

8. MaNGA TRM: Setting up and running the MANGA pipeline

Outlined here are the steps necessary to set up the MANGA pipeline. In this example, we'll set up the MPL-11/DR17 version. It is assumed that you already have a working version of IDL 7.1 or later.

8.1. Set up Directory Structure

Create a main MaNGA data directory inside which all data will be placed. This folder will be your MANGA_DIR [e.g. /Users/dlaw/MANGApipe/] in your bash or cshrc files. Note that the path to this root MaNGA data directory must not include the '-' character anywhere.

```
mkdir Manga [e.g. /Users/dlaw/MANGApipe/]
```

Inside this folder create the spectro and gcam subfolders. Inside the spectro subfolder, create folders for data and redux. Data is the directory where the raw data will be stored. Redux is the directory where the 2d output of the full MANGA pipeline will be stored.

```
mkdir data redux [e.g. /Users/dlaw/MANGApipe/spectro/data/, etc ..]
```

Note: Inside your data and gcam directories, the subdirectory structure must be [MJD] where each subdirectory is the MJD of the raw observations. [e.g. /Users/dlaw/MANGApipe/spectro/data/56290/]

8.2. Checking Out the Necessary Products

From the SDSS4 SVN, you need to check out a number of different products that contain various codes and reference files. The default below checks out the product into the directory you ran the svn command in. For help getting svn set up, see <https://trac.sdss.org/wiki/Software/SVNTutorial>

I like to separate the MaNGA pipeline data from the svn products, so I install all of the svn products into a separate SDSS_DIR directory:

```
setenv SDSS_DIR /Users/dlaw/sdss
```

8.2.1. Note on versions

Many of the SVN products will continually be updated over the lifetime of the survey, and as such you'll need to check out different versions as the project evolves. The detailed instructions below are given for the MPL-11 tagged software version.

```
mangadrp: v3_1_1
mangacore: v1_9_1
idlutils: v5_5_36
specflat: v1_33
mangapreim: v2_9
dust: v0_2
platedesign: v5_1
mangatarget: v1_2_27
mangatile: v2_3
```

8.2.2. Download instructions for svn products

- **MANGADRP**

```
svn co https://svn.sdss.org/repo/manga/mangadrp/tags/v3_1_1 $SDSS_DIR/mangadrp/v3_1_1 ; MaNGA data reduction pipeline code
```

- **MANGACORE**

```
svn co https://svn.sdss.org/repo/manga/mangacore/tags/v1_9_1 $SDSS_DIR/mangacore/v1_9_1 ; MaNGA metadata files (~ 8 GB)
```

- **IDLUTILS**

```
svn co https://svn.sdss.org/repo/sdss/idlutils/tags/v5_5_36 $SDSS_DIR/idlutils/v5_5_36 ; Shared code between all
```

- **SPECFLAT**

```
svn co https://svn.sdss.org/data/sdss/specflat/tags/v1_33 $SDSS_DIR/specflat/v1_33 ; This product is ~6 GB and co
```

- **DUST**

```
svn co https://svn.sdss.org/data/sdss/catalogs/dust/tags/v0_2 $SDSS_DIR/dust/v0_2 ; This product is ~1 GB and cont
```

- **PLATEDESIGN**

```
svn co https://svn.sdss.org/repo/sdss/platedesign/tags/v5_1 $SDSS_DIR/platedesign/v5_1 ; This product is ~10 MB and cor
```

- **MANGAPREIM**

```
svn co https://svn.sdss.org/data/manga/mangapreim/tags/v2_9 $SDSS_DIR/mangapreim/v2_9 ; MaNGA preimaging files (~ 1
```

8.3. Set up Environment Variables and Path Dependencies

In this step, you need to set up your environment variables and path dependencies in your .bashrc file or .cshrc file to find all of the downloaded products and input/output directories. The following example is for a .cshrc file.

- Make sure a directory and path to IDL exists. IDL_DIR is your path to where IDL is installed on your machine. IDL_PATH is any folder where your IDL files are located. IDL searches for code files inside this path. In these examples, we'll assume default install locations for all products from Step 1, modify these as appropriate

```
setenv IDL_DIR /usr/local/itt/idl
setenv IDL_PATH +/usr/local/itt/idl
```

- 8. MaNGA TRM: Setting up and running the MANGA pipeline
- 8.1. Set up Directory Structure
- 8.2. Checking Out the Necessary Products
 - 8.2.1. Note on versions
 - 8.2.2. Download instructions for svn products
- 8.3. Set up Environment Variables and Path Dependencies
- 8.4. Compile C Libraries
- 8.5. Download the necessary data
 - 8.5.1. Raw spectroscopic data
 - 8.5.2. Guider data
 - 8.5.3. Flux calibration templates
- 8.6. Running the MANGA pipeline
 - 8.6.1. 2d Extraction
 - 8.6.2. 3d Cube Production
- 8.7. Known Issues

Note that the '+' sign in the above path is important, since it specifies that IDL should descend into subdirectories when looking for code.

- Set up product aliases:

```
setenv SDSS_DIR /Users/dlaw/sdss
setenv MANGA_DIR /Users/dlaw/MANGApipe
setenv MANGA_SPECTRO_DATA $MANGA_DIR/spectro/data
setenv MANGA_SPECTRO_REDUX $MANGA_DIR/spectro/redux
setenv GCAM_DATA $MANGA_DIR/gcam
setenv MANGA_SPECTRO_TEMPLATES $MANGA_DIR/spectro/templates

setenv MANGACORE_DIR $SDSS_DIR/mangacore/v1_9_1
setenv MANGADRP_DIR $SDSS_DIR/mangadrp/v3_1_1
setenv MANGAPREIM_DIR $SDSS_DIR/mangapreim/v2_9
setenv IDLUTILS_DIR $SDSS_DIR/idlutils/v5_5_36
setenv DUST_DIR $SDSS_DIR/dust/v0_2
setenv SPECFLAT_DIR $SDSS_DIR/specflat/v1_33
setenv PLATEDESIGN_DIR $SDSS_DIR/platesign/v5_1
```

- Include these aliases in paths:

```
setenv IDL_PATH ${IDL_PATH}:+$MANGADRP_DIR/
setenv IDL_PATH ${IDL_PATH}:+$MANGACORE_DIR/
setenv IDL_PATH ${IDL_PATH}:+$MANGAPREIM_DIR/
setenv IDL_PATH ${IDL_PATH}:+$IDLUTILS_DIR/pro
setenv IDL_PATH ${IDL_PATH}:+$PLATEDESIGN_DIR/pro
setenv IDL_PATH ${IDL_PATH}:+$SPECFLAT_DIR/pro
setenv IDL_PATH ${IDL_PATH}:+$IDLUTILS_DIR/goddard/pro ; This is the Goddard IDL library within idlutils
set path = ($MANGADRP_DIR/bin $path)
set path = ($MANGACORE_DIR/bin $path)
set path = ($MANGAPREIM_DIR/bin $path)
set path = ($IDLUTILS_DIR/bin $path)
set path = ($SPECFLAT_DIR/bin $path)
set path = ($PLATEDESIGN_DIR/bin $path)
```

Note that, in some cases, the MANGADRP library uses functions with the same name as in the BOSS IDLSPEC2D library, but whose structure is slightly different. Therefore, if you also have the BOSS pipeline installed on your machine, be sure that it is not present in your IDL path to avoid the DRP grabbing the wrong version of these files.

8.4. Compile C Libraries

The Manga pipeline relies on some C routines that need to be compiled into the necessary library files. You must do this twice, once for idlutils and once for mangadrp. If you have evilmake already installed, then simply navigate to inside your idlutils and mangadrp directories and run the command evilmake. Note that for this to work you will need to have a standard set of C, C++, etc compiler functions installed. One non-standard compiler function that is needed to fully install idlutils is gfortran, which can be found at <https://gcc.gnu.org/wiki/GFortranBinaries>

```
cd $IDLUTILS_DIR
evilmake
cd $MANGADRP_DIR
evilmake
```

If you do not have evilmake installed, or no alias exists, evilmake is included in idlutils/bin/evilmake.

```
cd $IDLUTILS_DIR
./bin/evilmake
cd $MANGADRP_DIR
$IDLUTILS_DIR/bin/evilmake
```

Successfully running this step should produce .dylib library files inside your idlutils/lib/ directory as well as inside your mangadrp/lib/ directory. If it does not, then something has gone wrong. If you receive an error regarding the file export.h not being found, then your IDL_DIR and/or IDLUTILS_DIR and/or MANGADRP_DIR environment variable is not set, or not set correctly. Double check that this is pointing to your IDL installation, and that you've sourced your .chsrc (or equivalent) before running this command. If you still have trouble, one solution is to manually copy export.h from your \$IDL_DIR/external/ directory into each subdirectory in \$IDLUTILS_DIR/src/ and into \$MANGADRP_DIR/src/ before running evilmake.

You have now successfully installed and set up the MANGA pipeline.

8.5. Download the necessary data

We'll assume that there are specific plate, MJD combinations for which a user wants to run the DRP. You will need to know both the MJD of your observations, and the PLATEID of the plate if you want to run the full pipeline.

A detailed description of how to access data from the SAS can be found here: <https://trac.sdss.org/wiki/DataArchive/DataTransfer>

You can browse the MaNGA section of the SAS by navigating to <https://data.sdss.org/sas/mangawork/manga>

In the sections below are some specific rsync commands that can be used to pull data from the SAS for common MaNGA usage cases.

8.5.1. Raw spectroscopic data

Download the raw spectroscopic data from the SAS to your local directory for a given MJD (let's say 56741). You can do this by hand via the web browser interface (tedious), or you can grab the entire directory by doing:

```
rsync -avz rsync://sdss@dtn01.sdss.utah.edu/sas/mangawork/manga/spectro/data/56746/ $MANGA_SPECTRO_DATA/56746/
```

You will be prompted for the SDSS-IV data password. Note that if you are trying to download public data you can do so without using the collaboration password by instead using the command:

```
rsync -avz rsync://data.sdss.org/dr15/manga/spectro/data/56746/ $MANGA_SPECTRO_DATA/56746/
```

8.5.2. Guider data

Guider data (which you will need for each MJD) can be obtained for MJD 56741 for instance by

```
rsync -avz rsync://sdss@dtm01.sdss.utah.edu/sas/sdsswork/apo/gcam/56741/ $GCAM_DATA/56741/
```

You will be prompted for the SDSS-IV data password. The public version of this command is:

```
rsync -avz rsync://data.sdss.org/dr15/apo/gcam/56741/ $GCAM_DATA/56741/
```

8.5.3. Flux calibration templates

The MaNGA flux calibration routine uses as reference a variety of flux calibration templates. Since MPL-6, these are required for the pipeline to run. They can be downloaded by:

```
rsync -avz rsync://sdss@dtm01.sdss.utah.edu/sas/mangawork/manga/spectro/templates/ $MANGA_SPECTRO_TEMPLATES/
```

You will be prompted for the SDSS-IV data password. The public version of this command is:

```
rsync -avz rsync://data.sdss.org/dr15/manga/spectro/templates/ $MANGA_SPECTRO_TEMPLATES/
```

8.6. Running the MANGA pipeline

Before running the DRP, make sure to download the raw spectroscopic data and the guider data. If you are running a trunk version of the code, you should also svn update mangacore before running to be sure that you get any new metadata files that might be required.

8.6.1. 2d Extraction

To run the pipeline, start IDL from any directory. There are many ways to run the pipeline, but the simplest is to reduce a single night of data for one plate using the `mdrp_runmanga2d` calling routine. To reduce, for example, the all-sky plate from MJD 56693, you would run

```
mdrp_runmanga2d,inop='56693',outop='56693',plate='7341',/full
```

Plate specifies the plate of the observations you are reducing. Inop is a string specifying the MJD of the input raw data directory. Outop is a string name that specifies the output location inside the redux folder, for the reduced files. It is customary to also use the MJD of the observations to keep things orderly. `mdrp_runmanga2d.pro` will automatically create the appropriate directory structure inside the redux folder at runtime, as well as set all required input path names for the rest of the pipeline. The output folder structure is created as `$MANGA_SPECTRO_REDUX / mangadrp_version / [PLATE] / [MJD] /`. In the above example, the output folder containing the reduced files is `$MANGA_SPECTRO_REDUX/trunk/7341/56693/`. The keyword `/full` specifies to run the full reduction pipeline, as opposed to the quick DOS pipeline. This will reduce exposures through the entire 2d DRP; the outputs of this code are the `mgFlat`, `mgArc`, `mgFrame`, `mgSFrame`, `mgFFrame`, `mgCFrame` files, which contain the 2d row-extracted spectra.

8.6.2. 3d Cube Production

Once the 2d pipeline has been fully run for all MJD, the 3d pipeline can be called (for, e.g., plate 7443) by

```
mdrp_runmanga3d,'7443'
```

This will search the MJD subdirectories for `mgCFrame`-type files and use this to construct the per-IFU RSS and datacube files.

The 3d pipeline can also be run only for individual IFUs (or a subset of them) by specifying, e.g.,:

```
mdrp_runmanga3d,'7443',doifu='12703'
```

8.7. Known Issues

If you're running IDL 8, you might run into errors about 'legend' or 'colorbar' such as

```
attempt to call undefined procedure/function 'legend'
```

See discussion of this at http://www.idlcoyote.com/ng_tips/idl8_name_conflicts.html

To mitigate this the DRP has moved to using the renamed versions `al_legend` and `al_colorbar`

Last modified on 02/01/21 18:19:58