# Raingauge quality control

## The sites and raingauges

There are three types of raingauge which require quality control: drop counting, tipping bucket and weighing raingauges.

There are 2 sites at which measurements are made: Chilbolton Observatory and Sparsholt Agricultural College.

|  |  |
| --- | --- |
| 100_2407_rg_cabin_aug12 | 100_0342_spgauges |

Raingauge sites at Chilbolton (left) and Sparsholt (right)

All of these raingauges collect the rain in a funnel. The tipping bucket has a divided bucket which tips every time it fills up. Each tipping movement is measured by the gauge. The drop counting gauge passes the rain through a block into a fine dropper tube. Each drop of water breaks a small light beam and is detected by the gauge. The weighing raingauge measures the weight increase in the raingauge reservoir.

Tipping buckets are simple gauges which are widely used. 1 tip of the bucket is equivalent to 0.2 mm of rain. The drop counting gauges were developed at Chilbolton to give better resolution data. 1 drop is equivalent to 0.0033mm of rain for most of the gauges (with the green funnels) or 0.00189 mm for a high sensitivity gauge (with a larger metal funnel). The sensitivity of the weighing raingauge is slightly more variable, but it is typically around 0.05 mm.

The Campbell Scientific PWS100 present weather sensor is used in the quality control process. It can also require quality control itself.

## What goes wrong with the rain measurements?

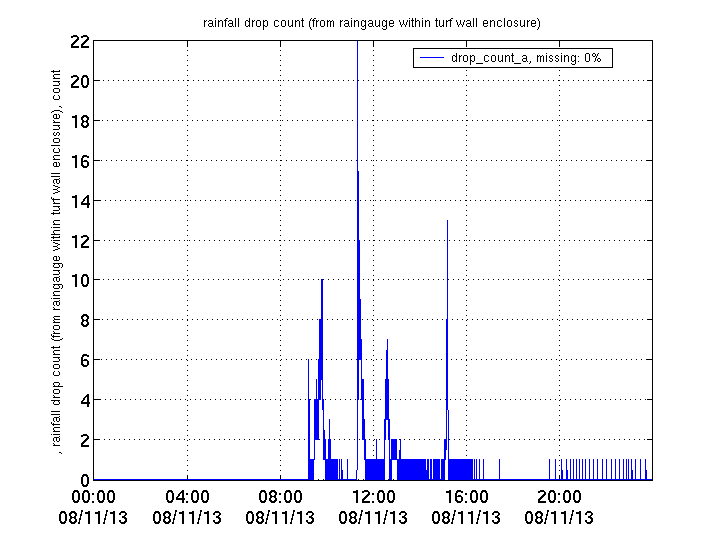
Raingauges can produce counts when there is no rain for several reasons:

1. **Condensation and frost.** These collect on the funnels then produce a slow rate of droplets (appearing very like light drizzle). Frost can look like a short period of rain when it melts. The high sensitivity gauge shows this more strongly than the others because of its metal funnel. Drops due to condensation and frost must be removed from the data. Looking at the air temperature data at <https://gate.chobs.rl.ac.uk/amof-netCDF/ncas-temperature-rh-1_v1.0/index.html> can be helpful. Note that the graphs here are in Kelvin, not centigrade! If the temperature is below about 280 K (7°C) then condensation is likely.
2. **Pump cycles.** Every 12 hours the drop counting raingauge automatically flushes itself with water from its reservoir. The controller is set not to produce counts during that period as they would be wrong, however often 1 or 2 extra counts are seen after the cycle has finished. They are easily spotted as they are 12 hours apart (although can be at different times for different gauges).
3. **Blocked raingauge.** This often looks quite similar to drizzle or condensation, i.e. a low, almost continuous series of counts. Real rain events may not show as a higher rain rate. The data from the affected gauge will be noticeably different to the other raingauges. Pump cycles may show more counts than normal (more than 10). I also watch for this problem occurring, so I can organise cleaning the gauge, so I’ll warn you and remove the data from the record before quality control is done.
4. **Snow.** Measurements of snow can behave erratically depending on the conditions. If the air temperature is relatively high, it may melt on contact with the funnel and be detected. In this case it’s OK to leave the data as it is. However, it may not melt, or may freeze at night, in which case it can behave more like frost. In this case the counts should be removed, as they may well occur well after the snow fell. Going to look at the Chilbolton gauges to see whether snow has accumulated in the funnels can be useful!
5. **Maintenance work.** Erroneous drops can be produced when gauges are cleaned. As with the blocked raingauge cases, I normally look for these counts and remove them before quality control.

All of these occur much more often for the drop counting gauges than the tipping bucket gauges. For the weighing raingauge they are less likely, but it does occasionally produce counts during dry conditions. The cause of this is unknown, but it may at least partly occur when the weather is windy and has been dry for some time, due to wind causing the reservoir to move.

## Rain and condensation

We record all raingauge data as the number of counts (y-axis) v. the time of day (x-axis). Most data are recorded at 10 second intervals (except for the weighing raingauge which is at 60 s intervals), so the plots show the number of counts recorded in each 10 s time period through the day.

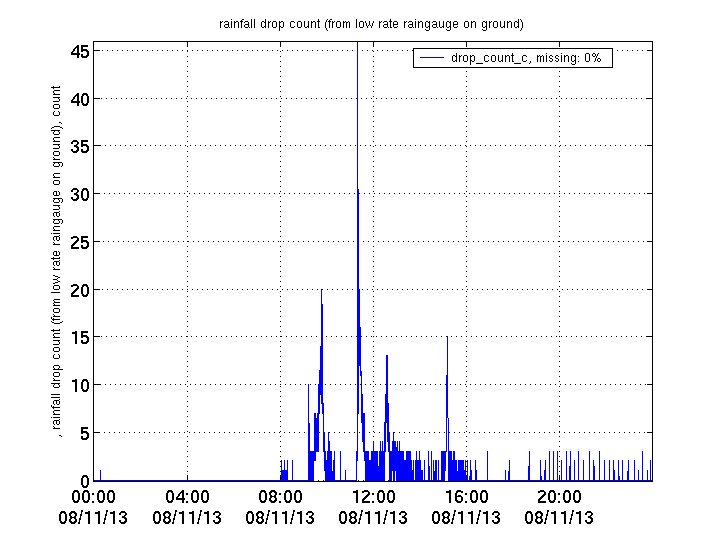


Condensation

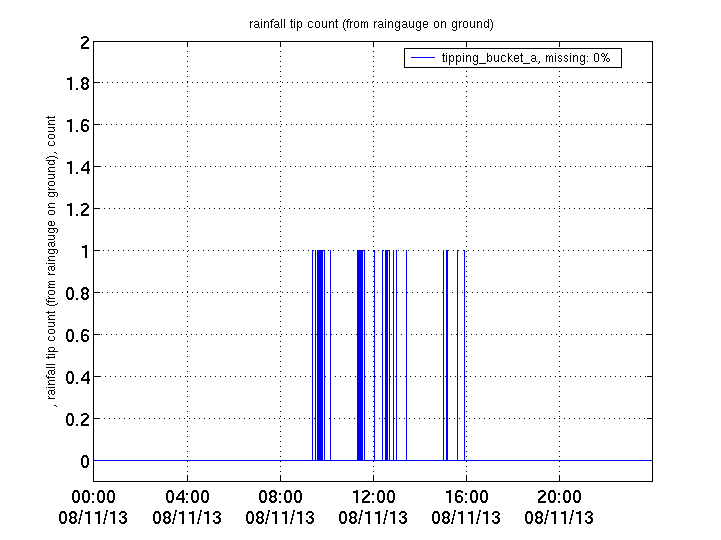
Light rain

Heavy rain

Data from standard rate drop counting gauge (rg001dc\_ch)



Data from high sensitivity drop counting gauge (rg008dc\_ch). Data are quite similar, but there are more condensation counts between 19:00 and 00:00 (midnight).

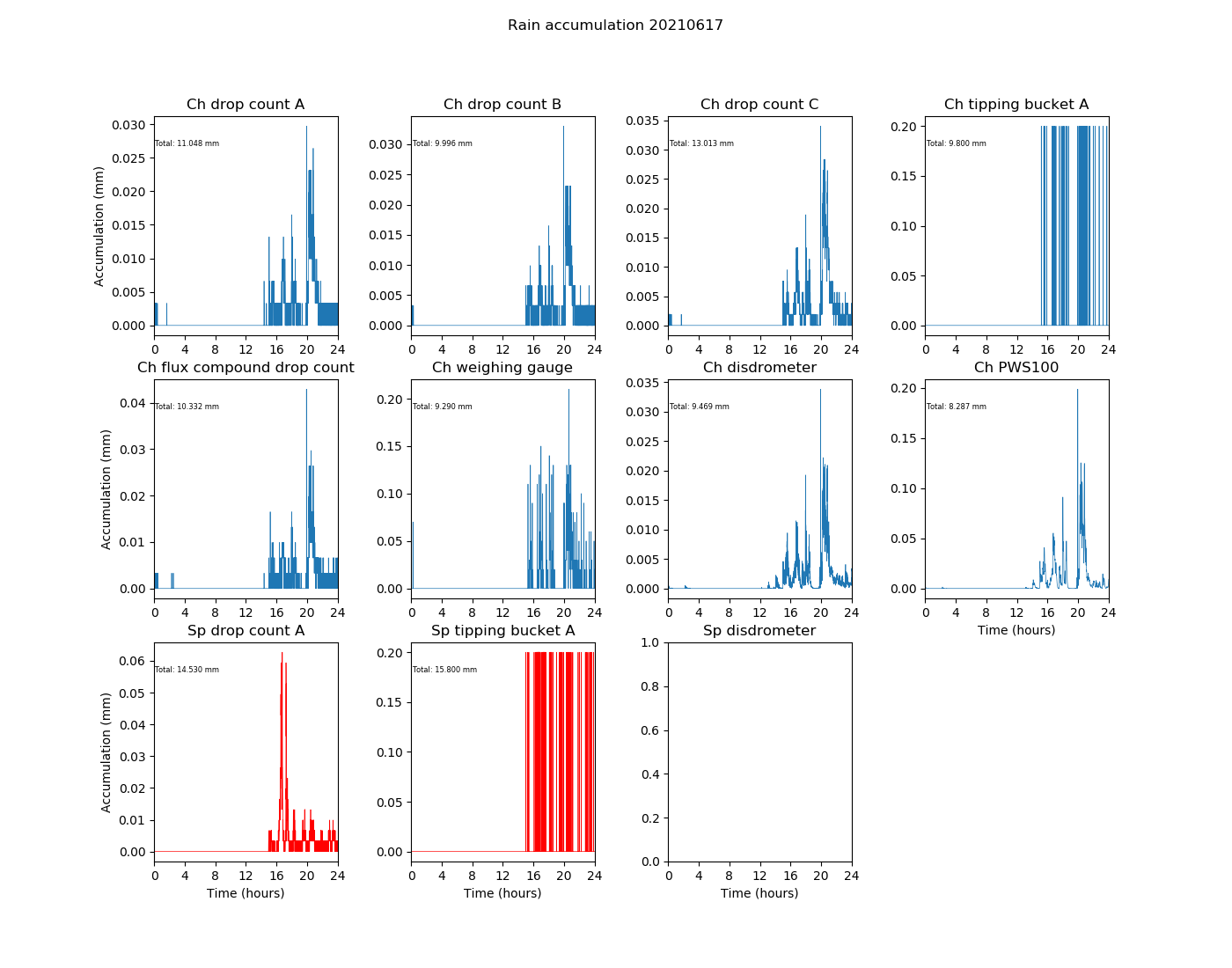


Data from tipping bucket gauge (rg004tb\_ch). Because the tipping bucket gauge requires more rain before it “counts” than the drop counting gauge, only 1 count is ever detected in the 10 second time windows that the raingauges all use to record. The condensation is not visible, whereas the light rain still is, but with quite widely spaced counts. Tipping bucket gauges are not sensitive enough to detect condensation.

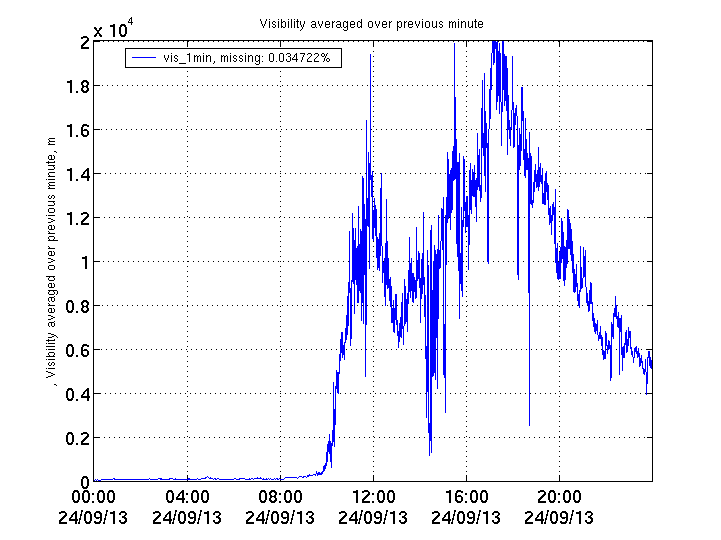
## How to tell if it was really raining

The above example prompts the question “How can you tell that the measurements labelled as condensation weren’t rain”? To do this we have to use some of the other Chilbolton instruments and the Met Office weather radar network.

Every day the program raingauge\_comparison.py is run to produce a combined plot comparing all the Chilbolton and Sparsholt raingauges, disdrometer and present weather sensor. This is useful in assessing differences between raingauges. These plots can be seen in <https://gate.chobs.rl.ac.uk/amof-netCDF/ncas_rain_quality/index.html>. An example is shown below:



1. **Disdrometers and present weather sensor.** These are also raingauges, but disdrometers work by detecting the “taps” as raindrops hit a polystyrene cone and present weather sensors detect the size of the droplet as it passes through 2 light beams. There is a disdrometer at both Chilbolton and Sparsholt and a present weather sensor at Chilbolton. Because they measure individual raindrops, they are insensitive to condensation. They are also very accurate for measuring when rain starts, as they don’t need a funnel to get wet and water to run through it to detect rain. Note that disdrometers are insensitive to snow as it makes a negligible impact when hitting the cone.
2. **Radar data.** There are radar data at <http://gate.chobs.rl.ac.uk/frontiers/>. Open that page and choose the “Frontiers archive and animation” link. It opens showing the most recent data (usually today). Use the buttons at the bottom of the window to go backwards and forwards through the archive, by 5 minute intervals, hours, days, months and years. If there was rain, it will appear as a colour on the graph. The cross hairs are centred on Chilbolton. Note that very light rain or drizzle is the darkest blue shade and that sometimes the radar isn’t very sensitive to the lightest drizzle. It sees heavier rain well and is always useful for seeing if rain was anywhere in the vicinity on a particular day.
3. **Visibility.** The Campbell PWS100 present weather sensor at Chilbolton provides a measurement of visibility. This isn’t a 100% proof of rain or not, but if the visibility is low, e.g. less than 1km (the vertical scale on the graphs is 20 km), the conditions are such that there is likely to be condensation on surfaces. This is particularly true if the temperature is low, less than around 10°C. Note that in the temperature plots described in the next section, the temperature is shown in Kelvin, not °C - 10°C is equivalent to 283 K.



In the above example there was fog from midnight until 10am. The visibility increased rapidly after that. While the visibility was low, conditions would have been particularly conducive to the formation of dew, although it can still happen at other times.

1. **Chilbolton radars and lidars.** These can also show whether it was raining, but the data are more complicated to interpret and they are only necessary for a few particularly confusing rain cases.

## How to make corrections

A new code has been written using Python to enable raingauge quality control. It uses a similar process to the previous Matlab code (plot\_raingauge) but there are some improvements and also some changes which were required due to differences in functionality between Python and Matlab.

The new code still operates in a similar way to previously, in that when erroneous points are highlighted in the window, the corresponding corrections file (instrument\_id.corr) in /data/netCDF/corrections is edited to include the time window of that correction. These corrections apply to both the original format cfarr\*.nc netCDF files and the new ncas\*.nc files. The way in which the corrections are applied is slightly different, but the same need to identify false points still exists.

This correction process identifies data points which are suspicious, via the use of the HOLDCAL label in the corrections file. These are usually rain counts which were recorded when there is no evidence of rain being present. In the old cfarr\*.nc netCDF files, these were set to a value of zero over the identified time period. In the new ncas\*.nc files the data value is not changed, but a flag is set to indicate that the value is suspicious if it is above zero.

This code does not handle cases where there is known to be a fault with the gauge, such as a blockage or failure. These cases currently have to be manually entered in the correction file as BADDATA. Alternatively they can be marked using the procedure shown below, then the correction file must be edited manually to change that correction from HOLDCAL to BADDATA. They are then shown as a fill value in the netCDF file.

Text shown in bold is a command to be typed. Exceed needs to be running on your computer.

1. In a browser, open <https://gate.chobs.rl.ac.uk/amof-netCDF/ncas_rain_quality/index.html> to access raingauge comparison plots. Other sensors such as air temperature from ncas-temperature-rh-1 and visibility from the PWS100 present weather sensor can also be useful
2. In another tab in the browser, open the Met Office weather radar images <http://gate.chobs.rl.ac.uk/frontiers/> and choose the Image Archive and Animations link. Both of these pages will be useful as you work through the data.
3. Start Exceed on your Windows computer if it hasn’t been started automatically.
4. Use PuTTY to start a terminal window connected to Wilma.
5. Log in to Wilma using usual username and password.
6. **ssh python**
7. This transfers you from Wilma to the “python” computer.
8. Enter your password again
9. Enter the command **conda activate chil\_3\_8**
10. Connect to the directory where the code is located using
11. **cd /home/chilbolton\_software/python/ncas\_python/raingauge**
12. Enter **./raingauge\_click\_plots.py -s *yyyymmdd***
13. Substitute the date you want to process for *yyyymmdd* e.g. 20210425
14. After a few seconds a plot screen will appear. If it doesn’t automatically appear as the foreground window you will need to select the Exceed icon in the taskbar to see it.
15. Click on each plot to either side of any data points that you want to remove. The pointer stays as a rather clunky pointer rather than neater crosshairs, but position it to either side of the point as best you can.
16. Unlike with the Matlab version, don’t press Return when moving on from one plot to the next. This code recognises which plot you have clicked on by its position in the window.
17. Proceed through the plots along the top row, then the bottom row. If you realise you missed something, don’t go back. Finish this procedure then start again from command 11, just correcting the missed point.
18. When you have reached the end of the series of plots, click the Close (X) button of the window to finish. At this point the corrections get written to the relevant corrections files, provided no errors were detected.
19. The code will spot if there have been a total of an odd number of clicks on plots, or if there are any where the 2nd point of a pair is earlier than the first. If this happens the program will end and no corrections will be written. An error message which begins “Error in clicking plots” will be displayed in the text window where you enter commands.
20. If no errors in the corrections were detected, after a few seconds a window will appear which shows which data were removed by outlining that region in red. Once you have checked whether you are happy with the corrections, click the Close(X) button and the program will end. The message “Corrections successfully written” will be shown in the text window where you enter commands.
21. If you spot an error in the corrections that you made, or the code reports an error, make a note of the day and which gauges had a problem.
22. You should have a command prompt at the start of the line, as at the start. If not, there may still be a plot window which needs to be closed. Return to command 11 to process other days.
23. When you have finished work, enter exit to leave the Python machine, then exit again to leave Wilma. Or you can stay on the Python machine to run other programs.

Behind the scenes, the program is added information to a corrections file for each raingauge, stored at /home/data/netCDF/corrections. They all have the name: \*.corr, where the \* is the name of each raingauge.

Table x in the document “Use of generate\_days\_netcdf\_metsensors.py” describes the IDs and .nc data file names of each raingauge.

## Manually editing corrections files

These can be edited with any Unix editor such as nano or vi, or with a Windows-style editor such as Pluma. The instructions below assume Pluma is being used.

1. Open a new Putty window and login.
2. Enter the command “pluma”. This opens a program quite like Notepad on a Windows computer.
3. Move around the directory structure to /home/data/netCDF/corrections, then select and open the file you require, e.g. rg001dc\_ch.corr.
4. The corrections that you just generated will be at the end of this file, so scroll down until you see them.
5. Delete or modify lines as you wish, then save the file and exit the editor.
6. You can then run raingauge\_click\_plots.py again and re-do the days that you need.

## Producing new netCDF files

You are now ready to apply the corrections to the netCDF files on the computer. This is done with generate\_days\_netcdf\_metsensors.py, as described in “cao\_metinstruments\_netcdf\_production.docx”.

Any data points marked as HOLDCAL will be highlighted with a red hollow circle in the plots at <https://gate.chobs.rl.ac.uk/amof-netCDF/>. Any marked as BADDATA will no longer be visible in the plots.

## PWS100 present weather sensor corrections

The most common fault that occurs with the Campbell Scientific PWS100 present weather sensor is that cobwebs occur in one or more arms of the instrument. Dead insects and spiders can also accumulate. This appears to occur most frequently in autumn but can occur at any time of year.

This fault appears to have no significant effect on rainfall measurements, and has no effect on temperature and relative humidity measurements which are recorded using a separate small weather station. The measurements which are most impacted are visibility and synoptic code. The visibility that is reported is lower than the values that would be expected and the synoptic code in clear conditions will often not be zero as it should be. Typically values of around 10 or 20 occur in clear conditions instead. Values will still be 61 in light rain, backing up the view that rain measurements are less affected by debris in the instrument.

The older Vaisala PWD21 present weather detector (cfarr-pwd21\_chilbolton), also located on the receive cabin roof, can be used to compare visibility and synoptic code. For unknown reasons, this instrument is less likely to be affected by insects and spiders.

There are 3 different correction files for this instrument: pws100.corr, pws100\_met.corr and pws100\_rain.corr. These cover faults in visibility and synoptic code, temperature and relative humidity, and rain and hydrometeor corrections respectively. If you wish to remove visibility and synoptic code data, manually edit pws100.corr, using the “BADDATA” correction. If you click on the rainfall plot in raingauge\_click\_plots.py, this will add a “BADDATA” correction to pws100\_rain.corr. This will remove any data relating to precipitation, including hydrometeor type and drop size distribution. If you wish to remove temperature and relative humidity data, manually edit the pws100\_met.corr file.