# SENTINEL R Shiny Application User Guide

Version: 1.1 (June 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 low-cost fenceline sensor data. These sensors can generate a large amount of data, which can be overwhelming for users to process manually. SENTINEL provides users with a standard framework for processing data and generating visualizations that can lead to insights. This application is also useful for generating Quality Assurance (QA) tables for individual or co-located sensors.

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.

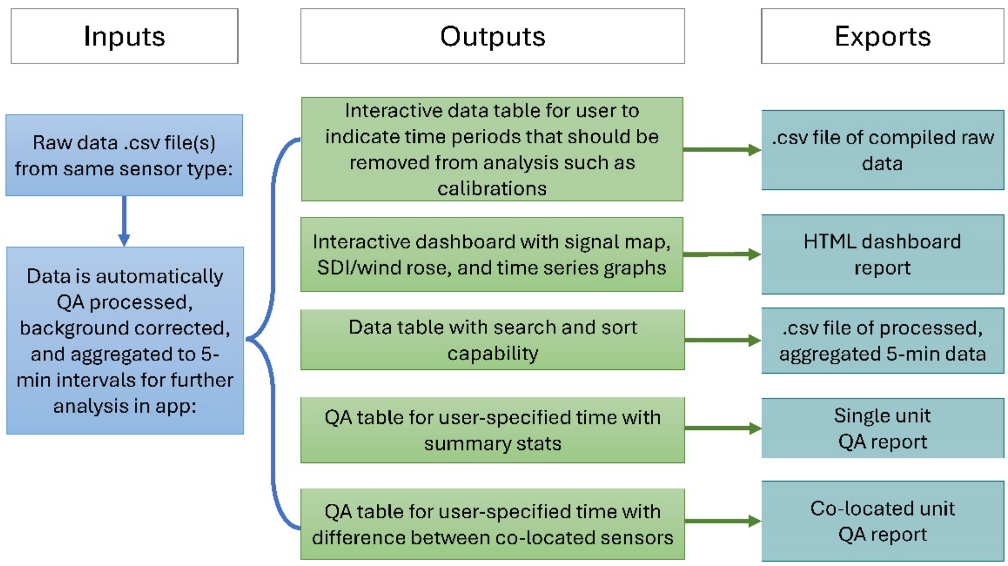



Figure 1: Application Map

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.

# Accessing Code & Deploying the App

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. This user guide is contained in this folder as a pdf document.

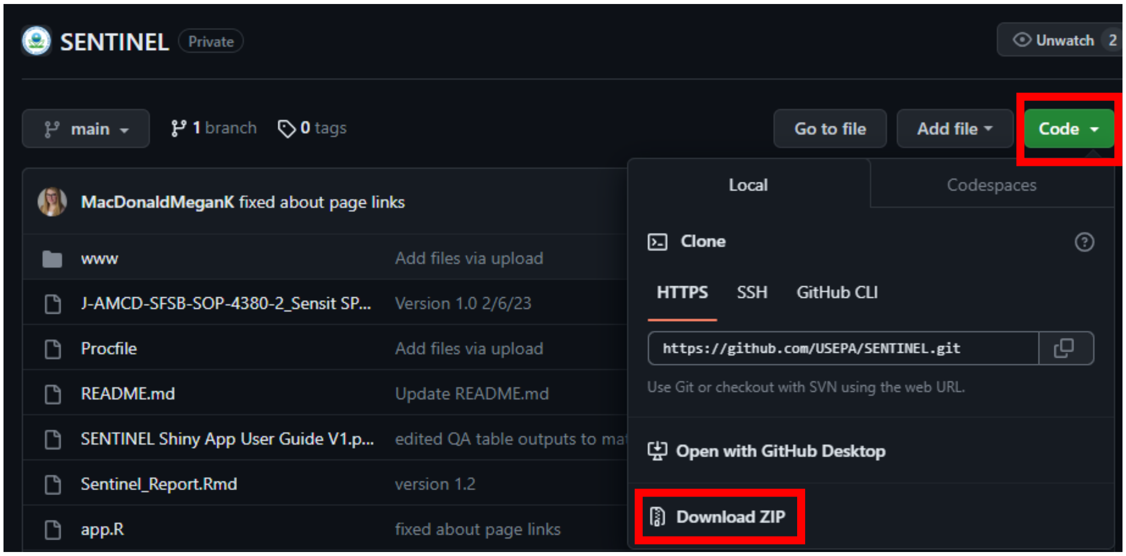


Figure 2: Github Repository for SENTINEL

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 all of the downloads in a folder on their computer.

Users should have [RStudio] (Posit) installed on their machine. Inside of the RStudio environment, the user should navigate to Session > Set Working Directory > Choose Directory and then select the folder where the app files are stored. The folder should have all of the components from the Github zip download.

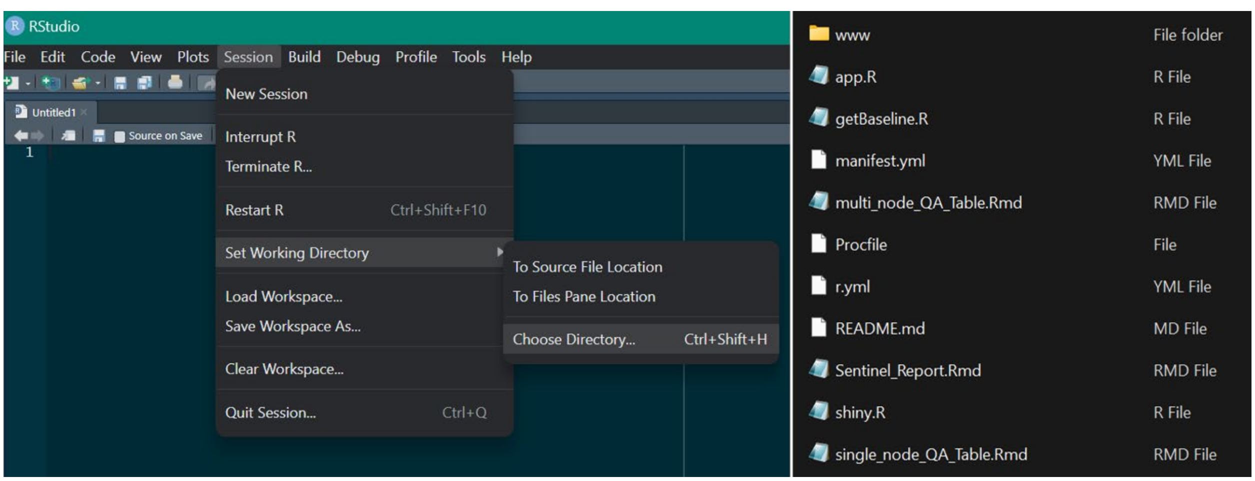


Figure 3: RStudio Environment and Working Directory selection.

Once the correct working directory is selected, the user should open the app.R file in R studio. At the top right corner of the editor, there should be a small green arrow labeled “Run App”. After pressing this button, the code should deploy and the app will open in a new R Studio window.

# Uploading Data & QA Processing

Data can be entered into the "Upload Data" tab from any lower cost fenceline sensor. This application was originally designed for use with a specific Fenceline VOC sensor (SPod) but has since been expanded to be sensor agnostic. The only requirements for files being input into the app are that the format must be.csv (not excel or .txt files) and there must be a timestamp column and sensor ID column. Columns can have any names, as these will be changed in the app. **The user must click through the 4 tabs at the top of the input box to ensure column names get accepted by the app before moving on to the data check page** (See Figure 5)**.**

For files downloaded from Sensit Connect, do not change the file name that is automatically generated by the site: ("SPOD_Data_Export_1181), where "SPOD_Data_Export_" is automatically generated and "1181" is the Sensor ID. The unchanged files downloaded from Sensit connect can be uploaded directly into SENTINEL. This naming convention is how the app is determining the ID column, which will be called *spod_check* in the app.

## For Sensit users: Downloading Data with Sensit Connect

Data should be downloaded from the Sensit Connect Website ([sensitconnect.net](sensitconnect.net)) 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.

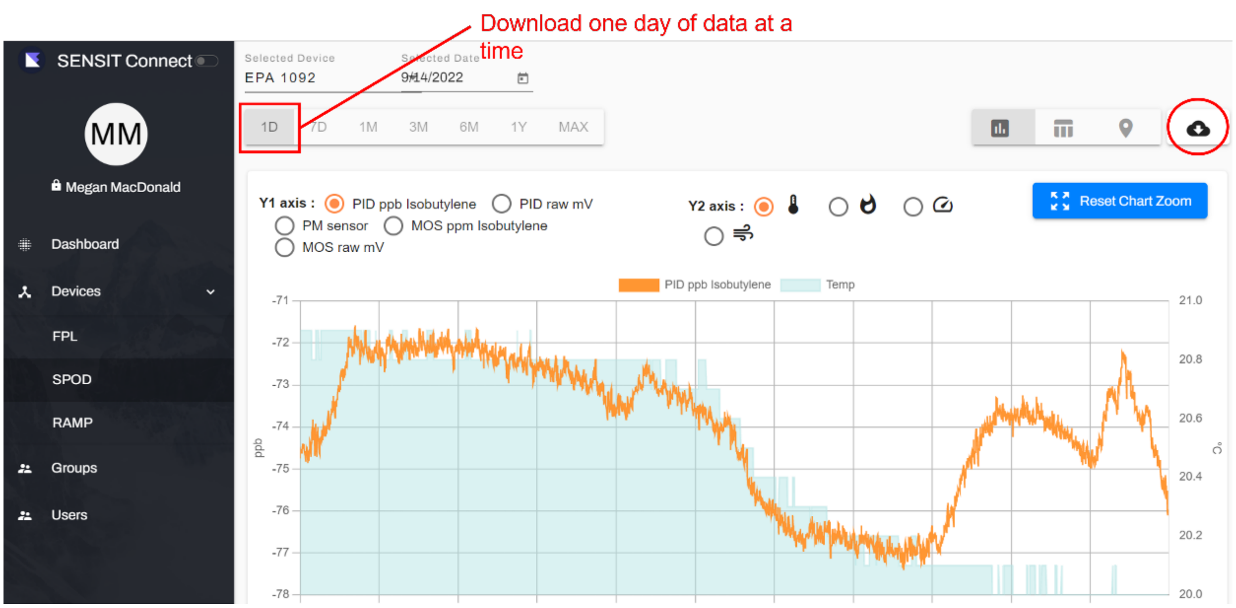
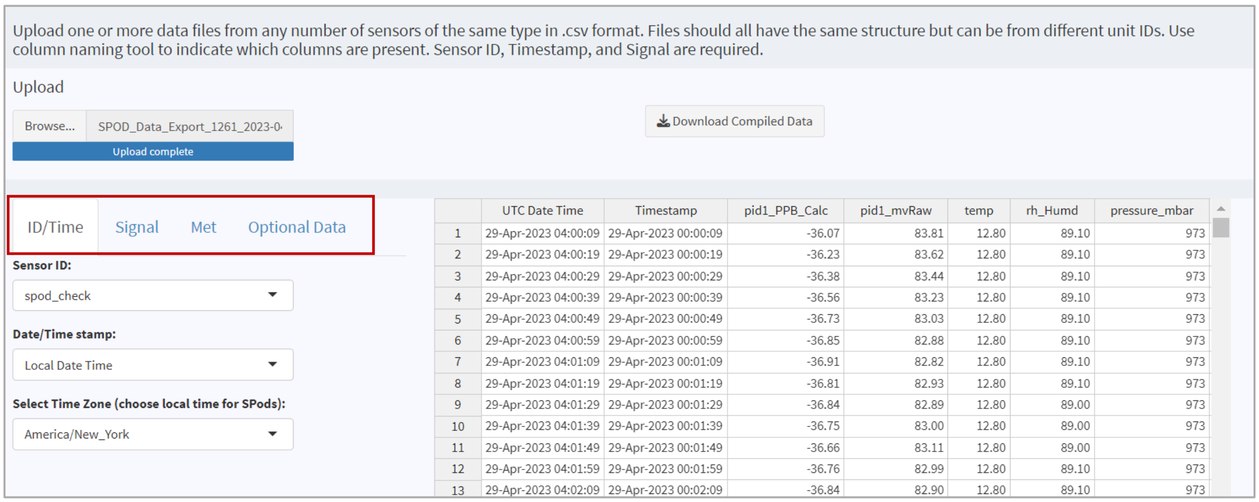



Figure 4: Sensit Connect Site

Once the user has selected their files and entered them through this page of the app, the columns can be changed to indicate which column the app should expect for certain analysis variables. The options in the drop-down menus are generated from the entered file(s)' headers. The Sensor ID column will default to *spod check* which is a function in the app that parses the Sensor name from Sensit Connect file names, since these files do not have a sensor unit column. In order for the table to appear, the date/time stamp column must be identified. Otherwise, a red warning will appear asking for this column to be identified.

The app will prompt the user to select the time stamp format from a drop-down list. The separators must match (i.e.; - or /) and the components of the time stamp must be in the proper order (Y = four digit year, y = two digit year, m = month (numeric, such as 12), b = month (character, such as dec), d = day, H = hour, M = minute, S = second). If a red error text appears anywhere in the app saying that timestamp is needed to generate a data frame, it is possible a data file you have entered has a timestamp that is unsupported. Send an email to macdonald.megan@epa.gov with a sample data file to get your file's time format added in future updates.




| | UTC Date Time | Timestamp | pid1_PPB_Calc | pid1_mvRaw | temp | rh_Humd | pressure_mbar |
|---|---|---|---|---|---|---|---|
| 1 | 29-Apr-2023 04:00:09 | 29-Apr-2023 00:00:09 | -36.07 | 83.81 | 12.80 | 89.10 | 973 |
| 2 | 29-Apr-2023 04:00:19 | 29-Apr-2023 00:00:19 | -36.23 | 83.62 | 12.80 | 89.10 | 973 |
| 3 | 29-Apr-2023 04:00:29 | 29-Apr-2023 00:00:29 | -36.38 | 83.44 | 12.80 | 89.10 | 973 |
| 4 | 29-Apr-2023 04:00:39 | 29-Apr-2023 00:00:39 | -36.56 | 83.23 | 12.80 | 89.10 | 973 |
| 5 | 29-Apr-2023 04:00:49 | 29-Apr-2023 00:00:49 | -36.73 | 83.03 | 12.80 | 89.10 | 973 |
| 6 | 29-Apr-2023 04:00:59 | 29-Apr-2023 00:00:59 | -36.85 | 82.88 | 12.80 | 89.10 | 973 |
| 7 | 29-Apr-2023 04:01:09 | 29-Apr-2023 00:01:09 | -36.91 | 82.82 | 12.80 | 89.10 | 973 |
| 8 | 29-Apr-2023 04:01:19 | 29-Apr-2023 00:01:19 | -36.81 | 82.93 | 12.80 | 89.10 | 973 |
| 9 | 29-Apr-2023 04:01:29 | 29-Apr-2023 00:01:29 | -36.84 | 82.89 | 12.80 | 89.00 | 973 |
| 10 | 29-Apr-2023 04:01:39 | 29-Apr-2023 00:01:39 | -36.75 | 83.00 | 12.80 | 89.00 | 973 |
| 11 | 29-Apr-2023 04:01:49 | 29-Apr-2023 00:01:49 | -36.66 | 83.11 | 12.80 | 89.00 | 973 |
| 12 | 29-Apr-2023 04:01:59 | 29-Apr-2023 00:01:59 | -36.76 | 82.99 | 12.80 | 89.10 | 973 |
| 13 | 29-Apr-2023 04:02:09 | 29-Apr-2023 00:02:09 | -36.84 | 82.90 | 12.80 | 89.10 | 973 |

Figure 5: Data Upload tab of app. User must click through red boxes to make sure column names are accepted!

For Sensit users, the Sensit SPod variables will automatically map to these columns if the file is downloaded from Sensit Connect and the file name is unchanged, and users just need to click through the tabs to see the columns re-name. Additionally for Sensit users, the Timestamp column is based on the local time column, so set the time zone to the local time of the sensors' location. Check that signal units are in ppb!

As users change values, they can see the table column names updating. It is important to set correct units for windspeed, as a multiplier will be added to any data in mph. (a division by 2.237 to put all wind speed values in m/s for analysis). The unit choices for temperature and signal do not impact the data and only serve to be edited for graph titles and axis labels.

The active canister column only pertains to Sensit Spods that have an associated canister grab sampling system.

Ultimately, the app will run with only Sensor ID, Timestamp, and Signal_1. To see the full extent of the app's visualizations, the user will also need to identify wind direction, wind speed, wind speed units, lat/long, temperature, and RH.

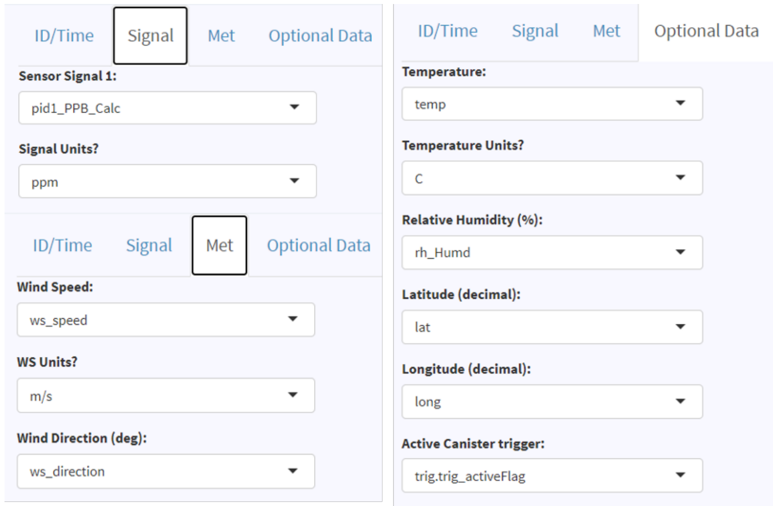



Figure 6: All column options for analysis.

Once the files are uploaded and columns are identified, the user can use this table to indicate any time periods during which there is a reason to discount that data. The user can scroll all the way over to the last column in this table on the right, *QA*, which is appended by the app. The user can select the drop down box for time periods they want to discount for the various options listed.

| Lat | Long | PSI_1 | PSI_2 | PSI_3 | PSI_4 | V26 | Sensor_ID | QA |
|---|---|---|---|---|---|---|---|---|
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | Calibration |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | Calibration |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | Calibration |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | Calibration |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | Calibration |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | None |
| 0.00 | 0.00 | -12.17 | 0 | -12.09 | -12.16 | | 1261 | None |

Figure 7: QA column with calibration period added.

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.

- **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.
- **WD 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.
- **WD error** applies to time periods where the anemometer was misaligned or malfunctioning.
- **Other** can be user defined in your own QA planning.

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. This stage is also a good time to add Lat and Long through this interactive table if it isn't included in the files.

The user has the option to download this QA-checked data as a .csv file. These QA-checked files can be read into the data upload page in combination with raw files for processing and use in the app in the future, to save calibration and other QA entries.

When the user navigates away to the Data Check page, this data is processed and aggregated to 5 minutes.

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 |
|---|---|
| None | auto |
| Calibration | user-defined |
| Interference | user-defined |
| Maintenance | user-defined |
| Malfunction | user-defined |
| Other | user-defined |
| WD Interference | user-defined |
| WD Error | user-defined |
| WS Repeat (> 30 WS values consecutively) | auto |
| WD Repeat (> 30 WD values consecutively) | auto |
| Sig Repeat (> 30 Signal consecutively) | auto |
| WS Off scale (> 12 m/s) | auto |
| WD Off scale (< 0 or > 360 degrees) | auto |
| Missing Signal (Signal value = NA) | auto |

The script will then conduct baseline correction 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 wind direction and wind speed. These will be vector averaged in the final step of summarizing the data to 5-minute averaged values. Because of this, it is important to not input data at a frequency larger than 5 minutes. 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:

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

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

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 NA 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). Based on the data provided, a data frame will be generated with the standardized column names. This data table can be output to a .csv file in the "Data Table" page or viewed interactively in the app there.

# Data Check

After data is uploaded, the user can navigate to the "Data Check" page. This page provides the user with a summary table of the data that was entered and processed. The user can see separate Sensor IDs, time ranges, latitude/longitude (if applicable), and any unique QA flags that were appended.

| Sensor_ID | Start_Time | End_Time | Lat | Long | Count | QA | Canister |
|---|---|---|---|---|---|---|---|
| 1181 | 2023-04-28 20:00:00 | 2023-04-29 19:55:00 | 0 | 0 | 288 | None | NA |
| 1261 | 2023-04-28 20:00:00 | 2023-04-29 19:55:00 | 0 | 0 | 288 | None | NA |

Figure 8: Data check summary table. Note that Lat/Long are not required to run the application.

# 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

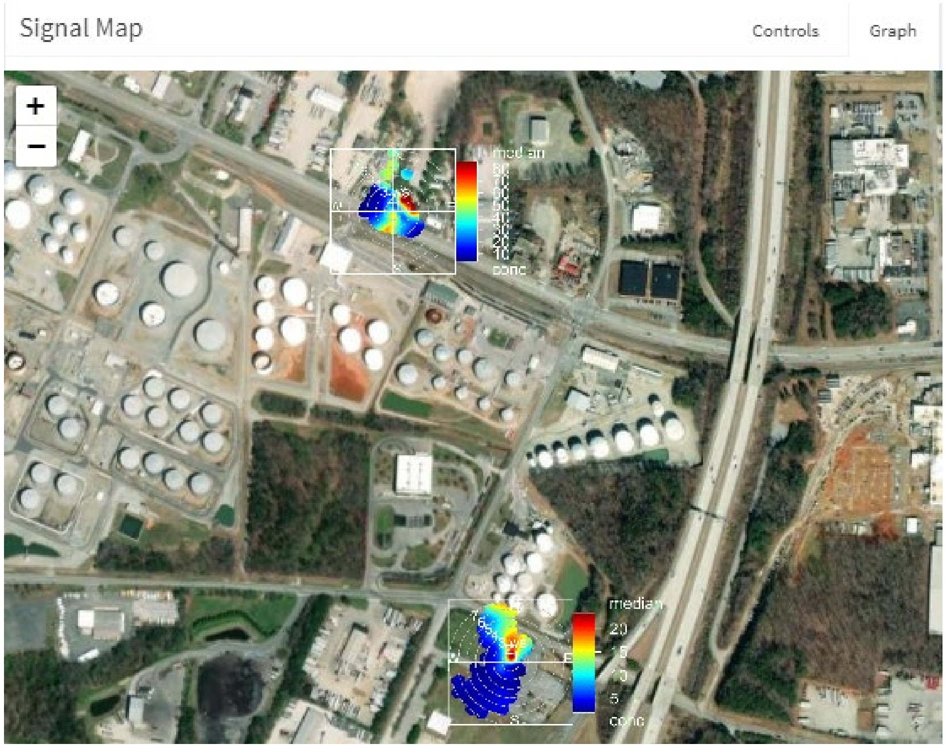



Figure 9: Signal Map with SDI plots

The signal map pane (Figure 9) 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

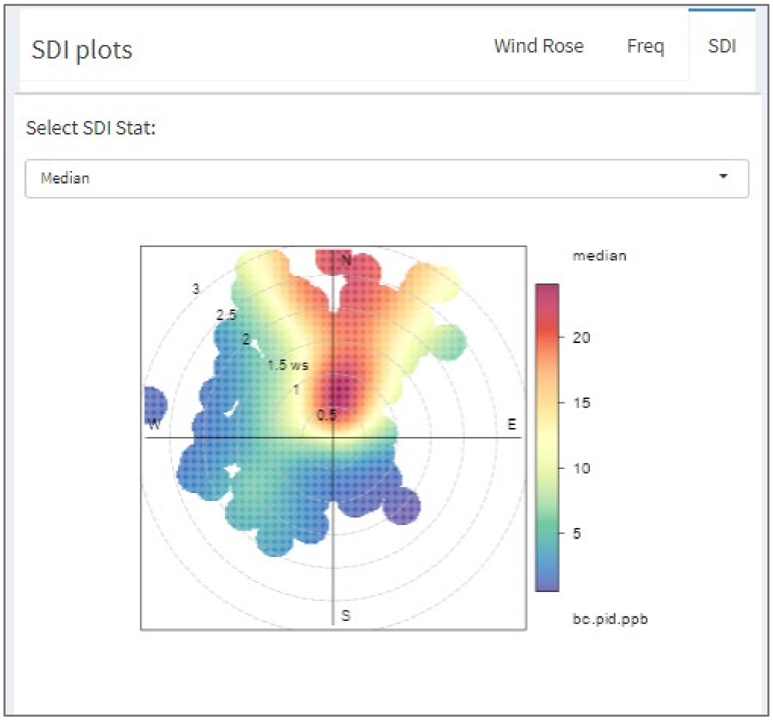



Figure 10: SDI plot tab in Dashboard page

This panel contains three tabs, one containing a wind rose and the other two containing forms of SDI plots (Figure 10)[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

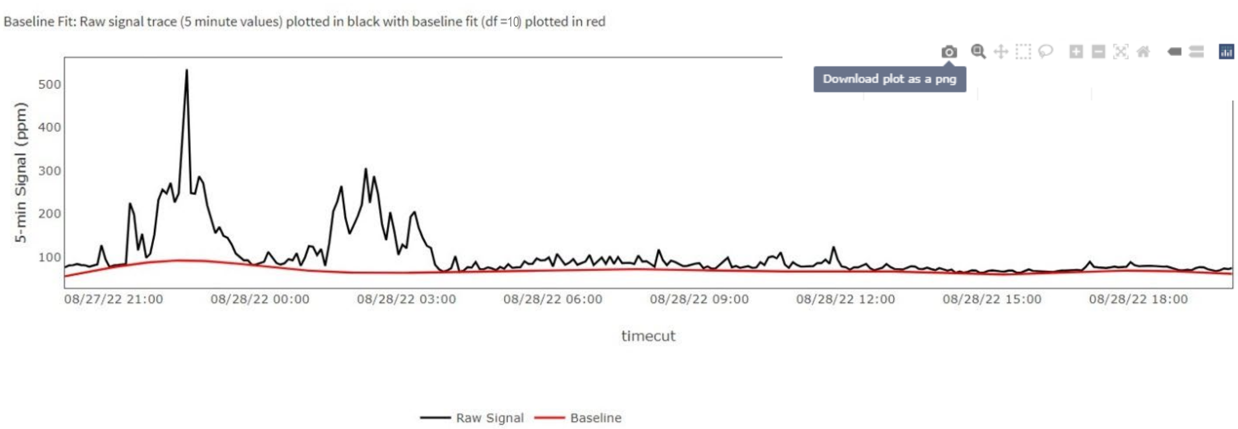



Figure 11: Baseline Fit time series plot

There are several possible tabs in the time series pane: Baseline Fit, Wind Direction, Wind Speed, Relative Humidity, Temperature, Calibrations, 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 11). 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 11). The *Wind Direction* tab shows the wind direction points as green points and the signal as a black trace (Figure 12). This is useful for matching the periods of elevated signal with wind direction. The *Wind Speed* tab shows wind speed in m/s as grey points (Figure 13). The *Temperature* tab shows temperature in blue points with units indicated by the original data upload option (Figure 14). The *RH* tab shows relative humidity in purple points as a percentage (Figure 15). The *Canister Triggers* graph is specific to SPod data downloaded through Sensit Connect and shows the signal as a black line trace and any canister acquisitions during that period color coded by port (Figure 16). 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. Finally, the *Calibration* tab shows an orange point on 5-minute time periods where the user indicated a calibration appeared in the QA column on the Data Upload page (Figure 17). These traces will not be present if the data does not contain these fields.

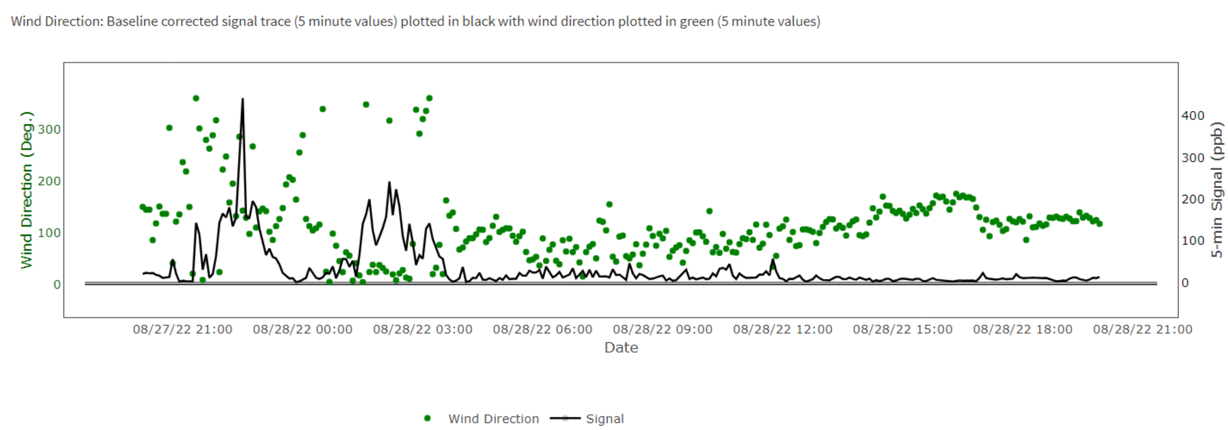



Figure 12: Wind Direction and Signal time series plot

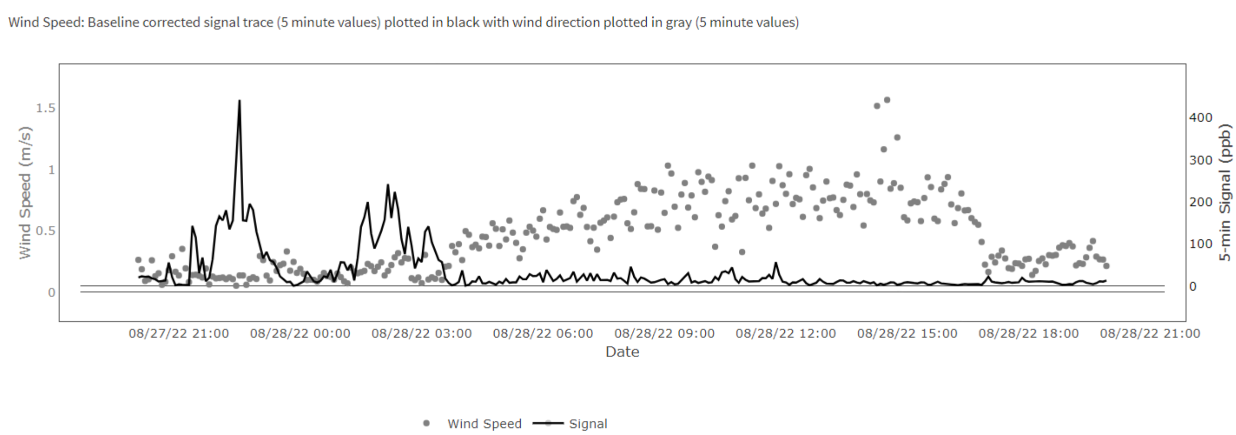



Figure 13: Wind Speed and Signal time series plot

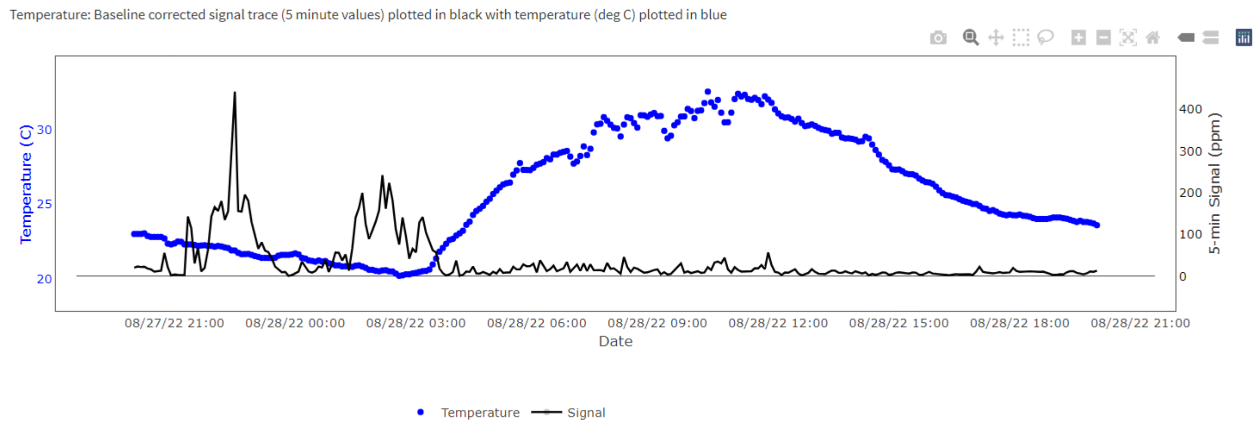



Figure 14: Temperature and Signal time series plot

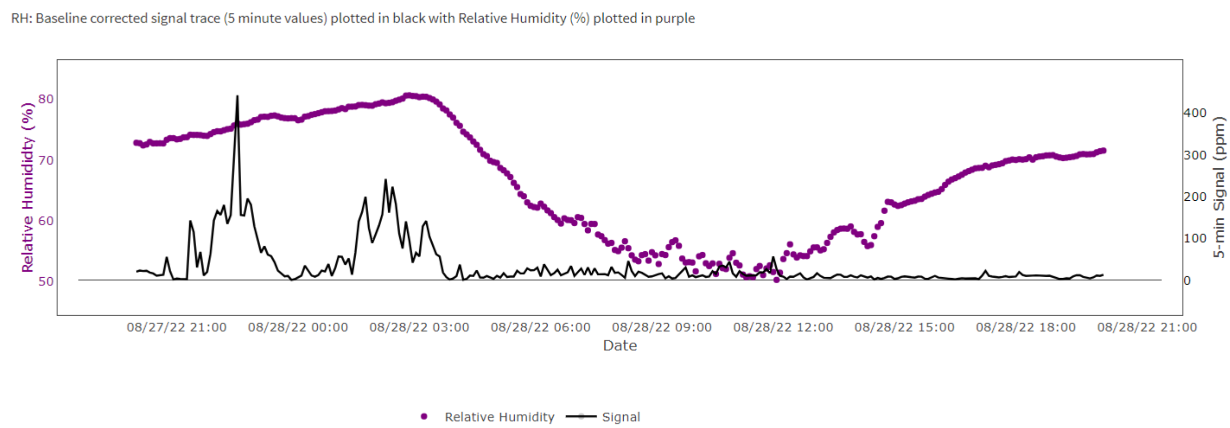



Figure 15: RH and Signal time series plot

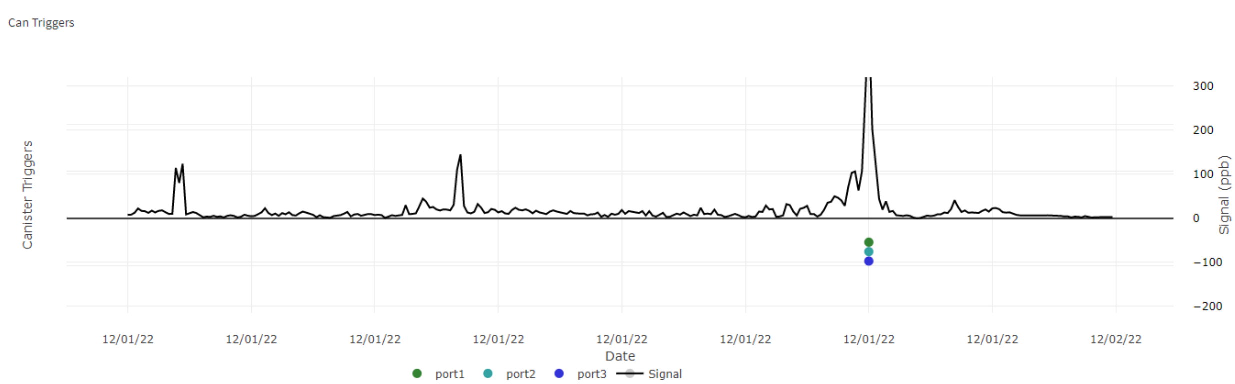



Figure 16: Canister triggers (3) and Signal time series plot

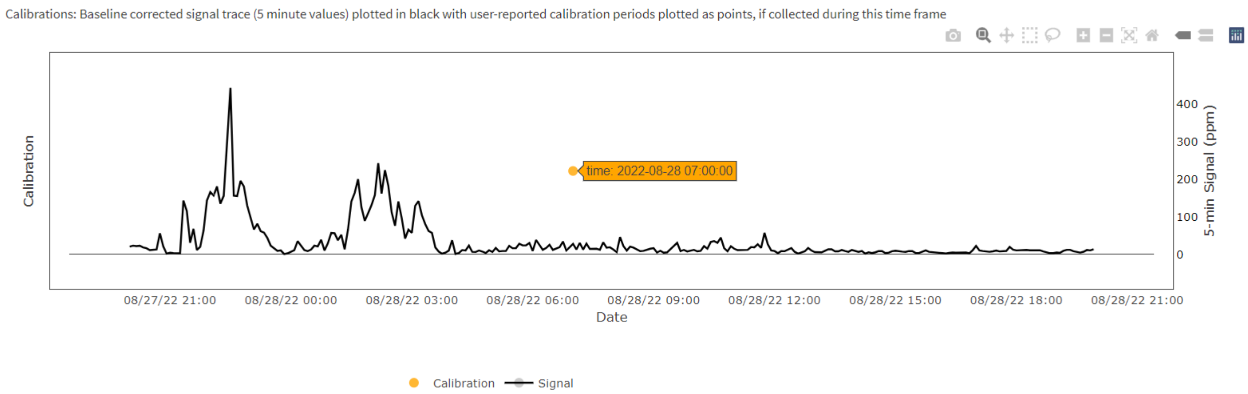



Figure 17: Calibration and Signal time series plot

## 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). The user can save this HTML file to a .pdf file for recordkeeping.

# Data Table

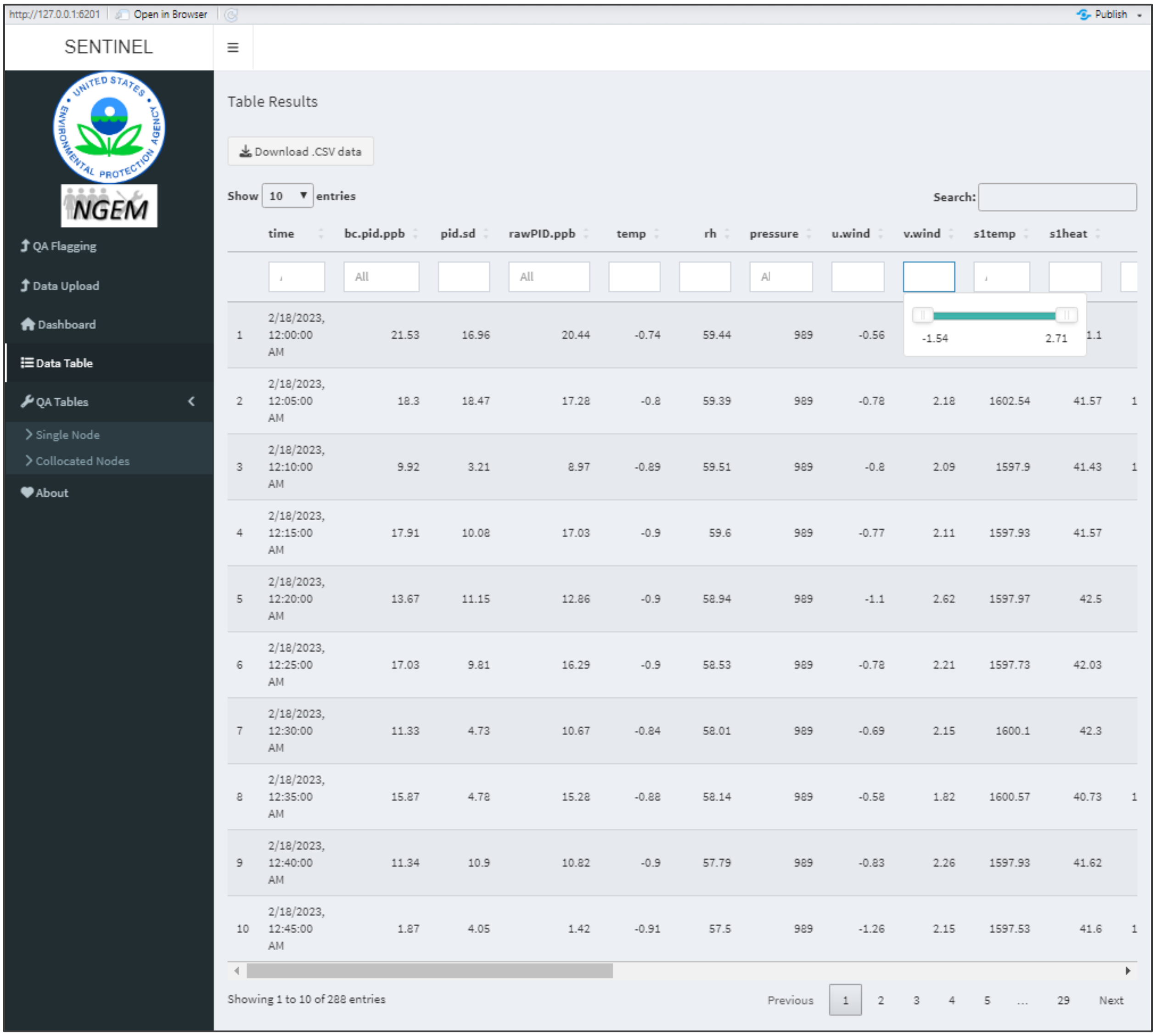


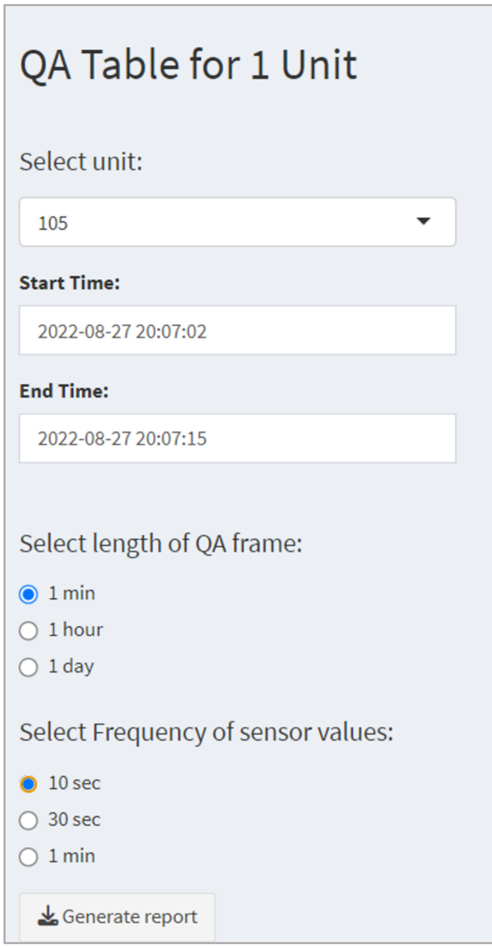

Figure 18: Interactive Data Table

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.

# 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



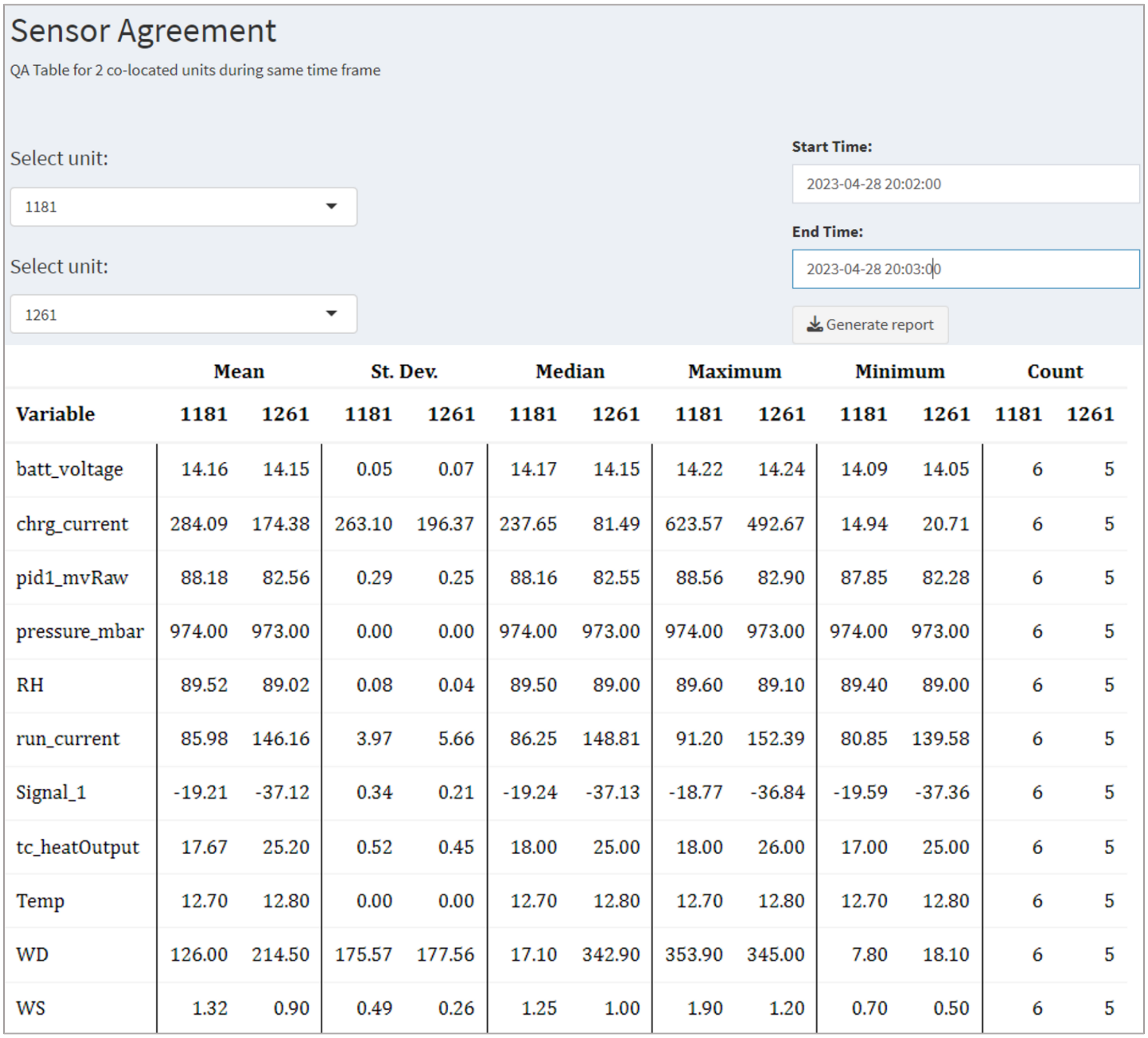

| Variable | Mean | SD | Median | Max | Min | % Complete |
|---|---|---|---|---|---|---|
| Canister | NaN | NA | NA | -Inf | Inf | 0.00 |
| RH | 72.80 | 0.00 | 72.80 | 72.80 | 72.80 | 33.33 |
| Signal_1 | 80.20 | 0.40 | 80.20 | 80.48 | 79.92 | 33.33 |
| Temp | 23.00 | 0.00 | 23.00 | 23.00 | 23.00 | 33.33 |
| WD | 150.35 | 9.26 | 150.35 | 156.90 | 143.80 | 33.33 |
| WS | 0.45 | 0.07 | 0.45 | 0.50 | 0.40 | 33.33 |
| batt_voltage | 14.27 | 0.01 | 14.27 | 14.27 | 14.26 | 33.33 |
| chrg_current | 903.32 | 150.25 | 903.32 | 1009.56 | 797.07 | 33.33 |
| pid1_mvRaw | 154.90 | 0.30 | 154.90 | 155.11 | 154.69 | 33.33 |
| pressure_mbar | 984.00 | 0.00 | 984.00 | 984.00 | 984.00 | 33.33 |
| run_current | 104.09 | 1.28 | 104.09 | 104.99 | 103.18 | 33.33 |
| tc_heatOutput | 23.00 | 0.00 | 23.00 | 23.00 | 23.00 | 33.33 |
| tc_setPoint | 2511.00 | 0.00 | 2511.00 | 2511.00 | 2511.00 | 33.33 |
| tc_temp | 2515.00 | 1.41 | 2515.00 | 2516.00 | 2514.00 | 33.33 |
| trig.canister_status | 17.00 | 0.00 | 17.00 | 17.00 | 17.00 | 33.33 |
| trig.trig_eventFlag | 33.00 | 0.00 | 33.00 | 33.00 | 33.00 | 33.33 |
| trig.trig_value | NaN | NA | NA | -Inf | Inf | 0.00 |

Figure 7: Single node QA Table screen

The Single node calibration page (Figure 19) requires the user to select the unit they would like to QA from a drop-down list. This list is populated by all unique Sensor IDs that were detected when the user uploaded data files on the "Data Upload" page. The time start will automatically generate based on the first available timestamp for that data file. The user should type the date/time start and end they wish to analyze into these text boxes without altering the format. The user should then select the length of time they entered in the duration radio buttons (1 min, 1 hour, or 1 day are the current options for data completeness checks – other time frames can be entered but the data completeness will not be accurate). The frequency of the data readings in the file should be selected in the Frequency radio buttons. For a calibration, this will likely be only 1 minute. For other QA purposes, this could be 1 hour.

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.

## MULTI NODE




| | Mean | | St. Dev. | | Median | | Maximum | | Minimum | | Count | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Variable** | **1181** | **1261** | **1181** | **1261** | **1181** | **1261** | **1181** | **1261** | **1181** | **1261** | **1181** | **1261** |
| batt_voltage | 14.16 | 14.15 | 0.05 | 0.07 | 14.17 | 14.15 | 14.22 | 14.24 | 14.09 | 14.05 | 6 | 5 |
| chrg_current | 284.09 | 174.38 | 263.10 | 196.37 | 237.65 | 81.49 | 623.57 | 492.67 | 14.94 | 20.71 | 6 | 5 |
| pid1_mvRaw | 88.18 | 82.56 | 0.29 | 0.25 | 88.16 | 82.55 | 88.56 | 82.90 | 87.85 | 82.28 | 6 | 5 |
| pressure_mbar | 974.00 | 973.00 | 0.00 | 0.00 | 974.00 | 973.00 | 974.00 | 973.00 | 974.00 | 973.00 | 6 | 5 |
| RH | 89.52 | 89.02 | 0.08 | 0.04 | 89.50 | 89.00 | 89.60 | 89.10 | 89.40 | 89.00 | 6 | 5 |
| run_current | 85.98 | 146.16 | 3.97 | 5.66 | 86.25 | 148.81 | 91.20 | 152.39 | 80.85 | 139.58 | 6 | 5 |
| Signal_1 | -19.21 | -37.12 | 0.34 | 0.21 | -19.24 | -37.13 | -18.77 | -36.84 | -19.59 | -37.36 | 6 | 5 |
| tc_heatOutput | 17.67 | 25.20 | 0.52 | 0.45 | 18.00 | 25.00 | 18.00 | 26.00 | 17.00 | 25.00 | 6 | 5 |
| Temp | 12.70 | 12.80 | 0.00 | 0.00 | 12.70 | 12.80 | 12.70 | 12.80 | 12.70 | 12.80 | 6 | 5 |
| WD | 126.00 | 214.50 | 175.57 | 177.56 | 17.10 | 342.90 | 353.90 | 345.00 | 7.80 | 18.10 | 6 | 5 |
| WS | 1.32 | 0.90 | 0.49 | 0.26 | 1.25 | 1.00 | 1.90 | 1.20 | 0.70 | 0.50 | 6 | 5 |

Figure 20: 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 unit to be compared to the first unit (Figure 20). These Sensor ID's are generated by the Data Upload files. For this comparison to be effective, the two selected nodes should be co-located and operating during the same time period, so sensor agreement can be evaluated. The user once again enters the start and end time values. Only variables contained in both data frames will be considered. This table compares the two values between the two nodes and counts the values recorded by each unit during this time frame as opposed to a percentage of data completeness. The user has the option to export this QA table record keeping. The multi-node QA page also includes a time series graph showing sensor agreement between these two units.

# 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 this User Guide is also included on this page, as well as an overview of the app and acknowledgements to contributors to this project.

# 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)