

Generating a profile of mean CALIPSO aerosol extinction: an example of basic quality screening techniques for CALIPSO level 2 profile extinction products using MATLAB

In this example we demonstrate how to generate a vertical profile of mean extinction from the CALIPSO level 2 profile products with some basic data quality screening. The process of averaging extinction profiles is quite straight forward. However, applying the CALIPSO profile descriptive flags for quality screening is a bit tricky because they are stored in a three dimensional array while extinction coefficients are stored in a two dimensional array. More on this later... Since it is important to pay attention to these flags, this document provides a practical example of how to apply profile descriptive flags to screen extinction data.

The case we will use to demonstrate this process was acquired April 17, 2010 at about 20Z, shown in Figure 1 below. The CALIPSO total attenuated backscatter measurements reveal an aerosol layer over India extending from the Ganges Valley near Kathmandu (left) to the Bay of Bengal, just south of Vishakhapatnam (right). The vertical feature mask in Figure 1b shows that CALIPSO level 2 algorithms have detected this aerosol layer from about 5 km in altitude to the surface. We will use this case to demonstrate how to compute a vertical profile of mean extinction with basic quality screening.

The MATLAB code and associated sample data that accompanies this example can be downloaded as a zip file from the Tools section of the CALIPSO Data User's Guide. Please unzip the .mat files and .hdf file to your active MATLAB directory to follow along. The main function is called *quality_screen_CALIPSO_profile_products.m*.

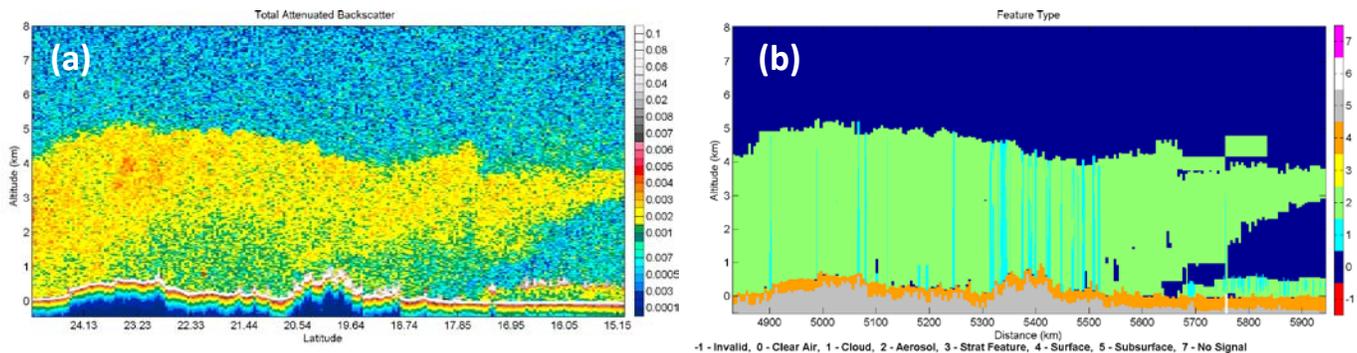


Figure 1. (a) CALIPSO total attenuated backscatter in $\text{km}^{-1}\text{sr}^{-1}$ and **(b)** level 2 vertical feature mask of feature type for an aerosol layer over India on April 17th, 2010.

Step 1. Get relevant CALIPSO data.

For this example, we are using the level 2 aerosol profile granule, CAL_LID_L2_05kmAPro-Prov-V3-01.2010-04-17T20-09-13ZN.hdf. This granule is included in the zip file that accompanies this example.

If you would like to use the [CALIPSO Search and Sub-setter Tool](#) to get a smaller, subsetted version of this granule, follow the link and select Granule Type to be night. Next, select Extinction Coefficient 532 nm from the LIDAR Level 2 5-km Aerosol Profile Product. By default, the subsetter tool will include Profile Information which contains the profile descriptive flags that we will use for quality screening later. Now, set the temporal range to April 17, 2010, and for the geospatial range, select a bounding box of (15°N,25°N; 80°E, 90°E). Once you submit your order, follow the instructions to download the .hdf file that will be sent to you via email from ASDC User Services.

Step 2. Open the CALIPSO profile file and extract relevant data.

The extinction coefficient array and all profile descriptive flag arrays can be read into MATLAB using the readHDF.m tool available in the Tools section of this User's Guide. For this example, the extinction coefficient and its uncertainty are read in as the variables `sigma` and `sigma_uncer`. Additionally, the profile descriptive flags (for quality screening) CAD score, atmospheric volume description, and extinction QC flag 532 are all read in.

A note about the array sizes...

The two dimensional arrays of extinction and its uncertainty have size `numberAlts x numberProfiles`, indicating the number of altitudes and number of profiles, respectively. On the other hand, profile descriptive flags such as CAD Score, Extinction QC, and the Atmospheric Volume Description are all three dimensional arrays of size `2 x numberAlts x numberProfiles` in the CALIPSO profile products. This is because the level 2 feature finder and scene classifier operate at 30 m vertical resolution in the lower troposphere, but the level 2 profile products are averaged and reported at 60 m vertical resolution at those altitudes. So, to keep track of the two 30 m resolution layers in each 60 m extinction range bin, the descriptive flags in the profile products are three dimensional with the first dimension discriminating between the higher of the two 30 m layers (index = `1 x numberAlts x numberProfiles`) and the lower of the two 30 m layers (index = `2 x numberAlts x numberProfiles`) that comprise each 60 m bin. For more details about how profile descriptive flags are stored, see the Atmospheric Volume Description entry in the Level 2 Profile Products Data Quality Summary.

Step 3. Selectively screen the extinction coefficients.

Now that we have our array of extinction coefficients `sigma`, we need to screen out samples with which we have less confidence. Here, we will use a logical array called `useSamples`, having the same size as `sigma`, to indicate which samples to keep. Elements in this array that are 'true' will be kept for averaging later, while those that we specify as 'false' will be screened out of the average. If you prefer a more technical example, suppose an element in `useSamples` is 'false' at indices (i, j) . Then we will exclude the corresponding element in `sigma` at indices (i, j) from the average.

Screen out fill values

This is the easiest thing to screen out! Fill values indicate that there are no samples to report. If you are new to MATLAB, the following expression will set every element in `useSamples` to false where the corresponding elements in `sigma` are -9999.

```
useSamples(sigma == -9999) = false;
```

Screen out all features that are not aerosols

It is important to recognize that profile descriptive flags stored in the level 2 aerosol profile products not only contain information about aerosols, but they also contain information about clouds, stratospheric features, and every other type of feature the scene classifier identifies. This is so you can know the state of the entire atmosphere all at once using the same array, the most fundamental of which is the atmospheric volume description (AVD).

The AVD is very similar to the feature classification flag in the CALIPSO level 2 layer products in that it is a bit-mapped integer that stores the feature type and subtype (amongst other things) for each layer. We could also use this array to screen out clouds overhead or to identify bad profiles. For AVD definitions, see the CALIPSO Data Products Catalog.

Since stratospheric features are also reported in the aerosol profile extinction array and we are not interested in them, let's use the AVD to screen out everything that is not an aerosol. For our purposes, once the feature type has been decoded out of the AVD, aerosol layers have feature type = 2. The following function extracts the feature type and subtype from the AVD array; both having the same size as AVD.

```
[ftype, subtype] = get_feature_type_and_subtype(AVD);
```

Now recall that our logical array `useSamples` is two dimensional, but `ftype` is three dimensional to account for the two 30 m features per 60 m `sigma` range bin. It is possible that in a given `sigma` range bin that one 30 m element is an aerosol and the other 30 m element is a cloud, stratospheric feature, or even 'clear air'. In this example, we are going to use every range bin containing at least one aerosol feature. The following MATLAB code uses an OR operator such that if either the higher 30 m range bin OR the lower 30 m range bin contains an aerosol, then we use the sample. Otherwise we do not. The `squeeze` function simply makes the logical expression in parentheses into a two dimensional array. Similar to how we screened out fill values, we will make every element in `useSamples` 'false' where the AVD does not contain aerosols (`~` is the MATLAB NOT operator).

```
isAerosol = squeeze(ftype(1, :, :) == 3 | ftype(2, :, :) == 3);  
useSamples(~isAerosol) = false;
```

Screen by CAD Score

CAD Score reflects our confidence that the feature under consideration is either an aerosol or a cloud. When CAD = -100, we have the most confidence that the feature is an aerosol and when CAD = 100, we have the most confidence that the feature is a cloud. CAD = 0 says we really do not know. So, what

CAD values should we accept, then? This sometimes depends on the situation and aerosol subtypes involved, but a good rule of thumb is to reject aerosol layers with CAD scores between 0 and -20. This is not guaranteed to screen out all bad stuff, but our analysis of version 3.01 data suggest most layers whose retrievals were suspicious due to possible type misclassification had CAD values in this range. If you want to be very conservative, you can screen out features with CAD scores > -80 and preserve only the layers which we have higher confidence that they are aerosols.

For this example, we choose to screen out CAD > -20. This time, if either of the two 30 m samples contain aerosols with CAD > -20, we will discard the entire 60 m extinction sample. This can be accomplished with a simple AND statement. But! The situation is a little more complicated because CAD scores for clouds are also reported in the CAD scores array in the level 2 profile products. To ensure that we do not discard cases where one of the 30 m elements contains an aerosol layer with a CAD that is < -20 and the other 30 m element contains a cloud, we will first set all of the cloud CAD scores to a fill value. Bear in mind that this is not the only way to accomplish this! This is just one example to highlight the problem. After this step, we will keep cases where the CAD in the first element (1, :, :) is less than -20 OR equal to the fill value AND the second element (2, :, :) is less than -20 OR equal to the fill value. Any elements not meeting these criteria are set to 'false' in the last line.

```
CAD_fill = -127;
CAD(CAD > 0) = CAD_fill;
goodCAD = squeeze( (CAD(1, :, :) < -20 | CAD(1, :, :) == CAD_fill) &...
                  (CAD(2, :, :) < -20 | CAD(2, :, :) == CAD_fill) );
useSamples(~goodCAD) = false;
```

Screen with extinction QC flags

CALIPSO extinction QC flags summarize the final state of the extinction retrieval. During the iterative process to find a solution, the lidar ratio (an assumed parameter which is required to derive extinction from backscatter) may be adjusted. Often, we are most confident in solutions where the lidar ratio is unchanged during the extinction retrieval (extinction QC = 0) or if the retrieval is constrained (extinction QC = 1). For more information on the retrieval process and the information stored in extinction QC flags, see the final lidar ratio and extinction QC entries in the CALIPSO Layer Products Data Quality Summaries.

For this example, we will keep all 60 m elements where both 30 m element which comprise it have and extinction QC value of 0, 1, or a fill value (when there are no features). Similar to the CAD score array, cloud extinction QC values are included in the extinction QC array, so we will first set all elements corresponding to clouds equal to a fill value using the feature type array ftype from earlier. Again, this is not the only way to account for this! It's just one way.

```
extQCfill = 32768;
extQC(ftype ~= 3) = extQCfill;
goodExtQC = squeeze(...
    (extQC(1, :, :) == 0 | extQC(1, :, :) == 1 | extQC(1, :, :) == extQCfill) & ...
    (extQC(2, :, :) == 0 | extQC(2, :, :) == 1 | extQC(2, :, :) == extQCfill));
useSamples(~goodExtQC) = false;
```

Screen by extinction uncertainty

The CALIPSO extinction retrieval is an iterative process that sometimes does not converge to a solution. When the retrieved extinction begins to grow without bound, the extinction uncertainty is assigned a value of 99.99 to indicate that the solution has “blown up”. Extinction values should not be used when the extinction uncertainty is 99.99. So, we will screen them out with the following statement.

```
useSamples(sigma_uncer > 99.9) = false;
```

Step 4. Average the screened extinction coefficients.

For this example, we are going to plot vertical profiles of mean extinction with and without quality screening so we can compare the two. Furthermore, we will plot vertical profiles of the extinction relative uncertainty (extinction uncertainty divided by the mean extinction) with and without quality screening.

To accomplish this in MATLAB, we will go through the arrays one altitude at a time, screen out all samples that do not meet our criteria, and calculate the mean for that altitude. For the unscreened mean extinction (`meanSigmaUnscreened`), we will simply filter out all extinction samples that are not fill values (-9999) using the logical array `t` below. For the screened mean extinction (`meanSigmaScreened`), we will screen out all extinction values whose corresponding elements in `useSamples` are not ‘true’.

```
for k = 1:numberAlts
    t = sigma(k,:) ~= -9999;
    meanSigmaUnscreened(k) = mean(sigma(k,t));
    meanSigmaScreened(k) = mean(sigma(k,useSamples(k,:)));
end
```

...

In order to calculate the relative uncertainty, we must propagate the extinction uncertainty over the mean. Using standard error propagation techniques, the propagated uncertainty is the square root of the sum of the squares of the uncertainties divided by the number of samples in the mean. The example below is for the unscreened uncertainty – the screened uncertainty case is similar. The first two lines ensure division by zero does not occur when there are no samples to average.

```
numSamplesUnscreened(k) = sum(t);
if numSamplesUnscreened(k) > 0
    meanSigmaUnscreened_uncer(k) =
        sqrt(sum(sigma_uncer(k,t).^2) ./ numSamplesUnscreened(k));
end
```

Step 5. Plot the figures.

The final section of the example MATLAB code plots the mean extinction as a function of altitude, the number of samples in the average, and the relative uncertainty in percent for the unscreened (blue dashed lines) and screened (red lines) cases. The figures are reproduced below. As shown by the reduction in relative uncertainty, quality screening with these filters has improved the quality of the mean extinction profile.

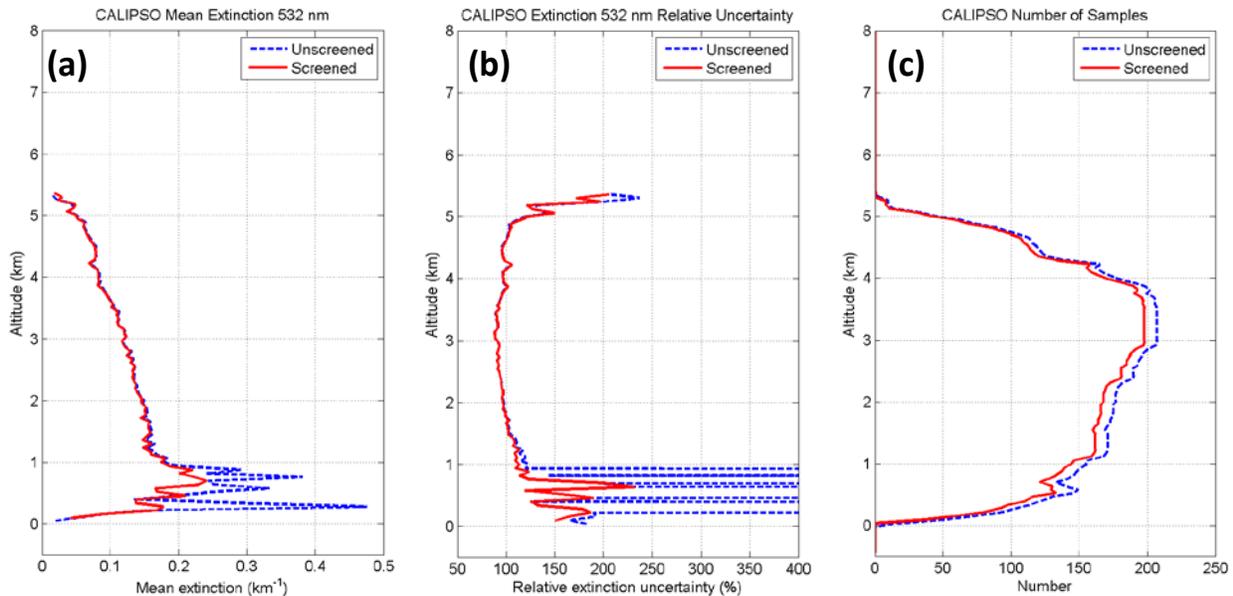


Figure 2. (a) Mean extinction, (b) relative uncertainty in percent, and (c) number of samples in the average for the example.

Summary and discussion

In this example, we generated a vertical profile of CALIPSO mean aerosol extinction using profile descriptive flags included in the level 2 profile products for basic quality screening. We learned how to interpret and apply the three dimensional profile descriptive flags AVD, CAD score, and extinction QC to screen out extinction samples which we have less confidence.

The mean extinction and relative uncertainty profiles in Figure 2 demonstrate why it is important to pay attention to the quality flags built into the profile products when interpreting the data. For instance, the relative uncertainty below 1 km exceeded 400% before screening, and after removing a relatively small number of troublesome samples, it was reduced to a more manageable (albeit still large) 125-200%.

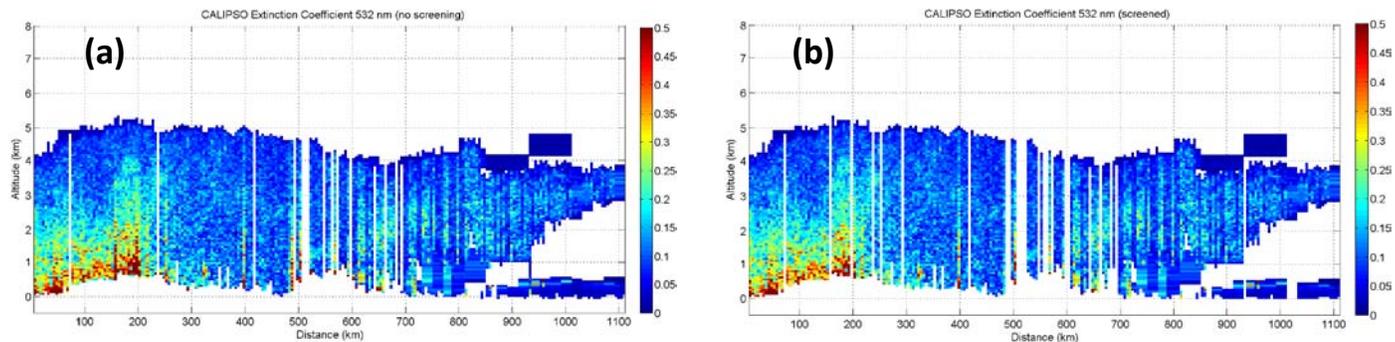


Figure 3. CALIPSO aerosol extinction coefficients in km^{-1} **(a)** unscreened and **(b)** screened using the profile descriptive flags for the April 17th, 2010 case examined here.

To illustrate which samples were excluded by the screening process, we show plots of the extinction coefficients with and without screening in Figure 3. It is clear that not many samples were removed, but those that were tended to have high extinction values at the lowest altitudes of the layer. No doubt their uncertainty values were large too, since it is the nature of a downward pointing lidar to have increasing uncertainty with decreasing altitude.

As a final note, it must be stressed that these are basic quality screening approaches. This example only serves to highlight the tools available to the user in the profile products so that they can begin quality assessment on their own. For instance, the CAD threshold that we chose, requiring aerosols to have $\text{CAD} < -20$ is only a basic rule of thumb. Some users may want to be more restrictive. Also, we chose to use only extinction solutions where the lidar ratio was not adjusted or the solution was constrained (extinction QC = 0, or 1). Advanced users who are familiar with CALIPSO extinction retrieval algorithms may want to explore extinction values with extinction QC values beyond this range, but they should do so carefully. Another issue at hand is how extinction uncertainties propagate to lower layers in the retrieval process. If something goes wrong in the extinction retrieval for a particular layer up high, all layers below it will be influenced. Furthermore, retrievals for aerosol beneath clouds may be less trustworthy, depending on how well the extinction was retrieved in the cloud. We did not address these issues here as they are worthy of another document on their own which we hope to provide in the near future as part of the Data User's Guide.

Revision history.

Original document: Oct. 1, 2010.

Revision 1 (this document): Oct. 21, 2010.

Revision 2 (this document): Jan 3, 2019.