EGI Manual

Apr 21, 2017

Created by Jacob Hedelius
jhedeliu@caltech.edu

The EGI suite has been created to make processing of our EM27 data easier and more automated. The EGI software suite is meant to process from interferograms to retrieved DMF using I2S (optional) and GGG by in a simple one-line manner.
If you would like to use PROFFIT for retrievals, contact Frank Hase and or Matthias Frey at KIT.

Contents

Acknowledgements

Many people have contributed to development directly or indirectly. Thanks to Debra Wunch for introducing me to scripting in bash and tcsh, and for generating some basic scripts that I was able to expand to make EM27/SUN retrievals hopefully more user-friendly. Thanks to her and to Geoff Toon for working on making a version of I2S that works for EM27/SUN instruments. Thanks to Camille Viatte for being the first to try EGI and for helping me eliminate many, many bugs and for suggestions features. Thanks to Harrison Parker for working with me on making EGI Mac compatible for the first time and to Jia Chen for testing on Mac, and to both on suggestions of new features. Thanks to Hiroshi Suto-san for feedback on installing on MacOSx Yosemite. Thanks to Elaine Gottlieb, Jonathan Franklin, Hai Nguyen, Niki Jacobs, and Michael Wedrat for bug reporting and feature suggestions.

1. Basic Description

This **E**M27 **G**GG **I**nterferogram processing (EGI) suite simplifies running GGG and I2S with the similar each time. It has been used with 7+ different EM27/SUN spectrometers and by 6+ different institutions. This suite requires 1) raw interferogram or spectral data is divided into daily folders named YYYMMDD as well as 2) daily local meteorological files in an ascii/csv format. After an initial setup it should only require one line to process retrievals from start to finish.

Features include: Creates all necessary input files and runs routines for you, a basic cloud filter, a SNR calculation (for single detector instruments).

2. Required Functionality

This relies on a lot of functionality that seems to be included on most systems. At Caltech this suite is run on ccycle, a 64-processor central computer that is equipped with enough RAID disk arrays to store and write our data. EGI has also been run on local computers (MacOS) as well. If unaveraged interferograms are saved, it takes about 12 hours to run a day worth of data on a Mac.

2.1 Mercurial

Repositories are cloned using hg. Mercurial is also used for the GGG repository, and you will likely use it for I2S for EM27s too. There will be 3 different repositories you will need (GGG2014, IPP-developmental, and EGI). You should already have Mercurial installed on your system. To check type:

```
$ hg help
```

An error message indicates it is not installed. If it is not installed visit
http://mercurial.selenic.com/wiki/Download
to setup/install.

If you are new to Mercurial I encourage you look over this tutorial.
http://hginit.com/

Good practice is to not place one repository within another. There are other features to Mercurial you may want to look into that I am not covering here. I submit EGI updates using this.

2.2 System functionality

Most of these seem to be included on systems – though some users have had trouble with GNU parallel.

- Python or MATLAB must be installed. To check if python is installed run

```
$ python -V
```

An error message indicates it is not installed.

- GNU parallel must be installed. You can check by looking for the manual (q to quit when open):

```
$ man parallel
```

- An internet connection (to get .mod files from Caltech ccycle) or IDL.

- Some specific Linux functions (already installed on most systems) used include :
    - Bash
    - Grep
    - egrep
    - awk
    - sed
    - cut
    - expr
    - date
    - bc
    - md5sum
    - Fortran compiler (used for GGG and I2S)

Note that sed and date are different for MacOS and Linux systems, but EGI should use the appropriate syntax based on your OS.

2.3 GGG and I2S installed and paths set

2.3.1 GGG2014, and default I2S

- GGG must be installed (see the TCCON wiki: https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2014_Release_Notes). Use the "clone" version to get GGG for the first time (https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2012_Release_Notes#.22Clone.22_method). GGG and I2S need to be compiled too (https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2014_Release_Notes#Install).

  Note: If you run GGG for other purposes (e.g. TCCON) there are a few small differences. They should be compatible, except you lose one small feature (a check for spectra too temporally close).

- You can compile the default I2S too (see the TCCON wiki: https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2014_Release_Notes/I2S_Release_Notes). This is not strictly needed because the ipp_double_sided developmental version is more compatible with EM27/SUN instruments.

2.3.2 ipp_double_sided developmental version installation

This is the program to run FFTs on interferograms to obtain spectra. This is required even if you use OPUS spectra because EGI uses I2S utilities.

Notice: The default GGG2014 version of I2S is not compatible with EM27s. In the meantime this is a pre-release of the GGG2017 (GGG2018?) version. Thanks to Debra and Geoff who have provided early access.

Note: This is not currently compatible with midIR-EM27s (inputs are different).

1) Create a new directory you would like to put this repository in (good practice is to not put one repository within another repository). E.g.

```
$ mkdir DIRNAME
```

2) Clone the repository from bitbucket to the directory you just created. Password is: co2ch4co

```
$ hg clone
https://jhedeliu@bitbucket.org/em27ggg/i2s_double_developmental
DIRNAME
```

3) Compile I2S by running ./`master.sh` (see https://tccon-wiki.caltech.edu/Software/IPP/IPP_2014_Release_Notes)

2.3.3 set system paths "gggpath" and "i2spath" (and "matlabpath" for MacOS)

The paths $gggpath and $i2spath (all lower-case) need to be set your GGG and I2S paths respectively. Point i2spath to the ipp_double_sided developmental version.
See https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2012_Release_Notes#Installation_Instructions for details on how to. This varies with OS, but some examples that work on our Linux system are:

Ex. b-shell: add lines to .bash file "`export GGPATH=/home/toon/ggg`" (then `source ~/.bash`)
Ex. c-shell: add lines to .cshrc file "`setenv GGGPATH /home/toon/ggg`" (then `source ~/.cshrc`)
Note : "$gggpath" and "$i2spath" are lower case!

To see how this was set up on a MacOS, see appendix A. If running with MATLAB (optional) on a Mac, $matlabpath must be set as well. E.g.: '/Applications/MATLAB_R2014b.app/bin'

To check these are set, you can type something like

```
$ echo $gggpath
```

3. EGI Installation Instructions

To get the EGI code you can clone the repository. You should have set a path that is not within another repository to clone the EGI suite to.

3.1 Clone the repository (first time only)

To clone (download) the repository for the first time, pick a local directory (LOCAL_DIR) to save it to. Enter:

```
hg clone https://jhedeliu@bitbucket.org/em27ggg/egi2014
LOCAL_DIR
```

You will be prompted for a password (co2ch4co)
This should add the files to LOCAL_DIR.

3.2 Future changes (skip this the first time)

You can see when changes have been made to the files by navigating to your LOCAL_DIR and entering the command

```
hg incoming
```

To pull (download) these changes enter

```
hg pull
```

Run `hg update` to get a working copy.

If you ever want or need to revert to a previous version you can always revert everything or even just specific files.

3.3 Python users (if not running MATLAB)

(Note: I changed this process in early 2017). Do this before attempting to run anything else.

If you would like to run using python (instead of MATLAB) navigate to "univ_vars" and swap line (~33) from:

```
matlaborpython=0 #0 = matlab, 1=python
```

to

```
matlaborpython=1 #0 = matlab, 1=python
```

3.4 Running the setup_test

EGI does not require compilation, but there are some GGG files that need to be modified, so the benchmark run should be completed.

This test runs through some example representative Caltech interferograms and spectra. These data represent measurements taken under a variety of conditions (wetter/drier, lower/higher airmass, warmer/cooler, higher/lower signal). To run, navigate to where you have your egi repository and run

```
./setup_test
```

You will be asked 4 input questions. Here's an example of how it looks when you run it and the most likely/common answers for 1–3.

```
[jhedeliu@ccycle ~/working_egi]$ ./setup_test
EGI test
IPP-double-sided installed? (y/n):y
IDL installed? (y/n):n
Number of processors to use:1
Create directories and a file_loc.lst for your instrument (first time
users only)? (y/n):
```

Question 4 asks about making a file named "`file_loc.lst`" (one required per EM27/SUN instrument). You can also make your own based on examples, more details in Sect. 3.6. The setup test can make directories for you to put data in and a `file_loc.lst`. It also makes required `$gggpath/tccon` files for you ("`xx_qc.dat`", "`xx_sunrun.dat`", and "`xx_oof_header.dat`"). More details on this feature in Sect. 3.5.

The `setup_test` will make some common edits and additions to the ggg scripts that are useful for EM27 instruments (i.e. adds `bin2asc_wlims`, modifies `collate_results`, modifies `write_official_output_file`, and modifies `get_opusigram_params`). If you have previously edited a file in GGG, EGI will not make the corrections until you commit your changes in GGG. This may cause you problems later otherwise.

Finally, `setup_test` will attempt to run some files through full processing. This takes ~10–20 minutes. This is useful to make sure you have all dependencies. If successful, you will get this file:

```
results/cumulative/cn20140613_20141021/cn20140613_20141021.vav.ada.aia
.oof.csv
```

It also compares with benchmark results, but the benchmark outputs have some problems so don't expect them to match perfectly.

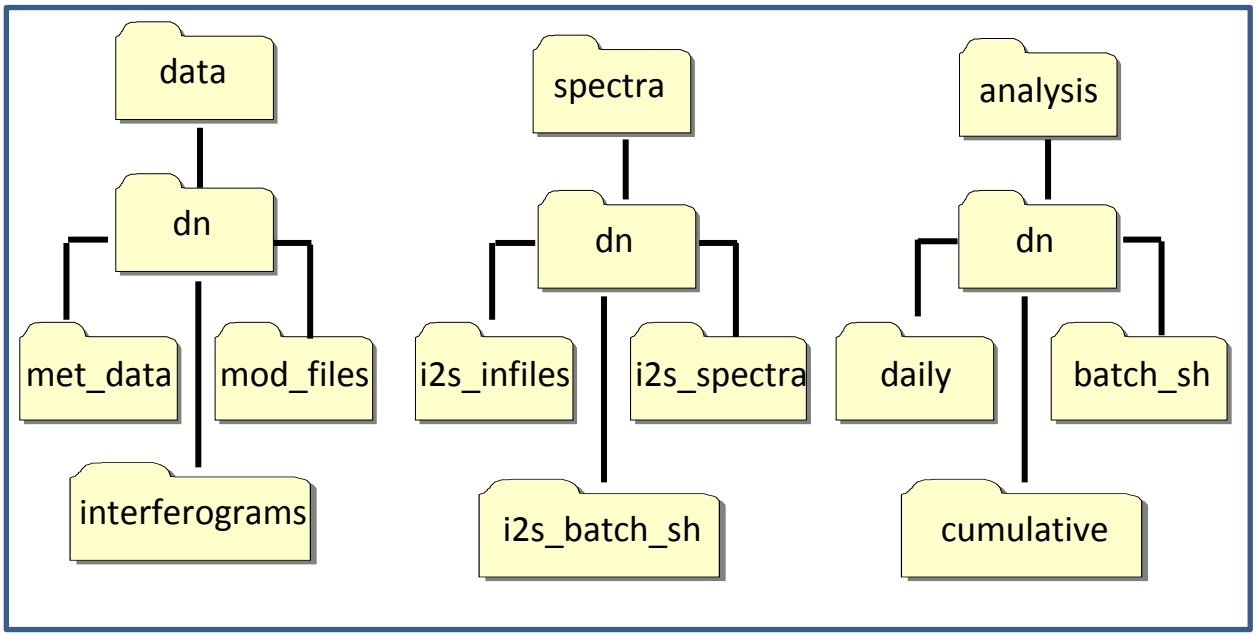
3.5 setup_fileloc

This script is run when "y" is answered to question 4 in the setup_test.

Here's how it looks (3 questions in bold):

```
file_loc.lst creator,
this assumes you have a standard EM27 instrument.
if you have a wide-band detector you will need to change
parameter 14 (windows used) to em27wb.gnd
Instrument 2 letter id:dn
Lat,long,altitude_masl (no spaces). e.g. 34.14,-118.13,237:34.14,-
118.13,237
Directory location, or 0 for default (one level up, em27_data):0
Freq limit file: /home/jhedeliu/ipp_double_sided/opus-
ipp/flimit_EM27.ipp already found
Your file_loc.lst has been created, please see instructions in that
file on where to place data
Now creating some TCCON files for your instrument if they do not yet
exist
though you really should go through and check/modify these yourself
too
```

It creates the following directory structure (in my case, in `/home/jhedeliu/em27_data`).



It also creates files "`xx_qc.dat`", "`xx_sunrun.dat`", and "`xx_oof_header.dat`" in `$gggpath/tccon` which you may want to check (see Sect. 3.7). It also creates a `file_loc.lst` for you.

3.6 Create file_loc.lst (if not done with setup_test, Sect. 3.4/3.5)

You will need to set your file locations in a file_loc.lst file, following the format in one of the example files included in the repository. Below is an example of a file_loc.lst file. The line numbers should be exactly the same in your file. You should have a space (NOT tab) after your parameters. After the space you may put in comments (optional). You will need to change/save a different file_loc.lst for each instrument. At Caltech we treat our EM27 as 2 instruments when we are on campus (dn) and off (nn). We have a different file_loc.lst for each.

1	This contains a list of all the relevant file directory locations.
2	Note: the order must not change. There are five lines of header before the directory locations.
3	Requires a space after each entry. $gggpath and $i2spath should already be appropriately set
4	Edit your example .in file to match your instrument
5	
6	cn : (1) inst 2 letter ID
7	/oco2-data/tccon/data/caltech_em27/cn_sun/ : (2) raw interferogram directory (separated in daily folders), or enter 0 if using OPUS spectra
8	/oco2-data/tccon/spectra/caltech_em27/cn_sun/i2s_infiles/ : (3) output dir for inputs to IPP, including .in files, etc.
9	/oco2-data/tccon/spectra/caltech_em27/cn_sun/ : (4) If using I2S where spectral files will go. If using OPUS, where spectra are at.
10	/oco2-data/tccon/data/caltech_em27/Retrieved_Caltech_Zeno/ : (5) Where weather station meteorological .txt files are at
11	34.14,-118.13,237 : (6) Location files or coordinates (enter lat,long,masl seperated by comma, no spaces!)
12	/home/jhedeliu/ggg/supp_gggcode/automating/em27gi/opus-ipp.cn_caltech_em27.top : (7) I2S only - An original .top file that can be used in creating .in files
13	/oco2-data/tccon/spectra/caltech_em27/cn_sun/i2s_batch_sh/ : (8) location to put batch files for use with I2S
14	/oco2-data/tccon/analysis/caltech_em27/cn_sun/daily/ : (9) Main output directory where runlogs, and final results go. In daily YYYYMMDD folders.
15	s0e00a : (10) spectral naming convention SIBDOD
16	gnd : (11) set as gnd for ground
17	/oco2-data/tccon/analysis/caltech_em27/cn_sun/ batch_sh/ : (12) Batch files for running GFIT
18	/oco2-data/tccon/analysis/caltech_em27/cn_sun/cumulative/ : (13) where cumulative outputs go
19	/home/jhedeliu/ggg/windows/gnd/em27.gnd : (14) windows used
20	0 : (15) indicator if averaging kernels should be calculated.
21	0 : (16) indicator if all spectral fits should be saved.
22	1 : (17) Indicator for obtaining .mod files. 1=IDL, list directory path, or use “caltechccycle”
23	0 : (18) wavenumber to split at if multiple detectors are used (set to 0 or blank if using standard single detector)

Detailed parameter descriptions:

(1) 2 letter ID. See https://tccon-wiki.caltech.edu/Software/GGG/TCCON_Spectral_Naming_Convention#Current_list_of_two-letter_identifiers for currently used TCCON identifiers. EM27/SUN identifiers I've seen include:

cn	Caltech, on campus, ext. detector	pl	LANL in SoCal
mn	Caltech, off campus, ext. detector	al	LANL at AFRC
dn	Caltech, on campus, stnd. detector	Ok	LANL at Lamont
nn	Caltech, off campus, stnd. detector	wi	LANL at Wisconsin
ha	Harvard 1	mu	TUM, 2 detectors
hb	Harvard 2	ma	TUM
ka	KIT at Indianapolis	jz	JAXA
ak	LANL in Alaska	jm	JAXA

(2) Location for raw incoming data if using interferograms (set as 0 for OPUS spectra). Data should be organized in daily folders named "YYYYMMDD" (e.g. 20140602 for June 2, 2014). See Sect. 4 for interferogram naming convention

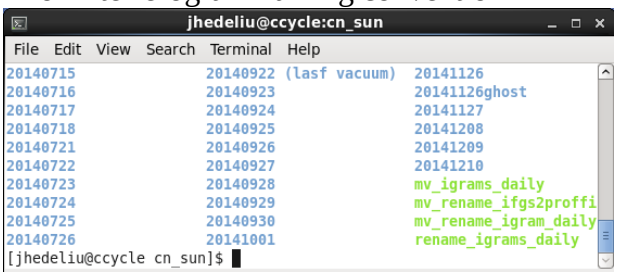


Figure Accepted folder options and files named as ifgs. Note that the dates that are not exactly in YYYYMMDD (e.g. "20141126ghost") format are excluded upon analysis.

(3) Output dir for various inputs prior to running I2S/IPP (daily subfolders). Required even if using OPUS spectra Outputs include .in files, day (combined) zeno/weather, day (combined) GPS if applicable, list of interferograms, and interferogram name+internal time+signal files.

(4) For I2S, generated spectra go here in monthly subfolders. For OPUS generated spectra, this should have daily subfolders containing the spectra (see Sect. 4.1 for naming convention).

(5) Directory containing all meteorological data text files. See Sect. 4.3 for naming convention and file format.

(6) Location files or coordinates. Coordinates should be latitude, longitude, altitude (m above sea level) separated by commas with no spaces in-between. If you make

measurements with your instrument at multiple locations I recommend using a "GPS" data text files to keep track of where your instrument was on various days (similar to the meteorological data text files). See Sect. 4.4 for naming convention and file format. Using "`none`" defaults to Caltech coordiantes.

(7) A .top file is required if using I2S (but not OPUS). Generally the default file is acceptable for EM27s using a standard detector. EGI should correctly parameterize incoming and outgoing file directories, UTC time offset, and the instrument id/outgoing naming convention.

(8) Directory where batch files for running I2S go.

(9) Main output directory where sunruns, runlogs and other files go (in sub-folders named YYYYMMDD/runlogs). Other output files go in the YYYYMMDD folder (e.g. *.oof.csv files).

(10) Naming convention with spectra, Source to detector. See https://tccon-wiki.caltech.edu/index.php?title=Software/GGG/TCCON_Spectral_Naming_Convention For standard EM27/SUN instruments this is "`s0e00a`", for instruments with dual detectors (the CO channel) this is "`s0e00a`", for instruments using a mid-IR InSb detector this is "`s0e00c`".

(11) Set as "`gnd`" (currently the only supported option).

(12) Directory to put batch files for running groups of GGG and EGI related routines (besides I2S). Separates outputs into grouped directories.

(13) Directories where cumulative outputs go (concatenated across multiple days). Output subdirectories named "xxYYYYMMDD_YYYYMMDD" where the 2 dates are the start and end dates.

(14) File that lists the windows used by GFIT. In the directory setup_examples_jkh are some examples. You can use "`em27.gnd`" with the standard InGaAs detector, and "`em27wb.gnd`" with the extended-band detector, or if using 2 detectors.
You can adapt "`mir_windows_wip.gnd`" for the MID-IR instruments, note that this is a work-in-progress.

(15) Indicator if averaging kernels should be calculated. 0=no, 1= create directories only, 2=yes

(16) Indicator if all spectral fits should be saved. 0=no, 1= create directories only, 2=yes

(17) How to obtain .mod files. Indicate if IDL is installed with 1=yes, 0=no (.mod files in `$gggpath/models/gnd/`), list a directory where .mod files are to copy over from, or use the option `caltechccycle` to give EGI permission to automatically upload runlogs to the Caltech ccycle server and pick up .mod files via ftp.

(18) Wavenumber to split at if multiple detectors are used (set to `0` if using standard single detector). A value of `5300` is a good choice if using the dual detectors with the CO channel.

3.7 Create/modify/check instrument specific files

If you have not been running GFIT/I2S yet manually for your instrument, you may want to check some tiles.

Running setup_test (Sect. 3.4) first which will copy cn versions of these files to their correct locations. Running the optional setup_fileloc (Sect. 3.5) will create default files for your instrument. You can edit/rename the copied files.

1) If you use 2 detectors and I2S, you will want to use a .top file similar to the `setup_examples_jkh/example_2detector.top`. (Parameter #7 in the file_loc.lst.) Most .top/.in file parameters will be changed for you (including file paths, UTC time correction, and site ID).

2) Sunrun parameters, file: **$gggpath/tccon/xx_sunrun.dat** where the "xx" matches your site id. You can change parameters you deem necessary (e.g. a pressure correction).

3) QC file for write_official_output, file: **$gggpath/tccon/xx_qc.dat dat**. Set the quality control parameters (what gets flagged).

4) OOF header for write_official_output, file: **$gggpath/tccon/xx_oof_header.dat**. Create your own header for your oof files containing what you like that matches your instrument policy. This is written to the top of all these files with the retrieved values below.

4. Incoming Data Formats

At a minimum you will need interferograms (recommended, 4.1) or spectra (4.2). You also need meteorological files (4.3). GPS files (4.4) are optional, but recommended if you move your instrument frequently.

4.1 Interferograms – generating I2S Spectra

If I2S (Caltech/JPL developed FT software) is being used (recommended) then raw data format should be in opus interferogram format with the interferograms in daily folders. Files should be named as xxYYYYMMDD.ifg.nnnn format

xx	unique 2 letter site identifier
YYYY	year (e.g. 2014)
MM	month (e.g. 06)
DD	day (e.g. 31)
nnnn	file number/scan number of the day (e.g. 0001)

Some examples are:
cn20140528.ifg.0002, mn20140830.ifg.0130, cn20140701.ifg.1347

Spectra will be written and later looked for in monthly (YYYYMM) folders by GGG.

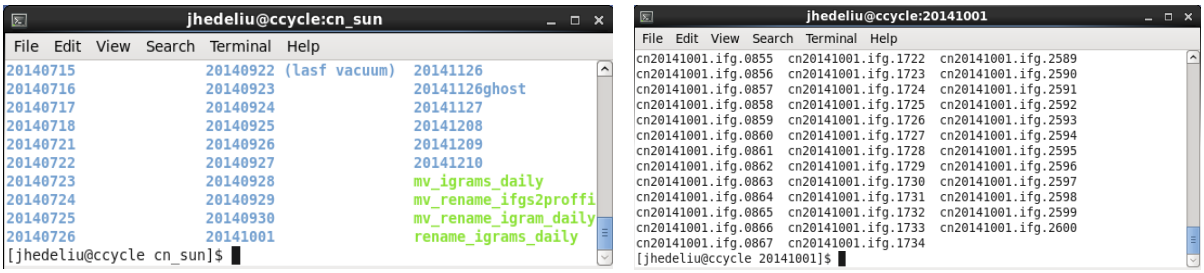


Figure Accepted folder options and files named as ifgs. Note that the dates that are not exactly in YYYYMMDD (e.g. "20141126ghost") format are excluded upon analysis.

If you are acquiring data with and EM27/SUN, I highly recommend setting up your XPM files to create this correct naming convention for you. You may also consider modifying and using the files in `utilities/opus_macros` for use with your instruments.

4.2 OPUS Spectra (not req'd if using I2S)

You can directly use OPUS spectra if you do not want to run I2S. Spectra should named in a TCCON format. https://tccon-wiki.caltech.edu/index.php?title=Software/GGG/TCCON_Spectral_Naming_Convention

Examples: cn20140709s0e00a.1247, mn20140830s0e00a.0261, pl20140424s0e00a.0005.

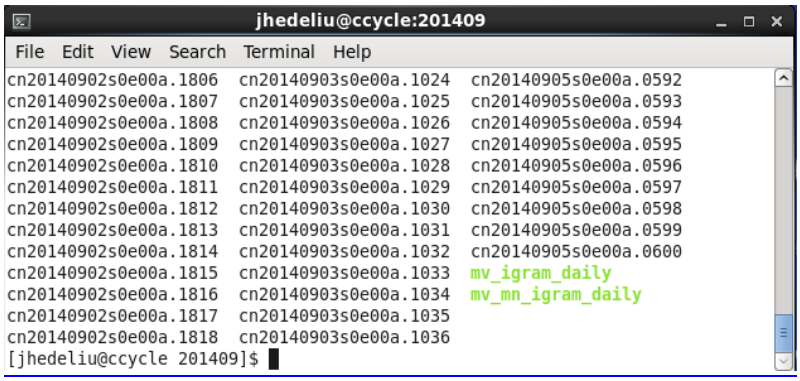


Figure Accepted file name options if using spectra generated directly from OPUS. Note that files are still divided into daily folders.

4.3 Weather/Meteorological Station Data

Weather data should be in a plain text delimited (comma or space) file. There must be time data, either UTCDate (format YY/MM/DD or YYYY/MM/DD) & UTCTime (format HH:MM:SS) or CompSrlDate (example 735854.84046) for time (CompSrlDate takes precedence). Time in the UTCDate/UTCTime should be in UTC (GMT+0) time. CompSrlDate should match the time on the acquisition computer. It also requires headers and data of Temp (or Tout) in C, RH in %, Pout in hPa/mbar. WSPD, and WDIR are optional headers. These files need to be named `YYYYMMDD*.txt`, where "*" is a wildcard and can be anything. These files currently can only have one row for the headers. They also should only contain numbers below the header line. For example files look in the directory: `example_data/zeno_weather/`

Example:

```
CompSrlDate, Unit, UTCDate, UTCTime, WSPD, WDIR, SigTheta, Gust, Temp, RH, SFlux, Pout, Precip, LeafWet, Battery, Bit,
736005.73038, 4449, 2015/02/10, 18:04:46,  0.0,   0,  0.0,  0.0, 19.9,  46,    0.0,  985.9, 0, 15, 13.7,      0,
736005.73041, 4449, 2015/02/10, 18:04:48,  0.0,   0,  0.0,  0.0, 19.9,  46,    0.0,  985.9, 0, 19, 13.7,      0,
736005.73043, 4449, 2015/02/10, 18:04:50,  0.0,   0,  0.0,  0.0, 19.9,  46,    0.0,  985.9, 0, 19, 13.7,      0,
736005.73045, 4449, 2015/02/10, 18:04:52,  0.0,   0,  0.0,  0.0, 19.9,  46,    0.0,  985.9, 0, 15, 13.7,      0,
```

4.4 GPS Data (Optional)

You will have the option to include a static GPS location, or to have text files containing locations (this is very useful if you move around during a given day). These are similar to the Meteorological data files. Time is needed (CompSrlDate or UTCDate + UTCTime) as well as Lat and Long.

Example:

```
CompSrlDate, GPSUTCDate, GPSUTCTime, Lat, Long, Quality, Flag, Satellites, HDOP, masl_inacc, geoidal_WGS84_sep, s_since_update,
ref_stat_ID, checksum, spd_ms, dir, magvar,
735988.88479, 2015/01/24, 21:14:08,  34.004343, -117.653770, 2, 0, 11,   0.9,  223.1,  -33.2,    3, 0000, 46,   0.0,   0.0,   NaN,
735988.88480, 2015/01/24, 21:14:09,  34.004343, -117.653770, 2, 0, 11,   0.9,  223.1,  -33.2,    2, 0000, 46,   0.0,   0.0,   NaN,
735988.88481, 2015/01/24, 21:14:10,  34.004343, -117.653770, 2, 0, 11,   0.9,  223.1,  -33.2,    2, 0000, 4E,   0.0,   0.0,   NaN,
```

If you have a MATLAB license, you can create "synthetic" GPS files using the script `utilities/sythetic_gps_file.m`. These are useful if you move from day to day so you can process multiple days of data at once.

5. How to Run – One line entry

I suggest looking into the **screen utility** before processing large datasets on a Linux system.

The main file is egi_full. The syntax is to enter (after navigating to folder):
./egi_full
Followed by the start date (format YYYYMMDD) end date (format YYYYMMDD) and maximum number of processors to be used (default is 1). Some example input arguments:

`20140602`	Processes all data from June 2nd on 1 processor
`20140602 15`	June 2nd data on 15 processors
`20140602 20140630 15`	All data from June 2nd to 30th on 15 processors
	An empty input will bring up a prompt where you specify start and end dates. Uses 1 processor.

```
./egi_full 20140602 20140831 40
```

Wait for the code to complete, and that's it to get daily output files!

If you want to get cumulative files (recommended, see https://tccon-wiki.caltech.edu/Software/GGG/FAQs, look for average_results) you also need to run egi_cumulative

```
./egi_cumulative 20140602 20140831 40
```

This script also requires a cumulative_exclude.lst that includes dates and specific spectra (or rather line-numbers) to exclude. For example
20140528
20140529:1,2,30-50
Excludes any data from May 28, 2014 and excludes spectra (rather line-numbers) 1, 2, and 30 through 50 on the 29th.

6. Other info available

I also have instructions for the following (available upon request, later I may implement in this manual)

- Automatically uploading/backing up data (iterferograms, etc.) from acquisition Windows computer to server
- Additional time syncs for windows
- Setting up automatic uploads to Google drive (e.g. text files) from linux system
- Setting up IP camera connected to acquisition computer via router to take and save pictures
- Minimal instructions on connecting EM27/SUN and acquisition computer to router (so it can also be connected to Ethernet).

7. AOD estimates (work in progress)

I have begun work on trying to estimate AOD from EM27/SUN instruments, likely to be presented at the 2017 TCCON meeting. I have a draft of a paper on this. Results may not be reliable depending on your situation. I only have MATLAB versions currently, but do intend to make equivalent python scripts in the future. If you would like to test this (with the caveat that these estimate are very low accuracy), navigate to `aod_calc`. Then run `em27aodest` with similar inputs as before:

```
./em27aodest 20140602 20140831 40
```

8. Some descriptions of sub-routines

egi_full
The script egi_full runs all sub-scripts. If you only want to run some of the scripts, you can comment out the ones you don't want with # or you can use the “exit” command to stop the code prematurely.
The sub-scripts can each be run independently as well. (See Flowcharts).

datein_sub
Takes your inputs of start and end-dates and processor numbers and makes any fixes if necessary (e.g. start/end date backwards, or only one date given, or no date given). This script also prevents you from using more than 40 processors (this can easily be changed). Some example inputs this allows are:

20140602	Processes all data from June 2nd on 1 processor
20140602 15	June 2nd data on 15 processors
20140602 20140630 15	All data from June 2nd to 30th on 15 processors
	An empty input will bring up a prompt where you specify start and end dates. Uses 1 processor.

ep2date_sub
Takes YYYYMMDD dates and converts to an epoch time (seconds since Jan 1, 1970) or vice versa depending on input type (detects if input is epoch or YYYYMMDD). Calls in a different date syntax depending on OS.

textreplace_sub
Single location for sed with all its various options used in the EGI suite. Calls in the correct syntax for Mac or Linux to accomplish the same purpose. There are 7 different sed options currently supported. See the header of the script for more information. Not yet fully implemented.

(1) egiprepdaily and egi_prepdaily_multi
egiprepdaily has the purpose of creating .in files with given weather files, GPS files (optional, just enter location in file_loc.lst), and opus interferogram files. Creates lists of spectra, and lists that include number of scans+time stamp+interferogram peak. Uses a matlab script which will average up to 5 of the nearest weather points as long as they are within two minutes time, and also creates a file for arbitrary solar intensity based on the interferogram peaks. Requires you to have already set a .top file to the .in.
egi_prepdaily_multi is the multi-processor version (both scripts required).

(2) egi_i2sdaily
Runs i2s using .in files created by egiprepdaily

(3) egirunlogdaily and egi_runlogdaily_multi
Creates sunruns, runlogs, and lse files for each day from spectra created after running egi_i2sdaily. This also includes an OSDS correction (gets written to a different, unused runlog file). It also includes a static correction to the delta_nu value that is likely specific to the Caltech EM27 only (changes 0.24105889405 to 0.24105490224, which was found empirically from 3 months of data).
egi_runlogdaily_multi is the multi-processor version (both scripts required).

(4) egi_idlfree_mod
This copies over pre-made .mod files to the correct location if they have been created previously (e.g. by the Caltech centralized mod_maker). If IDL is specified to be installed in the **file_loc.lst** this function will call in egi_modmaker instead. I am considering eventually linking this with the centralized mod_maker to run files automatically.

(5) egi_modmaker
This script redownloads all of the NOAA NCEP data for the given years (may be slow if system has a slow connection). Then creates (or attempts to create) .mod files for every day.

(6) egigsetup and egi_gsetup_multi
Creates .gsin files (inputs to gsetup) and runs gsetup to create .ggg files for everyday for every window.
egi_gsetup_multi is the multi-processor version (both scripts required).

(7) egi_gfitdaily
First creates a master multiggg_*.sh file and then runs gfit on multiple processors. The master multiggg_*.sh file is optimized so that O2 will run first, then CO2, then CH4. Finally CO, H2O, HDO, N2O, and air are run in a mixed up order. Note: these are currently all the species allowed, but it could be easily modified to include other species.
After gfit is run, post-processing is run on a daily basis.

(8) egiopusrldaily

Creates sunruns based on OPUS spectra. Similar functionality to egiprepdaily - finds and coordinates proper weather data and gps data. This runs automatically instead of egiprepdaily in egi_full if parameter 2 in file_loc.lst is set as 0.

(9) egi_postgfit

Checks for NaN values in col files and will rerun these windows in gfit. Will calculate the averaging kernels as well if option is on. However these are individual files and it is recommended they be combined before trying to work with them for speed in reading in. They can be combined by running the aks_combine_files script.

egi_cumulative
Inputs are not explicity required on this one, rather it just looks for all folders that contain data that has gone through processing. Combines all the runlog, lse, mav, ray, and col files. Then runs post-processing on combined files. Note: for the EM27 outputs can get huge.
This script also requires a cumulative_exclude.lst that includes dates and specific spectra (or rather line-numbers) to exclude. For example
20140528
20140529:1,2,30-50
Excludes any data from May 28, 2014 and excludes spectra (rather line-numbers) 1, 2, and 30 through 50 on the 29th.

This script has been updated to allow specified date inputs if for some reason you do not want to concatenate all files.

9. Other scripts:

pfit_jkh
An IDL utility. Same as pfit with GGG, but allows user to specify spt locations, enter in new runlogs while running, and change file-save names. Note: any enter settings are still reset upon closing pfit

aks_combine_files
Calls in akscombineday on multiple processors. These scripts together combine the averaging kernel files for each run and microwindow and instead create one large file (extension .akf for averaging kernel full) for each microwindow for a day. These files are much easier to work with and faster to read in. However the initial combining/running of these programs takes a long time (max is one processor per day, took about 1 cpu-day per extended detector EM27 day). Averaging kernels should not need to be computed very often for a given site though.
The aks_combine_files combines the daily files together as well as the daily pressure files, and is adapted for multiprocessor use. Input syntax is the same for all programs (see section 7 – "datein_sub).

snrspectra
Calls in/uses em27_signoise.m matlab script. This script was developed to calculate signal to noise ratios of spectra and add them to the printed value of 1000 in the runlogs. Originally it was created to look only at sparse spectra to see general trends of S/N ratios with time. How frequently the snr was/is calculated is set by the sample_freq parameter within the script.

This script is not yet fully utilized, but can be by setting sample_freq=1 within the script and adapting the ggg bin2asc script to accept frequency limits in a new file called bin2asc_wlims. (Note: signal and noise frequency limits are hard-coded within the snrspectra script by may need to adapted based on your detector). This script does not yet work with instruments employing 2 detectors.

10. Updates:

10.1 Possible Future Updates

These are listed 1) to help me keep track and 2) if I get enough requests for one of these I'm more likely to implement it.

- Include making NetCDF files in egi_cumulative, Debra already has a (Python?) script for this I think (increasing need with increasing .eof file sizes).
- Move/copy model (.mod) files to a specified directory after processing.
- Also calculate SIA based on pre-amp. I can't decide if I want this or not... Possibly may want the amplification factor to change based on instrument.
- Make an active delta_nu correction calculation when there are frequency shifts. - May not be needed, using static right now for Caltech only.
- Allow for header/descriptions in GPS/weather csv files
- Implement the textreplace_sub function
- Create a mobile version to do real-time retrievals and link with a webpage viewer
- Add in a folder with some utilities – make averaging kernels with sza (matlab script), averaging oof files, plotting, renaming files, etc.
- After creating cumulative oof files, allow them to be split again (e.g. monthly) to take advantage of average_results but to not have too large of files.
- Flag for forward/backward interferograms
- Create an option to automatically run the centralized Caltech mod_maker
- Check for Infs in .vsw or perhaps .col files. Check for 0.000 in O_2
- Save scanner temp as the new Tins
- more generic sig/time reader or at least crash earlier if improperly read in
- include option to automatically generate plots of many eof parameters

10.2 November 2014 updates:

- Allowing OPUS generated spectra to also be processed with GFIT without using I2S. Requires that spectra are appropriately named (e.g. cn20140602s0e00a.0001) and separated into daily folders. Change file_loc.lst parameter 2.
- Easily include averaging kernel calculations (entry 15 in file_loc.lst)
- Easily save all fitted spectra from windows rather than just saving the last micro-spectral windows fitted (in $gggpath/spt) for each full spectrum (entry 16 in file_loc.lst).

- Updated mod_maker9b.pro that works with the new version made by Dietrich Feist that works on the new NOAA NCEP data format.
- Mildly improved readability of ground weather station data.
- Releasing first repository version
- Fixed bug on finding spectra files for commented out lines in data_part.lst
- Restructured how mod files are created to increase speed
- Only redownload NCEP data if the last file change date is within ~9 days of end date. This saves on the need to redownload every time.
- Restructured sunrun/runlog creation to increase speed
- Added ability to create runlogs from I2S generated spectra if without original files/headers
- Silenced daily email outputs. Post-processing outputs have been redirected to files of the form cn20140613_ppoutput.dat. This can be undone in the egi_gfitdaily line ~237 comment out the line, and uncomment the next line

```
# echo "./silent_post_processing.sh" | at now
at now < post_processing.sh #this by itself sends out a lot of emails, one per day
```

- (Possibly) fixed parallel processing error (with ideas from http://unix.stackexchange.com/questions/168757/concurrency-problems-with-gnu-parallel). If there are further issues I would like to know.

10.3 January/February 2015 updates:

- Fixed various MacOS incompatibility issues with sed and date functions.
- Completes averaging kernel calculations instead of just Jacobians
- Made an option in file_loc.lst to no longer need IDL (note: mod files are still needed, but these are automatically created and uploaded to a google drive).
- Gave egi_cumulative the option of having start and end dates input
- Allow windows to be in non-default ($gggpath/windows/gnd/) directory

10.4 March-June 2015 updates:

- Added in aks_combine_files (and akcombineday). Combines ak files together for faster reading later.
- Automatically reruns windows (once) if NaNs are found.
- Moves incomplete spectra/interferograms to subdirectories.
- Changed air window (was having problems with LANL) to 6200
- Began implementing a calculation for signal to noise ratios with new scripts

10.5 July- 2015 updates:

- Automatic fortran code fixing (replaces original files with modified files)
- Fixed a bug that was moving setup_test “OPUS” spectral files because they were 131kb (changed threshold)

- Created utility to create a file_loc.lst and directories for first time users.
- Added in a kludge to rerun gsetup if the wrong files get made in the wrong directories while using parallel. I'm still not sure what the origin is, but this kludge should reduce the number of errors.
- Added in approximate runtimes to egi_prepdaily_multi and egi_opus_rl_multi. These timers are for egi_full

10.6 Aug 15-May 2016 updates:

- Added the ability to use Python instead of Matlab
- Added the ability to use 2 detectors simultaneously collecting data
- Restructured how the order of gfit windows that are ran is decided. This was based on the work of Dietrich Feist (https://tccon-wiki.caltech.edu/Software/GGG/Running_GFIT_and_I2S_on_a_Cluster#3rd_approach_%28Dietrich_Feist%29)
- Checks to the met data are now moved within files called in by egi_full rather than egi_full itself
- Updated benchmark results (previous benchmark had known issues with "sia" and wind dir data)
- Numerous bug fixes

11. Noted Problems (unfixed)

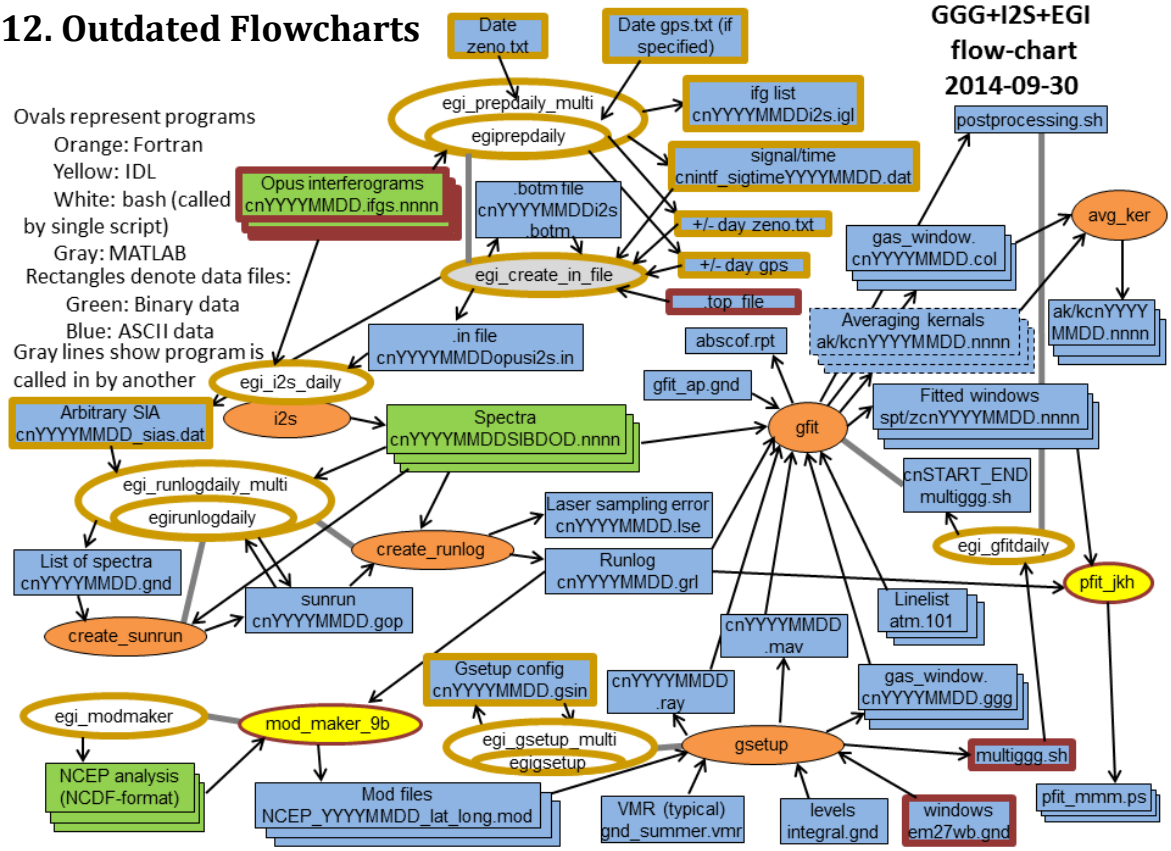
EGI will now rerun windows with NaN values once. If they occur twice errors will still result.

Added in a fix to rerun gsetup once if there was failure the first time, but it will fail if error occurs twice.

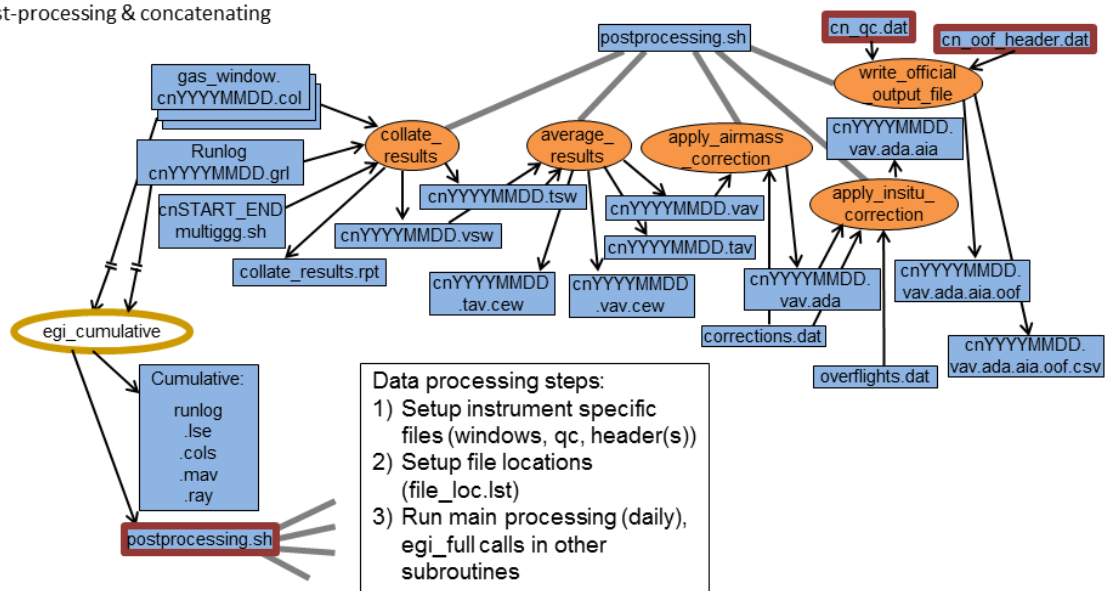
1) IEEE signaling – occurs in MacOS systems and deals with how the different operating systems work with numbers. This occurs in GFIT and hasn't caused any real problems.
2) Estimated runtime is often quite far off.

If you run into problems on installation or regular execution please let me know.

12. Outdated Flowcharts



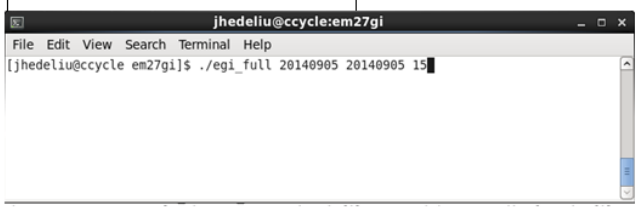
Post-processing & concatenating



Data processing steps:

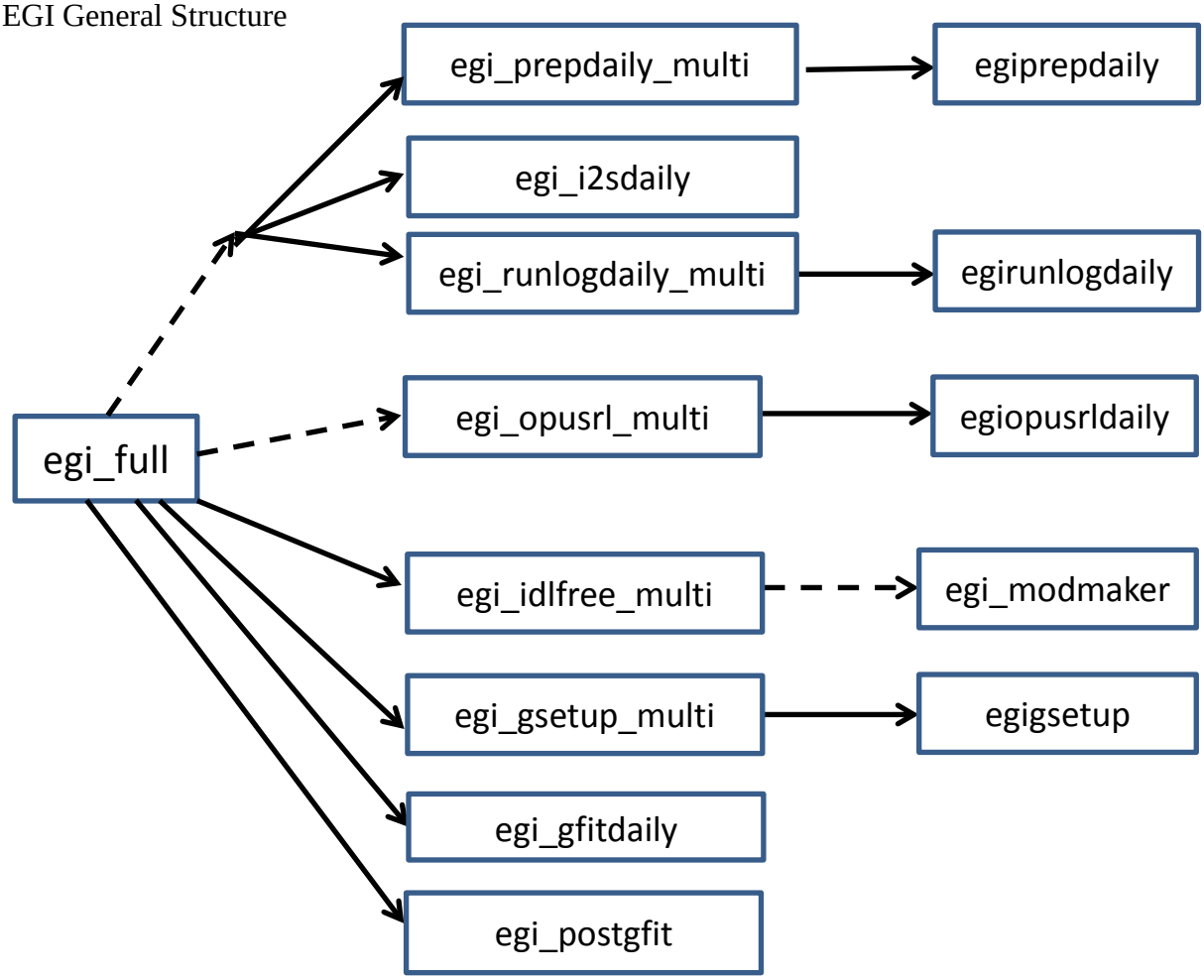
1) Setup instrument specific files (windows, qc, header(s))
2) Setup file locations (file_loc.lst)
3) Run main processing (daily), egi_full calls in other subroutines

4) Run cumulative processing



Golden outline: New script or file-type
Crimson outline: file/program changed to meet our needs

EGI General Structure



egi_cumulative

Egi_cumulative is run after the xxYYYMMDD_ppoutput.dat files are all checked (filesize is good indication) to make sure there were no errors.

Appendix A. Full install on Mac system, 10.9

We noted in January there were a number of incompatibilities when originally installing EGI on Mac systems. This details how to do a full install on a MacOS, including some additional recommended utilities. **Please note:** These instructions are likely not specific to your system! You should use equivalent commands that meet your needs.

MacOS install notes: (Using Mavericks)

1) Need Xcode (from app store)
2) Will need python (came preinstalled in Mavericks)

3) Installed homebrew:

```
ruby -e "$(curl –fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Installed with:

```
sudo brew doctor
```

Next I updated with

```
brew update
brew upgrade
brew doctor
```

4) Mercurial installed

```
brew install hg
```

5) GNU parallel installed, and citation notification silenced

```
brew install parallel
parallel --bibtex
```

6) Fortran install

```
brew install gcc
```

7) Downloaded GGG according to TCCON wiki (https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2012_Release_Notes#Download_Instructions) User=TCCON, password=!co2

```
hg clone https://lamont.gps.caltech.edu/tccon/stable/hg/ggg-stable
ggg_location --insecure
```

8) Had the repository remember name and password according to directions on the wiki

Also removed error message (about insecure) by creating and editing my $HOME/.hgrc file (see https://tccon-wiki.caltech.edu/Software/GGG/Download/GGG_2014_Release_Notes#Download_and_Installation_Instructions)

9) *Note: setting paths has changed from OSX 10.9 to OSX 10.10.* Set environmental variables according to instructions on http://apple.stackexchange.com/questions/106355/setting-the-system-wide-path-environment-variable-in-mavericks e.g.

```
echo setenv GGGPATH /code_tests/ggg2014|sudo tee -a /etc/launchd.conf
echo setenv gggpath /code_tests/ggg2014|sudo tee -a /etc/launchd.conf
echo setenv i2spath /code_tests/ggg2014|sudo tee -a /etc/launchd.conf
```

Then restarted computer.

10) Installed GGG according to instructions on wiki

```
cd install
./master.sh
```

~~b. Made some changes to GGG according to instructions in "Changes to GGG" section.~~ Should be obsolete with newest setup_test

11) Installed I2S

```
hg clone https://jhedeliu@bitbucket.org/em27ggg/i2s_double_developmental ./
```

Then compiled

```
./master.sh
```

~~b. There is an optional patch for this. See the "Changes to I2S Developmental" section. .~~ Should be obsolete with newest setup_test

12) Installed EGI

```
hg clone https://jhedeliu@bitbucket.org/em27ggg/egi2014 ./
```

13) Also needed matlab for EGI, so I installed matlab, according to instructions on Caltech software website

14) Added EGI folder to matlab path as well as the subfunctions folder to the matlab search path (may not be necessary).

15) a. I also installed Dropbox while I was at it.

b. I also installed kdiff3 while I was at it

```
brew install kdiff3
```

16) Changed colors of bash terminal (optional)

Added the following lines to my $HOME/.bash_profile file

```
export
PS1="\[\033[36m\]\u\[\033[m\]@\[\033[32m\]\h:\[\033[33;1m\]\w\[\033[m\]\$ "
export CLICOLOR=1
export LSCOLORS=ExFxBxDxCxegedabagacad
alias ls='ls -GFh'
```

17) While I was at it I also added a matlab alias to my bash_profile file

17a) I found the matlab location by opening up matlab and running matlabroot. (example results: /Applications/MATLAB_R20xxx.app). *The path will be different depending on you OSX version and where you installed matlab.* Then I added the following to my $HOME/.bash_profile

```
alias matlab='/Applications/MATLAB_R2014b.app/bin/matlab'
```

17b) I also set

```
echo setenv matlabpath /Applications/MATLAB_R2014b.app/bin/matlab|sudo tee -a
/etc/launchd.conf
```

Computer had to be restarted for this last part to take effect.

18) *Note: In OSX 10.10 md5sha1sum had to be installed instead of coreutils.* Had to install md5sum by running:

```
brew install coreutils
```

Added files to path

Edited /etc/paths
Added the line
`/usr/local/opt/coreutils/libexec/gnubin`

19) Ran setup test on EGI

```
./setup_test
```

Appendix B. Install on JAXA MacOS, 10.10

Suto-san has kindly provided the following instructions on installing on MacOS 10.10 (Yosemite). Again note that these instructions may or may not match exactly with your system.

//
How to install egi2014 with MAC OS 10.10
1. Install Xcode
Download Xcode 6.4 from apple developer site and install

2. Install command line tools
Download command line tools for Xcode6.4 from apple developer site and install

3. Install homebrew (a tool for downloading and installing common software packages and utilities)
To install brew v 0.9.5,
Execute the following commands on terminal

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
sudo brew doctor
```

The following message will be displaed, but it might be negligible.
Warning: Unbrewed dylibs were found in /usr/local/lib.
If you didn't put them there on purpose they could cause problems when
building Homebrew formulae, and may need to be deleted.
Unexpected dylibs:
/usr/local/lib/libecomlodr.dylib

```
brew update
brew upgrade
brew doctor
brew install hg
brew install parallel
parallel --bibtex
brew install wget
brew tap homebrew/versions
brew install gcc
```

* Regarding " brew install gcc", the latest version of gcc will be installed.
if you would like to use the old version of gcc, such as gcc46,

you have to execute " brew install gcc46".
** Regarding gfortran, it's included on gcc
The latest version of homebrew, it don't have the tap for "gfortran".

4. Generic "homebrew" basic commands (how to use it)

```
brew list
brew search
brew remove xxxx   (xxxx=gcc, wget, or... package name)
brew install gcc
brew uninstall xxxx
```

uninstall "homebrew"
Execute the following command

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/uninstall)"
```

5. Check the software version
gcc -v , 5.2.0
g++ -v , 5.2.0
gfortran -v , 5.2.0

6. Install "GGG2014"
In my case, I installed the GGG2014 on the following path; /WORK/EM27/GGG2014

```
GGGPATH = /WORK/EM27/GGG2014
gggpath = /WORK/EM27/GGG2014
i2spath = /WORK/EM27/GGG2014/i2s

cd /WORK/EM27/
hg clone https://lamont.gps.caltech.edu/tccon/stable/hg/ggg-stable ./GGG2014
--insecure
```

**** **The setting procedure for environmental variables has drastically changed from OS10.9 (Mavericks) to OS10.10 (Yosemite).**
You have to make a startup list on /Library/LaunchAgents

In my case, the following procedure was applied.

```
cd /Library/LaunchAgents
```

make "my.startup.plist" by text editor
write the following text from <?xml > to </plist>
****

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>Label</key>
 <string>my.startup</string>
 <key>ProgramArguments</key>
```

```
 <array>
  <string>/bin/launchctl</string>
  <string>setenv</string>
  <string>GGGPATH</string>
  <string>/WORK/EM27/GGG2014</string>
  <string>/bin/launchctl</string>
  <string>setenv</string>
  <string>gggpath</string>
  <string>/WORK/EM27/GGG2014</string>
  <string>/bin/launchctl</string>
  <string>setenv</string>
  <string>i2spath</string>
  <string>/WORK/EM27/GGG2014/i2s</string>
  <string>/bin/launchctl</string>
  <string>setenv</string>
  <string>matlabpath</string>
<string>/Applications/MATLAB_R2015a.app/bin</string>
 </array>
 <key>RunAtLoad</key>
 <true/>
</dict>
</plist>

**********
```

After that, execute the following command

```
launchctl load /Library/LaunchAgents/my.startup.plist
```

and, restart the computer

7. Run the setup test for GGG2014

```
cd /WORK/EM27/GGG2014/install
  ./master.sh
```

8. Modify the GGG2014 program (**note: these fixes should be obsolete for new users with the latest July 2015 version of setup_test)**

/src/collate_results/collate_results.f
Change line 211 from:
max_delta_t=0.0025 ! 9.0s (ground-based TCCON)
to
max_delta_t=0.00001 ! 0.04 s (EM 27)

/src/write_official_output_file/write_official_output_file.f
Change line 55 from: & cc*20
To
& cc*21
Then change line 388 from: cc=sarr(k)(:20)
to
cc=sarr(k)(:21)

make the recompile

```
./compile_ggg.sh
```

9. install i2s_double
hg clone https://jhedeliu@bitbucket.org/em27ggg/i2s_double_developmental ./i2s_double
cd i2s_double
./master.sh

10. modify the i2s_double (**note: these fixes should be obsolete for new users with the latest July 2015 version of setup_test)**
$i2spath/opus-comn/get_opusigram_params.f.
line 1245 to 1246
```
      if(INSstr.eq.'EM27SUN'.or.INSstr.eq.'EM27/SUN'
     &     .or.INSstr.eq.'...'.and.em27flag.gt.0)then
```

recompile i2s_double
```
./master.sh
```

11 Renamed i2s and i2s_double **(note: you could also change your system i2spath variable, whichever is easier. I discourage you from placing one mercurial repository within another.**
renamed i2s folder as i2s_single
renamed i2s_double folder as i2s

12. Install EGI2014
```
cd /WORK/EM27GGG2014
pwd
      /WORK/EM27/GGG2014
hg clone https://jhedeliu@bitbucket.org/em27ggg/egi2014 ./egi2014
```

13. Add the following matrab script on egi2014 folder
MatlabFiles (include "mapdisp" and "maputils")
and
timezone.m, nanmean.m, abstractAngleConv.m, npi2pi.m, toDegrees.m
if you don't put these files, you will have error

14. Install the Matlab
in our case, license xxxx89 will be activated.

15 add the path
$GOSAT
```
vi .bash_profile
```
i

```
export PATH=/usr/share/bin:$PATH
export PATH=/usr/bin:$PATH
export PATH=/bin:$PATH
export PATH=/usr/sbin:$PATH
export PATH=/usr/local/opt:$PATH
export PATH=/usr/local/bin:$PATH
if which pyenv > /dev/null; then eval "$(pyenv init -)"; fi
```

ESC

:w
:q

alias matlab='/Applications/MATLAB_R2015a.app/bin/matlab'

16. Install kdiff3

```
brew install kdiff3
```

17. Install md5sum

```
brew install coreutils
```

add path;

```
vi /etc/paths
```

i

/usr/local/bin
/usr/bin
/bin
/usr/sbin
/sbin
/usr/local/opt/coreutils/libexec/gnubin

ESC
:w
:q

restart the computer

18. Test for egi2014

```
cd /WORK/EM27/GGG2014/egi2014
./setup_test
```

use double side interferogram -> yes
use IDL -> NO
CPU number -> 5 *(depends on you system)*

19. Executing egi2014

```
./egi_full 20140628 1
```