NEON ALGORITHM THEORETICAL BASIS DOCUMENT (ATBD):

OS DATA QUALITY CONTROL

PREPARED BY	ORGANIZATION	DATE
Caren Scott	SCI	03/03/2026

APPROVALS	ORGANIZATION	APPROVAL DATE
Kate Thibault	SCI	03/03/2026

RELEASED BY	ORGANIZATION	RELEASE DATE
Tanisha Waters	CM	03/03/2026

See configuration management system for approval history.

The National Ecological Observatory Network is a project solely funded by the National Science Foundation and managed under cooperative agreement by Battelle. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Change Record

REVISION	DATE	ECO #	DESCRIPTION OF CHANGE
A	03/03/2026	ECO-07195	Initial release

TABLE OF CONTENTS

LIST OF TABLES

1 DESCRIPTION

This document describes Quality Control details for NEON Observation System data.

1.1 Purpose

Data quality assurance (QA) and quality control (QC) are integrated throughout the data collection, ingest, and processing of NEON Observation System (OS) data. This document focuses on post-hoc quality assessment procedures. For information about data quality at the collection and ingest steps, see the NEON Science Data Quality Plan (AD [05]). For information about the processing of observational data, see OS Generic Transitions (AD [04]). The post-hoc quality assessment procedures for OS data include reports specific to each Data Product (DP) that are generally run monthly. Each report contains checks on the completeness, timeliness and plausibility of the data. Several generic functions have been developed in each category, which are used by many or most of the OS data products. Each DP-specific report also contains custom checks unique to that DP. The goals for post-hoc QC include establishing a baseline for data quality across and within products, ensuring NEON is meeting data quality benchmarks (see NEON Science Availability Plan (AD[03]) for more details), identifying and correcting erroneous data (when possible), flagging erroneous data that cannot be corrected, and improving the data generating or ingesting processes.

1.2 Scope

This document describes the overall approach to quality control for NEON OS data, the types of checks that have been developed, and how the plan is implemented across data products. While there is much work done to constrain the data before ingest (i.e., prevent bad data from getting into the system), those upstream QC checks are described elsewhere (AD[05]). The focus of this document is to describe the QC procedures after the data have been ingested.

2 RELATED DOCUMENTS, ACRONYMS AND VARIABLE NOMENCLATURE

2.1 Applicable Documents

AD[01]	NEON.DOC.000001	NEON Observatory Design
AD[02]	NEON.DOC.002652	NEON Data Products Catalog
AD[03]	NEON.DOC.004764	NEON Science Availability Plan
AD[04]	NEON.DOC.004825	OS Generic Transitions
AD[05]	NEON.DOC.004104	NEON Science Data Quality Plan

2.2 Reference Documents

RD[01]	NEON.DOC.000008	NEON Acronym List
RD[02]	NEON.DOC.000243	NEON Glossary of Terms

2.3 Acronyms

Acronym	Explanation
ATBD	Algorithm Theoretical Basis Document
DP	Data Product
GCP	Google Cloud Platform
GCS	Google Cloud Storage
L0	Level 0
L1	Level 1
LOV	List of Values
LUT	Look Up Table
OS	Observation System
QA	Quality Assurance
QC	Quality Control

3 SCIENTIFIC CONTEXT

QC for OS data includes executing a custom report for each DP. The reports include generic checks, i.e., those that are used by many or most DPs, as well as custom checks unique to specific data products. These DP-specific reports are generally run monthly (although some are only run annually) and check recently published provisional data available on the NEON Data Portal. The reports produce an HTML document, summary metrics, and tables of flagged records, all of which can be viewed on the internal OS quality dashboard.

NEON OS science staff review reports and quality metrics monthly and take steps to address or correct data issues as identified.

Each report contains checks that are grouped into three data quality categories: completeness, timeliness, and plausibility. Each category is explained briefly here with examples of the types of checks that fall within each. See the full list of generic functions in **Section 3.1.2** for more details. The prefix of each check name indicates the data quality category to which it belongs.

These metrics present detailed views of the two basic measures of availability that NEON has adopted to monitor performance across all data products (AD[03]); Sebastian-Coleman 2013):

- Completeness (technical availability): The quantity of data (e.g., number of records, data files) published over a period of time, compared to the amount of data expected.
- Validity (scientific availability): The proportion of data published over a period of time that has passed all quality checks.

3.1 Theory of Algorithm

3.1.1 Data Quality Metrics

3.1.1.1 Completeness

In addition to the comprehensive metric of completeness defined above, the OS routines also include checks on several ‘levels’ of completeness. For example:

- In a given year, were the correct number of bouts (i.e., distinct sampling events) collected, according to the NEON Science design and sampling protocols?
- Within each of those bouts, were records collected from the correct number of sampling locations?
- Within each of those records, was information recorded for all of the high-priority fields?
- Are there data records present in the tables of laboratory measurements for all of the field-collected samples?

3.1.1.1.1 Subcategory: Pipeline Completeness

A few of the generic functions within the completeness category also provide metrics for a subcategory of completeness called 'pipeline completeness'. The pipeline completeness category captures completeness of recording and processing data, regardless of the contents of the data; specifically, whether the correct numbers of data records are present for all expected sampling times and locations, even if some of those records are in fact 'placeholders' for sampling that was scheduled but could not be carried out. For almost all OS DPs, NEON includes records as placeholders for data that could not be collected for a variety of reasons (e.g., due to local conditions such as dry stream beds, lack of site access due to road closure, etc.). These placeholder records allow users to know that sampling was intended but missed and they should not expect actual data records for those locations and time periods. These records are labeled as sampling impractical (i.e., they include a value other than 'ok' or blank in the 'sampling impractical' field in the most upstream table in the DP).

Due to sampling impractical records, metrics based on counts of bouts or counts of records in each bout would be inflated relative to the amount of sampling actually performed. Therefore, we split these metrics into 2 categories:

1) Did NEON collect the correct number of bouts/records/locations as defined in the protocol (Category = Completeness, described above)
2) Did NEON also include the right number of records to represent the missing data, i.e., records that are labeled as sampling impractical (Category = Pipeline Completeness).

3.1.1.2 Timeliness

The timeliness category includes a variety of checks; however, there are some DP-specific scripts that do not include any timeliness checks if none are relevant for the DP. Examples of timeliness checks:

- Were bouts completed during designated sampling windows?
- Did bouts have the appropriate spacing and/or duration?
- Were samples processed within relevant time limits?

3.1.1.3 Plausibility

The plausibility category includes a variety of checks. For example:

- Are any numeric values outliers?
- For repeated measures (taxonomic ID of a plant or animal, diameter of a tree), are values consistent or display a realistic change over time?
- Is there consistency compared to historical ranges?
- Are taxonomic data of adequate resolution according to the NEON Science design?

- Do values in one field make sense given values in other fields?
- How often are values other than 'ok' used for fields describing condition of the samples?

This category tends to have the largest number of custom checks specific to each DP script, given the diverse requirements needed to understand DP specific issues that indicate data plausibility concerns.

3.1.1.4 Validity

Validity for OS data is calculated from a combination of all timeliness and plausibility checks for each DP.

3.1.2 Generic Functions

To standardize OS QC and support aggregated reporting and metrics, a function library was developed in R via an internal (i.e., not publicly available) package called 'neonOSqc'. Within the neonOSqc package, there are functions that require user-specified inputs and variables to conduct standard checks on OS data tables. Below are details for each of the generic functions used across the DP-specific reports. See **Table 1** for a list of which functions are used in each DP.

3.1.2.1 complete_bout

The 'complete_bout' function checks if the number of bouts per year meets expectations according to the relevant NEON sampling protocol. Bout number requirements are listed in the companion look-up table (LUT) that accompanies the function. The function also checks if all sites in the LUT for that year are in the data, and vice versa. Alerts from this function indicate that too many or too few bouts were collected for that site for that year. For example, for the 'Sediment chemical and physical properties' product (DP1.20194.001), the LUT indicates that all aquatic sites expect two bouts per year; therefore, the function will return an alert for any site that has more than or fewer than two bouts.

This function returns metrics for both 'pipeline completeness' (includes sampling impractical) and 'completeness' (excludes sampling impractical).

3.1.2.2 complete_cross_table

The 'complete_cross_table' function checks if records from an upstream table are present in a downstream table, usually using a sample ID as the linking variable. Upstream tables are the point of sample creation in the field, whereas downstream tables include follow-on activities (measurements, subsampling, etc.) on those same samples. Alerts from this function indicate a sample that exists in the upstream creation table is missing from a downstream measurement table. For example, for the 'Stable isotopes in surface water' product (DP1.20206.001), we expect all samples collected in the 'asi_fieldData' table to exist in the 'asi_externalLabH2OIsotopes' table; therefore, this function will return an alert for any sample that is missing from the downstream table.

3.1.2.3 complete_within_bout

The ‘complete_within_bout’ function checks if the number of records collected per bout meets expectations according to the protocol, and/or if the number of named locations (plots, stream reaches, etc.) per bout meets expectations, as specified in the LUT. Alerts from this function indicate that too many or too few records were collected within a bout. For example, for the ‘Periphyton, seston, and phytoplankton collection’ product (DP1.20166.001), the LUT indicates that all aquatic stream sites expect 8 records per bout in the ‘alg_fieldData’ table; therefore, the function will return an alert for any bout that has more than or fewer than 8 records.

This function returns metrics for both 'pipeline completeness' (includes sampling impractical) and 'completeness' (excludes sampling impractical).

3.1.2.4 complete_within_rec

The ‘complete_within_rec’ function checks if expected variables contain values for each record. Alerts from this function indicate a value is missing when it should be filled in. The pre-ingest quality checks in place through mobile data entry apps will usually enforce required fields (see AD [05]), so this function double-checks if pre-ingest quality checks have been implemented correctly. For example, for the ‘Soil physical and chemical properties, periodic’ product (DP1.10086.001), we expect values in each of these fields: sampleID, collectDate, sampleTopDepth, sampleBottomDepth, soilTemp, boutType, horizon, etc.; therefore, the function will return an alert for any records where any of those fields are not filled in.

3.1.2.5 complete_within_rec_null

The ‘complete_within_rec_null’ function is the opposite of the ‘complete_within_rec’ function, it checks if specific variables are blank given certain conditions. Alerts from this function indicate a value is filled in when it should be blank. For example, in the ‘Macroinvertebrate collection’ product (DP1.20120.001), ponar depth should not be filled in at a stream site because the ponar sampler type is not used at streams; therefore, this function will return an alert if ponar depth is filled in at a stream site.

3.1.2.6 timely_bout

The ‘timely_bout’ function checks if bouts have been conducted within the appropriate time window, according to the NEON protocol or, for aquatic data products, the Site Sampling Design used to collect the data and specified in the LUT created for each DP. Alerts from this function indicate collection events that occurred before or after the sampling windows specified in the LUT. For example, the ‘Soil physical and chemical properties, periodic’ (DP1.10086.001) protocol defines each site’s sampling window to capture the correct biophysical criteria; therefore, this function will return an alert for any sample collected outside the sampling window.

3.1.2.7 timely_bout_duration

The 'timely_bout_duration' function checks if bout duration, in days, meets expectations. Alerts from this function indicate bouts lasted longer than the threshold specified in the function call. For example, for the 'Root biomass and chemistry, periodic' product (DP1.10067.001), the protocol specifies bouts should last no more than 42 days; therefore, this function will return an alert for any bout that lasted longer than 42 days.

3.1.2.8 timely_bout_spacing

The 'timely_bout_spacing' function checks if spacing between bouts, in days, meets expectations. Alerts from this function indicate bouts were completed too close to each other in time. For example, for the 'Small mammal box trapping' (DP1.10072.001) product, sampling bouts are required to be scheduled at least 21 days after the end of the previous bout; therefore, this function will return an alert for any pair of bouts with less than 21 days in between.

3.1.2.9 timely_process

The 'timely_process' function checks if activities associated with sample processing (sorting, oven drying, extracting, measuring, analyzing) are conducted within expected timelines. Alerts from this function indicate the processing step took more time than specified in the LUT. For example, for the 'Soil inorganic nitrogen pools and transformations' product (DP1.10080.001), the protocol specifies to perform extraction by the end of the day following the collection date; therefore, this function will return an alert for any samples where the extraction occurred more than 36 hours after collection.

3.1.2.10 plausible_count_lov

The 'plausible_count_lov' function summarizes the prevalence of anomalous conditions for one or more data columns that are populated by list of value (LOV) options. Alerts from this function indicate the number of records that contain the anomalous conditions in the specified column. For example, in the 'Macroinvertebrate collection' product (DP1.20120.001), incorrect sample timing can be recorded in the biophysical criteria column; therefore, this function will return an alert for any record that contains a value other than 'OK - no known exceptions' in this field.

3.1.2.11 plausible_cross_table_taxon_check

The 'plausible_cross_table_taxon_check' function checks if a taxon ID is consistent across tables. Alerts from this function indicate a mismatch in taxon IDs between tables. For example, in the 'Plant foliar traits' product (DP1.10026.001), we expect the taxon ID of the individual ID to match the taxon ID of the same individual ID in the 'Vegetation structure' product (DP1.10098.001) 'vst_mappingandtagging' table; therefore, this function will return an alert for any individual ID where the taxon ID does not match between tables.

3.1.2.12 plausible_percentile_range

The ‘plausible_percentile_range_check’ function checks numeric values against expected ranges as specified in the LUT. These expected ranges can come from several sources, including NEON-measured historic ranges, external data sources, or protocol requirements. Alerts from this function indicate a numeric value in the current data is outside the expected range from the LUT. For example, in the ‘Chemical properties of surface water’ product (DP1.20093.001), historic ranges have been calculated for each analyte in the ‘swc_externalLabDataByAnalyte’ table for each variable-site-season combination using an interquartile range approach; therefore, this function will return an alert for any sample with an analyte outside the specified range.

3.1.2.13 plausible_repeat_cat

The ‘plausible_repeat_cat’ function checks if repeated categorical measurements of the same sample/individual/location/etc. deviate from expectations. Alerts from this function indicate an unacceptable change in a repeated measurement. For example, in the ‘Small mammal box trapping’ product (DP1.10072.001), the sex of a recaptured mammal is expected to be consistent through time; therefore, this function will return an alert for any pair of records where the sex has changed between consecutive captures.

3.1.2.14 plausible_repeat_num

The ‘plausible_repeat_num’ function checks if repeated numerical measurements of the same sample/individual/location/etc. deviate from expectations. Alerts from this function indicate a numeric value has changed more than the threshold indicated in the function call. For example, in the ‘Reaeration field and lab collection’ product (DP1.20190.001), wetted widths are expected to be relatively consistent between consecutive bouts; therefore, this function will return an alert for any two bouts where the wetted width at the same stream transect has changed by more than 2 meters.

3.1.2.15 plausible_taxon_rank_by_record

The ‘plausible_taxon_rank_by_record’ function checks if the taxon in each record is identified to the desired taxonomic resolution. Alerts from this function indicate the taxon rank is coarser than expected. For example, in the ‘Mosquitoes sampled from CO2 traps’ product (DP1.10043.001), the expert taxonomist is expected to identify all mosquitoes to the species level; therefore, this function will return an alert for any record in which the taxon rank is coarser than species in the ‘mos_expertTaxonomistIDProcessed’ table.

3.1.2.16 plausible_taxon_rank_within_bout

The ‘plausible_taxon_rank_within_bout’ function checks if the taxa in a bout are identified to the desired taxonomic resolution for an acceptable percentage of records. Alerts from this function indicate too few records in the bout (specified in the function call) meet the desired taxon rank. For example, in

the 'Macroinvertebrate collection' product (DP1.20120.001), at least 75% of the taxa are expected to be identified to at least the genus level; therefore, this function will return an alert for any bout where less than 75% of the taxa were identified to the genus level.

3.1.2.17 plausible_within_group

The 'plausible_within_group' function checks for outliers (as defined by the user) within a group of related measurements. Alerts from this function indicate a possible outlier, a measured value that is suspect compared to the rest of the group. For example, in the 'Groundwater and active layer measurements at permafrost sites' product (DP1.20099.001), thaw depth values are expected to be similar throughout the site at the same time; therefore, this function will return an alert for any thaw depth records that are suspect compared to the central tendency values of the group of samples from the same sampling event.

Table 1. List of functions used in each DP-specific report.

dpID	complete_bout	complete_cross_table	complete_within_bout	complete_within_rec	complete_within_rec_null	timely_bout	timely_bout_duration	timely_bout_spacing	timely_process	plausible_count_lov	plausible_cross_table_taxon_chec	plausible_percentile_range_check	plausible_repeat_cat	plausible_repeat_num	plausible_taxon_rank_by_record	plausible_taxon_rank_within_bout	plausible_within_group
DP1.00013.001	X	X		X				X	X	X		X					
DP1.00038.001	X	X		X						X		X					
DP1.10003.001	X	X	X	X		X	X					X					X
DP1.10010.001	X		X	X	X	X	X				X	X	X		X		
DP1.10014.001	X	X	X	X					X		X	X	X				
DP1.10017.001	X	X	X				X										
DP1.10020.001		X		X						X							
DP1.10022.001	X	X	X	X	X	X	X	X	X	X	X				X	X	X
DP1.10023.001		X	X	X	X	X	X	X		X		X					
DP1.10026.001	X	X	X	X	X	X	X		X	X	X	X	X		X	X	
DP1.10033.001	X	X	X	X	X	X	X	X	X	X		X					
DP1.10038.001		X		X						X							
DP1.10055.001	X	X						X		X					X		X
DP1.10058.001	X	X	X			X	X			X		X			X	X	
DP1.10064.002		X		X								X					
DP1.10067.001	X	X	X	X	X	X	X		X	X		X					

dpID	complete_bout	complete_cross_table	complete_within_bout	complete_within_rec	complete_within_rec_null	timely_bout	timely_bout_duration	timely_bout_spacing	timely_process	plausible_count_lov	plausible_cross_table_taxon_chec	plausible_percentile_range_check	plausible_repeat_cat	plausible_repeat_num	plausible_taxon_rank_by_record	plausible_taxon_rank_within_bout	plausible_within_group
DP1.10072.001	X	X	X	X	X		X	X		X		X	X	X		X	
DP1.10076.001		X		X						X							
DP1.10086.001	X	X	X	X	X	X	X	X	X	X		X		X			X
DP1.10092.001			X	X													
DP1.10093.001	X	X	X	X		X	X	X		X		X					
DP1.10098.001		X	X									X					
DP1.10104.001				X					X	X		X					
DP1.10111.001		X		X													
DP1.20048.001	X	X	X	X						X							
DP1.20063.001	X	X	X	X		X			X	X		X					
DP1.20066.001	X	X	X	X		X		X		X		X	X			X	
DP1.20072.001	X	X	X	X		X		X		X			X			X	
DP1.20092.001	X	X	X	X		X			X	X		X					
DP1.20093.001	X	X	X	X					X	X		X		X			
DP1.20097.001	X	X	X	X					X	X		X					
DP1.20099.001			X	X		X		X						X			X
DP1.20105.001		X		X						X							
DP1.20107.001	X	X	X	X		X	X	X		X		X	X	X	X		X
DP1.20120.001	X	X	X	X		X		X		X		X	X		X	X	
DP1.20126.001	X	X		X		X				X					X	X	
DP1.20138.001	X	X	X	X		X		X		X		X					
DP1.20163.001	X	X	X	X		X			X	X		X					
DP1.20166.001	X	X	X	X		X		X	X	X		X	X		X	X	
DP1.20190.001	X	X	X	X	X					X		X					X
DP1.20191.001	X		X	X			X					X					
DP1.20193.001	X	X															
DP1.20194.001	X	X	X	X		X			X	X		X					
DP1.20206.001	X	X	X	X				X	X	X		X					
DP1.20219.001	X	X	X	X		X		X		X		X	X		X		
DP1.20221.001	X	X		X		X				X					X	X	
DP1.20252.001	X			X	X					X		X					

dpID	complete_bout	complete_cross_table	complete_within_bout	complete_within_rec	complete_within_rec_null	timely_bout	timely_bout_duration	timely_bout_spacing	timely_process	plausible_count_lov	plausible_cross_table_taxon_chec	plausible_percentile_range_check	plausible_repeat_cat	plausible_repeat_num	plausible_taxon_rank_by_record	plausible_taxon_rank_within_bout	plausible_within_group
DP1.20254.001	X	X	X	X	X					X		X					
DP1.20267.001	X			X						X							
DP1.20275.001	X		X	X		X						X					
DP1.20276.001	X	X	X	X		X				X		X					
DP1.20279.001	X	X		X													
DP1.20280.001	X	X	X	X		X		X		X		X					
DP4.00131.001	X	X	X	X													
DP4.00132.001	X		X	X		X											
DP4.00133.001	X																

4 ALGORITHM IMPLEMENTATION

4.1 QC Pipeline

For each DP, there is a script in a docker container stored in GitHub, and a corresponding job in NEON's Google Cloud Platform (GCP) environment. When each script is executed on an automated schedule (generally monthly), it creates report outputs that are stored in NEON's Google Cloud Storage (GCS). These outputs include HTML documents containing the outcome of every check, as well as summary metrics and tables of alerts from each check. The outputs can be viewed and/or downloaded from NEON's internal OS dashboard or pulled directly from GCS.

4.1.1 OS Dashboard

The internal OS dashboard displays compiled metrics from all DPs and all sites. In contrast to the individual alerts, which are used on a granular level to identify specific problems in data collection and sample processing (see **Section 4.1.2** below), the compiled metrics are used to evaluate the quality of observational data at a high level. Every function within a DP-specific script, including the custom checks, creates standard outputs which contain a summary table that includes the count of alerts vs the total count of data records (or bouts, or data tables, depending on the resolution of the alert). These counts are summarized per site as well as across all sites combined. As there are several checks in each category in each script, summaries are grouped together to roll up into one metric per category per DP per site. See **Table 2** for a count of custom checks in each category for each DP. The metrics are compiled as described in the sections below.

4.1.1.1 Completeness and Pipeline Completeness

Metrics for Completeness and Pipeline Completeness are calculated by a series of steps at several levels.

- Date by DP by Function by Table:
    - The summary tables are all combined and a new percent complete is calculated as the inverse of a new count of alerts divided by a new total number of records. This is done instead of averaging the percent complete across the original summary tables for each check to avoid excessive weighting caused by running checks on small subsets of data.
- Date by DP by Function:
    - These new percent complete values per table are then averaged within a function (across tables) to get a single value for each function for each DP for each date.
- Date by DP:
    - These averaged values per function are then multiplied across all completeness, or pipeline completeness, functions within a DP (including custom functions) to get an overall completeness metric for each DP for each date. Percent complete values are multiplied

across functions in Completeness and Pipeline Completeness because only the records/bouts/locations etc that passed a previous check can be run in an additional check. For example, if complete_bout returns 50% complete (only half the bouts were sampled), then only those 50% of bouts are evaluated in complete_within_bout; if only half of the bouts were completed, the maximum completeness should be 50%.

- Date:
    - Finally, for the overall NEON metrics, the values are then averaged across DPs to get a single value per date and then averaged across dates to get the final NEON Summary metric.

4.1.1.2 Plausibility and Timeliness

Metrics for Plausibility and Timeliness are calculated by combining all summary tables within each category for each Date and DP and taking the average of the percent plausible/timely values. We are currently exploring options to remove excessive weighting caused by running checks on small subsets of data.

Note, these categories do not include a multiplication step in the metrics rollup because each function evaluates the appropriate data regardless if they passed a previous check. For example, records are evaluated for 'plausible_percentile_range_check' regardless of whether they passed the check for 'plausible_repeat_num'.

4.1.1.3 Validity and Availability

Validity is the average of the combined set of Plausibility and Timeliness summaries (not the average of the Plausibility and Timeliness averages).

Availability = Validity * Completeness

4.1.2 Interpretation of QAQC Metrics

The 'flags' resulting from each function are considered alerts and do not necessarily indicate a problem in the data. The Science lead for each DP investigates each flagged record and determines the appropriate action. Possible actions include, but are not limited to:

- Edit the data to fix a problem that resulted from a data entry error, if possible.
- Edit the data to include a quality flag, sample condition, or other value so the end user is aware of a potential problem.
- Delete the data if the error is egregious.
- Do nothing because an investigation determines that the data are fine or potentially fine.

- Update protocols, training materials, and/or data collection and ingest procedures to better constrain incoming data and prevent future issues.

Through time, the DP-specific reports will be updated to reflect better thresholds in each of the function calls to reduce the number of flagged records which are determined to be fine upon NEON staff review. At this time, results from these functions will not be visible to end users; they are used by NEON staff only.

Table 2. Count of custom quality checks in each category.

dpID	Completeness	Plausibility	Timeliness
DP1.00013.001	1	2	1
DP1.00038.001	3	1	0
DP1.10003.001	4	3	0
DP1.10010.001	4	0	0
DP1.10014.001	4	0	0
DP1.10017.001	1	3	1
DP1.10020.001	0	5	0
DP1.10022.001	0	3	0
DP1.10023.001	2	0	0
DP1.10026.001	7	4	1
DP1.10033.001	5	6	0
DP1.10038.001	0	6	0
DP1.10055.001	4	2	1
DP1.10064.002	3	2	0
DP1.10067.001	5	3	0
DP1.10072.001	0	22	1
DP1.10076.001	0	6	0
DP1.10086.001	1	21	2
DP1.10092.001	0	5	0
DP1.10093.001	0	7	0
DP1.10098.001	6	7	1
DP1.10104.001	4	5	0
DP1.10111.001	4	4	0
DP1.20048.001	0	1	0
DP1.20063.001	3	0	0
DP1.20066.001	5	1	0
DP1.20072.001	8	1	0
DP1.20092.001	1	2	0
DP1.20093.001	1	1	0

dpID	Completeness	Plausibility	Timeliness
DP1.20097.001	1	0	0
DP1.20099.001	2	0	0
DP1.20105.001	0	6	0
DP1.20107.001	5	0	0
DP1.20120.001	6	2	0
DP1.20126.001	13	1	0
DP1.20138.001	3	1	0
DP1.20163.001	6	1	0
DP1.20166.001	6	1	0
DP1.20190.001	0	3	0
DP1.20206.001	0	1	0
DP1.20219.001	6	2	0
DP1.20221.001	12	1	0
DP1.20252.001	1	1	0
DP1.20254.001	2	4	0
DP1.20276.001	0	0	1
DP1.20279.001	1	0	0
DP1.20280.001	1	1	0
DP4.00131.001	1	16	0
DP4.00132.001	0	8	0
DP4.00133.001	2	0	0

4.2 Timing of Report Execution

4.2.1 Monthly Reports

Reports are executed monthly, evaluating the month of data that was just recently published. Note that there is a lag between the data collection and data ingest and publication, so the collection dates of the records being evaluated will be older than one month. How much older depends on the data product. For some data products, field data are ingested and published within a few weeks, and therefore, the report evaluates data that were collected 1-2 months prior. For other data products, processing occurs after sample collection (either in the domain support facility or at an external lab) and this processing can take weeks to months. For data products with sample processing, the reports evaluate data that were collected several months prior. Many data products have multiple lags, one for the field data and one for the lab data. See **Table 3** for a list of the lags for each data product.

Table 3. Lag times (in months) for each data product.

dpID	Field	Lab	Full
DP1.00013.001			5
DP1.00038.001			5
DP1.10003.001			2
DP1.10010.001			3
DP1.10014.001			3
DP1.10017.001			2
DP1.10020.001			24
DP1.10022.001	2	24	
DP1.10023.001			3
DP1.10026.001	4	7	
DP1.10033.001	2	7	
DP1.10038.001			24
DP1.10055.001			2
DP1.10058.001			5
DP1.10064.002			9
DP1.10067.001	9	12	
DP1.10072.001			3
DP1.10076.001			24
DP1.10086.001	4	5 or 7	
DP1.10092.001			7
DP1.10093.001	2		7
DP1.10098.001			6
DP1.10104.001			4
DP1.10108.001			9
DP1.10111.001			2
DP1.20048.001			3
DP1.20063.001	2		7
DP1.20066.001	2		7
DP1.20072.001	2		7
DP1.20092.001	2		7
DP1.20093.001	2		7
DP1.20097.001	2		7
DP1.20099.001			2
DP1.20105.001			24
DP1.20107.001			2
DP1.20120.001	2		7
DP1.20126.001			7
DP1.20138.001	2		7
DP1.20163.001	2		7

dpID	Field	Lab	Full
DP1.20166.001	2		7
DP1.20190.001			2
DP1.20191.001			2
DP1.20193.001			2
DP1.20194.001	2		4
DP1.20206.001	2		8
DP1.20219.001	2		7
DP1.20221.001			7
DP1.20252.001			2
DP1.20254.001			2
DP1.20267.001			1
DP1.20275.001			2
DP1.20276.001	2		6
DP1.20279.001	2		7
DP1.20280.001	2		9
DP1.20282.001			9
DP4.00131.001			12
DP4.00132.001			12
DP4.00133.001			3

4.2.2 Annual Reports

In addition to the reports that run on a monthly cadence described above, annual reports are produced automatically, generally when the monthly script that examines data from December of that year is run. Therefore, the annual report for every DP is executed automatically on a different time lag. Due to the desire to QC all data prior to NEON's annual Data Release every year, annual reports are also manually executed for all DPs in the couple of months leading up to the Release, regardless of the typical time lag for each DP. All of the checks included in the monthly reports are also found in the annual reports, but, in addition, the annual reports often contain checks that can only be conducted when a full year of data are examined, for example number of bouts, spacing of bouts, etc.

5 FUTURE PLANS AND MODIFICATIONS

DP-specific QC reports will evolve and improve over time. Checks will be refined, thresholds will be adjusted, and additional checks will be included.

6 BIBLIOGRAPHY

Sebastian-Coleman, Laura. (2013). Measuring Data Quality for Ongoing Improvement. Measuring Data Quality for Ongoing Improvement. 10.1016/B978-0-12-397033-6.00020-1.