NEON Algorithm Theoretical Basis Document: TOS Vegetation Structure - QA/QC of Raw Field Data

|  |  |  |
| --- | --- | --- |
| **PREPARED BY** | **ORGANIZATION** | **DATE** |
| Katherine Jones | FSU |  |
| Dr. Courtney Meier | FSU |  |

|  |  |  |
| --- | --- | --- |
| **APPROVALS (Name)** | **ORGANIZATION** | **APPROVAL DATE** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **RELEASED BY (Name)** | **ORGANIZATION** | **RELEASE DATE** |
|  |  |  |

See Configuration Management System for approval history.

**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **REVISION** | **DATE** | **ECO #** | **DESCRIPTION OF CHANGE** |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

[1 Description 2](#_Toc380338854)

[1.1 Purpose 2](#_Toc380338855)

[1.2 Scope 2](#_Toc380338856)

[2 Related Documents and Acronyms 3](#_Toc380338857)

[2.1 Applicable Documents 3](#_Toc380338858)

[2.2 Reference Documents 3](#_Toc380338859)

[2.3 Acronyms 3](#_Toc380338860)

[3 Data Product Description 5](#_Toc380338861)

[3.1 Variables Reported 6](#_Toc380338862)

[3.2 Temporal Resolution and Extent 10](#_Toc380338863)

[3.3 Spatial Resolution and Extent 10](#_Toc380338864)

[3.4 Associated Data Streams 11](#_Toc380338865)

[3.5 Product Instances 11](#_Toc380338866)

[4 Scientific Context 11](#_Toc380338867)

[4.1 Theory of Measurement/Observation 12](#_Toc380338868)

[4.2 Theory of Algorithm 12](#_Toc380338869)

[4.3 Special Considerations 13](#_Toc380338870)

[5 Algorithm Implementation 14](#_Toc380338871)

[6 Uncertainty 18](#_Toc380338872)

[6.1 Analysis of Uncertainty 18](#_Toc380338873)

[6.2 Reported Uncertainty 18](#_Toc380338874)

[7 Validation and Verification 18](#_Toc380338875)

[7.1 Algorithm Validation 18](#_Toc380338876)

[7.2 Data Product Validation 18](#_Toc380338877)

[7.3 Data Product Verification 18](#_Toc380338878)

[8 Scientific and Educational Applications 18](#_Toc380338879)

[9 Future Modifications and Plans 18](#_Toc380338880)

[10 Bibliography 19](#_Toc380338881)

[11 Changelog 19](#_Toc380338882)

# Description

## Purpose

This document details the algorithms used for creating a subset of NEON Level 1 data products that are the quality controlled products generated from raw Level 0 data, and associated metadata. In the NEON data products framework, the raw data collected in the field, for example, hind foot length of an individual small mammal, are considered the lowest level (Level 0). Raw data that have been quality checked via the algorithms detailed herein, as well as simple metrics that emerge from the raw data, such as total species richness of small mammals at a particular site, are considered Level 1 data products. This document relates only to the former group of L1 data products, the quality controlled pass-through products from the Level 0 data products.]

It includes a detailed discussion of measurement theory and implementation, appropriate theoretical background, data product provenance, quality assurance and control methods used, approximations and/or assumptions made, and a detailed exposition of uncertainty resulting in a cumulative reported uncertainty for this product.

## Scope

This document describes the theoretical background and entire algorithmic process for creating a subset of quality controlled and calibrated L1 data products and associated metadata from input data. These data products include vegetation structure for trees, shrubs, lianas, and other non-herbaceous perennial plants (Tables 1 - 4). It does not provide computational implementation details, except for cases where these stem directly from algorithmic choices explained here. This document also provides details relevant to the publication of the data products via the NEON data portal, *NEON Data Publication Workbook for TOS Vegetation Structure: QA/QC of Raw Field Data* (AD[12]).

This document describes the algorithms for ingesting and performing automated quality assurance and control procedures on the data collected in the field pertaining to Field and Lab Protocol: Vegetation Structure *(AD[10])*. The raw data that are processed in this document are detailed in the NEON Raw Data Ingest Workbook for *NEON Raw Data Ingest Workbook for TOS Vegetation Structure (AD[11])*.

# Related Documents and Acronyms

## Applicable Documents

|  |  |  |
| --- | --- | --- |
| AD[01] | NEON.DOC.000001 | NEON Observatory Design (NOD) Requirements |
| AD[02] | NEON.DOC.005003 | NEON Scientific Data Products Catalog |
| AD[03] | NEON.DOC.005004 | NEON Level 1-3 Data Products Catalog |
| AD[04] | NEON.DOC.005005 | NEON Level 0 Data Product Catalog |
| AD[05] | NEON.DOC.005011 | NEON Coordinate Systems Specification |
| AD[06] | NEON.DOC. 001247 | NEON Algorithm Theoretical Basis Document for QA/QC Plausibility Testing of Organismal and Observation Based Field and Lab Data |
| AD[07] | NEON.DOC.004309 | NEON Field Site Information |
| AD[08] | NEON.DOC.00XXXX | NEON Algorithm Theoretical Basis Document: Taxonomic Consensus Identifications and Uncertainty |
| AD[09] | NEON.DOC.000914 | TOS Science Design for Plant biomass, productivity, and leaf area index |
| AD[10] | NEON.DOC.000987 | Field and Lab Protocol : Vegetation Structure |
| AD[11] | NEON.DOC. 001928 | NEON Raw Data Ingest Workbook for TOS Vegetation Structure |
| AD[12] | NEON.DOC. 001939 | NEON Data Publication Workbook for TOS Vegetation Structure: QA/QC of Raw Field Data |
| AD[13] | NEON.DOC.00XXXX | TOS Plot-Level Spatial Data |
| AD[14] | NEON.DOC.00XXXX | TOS Point-Level Spatial Data |
| AD[15] | NEON.DOC.00XXXX | Master Taxonomy for |
| AD[16] | NEON.DOC.00XXXX | Species Lists by Domain for Vascular Plants |
| AD[17] | NEON.DOC.00XXXX | NEON Algorithm Theoretical Basis Document: TOS Vegetation Structure - Structure and Biomass Calculations |
| AD[18] | NEON.DOC.00XXXX | Data Entry Protocol and QA/QC |
| AD [19] | NEON.DOC.00XXXX | Point ID Azimuth validation lookup table |

## Reference Documents

|  |  |
| --- | --- |
| RD[01] | NEON.DOC.000008 NEON Acronym List |
| RD[02] | NEON.DOC.000243 NEON Glossary of Terms |
| RD[03] |  |
| RD[04] |  |

## Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| ANPP | Annual Net Primary Productivity |
| DBH | Diameter at Breast Height |
| ddh | diameter at decimeter height |
|  |  |

# Data Product Description

Vegetation structure data includes information related to structure, spatial location, and biomass of the perennial non-herbaceous plant community, including tree, sapling/shrub, liana, and other growth forms. Stem mapping activities and the collection of vegetation structure data will take place in all Distributed and Tower Plots, and may also occur in Gradient Plots if Gradient Plots are required at a given site (Figure 1). Across a site, as many as 50 plots may be sampled. Tower plots will be re-measured annually; Distributed and Gradient plots may be re-measured less frequently depending of logistical constraints. Sampling occurs while plants are dormant at temperate sites with defined growing season. Tropical or moisture dependent sites at which growth may occur throughout the year, sampling occurs at the same time each year (± 2 weeks) in order to capture a single years’ growth.

Specific data collected include, diameter at breast height (DBH), diameter at decimeter height (ddh), total stem height, canopy diameter, species ID, stem status (i.e. healthy, snag, damaged, etc.), and the location of measured stems. Parameters such as DBH, ddh, canopy diameter and total stem height can then be used to estimate aboveground biomass and carbon (C) density values, on both a per stem and a per unit area basis (algorithm provided in higher lever ATBDs). Map location of measured individuals are collected as relative coordinates from high resolution GPS points (Figure 2and Figure 3) and are reported as lat/long coordinates for each individual. For additional details on the sampling design and associated protocol, see the TOS Science Design for Plant biomass, productivity, and leaf area index (AD[08]) and NEON Field & Lab Protocol for Vegetation Structure (AD[09]). Products resulting from this sampling include the species identification and unique identifier for each individual measured, as well as a suite of standard size measurements and reproductive condition data.

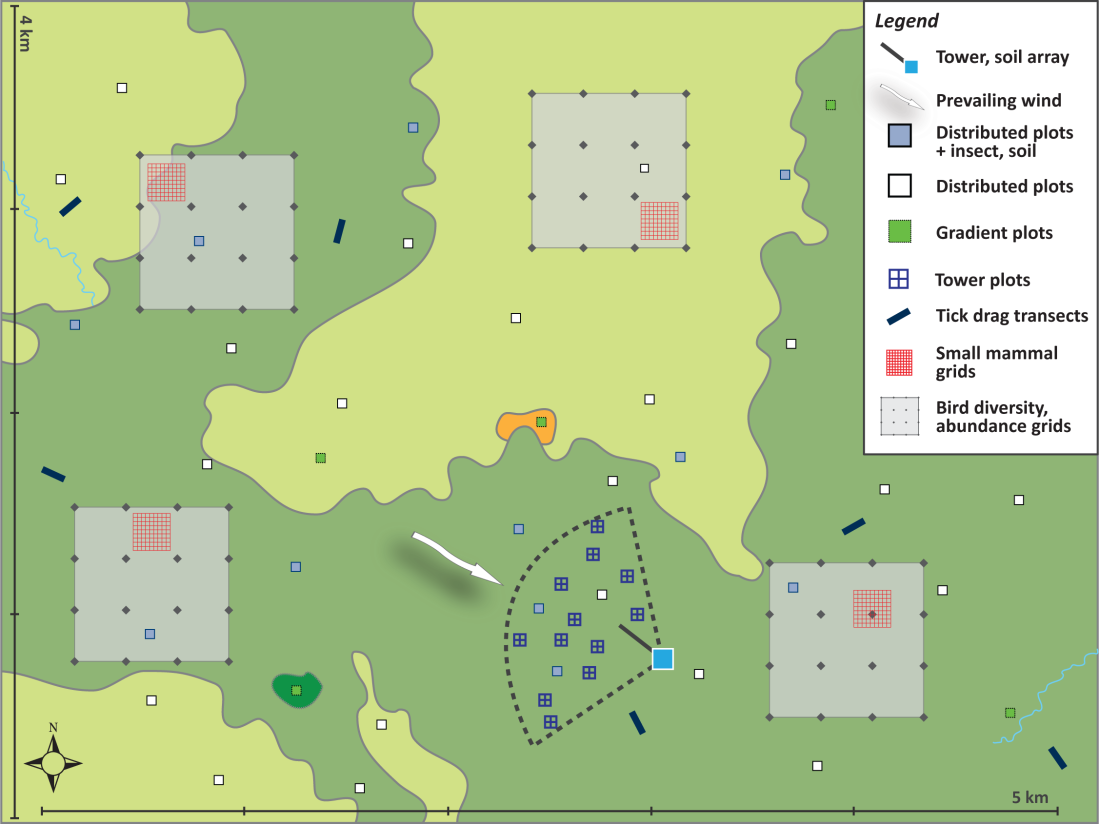


Figure . Generalized TOS sampling schematic, showing the placement of Distributed, Tower, and Gradient Plots

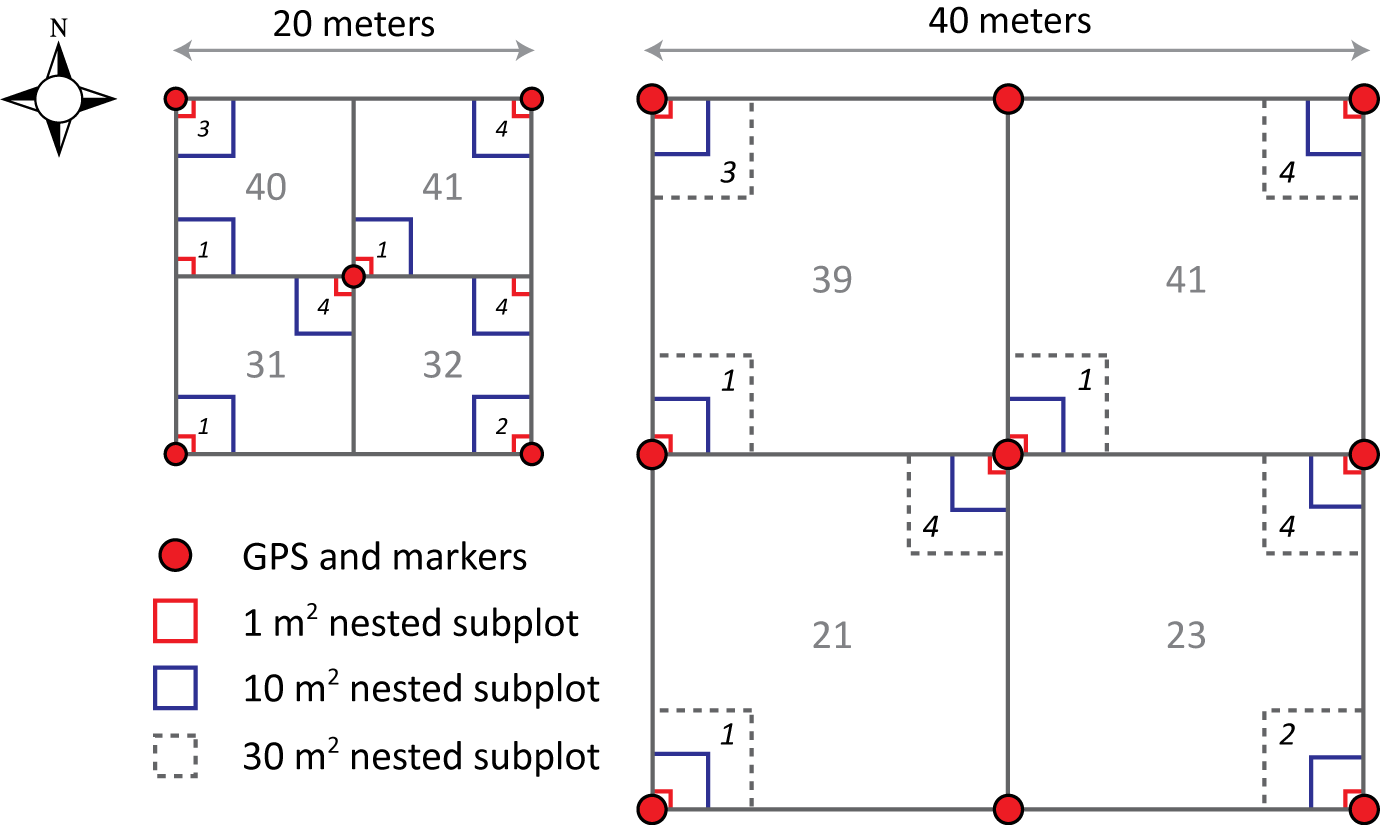


Figure 2. Illustration of a 20 m × 20 m Distributed/Gradient base plot (left), a 40 m × 40 m Tower base plot (right), and associated nested subplots used for measuring woody stem vegetation. Locations of high level subplots are numbered in plain text, and locations of lower level subplots are numbered with italics

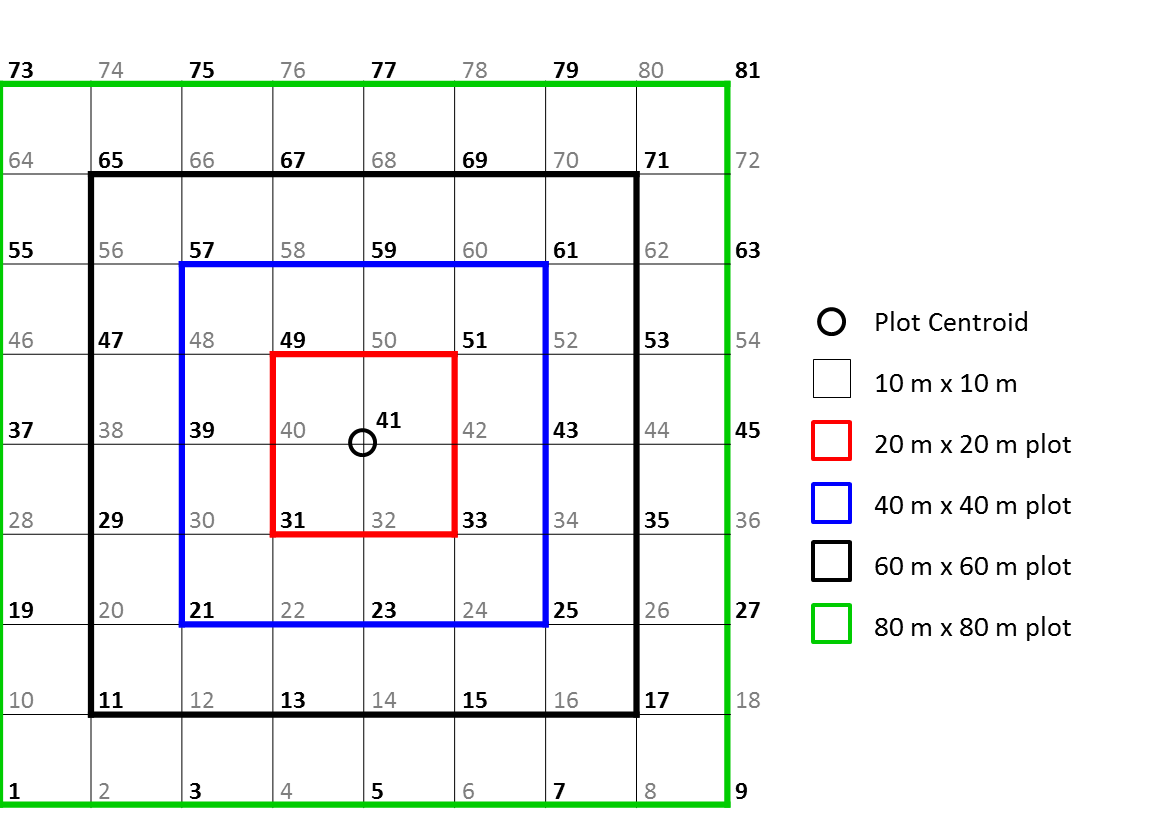


Figure 3. Plot pointIDs are numbered according to the largest possible plot size, 80 m x 80 m, such that a plot of any size will use a consistent numbering scheme. A pointID has been for defined for every 10 m spacing, this design is used in initial plot establishment but permanent, primary and secondary markers, are placed at 20 meter intervals. During stem mapping, technicians will only use the points that have permanent markers in place, those identified as bold in this diagram. Plot subplotIDs are determined by the pointID of the SW corner of the subplot area. Plot center will always be pointID 41.

## Variables Reported

This ATBD describes the steps needed to generate the L1 data products, Tree vegetation structure, Sapling and shrub vegetation structure, Liana vegetation structure, and perennial non-herbaceous vegetation structure. These products are used to estimate biomass on the individual, plot and site scale and to calculate Annual Net Primary Productivity (ANPP) at similar resolutions.

Subproducts for these data products are listed below (Table 1 – 4). Detailed lists of the associated subproducts and metadata products are provided separately, along with example data in publication-ready spreadsheets *NEON Data Publication Workbook for TOS Vegetation Structure: QA/QC of Raw Field Data (AD[12])*. Field names have been standardized with Darwin Core terms (<http://rs.tdwg.org/dwc/>; accessed 16 February 2014), the Global Biodiversity Information Facility vocabularies (<http://rs.gbif.org/vocabulary/gbif/>; accessed 16 February 2014), the VegCore data dictionary (<https://projects.nceas.ucsb.edu/nceas/projects/bien/wiki/VegCore>; accessed 16 February 2014), and with the Bird Monitoring Data Exchange standards (<http://www.avianknowledge.net>; accessed 16 February 2014), where applicable. Geospatial data shall conform to the standards set forth in the NEON Coordinate Systems Specification (AD[05]).

Table 1. List of subproducts produced in this ATBD for the data product, Tree structure. The list is not exhaustive and a variety of supporting data will also be made available.

|  |  |  |
| --- | --- | --- |
| **Number** | **Field Name** | **Description** |
|  | individualID | Domain-level unique identifier based on domain number: NEON.VST.DXX.123456 |
|  | scientificName | Binomial latin name and authority |
|  | idqCode | Code reflecting a standard term to express the determiner's doubts about the Identification. |
|  | growthForm | 3-character code indicating woody plant structure; |
|  | stemDiameter | Cross-sectional stem diameter; centimeters |
|  | measurementHeight | Distance from ground to stem\_diameter measurement point; centimeters |
|  | canopyPosition | Code describing the position of the individual relative to the average canopy within the plot |
|  | MaxCanopyDiameter | Maximum canopy diameter; meters |
|  | NinetyCanopyDiameter | Canopy diameter perpendicular to MaxCanopyDiameter; meters |
|  | height | Height from the base of the individual to the highest point in the canopy; meters |
|  | stemStatus | Description of stem condition |

Table 2. List of subproducts produced in this ATBD for the data product, Shrub/Sapling structure. The list is not exhaustive and a variety of supporting data will also be made available.

|  |  |  |
| --- | --- | --- |
| **Number** | **Field Name** | **Description** |
|  | individualID | Domain-level unique identifier based on domain number: NEON.VST.DXX.123456 |
|  | scientificName | Binomial latin name and authority |
|  | idqCode | Code reflecting a standard term to express the determiner's doubts about the Identification. |
|  | growthForm | 3-character code indicating woody plant structure; |
|  | stemDiameter | Cross-sectional stem diameter; centimeters |
|  | measurementHeight | Distance from ground to stem\_diameter measurement point; centimeters |
|  | canopyPosition | Code describing the position of the individual relative to the average canopy within the plot |
|  | MaxCanopyDiameter | Maximum canopy diameter; meters |
|  | NinetyCanopyDiameter | Canopy diameter perpendicular to canopyDiameterMax; meters |
|  | stemStatus | Description of stem condition |
|  | shrubShape | code describing the shape of a measured shrub |
|  | baseCanopyDiameterMax | Maximum canopy diameter, measured at ground level; meters |
|  | baseCanopyDiameter90 | Canopy diameter perpendicular to basCanopyDiameterMax, measured at ground level; meters |
|  | canopyArea | size of the group canopy, square meters |
|  | canopyPosition | Code describing the position of the individual relative to the average canopy within the plot |
|  | Height | height from the base of the stem to the highest point on the individual; meters |
|  | aGroupHeight | Height of tallest 5 stems in group; meters |
|  | bGroupHeight | Height of tallest 5 stems in group; meters |
|  | cGroupHeight | Height of tallest 5 stems in group; meters |
|  | dGroupHeight | Height of tallest 5 stems in group; meters |
|  | eGroupHeight | Height of tallest 5 stems in group; meters |
|  | taxonID | taxon-specific code, linked to scientificName via a look-up table |
|  | percentVolume | percent of a shrub group volume attributed to a particular species |
|  | percentLive | percent of given species within a shrub group that is alive |
|  | percentDead | percent of a given species within a shrub group that is dead |

Table 3. List of subproducts produced in this ATBD for the data product, Liana structure. The list is not exhaustive and a variety of supporting data will also be made available.

|  |  |  |
| --- | --- | --- |
| **Number** | **Field Name** | **Description** |
|  | individualID | Domain-level unique identifier based on domain number: NEON.VST.DXX.123456 |
|  | supportingStemIndividualID | Unique individualID for stem on which a liana is growing |
|  | scientificName | Binomial latin name and authority |
|  | idqCode | Code reflecting a standard term to express the determiner's doubts about the Identification. |
|  | growthForm | 3-character code indicating woody plant structure; |
|  | stemDiameter | Cross-sectional stem diameter; centimeters |
|  | measurementHeight | Distance from ground to stem\_diameter measurement point; centimeters |
|  | canopyPosition | Code describing the position of the individual relative to the average canopy within the plot |
|  | MaxCanopyDiameter | Maximum canopy diameter; meters |
|  | NinetyCanopyDiameter | Canopy diameter perpendicular to canopyDiameterMax; meters |
|  | stemStatus | Description of stem condition |

Table 4. List of subproducts produced in this ATBD for the data product, non-herbaceous perennial structure. The list is not exhaustive and a variety of supporting data will also be made available.

|  |  |  |
| --- | --- | --- |
| **Number** | **Field Name** | **Description** |
|  | individualID | Domain-level unique identifier based on domain number: NEON.VST.DXX.123456 |
|  | scientificName | Binomial latin name and authority |
|  | idqCode | Code reflecting a standard term to express the determiner's doubts about the Identification. |
|  | growthForm | 3-character code indicating woody plant structure; |
|  | numberOfLeaves | the total number of leaves on an individual |
|  | canopyDiameterMax | Maximum canopy diameter; meters |
|  | canopyDiameter90 | Canopy diameter perpendicular to canopyDiameterMax; meters |
|  | height | height from the base of the stem to the highest point on the individual; meters |
|  | basalStemDiameter | Cross-sectional stem diameter and soil surface; centimeters |
|  | avLengthOfLeaf | the average length of an adult leaf, used to estimate fern biomass; centimeters |
|  | numberOfPads | count of pads on an *Opuntia* |

## Temporal Resolution and Extent

The finest resolution that records will be tracked is the level of the day (date – YYYY-MM-DD). All sampling at a site that occurs within a calendar year will be grouped into a single eventID, the temporal resolution that will be used for calculating higher level data products such as biomass and annual net primary production (ANPP).

## Spatial Resolution and Extent

Individuals, mapped as points within a plot will be tracked and regularly re-measured. Shrub groups cannot be mapped as a point so the finest spatial resolution this growth form will be tracked at is the level of the nested subplot which may be 1,10 or 30 m2. It is anticipated that the finest spatial resolution at which summary statistics will be produced is the level of the plot. Tower, Gradient and Distributed base plots will be sampled for vegetation structure. Tower plots will generally be either 20 m x 20 m or 40 m x 40 m square plots but may be even larger if some systems. Gradient and Distributed base plots are 20 m x 20 m square plots. Plot level data will be aggregated to produce site level estimates of biomass and ANPP.

Spatial hierarchy from finest to coarsest resolution may be defined in a couple different ways depending on the growth form being measured.

Saplings and groups of shrubs, : nested subplot ID -> nested subplot size -> subplot ID -> plot ID -> site ID -> domain ID

Trees, individual shrubs: points, *decimalLatitude, decimalLongitude* (CI-derived latitude and longitude of the individual being measured -> subplot ID -> plot ID -> site ID -> domain ID

## Associated Data Streams

All of the above data products are directly linked to the herbaceous clip harvest, LAI, litterfall and fine woody debris, coarse woody debris, and phenology data products described in each protocol’s respective ATBD NEON ATBD XXXXX (RD[XX]).

|  |  |  |
| --- | --- | --- |
| **Protocol** | **ATBD document #** | **Linking variable** |
| Herbaceous clip harvest | NEON.DOC.01843 | Plot ID |
| Leaf area index |  | Plot ID |
| Litterfall and fine woody debris | NEON.DOC.01851 | Subplot ID + Plot ID |
| Coarse woody debris |  | Plot ID |
| Plant phenology | NEON.DOC.001246 | Tag ID |

The products from each of these ATBDs will be used to estimate biomass and ANPP at varying spatial resolutions.

## Product Instances

Sampling will occur once a year at 40-50 plots per site at all NEON terrestrial sites.

# Scientific Context

## Theory of Measurement/Observation

The measurement of and mapping woody stems is an important complement to data streams generated by the NEON AOP and TIS. These ground-collected data will validate LiDAR data used to map the structural complexity of vegetation, will enable mapping of plant biomass at the site scale, and in conjunction with carbon flux data, will facilitate understanding how biomass in different plant growth forms contributes to ecosystem level carbon flux.

The protocols for measuring vegetation structure provide procedures for measuring wood and non-woody, non-herbaceous perennial vegetation. The term ‘woody vegetation’ applies to any perennial plant that produces persistent lignified vascular tissue that remains aboveground throughout the dormant season (Van Buren et al. 2011, Jepson 2012). In addition to woody vegetation, perennial plants which produce persistent aboveground structures (e.g. ferns and agave) will be measured according to AD[11]. NEON will sample vegetation biomass and productivity across tree, sapling, shrub, liana and non-herbaceous perennial growth forms, with varying methods and varying thresholds for inclusion in the sample depending on growth form (see AD[11] for details). These growth forms are defined, approaches for estimating their aboveground biomass (AGB) are evaluated, and an approach suitable for estimating AGB within Observatory constraints is selected and justified in the TOS Science Design for Plant Biomass, Productivity, and Leaf Area Index AD[09]. *ANPP* of these growth forms will be calculated using annual changes in AGB; the process for making these calculations will be outlined in a higher level data product ATBD, NEON Algorithm Theoretical Basis Document: TOS Vegetation Structure - Structure and Biomass Calculations AD[17].

## Theory of Algorithm

This document provides the algorithm for automated quality checks on field collected data. Data related to individuals ingested through this ATBD collected by field technicians according to the vegetation structure protocol AD[10] and will be re-measured annually. Consistency across years is essential to providing quality data to end users and is one of the primary functions of the ATBD. In addition to quality checking, ‘pass-through’ routines some new variables will be generated though this algorithm including unique identifier and absolute location from relative position measurements.

**Summary of Algorithm for all Data**

1. QC data for:
   1. completeness
   2. data type
   3. date ranges
   4. technician IDs
   5. duplicate values
   6. valid code entries
   7. valid plot IDs
2. Generate unique record IDs
3. Evaluate all validation rules for all columns and check that all fields that should not be null are populated, as specified in the NEON Raw Data Ingest Workbook for TOS Vegetation Structure (AD[11]).
4. Generate spatial metadata fields:
   1. Lookup domain, site, in the spatialDataLookupTable (AD[14]).
5. Convert all dates to standard format.
6. Generate eventID for each plotID:
   * 1. Generate eventID of the form: VST.SITE.year
        + Only a single bout occurs during a single calendar year but a single bout may include many different sampling dates (bouts)
7. Verify scientificName and/or taxonID from domain-specific species lookup table (Species Lists by Domain for Vascular Plants lookup table (AD[16])).
8. Transfer data from the ingest workbook to the NEON database

**Summary of Algorithm for plot Data**

1. Generate spatial metadata describing the plot and plot points
   1. Lookup and assign nlcdClass, plotType, pointID, latitude, longitude and elevation

**Summary of Algorithm for Bout Data**

1. Link bout data to records in other tables by assigning publication table names to vegetationCategoryTable based on values in vegCategory field

**Summary of Algorithm for the Mapping Data – perindividual table**

1. Generate a unique identifier for each individual.
   1. To ensure that the identifier for an individual is a NEON unique identifier across space and time, generate values for the publication field, individualID. These should be of the form, NEON.VST.DXX.123456, where DXX = domain number and 123456 represents an auto-increment number.
   2. some individuals have multiple tagIDs associated with them which is dealt with by splitting the tagID into stemDigits and stemID before generating individualID
2. Validate pointID values.
3. Validate stemAzimuth values
4. Calculate relative position of mapped individuals to pointID; derive Easting and Northing coordinates for each individual
   1. Generate decimalLatitiude and decimalLongitude for each mapped individual

**Summary of Algorithm for perindividual\_perbout table**

1. Validate individualID
2. Validate location of tagID within a plot
3. Repeat measurement checks on marked individuals and generate values for individualID:
   1. Check for internal and temporal consistencies in the tagID and related fields: scientificName (or taxonID), and plotID
4. Assign plotType
5. Assign nestedSubplotArea from values in plot level data table
6. Replace canopyPosition code with Name
7. Replace stemStatus code with Name
8. Check conditional values
9. Calculate height from VD1Height and VD2Height

**Summary of Algorithm for pershrubgroup\_perbout table**

Fill null fields in multispecies data rows

Split shrubgroupID into subplot and group ID fields

Replace canopyPosition code with Name

Calculate average group height

Check that sum of group percentage estimates of a multi-species shrubgroup add up to 100

Check that sum of live and dead estimates of a single taxonID within a group add up to 100

**Summary of Algorithm for perother\_perbout table**

1. Assign measurement sampling area.

**Final steps**

1. convert growthForm codes to full name

## Special Considerations

# Algorithm Implementation

**Automated Processing Steps for Field Collected Data**

* 1. Run the following processing steps for all data in the vegetation structure data ingest workbook

1. Verify that all records in the vst\_dataingest\_2014 workbook are complete
   * 1. Run Validation Test: Complete Records, in AD[06], where: List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
     2. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
2. Verify that all records of all fields are of the correct data type in the vst\_dataingest\_2014 workbook
   1. Run Validation Test: Data Type, in AD[06], where:
      1. List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
3. Convert **date** to correct format and verify that this is within acceptable range in the vst\_dataingest\_2014 workbook
   1. Run Validation Test: Date, in AD[06], where:
      1. List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of date fieldnames= (**date**)
      3. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
4. Verify that technician IDs are valid in any field in which they are entered in the vst\_dataingest\_2014 workbook
   1. Run Validation Test: Technician IDs, in AD[11], where:
      1. List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of technician fieldnames= (measuredBy, recordedBy)
      3. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
5. Check for duplicate values in the vst\_dataingest\_2014 workbook
   1. Run Validation Test: Duplicate Records, in AD([06]), where:
      1. QF Name= tagID
      2. Database table= vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
      3. Data ingest sheet= vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      4. List of ‘Fieldnames’= (**siteID, tagID, date**)
6. Verify that code values in the vst\_dataingest\_2014 workbook are valid, as specified by validation rules
   1. Run Validation Test: Validation Rules, in AD[06], where:
      1. List of data ingest sheets= vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of date fieldnames= (**date, growthForm, canopyPosition, stemStatus, shrubShape**)
      3. List of database tables= vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
7. Verify that **plotID** or **siteID** values are valid for the site and domain in which data were collected in the vst\_dataingest\_2014 workbook
   1. Run Validation Test: Location, in AD[06], where:
      1. List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
8. Generate a unique ID (**uid**) for each record in the vst\_datapub\_2014 workbook
   1. Run Generate: Unique ID, in AD[06], where

List of data ingest sheets= vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in

* + 1. List of database tables= vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db

1. Assign domain and site identifications for each record in the vst\_datapub\_2014 workbook
   1. Run Assign: Location IDs, in AD[06], where

List of data ingest sheets= vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in

* + 1. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db

1. Generate anonymous technician IDs for reporting in the vst\_datapub\_2014 workbook
   1. Run Generate: Technician IDs, in AD[06], where:
      1. List of data ingest sheets=vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of technician fieldnames=measuredBy, recordedBy
      3. List of database tables= vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
      4. List of database tables=vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
2. Generate **eventID**
   1. For each row in *vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db, vst\_perbout\_db*:
      1. Concatenate: ‘VST.’ + first four characters in **plotID** field of *vst\_perindividual\_perbout\_in* + ‘.’ + first four characters in **date** field of *vst\_perindividual\_perbout\_in*
      2. Insert string into **eventID** field of *vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db, vst\_perbout\_db*
3. Verify that recorded **scientificName** or **taxonID** values are valid
   1. Run Validation Test: Lookup, in AD(06), where:
      1. List of data ingest fieldnames=**scientificName, taxonID**
      2. Database table= vst\_perindividual\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
      3. Data ingest sheet= vst\_perindividual\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      4. List of lookup fieldnames=**scientificName** OR **taxonID**
      5. List of lookup tables= AD[16]
4. Assign data values, in vst\_datapub\_2014 workbook, from L0 data values, in the vst\_dataingest\_2014 workbook
   1. Run Assign: L1 Data from L0 Data, in AD[06], where:
      1. List of data ingest sheets=vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of date and time fieldnames=(**date,**)
      3. List of technician fieldnames=( **measuredBy, recordedBy,**)
      4. List of non-transferring fieldnames= (**previouslyMeasuredDate)**
   2. Run the following processing steps for vst\_perplot\_db
5. Assign spatial data from AD [14] to vst\_perplot\_db
   1. For each value in **plotID** field of vst\_perplot\_db
      1. Locate value in plotID field of AD [14]
      2. Count = Count # of unique values in **pointID** of AD [14] field that correspond to value in plotID field of spatial table
      3. duplicate row in DB where **plotID** = value (Count – 1) times
      4. Populate **pointID** field with **pointID** values from spatial table
      5. Populate **plotType** field
6. Generate: Spatial Information and Uncertainty:
   1. Run Generate: Spatial Information and Uncertainty in AD[06], where:
      1. Database table= vst\_perplot\_db
      2. Fieldname = plotID, pointID
      3. Spatial table = TOS Point-Level Spatial Data (AD [14])
      4. Subtype = Base

## Run the following processing steps for vst\_perbout\_db

1. Generate **vegetationCategoryTable** field in vst\_perindivdiual\_db,
   1. For each row in **vegCategory** field of vst\_perindividual\_perbout\_db:

if value in **vegCategory** field == ‘apparent individual’

insert ‘vst\_perindividual\_perbout\_pub’ into cell in **vegetationCategoryTable** field

Else:

if value in **vegCategory** field == ‘shrub group’

insert ‘vst\_pershrubgroup\_perbout\_pub’ into cell in **vegetationCategoryTable** field

Else:

if value in **vegCategory** field == ‘other’

insert ‘vst\_pernonherbaceousperennial\_perbout\_pub’ into cell in **vegetationCategoryTable** field

1. Check for whether all tower plots were sampled for at least one vegetation component in a given bout:
   1. Generate **missingRecordsQF** field in vst\_perbout\_db
      1. populate with zeros
   2. Generate **plotType** field
      1. insert value from **plotType** field for corresponding **plotID** field in AD [13]
   3. Generate temporary field : **eventPlot**
      1. For each **plotID**
      2. Concatenate **eventID** from + **plotType**
      3. insert value in **eventPlot field**
   4. For each unique value in the **eventPlot** field

if plotType = ‘tower’

* + 1. Count = the number of unique **plotID** values
    2. Count2 = the number of unique **plotID** values for each **site** where **plotType** = ‘tower’ in AD [13]

if (Count – Count2) does NOT equal zero AND nlcdClass is NOT = GrasslandHerbaceous OR PastureHay OR CultivatedCrops OR EmergentHerbaceousWetlands OR Developed\*;

insert ‘1’ in **missingRecordsQF**

* + 1. delete temporary field
  1. Run the following processing steps for the *vst\_perindividual\_in* sheet of the *vst\_dataingest\_2014* workbook

1. Generate **stemID** field, leave blank
   1. Generate **stemDigits** field, leave blank
   2. For all records in vst\_perindividual\_db:

For each value in which the first four characters in the **tagID** field are identical:

Count= number of occurrences of first four characters in **tagID** field

if Count > 1:

If last character is in range ‘a’ – ‘j’:

* + - 1. insert first four characters into **stemDigits** field of database
      2. Insert last character into **stemID** field of database

Else:

1. Insert ‘x’ into **stemID** field

Else:

* + - 1. insert first four characters into **stemDigits** field of database

1. Generate **individualID,** a NEON unique identifier across space and time (**Generate: Individual ID**)
   1. CI shall maintain a lookup table (hereafter lookup:NEONMODID, where ‘MOD’ is the 3-letter module abbreviation for the module in question) in which unique sets of **domainID** and **stemDigits** values are linked for each domain, through a 6-digit CI-autogenerated **individualID** value of the form NEON.MOD.DXX.123456, where:
      1. MOD= vst
      2. DXX = domain number
      3. 123456= auto-increment number for each unique **domainID**/**stemDigits** combination
   2. For each row in vst\_perindividual\_db:
      1. Concatenate values in **domainID** and **stemDigits** fields

If value from b.1 is in **individualIDs** field of lookup:NEONMODID:

1. Insert value from **individualID**  field of lookup: NEONMODID into **individualID** field of database table

Else:

1. Auto-increment **individualID** value as specified in step a
2. Insert auto-incremented number into **individualID** field of database table
3. Verify that **pointID**s are valid for the given plotID
   1. Generate **pointIDQF** field in vst\_perindividual\_db and populate with zeros
   2. for each row in **pointID** field in vst\_perindividual\_db

for each **pointID** in the vst\_perindivdiual\_db:

if value is not in AD [14] for the given **plotID**:

1. insert -9999 into cell in **pointID** field of ith database table
2. insert 1 into cell in **pointIDQF** field of ith database table
3. Assign **plotSize** values
   1. for each row in **plotID** field of vst\_perindividual\_db
      1. find value in **plotID** field of AD [13]
      2. get corresponding value in **plotSize** field of AD [13]
      3. insert value from a.ii into **plotSize** field of vst\_perindividual\_db
4. Verify that recorded stemAzimuthvalues are valid
   1. Generate **azimuthQF** field in vst\_perindivdiual\_db, and populate with zeros
5. For each row in **stemAzimuth** field of vst\_perindividual\_db:
6. if value in **plotSize** field == value in **plotSize** field in AD [13] AND value in **pointID** field == value in **pointID** field of AD [14] for a given **plotID**:
   1. aAzimuth= value from **aAzimuth** field of lookup table AD [19]
   2. bAzimuth= value from **bAzimuth** field of lookup table AD [19]

If aAzimuth < bAzimuth:

If value in **stemAzimuth** field is not in range: aAzimuth to bAzimuth:

* + 1. Insert -9999 into cell in the **stemAzimuth** field of the vst\_perindividual\_db
    2. Insert 1 into cell in **azimuthQF** field of vst\_perindividual\_db

Else:

If value in aAzimuthand/or bAzimuth == 360:

* + 1. Change aAzimuth and/or bAzimuth to 0

if absolute value of (bAzimuth – aAzimuth) == 270:

if aAzimuth < bAzimuth:

if value in **stemAzimuth** field is not in range: aAzimuth to bAzimuth field:

1. Insert -9999 into cell in the **stemAzimuth** field of the vst\_perindividual\_db
2. Insert 1 into cell in **azimuthQF** field of vst\_perindividual\_db

Else:

If value in **stemAzimuth** field is not in range: aAzimuthto (aAzimuth+90)

1. Insert -9999 into cell in the **stemAzimuth** field of the vst\_perindividual\_db
2. Insert 1 into cell in **azimuthQF** field of vst\_perindividual\_db

Else:

if aAzimuth < bAzimuth:

if value in **stemAzimuth** field is not in range: aAzimuth to bAzimuth:

1. insert -9999 into cell in the **stemAzimuth** field of the vst\_perindividual\_db
2. Insert 1 into cell in **azimuthQF** field of vst\_perindividual\_db

Else:

If value in **stemAzimuth** field is not in range: 270 to 360 OR 0 to 90:

1. Insert -9999 into cell in the **stemAzimuth** field of the vst\_perindividual\_db
2. Insert 1 into cell in **azimuthQF** field of vst\_perindividual\_db
3. Calculate **easting**, for each **individualID** in vst\_perplot\_db table
   1. Generate new **easting** field in vst\_perindividual\_db
   2. for each row in vst\_perplot\_db
      1. find **plotID + pointID** in TOS Plot
      2. get corresponding **easting** values
      3. Calculate temporary variable xOffset = (**stemDistance** \* sin(**stemAzimuth**)
      4. insert value for (**pointIDEasting** from AD[13] + xOffset) into **easting** cell of vst\_perplot\_db
4. Calculate **northing**, for each **individualID** in vst\_perplot\_db table
   1. Generate new **northing** field in the database table vst\_perindividual\_db
   2. for each row in vst\_perplot\_db
      1. find **plotID + pointID** in plot\_lookup\_table
      2. get corresponding **pointIDNorthing** values
      3. Calculate temporary variable yOffset = (**stemDistance** \* cos(**stemAzimuth**))
      4. insert value for (**pointIDNorthing** from AD[13] + yOffset) into **northing** cell of vst\_perplot\_db
5. Convert **easting** to **decimalLatitude** in vst\_perindividual\_db
   1. Get **utmZone** from AD[13]
6. Convert **northing** to **decimalLongitude** in vst\_perindividual\_db
   1. Get **utmZone** from AD[13]
7. Assign data values, in vst\_perindividual\_db, from L0 data values, in the vst\_dataingest\_2014 workbook
   1. Run Assign: L1 Data from L0 Data, in AD[06], where:
      1. List of data ingest sheets=vst\_perplot\_in, vst\_perindividual\_in, vst\_perindividual\_perbout\_in, vst\_pershrubgroup\_perbout\_in, vst\_perother\_perbout\_in
      2. List of date and time fieldnames=(**date,**)
      3. List of technician fieldnames=( **measuredBy, recordedBy,**)
      4. List of non-transferring fieldnames= (**previouslyMeasuredDate)**
      5. List of database tables=vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
   2. Run the following processing steps for vst\_perindivdiual\_perbout\_db
8. Generate **indivdiualIDQF** field, populate with zeros
   1. Check combination of **domainID** + first four characters of **tagID** invst\_perindividual\_perbout\_dbagainst **domainID** + first four characters of **tagID** in vst\_perindividual\_db:
      1. Concatenate values in **domainID** and first four characters of **tagID** fields

If value from 1.a is in **individualID** field of vst\_perindividual\_db:

Insert value from **individualID**  field of lookup: NEONMODID into **individualID** field of vst\_perindividual\_perbout\_db

Else:

Insert value of 1 in **individualIDQF** field of vst\_perindividual\_perbout\_db

1. Verify that **tagID** and **subplotID** in the vst\_perindidivdiual\_perbout are consistent across measurement bouts.
2. If current **eventID** is not the first **eventID** in vst\_perindividual\_perbout\_db for a given **siteID**
3. Run Validation Test: Consistency, in AD[06], where:
   * + 1. List of data ingest sheets= vst\_perindividual\_perbout\_in,
       2. List of date fieldnames= **siteID, tagID**, **subplotID**
       3. List of database tables= vst\_perindividual\_perbout\_db,

Else:

Do not run consistency tests

1. Generate **plotIDQF** field in vst\_perindividual\_perbout\_db; populate with zeros
   1. Generate **plotType** field in vst\_perindividual\_perbout\_db
2. For each row in vst\_perindividual\_perbout\_db:

lookup plotType for specified **plotID** in AD[13]

if value exists:

Assign value to vst\_perindivdiual\_perbout\_db

Else:

insert ‘-9999’ in the **plotType** field

and

insert ‘1’ in the **plotIDQF** field

1. Generate **nestedSubplotArea** field in vst\_perindividual\_perbout\_db
2. For each row in **plotID** field of vst\_perindividual\_perbout\_db
3. if value in **growthForm** field of perindividual\_perbout\_db = ‘sis’ OR ‘sap’ OR ‘sms’

insert value from **nestedSubplotAreaShrubSapling** field of the vst\_perplot\_db into **nestedSubplotArea** of the vst\_perindividual\_perbout\_db

Else:

if value in the **growthForm** field = ‘lia’

insert value from **nestedSubplotAreaLiana** field of the vst\_perplot\_db

1. Generate **canopyPositionName** field in vst\_perindivdiual\_db,
2. For each row in **canopyPosition** field of vst\_perindividual\_perbout\_db:

if value in **canopyPosition** field == ‘1’

insert ‘Solitary - full sun’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘2’

insert ‘Dominant – full sun’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘3’

insert ‘Co-dominant – partially shaded’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘4’

insert ‘Intermediate – mostly shaded’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘5’

insert ‘Overtopped – full shade’ into cell in **canopyPositionName** field

1. Generate **stemStatusName** field in vst\_perindivdiual\_db,
2. For each row in **stemStatus** field of vst\_perindividual\_perbout\_db:

if value in **stemStatus** field == ‘1’

insert ‘Live’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘2’

insert ‘Dead’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘3’

insert ‘Removed’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘4’

insert ‘Damaged - insect’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘5’

insert ‘Damaged - disease’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘6’

insert ‘Damaged - abiotic’ into cell in **stemStatusName** field

Else:

if value in **stemStatus** field == ‘7’

insert ‘Damaged - other’ into cell in **stemStatusName** field

1. Verify that conditional null values in the vst\_perindividual\_perbout\_in sheet are valid
   1. generate **completeRecordCanopyPositionQF, completeRecordMaxCanopyDiameterQF, completeRecordNinetyCanopyDiameterQF, completeRecordVD1HeightQF, completeRecordVD2HeightQF, completeRecordShrubShapeQF, completeRecordMaxBaseCanopyDiameterQF, completeRecordNinetyBaseCanopyDiameterQF** populate with zeros

For each row in **tagID** field of vst\_perindividual\_perbout\_db:

If value in **canopyPosition** is NULL

If **growthForm** is not 'lia' OR 'sms' OR 'sis' OR 'sgr' OR 'smt' OR 'sap'

insert 1 in **completeRecordCanopyPositionQF** field of vst\_perindividual\_perbout\_db

If value in **maxCanopyDiameter** is NULL

If **growthForm** is ‘sis’ OR ‘sms’

insert 1 in **completeRecordMaxCanopyDiameterQF** field of vst\_perindividual\_perbout\_db

Else:

If value in **maxCanopy** is NULL

If **growthForm** is ‘sbt' OR 'mbt

If **plotType** is ‘Distributed’

insert 1 in **completeRecordMaxCanopyDiameterQF** field of vst\_perindividual\_perbout\_db

If value in **ninetyCanopyDiameter** is NULL

If **growthForm** is ‘sis’ OR ‘sms’

1. insert 1 in **completeRecordNinetyCanopyDiameterQF** field of vst\_perindividual\_perbout\_db

Else:

If **growthForm** is ‘sbt' OR 'mbt

If **plotType** is ‘Distributed’

1. insert 1 in **completeRecordNinetyCanopyDiameterQF** field of vst\_perindividual\_perbout\_db

If value in **VD1Height** is NULL

If **growthForm** is not 'lia' OR ‘mbt’

1. insert 1 in **completeRecordVD1HeightQF**

If value in **VD2Height** is NULL

If **growthForm** is not 'lia' OR ‘mbt’ OR ‘sms’ OR ‘smt’ OR ‘sis’

1. insert 1 in **completeRecordVDHeightQF**

If value in **shrubShape** is NULL

If **growthForm** is 'sms' OR ‘sis’

1. insert 1 in **completeRecordShrubShapeQF**

If value in **maxBaseCanopyDiameter** is NULL

If **maxBaseCanopyDiameter** is 'icn’

1. insert 1 in **completeRecordMaxBaseCanopyDiameterQF**
2. Generate **height** field in vst\_perindividual\_perbout\_db
3. If cells for BOTH **VD1Height** and **VD2Height** are non-null,
4. Calculate **height** = **VD1Height** – **VD2Height**
5. insert value for **height** into **height** cell of vst\_perindividual\_perbout\_db

Else

if only **VD1Height** is non-null

* + 1. insert value from **VD1Height** in to **height** cell of vst\_perindividual\_perbout\_db

## Run the following processing steps for vst\_shrubgroup\_perbout\_db

1. Duplicate **canopyArea, canopyPosition, aGroupHeight, bGroupHeight, cGroupHeight, dGroupHeight** and **eGroupHeight** fields for duplicate **date,** **plotID, shrubgroupID** combinations
2. Generate **subplotID** field in vst\_pershrubgroup\_perbout\_db
   1. For each row in **shrubGroupID** field of vst\_pershrubgroup\_perbout\_db
      1. insert first two characters from **shrubGroupID** field in **subplotID** field of vst\_pershrubgroup\_db
3. Generate **groupID** field in vst\_pershrubgroup\_perbout\_db
   1. For each row in **shrubGroupID** field of vst\_pershrubgroup\_perbout\_db
      1. insert last two characters from **shrubGroupID** field in **groupID** field of vst\_pershrubgroup\_db
4. Generate **canopyPositionName** field in vst\_pershrubgroup\_perbout\_db,
5. For each row in **canopyPosition** field of vst\_perindividual\_perbout\_db:

if value in **canopyPosition** field == ‘1’

insert ‘Solitary - full sun’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘2’

insert ‘Dominant – full sun’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘3’

insert ‘Co-dominant – partially shaded’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘4’

insert ‘Intermediate – mostly shaded’ into cell in **canopyPositionName** field

Else:

if value in **canopyPosition** field == ‘5’

insert ‘Overtopped – full shade’ into cell in **canopyPositionName** field

1. Generate **averageGroupHeight** field in vst\_pershrubgroup\_perbout\_db,
   1. For each row in vst\_pershrubgroup\_perbout\_db
      1. sum values in **aGroupHeight, bGroupHeight, cGroupHeight, dGroupHeight** and **eGroupHeight**

if cell is null, ignore

* + 1. divide sum from 2.a.i by the number on non-null cells included in the calculation
    2. insert value in **averageGroupHeight** field

1. Generate **multipleSpeciesShrubGroupQF** field in vst\_pershrubgroup\_perbout\_db, populate with zeros.
   1. Concatenate **date, plotID** and **shrubgroupID** fields to create temporaray **datePlotGroup** field
   2. Generate temporary field **groupPercent**
   3. For each unique **datePlotGroup** value
      1. Add values in the **percentVolume** field in vst\_pershrubgroup\_perbout\_db together
      2. insert summed value in **groupPercent** field

if value in **groupPercent** is NOT 100

insert ‘1’ in **multipleSpeciesShrubGroupQF** field

1. Generate **taxonIDStatusQF** field in vst\_pershrubgroup\_perbout\_db, populate with zeros
   1. Concatenate **date, plotID, shrubgroupID, taxonID** fields to create temporary **datePlotGroupTaxon** field
   2. Generate temporary field **taxonStatus**
   3. For each unique **datePlotGroupTaxon** value
      1. Add values from **percentLive** and **percentDead**
      2. insert summed value in **taxonStatus** field

if value in **taxonStatus** is NOT 100

insert ‘1’ in **taxonIDStatusQF** field

## Run the following processing steps for vst\_perother\_perbout\_db

1. Generate **nestedSubplotArea** field in vst\_perother\_perbout\_db
   1. For each row in **plotID** field of vst\_perother\_perbout\_db

insert value from **nestedSubplotAreaOther** field of the vst\_perplot\_db into **nestedSubplotArea** of the vst\_perother\_perbout\_db

if value in **nestedSubplotAreaOther** field of vst\_perplot\_db is NULL

leave **nestedSubplotArea** field of vst\_perother\_perbout\_db NULL

## All Databases: Run the following processing steps for all databases produced through the algorithms above

1. Generate **growthFormName** field in vst\_perindivdiual\_db, vst\_perindividual\_perbout\_db, and vst\_perother\_perbout\_db; populate with zeros
   1. For each row in **growthForm** field of in vst\_perindivdiual\_db, vst\_perindividual\_perbout\_db, and vst\_perother\_perbout\_db:

if value in **growthForm** field == ‘lia’

insert ‘Liana’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘sbt’

insert ‘Single Bole Tree’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘mbt’

insert ‘Multi-bole Tree’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘sms’

insert ‘Small Shrub’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘sis’

insert ‘Single Shrub’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘sgr’

insert ‘Shrub Group’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘sap’

insert ‘Sapling’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘frn’

insert ‘Fern’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘cac’

insert ‘cactus’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘yuc’

insert ‘Yucca’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘xer’

insert ‘Xerophyllum’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘agv’

insert ‘Agave’ into cell in **growthFormName** field

Else:

if value in **growthForm** field == ‘plm’

insert ‘Palm’ into cell in **growthFormName** field

1. Assign L1 data values, in vst\_datapub\_2014 workbook, from L0 data values, in the vst\_dataingest\_2014 workbook
   1. Run Assign: L1 Data from L0 Data, in AD[06], where:
      1. List of database tables =vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
      2. List of date and time fieldnames=(**date,**)
      3. List of technician fieldnames=( **measuredBy, recordedBy,**)
      4. List of non-transferring fieldnames= (**previouslyMeasuredDate)**
2. Generate quality flag summary
   1. Run Generate: Quality Flag Summary in AD[06], where:
      1. List of database tables = vst\_perplot\_db, vst\_perindividual\_db, vst\_perindividual\_perbout\_db, vst\_pershrubgroup\_perbout\_db, vst\_perother\_perbout\_db
3. Do not publish:
   * eventPlot
   * tagID
   * supportingStemTagID
   * point ID
   * Stem Azimuth
   * StemDistance
   * VD1Height
   * VD2Height
   * stemDigits
   * previouslyMeasuredDate
   * growthForm
   * datePlotGroup
   * groupPercent

# Uncertainty

Uncertainty in vegetation structure data may arise from field activities or from data entry (transcription) errors. Opportunity for mistakes that lead to increased uncertainty is greatest during field data collection. Misidentifying a pointID from which an individual is mapped can lead to placement of an individual in a different relative location than it actually occurs. This error can be identified if the incorrect pointID leads to a situation in which an individual is mapped to be outside the plot boundaries, however, if error leads to mapping an individual in a different subplot than it actually occurs, the error may not be caught until the following measurement bout, up to a year later. Failure to properly calibrate the laser range finder used to measure azimuth and distance of an individual from a known point may lead to erroneous measurements for both values causing uncertainty in the position within the plot of all mapped individuals. During annual re-measurement, differences in measurement technique between technicians can lead to differences in values for diameter, height, canopy etc. with a downstream effect of differences in estimated biomass and productivity based on the technician making the measurement.

## Analysis of Uncertainty

Technician errors associated with field data collection may be prevented with careful training and clearly worded, detailed standard operating procedures such as those provided in AD[10]. A system of hot and cold field checks may be used identify errors or inconsistencies in implementation of field protocols but are still essentially preventative in nature in that they cannot aid in quantifying the level of uncertainty introduced during field data collection.

## Reported Uncertainty

Data entry error rates can be quantified and reported through the system of re-checking entered data against data sheets (AD[18]) and, potentially, through the number of flags generated by the QA/QC algorithm provided here. Errors in species identification are possible and will be addressed and quantified according to the procedures laid out in the NEON Algorithm Theoretical Basis Document: Taxonomic Consensus Identifications and Uncertainty, AD[08].

# Validation and Verification

## Algorithm Validation

## Data Product Validation

## Data Product Verification

# Scientific and Educational Applications

NEON’s vegetation structure measurements are one component of measurement of aboveground net primary productivity at terrestrial NEON sites. The 30 year dataset that is generated by these measurements represents a significant contribution to the field of ecology. When analyzed in conjunction with data streaming from the Terrestrial Instrument System (TIS), the Airborne Observation Platform (AOP), and other terrestrial observations, vegetation structure measurements will help address grand challenge questions related to the responses to climate change, land use change and invasive species. The measurement of vegetation structure, as well as the mapping of free-standing woody stems, is an important complement to data streams generated by the NEON AOP and TIS. These ground-collected data will validate LiDAR data used to map the structural complexity of vegetation, will enable mapping of plant biomass at the site scale, and in conjunction with carbon flux data, will facilitate understanding how biomass in different plant growth forms contributes to ecosystem level carbon flux. Unique to NEON’s measurement of vegetation structure is the level of standardization employed across ecosystem types; it is this standardization that will enable scaling across the NEON observatory from the plot to site, domain, and continental scales.

# Future Modifications and Plans

As the QA/QC of raw vegetation structure data relies heavily on the NEON Algorithm Theoretical Basis Document for QA/QC Plausibility Testing of Organismal and Observation Based Field and Lab Data, AD[06], significant changes to that document will result in changes to this one. Planned additions to the plausibility ATBD include the inclusion of quantitative range tests; it is anticipated that these will be utilized heavily in future versions of this document. Setting *a priori* ranges for anticipated measurements by species is impractical given the number of species that will be measured across the Observatory; reasonable thresholds for testing measurements will be derived from existing NEON data and will be updated as more data become available.

The complete bout test, as it is currently written, only checks whether all tower plots were sampled for at least one vegetation category for nlcd classification likely to have woody vegetation present. In future iterations of this algorithm we intend to include greater specificity which may include a consistency test across calendar years or a complete records test associated with vegetation category and subplot. At this time, including this resolution would require modifications to both the sampling protocol and field datasheets which is not currently feasible.

# Bibliography

Jepson, W. L. 2012. The Jepson Manual: Vascular plant of California. 2nd edition. University of California Press.

Van Buren, R., J. G. Cooper, L. M. Shultz, and K. T. Harper. 2011. Woody plants of Utah. 1st edition. Utah State University Press.

# Changelog