Jan 2014

Sfit4 Layer1 Documentation

Network for the Detection of Atmospheric Composition Change (NDACC) / Infrared Working Group (IRWG)

Author(s)	Reviewer	Approver
Eric Nussbaumer Stephanie Conway	Jim Hannigan	Jim Hannigan

Table of Contents

1 **Scope** **3**

1.1 Reference Documents and Material 3

2 **Layer1 Pre-requisites** **4**

2.1 Spectral Database 4

2.2 Creating the Spectral Database 4

2.3 Input Data File Structure 5

2.4 Collecting HITRAN Data 5

3 **Layer1** **6**

3.1 Layer1 Input File 6

3.2 Output File Structure 7

3.3 Running Layer1 8

4 **Error Analysis** **9**

4.1 Sb ctl File 9

4.1.1 Error Analysis Outputs 9

4.1.2 Error Analysis Inputs 9

4.1.3 Sb Values 9

5 **Plotting and HDF creation** **10**

Revision History **11**

Figures and Tables

Figure 1 File structure for input data example 5

Figure 2 Output File Structure 8

1 Scope

The purpose of this document is to outline and detail the code and procedure for sfit4-Layer1. Sfit4-Layer1 is a collection of programs, which allow for the batch processing and error analysis of atmospheric retrievals accomplished with the Sfit4 program. Plotting and conversion of data to HDF is currently not supported in Layer1.

1.1 Reference Documents and Material

- Sfit4 wiki site: https://wiki.ucar.edu/display/sfit4/Infrared+Working+Group+Retrieval+Code%2C+SFIT
- NDACC/IRGW website: http://www.acd.ucar.edu/irwg/
- Sfit4 bug reporting website: http://acd.ucar.edu/~ebaumer/bugzilla/
- Sfit4 Documentation:
    - Code_desc
    - EFAPOD_v3.5
    - Sfit4_init
    - Sfit4-get.start.quick
    - Sfit4-hbin_descrip
    - Sfit4-isotope_descrip
    - Sfit4-output_descrip

2 Layer1 Pre-requisites

2.1 Spectral Database

The spectral database is a space-delimited file that contains information (columns) related to each observation (rows). This file is essential for the operation of Layer1 and must contain a certain parameters of the observation; however, it can also contain information not necessarily used by Layer1.

The following parameters currently need to be included in the spectral database for Layer1:

1. Date (YYYMMDD)
2. Time (HH:MM:SS)
3. TStamp (HHMMSS) (Layer1Mods t15ascPrep) Redundant?!?
4. LWN (Lower Wave Number)
5. HWN (Upper Wave Number)
6. Flt (Filter ID)
7. SZen (Solar Zenith Angle)
8. FOV (Field of View)
9. N_Lat (North Latitude of instrument)
10. W_Lon (West Longitude of instrument)
11. Alt (Altitude of instrument)
12. ROE (Radius of Earth)
13. ExtStatTemp (Surface Temperature from external source i.e. NOAA)
14. HouseTemp (Surface Pressure from external source i.e. NOAA)
15. ExtStatPres (Surface Temperature from instrument)
16. HousePres (Surface Pressure from instrument)

The parameters: ExtStatTemp, HouseTemp, ExStatPres, HousePres are used for creating the reference profile. Values of -9999 can be used in the database if these parameters do not exist or are unknown. If one uses a different refmaker to create the reference profile then these values may not be needed in the spectral database.

Each row of the spectral database corresponds to an OPUS or bnr type file. Layer1 uniquely associates a bnr file to an entry in the spectral database through a date and time-stamp. The date is indicated by the folder name in which the file is located (as discussed later) and bnr file name corresponds to the time-stamp.

2.2 Creating the Spectral Database

Software programs exists to create the spectral database

- ckopus.c – A C program which converts an OPUS file into a binary file and extracts the relevant spectral database parameters from the binary file.
- mkSpecDB.py – A python program that builds the spectral database using outputs from the ckopus.c program
- specDBInputFile.py – The input file for mkSpecDB.py program

- appendSpecDB.py – A python program to append weather data (ExtStatTemp, HouseTemp, etc) to the spectral database
- appndSpecdBInputFile.py – The input file for appendSpecDB.py

To create a spectral database, complete the following steps:

1. Organize input data into file structure discussed in section 2.3
2. Run mkSpecDB.py to create database
3. Run appendSpecDB.py to append database with external weather data

2.3 Input Data File Structure

The input file structure for Layer1 is assumed to be organized by date. The binary and/or opus files are expected to reside under the folder with the appropriate date as the name. For example, spectra taken on January 8th, 2014 would reside under the folder 20140108 (YYYYMMDD). The date folders are expected to reside under folder with the three letter abbreviate for the station location. For example, the file structure for the station 'MLO' with observations on the dates January 1st, 2012 and January 2nd, 2012 are depicted in Figure 1.

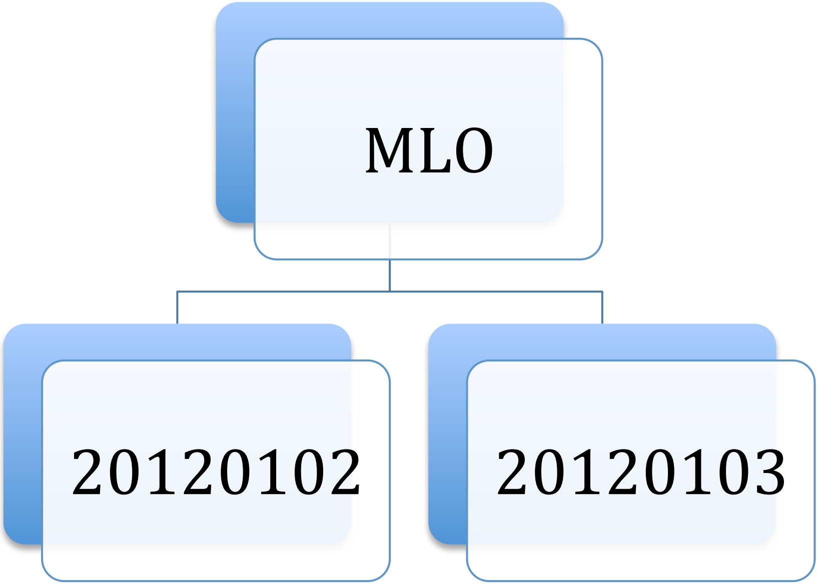

Figure 1 File structure for input data example

2.4 Collecting HITRAN Data

Layer1 assumes that the relevant HITRAN data has already been collected into the hbin file. One must run the hbin.f90 code prior to running Layer1. The hbin program relies on information held within the sfit4.ctl file and thus the program must be called from the folder the file.

3 Layer1

Layer1 allows for batch processing of spectra under the sfit4 framework. Layer1 is controlled by an input file and command line arguments. Each version of layer1

The station location, time period to process, base directories for input output data, various input files, and flags for processing are specifies within the input file. The Layer1 input file is commented to include explanations of each of the variables.

3.1 Layer1 Input File

The Layer1 input file is a python file and therefore must adhere to python syntax convention. The following is an explanation of the variables found in the Layer1 input file:

- loc – The three letter abbreviation for the station location (example: Mauna Loa = mlo)
- iyear,imnth,iday – Year, month, day to begin processing (Inclusive)
- fyear,fmnth,fday – Year, month, day to end processing (Inclusive)
- BaseDirInput – The input base directory for processing. Below this directory Layer1 will expect to find a folder named after the three letter abbreviation for the station location given as the loc variable.
- BaseDirOutput – The output base directory. Layer1 will create output folders under this directory
- binDir – This is the directory where all the sfit4 executable files should be
- ilsDir – This can either point to a folder containing ILS files or point directly to an ils file. If this is an empty string then Layer1 defaults to the ils file specified in the sfit4.ctl file
- RatioDir – Currently not used
- logDirOutput – This is the directory where any log files are written to
- ctlList – Contains three elements. The first element in the list is the location and name of the corresponding sfit4.ctl file. The second element is the filter ID to be used. If this is an empty string, Layer1 filters the spectral database based on micro-windows specified in the sfit4.ctl file. The third element is the version name. A folder with this name is created under BaseDirOutput to which all data is output.
- spcdbFile – This is the path and name of the spectral database file
- WACCMfile – This is the path and name of the WACCM file containing gas profiles. Note that the format of the WACCM file must follow the reference.prf format, i.e. the header for has an 'i5 a8' format where the gas number must be contained within the first 5 columns and the gas name in the following 8 columns. More information can be found in the reference.prf description.
- sbCtlFile – This is the path and name of the sb.ctl file (discussed in error analysis section)
- The explanation of the various flags are commented in the input file

Some files that are necessary for sfit4 to run are not specified in the Layer1 input file. These files are specified through the sfit4.ctl file. They include:

- Station layer file

- Reference profile file – This file is actually created by Layer1 through the refMkrNCAR (NCAR's version of refmaker), is named reference.prf and is put in the same directory as the ctl file. In the sfit ctl file the follow should be used: 'file.in.refprofile = reference.prf'
- Spectrum file – This file is created by Layer1 using the pspec program, is given the name t14asc.4, and is written to the same directory as the sfit4.ctl file. In the sfit4.ctl file the following should be used: 'file.in.spectrum = t15asc.4'
- ILS modulation and phase function files – These files can either be specified through Layer1 or through the sfit4.ctl file depending on the flags set in the Layer1 input file
- Sa matrix file
- Isotope file
- Solar Lines list file
- Linelist file – This file is created by the hbin program and is given a name based on the microwindows described in the sfit4.ctl file.

3.2 Output File Structure

The user specifies the base directory for the output in the Layer1 input file. Within this base directory Layer1 will create a folder with the version name specified in the Layer1 input file. Under the version name folder for each retrieval Layer1 will create a folder with the observations time stamp in the format YYYYMMDD.HHMMSS. These folders are where the sfit4 executable is actually called. The original binary spectra files are copied into these folders.

A folder called inputFiles under the version name folder is automatically created. The original sfit4.ctl, hbin.input, hbin.dtl, and output hbin files are copied to this folder for record keeping purposes. An example of an output data structure is given in Figure 2.

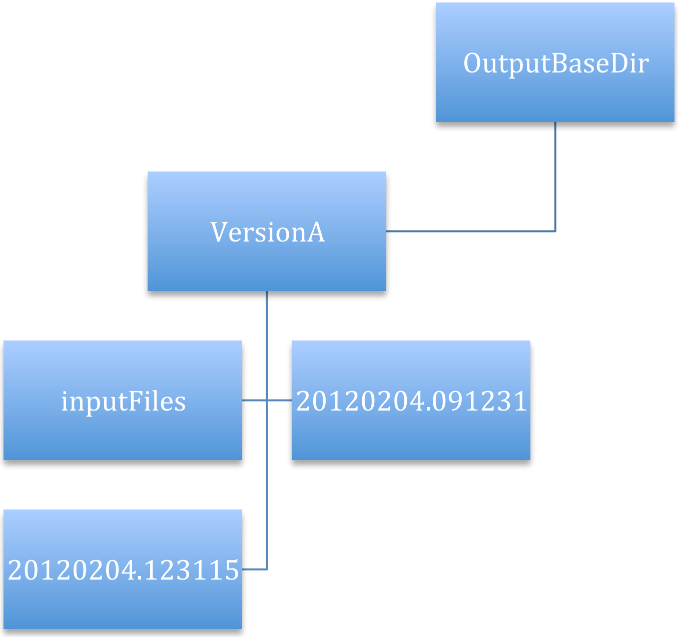

Figure 2 Output File Structure

3.3 Running Layer1

Other than the input file, there are several supporting files necessary to run Layer1. These files reside in the folder ModLib. This folder should be added to the pythonpath environmental variable so the program can access these files.

Layer1 can be run from any directory at the command line once the ModLib folder is added to the pythonpath environmental variable. There are several command line arguments for Layer1 which point Layer1 to the input file, turn off/on logging, and allow for pausing the batch processing.

- -i <file> Flag to specify location and name of layer1 input file
- -l Flag to turn on logging
- -L <0/1> Flag to turn on list-file, where the list-file is a file containing a list of directories processed. The list file is named testing.lst when 0 is used. The list file is named using the date and time stamp (of Layer1 execution) when 1 is used.
- -P <int> will pause the batch processing. If an integer <int> is specified the batch processing will skip n runs, where n = <int> -1 and pause after the n + 1 run.
- -? Shows all flags

The following is an example for running Layer1

```
>sfit4Layer1.py -i /home/data/layer1Input.py –l –L1 –P1
```

This example runs layer1 with the input file layer1Input.py located at /home/data/. It creates a log file and a list file. The list file is named with date and timestamp of when sfit4Layer1.py was called. It runs the first observation found within the specified time period and then pauses for user input.

4 Error Analysis

The error calculations are partially done in sfit4 and completed in supplemental python code. Sfit4 calculates the Jacobian matrix (K) and the Gain matrix (G). It also assembles the a priori covariances (Sa). The python code constructs the instrument covariance matrix (Se), calculates the averaging kernel (AVK), and calculates the systematic, random, and total errors in various units. The errors are calculated as described in Rodgers (2000) and Rodgers and Connor (2003).

The Sb.ctl file is the input file for the error analysis. The Layer1 input file should point to this file and have the appropriate error analysis flag switched on for error analysis to be performed.

4.1 Sb ctl File

The sb.ctl file has several functions including outputs from error analysis, inputs to the error analysis, and definition of the Sb values.

4.1.1 Error Analysis Outputs

Output flags in the sb.ctl file control the output units of the covariance matrices. Output matrices can either be specified in units of VMR or molecules cm^{-2}. Output flags also control which covariance matrices are written (i.e. total, random, and systematic). In addition, one controls the output file names through the sb.ctl file.

4.1.2 Error Analysis Inputs

The input source of the instrument covariance matrix (Se) is determined through the sb.ctl file. There are two options for constructing the Se matrix. The program can read the Se matrix from the sfit4 output file, filename specified in file.out.seinv_vector. This option will take into account the weighting of the SNR done in the sfit4.ctl file. The other option is to construct the Se from the SNR value given in the summary file. These SNR values come directly from the t14acs files and are not influenced by the weighting of SNR in the sfit4.ctl file.

4.1.3 Sb Values

Variance-covariance matrices for parameters not retrieved by sfit4 are constructed through the inputs specified in the sb.ctl file. Currently, only the diagonal elements (the variances) of the variance-covariance matrices are specified in the sb.ctl file. The user should input the values as the square root of the variance. The units for the Sb values are either in native units or fractional. A legend of the units is located at the top of the sb.ctl file. There are several parameters in which the user can either specify native or fractional units, which is controlled by the unit's flags in the sb.ctl file.

The Sbs in the sb.ctl file are grouped according to their dependence. The size of an Sb can depend on number of atmospheric layers, number of micro-windows, a single value, or an individual gas.

There are some important nuances for specifying some of the Sbs in the sb.ctl file.

- Atmospheric layer dependent Sb should be specified in descending order, such that the first value of Sb given corresponds to the top atmospheric layer.
- Micro-window dependent Sbs should be specified in the same order as the micro-windows specified in the sfit4.ctl file parameter 'band ='
- If the apodization and phase function are specified in the forward model (fw.apod_fcn.type and fw.phase_fcn.type = 2 or 3) then the Sbs correspond to each term of the polynomial or Fourier series.. If the apodization or phase function is not included in the forward model the Kb for these is taken as a third order polynomial, therefore three values are necessary for the Sb.
- Both 'phase' and 'phase_fcn' are different ways to describe the same parameter, therefore it is not recommended to calculate an error for both simultaneously.
- The Sbs for zshift are micro-window dependent; however, Sbs should only be specified for micro-windows where the zshift is not retrieved.

5 Plotting and HDF creation

Currently there are no official programs to plot results and convert data to HDF.

Revision History

Rev.	Author	Date	Section	Description
01	Eric Nussbaumer	08-January-2014	All	Initial Release.