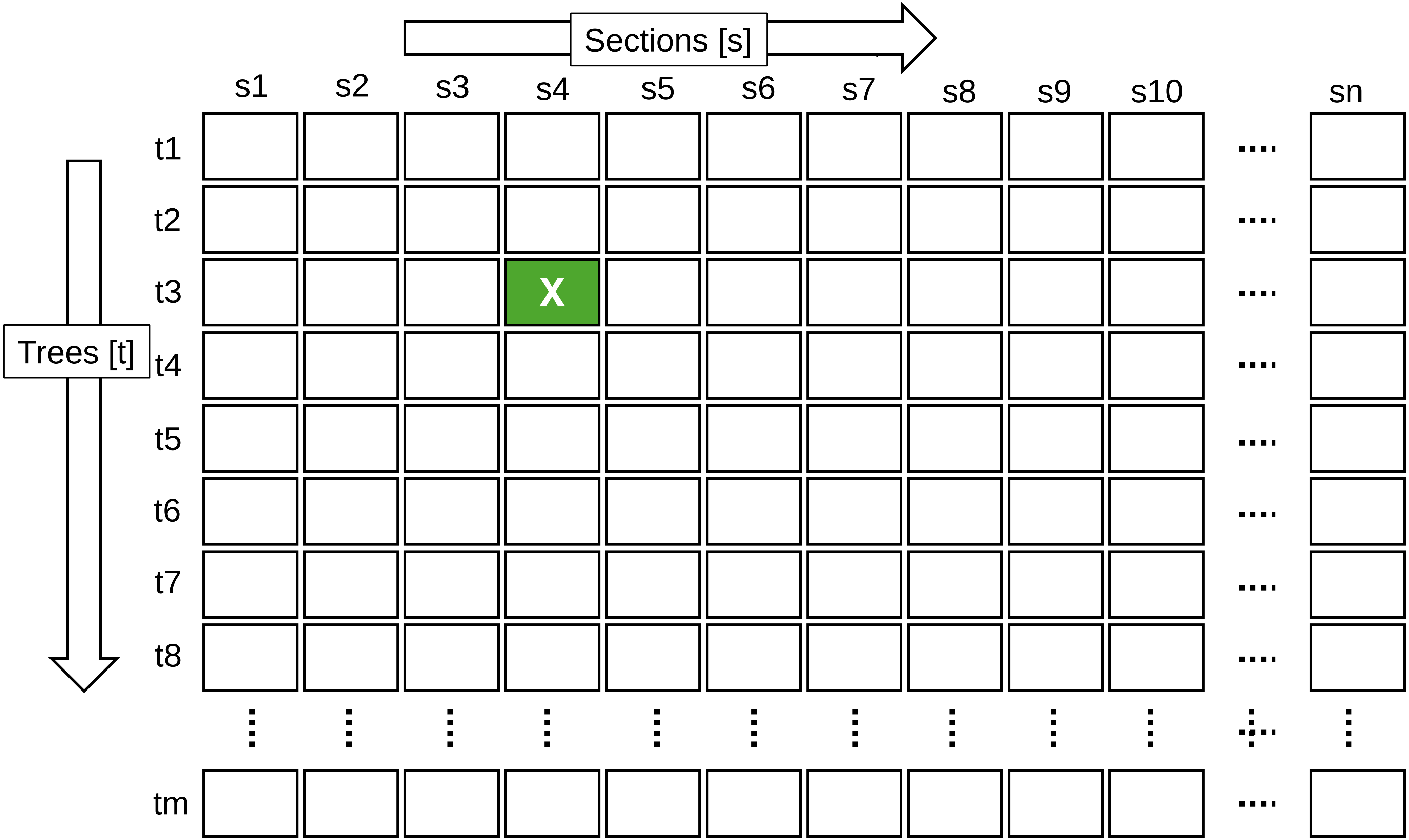
Original point cloud			Added fields		
X	**Y**	**Z**	Norm. **height**	Tree **ID**	**Dist.** closest stem
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-

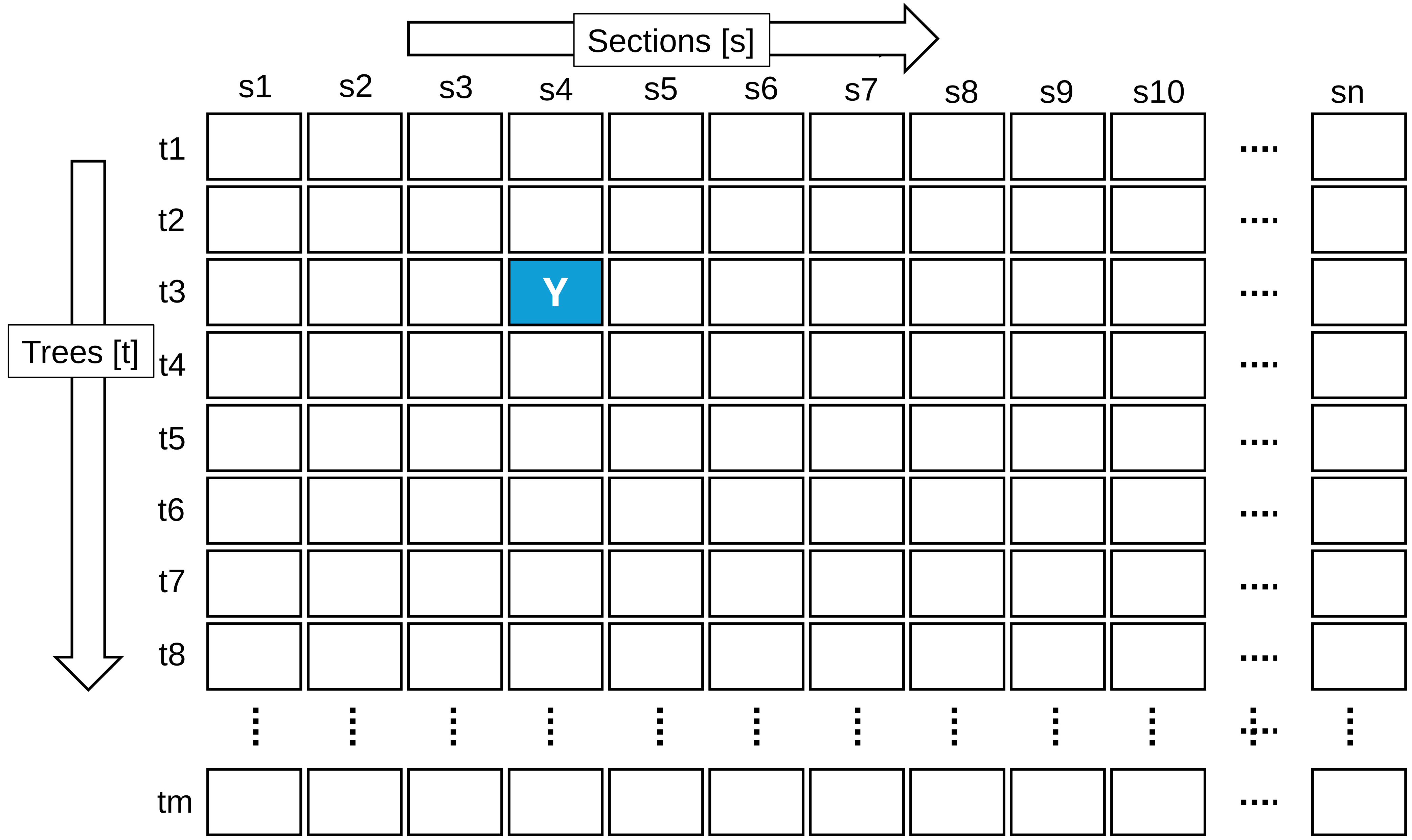
Structure for the files: D, X, Y, Z, Quality



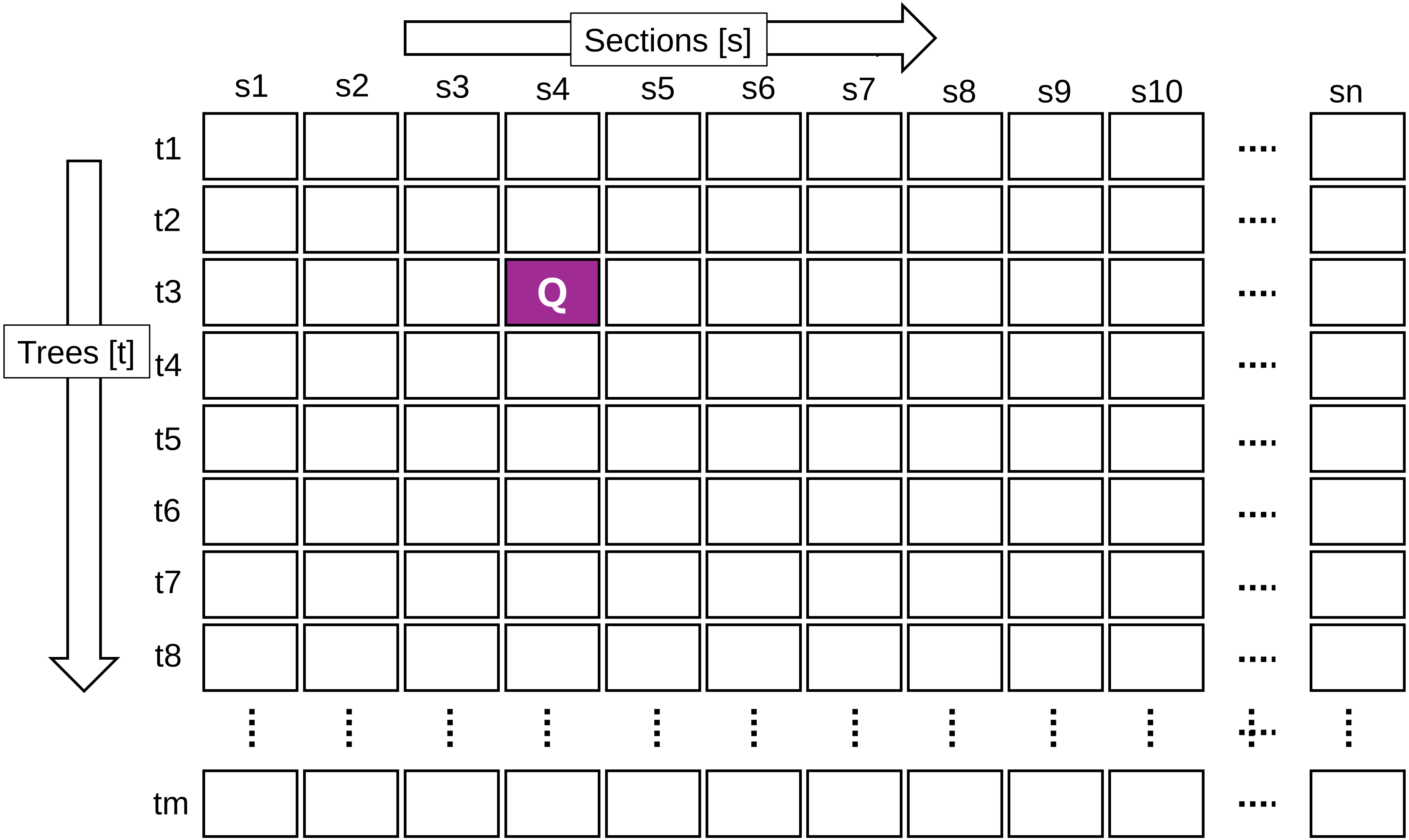


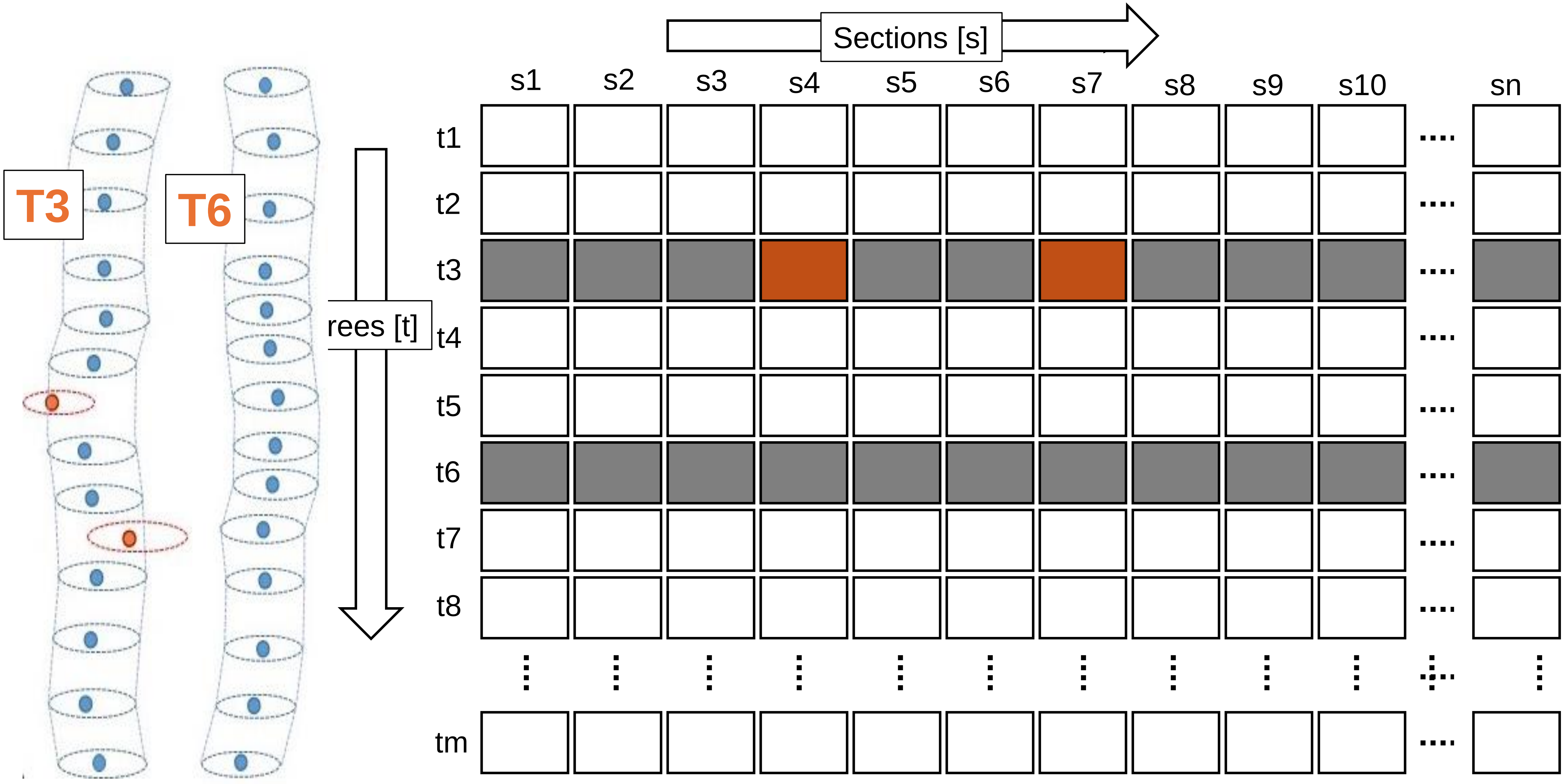
Sections [s]
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
sn
Trees [t]
t1
t2
t3
t4
t5
t6
t7
t8
tm
D


Sections [s]
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
sn
Trees [t]
t1
t2
t3
t4
t5
t6
t7
t8
tm
X



Sections [s]
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
sn
Trees [t]
t1
t2
t3
t4
t5
t6
t7
t8
tm
Y


Sections [s]
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
sn
Trees [t]
t1
t2
t3
t4
t5
t6
t7
t8
tm
Q


Sections [s]
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
sn
T3
T6
rees [t]
t1
t2
t3
t4
t5
t6
t7
t8
tm

Forest inventory
Implementation 3DFin
Experimental **tests**

Silvilaser dataset
25m in radius
Variable density and conditions

10 datasets

- 4 TLS -static-
- 4 MLS -mobile-
- 2 SfM -photogrammetric-

- **30-70 million points**

Forest inventory

Implementation 3DFin

Experimental **results**

Completeness:	**95-100%**
Correctness:	**98-100%**
DBH RMSE/BIAS:	17mm / 10mm
Processing time:	**2-7min**

Are we alone?

*In the universe of forest inventory from terrestrial point clouds

3DFin

Forest inventory

Are we alone?

FORTLS

rTLS

opals

3DFin
Forest inventory

Computree

3D FOREST

R

DendroCloud
point cloud processing software for forestry

MATLAB

LiDAR360
LiDAR360 V7

python™

How to assess and compare them?

COST Action CA20118

3DForEcoTech

Three-dimensional forest ecosystem monitoring & better understanding by terrestrial-based technologies

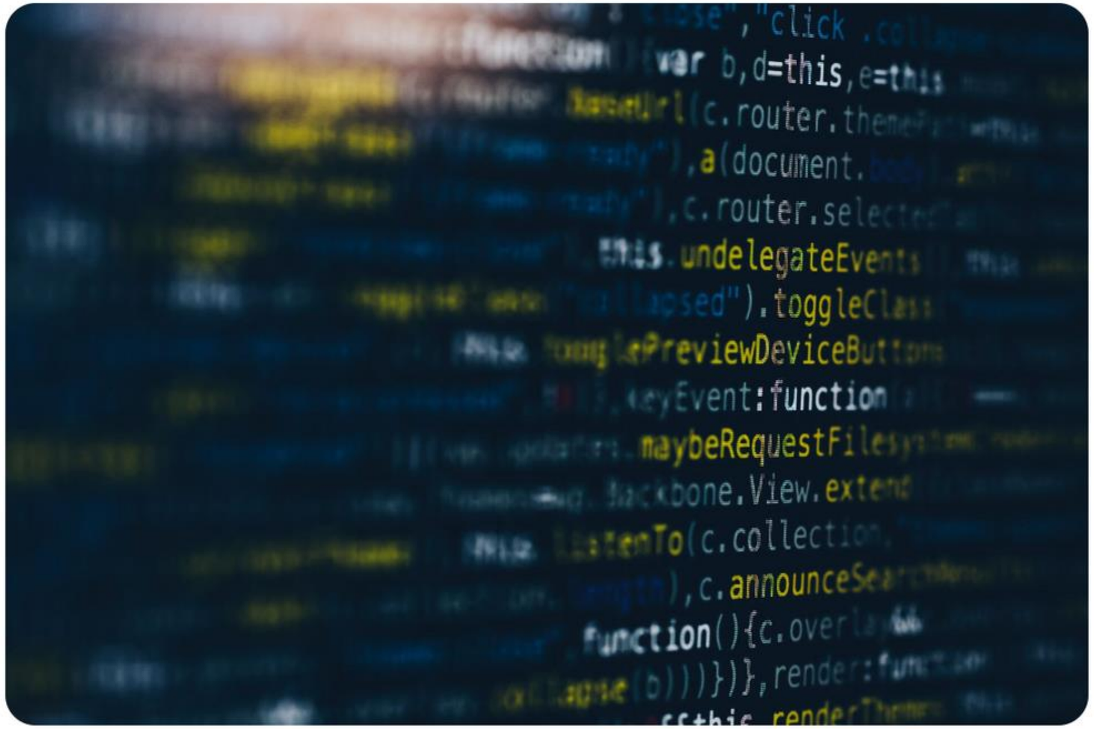
nov**2021** - nov**2025**

500 participants

50 countries

← NEWS

Hackathon: a benchmark of software solutions for processing close-range forest point clouds



In WG3 we are **looking for participants for a benchmark hackathon** for software solutions that process close-range point clouds from forests. It will take place on 25-29 September, 2023, at TU Wien (Austria).

One of the main aims of WG3 is to compile and evaluate algorithms and software that is relevant for 3DForEcoTech. We have already collected all software, algorithms and other solutions for point cloud processing with a focus on forest and individual trees. Altogether we have 24 such implementations; we have been successful at running and testing them, and we are working on an overview scientific paper and webpage focusing on the workflows of all solutions and their potential usage.

With this hackathon, we want to benchmark all the implementations using various point clouds from different European forest ecosystems.During the hackathon we will process them, assess their accuracy, and evaluate their pros and cons with the different available datasets. Based on the whole process, we will also create a standard protocol for the future assessment of other software, algorithms and solutions.

We are looking for participants with skills/experience in:

- point cloud technologies: data collection (desirable) and data processing (required)

Software solutions:

The list used in the hackathon + participant asignments

Hackathon task assignments		
Supervising	**In charge**	**Software**
Participant 2	Participant 1	TreeLS
Participant 3	Participant 2	PointCloudTools
Participant 4	Participant 3	FSCT
Participant 5	Participant 4	treetool
Participant 6	Participant 5	OPALS
Participant 7	Participant 6	FORTLS
Participant 8	Participant 7	LiDAR 360
Participant 9	Participant 8	Computree
Participant 10	Participant 9	3DForest
Participant 11	Participant 10	AID-FOREST
Participant 12	Participant 11	dendrocloud
Participant 1	Participant 12	3DFIN

Benchmark paper to come soon

Gracias!

carloscabo@uniovi.es