

SFit4 Pre-Processing

Ivan Ortega, James Hannigan

NCAR, July 2016



Introduction

- This presentation describes the step-by-step procedures used by the NCAR OT group in the pre-processing of ground-based high resolution spectral data from the sites located at Thule Greenland, Mauna Loa, Hawaii and Boulder Colorado.
- The necessary tools (mostly python and IDL scripts) are introduced.
- These processing tools can be applied (with hopefully minimal changes) to any similar data set.
- The Pre-Processing is an essential part prior to running SFIT4

Where to find the tools?

1) sfit-processing-environment

- Location:

Inside NCAR LAN: <https://git.ucar.edu/?p=sfit-processing-environment.git>

Outside NCAR LAN: <https://proxy.git.ucar.edu/?p=sfit-processing-environment.git>

- Description: This repository contains codes for pre and post processing sfit4 retrieval data. These codes also contain the error analysis for sfit4.
- Permissions: This repository is global read with write permissions only to registered users.

2) SFIT4 Wiki

- Location:

<https://wiki.ucar.edu/display/sfit4/Infrared+Working+Group+Retrieval+Code%2C+SFIT>

General overview of Pre-Processing

1) Initial quality check of all spectra

2) Initial spectral database

3) Prepare house and external station data (weather data)

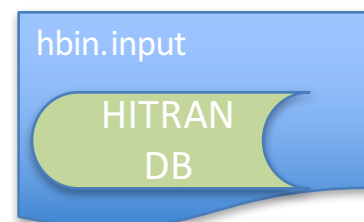
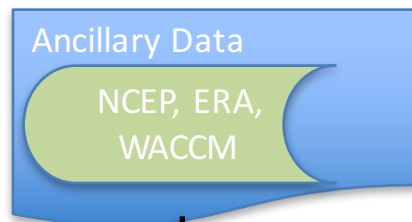
4) Append initial spectral data base with weather data

5) ZPTW Profiles (WACCM, NCEP, ERA-Interim)

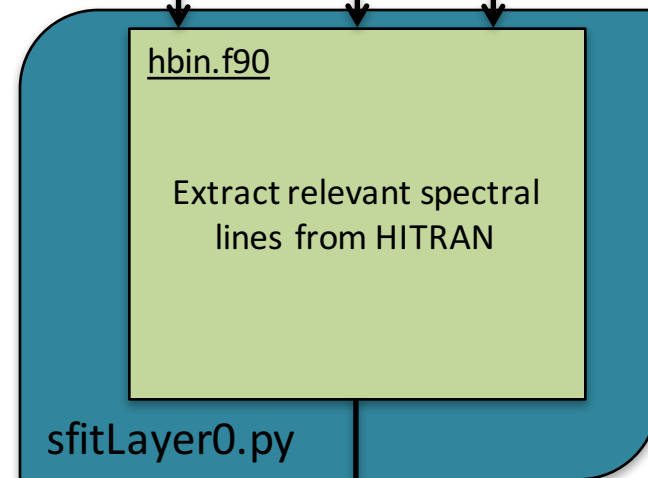
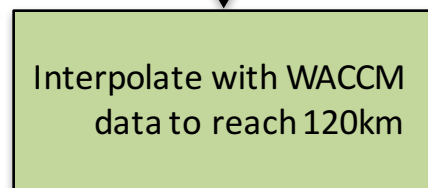
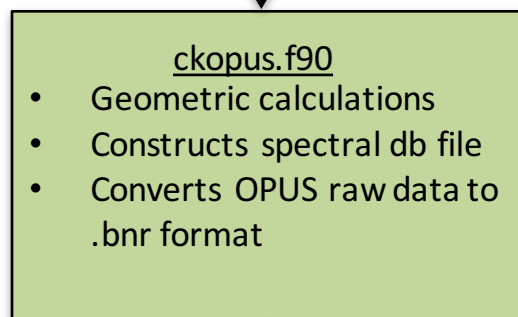
6) Extract relevant spectral lines from HITRAN/ATM

Input and Output Flow for Pre-Processing

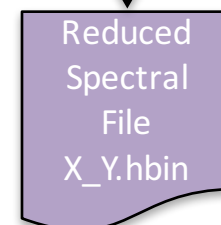
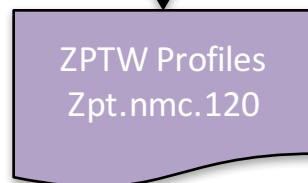
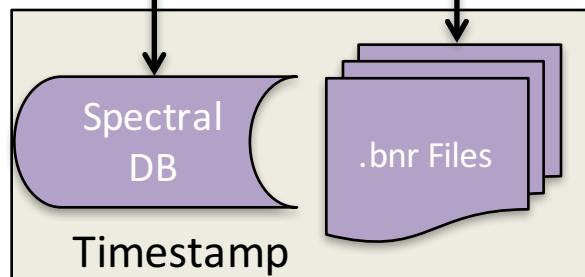
Inputs



Processes



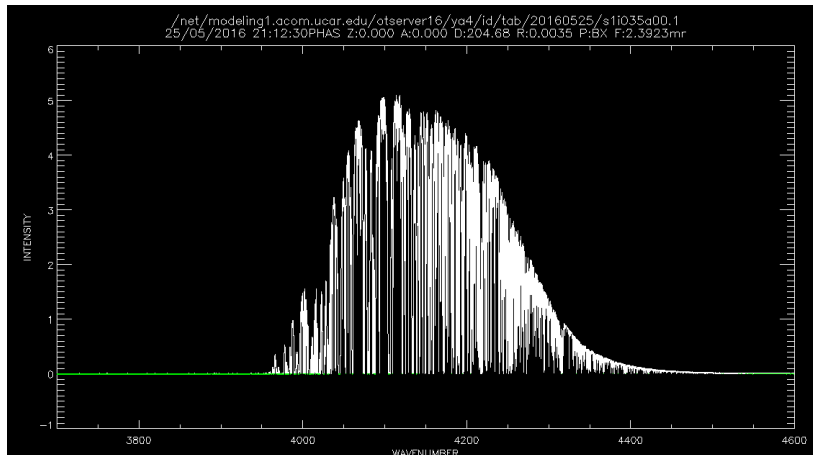
Outputs



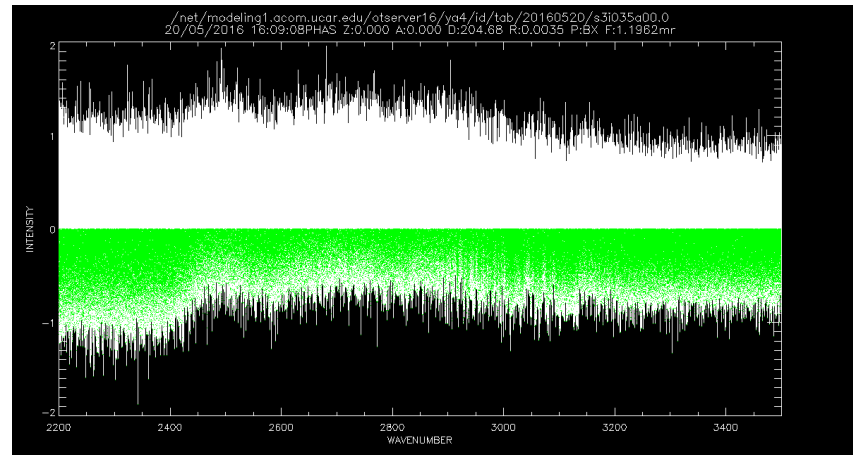
1) Initial quality check

An initial quality check on the spectrum is done using the IDL program `ckop.pro`.

Good



Bad



Would you like to change the x range (p,d,n,f,q,y) : d

```
% FILE_DELETE: Removed file:
/net/modeling1.acom.ucar.edu/otserver16/ya4/id/tab/20160520/s6h035a00.0.
```

- Note: 1) The `OPUSREADERPATH=/data/bin` needs to be set in the environment variables
2) Some changes need to be implemented for other sites

2) Initial spectral database

A python program (mkSpecDB.py) is used to manage the creation of the initial spectral database file:

Program	Description
mkSpecDB.py	Main program to create initial spectral database
specDBInputFile.py	Input file for mkSpecDB.py program

There are logical flags which control the creating of a file which list the folders processed and whether bnr files are created. The output files have the names spDB_loc_YYYY.dat.

Example:

Filename	Site	SBlock	TOffs	TStamp	Date	Time	SNR	N_Lat	W_Lon	Alt	SAzm	SZen	ROE	Dur	Reso	Apd	FOV	LWN	HWN	Flt	MaxY	MinY	FLSCN	EXSCN	GFW	GBW
s1fml1a.0	MLO	SNGC	0.0135	211530	20160101	21:15:30	0.0	19.54	155.57	3396.0	337.21	45.86	6348.2510	204.69	0.0035	BX	1.9139	3899.999	4349.998	1	7.773e+00	-2.333e-02	2	2	1	1
s2fml1a.0	MLO	SNGC	0.0135	211935	20160101	21:19:35	0.0	19.54	155.57	3396.0	338.42	45.50	6347.6927	204.69	0.0035	BX	1.9139	2849.999	3599.996	2	3.924e+00	-1.619e-02	2	2	1	1
s3fml1a.0	MLO	SNGC	0.0135	212343	20160101	21:23:43	0.0	19.54	155.57	3396.0	339.66	45.15	6347.1448	204.68	0.0035	BX	1.1962	2350.000	3199.997	3	1.734e+00	-1.317e-02	2	2	1	1
s4fml1a.0	MLO	SNGC	0.0135	212757	20160101	21:27:57	0.0	19.54	155.57	3396.0	340.96	44.81	6346.6041	204.69	0.0035	BX	1.9139	1849.999	2699.996	4	4.497e+00	-1.265e-02	2	2	1	1
s5fml1a.0	MLO	SNGC	0.0135	213205	20160101	21:32:05	0.0	19.54	155.57	3396.0	342.24	44.50	6346.0987	204.69	0.0035	BX	3.5885	1749.999	2099.998	5	1.251e+01	-2.121e-02	2	2	1	1
s6hfm1a.0	MLO	SNGC	-0.0149	210501	20160101	21:05:01	0.0	19.54	155.57	3396.0	334.19	46.88	6349.7453	0.00	0.0035	BX	4.0670	699.999	1349.996	6	2.527e+00	1.575e-03	2	4	1	1
s9ifml1r.0	MLO	SNGC	0.0135	213625	20160101	21:36:25	0.0	19.54	155.57	3396.0	343.60	44.20	6345.5956	204.69	0.0035	BX	1.1962	4199.999	5299.997	9	9.220e-01	-8.305e-03	2	2	1	1
saifml1r.0	MLO	SNGC	0.0135	214029	20160101	21:40:29	0.0	19.54	155.57	3396.0	344.89	43.94	6345.1507	204.69	0.0035	BX	1.1962	3800.000	4799.998	A	9.868e-01	-1.094e-02	2	2	1	1
s1fml1a.0	MLO	SNGC	0.0135	174437	20160102	17:44:37	0.0	19.54	155.57	3396.0	298.46	80.74	6371.8685	204.69	0.0035	BX	1.9139	3899.999	4349.998	1	6.346e+00	-3.613e-02	2	2	1	1

3) Prepare house data

House data is data that is recorded by the FTS autonomous system, such as outside temperature, pressure, wind direction, etc. A python program (station_house_reader.py) is created to read the various formats and create a standardized file.

Program	Description
station_house_reader.py	Main program to read house data files
HouseReaderC.py	Supporting program with formats of previous house data files

There are logical flags which control the creating of a file which list the folders processed and whether bnr files are created. The output files have the names spDB_loc_YYYY.dat.

Example:

Filename	Site	SBlock	TOffs	TStamp	Date	Time	SNR	N_Lat	W_Lon	Alt	SAzm	SZen	ROE	Dur	Reso	Apd	FOV	LWN	HWN	Flt	MaxY	MinY	FLSCN	EXSCN	GFW	GBW
s1fml1a.0	MLO	SNGC	0.0135	211530	20160101	21:15:30	0.0	19.54	155.57	3396.0	337.21	45.86	6348.2510	204.69	0.0035	BX	1.9139	3899.999	4349.998	1	7.773e+00	-2.333e-02	2	2	1	1
s2fml1a.0	MLO	SNGC	0.0135	211935	20160101	21:19:35	0.0	19.54	155.57	3396.0	338.42	45.50	6347.6927	204.69	0.0035	BX	1.9139	2849.999	3599.996	2	3.924e+00	-1.619e-02	2	2	1	1
s3fml1a.0	MLO	SNGC	0.0135	212343	20160101	21:23:43	0.0	19.54	155.57	3396.0	339.66	45.15	6347.1448	204.68	0.0035	BX	1.1962	2350.000	3199.997	3	1.734e+00	-1.317e-02	2	2	1	1
s4fml1a.0	MLO	SNGC	0.0135	212757	20160101	21:27:57	0.0	19.54	155.57	3396.0	340.96	44.81	6346.6041	204.69	0.0035	BX	1.9139	1849.999	2699.996	4	4.497e+00	-1.265e-02	2	2	1	1
s5fml1a.0	MLO	SNGC	0.0135	213205	20160101	21:32:05	0.0	19.54	155.57	3396.0	342.24	44.50	6346.0987	204.69	0.0035	BX	3.5885	1749.999	2099.998	5	1.251e+01	-2.121e-02	2	2	1	1
s6hfm1a.0	MLO	SNGC	-0.0149	210501	20160101	21:05:01	0.0	19.54	155.57	3396.0	334.19	46.88	6349.7453	0.00	0.0035	BX	4.0670	699.999	1349.996	6	2.527e+00	1.575e-03	2	4	1	1
s9ifml1r.0	MLO	SNGC	0.0135	213625	20160101	21:36:25	0.0	19.54	155.57	3396.0	343.60	44.20	6345.5956	204.69	0.0035	BX	1.1962	4199.999	5299.997	9	9.220e-01	-8.305e-03	2	2	1	1
saifml1r.0	MLO	SNGC	0.0135	214029	20160101	21:40:29	0.0	19.54	155.57	3396.0	344.89	43.94	6345.1507	204.69	0.0035	BX	1.1962	3800.000	4799.998	A	9.868e-01	-1.094e-02	2	2	1	1
s1fml1a.0	MLO	SNGC	0.0135	174437	20160102	17:44:37	0.0	19.54	155.57	3396.0	298.46	80.74	6371.8685	204.69	0.0035	BX	1.9139	3899.999	4349.998	1	6.346e+00	-3.613e-02	2	2	1	1

3) Prepare external station data (weather data)

There are currently two external station data sources used (EOL for FLO, and CMDL for MLO) only the EOL data needs to be pre-processed. The original format of this data is in netcdf files. The program `read_FLO_EOL_data.py` reads the daily netcdf files and creates a yearly text file

Program	Description
<code>pullAncillaryData.py</code>	Program to automatically pull EOL and CMDL data
<code>read_FLO_EOL_data.py</code>	Main program to read EOL and CMDL data

4) *Append initial spectral data base with weather data*

A python program was created to append the initial spectral database with the house and external station weather data.

Program	Description
appendSpecDB.py	Program to create the append spectral database file
appndSpecDBInputFile.py	Input file for appendSpecDB.py

The program reads in the initial spectral database file. It then searches the house and external station files for weather data at the time of observation, plus a certain number of minutes specified by the user. The mean of the data collected is calculated and a new spectral database file is created.

The sfit4Layer1 processing looks for a database file with all years. Once you create a year appended spectral database file you should copy or append this to the file which contains all years processed

Coadding spectra?

The program `mkCoadSpecDB.py` co-adds two bnr files together with the appropriate forward and backward scans. A new coadded bnr file is created. The program `mkCoadSpecDB.py` calls the C program `coadd.c` to co-add the files. The program `mkCoadSpecDB.py` requires an input file (`CoadSpecDBInputFile.py`).