###
###GMOS MOS ---> GUIDELINES POSTED IN THE DR FORUM###
###

To download the attached documents, please access the GMOS DR Forum:

http://drforum.gemini.edu/topic/gmos-mos-guidelines-part-1/

################### PART #1 ###################

CONTEXT:

One of the difficulties (actually the first one!) that a new GMOS user finds is to establish a clear sequence of reduction steps. There is a huge documentation on IRAF (Gemini package) tasks and parameters, which requires considerable time to gain insight into the process. Some basic steps (e.g., bias and overscan subtraction, trimming) are quite obvious to anyone who deals with reduction of astronomical data, but there are some subtleties regarding spectroscopic data treatment that deserve to be discussed.

AIMS:

The objective of this post is to summarize a possible (not definitive!) way to reduce GMOS multi-object spectroscopy (MOS) data (a more complete document, containing examples and a longer discussion, is under preparation and will be posted in this forum). USERS ARE REALLY WELCOME TO SHARE THEIR OWN EXPERIENCE, to highlight critical tasks parameters and issues found during the process. In what follows, we assume that you have the latest version of Gemini Iraf Package (https://www.gemini.edu/node/11823) installed on your machine.

DATA FORMAT AND DETECTORS:

Images are stored in multi-extension FITS format, containing a Principal Header Unit (PHU, containing informations about the instrument, weather conditions at the time of frame acquisition, target coordinates, etc) and 12 "Science extensions" (one for each amplifier, in the case of the Hamamatsu detectors), which contains the data itself.

In order to see the structure of your raw FITS files, just type:
fxhead %YOUR_RAW_FITS_IMAGE%.fits

The disposal of the amplifiers is shown in http://www.gemini.edu/sciops/instruments/gmos/data-format-and-reduction.

Informations about gain, readnoise and bias level for each amplifier for all readout modes are registered in the file gmosamps.dat inside gmos$data folder (where gmos$data indicates the "data" directory inside the "gmos" folder, e.g., %YOUR_HOME_FOLDER%/anaconda/envs/astroconda/iraf_extern/gemini/gmos/data/gmosamps.dat). More detailed information about the detectors (saturation levels, plate scale, etc) can be acessed in the links https://www.gemini.edu/sciops/instruments/gmos/imaging/detector-array/gmosn-hamamatsu-array (for Gemini-N) and https://www.gemini.edu/sciops/instruments/gmos/imaging/detector-array/gmoss-array-hamamatsu (for Gemini-S).

DATA ORGANIZATION:

As we get more and more data for different targets during different nights, the data organization becomes a critical task. Besides, during the data reduction process, the number of produced files increases considerably.

In the file "files_and_folders_organization.txt" attached to this post, we suggest an useful disposal of folders and subfolders, for a good organization of your data.

################### PART #2 ###################

DATA REDUCTION STEPS:

In this section, we propose a summarized sequence of reduction steps and give a short description of each task. As said above, a more complete documentation will be made available in this forum.

1) GPREPARE
Run the task gprepare on all images (arcs, GCAL flats, science frames, twilight flats) in order to:

- prepare raw GMOS data for reductions;
- identify the PHU and the science extensions (see section DATA FORMAT AND DETECTORS above);
- update the RDNOISE and GAIN values;
- attach the content of the Mask Definition File (MDF) in a separate FITS extension (BINTABLE).

2) GSREDUCE
Run GSREDUCE on the GCAL flats in order to:
- perform overscan subtraction;
- trim the frames;
- perform bias subtraction (an already processed bias can be obtained from the Gemini Science Archive).

The reduced images are multiplied by the respective gain values, so the outputs are in electrons.

3) GMOSAIC
Run this task on the GCAL flats in order to:
- Mosaic the different science extensions in only one extension.

It is recommended to set fl_paste-, so the task will take into account the relative orientations of the 3 CCDs; with fl_fixpix+, the task will interpolate across the CCDs gaps, which are specified in the file gmos$data/chipgaps.dat (taking into account the detectors binning, specified in the keyword CCDSEC of each frame).

4) GSFLAT
Run GSFLAT on the GCAL flats in order to:
- Combine the GCAL flats;

In order to produce the output combined and not normalized GCLAl flat (task parameter combflat), we must set fl_keep+. The combflat will be used in the next step, as a reference for cutting the slitlets.

5) GSCUT: 1st run
Run GSCUT on the combined GCAL flats (combflat, see step 4 above) in order to:
- Identify automatically the edges of each slitlet using the gradient method.

The task identifies the edges of each slitlet from the gradients of counts in the combined GCAL flats. It also uses the information contained in the file gmos$data/GMOSgratings.dat, which must be specified in the task parameter gratingdb. Use the output files (task parameters outimage and secfile) to check if the slit edges were properly identified. Using fl_over+ (default) allow the task to use 1.05x slit length to accommodate distortions.

6) GSCUT: 2st run
Run GSCUT on the combined GCAL flats (combflat, see step 4 above) in order to:
- Identify the slits edges and update the MDF extension of the combflats with the positions of each slit.

During this run, with no output files specified (that is, outimage="" and secfile=""), the task with update the MDF extensions of the input images with the slits positons.

7) GSREDUCE
Apply GSREDUCE to the arc frames in order to:
- perform overscan subtraction;
- trim the images;
- perform bias subtraction;
- mosaic the science extensions (interpolating across the gaps: fl_fixpix+);
- cut the slitlets and paste them in different science extensions.

As normally the the arc frames are obtained in fast readout mode, a proper bias image should be specified. The cut of the slitlets is performed according to the information contained in the MDF extension of the combflats (step 6 above).

8) GSAPPWAVE
Run GSAPPWAVE on the reduced arcs in order to:
- Add approximate wavelength calibration to the headers of GMOS spectra.

This approximate wavelength calibration is performed based on the informations contained in the files gmos$data/GMOSgratings.dat (which must be specified in the task parameter gratingdb) and gmos$data/GMOSfilters.dat (task parameter filterdb).

9) GSWAVELENGTH
Run GSWAVELENGTH on the reduced arcs in order to:
- perform wavelength calibration.

A high-resolution line list of CuAr is available at http://ast.noao.edu/sites/default/files/GMOS_Cookbook/_downloads/cuarHiRes.txt. A linelist including 2nd order CuAr lines, useful for calibrating red wavelength settings, is available at http://www.gemini.edu/sciops/instruments/gmos/calibration/example-cal-data/cuar-spectra.

The task should be run in interactive mode. It should be noted that it is not enough to simply analyze the RMS of the residues; the coverage of the solution must be verified: calibration lines must be present along the whole fit domain, whenever possible, in order to provide adequate rectification (see step 10 below) of the spectral lines. For a proper correction of possible distorsions along the spatial direction (that is, y-axis), it is important to specify low values for the task parameters step and nsum; putting a value of 2 for both parameters, the task will calculate a function lambda (x_pixel) for each 2 lines along the spatial direction, thus providing good sampling. See the files attached to this post for illustration purposes.

################### PART #3 ###################

In addition to what was said above, regarding the wavelength calibration, other 3 parameters should be set to yes when running GSWAVELENGTH, namely refit, trace and fl_addfeat.

The parameter refit+ allows the task REIDENTIFY (which is called by GSWAVELENGTH) to refit the coordinate function. According to the manual of the task REIDENTIFY, a new coordinate function of the same type as in the reference is fit using the new pixel positions. Otherwise only a zero point shift is determined for the revised coordinates without changing the form of the coordinate function.

When the parameter trace is set to yes then as the reidentifications step across the image the last reidentified features are used a the reference. This "tracing" is useful if there is a coherent shift/distorsion in the features.

The option fl_addfeat allows REIDENTIFY to automatically add new features from a line   list during each reidentification. This option can be used to compensate for lost features from the reference solution, particularly when tracing.

Try to use low orders (parameter order of GSWAVELENGTH) during the lamba (x_pixel) fitting procedures. Order is the number of polynomial terms (coefficients) or the number of spline pieces. At the end of the procedure, we have a dispersion solution every 2 rows (if parameter step is 2) throughout the data. GSWAVELENGTH then finds a surface lambda (x_pixel,y_pixel): the parameters for this run are fitcxord and fitcyord; again, it is recommended to choose values that are as low as possible without there being obvious systemic residuals. The results of the fits for each science extension are recorded in the database folder.

10) GSTRANSFORM
Apply GSTRANSFORM on the arc frames using the wavelength solutions found in step 8 in order to:
- check the rectification of calibration (CuAr) lines;
- linearize the data in the wavelength direction.

If proper wavelength solutions were derived, the emission lines will be perpendicular to the dispersion axis (X-axis). Use task IMPLOT across some comparison lines. It is important to highlight that the better the spectra rectification, the better your sky subtraction. If some of the science extensions were not properly rectified, you will need to re-run GSWAVELENGTH on these extensions (NOTE THAT, WHEN THE PARAMETER fl_overwrite IS SET TO yes, EACH GSWAVELENGTH RUN WILL OVERWRITE EXISTING DATABASE ENTRIES WITH THE NEW SOLUTION).

11) GQECORR

Run GQECORR on the reduced (non-normalized) GCAL flats (output of GSREDUCE in step 2 above) in order to:

- correct differences in quantum efficiency (QE) between the 3 CCDs;
- create correction images to be used to correct the science images.

The task employs the wavelength solution obtained in step #9 above and the content of the file gmos$data/gmosQEfactors.dat, which must be specified in the task parameter qecorr_data. This database file contains the calculated GMOS QE data based on supplied or empirically determined QE data for each CCD. The file contains fit coefficients for spectroscopic data. The correction values were empirically determined and they are simple polynomials (order 4 for EEV CCDs and order 9 for Hamamatsu):

```
     correction_value =
             C0 + (C1 * lambda) + (C2 * lambda**2) + ... + (Cn * lambda**n))
```

where lambda is the calculated wavelength of the pixel for the reference image and C0-n are the coefficients of the polynomial fit. The default coefficients are stored in the file gmos $data/gmosQEfactors.dat, given by qecorr_data.

12) GMOSAIC

Run GMOSAIC on the QE corrected GCAL flats, in order to

- paste the 12 science extensions into only one science extension.

Use fl_fixpix+ to allow interpolations across the chips gaps (as specified in the file gmos $data/chipgaps.dat); also use fl_paste- so the task will take into account the relative orientations of the 3 detectors.

13) GSFLAT

Run GSFLAT on the mosaiced and QE corrected GCAL flats in order to

- generate normalized flats, which will be applied to correct the science images;

The response function for every single line (task parameter fl_seprows+) in the region of each slitlet will be fitted. Then the data/fit ratios are obtained and registered in the output, so the pixel-to-pixel variations are determined. For the edges of the slitlets to be properly identified, the user needs to specify the (mosaiced) combflats generated in step #6; these combflats contain the slits positions recorded in their MDF extensions. The pixels in the regions between the slits are replaced with "1.0".

################## PART #4 ##################

14) Bad pixel Masks

At this stage, the normalized GCAL flats (output from step #13 above) can be used to identify bad pixels. For example, according to the GMOS MOS example (load GMOS package in your IRAF and type gmosexamples MOS), flats taken before late 2004 contain a pair of emission lines due to a fluorescent surface in GCAL. This way, we have to identify those pixels and correct them, in order to avoid introducing spurious features in your science images. Below we give some examples using IRAF tasks IMREPLACE and FIXPIX.

------> Normalized flat (output of step #13): normflat_maskXXX_centrallambdaYYY.fits
------> Create a copy of this file:

```
cl > cp normflat_maskXXX_centrallambdaYYY.fits
MODIFIED_normflat_maskXXX_centrallambdaYYY.fits
```

------> Execute the commands (in the examples below, the fits extension [2] corresponds to the SCI extension
------> of the image; remember that IRAF tasks perform operations on one fits extension at a time):

```
cl > imreplace MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2] lower=1.02 upper=INDEF
value=0.00000
cl > imreplace MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2] lower=INDEF upper=0.98
value=0.00000
```

------> With the two executions above, pixels with normalized counts above 1.02 or below 0.98 will be replaced by zero. Now execute the commands:

```
cl > imarith   MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2]  /
MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2]
MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2,overwrite]  divzero=0.00000

cl > imarith   MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2]  -   1.0
MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2,overwrite]

cl > imarith   MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2]  *  -1.0
MODIFIED_normflat_maskXXX_centrallambdaYYY.fits[2,overwrite]
```

------> With the three executions above, we will have a bad pixel mask (BPM) with the IRAF standard: badpixels=1.0; goodpixels=0.0. Now you can rename the file to make the procedure clearer:

```
 cl > mv MODIFIED_normflat_maskXXX_centrallambdaYYY.fits
BPM_normflat_maskXXX_centrallambdaYYY.fits
```

15) FIXPIX

Now you can run FIXPIX on the normalized GCAL flats in order to get rid of bad pixels (thus avoiding the introduction of spurious features on the science images, when these are divided by the flats). To make the procedure clearer, you can first rename the original normalized flats:

```
cl > mv normflat_maskXXX_centrallambdaYYY.fits
fixpixed_normflat_maskXXX_centrallambdaYYY.fits
```

After that, execute:

```
cl > fixpix fixpixed_normflat_maskXXX_centrallambdaYYY.fits[2]
mask=BPM_normflat_maskXXX_centrallambdaYYY.fits[2] linterp=INDEF cinterp=INDEF
```

Again, in this example we are assuming that extension [2] corresponds to the SCI extension. You must check this for the task to work properly.

################## PART #5: Reducing the science frames ##################

16) Run GSREDUCE on the frames containing science spectra in order to:

- Perform overscan and bias subtractions;
- Trim the images.

17) Before mosaicing and flat fielding the science images, run GQECORR in order to:
- Correct the frames for quantum efficiency (QE) differences among the detectors.

Use the same corrections derived for the flats, that is, use the correction images (parameter corrimages of GQECORR task) that were created in step #11 above.

18) Run GSREDUCE again on the science frames in order to:
- Flat field and mosaic the frames;
- Cut and paste the individual spectra in different fits extensions.

In the parameter flatim of GSREDUCE task, we should specify the normalized, GMOSAICed and GQECORRected flats generated in step #15 above. The cut of the slitlets is performed according to the information contained in the MDF extension of the combflats generated in step #6; these combflats should be specified in the parameter refimage of the task GSREDUCE.

Here we highlight a short note about the fl_fixpix parameter in GSREDUCE task: setting fl_fixpix+, the task GMOSAIC will interpolate the spectra across the CCDs gaps, which are specified in the file gmos$data/chipgaps.dat. Setting fl_fixpix- will avoid interpolations and leave null counts in the regions of the gaps. The choice depends on the kind of data one has at hand: if the project was planned to cover the intervals between the detectors, in order to obtain a smooth continuum, then we have multiple spectra obtained with different grating central wavelength (lambda_c) for each object. In this case, we should leave null counts in the gaps (fixpix-); the information lost in the gaps will then be recovered in the last reduction step, after combining (using, e.g., SCOMBINE and a proper rejection algorithm, such as MIN MAX or SIGCLIP) the final spectra centred in different lambda_c.

19) Apply GSTRANSFORM in order to:

- rectify the science spectra:
- linearize the data in the wavelength direction.

We must specify the task parameter fl_wavtran+ in order to apply the wavelength calibrations. In the task parameter wavtraname we need to specify the lamp image root name (suppressing the extension .fits) of the image that was used to establish the wavelength calibration with GSWAVELENGTH (step #9). The name of the directory which contains the calibration files (features input data and fitting output) is specified in the database parameter. The option fl_flux+ conserves the flux per pixel during the transform. The interpolation type applied by GSTRASFORM is specified in the parameter interptype.

################### PART #6 ###################

20) Run GSSKYSUB in order to:
- subtract the background from the science spectra.

The first step is to determine the size of the star PSF on the images. Choose one of the brightest star in your sample and run the IRAF task IMPLOT to take a look at the counts along the spatial direction. We want to determine an adequate region for the sky sampling. A good reference is to determine the points at which the star PSF falls to ~10% of its peak. Choose a background sampling window outside this region. This way, we are selecting a region that is not heavily affected by the light of the star. It is also recommended to restrict the background window to pixels that are not too close to the slit edges (which are affected by the light loss).

Here is an example: suppose that the full star PSF have an width of 13 pixels on the image. Suppose also that pixels are 2x2 binned (plate scale results 0.16 "/pixel in the case of the GMOS-S Hamamatsu array). This way, the parameter mosobjsize of the GSSKYSUB task should receive the value (0.16"/pixel) x 13 pixels = 2.1. An aperture of this size centered on the object is excluded from the sky sample. The parameter mos_sample specifies the maximum fraction of the MOS slit length that will be used as the sky sample. Use 0.90 or 0.95, to eliminate the edges of the slitlets.

The task parameter naverage specifies the number of sample points to be combined to create a fitting point. A positive value specifies an average and a negative value specifies a median. Here you can specify a huge negative number (e.g., -100) if you want to simply use two good values, one from either background window. Using a median instead of average is recommended in order to avoid avoid the possible introduction of cosmic ray spikes in your background sample.

Using a chebyshev polynomial (function parameter) with order=1 (that is, fitting a constant to the points) is enough for a good background subtraction. It is also recommended to run GSSKYSUB task in interactive mode, so the investigator can check with there are large differences (due to illumination effects) in the background level on each side of the science spectrum.

21) At this stage, it is useful to perform a check on each science extension, paying special attention to the columns close to the CCDs gaps. Since at this stage we have performed many operations on the images (e.g., linear interpolations to rectify the spectra; see step #19 above), some columns close very to the gaps may have been affected with bad counts. We can flag this bad columns and the whole set of pixels, as shown in the example below:

```
imreplace science_image.fits[SCI,1][327:359,*]    value=-100 #gap1
imreplace science_image.fits[SCI,1][1380:1413,*]  value=-100 #gap2
```

In this example, we are flagging the pixels inside the each gap (and few columns on its borders) with -100 counts. This is an interactive step. The complete set of commands (analogous to those above) for the different science extensions can be specified in a proper .cl file and executed on IRAF command line:

```
cl> cl < myscript_flagging_bad_columns_CCDs_gaps.cl
```

The information lost in these regions will be recovered in the last step of the reduction procedures, by combining multiple spectra obtained with different grating central wavelength (lambda_c) for each object (if the project was planned to cover the CCDs gaps and obtain full wavelength range).

################## PART #7 ##################

22) Run GSEXTRACT in order to
- Extract GMOS spectra to 1D (counts versus wavelength)

First, we need to specify the aperture size to extract the spectra. Following the same reasoning as that of step #20 above, we can determine the points at which the star PSF falls to ~10% of its peak. Use a bright source to estimate the aperture. The full width at this count level can be used as the aperture size (apwidth parameter, which have to be specified in arcseconds; in the case of GMOS-S Hamamatsu array we have, for unbinned pixels, a plate scale of 0.08"/pixel). Use background=none since the sky background was already subtracted in step #21 above. Use trace+ to allow the task to trace the spectrum.

In order to adequately trace faint spectra, it is important to increase the values of the task parameters tsum (number of lines summed along the dispersion direction when looking for the center of the spatial profile) and tstep (step along the dispersion axis between each determination of the spectrum positions). Run GSEXTRACT iteratively (fl_inter+) to make sure that all faint objects are traced correctly. Using low order (task parameter torder = ~3 - ~5) chebyshev or spline3 fitting functions (parameter torder) is normally adequate for a good tracing.

Again, it is important to run GSEXTRACT interatively to make sure that the proper tracing solution is found. This is particularly true in the case of faint targets affected by cosmic rays. Here we transcript an important paragraph taken from the GSEXTRACT help manual:

When a spectrum is too faint or otherwise difficult to trace, APALL (which is an IRAF task called by GSEXTRACT internally) can fail in one of several ways, including a crash. Sometimes (e.g., due to a cosmic ray) the trace is silently performed along a bad path, creating an incorrect output spectrum. When apall detects a problem, it may write a warning to the log file and the corresponding output spectrum may be blank or missing.

Note that once an image has been processed with gsextract, it is necessary to delete the corresponding database files (./database/) before re-reducing the image with different extraction parameters. Otherwise, the old parameter values in the database will silently override the user-specified values.

It is also recommended to activate the GSEXTRACT parameters find+ and recenter+, so the task will be able to automatically define and recenter the apertures. When we have to extract a series of similar spectra (e.g., sequential exposures centred on the same wavelength, which will then be summed up to increase the S/N), we can run GSEXTRACT on the first image and use it as a reference for tracing noninteractively the spectra in the other images. Below we put an example:

```
cl> GSEXTRACT  spectrum.fits reference= first_image_in_the_sequence.fits find- trace-
recenter+
```

This way, GSEXTRACT will employ the tracing solution determined for one of the images in a sequence of exposures, then recentering and resizing but not retracing.

################## PART #8: Finishing the data reduction ##################

23) Normalization

Once we have the extracted spectra, one of the following steps can be the continuum normalization. Here we need to run task continuum with a sufficiently flexible function (a cubic spline is a good function to perform this job) that will allow the fit of the continuum along the whole dispersion range, without fitting the true spectral features (absorption or emission lines).

24) Combine the spectra centered on different wavelengths

In this last step, we can run SCOMBINE task to generate the final (normalized) spectra. Suppose that we have 3 science spectra for each object, each one centered on a different wavelength. We can use the MIN MAX rejection algorithm (reject=minmax) to exclude the lowest (nlow=1) and highest pixel (nhigh=1) for the combination procedure. This is the same as taking the median of 3 values for each spectral pixel. Since the spectra are wavelength calibrated, the bad pixels, cosmic rays and the CCDs gaps will fall in different regions among the 3 spectra. For the final (SCOMBINEd) spectrum, only the good pixels will be kept.

With this strategy, we avoid the use of tasks (such as CRMEDIAN or COSMICRAYS) designed to identify and replace cosmic rays; such tasks perform additional operations on the images and may therefore introduce additional noise or even render some useful regions of the spectrum useless.