Check for updates

Received: 10 January 2023 | Accepted: 2 October 2023

DOI: 10.1111/2041-210X.14233

Methods in Ecology and Evolution

RESEARCH ARTICLE

TLS2trees: A scalable tree segmentation pipeline for TLS data

Phil Wilkes[1,2] | Mathias Disney[1,2] | John Armston[3] | Harm Bartholomeus[4] | Lisa Bentley[5] | Benjamin Brede[6] | Andrew Burt[7] | Kim Calders[8] | Cecilia Chavana-Bryant[1] | Daniel Clewley[9] | Laura Duncanson[3] | Brieanne Forbes[5] | Sean Krisanski[10] | Yadvinder Malhi[11] | David Moffat[9] | Niall Origo[12] | Alexander Shenkin[13] | Wanxin Yang[1]

[1]Department of Geography, University College London, London, UK; [2]NERC National Centre for Earth Observation, Leicester, UK; [3]Department of Geographical Sciences, University of Maryland, College Park, Maryland, USA; [4]Laboratory of Geo-Information Science and Remote Sensing, Wageningen University and Research, Wageningen, The Netherlands; [5]Department of Biology, Sonoma State University, Rohnert Park, California, USA; [6]Helmholtz Center Potsdam GFZ German Research Centre for Geosciences, Section 1.4 Remote Sensing and Geoinformatics, Potsdam, Germany; [7]Sylvera Ltd, London, UK; [8]CAVElab-Computational & Applied Vegetation Ecology, Department of Environment, Ghent University, Gent, Belgium; [9]Plymouth Marine Laboratory, Plymouth, UK; [10]ARC Training Centre for Forest Value, University of Tasmania, Sandy Bay, Tasmania, Australia; [11]University of Oxford, Oxford, UK; [12]Climate and Earth Observation Group, National Physical Laboratory, Teddington, UK and [13]School of Informatics, Computing, and Cyber Systems, Northern Arizona University Flagstaff, Flagstaff, Arizona, USA

Correspondence
Phil Wilkes
Email: p.wilkes@ucl.ac.uk

Funding information
Natural Environment Research Council, Grant/Award Number: NE/N00373X/1, NE/P011780/1 and NE/P012337/1; ESA ForestScan Project, Grant/Award Number: 4000126857/20/NL/AI; ERC-AG, Grant/Award Number: 321121; European Union, Grant/Award Number: 101039795; MetEOC-4; ESA IDEAS-QA4EO, Grant/Award Number: ESA AO/1-11035/21/I-DT; Australian Research Council, Training Centre for Forest Value, Grant/Award Number: IC150100004; NERC Earth Observation Data Acquisition and Analysis Service (NEODAAS); NCEO and UCL Geography

Handling Editor: Oscar Gaggiotti

Abstract

1. Above-ground biomass (AGB) is an important metric used to quantify the mass of carbon stored in terrestrial ecosystems. For forests, this is routinely estimated at the plot scale (typically 1 ha) using inventory measurements and allometry. In recent years, terrestrial laser scanning (TLS) has appeared as a disruptive technology that can generate a more accurate assessment of tree and plot scale AGB; however, operationalising TLS methods has had to overcome a number of challenges. One such challenge is the segmentation of individual trees from plot level point clouds that are required to estimate woody volume, this is often done manually (e.g. with interactive point cloud editing software) and can be very time consuming.
2. Here we present *TLS2trees*, an automated processing pipeline and set of Python command line tools that aims to redress this processing bottleneck. *TLS2trees* consists of existing and new methods and is specifically designed to be horizontally scalable. The processing pipeline is demonstrated on 7.5 ha of TLS data captured across 10 plots of seven forest types; from open savanna to dense tropical rainforest.
3. A total of 10,557 trees are segmented with *TLS2trees*: these are compared to 1281 manually segmented trees. Results indicate that *TLS2trees* performs well, particularly for larger trees (i.e. the cohort of largest trees that comprise 50% of total plot volume), where plot-wise tree volume bias is $\pm 0.4\,m^3$ and %RMSE is 60%. Segmentation performance decreases for smaller trees, for example where

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2023 The Authors. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

DBH ≤10 cm; a number of reasons are suggested including performance of semantic segmentation step.
4. The volume and scale of TLS data captured in forest plots is increasing. It is suggested that to fully utilise this data for activities such as monitoring, reporting and verification or as reference data for satellite missions an automated processing pipeline, such as *TLS2trees*, is required. To facilitate improvements to *TLS2trees*, as well as modification for other laser scanning modes (e.g. mobile and UAV laser scanning), *TLS2trees* is a free and open-source software.

KEYWORDS
above-ground biomass, Forest, FOSS, segmentation, terrestrial laser scanning

1 | INTRODUCTION

Above-ground biomass (AGB) is an important metric that quantifies the amount of carbon stored in terrestrial ecosystems, and as such has been identified as an essential climate variable (ECV). However, accurate quantification of forest AGB is a significant and ongoing challenge owing to a number of factors, including systematic errors when applying allometry to inventory data (Burt et al., 2020). To improve the accuracy of AGB quantification in forests, groups such as the International Panel on Climate Change (IPCC) and the Committee on Earth Observing Satellites (CEOS) have identified terrestrial laser scanning (TLS) as a disruptive technology (Duncanson et al., 2019; Ogle et al., 2019).

TLS is an active remote sensing technology that generates a detailed 3D point cloud of the surrounding area with centimetre to millimetre accuracy (Newnham et al., 2015; Calders et al. in Duncanson et al., 2019). A TLS-based approach has been used to estimate AGB across a range of forest types (Demol, Verbeeck, et al., 2022), for example; tropical forests (Beyene et al., 2020; Brede et al., 2022; Burt et al., 2021; Gonzalez de Tanago et al., 2018; Levick et al., 2021; Momo Takoudjou et al., 2018), temperate forests (Calders et al., 2015, 2022; Disney et al., 2020; Stovall et al., 2017), mangroves (Feliciano et al., 2014) and trees outside forests (Kükenbrink et al., 2021; Van Den Berge et al., 2021; Wilkes et al., 2018). Over the past decade, area scanned has increased from a few trees to systematic acquisition across multiple hectares that replicates forest inventory protocols (Wilkes et al., 2017).

The benefits of TLS based methods are that they are non-destructive, capture tree plasticity particularly of large trees (Burt et al., 2021) and are (mostly) free from errors and assumptions associated with allometric modelling (Chave et al. in Duncanson et al., 2019). When compared to destructive harvest, TLS has been shown to be more accurate than the application of existing allometries, particularly for larger trees that contribute disproportionately to plot-level AGB (Demol, Verbeeck, et al., 2022). However, operationalising a plot-level (i.e. ≥1 ha) TLS workflow as a "turn-key" solution to produce an AGB product has yet to be fully realised (Martin-Ducup et al., 2021).

A TLS survey of a forest plot to estimate AGB typically involves: (1) capturing scan data from multiple fixed positions across a plot, (2) co-register scans to produce a single plot-level point cloud P, (3) instance segmentation of P into a set of point clouds that represent individual trees S, (4) semantic segmentation of $s \in S$ into wood and leaf point classes, (5) estimate of woody volume for $s \in S$, for example using a Quantitative Structure Model (QSM) approach, and (6) conversion of volume to AGB via an estimate of wood basic density.

Steps 1–2 have been largely been solved with improvements in scanner technology and standardised scanning protocols (Calders et al., 2020; Wilkes et al., 2017). There are also a number of existing methods for semantic segmentation (Vicari et al., 2019; Wang et al., 2020) and QSM generation, where $s \in S$ are enclosed in a geometric primitives, for example a set of cylinders (Hackenberg et al., 2015; Raumonen et al., 2013; Stovall et al., 2017). Conversion of volume to mass is an ongoing challenge for all non-destructive methods of AGB estimation as wood density varies greatly within and between trees and geographic regions (Phillips et al., 2019). Therefore, it is suggested, that from a TLS workflow perspective at least, the most significant remaining challenge to operational plot level AGB estimation is the instance segmentation in step 3, that is $S \subset P$.

P is encoded with geometric information of whole-tree structure, for all trees in the surveyed area regardless of size. If P is of sufficient density and quality then S can be accurately segmented and an unbiased assessment of plot-level AGB, with associated uncertainties, can be produced (Burt et al., 2021; Calders et al., 2015; Momo Takoudjou et al., 2018). Currently, the most accurate method for $S \subset P$ is to manually segment individual trees from their neighbours and other vegetation using interactive point cloud editing software (Brede et al., 2022; Disney et al., 2020; Gonzalez de Tanago et al., 2018; Momo Takoudjou et al., 2018). Manually segmenting trees increases tree- and plot-level estimation accuracy by a factor of 10 and 3 respectively, compared to existing automated TLS pipelines (Martin-Ducup et al., 2021). However, manual segmentation can be very time consuming (10's of minutes per tree) as well as subjective and difficult to reproduce or validate. These factors have limited the number (and provenance) of segmented trees, where total

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

segmented trees is limited to 10's (and only in rare cases 100's) of trees per hectare.

Automated whole-tree instance segmentation methods have been previously demonstrated (Burt et al., 2019; Krisanski, Taskhiri, Gonzalez Aracil, Herries, & Turner, 2021; Martin-Ducup et al., 2021; Tao et al., 2015; Wang et al., 2021). Tao et al. (2015) presented a method where a graph is constructed through clustered P, then using a Dijkstra shortest-path method (Dijkstra, 1959) clusters are mapped to a base node resulting in segmented S. They applied their method to TLS data collected in three forest patches and reported high accuracy for number of trees correctly identified and completeness of extracted trees. Subsequent methods have built upon these graph based methods and have been applied in different forest types such as Schelorphyll, coniferous and tropical forests (Martin-Ducup et al., 2021; Wang et al., 2021). Burt et al. (2019) presented a rule constrained clustering with proximity testing to segment trees in a tropical forest plot; this method also requires an allometric assumption of DBH to crown height and extent. The *Forest Structure Complexity Tool* (*FSCT*) (Krisanski, Taskhiri, Gonzalez Aracil, Herries, Muneri, et al., 2021) is an end-to-end workflow that generates plot level estimates of structure parameters from laser scanning data. *FSCT*s innovative approach combines deep-learning powered semantic segmentation which is followed by a rule constrained cluster based instance segmentation. These methods have shown promise in demonstration plots, this paper aims to build on these and present a method that is scalable and is applicable across forest types.

There are a number of challenges that can hinder achieving sufficient quality of $S \subset P$ with an automated workflow (Demol, Verbeeck, et al., 2022). These include factors attributable to scanning protocol and scanner specification, such as scanner type and optics (Calders et al., 2020; Newnham et al., 2015), sufficient sampling density to minimise occlusion (Wilkes et al., 2017), co-registration accuracy and co-alignment errors (Demol, Wilkes, et al., 2022) and post-processing computational constraints, for example large data volumes (data volumes are typically $>65\,\mathrm{Gb\,ha^{-1}}$). Further, automated segmentation also needs to be sensitive to forest demography and composition where trees in a plot can range in size (height, diameter, etc.) over orders of magnitude, neighbouring crowns may intersect, parasite species such as lianas could be present (Moorthy et al., 2019) and understorey vegetation maybe dense.

To address the interoperability and scalability challenges of automated tree segmentation from plot-level point clouds, we present *TLS2trees*. *TLS2trees* is a set of Python command line tools that are free and open-source software (FOSS) and are specifically designed to be horizontally scalable (Kissling et al., 2022), for example, on a High Performance Computing (HPC) facility. The output is a set of segmented point clouds of individual trees, where points are classified into leaf and wood components. Below, *TLS2trees* is demonstrated at 10 forest plots that cover seven forest types; from open savanna to dense tropical forest. Additionally, segmented point clouds are compared to a set of 1053 manually segmented reference trees.

2 | MATERIALS AND METHODS

2.1 | Data acquisition

The TLS data used here were captured from 10 plots across seven different forest types (Table 1). At each plot, data were acquired with a RIEGL VZ-400 scanner (RIEGL Laser Measurement Systems GmbH, Horn, Austria) where a set of scans I were captured on a regular grid (Kissling et al., 2022) or, for RUSH plots, in a star formation (Calders et al., 2015). At each scan position two scans were acquired where the scanner rotation axis was orientated perpendicular then parallel to the ground surface. Manually-placed reflectors were used as tie points between each scan position to aid co-registration. Post-processing was done using RiSCAN Pro software (versions 2.1-2.9) where individual scans were co-registered to a common coordinate system on a per plot basis. Once registration was complete, a set of 4×4 transformation matrices M were exported for each plot.

2.2 | TLS2trees

The *TLS2trees* software package consists of a set of Python command line tools. *TLS2trees* can be considered Free and Open Source Software (FOSS) and is licensed under Creative Commons BY 4.0 (https://creativecommons.org/licenses/by/4.0/). Processing is horizontally scalable, that is can be scaled across multiple computing nodes (Kissling et al., 2022) where this is achieved using a tile-based workflow. The workflow is also modular so new or additional methods can be added or existing steps replaced. For more information and code see Wilkes et al. (2023).

This section presents a detailed description of the workflow. As is shown in Figure 1, a coregistered global point cloud P is passed through the *TLS2trees* pipeline to generate a set of segmented tree point clouds S. The pipeline is presented in three steps: (1) pre-processing (Section 2.2.1), (2) semantic segmentation (Krisanski, Taskhiri, Gonzalez Aracil, Herries, & Turner, 2021) where points are classified into different components (Section 2.2.2), and (3) an instance segmentation where P is segmented to a set of individual trees S (Section 2.2.3). Once $S \subset P$, structure traits, such as total woody volume, are computed via TreeQSM (Section 2.4). Exposed workflow parameters for each step are listed in the Appendix 0. It should be noted that, although data processed for this manuscript was only captured with a RIEGL scanner, the workflow can be adapted to other scanner types, for example Wielgosz et al. (2023).

2.2.1 | Step 1. Preprocessing

The workflow starts at the point of a set of individual scans I and corresponding rotation matrices M; the first step is $(I, M) \mapsto P$. During $(I, M) \mapsto P$, I are clipped to the plot extent determined by M, plus a 10 m buffer (Martin-Ducup et al., 2021). The buffer is required to capture the crowns of trees that have germinated inside the plot but

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 TLS plots.

Plot (subplot)	Year	Country	Forest type	Stem density ha^{-1} (>10 cm dbh)	Area (ha)	Scan pattern	#Trees manually segmented	Citation
RUSH	2012	Australia	Dry sclerophyll Box-Ironbark forest	221			65	Calders et al. (2015)
RUSH06				220[a]	0.3	5 scans	34	
RUSH07				222[a]	0.3	5 scans	31	
WYT	2015	UK	Temperate mixed deciduous (leaf-off)	379	1	20 m grid	559	Calders et al. (2018, 2022)
NOU	2016	French Guiana	Lowland tropical rainforest	501[b]	1	10 m grid	155	Burt et al. (2019)
KOG	2016	Ghana	Guinean woody savanna	193[c]	1	20 m grid	118	Unpublished
CALI	2018	USA	Coastal Redwood	168–996[d]			156	Disney et al. (2020)
CALI-01				–	1	10 m grid	25	
CALI-02				–	0.3	12.5 m grid	43	
CALI-07				–	1	10 m grid	88	
MLA	2019	Malaysia	Tropical lowland dipterocarp forest	403[c]	1	10 m grid	199	Unpublished
AEP	2019	Australia	Disturbed complex notophyll vine forest	1015[e]	0.6	10 m grid	29	Unpublished

[a]Murphy et al. (2014).
[b]Chave et al. (2008).
[c]Malhi et al. (2021).
[d]Fujimori (1977).
[e]Graham (2006).

where the crown overhangs the plot boundary. P is then projected onto a 10 m × 10 m grid to produce a set of tiled point clouds T where individuals tiles $t \in T$ are the processing unit for subsequent steps. A tile index of T is also generated to map the spatial location of neighbouring tiles.

As is inherent with TLS data, objects closer to the scanner are over sampled whereas objects further away (e.g. the top of the canopy, or at the edge of a scanned region) can be undersampled (Burt et al., 2019). To mitigate the impact of this difference and to reduce file size, P is downsampled to a common point density using PDAL's *Voxel Center Nearest Neighbour* method (PDAL Contributors, 2020) where here a voxel edge length of 0.02 m is used.

2.2.2 | Step 2: Semantic segmentation

Semantic segmentation is the process of labelling points into homogeneous groups; here this is into classes of different biophysical components, for example ground, leaf, wood and so forth. Previous workflows have classified points simultaneously with or after instance segmentation (Vicari et al., 2019; Wang, 2020; Wang et al., 2020). However, instance segmentation can be hampered by the presence of leaves that reduce gaps between tree crowns, or cause neighbouring crowns to intersect.

Here we use the *FSCT* semantic segmentation method (Krisanski, Taskhiri, Gonzalez Aracil, Herries, & Turner, 2021) prior to instance segmentation (Krisanski, Taskhiri, Gonzalez Aracil, Herries, Muneri, et al., 2021). The *FSCT* semantic segmentation uses the Pointnet deep-learning method (Qi et al., 2017) applied via Python's PyTorch. Using a pre-trained model, P is classified into 4 classes; ground (G), woody (W), leaf (L) or coarse woody debris (X) (Figure 1b and Equation 1)

$$P = P_G + P_W + P_L + P_X \quad (1)$$

The semantic segmentation is applied to $t + t_b \in T$, b refers to neighbouring tiles used to generate a 5 m buffer that mitigates any edge effect; once the semantic segmentation is complete only the class label for t are retained. It should be noted that the pre-trained model has not been modified since its initial release by Krisanski, Taskhiri, Gonzalez Aracil, Herries, and Turner (2021), that is the model is not trained for forest types specific to Table 1.

2.2.3 | Step 3: Instance segmentation

Instance segmentation is the process of identifying and segmenting individual trees $S = \{s_1, s_2, \ldots, s_k\}$ encoded in P, that is $P_{W+L} \mapsto S$. Here, a new two step process is presented where (1) P_W are grouped into a set of individual woody stems S (Figure 3c), then (2) P_L are assigned to $s \in S$ (Figure 3d).

Both steps use a Dijkstra's shortest path method (Dijkstra, 1959) where a graph $G = (N, E)$ is constructed; N are a set of nodes and

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

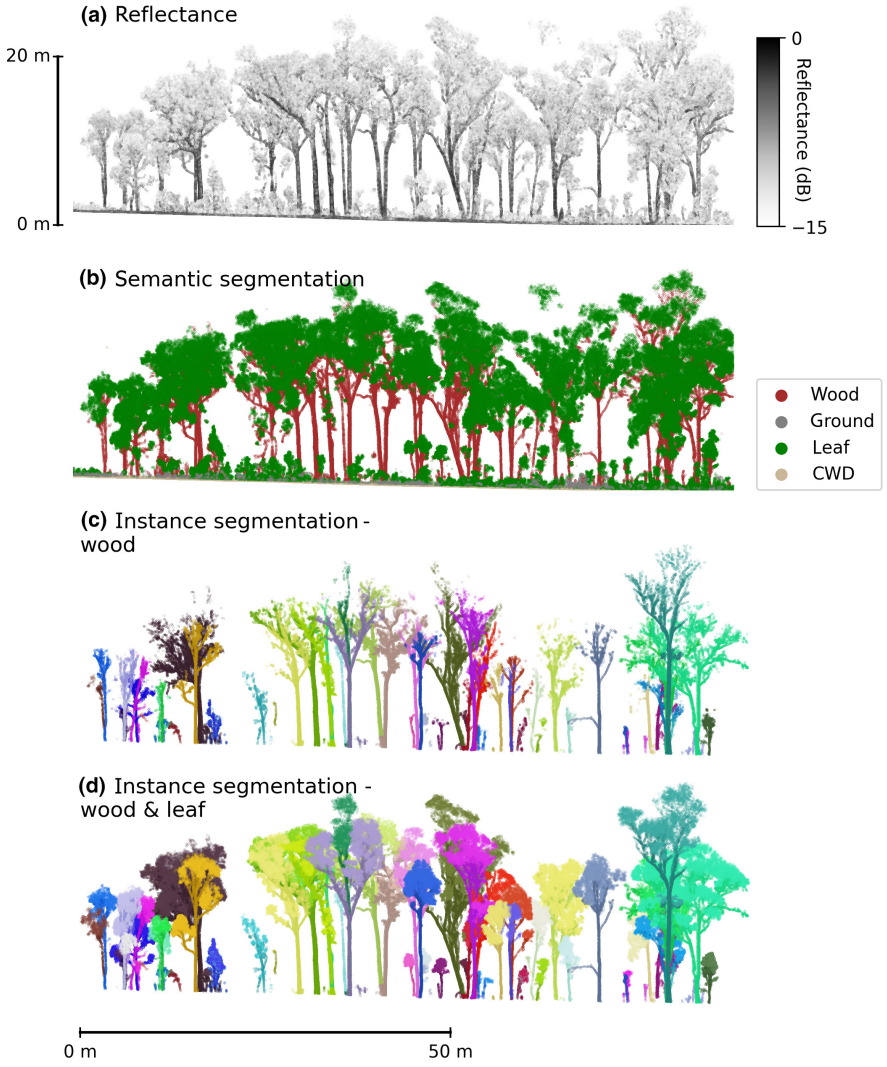


FIGURE 1 *TLS2trees* applied to a 70 m × 20 m strip of forest in plot RUSH where (a) *P* is coloured by calibrated reflectance, (b) semantic segmentation into four classes using a method adopted from *FSCT* Wielgosz et al. (2023), (c) instance segmentation where only points classified as wood are displayed and segmented trees are coloured randomly and (d) instance segmentation where leaf points are attributed to individual stems.

E are a set of edges that connect *N*. A path *p* can be defined that connects two nodes $(n, n_k) \in N$, where $p = [n, n_1, n_2, \ldots n_k]$; for each (n, n_k) pair there are multiple solutions to *p*. To determine the shortest path between the pair (n, n_k) a weight function $w: E \rightarrow \mathbb{R}$ is defined (Equation 2; Tao et al., 2015).

$$w(p) = \sum_{i=1}^{k} w(n_{i-1}, n_i) \quad (2)$$

Previous methods have constructed graphs where *N* is a set of vertices in $\mathbb{R}^3$ and *w* are the Euclidean distances between *N* (Brede et al., 2022; Tao et al., 2015; Vicari et al., 2019; Wang et al., 2021). Therefore the shortest path $min\{w(p): n \rightarrow n_k\}$ is analogue to a vascular system (Tao et al., 2015; Figure 2 left). However, experimentation found that if one or more tree crown envelopes intersect, the shortest path could be to the base of a smaller suppressed tree, resulting in a poor instance segmentation (Figure 3). To solve this, a different approach is taken to compute *w*.

To define a graph G_W for wood classified points P_W, a set of nodes N_W are first generated. To do this, P_W is sliced horizontally (relative to ground normalised height calculated from P_G) at intervals of 0.2 m; slicing is required to regularise the distance between clusters. Then for each slice, DBSCAN (Ester et al., 1996) is used to map P_W to a set of clusters C_W (Figure 2). DBSCAN parameters are dependent on acquisition and point cloud characteristics; after downsampling an eps value of 0.1 m and minimum_sample of 20 points are used here.

For each cluster $c \in C_W$, a convex hull is computed generating a set of hull vertices V_W where $V_W \subset C_W \subset P_W$ (Figure 2 right). A convex hull retains information on the occupation of space and proximity to neighbouring clusters, whereas collapsing a cluster to a single vertex does not. A *k*-nearest neighbour search is then performed on V_W to identify vertices in neighbouring clusters; within cluster connections are disregarded. The Euclidean distance where $\min\left(\text{dist}\left(V_{W_{c_x}}, V_{W_{c_y}}\right)\right)$ is used as the edge weight function *w* (Figure 2 right). $w(p)$ is therefore determined by distance between cluster edges (i.e. connectivity) as opposed to distance between cluster centroids (i.e. path length). A parameter is available to set allowable maximum distance between $V_{W_{c_x}}$ and $V_{W_{c_y}}$ (Appendix 0).

Dijkstra shortest path analysis requires a subset of source nodes $n_b \subset N_W$ from which to calculate distance from. Here n_b are generated by taking a slice through C_W $[z_1, z_2]$; z_1 and z_2 are upper and lower bounds of the slice relative to ground height and the median height

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

of $c \in C_W$ is used to determine intersection. RANSAC cylinder fitting is then used to identify stem bases from noise such as small branches or misclassified leaves (Burt et al., 2019). It is important to attempt to identify *all* stems, regardless of diameter, otherwise smaller stems can be erroneously included into larger neighbouring stems; here a lower limit of $\emptyset = 0.05$ m is used for stem detection.

Once the graph G_W is generated, the Dijkstra shortest path method is used where multiple source nodes n_b can be defined. The output of the shortest path analysis is a disjoint union of undirected acyclic graphs. Then $N_W \mapsto C_W$ and therefore P_W can be mapped to a set of individual stems S.

Once step 1 is complete, another graph G_L is created that maps leaf classified points P_L (and P_W that are not attributed to a stem) to $s \in S$. N_L are computed by first mapping P_L to a set of voxels C_L with an edge length of 0.5 m; C_L is equivalent to the clusters C_W above. A set of 6 vertices are generated for $c \in C_L$ where a vertex is assigned to the centroid of each voxel facet; this results in V_L which are equivalent to V_W above. E_L and corresponding w are determined by calculating the Euclidean distances to neighbouring $v \in V_L$. A set of source nodes $n_b \subset N_W$ are taken from G_W where $n \in N_W$ have no children, that is n is a branch tip; n_b are already associated with $s \in S$. Shortest path analysis is again used to connect N_L with n_b and therefore assign P_L to $s \in S$.

To run the instance segmentation, each $t \in T$ is buffered by neighbouring tiles (see Table 2 for example buffer size). Once complete, S is pruned so that only trees whose germination point is within t are retained; further, a first-pass filter to remove trees where DBH <0.1 m (as determined by the RANSAC cylinder fitting) is applied.

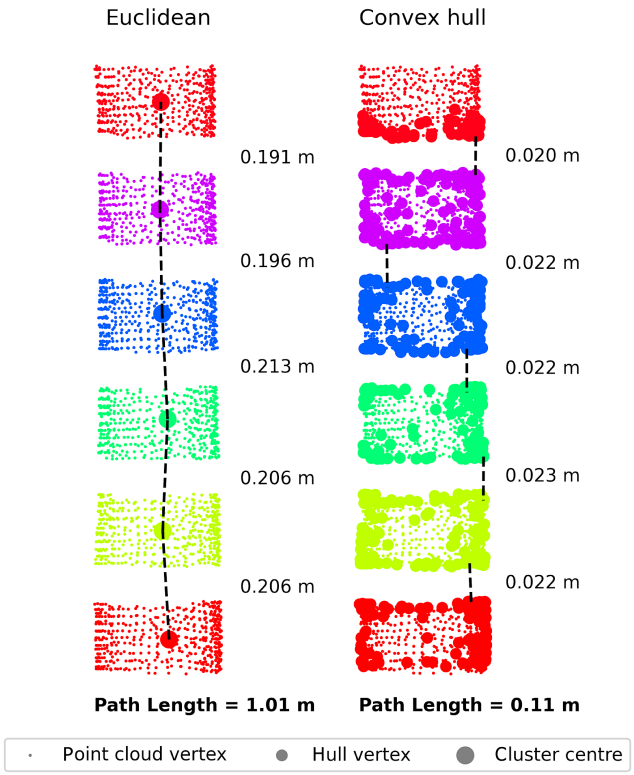


FIGURE 2 A comparison of methods for the construction of graphs through point clouds. Displayed are clusters (vertically orientated) along a stem, adjacent clusters have been exploded so connections between clusters are clearly visible. For the "Euclidean" method, edge weights (values to the right) are calculated as the Euclidean distance between cluster centres; this is synonymous with path length (Tao et al., 2015). For the "Convex hull" method, edge weights are calculated as the distance between the vertices of neighbouring convex hulls (i.e. connectivity).

2.3 | Manually segmented trees

A total of 1281 trees have been manually segmented in previous studies (Table 1). Methods for selecting trees to manually segment from P differed for each plot. For RUSH plots, trees were selected by the Victorian State Government to update statewide AGB allometry, where a subset of trees were harvested across a range of sizes and species (Calders et al., 2015; Murphy et al., 2014). For WYT, all trees within a central 1 ha plot were segmented from a larger 6 ha scanned area, this reduces edge effects inherent at other plots. Individual tree point clouds were also split into >1 tree if bifurcation occurred <1.3 m (Calders et al., 2018). All trees were segmented from NOU where DBH >0.2 m (Burt et al., 2019) and KOG where DBH >0.1 m. For AEP and MLA plots, tree species that comprised 80% of total basal area were selected and a 2–3 trees from each species were segmented (Shenkin et al., 2020). It should be noted no trees have been specifically manually segmented or modified for this manuscript, that is trees were segmented before the inception of this method.

Trees from plots RUSH, MLA, AEP and KOG were manually segmented from the plot-level point cloud using either RiSCAN Pro or CloudCompare v2.X (https://www.danielgm.net/cc/). Data from WYT (Calders et al., 2018, 2022), NOU (Burt et al., 2019) and CALI (Disney et al., 2020) were first segmented with *treeseg* (Burt et al., 2019), after which trees were modified manually, for example removing overlapping crowns. During this process all trees were manually verified for commission and omission errors by an experienced operator.

After segmentation, all tree point clouds underwent a semantic segmentation into leaf and wood points using the *TLSeparation* Python package (Vicari et al., 2019); the exception being WYT where data were captured in leaf-off conditions. Using the wood classified point cloud only, per tree structural traits were then modelled using *TreeQSM* (see Section 2.4).

2.4 | Quantitative structure models

Quantitative Structure Model (QSM) methods enclose the wood-only point clouds in a set of geometric primitives, for example a cylinder. This allows for the estimation of morphological and topological traits such as volume, length and surface area metrics (Raumonen et al., 2013). *TreeQSM* (version 2.3.1, Raumonen, 2019) is used to generate a QSM for all manually and automatically segmented trees.

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

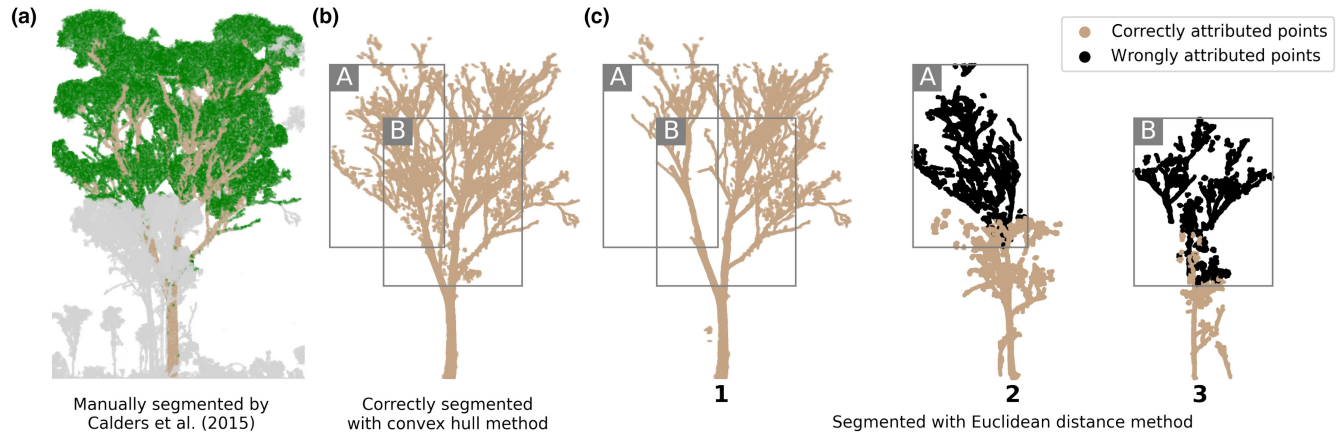


FIGURE 3 A tree segmented using (a) manual segmentation by Calders et al. (2015) (b) segmented using the convex hull method of *TLS2trees* and (c) segmented using a Euclidean distance method. Grey boxes A and B highlight branches that have been wrongly attributed to Trees 2 and 3, respectively [black points in (c)]. Trees 2 and 3 (c) can be seen at the base of the manually segmented tree [(grey points in (a)]. Leaf points identified in Panel a using *TLSeparation* (Vicari et al., 2019).

Since the version of *TreeQSM* and iterated parameter space may differ from previously published versions, it should be noted that modelled values for manually segmented trees may also differ.

A set of *TreeQSM* parameters control the overall fit of cylinders, following Raumonen et al. (2013) three parameters are iterated over here; PatchDiam1 = $[0.20, 0.22, \ldots, 0.3]$, PatchDiam2Min = $[0.5, 0.7, \ldots, 0.15]$ and *PatchDiam2Max* = $[0.15, 0.17, \ldots, 0.25]$. *TreeQSM* is run in Octave (Eaton et al., 2020) where, for each parameter set permutation 5 models are generated. This results in a total of 625 models per segmented tree. An optimal model is then selected by minimising the point to cylinder surface distance (Burt et al., 2019; Martin-Ducup et al., 2021).

2.5 | Comparing tree pairs

To assess the accuracy of segmentation, a corresponding pair of trees is identified in the manually and *TLS2trees* segmented sets. Pairs are identified by firstly taking a slice through all segmented trees between 2 and 3 m and computing the centroid of the slice. Tree pairs are then matched by selecting a tree from the *TLS2trees* data set that mostly closely matches the position of a manually segmented tree. Trees where more than one target tree is within 1 m of the reference tree or where a match is >2 m from the reference tree are disregarded from further analysis.

For matched tree pairs, QSM and point cloud metrics are compared. QSM metrics include total woody volume (m^3), trunk volume (m^3), total branch length (m) and *DBH* (m). Point cloud metrics include leaf-on crown height (m), leaf-on projected crown area (m^2), wood/leaf point classification ratios and Jaccard Index metrics. The Jaccard Index (Jaccard, 1912) is a measure of spatial concordance; values range from 0%–100% where 0% and 100% indicate no overlap and complete overlap of point clouds accordingly (Brede et al., 2022). A Weighted Jaccard Index is calculated here by first voxelising the segmented point clouds with an edge length of 0.5 m, a weight w is then assigned to each voxel corresponding to the number of points per voxel. Then for each tree pair the weighted intersection and union of voxel sets is computed using Python's sklearn (Pedregosa et al., 2011) *jaccard_index* method.

Owing to the unbalance in number of segmented trees per plot, when deriving metrics for all matched pairs, a bootstrap sampling approach is taken where for each iteration a sample of 10 trees per plot is drawn.

2.6 | Computing infrastructure and software

Insufficient computing infrastructure can be a bottle neck to processing large geospatial data sets, such as point cloud data. TLS data can generate particularly large data sets for relatively small regions of interest when compared to other laser scanning instruments, for example airborne. Therefore, for a new TLS data processing pipeline to be operationalised it would ideally be capable of being horizontally scaled.

Horizontal scaling is achieved in the semantic and instance segmentation steps of *TLS2trees* by mapping P to a set of tiles T; then $t \in T$ are run independently and in parallel. To process the plots in Table 1, the UK's Natural Environment Research Council's (NERC) JASMIN computing facility and the NERC Earth Observation Data Acquisition and Analysis Service (NEODAAS) MAssive GPU for Earth Observation (MAGEO) cluster were used. The JASMIN facility (https://jasmin.ac.uk/) is designed for data intensive computing and comprises a large volume of storage combined with general purpose batch computing capability. The MAGEO cluster (https://www.neodaas.ac.uk/) is a specialised system designed for earth observation data and comprises 40 GPUs, 200 GPU cores and 0.5 PB of fast storage. Data processing requirements and times for a subset of plots are presented in Table 2.

All steps use Python as a base programming language, this includes common scientific libraries such as Numpy, Scipy and Pandas

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Processing steps and benchmark times.

	GPU/CPU	Parallelisable	Software	RUSH	NOU	MLA
Plot area (ha)				0.3	1	1
Number of scans				10	242	242
Raw data size (Gb)				2.4	65.7	107.8
Preprocessing	CPU	Within machine 10× CPU used for benchmark	Python, PDAL			
Raw points per tile				400 K	33.2 M	67.7 M
Downsample points per tile				350 K	5.5 M	7 M
Wall time				47 s	45 min	1 h 20 min
Semantic segmentation	GPU	TRUE	Python, PyTorch			
Mean CPU time per tile				1 min	13 min	20 min
Maximum RAM usage per tile (gb)				4.6	39.7	82.3
Wall time				11 min	4 h	3.5 h
Instance segmentation	CPU	TRUE	Python, NetworkX			
Buffer (tiles)				3 × 3	5 × 5	7 × 7
Mean CPU time per tile				16 min	3 h	4 h 50 min
Maximum RAM usage per tile				2.8	51.6	129.8
Wall time				1 h 30 min	16 h	18 h

(see Wilkes et al. (2023) for full list) which are managed with the *conda* package and environment manager. In addition, the pre-processing step also uses the PDAL library (PDAL Contributors, 2020), semantic segmentation uses PyTorch libraries (version 1; Fey & Lenssen, 2019) and instance segmentation uses Networkx (Hagberg et al., 2008).

3 | RESULTS

3.1 | Plot wide segmentation

A total of 10,557 trees were segmented from 10 plots, of which 3908 trees were inside the plot boundary and have a DBH >0.1 m (Table 3). The tallest tree segmented is a 78.9 m dipterocarp from MLA and the largest tree by volume is a 211 m^3 coastal redwood from CALI (CALI-A, Figure 3). Trees from WYT have far longer total branch length where the longest is 8.5 km, this compares to the longest tropical branch length of <2 km. This could be as a result of a trees ontogeny, for example repeated pollarding followed by abandonment, or a systematic method bias where aggressive semantic segmentation misclassifies smaller branches as leaves for evergreen trees.

Estimated segmented stem density ranges from 247 stems ha^{-1} at the RUSH plots to 761 stems ha^{-1} at the AEP plot (Table 3). Compared to reported stem density values (Table 1) these are ±10% for RUSH, WYT, NOU and MLA; larger discrepancies are evident at AEP (−25%), KOG (+39%). Fujimori (1977) reported a large interval for stem density in CALI-type (coastal redwood) plots as a function of succession; values reported here fall within that range.

Figure 4 shows per plot the largest segmented tree point clouds (by total woody volume); for each tree, both wood and leaf points are displayed (and coloured accordingly). The Supplementary Material presents all trees segmented for plots where *TreeQSM* computed DBH >0.1 m. In general, tree crowns appear complete with only small omission and commission errors evident, for example small twigs/branches from neighbouring trees. The crowns of large tropical trees (plots MLA and NOU) are well segmented, even capturing the idiosyncrasies of crown morphology, for example tree MLA-Q has broken and repsprouted (top row Figure 4). Along the length of some tree stems there are erroneous leaf points that are either attributable to lianas or crowns of smaller mid and understorey trees (e.g. trees NOU-A and MLA-B in Figure 4). As described in Section 2.2.3, leaf points P_L are connected to a stem via branch tips, this would suggest that graph connections have been made to branch points surrounding the stem. It should be noted that P_L are not used to model volume.

TreeQSM appears to have overestimated the total volume of a few trees (e.g. trees KOG-B, RUSH-AG, AEP-H, AEP-J and NOU-L in Figure 4). This is a result of *TLS2trees* derived commission errors where small disconnected woody structures (e.g. shrubs) have inflated the size of cylinders used to estimate volume, particularly at the base. Further, a number of trees in the KOG plot have numerous smaller stems at the base of each tree (see Section 4).

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 Summary of *TLS2trees* segmented trees, only trees that are within the plot boundary and DBH >0.1 m are considered. 99th percentile is used to remove outliers.

Plot	N		Height (m)			Projected crown area (m^2)		Total volume (m^3)		Trunk volume (m^3)		Branch length (m)		DBH (m)	
	|S|	S ha^{-1}	Median	99th	Max	Median	99th	Median	99th	Median	99th	Median	99th	Median	99th
RUSH	162	247.0	12.5	24.5	26.1	15.8	156.7	0.3	4.2	0.2	2.0	54.0	594.8	0.3	1.3
WYT	393	393.0	18.4	27.0	30.5	36.1	291.4	1.3	20.9	0.5	5.2	337.7	3570.6	0.3	1.0
NOU	547	521.0	17.9	46.0	58.9	36.6	338.7	0.4	15.9	0.3	11.7	35.8	605.3	0.2	1.2
KOG	292	268.9	6.4	25.7	26.7	24.2	228.6	0.4	10.3	0.2	5.4	41.6	1166.8	0.3	1.6
CALI	804	326.4	23.9	68.4	72.9	51.1	283.9	2.3	85.7	1.8	75.4	111.0	1758.0	0.5	3.0
MLA	401	371.1	15.8	65.1	78.9	30.2	543.3	0.4	35.9	0.3	32.9	33.6	1066.1	0.3	1.5
AEP	462	760.7	11.5	22.5	23.4	19.8	128.1	0.2	3.3	0.1	2.4	24.3	270.7	0.2	0.9

3.2 | Comparison with manually segmented trees

A total of 1053 segmented trees were successfully paired to a manually segmented tree, this represents a match rate of 82% (Table 4). For the majority of plots a corresponding tree pair is found for >90% of trees and for all plots a pair is found for all segmented trees that constitute the largest 50% of total segmented volume. The lowest rates are observed at WYT: see Section 4 for discussion.

Scatter plots that compare key QSM and point cloud metrics are presented in Figure 5. *TLS2trees* derived point clouds result in *TreeQSM* total woody volume estimates (a precursor for estimating AGB) with a per plot bias of ±0.4 m^3 and RMSE of 0.5–3 m^3. The exception being CALI where total woody volume bias and RMSE are −1.4 and 16.7 m^3 (%RMSE = 70%) respectively. Segmentation of smaller trees by *TLS2trees* is less accurate, this results in a *TreeQSM* derived per tree total woody volume %RMSE of >1000%. However, when considering the largest trees that constitute 50% of total woody volume, %RMSE of per-tree woody volume reduces to 60%.

Presented in Figure 6 is a comparison of per plot summed total volume; summed volume from *TLS2trees* segmented trees were ±10% that of manually segmented trees for all plots. The exception is the RUSH and KOG plots where an 18% and 30% overestimate in total woody volume is evident.

If using an existing allometric equation to estimate tree volume or AGB then other structure metrics are important, for example *DBH* and tree height. Tree height is estimated with an %RMSE of 27% where the largest errors were at the tall tree sites (MLA and CALI) as well as AEP. For AEP, tree height is underestimated owing to semantic segmentation errors near the base of trees where wood points were classified as leaf, this also impacted estimates of *DBH*. Crown area is consistently overestimated by *TLS2trees* segmented trees with a bias of 20 m^2. *DBH* is also overestimated by *TLS2trees* segmented trees with a bias of 0.04 m and an %RMSE of 100%, this reduced to <40% when considering the cohort of largest trees. Inflation of crown area and *DBH* by *TLS2trees* segmented trees are attributed to outlier points that increase projected area; it is suggested that a method to filter outlier points could improve results.

Segmented tree pair similarity is tested using the Jaccard Index (see Section 2.5). Mean Jaccard Index results for all trees is 75%; this indicates that a pair of trees shared 75% of the voxel space (weighted by point density; Table 4). Differences were predominantly caused by the foliage around stems (e.g. top row Figure 4). Considering all points, a Jaccard Index of >90% is evident at 3 the least densely stocked plots the sites. Jaccard index values increase for all plots when considering only the largest trees that comprise 50% of total woody volume and are generally greater than 75% (except CALI plots; Table 4). Trees at AEP performed poorly when considering Jaccard Index values, with a median score of 13% increasing to 75% when considering the 7 largest trees; it is suggested this is again due to a poor semantic segmentation.

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

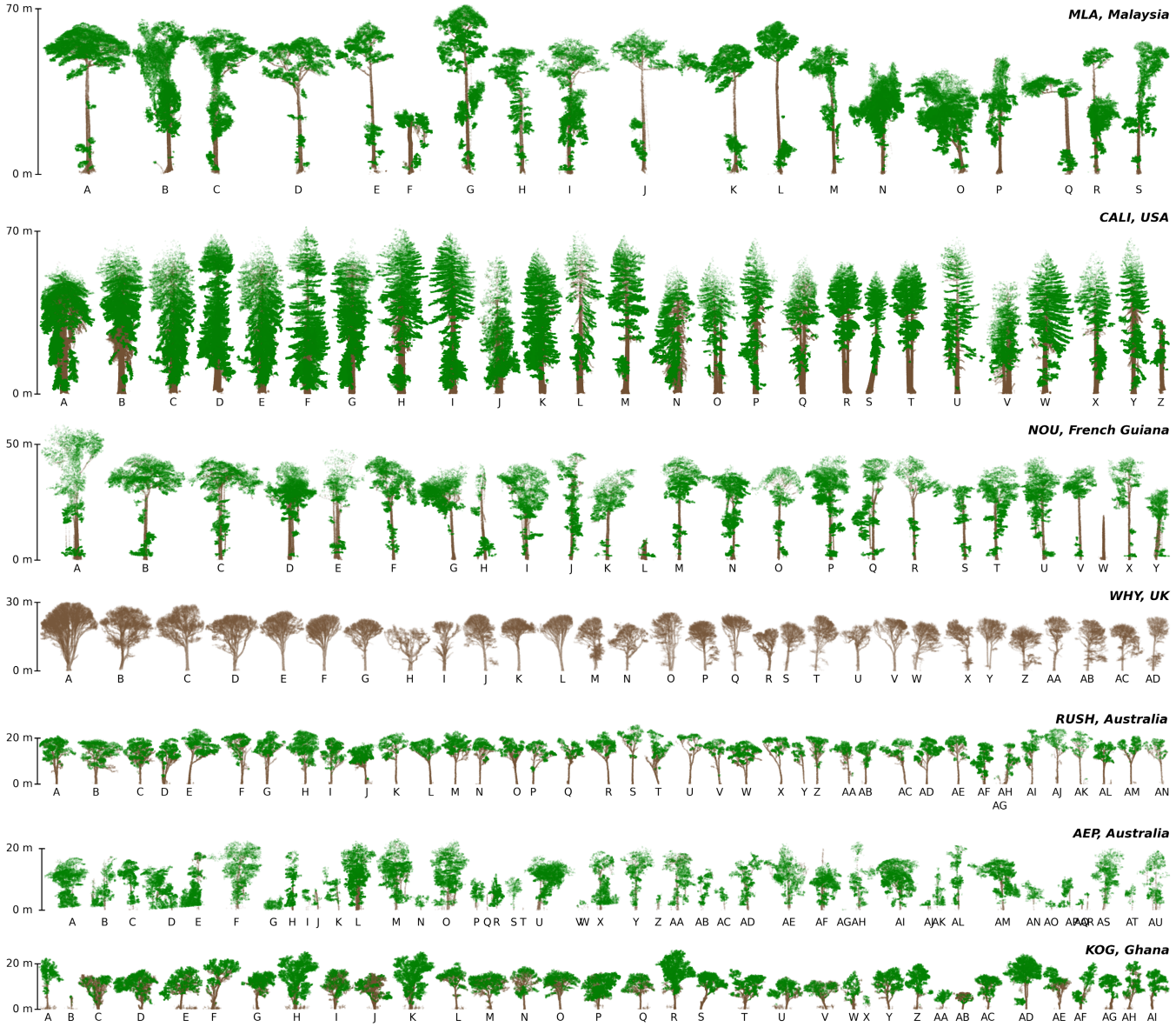


FIGURE 4 The largest trees (by volume) segmented using *TLS2trees* for different forest types. Points are coloured leaf or wood components as classified during the semantic segmentation step.

Considering wood and leaf point voxel occupancy (Table 4), the Jaccard Index values were lower than for all-points values. This indicates that there is a mismatch between wood/leaf point segmentation methods. For example, at MLA 67% of points were classified as leaf using *TLSeparation* whereas 91% of points were classified as leaf using the deep-learning segmentation (Table 4). This also impacts total branch length (Figure 5e) where *TLS2trees* significantly underestimates length compared to manually segmented trees.

4 | DISCUSSION

TLS methods are capable of generating accurate estimates of tree and forest AGB. Operationalising TLS workflows could therefore have implications for activities including National Forest Inventories (NFI) (Liang et al., 2018); measurement, reporting and verification (MRV) protocols; benchmarks; and reference datasets for AGB focused satellite missions (e.g. BIOMASS and GEDI; Chave et al., 2019) and new or updating allometry (Disney et al., 2020; Stovall et al., 2017). However, achieving operationalisation of TLS methods has proved challenging for a number of reasons. In particular, and the focus of this work, is the labour-intensive and time-consuming effort to accurately segment individual trees from plot-level point clouds which have caused a significant processing bottleneck. To address this issue, here we have presented *TLS2trees*, a FOSS and horizontally scalable Python-based pipeline for segmenting individual trees from plot-scale TLS point clouds.

Manually segmenting trees from plot-level point clouds is currently regarded as the most accurate method, and it is suggested

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 Matched tree pair point clouds comparison between manually and *TLS2trees* (T2t) segmented trees.

			All trees						Largest matched trees that contribute 50% of total volume						
			Jaccard index (%)			Points (%)				Jaccard index (%)			Points (%)		
PLOT	*N*	%*N*	All points	Leaf points	Wood points	Leaf Man	Leaf T2t	Wood Man	Wood T2t	*N*	All points	Leaf points	Wood points	Leaf Man	Leaf T2t	Wood Man	Wood T2t
RUSH	65	100	93.5	86.89	77.21	86.87	62.26	13.13	37.74	33	93.5	86.89	77.21	86.87	62.26	13.13	37.74
WYT	373	66.7	91.47	–	–	–	–	–	–	39	95.68	–	–	–	–	–	–
NOU	153	98.7	67.51	46.72	61.25	82.01	88.31	17.99	11.69	24	84.02	47.74	72.39	74.57	79.38	25.43	20.62
KOG	112	94.9	90.1	75.2	70.19	58.6	69.06	41.35	30.79	20	92.64	81.04	77.55	58.12	68.12	41.88	31.88
CALI	146	93.6	73.24	48.05	63.8	77.58	72.62	22.42	27.38	78	71.21	47.92	65.2	77.74	74.3	22.26	25.7
MLA	177	88.9	49.61	37.11	27.02	66.71	90.59	29.33	6.87	12	84.49	53.67	68.2	63.1	83.41	32.32	15.46
AEP	27	93.1	13.34	4.41	46.06	60.07	89.03	33.01	9.86	7	76.65	39.18	63.63	50.08	82.31	31.31	10.46
all	1053	82.2	74.43	46.08	59.66	71.45	80.78	26.06	15.5	–	–	–	–	–	–	–	–

that this is unlikely to change. However, such approaches are not reproducible and lack the traceability or transparency that would be required for carbon accounting and reporting programmes. Manual segmentation is also not scalable, where, along with issues of subjectivity, operator fatigue may degrade results. An automated approach, such as *TLS2trees*, aims to address these issues of reproducibility, subjectivity and scalability. However, routinely segmenting many hundreds of trees per hectare presents additional challenges, for example, quality assurance (QA). As seen here, different forest types present different challenges in terms of generating an accurate result and therefore an application focused QA strategy is suggested. For example, if the metric of interest is AGB, a strategy of sampling the largest trees, that disproportionately contribute to plot level AGB, as well as a subset of smaller trees across the AGB range should identify systematic issues.

TLS2trees performs well across a range of forest types, from open savanna (KOG) to tall tropical rainforest (MLA), the workflow also performs well within plots segmenting trees across a range of sizes (Table 3). This is despite the fact the semantic segmentation base model is only trained on a small area of Australian and New Zealand forest (Krisanski, Taskhiri, Gonzalez Aracil, Herries, & Turner, 2021) and *TLS2trees* model parameters were not adjusted for forest type (Appendix 0). Retraining the semantic segmentation model to be forest type or sensor specific has yielded improved results (Wielgosz et al., 2023); however, we suggest that using the *FSCT* base model is sufficient for a first pass. We would also like to stress that *TLS2trees* is a work in progress and there are a number of aspects of *TLS2trees* that could be improved. We hope this can be achieved through a user community (e.g. see Wielgosz et al., 2023) and have made the source code open-source to facilitate this.

The success of each step presented in Figure 3 depends on the success of the previous step. For example, as seen at the AEP plot, misclassification of wood points as leaf points around the base of trees caused errors in the instance segmentation. Similarly, QSM methods such as *TreeQSM* require a "clean" point cloud with minimal noise (Raumonen et al., 2013) and this is not always achieved, for example inclusion of small neighbouring stems leads to volume inflation. Model parameterisation will be forest type and scanner specific, for example, the height at which `--find-stems-boundary` (Appendix 0) is taken will differ depending on tree height and height and density of the understorey. It is therefore suggested that improved results can be achieved through an optimisation (Wielgosz et al., 2023) or heuristic approach where model parameters are tested on individual tiles before scaling up. Another approach could be to test different parameterisation in a simulated forest, for example a comparison of semantic segmentation methods (Morel et al., 2020).

One particular aspect where *TLS2trees* performed poorly is with the segmentation of smaller trees. Example errors include commission errors where multiple stems are grouped into a single tree or omission errors where smaller stems are missed. Again, a different parameterisation of *TLS2trees* may improve performance. An example is presented in Figure 7 where a change in

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

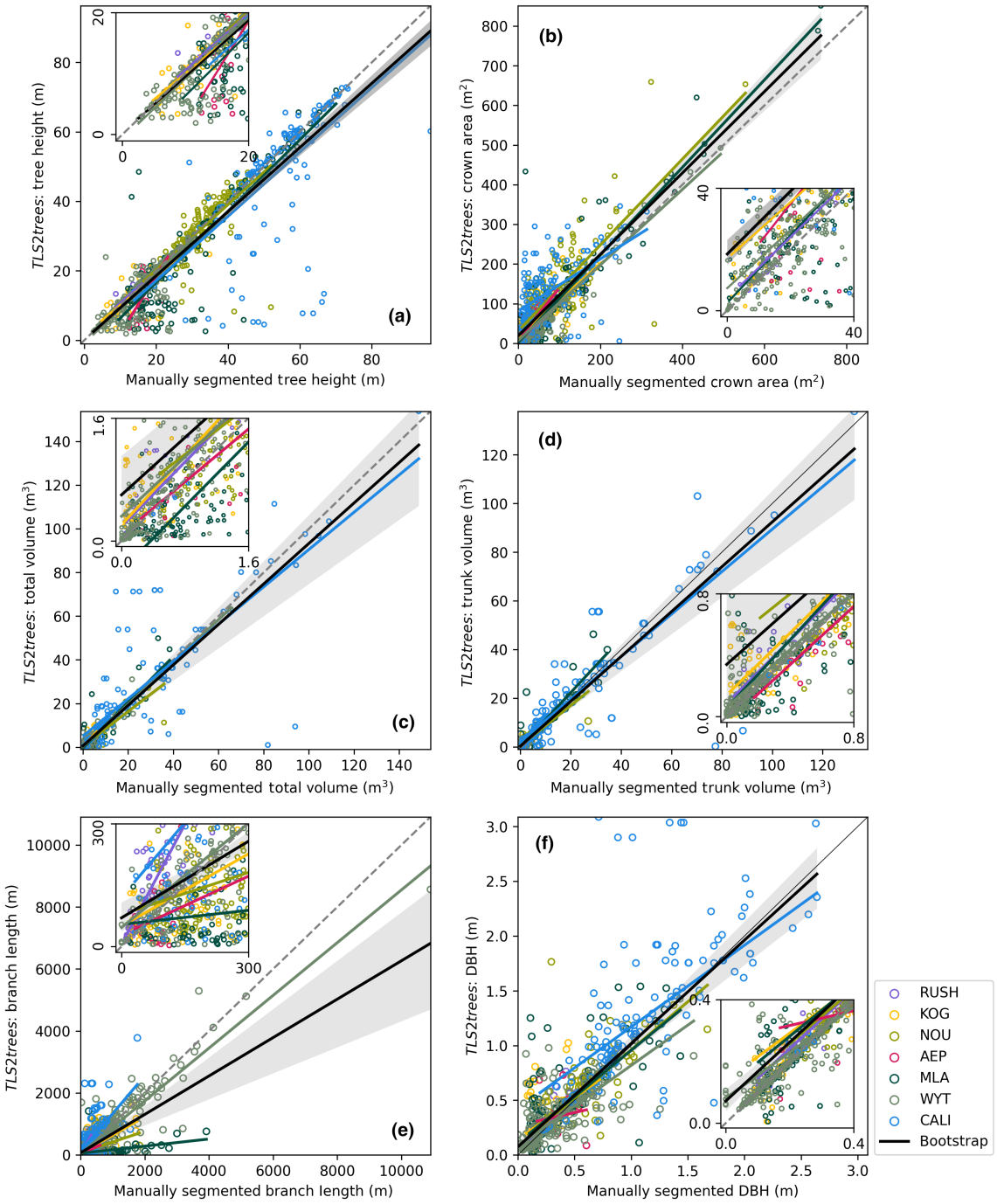


FIGURE 5 A comparison of manually and *TLS2trees* segmented tree pairs for metrics (a) tree height, (b) crown area, (c) total volume, (d) trunk volume, (e) total branch length and (f) *DBH*. Smaller panels are zoomed into the lower 50th percentile of tree pairs. The "boostrap" line represents a mean regression line for all plots, the grey shaded area is a 99% confidence interval for the regression.

`--maximum-cumulative-gap` reduces the number of erroneous stems at the base of the tree; however, if the parameters is reduced too far then portions of the crown are removed. Instrument limitations should also to considered when resolving smaller features, where laser beam and scanning characteristics can limit capability (Demol, Wilkes, et al., 2022). An example of the impact of the smallest trees is evident at WYT where total volume %RMSE is 2000% when compared to manually segmented trees. However, if considering only the largest trees that contributed 90% and 50% to total volume, %RMSE reduced to 50% and 25% respectively. Further, matching pairs of trees was a challenge at WYT where coppiced stools that forked below 1.3 m were often regarded as a single tree by *TLS2trees*; whereas they were manually extracted as individuals by Calders et al. (2018). A suggested solution to improving instance segmentation for small, forked trees is to perform a further instance segmentation based upon topological information derived

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

from the QSM. The output from *TLS2trees* is a point cloud file for each individually segmented tree; therefore any tree can be manually edited using software such as CloudCompare (https://www.danielgm.net/cc/).

Considering the scalability of *TLS2trees* there may be a number of options for improving code and routine efficiency. For example, a 10 m x 10 m processing unit (see Section 2.2.1) was chosen to allow high compute parallelisation of 1 ha plots; however, this may not be optimal for all forest types and processing architectures. Further, a high tile edge length to area ratio increases the likelihood of duplicate segmented trees. Other options for improved efficiency include larger vertical slice (and therefore reduced number of clusters) in the instance segmentation step (see Section 2.2.3) may improve compute times. Here, *TLS2trees* was run on a state-of-the-art computing facility that may not be available to all groups. An alternative option would be to run on a commercially available cloud computing service (e.g. Amazon Web Service or Microsoft Azure), to facilitate this we have provided a containerised version of *TLS2trees*.

Lastly, although the name *TLS2trees* implies the workflow is limited to the processing of just TLS data, we suggest that the framework presented here could be applied to other laser scanning modes. The horizontal scalability functionality, built in to *TLS2trees*, is well suited to the large area acquisitions possible with airborne platforms.

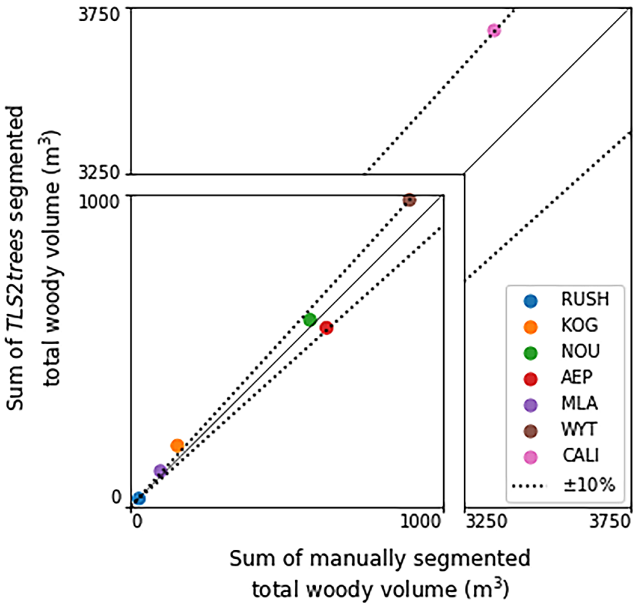


FIGURE 6 A comparison of per plot summed woody volume for manually and *TLS2trees* segmented trees.

5 | CONCLUSION

There is an estimated 260 ha of plot-level TLS data collected from forests across the globe (*pers. comm*. Dr. Atticus Stovall, NASA, 22nd November 2022); processing this data archive could yield upwards of 260,000 individuals trees. The biophysical and ecological insight that could be drawn from this data, including and beyond the estimation of AGB, could be significant. Further, there are a number of other potential uses for this data, including accurate 3D representations of forest plots in radiative transfer Calders et al. (2018) or other large area modelling approaches. However, much of this data remains unprocessed (to individual tree level) owing to instance segmentation bottlenecks, a problem we suggest is one of the remaining hurdles to operationalising TLS protocols. Here we have presented *TLS2trees*, a workflow

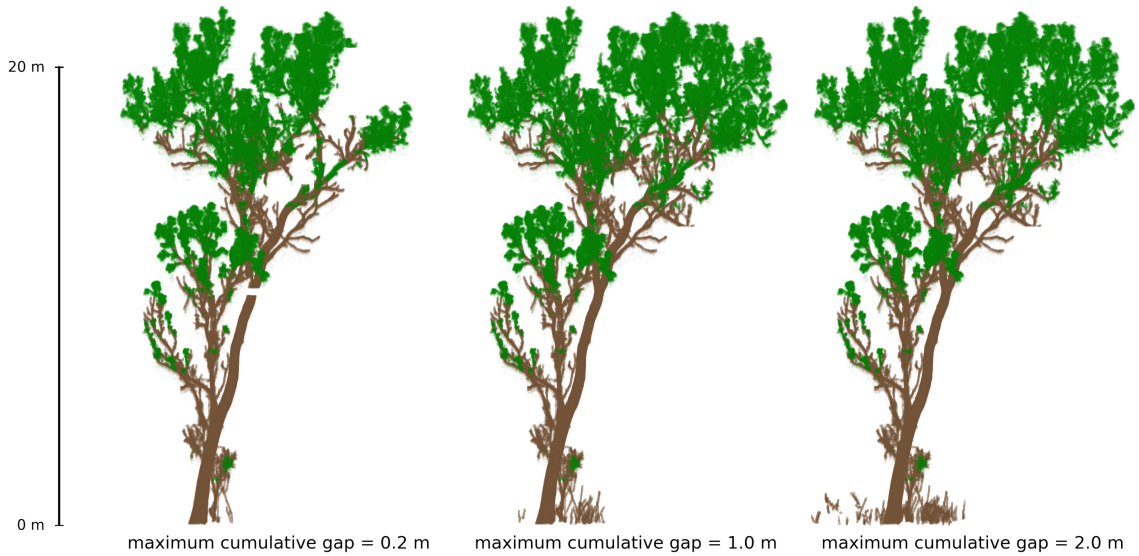


FIGURE 7 An example of different outcomes when altering model parameters. The parameter `--maximum-cumulative-gap` alters the maximum cumulative gap between node and a base node. In this example, a low value segments the base well but removes smaller branches at the top of the canopy whereas a higher value results in small stems around the base being included. The value used for this manuscript is `--maximum-cumulative-gap=2` m.

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

and processing pipeline to segment point clouds of individual trees from plot-level TLS point clouds, that we hope will begin to redress this issue.

AUTHOR CONTRIBUTIONS

Phil Wilkes and Mathias Disney conceived the project. Phil Wilkes, Mathias Disney, Andrew Burt, Kim Calders, Niall Origo, Laura Duncanson, John Armston, Benjamin Brede, Harm Bartholomeus and Alexander Shenkin collected data. Phil Wilkes, Cecilia Chavana-Bryant, Brieanne Forbes, Andrew Burt, Kim Calders, Benjamin Brede, Harm Bartholomeus, Daniel Clewley, David Moffat and Niall Origo processed data. Phil Wilkes, Sean Krisanski, Daniel Clewley, David Moffat and Wanxin Yang developed software. Phil Wilkes conducted analysis. All authors contributed to and approved the final manuscript.

ACKNOWLEDGEMENTS

P.W. and M.D. are funded by NERC National Centre for Earth Observation (NCEO). P.W., M.D., A.S., Y.M. and L.B. were funded under NERC grant NE/P011780/1. M.D. and A.B. were funded under NERC grant NE/N00373X/1. M.D. and C.C.-B. were funded by ESA ForestScan Project (ESA contract 4000126857/20/NL/AI). Y.M. was funded under NERC grant NE/P012337/1 as well as ERC grant GEM-TRAITS (321121). K.C. was funded by the European Union (ERC-2021-STG Grant agreement No. 101039795). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. N.O. was funded by the MetEOC-4 project which included funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme. B.B. was funded by ESA IDEAS-QA4EO (contract no ESA AO/1-11035/21/I-DT). S.K. was funded by the Australian Research Council, Training Centre for Forest Value (IC150100004). D.C., D.M. and computing infrastructure were fund by NERC Earth Observation Data Acquisition and Analysis Service (NEODAAS). Capital equipment was funded by NCEO and UCL Geography. Data were collected in Malaysia under permit JKM/MBS.1000-2/2 JLD.7 (87), we are grateful for project partners Sabah Biodiversity Center, Chief Minister's Department Office of Internal Affairs & Research, Land & Survey Department, Sabah Forestry Department, the Maliau Basin and Danum Valley Management Committees, Forest Research Center (Sabah). We are also grateful for the assistance of Matt Bradford and CSIRO Atherton.

CONFLICT OF INTEREST STATEMENT

AB is an employee of Sylvera Ltd.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14233.

DATA AVAILABILITY STATEMENT

The *TLS2trees* code is available from Wilkes et al. (2023). Code to run TreeQSM in Octave is available from https://github.com/philwilkes/run-treeqsm-in-octave. Raw scan data and manually segmented trees for RUSH plots are available from http://dx.doi.org/10.4227/05/542B766D5D00D. Raw scan data and manually segmented trees for WYT are available http://dx.doi.org/10.5285/ed9156e1697343e4ad82e83ed550e345 and https://doi.org/10.5281/zenodo.7307956, respectively. Data for all other sites is available upon request [Correction added on 7 November 2023, after first online publication: Data Availability Statement has been updated.].

ORCID

Phil Wilkes https://orcid.org/0000-0001-6048-536X
Mathias Disney https://orcid.org/0000-0002-2407-4026
John Armston https://orcid.org/0000-0003-1232-3424
Harm Bartholomeus https://orcid.org/0000-0002-1905-7678
Lisa Bentley https://orcid.org/0000-0002-6180-8842
Benjamin Brede https://orcid.org/0000-0001-9253-4517
Andrew Burt https://orcid.org/0000-0002-4209-8101
Kim Calders https://orcid.org/0000-0002-4562-2538
Cecilia Chavana-Bryant https://orcid.org/0000-0001-7353-9654
Daniel Clewley https://orcid.org/0000-0003-1243-3711
Laura Duncanson https://orcid.org/0000-0003-4031-3493
Brieanne Forbes https://orcid.org/0000-0001-9240-9687
Sean Krisanski https://orcid.org/0000-0003-0689-0051
Yadvinder Malhi https://orcid.org/0000-0002-3503-4783
David Moffat https://orcid.org/0000-0003-4885-7276
Niall Origo https://orcid.org/0000-0002-8475-6264
Alexander Shenkin https://orcid.org/0000-0003-2358-9367
Wanxin Yang https://orcid.org/0000-0002-4246-4802

REFERENCES

Beyene, S. M., Hussin, Y. A., Kloosterman, H. E., & Ismail, M. H. (2020). Forest inventory and aboveground biomass estimation with terrestrial lidar in the tropical forest of Malaysia. *Canadian Journal of Remote Sensing*, *46*(2), 130–145.

Brede, B., Terryn, L., Barbier, N., Bartholomeus, H. M., Bartolo, R., Calders, K., Derroire, G., Moorthy, S. M. K., Lau, A., Levick, S. R., Raumonen, P., Verbeeck, H., Wang, D., Whiteside, T., van der Zee, J., & Herold, M. (2022). Non-destructive estimation of individual tree biomass: Allometric models, terrestrial and UAV laser scanning. *Remote Sensing of Environment*, *280*, 113180.

Burt, A., Boni Vicari, M., Da Costa, A. C., Coughlin, I., Meir, P., Rowland, L., & Disney, M. (2021). New insights into large tropical tree mass and structure from direct harvest and terrestrial lidar. *Royal Society Open Science*, *8*(2), 201458.

Burt, A., Calders, K., Cuni-Sanchez, A., Gómez-Dans, J., Lewis, P., Lewis, S. L., Malhi, Y., Phillips, O. L., & Disney, M. (2020). Assessment of bias in pan-tropical biomass predictions. *Frontiers in Forests and Global Change*, *3*, 12.

Burt, A., Disney, M., & Calders, K. (2019). Extracting individual trees from lidar point clouds using *treeseg*. *Methods in Ecology and Evolution*, *10*(3), 438–445.

Calders, K., Adams, J., Armston, J., Bartholomeus, H., Bauwens, S., Bentley, L. P., Chave, J., Danson, F. M., Demol, M., Disney, M., Gaulton, R., Krishna Moorthy, S. M., Levick, S. R., Saarinen, N.,

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Schaaf, C., Stovall, A., Terryn, L., Wilkes, P., & Verbeeck, H. (2020). Terrestrial laser scanning in forest ecology: Expanding the horizon. *Remote Sensing of Environment*, *251*, 112102.

Calders, K., Newnham, G., Burt, A., Murphy, S., Raumonen, P., Herold, M., Culvenor, D., Avitabile, V., Disney, M., Armston, J., & Kaasalainen, M. (2015). Nondestructive estimates of above-ground biomass using terrestrial laser scanning. *Methods in Ecology and Evolution*, *6*(2), 198–208.

Calders, K., Origo, N., Burt, A., Disney, M., Nightingale, J., Raumonen, P., Åkerblom, M., Malhi, Y., & Lewis, P. (2018). Realistic forest stand reconstruction from terrestrial lidar for radiative transfer modelling. *Remote Sensing*, *10*(6), 933.

Calders, K., Verbeeck, V., Burt, A., Origo, N., Nightingale, J., Malhi, Y., Wilkes, P., Raumonen, P., Bunce, R. G. H., & Disney, M. (2022). Laser scanning reveals potential underestimation of biomass carbon in temperate forest. *Ecological Solutions and Evidence*, *3*, e12197.

Chave, J., Davies, S. J., Phillips, O. L., Lewis, S. L., Sist, P., Schepaschenko, D., Armston, J., Baker, T. R., Coomes, D., Disney, M., Duncanson, L., Hérault, B., Labrière, N., Meyer, V., Réjou-Méchain, M., Scipal, K., & Saatchi, S. (2019). Ground data are essential for biomass remote sensing missions. *Surveys in Geophysics*, *40*(4), 863–880.

Chave, J., Olivier, J., Bongers, F., Châtelet, P., Forget, P.-M., van der Meer, P., Norden, N., Riéra, B., & Charles-Dominique, P. (2008). Above-ground biomass and productivity in a rain forest of eastern south America. *Journal of Tropical Ecology*, *24*(4), 355–366.

Demol, M., Verbeeck, H., Gielen, B., Armston, J., Burt, A., Disney, M., Duncanson, L., Hackenberg, J., Kükenbrink, D., Lau, A., Ploton, P., Sewdien, A., Stovall, A., Takoudjou, S. M., Volkova, L., Weston, C., Wortel, V., & Calders, K. (2022). Estimating forest above-ground biomass with terrestrial laser scanning: Current status and future directions. *Methods in Ecology and Evolution*, *13*(8), 1628–1639.

Demol, M., Wilkes, P., Raumonen, P., Krishna Moorthy Parvathi, S., Calders, K., Gielen, B., & Verbeeck, H. (2022). Volumetric overestimation of small branches in 3D reconstructions of *Fraxinus excelsior*. *Silva Fennica*, *56*(1).

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Disney, M., Burt, A., Wilkes, P., Armston, J., & Duncanson, L. (2020). New 3D measurements of large redwood trees for biomass and structure. *Scientific Reports*, *10*(1), 1–11.

Duncanson, L., Armston, J., Disney, M., Avitabile, V., Barbier, N., Calders, K., Carter, S., Chave, J., Herold, M., Crowther, T. W., Falkowski, M., Kellner, J. R., Labriere, N., Lucas, R., MacBean, N., McRoberts, R. E., Meyer, V., Næsset, E., Nickeson, J. E., … Williams, M. (2019). The importance of consistent global forest aboveground biomass product validation. *Surveys in Geophysics*, *40*(4), 979–999.

Eaton, J. W., Bateman, D., Hauberg, S., & Wehbring, R. (2020). GNU Octave version 6.1.0 manual: A high-level interactive language for numerical computations. https://www.gnu.org/software/octave/doc/v6.1.0/

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Vol. 96 (pp. 226–231). Association for the Advancement of Artificial Intelligence.

Feliciano, E. A., Wdowinski, S., & Potts, M. D. (2014). Assessing mangrove above-ground biomass and structure using terrestrial laser scanning: A case study in the everglades national park. *Wetlands*, *34*(5), 955–968.

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428.

Fujimori, T. (1977). Stem biomass and structure of a mature *Sequoia sempervirens* stand on the pacific coast of northern California. *Journal of the Japanese Forestry Society*, *59*(12), 435–441.

Gonzalez de Tanago, J., Lau, A., Bartholomeus, H., Herold, M., Avitabile, V., Raumonen, P., Martius, C., Goodman, R. C., Disney, M., Manuri, S., Burt, A., & Calders, K. (2018). Estimation of above-ground biomass of large tropical trees with terrestrial lidar. *Methods in Ecology and Evolution*, *9*(2), 223–234.

Graham, A. W. (Ed.). (2006). *The CSIRO rainforest permanent plots of North Queensland—Site, structural, floristic and edaphic descriptions*. (p. 252) Cairns: CSIRO, the Cooperative Research Centre for Tropical Rainforest Ecology, and Management.

Hackenberg, J., Spiecker, H., Calders, K., Disney, M., & Raumonen, P. (2015). Simpletree—An efficient open source tool to build tree models from TLS clouds. *Forests*, *6*(11), 4245–4294.

Hagberg, A., Swart, P., & Schult, D. A. (2008). *Exploring network structure, dynamics, and function using networkx* Tech. rep. Los Alamos National Laboratory (LANL).

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, *11*(2), 37–50.

Kissling, W. D., Shi, Y., Koma, Z., Meijer, C., Ku, O., Nattino, F., Seijmonsbergen, A. C., & Grootes, M. W. (2022). Laserfarm–a high-throughput workflow for generating geospatial data products of ecosystem structure from airborne laser scanning point clouds. *Ecological Informatics*, *72*, 101836.

Krisanski, S., Taskhiri, M. S., Gonzalez Aracil, S., Herries, D., Muneri, A., Gurung, M. B., Montgomery, J., & Turner, P. (2021). Forest structural complexity tool—An open source, fully-automated tool for measuring forest point clouds. *Remote Sensing*, *13*(22), 4677.

Krisanski, S., Taskhiri, M. S., Gonzalez Aracil, S., Herries, D., & Turner, P. (2021). Sensor agnostic semantic segmentation of structurally diverse and complex forest point clouds using deep learning. *Remote Sensing*, *13*(8), 1413.

Kükenbrink, D., Gardi, O., Morsdorf, F., Thürig, E., Schellenberger, A., & Mathys, L. (2021). Above-ground biomass references for urban trees from terrestrial laser scanning data. *Annals of Botany*, *128*(6), 709–724.

Levick, S. R., Whiteside, T., Loewensteiner, D. A., Rudge, M., & Bartolo, R. (2021). Leveraging TLS as a calibration and validation tool for MLS and ULS mapping of savanna structure and biomass at landscape-scales. *Remote Sensing*, *13*(2), 257.

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., Huang, H., Jo, H. W., Katoh, M., Liu, L., Mokroš, M., Morel, J., Olofsson, K., Poveda-Lopez, J., Trochta, J., … Wang, Y. (2018). International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS Journal of Photogrammetry and Remote Sensing*, *144*, 137–179.

Malhi, Y., Girardin, C., Metcalfe, D. B., Doughty, C. E., Aragao, L. E. O. C., Rifai, S. W., Oliveras, I., Shenkin, A., Aguirre-Gutierrez, J., Dahlsjo, C. A. L., Riutta, T., Berenguer, E., Moore, S., Huasco, W. H., Salinas, N., da Costa, A. C. L., Bentley, L. P., Adu-Bredu, S., Marthews, T. R., … Phillips, O. L. (2021). The global ecosystems monitoring network: Monitoring ecosystem productivity and carbon cycling across the tropics. *Biological Conservation*, *253*, 108889. https://doi.org/10.1016/j.biocon.2020.108889. https://www.sciencedirect.com/science/article/pii/S0006320720309472

Martin-Ducup, O., Mofack, G., Wang, D., Raumonen, P., Ploton, P., Sonké, B., Barbier, N., Couteron, P., & Pélissier, R. (2021). Evaluation of automated pipelines for tree and plot metric estimation from TLS data in tropical forest areas. *Annals of Botany*, *128*(6), 753–766.

Momo Takoudjou, S., Ploton, P., Sonké, B., Hackenberg, J., Griffon, S., De Coligny, F., Kamdem, N. G., Libalah, M., Mofack, G. I., Le Moguédec, G., & P'elissier, R., & Barbier, N. (2018). Using terrestrial laser scanning data to estimate large tropical trees biomass and calibrate allometric models: A comparison with traditional destructive approach. *Methods in Ecology and Evolution*, *9*(4), 905–916.

Moorthy, S. M. K., Bao, Y., Calders, K., Schnitzer, S. A., & Verbeeck, H. (2019). Semi-automatic extraction of liana stems from

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

terrestrial lidar point clouds of tropical rainforests. *ISPRS Journal of Photogrammetry and Remote Sensing*, *154*, 114–126.

Morel, J., Bac, A., & Kanai, T. (2020). Segmentation of unbalanced and in-homogeneous point clouds and its application to 3D scanned trees. *The Visual Computer*, *36*(10), 2419–2431.

Murphy, S., Bi, H., Volkova, L., Weston, C., Krishnaraj, S. J., Fairman, T., & Law, R. (2014). *CCAP comprehensive carbon assessment program*. Victorian Centre for Climate Change Adaptation Research.

Newnham, G. J., Armston, J. D., Calders, K., Disney, M. I., Lovell, J. L., Schaaf, C. B., Strahler, A. H., & Danson, F. M. (2015). Terrestrial laser scanning for plot-scale forest measurement. *Current Forestry Reports*, *1*(4), 239–251.

Ogle, S. M., Kurz, W. A., Green, C., Brandon, A., Baldock, J., Domke, G., Herold, M., Bernoux, M., Chirinda, N., de Ligt, R., Federici, S., Garcia-Apaza, E., Grassi, G., Gschwantner, T., Hirata, Y., Houghton, R., House, J. I., Ishizuka, S., Jonckheere, I., … Waterworth, R. M. (2019). Generic methodologies applicable to multiple land-use categories. In E. C. Buendia, K. Tanabe, A. Kranjc, J. Baasansuren, M. Fukuda, S. Ngarize, A. Osako, Y. Pyrozhenko, P. Shermanau, & S. Federici (Eds.), *2019 refinement to the 2006 IPCC guidelines for National Greenhouse Gas Inventories* (pp. 2.1–2.96). IPCC. https://www.ipcc-nggip.iges.or.jp/public/2019rf/vol1.html

PDAL Contributors. (2020). Pdal point data abstraction library. https://doi.org/10.5281/zenodo.2556737

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Phillips, O. L., Sullivan, M. J., Baker, T. R., Monteagudo Mendoza, A., Vargas, P. N., & Vásquez, R. (2019). Species matter: Wood density influences tropical forest biomass at multiple scales. *Surveys in Geophysics*, *40*(4), 913–935.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv*, arXiv:1706.02413.

Raumonen, P. (2019). Inversetampere/treeqsm: Version 2.3.1. https://doi.org/10.5281/zenodo.3482908

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., & Lewis, P. (2013). Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sensing*, *5*(2), 491–520.

Shenkin, A., Bentley, L. P., Oliveras, I., Salinas, N., Adu-Bredu, S., Marimon-Junior, B. H., Marimon, B. S., Peprah, T., Choque, E. L., Trujillo Rodriguez, L., Clemente Arenas, E. R., Adonteng, C., Seidu, J., Passos, F. B., Reis, S. M., Blonder, B., Silman, M., Enquist, B. J., Asner, G. P., & Malhi, Y. (2020). The influence of ecosystem and phylogeny on tropical tree crown size and shape. *Frontiers in Forests and Global Change*, *3*, 501757.

Stovall, A. E., Vorster, A. G., Anderson, R. S., Evangelista, P. H., & Shugart, H. H. (2017). Non-destructive aboveground biomass estimation of coniferous trees using terrestrial lidar. *Remote Sensing of Environment*, *200*, 31–42.

Tao, S., Wu, F., Guo, Q., Wang, Y., Li, W., Xue, B., Hu, X., Li, P., Tian, D., Li, C., Yao, H., Li, Y., Xu, G., & Fang, J. (2015). Segmenting tree crowns from terrestrial and mobile lidar data by exploring ecological theories. *ISPRS Journal of Photogrammetry and Remote Sensing*, *110*, 66–76.

Van Den Berge, S., Vangansbeke, P., Calders, K., Vanneste, T., Baeten, L., Verbeeck, H., Krishna Moorthy, S. P., & Verheyen, K. (2021). Biomass expansion factors for hedgerow-grown trees derived from terrestrial lidar. *Bioenergy Research*, *14*(2), 561–574.

Vicari, M. B., Disney, M., Wilkes, P., Burt, A., Calders, K., & Woodgate, W. (2019). Leaf and wood classification framework for terrestrial lidar point clouds. *Methods in Ecology and Evolution*, *10*(5), 680–694.

Wang, D. (2020). Unsupervised semantic and instance segmentation of forest point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *165*, 86–97.

Wang, D., Liang, X., Mofack, G. I., & Martin-Ducup, O. (2021). Individual tree extraction from terrestrial laser scanning data via graph pathing. *Forest Ecosystems*, *8*(1), 1–11.

Wang, D., Momo Takoudjou, S., & Casella, E. (2020). Lewos: A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial lidar. *Methods in Ecology and Evolution*, *11*(3), 376–389.

Wielgosz, M., Puliti, S., Wilkes, P., & Astrup, R. (2023). Point2tree (p2t)–framework for parameter tuning of semantic and instance segmentation used with mobile laser scanning data in coniferous forest. arXiv preprint arXiv:2305.02651.

Wilkes, P., Disney, M., Vicari, M. B., Calders, K., & Burt, A. (2018). Estimating urban above ground biomass with multi-scale lidar. *Carbon Balance and Management*, *13*(1), 1–20.

Wilkes, P., Krisanski, S., Clewley, D., Moffat, D., & Yang, W. (2023). TLS2trees. https://doi.org/10.5281/zenodo.8406668

Wilkes, P., Lau, A., Disney, M., Calders, K., Burt, A., de Tanago, J. G., Bartholomeus, H., Brede, B., & Herold, M. (2017). Data acquisition considerations for terrestrial laser scanning of forest plots. *Remote Sensing of Environment*, *196*, 140–153.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Data S1. Additional figures showing all segmented tree point clouds from 10 plots where trees are within the plot boundary and have a DBH >0.1 m as calculated by TreeQSM.

How to cite this article: Wilkes, P., Disney, M., Armston, J., Bartholomeus, H., Bentley, L., Brede, B., Burt, A., Calders, K., Chavana-Bryant, C., Clewley, D., Duncanson, L., Forbes, B., Krisanski, S., Malhi, Y., Moffat, D., Origo, N., Shenkin, A., & Yang, W. (2023). *TLS2trees*: A scalable tree segmentation pipeline for TLS data. *Methods in Ecology and Evolution*, *14*, 3083–3099. https://doi.org/10.1111/2041-210X.14233

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

APPENDIX A

Python tools and exposed parameters

TABLE A.1 Python commands and parameter values used to segment trees. All code can be found at https://github.com/tls-tools-ucl/TLS2trees. *specific to RIEGL VZ scanners.

Parameter	Purpose	Value used
1. Preprocessing		
`rxp2ply.py`*		
`--tile`	Edge length of tile	10
`--deviation`	Upper and lower bounds of deviation values	0, 15
`--reflectance`	Upper and lower bounds of reflectance values	−20, 0
`downsample.py`		
`--length`	Edge length voxel where 1 point is retained per voxel	0.02
`tile_index.py`		
2. Semantic segmentation		
`semantic.py`		
`--buffer`	Size of buffer to use from surrounding tiles	5
`--model`	File path to trained model	*Defaults to FSCT model*
3. Instance segmentation		
`instance.py`		
`--n-tiles`	Number of tiles to use as a buffer, that is by 3×3 or tiles or 5×5 tiles	3, 5 or 7
`--slice-thickness`	Vertical slice thickness for constructing graph	0.5
`--find-stems-boundary`	Upper and lower bounds for extracting a slice within which to identify stems	2.0, 2.5
`--find-stems-min-radius`	Minimum radius of found stems	0.025
`--find-stems-min-points`	Minimum number of points for found stems	200
`--graph-edge-length`	Maximum distance between individual nodes in a graph	2
`--graph-maximum-cumulative-gap`	Maximum cumulative distance between a base and a node	3
`--min-points-per-tree`	Minimum number of points for a tree to be segmented	200
`-add-leaves`	Whether to add leaf points	TRUE
`-add-leaves-voxel-length`	Voxel size when add leaves	0.5
`-add-leaves-edge-length`	Maximum distance used to connect points in leaf graph	1

2041210x, 2023, 12, Downloaded from https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14233 by Oxford University, Wiley Online Library on [23/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License