BErkeley High Resolution (BEHR) NO_2 product - User Guide

Josh Laughner

May 26, 2015

1 References

2 Product overview

2.1 Product types

Currently, we have two data products available: one at the native OMI pixel resolution and one in which each swath has been gridded to a $0.05^\circ \times 0.05^\circ$ fixed grid. The gridded product is ideal for users who simply wish to obtain an NO_2 VCD, as the latitude and longitude of each grid point will remain fixed over time, whereas the native OMI pixels do not. However, the native OMI resolution files have additional variables compared to the gridded product, such as scattering weights, averaging kernels, our NO_2 *a priori* profile, etc. that will be useful to users wishing to modify the product in some way.

Plans to produce a third product with a single, daily grid are underway and we hope to have this available soon.

2.2 Version numbering

The BEHR version numbering system combines the OMNO2 version number with an internal version letter. So, v2.1A represents the first BEHR product based on version 2.1 of the NASA OMNO2 product. A subsequent version of BEHR still based on version 2.1 of OMNO2 would be v2.1B. If OMNO2 were to update to version 3, the BEHR product based on that would be v3.0A.

Should a minor change be made (e.g. one that adds information to the output but does not change the core algorithm), a revision number will be appended to the version number. For example, v2.1A and v2.1Arev0 will be the same, but v2.1Arev1 would indicate this sort of minor change.

2.3 File format

All products will be made available as HDF version 5 (`.h5`) files (*c.f.* `https://www.hdfgroup.org`). Please note when trying to open these that many programming languages and utilities have different commands and tools for opening version 4 and 5 HDF files. If you are having trouble opening these files:

1. Ensure that you are using the correct command for an HDF5 file, not an HDF4 file.
2. Try using HDFView (available from `https://www.hdfgroup.org/products/java/index.html` to browse the file. This will confirm that it downloaded properly.
3. Check if the utility or programming language you are using requires the HDF5 library (`https://www.hdfgroup.org/HDF5/`) to be installed.

For both the native and gridded products, the HDF files are organized similarly. Under the `/Data` group, each swath is contained within its own group, named as `Swath#`. There will be 3–5 swaths per day. Each swath will contain all relevant variables as datasets.

Starting with BEHR version 2.1A, fill values will be directly stored in the HDF FillValue information for each dataset, along with four additional attributes: Description, Range, Product, and Unit. *Description* is a brief, one-line description of the meaning of each variable. *Range* is the range of values that variable may correctly take on. *Product* indicates whether this dataset is copied directly from the NASA standard product (represented by **SP**) or is added by the BEHR product (unsurprisingly represented by **BEHR**). Finally, *Unit* is the physical unit assigned to each dataset.

The native product will also be provided as comma separated value files. Because part of the value of the gridded product is the 2-dimensional structure of the grid reflects the geographic distribution, and because this would be difficult to maintain in a .csv file, we will continue to provide the gridded product in HDF format only.

2.4 Tools for working with HDF files

The following programs or programming languages are known to be able to read HDF files:

- MATLAB: current versions have high-level functions such as `h5info` and `h5read` which can easily read in HDF5 files. This does not seem to rely on the external HDF library.
- Python: the `h5py` package (`http://www.h5py.org`) can read HDF5 files, however it does depend on having the HDF library installed, at least on Unix based systems.
- IDL: various users have sucessfully read HDF5 files in IDL; however since we do not use it ourselves, we cannot offer specific advice on the best way to do so.
- Igor Pro, v. $>$ 5.04: `http://www.wavemetrics.com/products/igorpro/dataaccess/hdf5.htm`
- GNU Octave: `https://www.gnu.org/software/octave/`

This is not an exclusive list, however these are common scientific software packages that indicate they have the capability to read HDF5 files. Note that our experience is focused on MATLAB and Python, so our ability to offer specific advice for other utilities is limited.

3 Variables

4 Considerations when working with BEHR files

4.1 Pixel filtering

For users wishing to identify high quality NO_2 column information to ground, there are six key criteria to look for:

1. The VCD must be > 0 (values < 0 are present in the OMNO2 data, but likely result from error in stratospheric separation)

2. The vcdQualityFlags field must be an even number. This indicates that the summary bit is not set, meaning there were no significant processing issues.

3. Cloud fraction: BEHR contains three cloud fractions: OMI geometric, OMI radiance, and MODIS cloud fraction. We generally filter for OMI geometric or MODIS fraction to be < 0.2 (20%). As discussed in Russell et al. 2011, MODIS cloud fraction is often less susceptible to identifying high albedo ground surfaces as clouds.

4. The column amount should be $< 1 \times 10^{17}$. Such values are expected to indicate that the pixel has been affected by the row anomaly.

5. The column amount must not be a fill value.

6. Filter for row anomaly, typically by requiring that the XTrackQualityFlags field = 0.

4.2 Weighting temporal averaging

When averaging the gridded product over time, use the *Areaweight* variable to weight each contribution. This is calculated for each grid cell as the average of the reciprocals of the areas of each pixel included in that grid cell. Thus, grid cells containing contributions from large pixels are weighted less than those from smaller, more representative pixels.

4.3 Using scattering weights/averaging kernels

4.3.1 Variable layout

These are for advanced users who might wish to either use their own *a priori* NO_2 profile or to compare modeled NO_2 VCDs correctly with BEHR VCDs. Note that the averaging kernels for each pixel are simply the scattering weights divided by the BEHR AMF; they are provided separately simply for user convenience.

For new users wishing to begin using these variables, several resources will be helpful. Palmer et al. (2001) contains the original formulation of the relationship between scattering weights, *a priori* NO_2 profiles, and air mass factors and should definitely be studied. The first several chapters of *The Remote Sensing of Tropospheric Composition from Space* assembled by Burrows, Platt, and Borrell (Burrows et al., 2011) also describes the relationship of scattering weights, averaging kernels, and air mass factors.

What follows will be a description of why the scattering weights are presented how they are. BEHRScatteringWeights, BEHRAvgKernels, BEHRPressureLevels, and BEHRNO2apriori are 3D variables; the first dimension is the vertical dimension, the second and third correspond to the dimensions of the normal 2D variables. To put this another way, if X were the array of values for these variables, then $X(:,1,1)$ would be the vector of values for the $(1,1)$ pixel, $X(:,1,2)$ the vector for the $(1,2)$ pixel and so on.

The BEHRPressureLevels variable gives the vertical coordinates for the other three. 28 of the pressure levels will be the same for every pixel; the remaining two will correspond to the terrain and cloud pressure. Should the terrain or cloud pressure match one of the standard 28 pressures, then the vector of values for this pixel will still be 30 elements long, but will end with fill values which should be removed.

4.3.2 Calculation of scattering weights

The scattering weights presented (w_{tot}) are calculated as the weighted average of a vector of clear and cloudy scattering weights obtained by interpolating the NASA OMNO2 TOMRAD look up table to the appropriate values of SZA, VZA, RAA (relative azimuth angle), albedo, and surface pressure. Mathematically:

$$w_{i,\text{tot}} = f_r w_{i,\text{cld}} + (1 - f_r) w_{i,\text{clr}} \tag{1}$$

$$w_{i,\text{clr}} = \begin{cases} 0 & \text{if } p_i > p_{\text{terr}} \\ w_{i,\text{clr}} & \text{otherwise} \end{cases} \tag{2}$$

$$w_{i,\text{cld}} = \begin{cases} 0 & \text{if } p_i > p_{\text{cld}} \\ w_{i,\text{cld}} & \text{otherwise} \end{cases} \tag{3}$$

where f_r is the radiance cloud fraction, $w_{i,\text{clr}}$ is the ith element in the clear sky scattering weight vector, $w_{i,\text{cld}}$ is the ith element in the cloudy scattering weight vector, p_i is the pressure level for the ith element in the vectors, p_{terr} is the terrain pressure for the pixel, and p_{cld} is the cloud top pressure for the pixel.

Therefore the final vector of scattering weights is the cloud radiance fraction-weighted average of the clear and cloudy scattering weight vectors after setting the clear sky vectors to 0 below the ground, and the cloudy vectors to 0 below the cloud top. This is an approximation of how the NASA scattering weights are reported. These scattering weights can be combined with an *a priori* NO_2 profile to get the total AMF as:

$$A = \int_{p_{\text{terr}}}^{p_{\text{trop}}} g(p) w(p) \, dp \tag{4}$$

where p_{terr} is the terrain pressure, p_{trop} is the tropopause pressure, and $g(p)$ is the *a priori* profile shape factor. Note that integration must be done w.r.t. number density, see Appendix B of Ziemka et al. (2001) for information on how mixing ratio integrated over pressure can be made equivalent to number density integrated over altitude.

4.3.3 Justification for including terrain and cloud pressure weights

Although we publish the scattering weights in this form, we calculate our AMF as:

$$A = f_r A_{\text{cld}} + (1 - f_r) A_{\text{clr}} \tag{5}$$

where

$$A_{\mathrm{cld}} = \int_{p_{\mathrm{cld}}}^{p_{\mathrm{trop}}} g(p) w_{\mathrm{cld}}(p)\, dp \tag{6}$$

$$A_{\mathrm{clr}} = \int_{p_{\mathrm{terr}}}^{p_{\mathrm{trop}}} g(p) w_{\mathrm{clr}}(p)\, dp \tag{7}$$

Mathematically, this should be the same as:

$$A_{\mathrm{tot}} = \int_{p_{\mathrm{terr}}}^{p_{\mathrm{trop}}} g(p) w_{\mathrm{tot}}(p)\, dp \tag{8}$$

however computationally it was not. This was because in Eqns. 6-7 we interpolate $w(p)$ to the terrain or cloud top pressure *before* setting $w(p)$ to zero below those pressures. This is done because the interpolation should not modify the scattering weights—surface pressure is already accounted for in the lookup table—but rather is only meant to find the scattering weight at our lower integration limit.

However, if the same approach is carried out with w_{tot} in Eqn. 8, we are not interpolating between the same values, because now we *have* set $w_{\mathrm{clr}}(p)$ and $w_{\mathrm{cld}}(p)$ to 0 below terrain and cloud top, respectively. To avoid this problem, $w_{\mathrm{clr}}(p)$ and $w_{\mathrm{cld}}(p)$ are each interpolated to both terrain and cloud top height before being used in Eqns. 1–3. Using data from 1 Aug 2013, with the same NO_2 profile, the difference between the AMFs calculated using Eqns. 5 and Eqn. 8 decrease from a median of 12.9% without pre-interpolating to 0.299% with pre-interpolating.

References

Burrows, J., Platt, U., and Borrell, P.: The Remote Sensing of Tropospheric Composition from Space, Springer, 2011.

Palmer, P., Jacob, D., Chance, K., Martin, R., Spurr, R., Kurosu, T., Bey, I., Yantosca, R., Fiore, A., and Li, Q.: Air mass factor formulation for spectroscopic measurments from satellites: Application to formaldehyde retrievals from the Global Ozone Monitoring Experiment, J. Geophys. Res., 106, 14 539–14 550, 2001.

Ziemka, J., Chandra, S., and Bhartia, P.: “Cloud slicing”: A new technique to derive upper tropospheric ozone from satellite measurements, J. Geophys. Res. Atmos., 106, 9853–9867, 2001.