**Crucial Needs:**

1. **Average blanks**

**1 Overview**

Ice Nucleating Particles (INP) are … One in a million particles act as INP at -20C

They are crucial to determining various properties of clouds, including precipitation rates, lifetime, and shortwave reflectivity and longwave emissivity. Since the effects of aerosols on cloud optical properties and radiative forcing is the single most uncertain component of radiative forcing of Earth’s climate, this makes understanding INP massively important.

The field is nascent and fast moving. Not only that, but Unncertainty is on the scale of orders of magnitude. Any time not spent wrangling and preprocessing data can be spent finding high impact results. Pycetools addresses these two problems. First, by offering a set of guidelines for INP data workup means less errors during data workup. Second, it’s easy to prepare, so you can spend less Spend less time working up your data, and more time finding high impact results.

**2 Definitions**

The **Pyce Tools** workflow has a very strict naming convention. For this reason, it’s important to first define some terms.

INP samples come in various **TYPES**, which can be collected from a range of **LOCATIONS,** and are subject to different **PROCESSES** before analysis with the LINDA instrument.

**TYPE:** *[aerosol, seawater, mq]* the category of the INP sample. This is most generally either seawater or aerosol. Note that if the sample is a seawater blank, it is defined as ‘mq’. If the sample is an aerosol blank, it is still classified as aerosol, since the blank used is a filter.

**LOCATION:** *[uway, wboat\_sml, wboat\_ssw, bubbler, coriolis]* where the sample was collected. For instance, seawater type samples can come from the boat’s underway or from workboat deployments. Workboat measurements are further specified as either wboat\_sml or wboat\_ssw depending on where in the water column the sample originated. Aerosol type INP samples can come from a bubbler, coriolis, or ambient measurements. For blanks samples, the options are bubbler, Coriolis, mq, mq\_wboat. This is kind of a weird naming convention. Sorry.

**PROCESS:** *[uf, f]* a technique used on an INP sample to gather more information. Possible processes include HEATING the sample or FILTERING it. Others include H2O2 but we do not do that in these samples. The way the code is written currently is that the process of either filtering or leaving samples unfiltered is described in the sample name, while heated and unheated is actually assumed to have been conducted during initial sample analysis. For this reason, only UF/F is specified in sample names.

These three terms defined above are used throughout the Pycetools workflow, which is described below.

**3 INP Workflow**

Throughout the workflow of an INP sample, several file types are made. They are termed raw, calculated, and cleaned. The analysis workflow and how each file type fits into it is generalized as follows:

1. An INP sample of a specific type is collected from a specific location.
2. The sample then undergoes whatever post-collection processes the experimenter decides.
3. The sample is analyzed with LINDA, and **raw** data files are saved to their appropriate folder with the appropriate naming convention.
4. Raw data files are then pre-processed into **calculated** report files using sample template spreadsheets. This step involves, among other things, subtraction of the blank (See Blank Correction section). Calculatedreport files are then saved to their appropriate folder.
5. Calculated report files from the calculation step are then cleaned and saved as **cleaned** output files, ready for analysis. During this step, error bars can be calculated and saved in separate files as well.

The specifics of these steps are further described in the following sections.

**3.1 Saving raw data files**

It is imperative that raw files are saved in the correct location and with the correct names. The rules are described for the two sample types (i.e., aerosol or seawater).

Seawater Sample Type Naming Convention:

*seawater\_[location]\_[process]\_[DDMMYY]\_[HHmm]*

These files are saved into the following folder: *[PROJECT\_ROOT]\\data\\raw\\IN\\[SAMPLE\_TYPE]\\[FILE]*

Aerosol Sample Type Naming Convention:

**IMPORTANT NOTE:** When I initially began creating this workflow, I named files with dayXX. This is bad and I should not have done it. For this reason, there’s a section of code that uses a hash table to allocate dayXX with specific dates and times. Unless you are analyzing these specific samples (i.e., Coriolis samples from Tan2020 S2C), you can ignore that section of code. Going forward, files should be saved using the following convention.

aerosol\_[location]\_[process]\_[size]\_[DDMMYYYY\_HHhmm]

These raw files are saved into the following folder: \\[PROJECT\_ROOT]\\data\\raw\\IN\\[SAMPLE\_TYPE]\\[FILE]

**3.2 Creating calculated sample report files from raw data**

Calculated report files are created by processing the raw data using either calculate\_raw\_blank or calculate\_raw functions. Raw blank calculation is described in the Blank Correction section below. Here we describe the process of creating a calculated sample report file for raw data, assuming that calculated blank data files are ready for use.\*\*

Raw data files are processed using template spreadsheets. These template spreadsheets have the necessary equations already inside of them, so that the code simply needs to place the values in the correct locations. There is a template for seawater and aerosol sample types. These should be placed in your project root directory.

The overall process of calculating report files is as follows: Raw file is loaded. Metadata is calculated and listed for raw data source, type, location, process, sample source name, sample collection date, sample analysis date, number of tubes, ml per tube, issues, and sigma. Sigma is used for error bar calculation and should be left at 1.96 for confidence intervals of 95%.

If the sample is of aerosol type, additional metadata is calculated and used in INP calculation. These parameters are rinse volume, size, average flow, sample collection time (calculated from start and stoptime) and total sampled air volume (calculated from sample collection time and average flow). Thenn, raw data is loaded into the template spreadsheet and calculations are made. Finally, a blank data file is loaded into the template file and used to subtract from raw data.

**Important Note:** Right now, only a single blank file is used. Eventually it would make sense to add functionality to average all relevant blank files into one file which is then subtracted from the data. For now, your best bet is to do this manually yourself by calculating several blank files individually (see **Section 3.3 Blank Correction**), and then averaging them into one blank file and passing that file as a parameter when calling the calculate\_raw function.

The calculated report file is then saved to its appropriate location according to the following convention: \\[PROJECT\_ROOT]\\data\\interim\\IN\\calculated\\[SAMPLE\_TYPE]\\[SAMPLE\_LOCATION]\\[type]\_[location]\_[process]\_[date]\_[time]\_calculated.xlsx

You will want to check over the calculated report file yourself as the template may not calculate across all temperatures conducted in your specific experiment. Simply extending the equation further down to lower temperatures by dragging a cell should suffice.

**3.3 Blank Correction**

INP raw data files from the LINDA need to be blank corrected. To do this, blanks are collected, analyzed with LINDA, calculated, and finally subtracted from experiment raw data files. The following functions are used in the preparation of blank files.

**calculate\_raw\_blank** function to calculate raw data files of blanks. When conducting LINDA experiments, the name of the blank file should follow this naming template:

For seawater type samples:

[type]\_blank\_[process]\_[collection\_date]

For aerosol type samples:

[type]\_blank\_[process]\_[size]\_[collection\_date]

It is crucial that naming conventions are followed, so see specific function documentation for more details.

The calculated report file will include metadata on the following: raw data source, type, location, process, sample source name, sample collection date, sample analysis date, number of tubes, volume (ml) per tube.

If the sample is type aerosol, the metadata will also include rinse volume and size.

The calculated report file is then saved to the appropriate folder. See function documentation for specifics.

**IMPORTANT NOTE: As of 02/11/2020 there is not yet any functionality to average the various blank files. This is obviously something that needs to be implemented eventually.**

**3.4 Cleaning**

The cleaning function loads and cleans each calculated report file into a format that’s easier to use in Python. Some of the steps include renaming columns and reshaping the dataframe. Most importantly, it combines all the calculated report files in a folder into a single combined time series. This means you will want to ensure each experiment has its own folder. The ouput of this function is a CSV where each row is an observation and each column is a temperature.

Seawater sample types will also have columns for datetime, time, process, type, location, and filtered/unfiltered indicator. The IN values are given in IN/mL of water.

In addition to these columns, aerosol sample types from the bubbler or Coriolis will also have size (bubbler only), start\_date and stop\_date. IN values are given in INP/L of air.

Cleaned files are saved to \\[PROJECT\_ROOT]\\data\\interim\\IN\\calculated\\[SAMPLE\_TYPE]\\[location]+[start\_date]\_[end\_date].csv

**3.4.1 Calculating Uncertainty**

Calculation of Wilson Errors (link to paper) is done using a combination of cells in the calculated report file and some additional python code. The output csv file is saved in the same location as the cleaned combined time series data file and with the same naming convention, but with ‘wilson\_error’ appended to the end.

Lower and upper bounds for blank subtracted frozen fraction of tubes (upperBound, lowerBound) are calculated using subfunctions (wilsonLower and wilsonUpper). These fractions are then converted to a number of blank subtracted tubes that are frozen (upper\_N-BLNK, lower\_N-BLNK, respectively). These bounds are then converted into INP/tube upper and lower bounds. Then they are converted to IN/mL and IN/L upper and lower bounds. Finally, the difference between each bound and the original observed value is calculated to determine the size of the error bars and saved as error\_y and error\_minus\_y. The confidence interval of the uncertainty can be changed by using a different sigma value in the template spreadsheets.

For seawater samples, the units are INP/L seawater. For aerosol samples, the units are INP/L air.

**3.5 Loading and Final Pre-Prepocessing**

While Pycetools does a bulk of the necessary manipulations and pre-processing, there’s a few steps that still should be completed after loading the cleaned combined timeseries files. This is left to the user’s discretion as the specifics of each experiment will vary considerably. For reference, the preprocessing steps for S2C data are shown below.

* Convert datetime column to a pandas datetime index
* Melt the dataframe
* Calculate different units
* Round to nearest hour
* Set to NZ time zone
* Merge uncertainty and concentration dataframes

The code for these steps can be found in **Section 4 Tutorial.**

**4 Particle Size Distributions**

**Inverted data from the scanotron is cleaned using pt.clean\_inverted() function. All inverted files should be saved into a single folder. Here, we choose the following directory path:**

\\[PROJECT\_ROOT]\\data\\interim\\scanotron\\inverted\\pro\\[FILE]

**Similarly, the outpath for the cleaned file can be defined by the user. Here we choose the following:**

\\[PROJECT\_ROOT]\\data\\interim\\scanotron\\combinedtimeseries\\

All files in the input folder will be appended into a single time series csv. Concentrations are lognormalized. As such, the function accepts the number of size bins.

Specifics of the function can be found in the code documentation.

After cleaning inverted scanotron data, it can be loaded using Load\_scano\_data and then further manipulated. Parameters for the load\_Scano\_data function include dates, which is the name of the combined time series file you want to load, and instr, which tells where the file is located.

In addition to the lognormalized size distribution, dN is also returned, which is simply the number of particles in each bin.

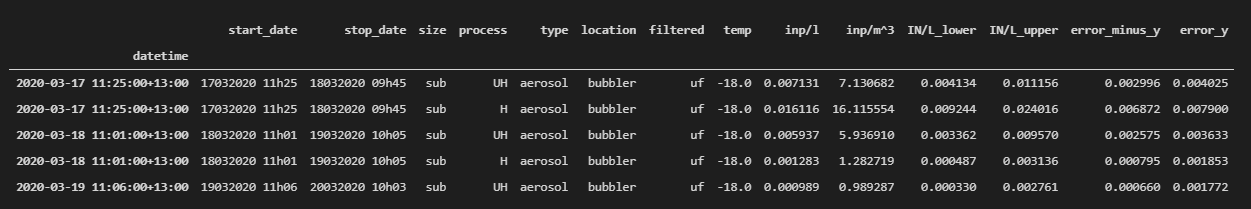
Surface area can be calculated using pt.surface\_area(). See documentation or tutorial for details on usage.

Finally, magic CPC data can be cleaned using pt.clean\_magic(), hwoever this will depend on the specific format your CPC data was saved in.

Plot number dist, plot\_surface\_dist,

**5 Analysis**

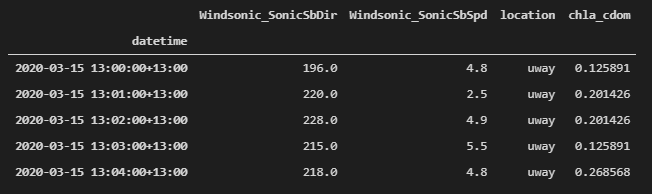
Analysis of INP data is achieved using the INP class. An INP class object is instantiated when given an inp\_type, inp\_location, cyto\_location, a cyto\_data dataframe, a uway\_bio dataframe, and inp dataframe.

Here is an example of what the INP dataframe should look like. This is easily achieved if you’ve followed the steps above.

We see here this is fairly similar to the INP dataframe, but it has been merged with the uncertainty dataframe. Code for how to do this is in the tutorial section.

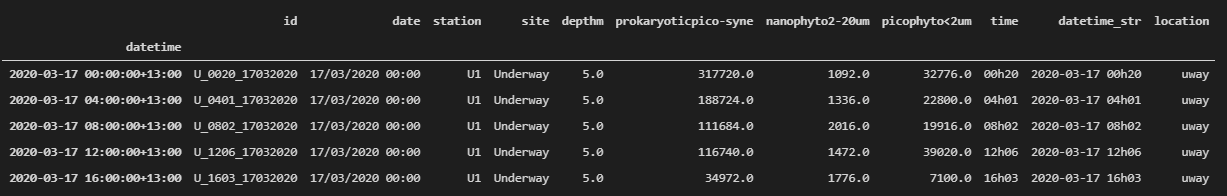
It is important that the index be titled ‘datetime’ and be a datetime object.

The uwaybio dataframe can include any data, but should have the index as datetime. Below is an example:



Furthermore, it’s crucial that a column labeled ‘location’ be defined.

The cyto dataframe will look similar, but it is not necessary (ACTUALLY right now it IS necessary. I should make it optional)

The cyto dataframe will also need to include location information.

When the object is instantiated, it will only take INP data that are of the specified inp location and type.

In this way, we have a final INP object consisting of INP data from a specific location and of a certain type. This means it will contain multiple processes and temperatures and filtered/unfiltered states.

**Calculate surface area normalize INP concentration**

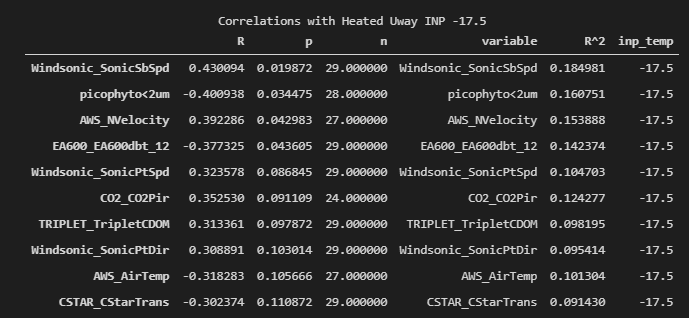
Surface area normalized INP concentrations can be swiftly calculated using inp.sa\_normalize function, and passing it the dA\_total dataframe which is returned from the pt.surface\_area() function. This function assumes you have already organized dA\_total to line up with your INP collection periods. See the tutorial for specifics.

**Plotting with error bars and previous studies**

Update this function

**Correlations and correlation scatter plots**

**Correlations are calculated using pt.correlations. A list of temperatures as strings are sent, as well as a specific process (H, or UH) and inp\_units, which indicates the column containing your INP concentrations. Below is an example of the returned dataframe of correlations.**



The correlations can also be viewed with scatter plots, using inp.plot\_corr\_scatter() function, which returns a figure object which can be further stylized. Below is an example.

**6 Tutorial**

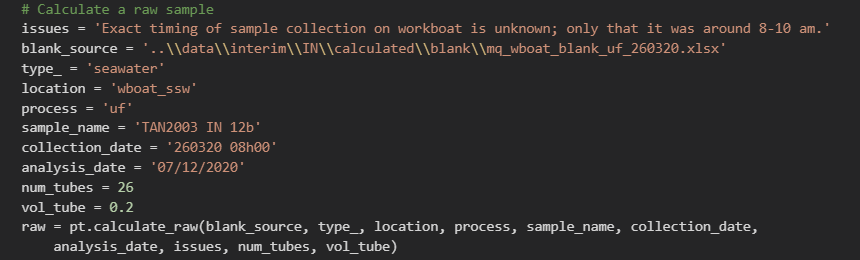
As an example, we will go through the entire process for an INP sample that was collected during the S2C Tangaroa cruise. The sample type is seawater and was collected at the wboat\_ssw sample location.

**Saving raw data files.** Show image of the raw csv and an image of where the csv is saved and its name. The following information should be logged during LINDA analysis:

Type, location, process, sample name, collection date, analysis date, number of tubes and volume (mL) in each tube, issues.

Filter aerosol samples need the following information in addition to the above: flow rate start, flow rate stop, rinsing volume

**Creating calculated report files.** Assume we have a blank already made (for now we manually take the average of each blank file and create a new one which we point to in the calculation).\*\*\*



This will create a calculated report file in which the blank corrected values are given. A separate tab for heated vs unheated samples is also given, with metadata in each. Finally, uncertainty calculations are given in the spreadsheet. This is used by the pt.calculate\_wilson\_errors to create an error spreadsheet.

**Creating cleaned files and uncertainty files.** Cleaning the available calculated report files is as sample as calling clean\_calculated\_in() function and passing the parameters for INP type and location. The script will automatically find all relevant files in the folder.

**Loading and preparing cleaned files for analysis.**

Load the cleaned inp data and uncertainty files. Do a final bit of preprocessing. Then combine them.

**Analysis.**