# SENTINEL R Shiny Application User Guide

Version: 1.0 (February 2024)

## Contents

## App Overview

Sentinel (**SEN**sor In**T**ell**I**ge**N**t **E**missions **L**ocator) is an application developed in R Shiny[1,2]. It is currently in a development/prototype phase. This application is intended to provide users with standardized visualization and graphical insights to Sensit low-cost fenceline sensor (SPod) data. These sensors can generate a large amount of data, which can be overwhelming for users to process manually. This application is also useful for generating manual and automatic Quality Assurance (QA) data flags as well as QA tables as described by the SOP for Sensit SPod Fenceline Sensor and Canister Grab Sample System Deployment and Operation (J-AMCD-SFSB-SOP-4380-2).

This app can be deployed by downloading and running the code on the Sentinel Github repository in an instance of R Studio. An overview diagram of the application inputs and outputs is shown below in Figure 1:

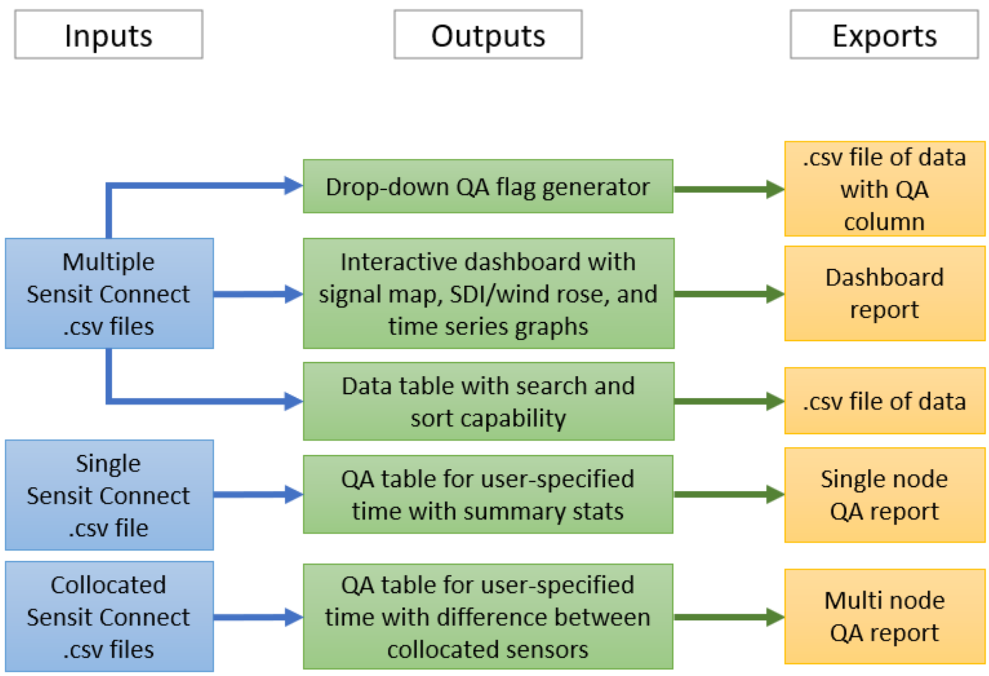


Figure 1: Application Map

All pages of SENTINEL contain the sidebar (which can be closed with the three horizontal lines button) and the main page. The sidebar options are QA Flagging (the landing page, where the user can add QA flags), Data Upload (Where files are uploaded for use in the app's visualization features), Dashboard (an interactive dashboard of uploaded data), Data table (a searchable and sortable tabular form of uploaded data), QA Tables (single node or multi node tables that summarize summary stats) and About (a page that contains resources). The user can click the NGEM logo to be taken to the [NGEM webpage](NGEM webpage).

# Downloading Data with Sensit Connect

Data should be downloaded from the Sensit Connect Website (sensitconnect.net) at a daily frequency for each individual sensor (Figure 2). Sensit SPod sensors are programmed automatically to report data at 30 seconds. This is acceptable by the app; however, the original code is programmed for 10 second frequency of data output. For more information on changing output settings on Sensit SPods, consult the Sensit Spod user manual.

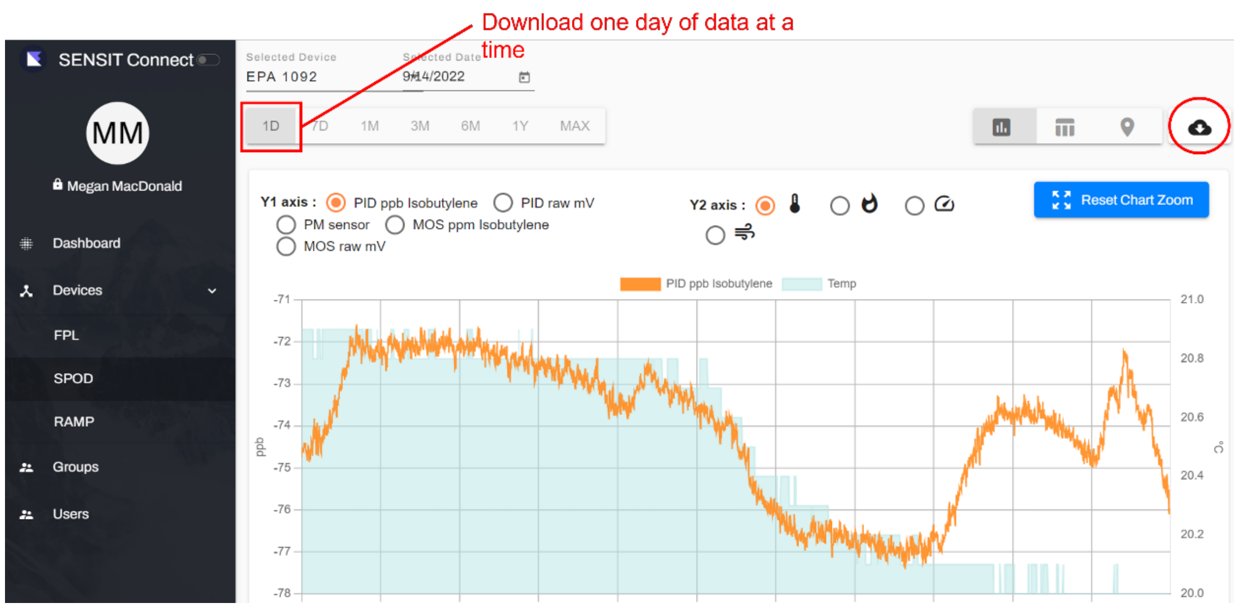


Figure 2: Sensit Connect Site

Data must be downloaded as a .csv file with the following naming convention:

**"SPOD_Data_Export_1181_2022-07-10.csv"**

Where "SPOD_Data_Export_" is automatically generated by Sensit Connect, "1181" is the Sensor ID, and "2022-07-10" is the date of the data collection. These values will be parsed by the processing code, so it is important to name the files in this way exactly. These files can be saved anywhere, but it is recommended to keep them grouped in a folder for ease of uploading to the SENTINEL Shiny app.

# QA Flagging

This interface allows the user to indicate time periods during which there is a reason to discount that data. The user must read in a data file downloaded from Sensit Connect in the standard naming convention. The autoQA features in the software will look for repeated values or offscreen values and flag those automatically when the data is read in. The user can make edits in the table using the dropdown menu for the QA column and typing the value into the box below to add offsets to the wind direction column (flag 7). Download the .csv file and read it in along with other downloaded Sensit Connect files to bring the QA edits into the SENTINEL. Add the flags shown on the app screen (Figure 3) in the QA column to designate specified times periods to be ignored in future analysis of this data file.

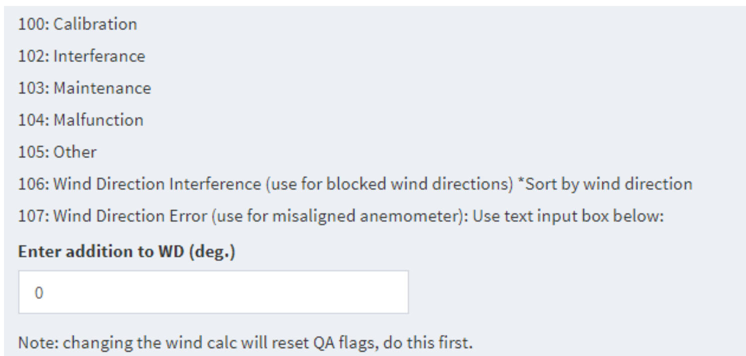

Figure 3: Manual QA Options

The flags described above can be applied to flag certain data rows that the user knows would not pass QA. The user can select the drop-down arrow to apply flags in the QA column and manually change values by typing into the cells. A calibration is when the user conducts calibration with zero air or calibration gas, or when the user conducts a cal check. Leaving these periods in the dataset could leave false signal in the analysis. Interference describes anything that could impact sensor readings or performance, such as a truck parking next to the sensor. Maintenance describes periods where the user is working on or updating the sensor or any attached power systems such as solar panels. Malfunction describes a period where the user knows the sensor is not working properly. Other is an empty flag that the user can define. Wind direction interference applies to time periods where there is an interference from a certain wind direction. For example, if the sensor is mounted on a pole, the direction of the pole is an impacted wind direction range, and the user might want to remove time periods where the wind came from this direction. Wind direction error applies to time periods where the anemometer was misaligned or malfunctioning. The user should apply this flag and then enter the amount (in degrees) to be added to the wind direction for all values in the table.

Like an excel table, the user can select multiple rows by highlighting a cell and dragging down from the right-hand corner. This can also be done in the QA column if the user doesn't want to use the drop-down functionality. The user should enter the precise latitude and longitude coordinates and drag the values down here if they were not logged by the unit. This is highlighted below in Figure 4.

| | ıs | trig.trig_value | trig.trig_activeFlag | trig.trig_eventFlag | lat | long | time | QA | SN |
|---|---|---|---|---|---|---|---|---|---|
| 19 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:09 | 0 | 1261 |
| 20 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:19 | 0 | 1261 |
| 21 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:29 | 100 | 1261 |
| 22 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:39 | 102 | 1261 |
| 23 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:49 | 103 | 1261 |
| 24 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:03:59 | 104 | 1261 |
| 25 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:09 | 105 | 1261 |
| 26 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:19 | 106 | 1261 |
| 27 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:29 | 107 | 1261 |
| 28 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:39 | 0 | 1261 |
| 29 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:49 | 0 | 1261 |
| 30 | 207 | | | 204 | 0.00 | 0.00 | 2023-03-07 00:04:59 | 0 | 1261 |

Figure 4: Interactive QA Data Frame

The user has the option to download this QA-checked data as a .csv file. The file will automatically be named with the correct naming convention and "_QA.csv" appended to the end of the name. These QA-checked files can be read into the data upload page in combination with raw files downloaded from Sensit Connect for processing and use in the app.

# Data Upload

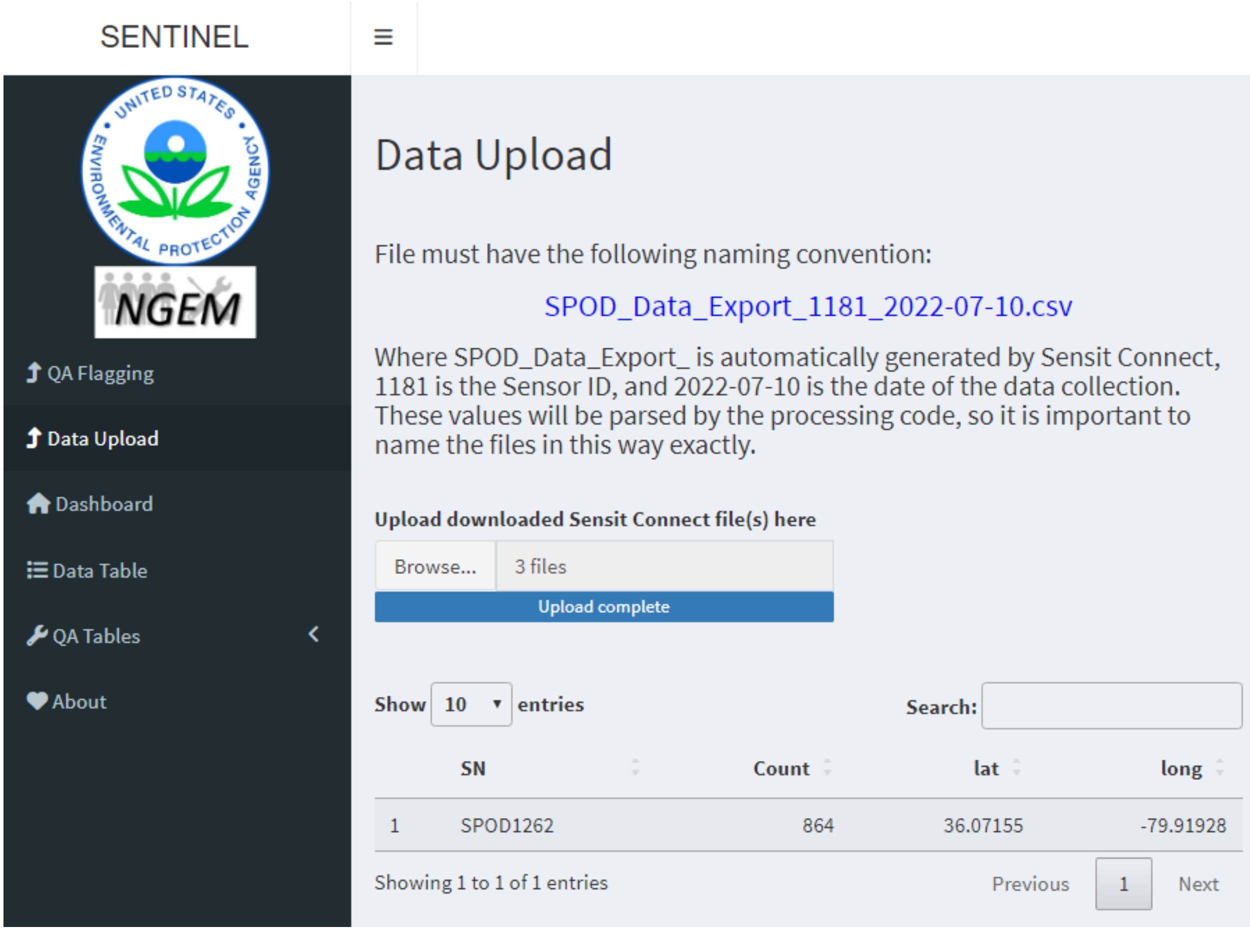


Figure 5: Data Upload page with 1 day of data entered

The user should upload files with the same naming convention they were saved in, described in the "Downloading Data with Sensit Connect" section. These files can be directly download from the Sensit site or run through the QA Flagging page of the app first. These files will have "_QA" appended to the end of the file name.

Once these files are uploaded through the Data Upload page (Figure 5), a status bar in the form of three blue vertical lines will appear as the files are processed. This processing involves applying automatic QA functions, that scan the data and look for values out range or repeating values and flag them accordingly in an appended QA column. This column is later visible in the Data Table viewer. Flag values in this column are shown below in Table 1.

Table 1: QA Flags (user-defined and auto QA)

| # | Description | Type |
|---|---|---|
| 0 | No QA issues, passing values | auto |
| 100 | Calibration period | user-defined |
| 102 | Interference | user-defined |
| 103 | Maintenance | user-defined |
| 104 | Malfunction | user-defined |
| 105 | Other - user defined | user-defined |
| 106 | Wind Direction Interference | user-defined |
| 107 | Wind Direction Error | user-defined |
| 108 | Repeated WS values (> 30 consecutively) | auto |
| 109 | Repeated WD values (> 30 consecutively) | auto |
| 110 | Repeated PID values (> 30 consecutively) | auto |
| 111 | Off screen WS check (> 12 mph) | auto |
| 112 | Off screen WD check (< 0 or > 360 degrees) | auto |

The script will then conduct baseline correction on the 10 second data using the getBaseline function, which is stored in the app folder. This function calculates a baseline with the “df” input set to 10, which is a slowly varying fit. It then subtracts this out to minimize any environmental drift present in the data.

The u and v vector directions are then calculated based on the 2D sonic wind direction and wind speed. These will be vector averaged in the final step of summarizing the data to 5-minute averaged values. If the wind direction data were averaged without converting to u and v vectors, there would be inaccuracies when averaging 360 and 0 degrees. The formulas for these conversions are below:

$$\mathrm{u} = Wind\ Speed \ \times \ \sin(2\pi \ \times \frac{Wind\ Direction}{360})$$

$$\mathrm{v} = Wind\ Speed \ \times \ \cos(2\pi \ \times \frac{Wind\ Direction}{360})$$

The code then checks for latitude and longitude values. It is not required to have GPS configured on the sensor to use the Sentinel Shiny App. If latitude and longitude columns are not detected, the code will input 0 values for these columns and the mapping capabilities in the dashboard will be limited.

Once these initial processing steps are done, the code will use the dplyr[3] package to group the data in 5-minute averaged values for data reduction purposes (Table 2). The user cannot currently change this. The following columns are created in this aggregation for each 5-minute period, which will be used for analysis throughout the application. The method Detection Limit (MDL), Wind Speed (ws), and Wind Direction (wd) columns are appended after the 5-min aggregation. Canister columns will only be populated if the sensor is configured for canisters and canisters have been collected previously/at that time.

Table 2: 5-min data frame fields

| Column | Definition | Calculation |
|---|---|---|
| bc.pid.ppb | Background corrected data (ppb) | 5-min mean |
| pid.sd | The standard deviation of background corrected data (ppb) | 5-min standard deviation |
| rawPID.ppb | Raw concentration data (ppb) | 5-min mean |
| temp | Temperature data (C) | 5-min mean |
| RH | Relative humidity data (%) | 5-min mean |
| pressure | Pressure data (mbar) | 5-min mean |
| u.wind | Calculated u values (see above formula) | 5-min mean |
| v. wind | Calculated v values (see above formula) | 5-min mean |
| s1temp | Sensor temperature (arbitrary units) | 5-min mean |
| s1heat | Sensor heater output (0 = off, 255 = fully on) | 5-min mean |
| set | Sensor setpoint (arbitrary units) | 5-min mean |
| Bat.volt | Battery voltage (volts) | 5-min mean |
| chg.current | Charge current in milliamps | 5-min mean |
| opp.current | Operating current in milliamps | 5-min mean |
| trigportstat | Port status (NA indicates no collections) | List of unique values in 5-min period |
| trigactivestat | Trigger status (1 indicates active trigger) | List of unique values in 5-min period |
| trigactiveflag | Active Port (numbers indicate which port is currently active) | List of unique values in 5-min period |
| trigsampleflag | Event status (current sampling event or complete sampling event) | List of unique values in 5-min period |
| lat | Latitude (deg.) | Unique values in 5-min period |
| long | Longitude(deg.) | Unique values in 5-min period |
| QA | QA flags (See table on page 3) | List of unique values in 5-min period |
| MDL | Method Detection Limit | 3 * median daily standard deviation |
| wd | Wind Direction (deg) | Atan2 function |
| ws | Wind Speed (mph) | Sqrt function |
| SN | Serial number of Unit | Parsed from file name |

The application will display a simple summary table once all files are uploaded showing the unique serial numbers, latitude/longitude values, and a row count of how many entries are associated with that serial number. Any sensors with no latitude or longitude columns (not GPS-enabled) will show a 0 for latitude and longitude values if the user did not define these in the QA frame.

# Dashboard

Once the user has uploaded their data through the data upload page, the dashboard components are loaded based on those files. The dashboard (developed using the shinydashboard package[4]) consists of a signal map, SDI plots, and Time series graphs. In the toolbar at the top of the screen, there is drop down menu that will be automatically populated with available units to display, a switch for only displaying data with wind speeds > 1 m/s, and a report export button. More detail on each of the sections is given below.

## SIGNAL MAP

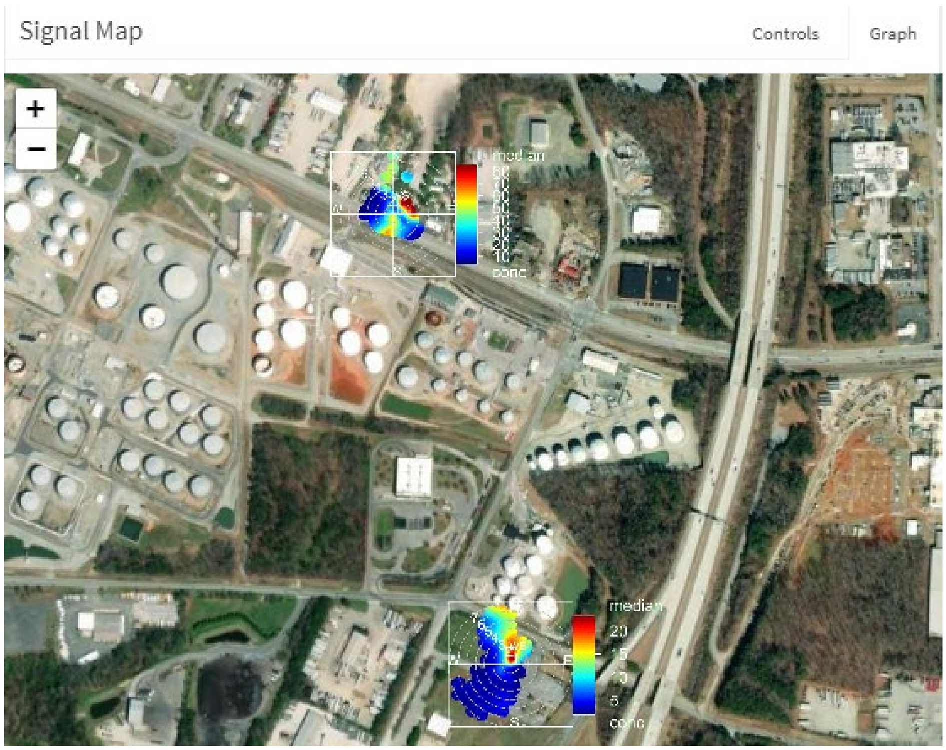


Figure 6: Signal Map with SDI plots

The signal map pane (Figure 6) consists of 2 tabs. The first is a controls tab, which allows the user to select some inputs, and the second is the graph display. The graph display can be panned and zoomed with the mouse as well as using the +/- controls in the upper left-hand corner. These plots are built using the polarmap function in the openair package[5]. This is run with a leaflet basemap[6], from Esri World Imagery. We refer to this kind of plot as a Source Direction Indicator (SDI) Plot. On the Controls tab, the user can use the slider to limit wind speed on the SDI plot to a certain range. The user can also choose the different statistic they would like to see applied to the graph, with the options of Median, weighted mean, maximum, or Nonparametric Wind Regression (NWR), which is done on datasets less than 200 datapoints. More information on these stats can be found in the SDI plot section. If the sensor did not include latitude and longitude data, and the user did not manually enter it in the QA flagging, the base map will not appear, and instead a grey screen will be displayed.

## SDI PLOTS

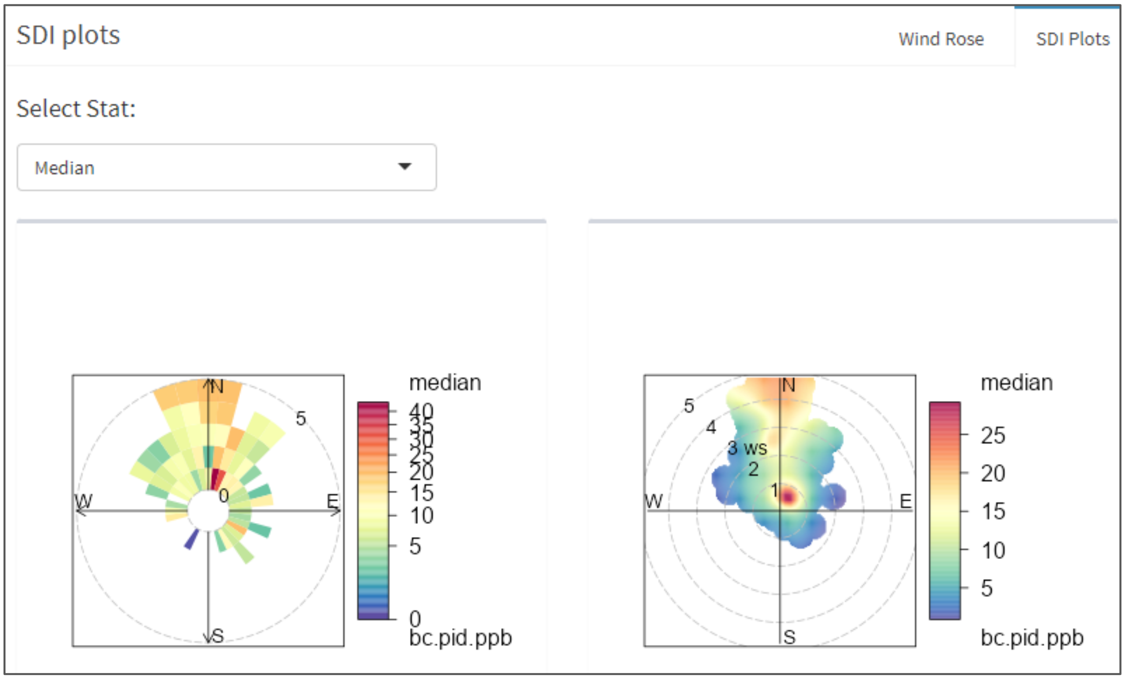


Figure 7: Polar Frequency and SDI plots

This panel contains two tabs, one containing a wind rose and the other containing two SDI plots. The SDI plot panel (Figure 7) contains a polar frequency plot on the left, and an interpolated SDI plot (polarplot) on the right[7,8]. The user has the option to select between several stats. These stats are each applied to the wind speed and wind direction bins shown in the polar frequency, and then interpolated with smoothing parameters in the SDI plot. The weighted mean stat is calculated as (concentration * frequency of occurrence). The wind rose plot (on the other tab) shows overall wind conditions as a frequency of counts by wind direction. Wind speed is binned by color in the wind rose plots.

## TIME SERIES GRAPHS

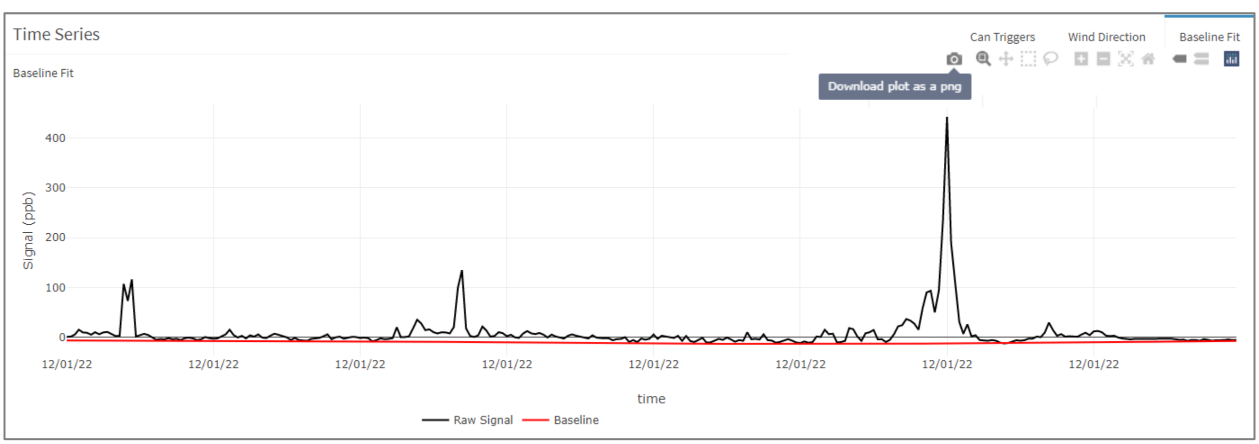


Figure 8: Baseline Fit time series plot

There are six possible tabs in the time series pane: Baseline Fit, Wind Direction, Relative Humidity, Temperature, and Can Triggers. All these plots are made using the plotly package[6], which allows for the user to hover over points and zoom/pan on the graph. If the user hovers the mouse in the upper right-hand corner of the graph, there are a suite of options available. One of these, the camera icon, can export a .png image of the graph (Figure 8). The user can also select an axis and drag it up or down to pan down only one axis at a time. The baseline fit tab shows the raw signal as a black trace and the baseline fit as a red trace (Figure 8). The wind direction tab shows the wind direction points as yellow dots and the signal as a black trace (Figure 9). This is useful for matching the periods of elevated signal with wind direction. The Canister trigger graph shows the signal as a black line trace and any canister acquisitions during that period color coded by port (Figure 10). Note that the y axis value of the canister trigger is not used – there are separated by distance in the event of multiple triggers in one 5-minute period (shown below). Also, no triggers will be present if the sensors are not configured to collect canister samples or if no triggers were recorded during this time frame. The Relative Humidity tab shows relative humidity in purple dots while the corrected signal is shown as a black trace (Figure 11). The Temperature tab shows temperature in blue dots while the corrected signal is shown as a black trace (Figure 12). These traces will not be present if the data does not contain these fields. Finally, the calibration is shown as an orange dot if the user entered it in the QA-flagged file and uploaded this file to the app (Figure 13).

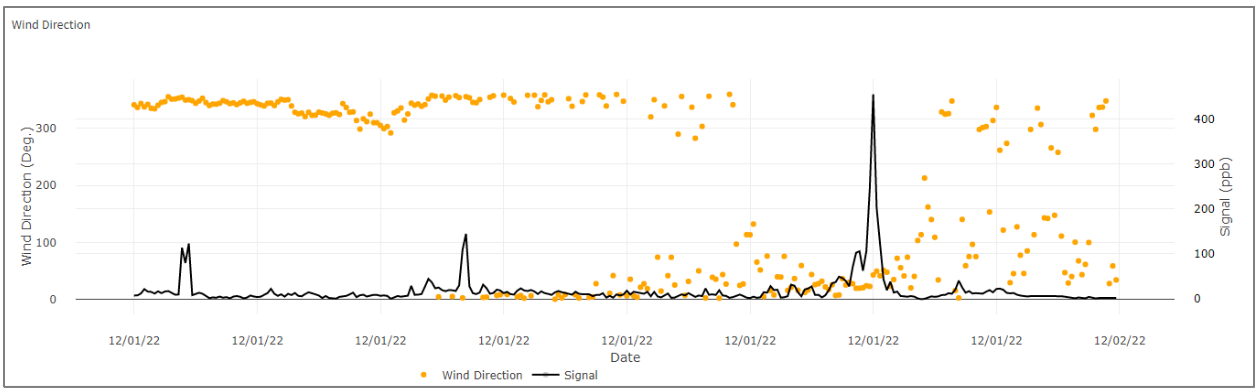


Figure 9: Wind Direction and Signal time series plot

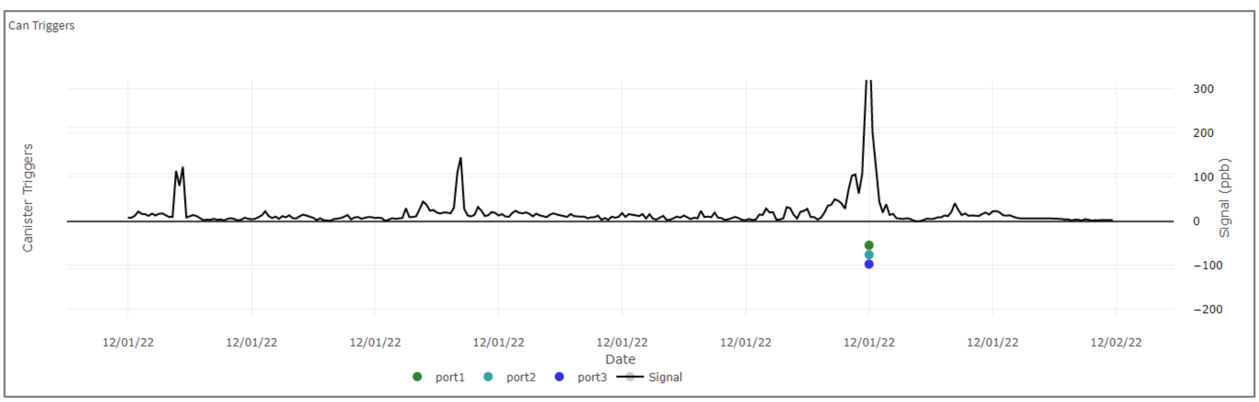


Figure 10: Canister triggers (3) and Signal time series plot

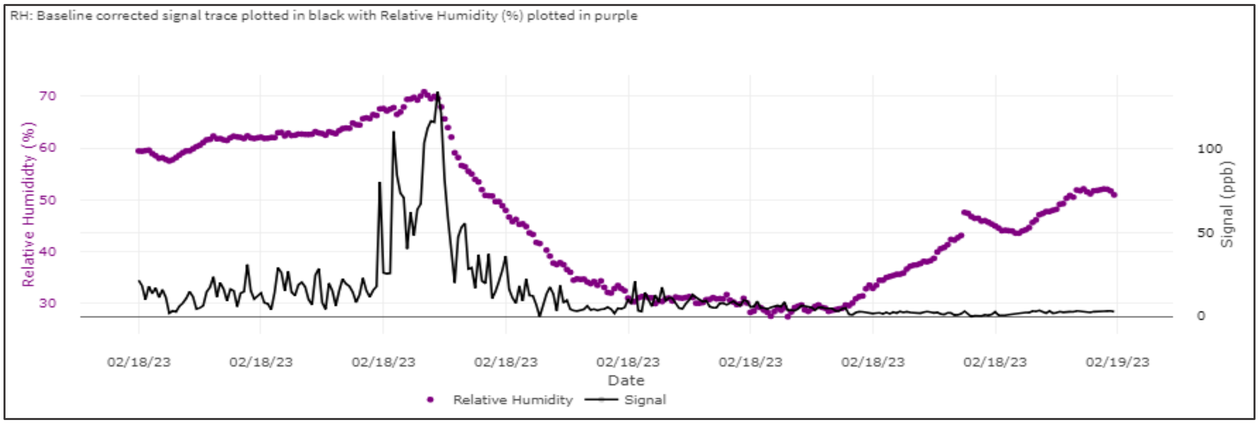


Figure 11: Relative Humidity and Signal time series plot

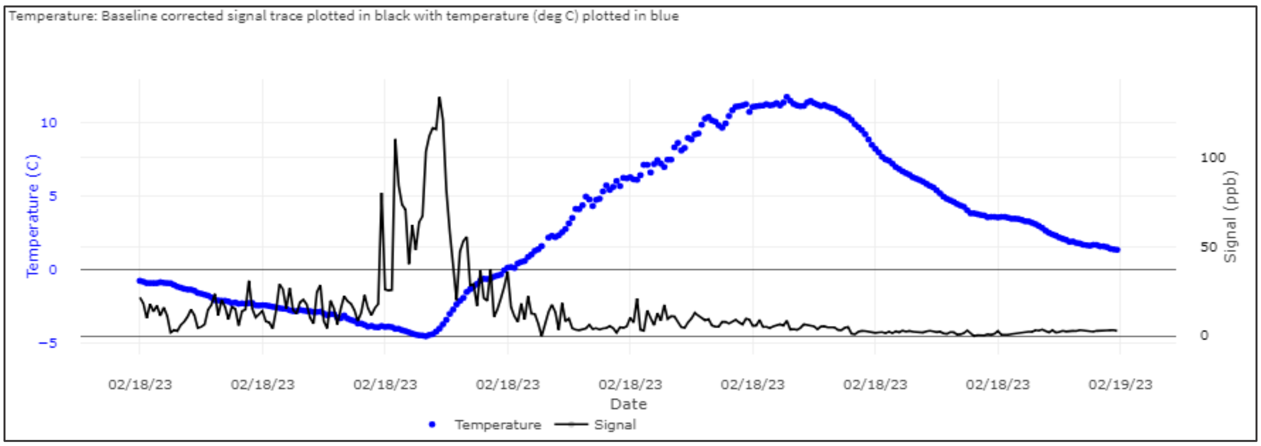


Figure 12: Temperature and Signal time series plot

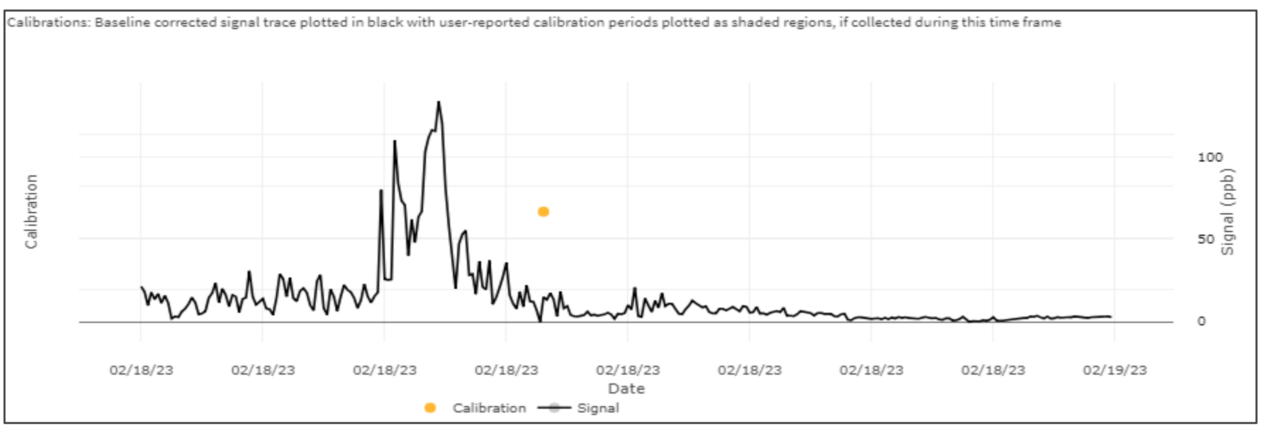


Figure 13: Calibration and Signal time series plot

## EXPORT TO REPORT

The top of the dashboard provides an option for a user to generate a report of the data displayed on the dashboard. This report will be output as an html file and will contain the SDI plot, the Wind Rose, and the time series graphs (no zoom options can be saved in that export).

# Data Table

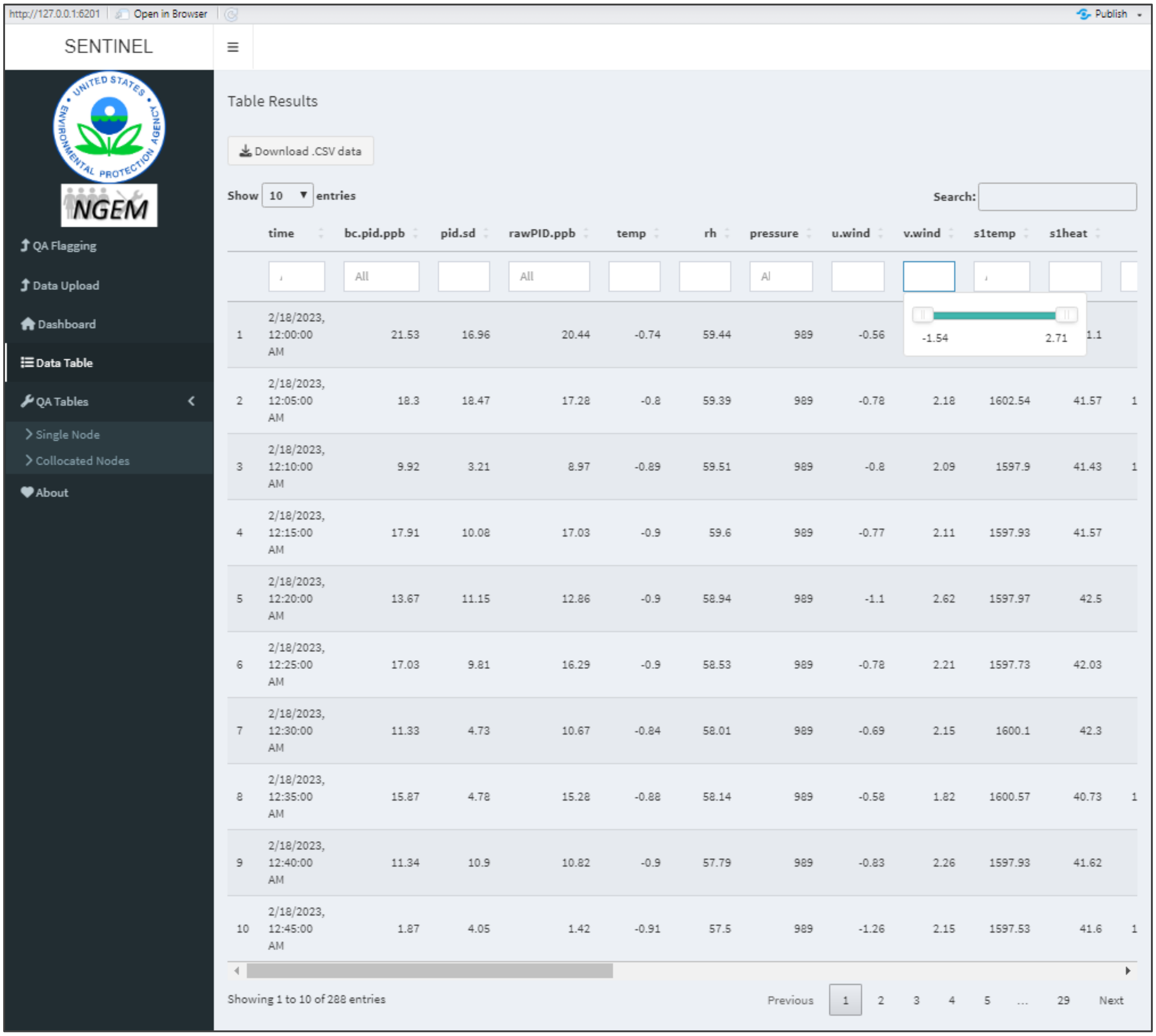

Figure 14: Interactive Data Table

The Data Table page allows the user to see the uploaded data in a 5-minute aggregated tabular form (Figure 14). Selecting the “Download .CSV data” button will export a .csv file of the compiled data and calculations. More entries can be displayed if the user changes the drop-down menu in the top left-hand corner. In the top right, there is a search bar where the user can search for values. The data is automatically organized by date, but the user can sort by other columns using the arrows at the top of each column. The user can search for values in the text entry boxes above each column and filter using the pop-up slider bars for each column. More rows can be seen by using the previous and next buttons in the bottom left. Scrolling to the right will show more columns. Definitions for these columns can be found on page 4 of this guide. Note that values in the canister trigger columns are encoded. Explanations of these values are shown in the Table 3.

Table 3: Canister Trigger Flag system

| Port Status (canister column 1) | |
|---|---|
| 15 | Port 1-4 installed; none collected |
| 31 | Port 1-4 installed; port 1 collected |
| 63 | Port 1-4 installed; port 1-2 collected |
| 127 | Port 1-4 installed; port 1-3 collected |
| 255 | Port 1-4 installed; port 1-4 collected |
| **Trigger Status (canister column 2)** | |
| 1 | Trigger occurring in this time frame |
| **Active Port (canister column 3)** | |
| 1 | Port 1 collecting |
| 2 | Port 2 collecting |
| 4 | Port 3 collecting |
| 8 | Port 4 collecting |
| **Event Status (canister column 4)** | |
| 1 | Port 1 started |
| 17 | Port 1 started; Port 1 complete |
| 19 | Port 2 started; Port 1 complete |
| 51 | Port 2 complete; Port 2 complete; Port 1 complete |
| 55 | Port 3 started; Port 2 complete; Port 1 complete |
| 119 | Port 3 complete; Port 2 complete; Port 1 complete |
| 127 | Port 4 started; Port 3 complete; Port 2 complete; Port 1 complete |
| 255 | Port 4 complete; Port 3 complete; Port 2 complete; Port 1 complete |

The user might come across some other, non-standard values if less than 4 canisters are installed in a non-standard order (i.e. canisters 2-3 are installed only). The user should consult the Sensit SPod Manual for more information on these flags.

# QA Tables

These options (single node and multi node) allow the user to create a QA table based on a time frame for a single sensor or two collocated sensors. These are useful for ensuring the sensor is running normally as well as determining how similar two sensors located side-by side are reporting. These tables are required to be collected during calibrations or cal-checks as defined in the SOP.

## SINGLE NODE

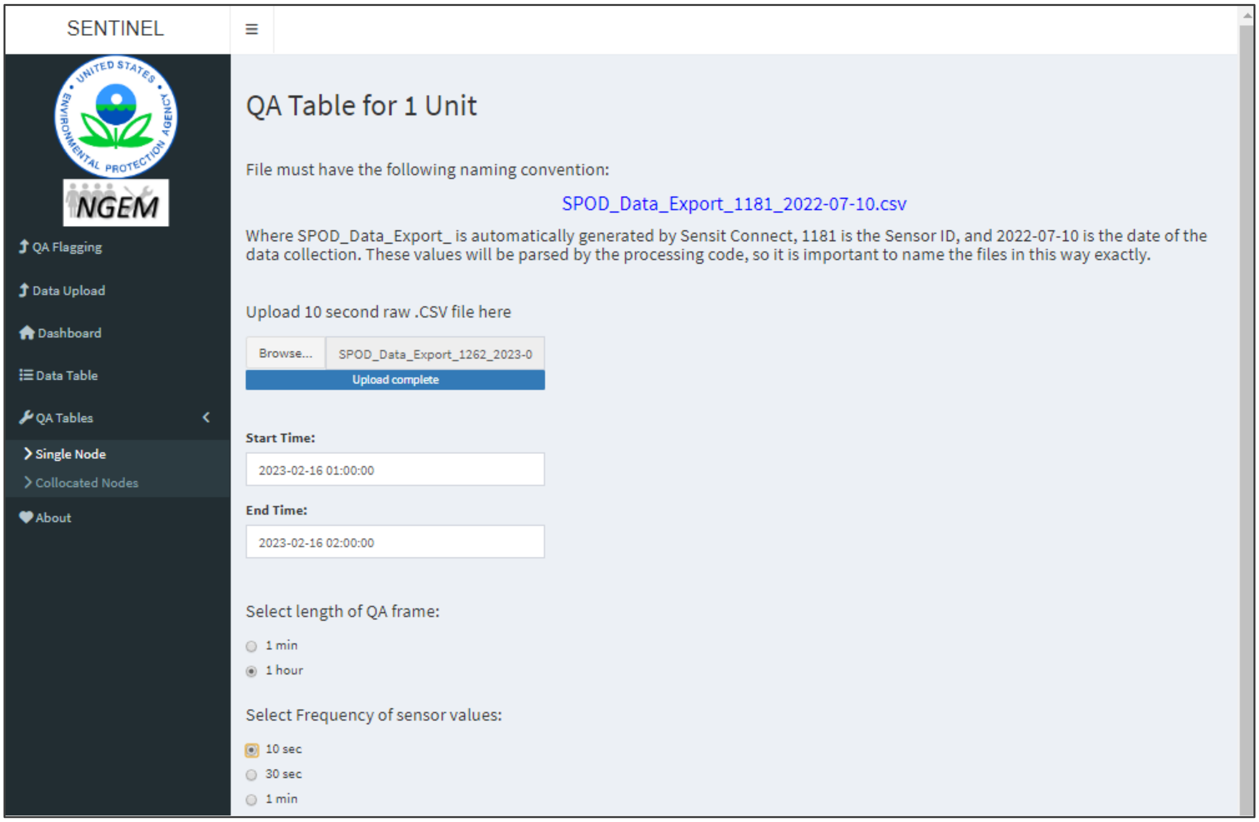


Figure 15: Single node QA Table screen

The Single node calibration page (Figure 15) requires the user to enter their Sensit connect raw file (named in the same way as general files are named) and the start and end time that they would like to QA. For a calibration on 10 second data, this will likely be only 1 minute. For other QA purposes, this could be 1 hour. The user should select the length of the QA frame (either 1 min or 1 hour) and the frequency of the data they uploaded (either 10 seconds, 30 seconds, or 1 hour), so the table shows the correct data completeness value. The table that is generated will show summary stats for data categories which can be used to judge if the sensor is calibrating or performing as expected during that time frame. The user can then select the “Generate report” button to get a report output with the QA table. This is an excellent option for record keeping. Values in the table will appear in red if they are outside of the recommended QA values. More information about these QA ranges can be found in the Sensit SPod SOP. See example output in Figure 16.

2023-02-16 01:00:00 to 2023-02-16 02:00:00

| Time periods with canister collections: FALSE | | | | | | |
|---|---|---|---|---|---|---|
| | **Mean** | **Median** | **StdDev** | **Min** | **Max** | **DataComp** |
| **Data Quality QA:** | | | | | | |
| Raw PID (ppb) | 32.9 | 29.2 | 15.1 | 11.7 | 68.2 | 100 |
| BC PID (ppb) | 24.1 | 20.4 | 15.2 | 2.6 | 59.6 | 100 |
| Raw PID (mV) | 107.5 | 104.8 | 11.1 | 91.9 | 133.5 | 100 |
| Temp (Deg C) | 10.7 | 10.7 | 0.2 | 10.4 | 11.2 | 100 |
| RH (percent) | 77.1 | 77.3 | 1.1 | 74.9 | 78.7 | 100 |
| Pressure (mBar) | 985 | 985 | 0 | 985 | 985 | 100 |
| WS (mph) | 0.3 | 0.3 | 0.2 | 0 | 0.7 | 100 |
| WD (deg) | 192.7 | 219.8 | 124.2 | 0 | 352.5 | 100 |
| **Operational QA:** | | | | | | |
| S1 temp (arb) | 2054.9 | 2054 | 7.3 | 2037 | 2075 | 100 |
| S1 Heat (0-255) | 36.3 | 36 | 1.1 | 34 | 39 | 100 |
| S1 Set (arb) | 2054.9 | 2054 | 6.7 | 2042 | 2074 | 100 |
| Bat volt (V) | 14 | 14 | 0.1 | 13.8 | 14.2 | 100 |
| Charge Current (mA) | 108 | 2.6 | 188.5 | 0 | 689.9 | 100 |
| Operate Current (mA) | 101.4 | 100.3 | 5.9 | 91.7 | 131.6 | 100 |

* Values determined by a location at 892 ft above sea level

Figure 16: Example output of a single node QA table

## MULTI NODE

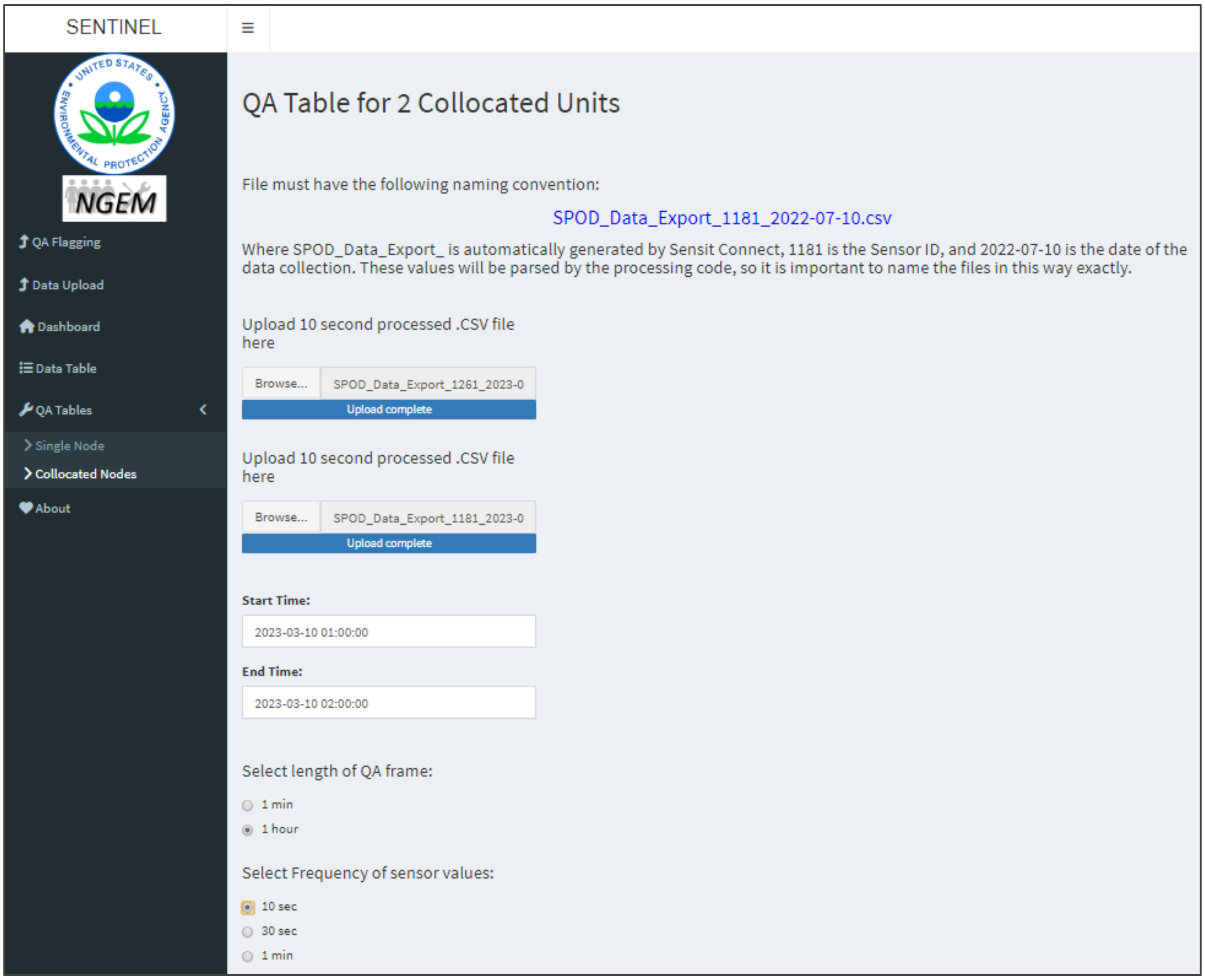


Figure 17: Multi Node QA Table screen

The multi node page operates similarly to the single node page except the user is prompted to enter a secondary node to be compared to the first node (Figure 17). For this comparison to be effective, the two selected nodes should be collocated, so sensor agreement can be evaluated. The user once again enters the start and end time and should select the length of the QA frame (either 1 min or 1 hour) and the frequency of the data they uploaded (either 10 seconds, 30 seconds, or 1 hour), so the table shows the correct data completeness value. This table is the simple difference between the two nodes (the first node – the second node). The user has the option to export this QA table record keeping (Figure 18).

2023-03-10 01:00:00 to 2023-03-10 02:00:00

| | Mean | Median | StdDev | Min | Max | DataComp |
|---|---|---|---|---|---|---|
| **Data Quality QA:** | | | | | | |
| Raw PID (ppb) | -13.2 | -13.1 | -1.4 | -10.6 | -22.2 | 0.3 |
| BC PID (ppb) | -3.1 | -2.9 | -1.4 | -0.4 | -12.3 | 0.3 |
| Raw PID (mV) | -3.1 | -3.1 | -0.6 | -2.0 | -8.3 | 0.3 |
| Temp (Deg C) | 0.2 | 0.2 | 0.0 | 0.1 | 0.1 | 0.3 |
| RH (percent) | -1.0 | -1.0 | 0.1 | -1.1 | -0.7 | 0.3 |
| Pressure (mBar) | -0.9 | -1.0 | -0.3 | 0.0 | -1.0 | 0.3 |
| WS (mph) | -0.1 | -0.2 | 0.1 | -0.1 | 0.1 | 0.3 |
| WD (deg) | 8.9 | 9.7 | 5.8 | -34.5 | -49.3 | 0.3 |
| **Operational QA:** | | | | | | |
| S1 temp (arb) | 7.1 | 9.0 | -1.1 | 4.0 | 7.0 | 0.3 |
| S1 Heat (0-255) | 10.1 | 10.0 | 0.0 | 11.0 | 10.0 | 0.3 |
| S1 Set (arb) | 7.1 | 8.0 | -1.4 | 4.0 | 4.0 | 0.3 |
| Bat volt (V) | 0.1 | 0.1 | 0.0 | 0.0 | 0.1 | 0.3 |
| Charge Current (mA) | 52.0 | 19.2 | 32.7 | 0.1 | 95.7 | 0.3 |
| Operate Current (mA) | 64.6 | 65.6 | -0.9 | 61.2 | 52.9 | 0.3 |

* Values determined by a location at 892 ft above sea level

Figure 18: Example output of a Multi node QA table

## About Page

The about page contains the version number of the code, and the contact information if any user finds questions or bugs (macdonald.megan@epa.gov). The link to the User Guide is also included on this page, as well as an overview of the app and acknowledgements to contributors to this project.

# Accessing Code

Users can access code at the Github Repository page (https://github.com/USEPA/SENTINEL). The Repository contains several files that are used in app deployment and posting. The app.R file is the code to build and deploy the user interface and the server portions of the app. The Sentinel_Report.Rmd, multi_node_QA_Table.Rmd, and single_node_Qa_table.Rmd files are templates for building out the reporting forms within the app. The getBaseline.R and screeningFunctions.R files are custom function code that define some of the functions the app utilizes that aren't defined in the common packages. The www folder contains logos that are used in the app. The user guide, app info sheet, and Sensit (J-AMCD-SFSB-SOP-4380-2) are contained in this folder as pdf documents.

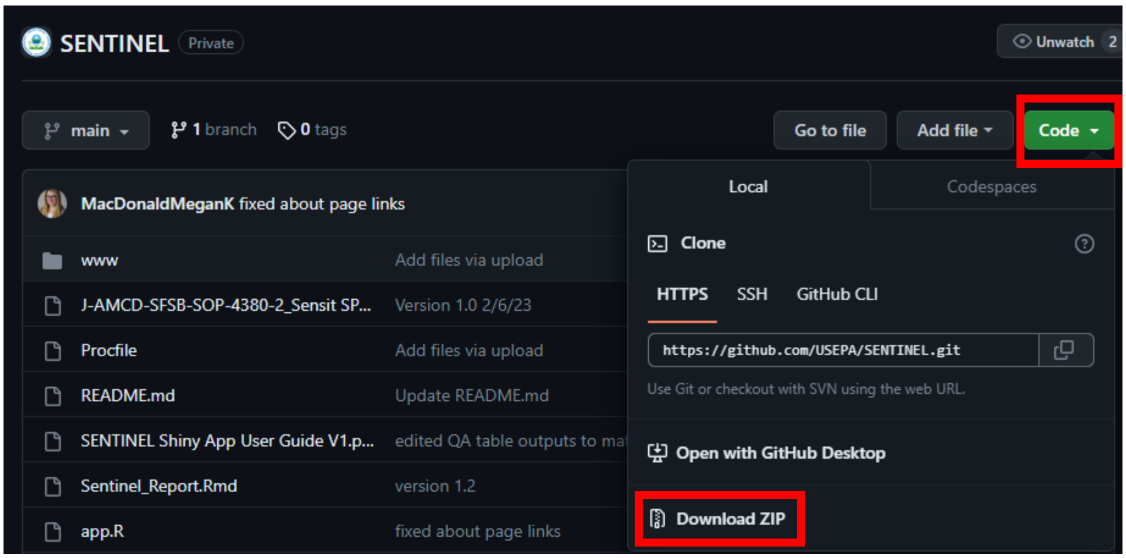


Figure 19: Github Repository for SENTINEL

To run code from this repository, the user will need to navigate to the page and click the green code button on the top left of the page and then select the Download ZIP option. The user should save it all in a folder and open the app in R Studio and set this folder as the working directory so all files can be traced. The app will not run if it cannot access the custom function files.

In the future, it is possible that the app will be accessible via a link. Currently, the app is only available via this open-source code.

# References


1. R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
2. Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert and Barbara Borges (2022). shiny: Web Application Framework for R. R package version 1.7.2. https://CRAN.R-project.org/package=shiny
3. Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2022). dplyr: A Grammar of Data Manipulation. R package version 1.0.9. https://CRAN.R-project.org/package=dplyr
4. Winston Chang and Barbara Borges Ribeiro (2021). shinydashboard: Create Dashboards with 'Shiny'. R package version 0.7.2. https://CRAN.R-project.org/package=shinydashboard
5. Carslaw, D. C. and K. Ropkins (2012). openair --- an R package for air quality data analysis. Environmental Modelling & Software. Volume 27-28, 52-61.
6. Joe Cheng, Bhaskar, Karambelkar and Yihui Xie (2022). leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.1.1. https://CRAN.R-project.org/package=leaflet
7. Ropkins, K., & Carslaw, D. C. (2012). openair-Data Analysis Tools for the Air Quality Community. *R Journal*, *4*(1). https://journal.r-project.org/archive/2012/RJ-2012-003/RJ-2012-003.pdf
8. Carslaw, D. C., & Ropkins, K. (2012). Openair—an R package for air quality data analysis. *Environmental Modelling & Software*, *27*, 52-61.
9. Carson Sievert (2020). Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC Florida.


# Resources

More information on R Shiny Applications:

- Mastering Shiny Bookdown Site

More information on SPod deployments:

- Rubbertown Next Generation Emissions Measurement Demonstration Project (Journal article)
- Demonstration of VOC Fenceline Sensors and Canister Grab Sampling near Chemical Facilities in Louisville, Kentucky (Journal article)
- Sensor Pod (SPod): An Approach for VOC Fenceline Monitoring and Data Analysis (EPA Tools and Resources Webinar)
- Next Generation Emission Measurements (NGEM) Advancements (EPA ORISE Meets the World Seminar)
- Fenceline and Community Sensor Applications and Comparisons (Air Sensors International Conference video)