Berkeley **H**igh **R**esolution (**BEHR**) Retrieval: Readme

Josh Laughner

December 5, 2014

Summary:

The Berkeley High Resolution Retrieval is a high resolution NO_2 retrieval based on the NASA OMNO2 product from the Ozone Monitoring Instrument (OMI) aboard the Aura satellite. This retrieval improves the standard OMNO2 product in several ways:

1. A higher resolution terrain product (GLOBE Terrain Database) is used to calculate the terrain pressure of the pixels
2. A more frequent and finer resolution albedo is used, taken from the Moderate Resolution Imaging Spectrometer (MODIS) instruments aboard the Terra and Aqua satellites.
3. A MODIS cloud fraction is available to use in rejecting cloudy pixels, as the OMNO2 cloud fraction tends to overestimate cloud fractions if highly reflective surfaces are present

This document will describe the current state of BEHR, including its file structure and a changelog. As of this writing, the code is maintained in a Git repository on my machine. I will try to keep the main matter of this document up-to-date; however, if there is a discrepancy between the changelog and the main matter, defer to the changelog.

Contents

1 Authors

BEHR was initiated by Ashley Russell, who completed her Ph. D. in 2012. She showed that using high-resolution albedo and terrain data improved the OMI NASA Standard Product retrieval (see her papers in the Literature section below). Check the group website for her current contact information.

Luke Valin also completed his Ph. D. in 2012; he helped Ashley run the WRF-Chem simulations needed to get the high-resolution NO_2 profiles.

Josh Laughner took over development in 2013.

If you contribute to the development of BEHR, add your name and contribution to this list and update the change log.

2 Literature

3 Retrieving NASA data

3.1 Web sites

3.2 Group server file locations

4 Version Control

The core code for BEHR is contained in a Git repository on our group's Synology DS1813+ NAS file server. As of December 5, 2014, this server is located at IP address **128.32.208.13**. Currently to connect to the server, your computer needs to also be on the 128 network, which means an ethernet connection to one of the wall ports or that you access wifi via the RC-Lab network. Credentials are:

Username: RCCohenLab
Password: SurfaceToStratopause

If you are not familiar with Git, I recommend reading chapters 1-4 at `http://git-scm.com/doc`, which includes information on installing Git on your system as well as the basic commands. If you are working on a Windows machine you make also want to look at `http://guides.beanstalkapp.com/version-control/git-on-windows.html` and follow their recommendations, although as I've never used Git on a Windows machine, I can't say for certain how well that works. Once you have Git installed and working on your machine, navigate to the folder that will be your working directory for the project in either Terminal (Mac) or Command Prompt (Windows) and run the command:

```
git clone ssh://RCCohenLab@128.32.208.13/volume1/share-sat/SAT/BEHR/
BEHR_GitRepo.git
```

This will mirror the repository as a new folder in that directory. A second repository, at `128.32.208.13/volume1/share-sat/SAT/BEHR/MiscUtils.git` contains some general utility Matlab scripts that I have found useful. Some were downloaded from the Matlab file exchange, many were functions I found myself needing rather often. A third repository is `128.32.208.13/volume1/share-sat/SAT/BEHR/AircraftProfiles.git`, with functions to work with aircraft data sets. Additional repositories will be added to Table 1. I recommend that these be downloaded to folders called "BEHR", "Utils", and "NO2 Profiles" within your main Matlab directory (which you can check with `userpath` at the Matlab command prompt. This way, any internal links that I've set up to be robust as `fullfile(userpath, x,y,z,...,file)` should work smoothly on either a Windows or Mac platform.

Address	Rec. folder	Contains
`128.32.208.13/volume1/share-sat/SAT /BEHR/BEHR_GitRepo.git`	BEHR	All the code needed to run BEHR
`128.32.208.13/volume1/share-sat/SAT /BEHR/MiscUtils.git`	Utils	Miscellaneous utilities not needed for BEHR but generally helpful
`128.32.208.13/volume1/share-sat/SAT /BEHR/AircraftProfiles.git`	NO2 Profiles	Functions to work with aircraft data sets, including code to validate satellite data against such data sets.
`128.32.208.13/volume1/share-sat/SAT /BEHR/BEHR_MatlabClasses_GitRepo. git`	Classes	Certain MATLAB classes I wrote to help manage certain tasks (e.g. error messages)

Table 1: Summary of the IP addresses, recommended folders within the main Matlab directory, and contents of the three Git repositories.

This repository is in place now and will be kept relatively up-to-date as I progress. I will also try to remember to make a simple copy of the BEHR code to the file server which will not require the use of Git to obtain. However, there are several reasons you should consider learning to use Git if you haven't yet. (If you have, you probably stopped reading this as soon as I told you where the repository was.)

1. *It keeps everything in one place, and makes it easy to keep everything up to date.* As long as you make changes in the directory that the Git repo consists of (and commit

the changes periodically), those changes are tracked. So, you can roll back to an earlier version if something breaks, or see what code you (or I) changed.

2. *Parallel development.* If multiple people are developing BEHR at one point, each person can create their own branch and develop simultaneously, while still being able to update the project on the server, and eventually merge the development lines together.

3. *Code sharing.* Conversely, if two (or more) people are both using BEHR, this makes it easier to synchronize code when and if you want.

5 File structure

5.1 Read data

The first step in running BEHR on new data is to read the NASA files into Matlab. This is done using the `read_omno2_v_aug2012.m` file. This is a Matlab script, ergo it is not called with any arguments; changes need to be made to the code itself. The "`v_aug2012`" part of the name indicates that this file is intended to work with OMNO2 v. 3, which was released in Aug. 2012 (or at least the technical specs for it were). This script serves several functions:

1. It reads all relevant variables from OMNO2 files into Matlab.

2. It averages the MYD06_L2 cloud product data to each OMI pixel.

3. It averages the MCD43C3 albedo product data to each OMI pixel.

4. It averages the GLOBE terrain data to each OMI pixel, converting from altitude to terrain pressure.

For each day that this script processes, a `.mat` file is saved with a single variable, `Data`. This variable is a data structure, in which each OMI swath for that day is stored under a different top-level index (i.e., `Data(1)` refers to the first swath of that day, and `Data(2)` the second, etc.). These data structures contain the data read from the OMNO2, MODIS, and GLOBE files as matrix fields. Each element of the matrix corresponds to an OMI pixel.

Production (i.e. not testing) files output from this script will generally have the name `OMI_SP_yyyymmdd.mat`. Filenames with additional information are generally testing files I have created in the course of various debugging runs, and should not be used to produce NO_2 data.

To run this file, enter the latitude and longitude boundaries for the area to retrieve (any OMI pixels outside this area will be discarded) and the date range to process. The first time you run this file, you may need to edit the paths to the various files; follow the instructions in the comments to identify which directory corresponds to which type of file. To specify a path to a folder on the lab server, Mac users should begin the path with `/Volumes`. PC users: I can't help you. Sorry! (Best guess, the path might need to start with the server IP address.)

Should a new version of the OMNO2 files be released, lines 192–325 (dealing with various H5 loading functions to read OMNO2) might need updated to reflect any change in dataset names in the OMNO2 he5 files. The MYD06 and MCD43C3 files are loaded further down in the code; follow the comments.

5.2 Recalculate AMF and Tropospheric Column

5.3 Weight pixels and map

6 Additional utilities

6.1 Verification

7 Changelog

Date	Version Number	Description
2012	2.0A	Original version by A. Russell, using OMI SP 2 with MODIS cloud and albedo, GLOBE terrain database, and WRF-Chem profiles (12 km for full US).
May 2013	2.0B	Version (unvalidated at this point) updated by J. Laughner to use OMI SP 2 and MODIS cloud collection 6. Albedo, terrain products, and WRF-Chem profiles unchanged. MODIS and GLOBE data import now done directly by `read_omno2_v_aug2012`, "preprocessing" of these files into Matlab .mat files no longer necessary.