Deep Extragalactic VIsible Legacy Survey

Tool for Analysis and Redshifting (TAZ) Version 0.1

1 Installation

1.1 External Packages

There are three main pieces of external software required to run TAZ. They are i) main coding language R, ii) the 2dfDR data reduction software provided by the the AAO, iii) the AAT targeting/fibre assignment software CONFIGURE.

1.1.1 Installing R

You can download the version of R specific to your operating systems and via your closest mirror here: `https://cran.r-project.org/mirrors.html`. Follow the instructions to install R. You can then check that R is installed correctly, by opening a terminal window and typing:

    % R

If an R session starts in your terminal you are up and working fine.

1.1.2 Installing 2dfDR

The 2dfDR software is used to reduce data from the 2df+AAOmega system at the AAT. Firstly, you can find detailed information regarding the processes in 2dfDR here: `https://www.aao.gov.au/get/document/2dF-AAOmega-obs-manual-69-88.pdf`. In order to install 2dfDR go here: `https://www.aao.gov.au/science/software/2dfdr` and download the appropriate version for your operating system. You will then need to follow the installation instructions.
You will then need to set the location of the installed 2dfDR software to your path by add the following (or similar) to your .cshrc/.bashRC/.profile file. For me this is:

    setenv PATH $PATH\:/Applications/2dfdr-6.46-MacOsX_ElCapitan/bin

For running TAZ we use the command line version of 2dfDR called AAORUN. To check this is running correctly, open a terminal and type:

    % aaorun help

This should display the AAORUN command line options. You can also test the 2dfDR GUI software interface is working, but typing:

    % drcontrol

This should give you a nice image of Saturn and tell you that there are no raw data files in your current directory.

1.1.3 Installing CONFIGURE

The AAT uses a piece of software called CONFIGURE to determine fibre assignments and produce the files required to drive 2df. You can find out about what CONFIGURE does here: `https://www.aao.gov.au/science/software/configure`. On this page follow the online instructions for installing configure.

You will also need to add the path to CONFIGURE:

    setenv PATH $PATH\:/Applications/configure-8.4-MacOsX_ElCapitan_x86_64

and the environmental variable for CONFIGURE's data files:

    setenv CONFIG_FILES /Applications/configure-8.4-MacOsX_ElCapitan_x86_64/data_files/

To your .cshrc, etc. You can then check that CONFIGURE is running correctly by opening a terminal and typing:

    % configure

You should get a pop-up GUI asking which instrument you are using. Don't worry about this, you should not need to do things manually, but TAZ will now be able to call CONFIGURE to allocate fibres for observing.

1.2 Installing R packages

There are a number of R packages that need to be installed. To do this you should just be able to run the INSTALLTAZ.R script provided. First move to the directory where INSTALLTAZ.R is located, open an R session in a terminal, using:

    % R

then run:

    > source('InstallTAZ.R')

This will install all of the required packages required to run TAZ. As a quick test type:

```
> ?TAZ
```

and you should get some helpful advice.

2   Setting Up the DEVILS Directory Structure

The first stage in running TAZ is to set up the the DEVILS directory structure. To do this, run the SETUPDIR function. This function will create the directory structure including all of the calibrations, IDX files (parameters for 2dfDR), and observability plots/logs for each night you are observing. To run this structure, you will need to input the run and night range for which you wish to generate the data structure. First open an R session. Then you will need to load the TAZ pacakge (this should be all you need to load whenever you start R):

```
> library('TAZ')
```

The SETUPDIR function is then run as:

```
> setUpDir(workingDir='.', runs=c('run1_2017_12','run2_2018_01'), $
    dateStart=c('2017_12_18','2018_01_09'),dateEnd=c('2017_12_26','2018_01_22')
```

where,

`workingDir`=directory location you wish to build this structure.
`runs`=the run names. They must have this format. If you do not know the run you are on, ask Luke (luke.j.davies@uwa.edu.au).
`dateStart`=First night of each run in year_month_day format
`dateEnd`=Last night of each run in year_month_day format

Try running this code and explore the directory structure. You should have:

data/
  biases/
    run1_2017_12/, ...
  calibrators/
    AutoZTemp/
    filters/
    GuideStars/
    sensfuncs/
    SkyFibres/
    stdstars/
  darks/
    run1_2017_12/, ...
  idxFiles/
  logs/

```
observing/
    D10_yrPlan2017.png, ....
    run1_2017_12/, ....
        2017_12_18/, ....
raw/
    run1_2017_12/, ....
        2017_12_18/, ....
reduced/
    run1_2017_12/, ....
        2017_12_18/, ....
```

You will populate the **biases/**, **darks/**, and **raw/** part of this structure with the data taken at the AAT, and TAZ will generate reduced data and analysis products in the other parts of the structure.

3 Adding Observations

When observing you will add raw data files to this directory structure. Firstly, at the start of each run, bias and dark frames will be taken. These need to be added to the **biases/*runName*** and **darks/*runName*** directories. Note that TAZ will aim to generate a master bias and master dark for all files in this directory, so if there are files that you do not wish to be used in making the master darks/biases please keep them in the **biases/*runName*/junk/** or **darks/*runName*/junk/** sub folders.

Next you will need to place the raw target data under in the relevant directory. This should be easily done as the date in in the file name produced by the AAT. Data for each night observed should be copped to the relevant **raw/*runName*/*date***. Once again, TAZ will reduce/analyse all data in this directory. So if there are files you do not wish to be included, add then to the **raw/*runName*/*date*/junk** folder. You do not need to separate ARC, FLAT and TARGET files in this directory. TAZ will identify and match the correct files based on the configuration used in the FITS header.

4 Running TAZ

Here I will just give a high level overview of running the TAZ software. More detailed descriptions of the individual functions can be accessed via the R help function by simply opening an R terminal and typing:

```
> ?*FUNCTION_NAME*
```

To run TAZ you will need to run the command similar too:

```
> TAZ(user='ldavies',workingDir='~/DEVILS/TAZ/',verbose=2, $
N_D02A=0,N_D02B=0, N_D03=1, N_D10=1, D02A_startPlate=0, D02B_startPlate=0, $
D03_startPlate=0, D10_startPlate=0, doReduce=T, doExtract=T, $
toExtractFiles=toExtractFiles, doStack=T, toStackIDs=toStackIDs, doAutoZ=T,
$
toAutoZStacks=toAutoZStacks, doUpdateMaster=T, doTiler=T, $
DODir='~/DEVILS/TAZ/data/observing/run1_2017_12/2017_12_18/DOCats/')
```

Below I will briefly explain what each variable in this input means in terms of running the high-level functionality of TAZ:

- `user`=A string identifier which will be added to log files to show who ran the code.
- `workingDir`=The directory location where your previously generated **data/...** data structure is located.
- `verbose`=How much information to give you about TAZ, verbose=0,1,2.
- N_D? = Number of new fibre configurations to generate in each field. Use for next night's tiling
- `D?_startPlate` = Starting plate number for each configuration (either 0 or 1)
- `doReduce` - TRUE/FALSE. Do you want to reduce new data? TAZ will look for new data as files where there is data in the **raw/** directory for a particular night, but nothing in the corresponding **reduced/** directory. For the advanced user, you can force this by running `run2dfDR(toReduce=toReduce)` where `toReduce` is a list of raw directories you want to reduce.
- `doExtract` - TRUE/FALSE. Do you want to extract 1D spectra from the files you have reduced? If you have set `doReduce`=F and `doExtract`=T you will need to provide a list of reduced files to extract.....
- `toExtractFiles` - A string vector list of reduced files for which you wish to extract 1D spectra. Set if `doReduce`=F and `doExtract`=T. These need to be the full directory path in the DEVILS data structure. *i.e.* data/reduced/run2_2018_01/2018_1_9/2018_1_9_config_1_reduced.fits.
- `doStack` - TRUE/FALSE. Do you want to stack extracted spectra? If you have set `doExtract`=F and `doStack`=T you will need to provide a list of spectra IDs to stack.....
- `toStackIDs` - A string vector list of ID for which you wish to stack. Here you just provide the IDs in a string. *I.e.* D10021811.
- `doAutoZ` - TRUE/FALSE. Do you want to run Auto-z over the stacked spectra? If you have set `doStack`=F and `doAutoZ`=T you will need to provide a list of stacked spectra files.....
- `toAutoZStacks` - A string vector list of stacked spectra files for which you wish to run Auto-z. These need to be the full directory path in the DEVILS data structure. *i.e.* data/reduced/stackedSpec/D10021811.Rdata
- `doUpdateMaster` - TRUE/FALSE. Do you want to update the master catalogue with new redshift. This will produce a new master catalogue with the current date (DMcats). And produce the observing catalogues for the next observation data DOcats.
- `doTiler` - TRUE/FALSE. Do you want to generate new tile configurations for the next night of observations?
- `DODir` - If you have set `doUpdateMaster`=F and `doTiler`=T you need to tell the code which observation catalogues to use. Set this to a string with the DOCat directory location. Something like: ~DEVILS/TAZ/data/observing/run1_2017_12/2017_12_18/DOCats/.

For the most common cases you would wish to run something like:

```
> TAZ(user='ldavies',workingDir='~/DEVILS/TAZ/',verbose=2, $
N_D02A=0,N_D02B=0, N_D03=4, N_D10=4, D02A_startPlate=0, D02B_startPlate=0, $
D03_startPlate=0, D10_startPlate=0, doReduce=T, doExtract=T, $
doStack=T, doAutoZ=T, doUpdateMaster=T, doTiler=T )$
```

This will run the code from end to end, finding the new data to reduce, reducing it using 2dfDR, extracting new 1D spectra, stacking new spectra with exisiting spectra, running Auto-z, updating the master catalogue, running the tiling using CONFIGURE, and producing the new tiling files.

4.1 EXAMPLE 1: Manually running 2dfDR using the GUI and extracting spectra, stacking and running AutoZ.

This will be similar to how the observers at the telescope will use this software, so please see the DEVILS_Observing_Manual.pdf for more details. However, I will briefly describe this process here.

Once the data has been reduced using 2dFDR (see DEVILS_Observing_Manual.pdf), there are certain things that need to be done for any other part of TAZ to be run. Firstly, the reduced files need to be given the correct naming convention and be placed in the correct files in the directory structure.

There must be a reduced file for the full spliced data, the blue arm and the red arm (these will be produced from 2dfDR). These must be named:

- Full spliced: *year*_*month*_*day*_config_*#*_reduced.fits. Where the # is the number of the configuration taken on that night (*e.g.* 2017_12_18_config_4_reduced.fits)
- Blue CCD: *year*_*month*_*day*_config_*#*_reduced_blue.fits. Where the # is the number of the configuration taken on that night (*e.g.* 2017_12_18_config_4_reduced_blue.fits)
- Red CCD: *year*_*month*_*day*_config_*#*_reduced_red.fits. Where the # is the number of the configuration taken on that night (*e.g.* 2017_12_18_config_4_reduced_red.fits)

You must then most these files to the correct part of the directory structure:
- Full spliced: **data/reduced/*runName*/*year*_*month*_*day*/** (*e.g.* **data/reduced/run1_2017_12/2017_12_18/**)
- Blue CCD: **data/reduced/*runName*/*year*_*month*_*day*/ccd1/**
- Red CCD: **data/reduced/*runName*/*year*_*month*_*day*/ccd2/**

Once you have done this, you should be able to run TAZ using the following:

```
> ReducList<-c('data/reduced/run2_2018_01/2017_12_18_config_4_reduced.fits')

> TAZ(user='ldavies',workingDir='~/DEVILS/TAZ/',verbose=2, $
N_D02A=0,N_D02B=0, N_D03=4, N_D10=4, D02A_startPlate=0, D02B_startPlate=0, $
D03_startPlate=0, D10_startPlate=0, doReduce=F, doExtract=T, $
toExtractFiles=ReducList, doStack=T, doAutoZ=T, doUpdateMaster=T, doTiler=T
)
```

You only need to give `toExtractFiles` the full spliced frame location and it will find the blue and red arms. In this example, only one file is being extracted. However, you can give `toExtractFiles` a list of files and it will extract all of them. If you only want to extract the spectra and run AutoZ (not update the catalogues/run the tiling), then run:

```
> TAZ(user='ldavies',workingDir='~/DEVILS/TAZ/',verbose=2, $
N_D02A=0,N_D02B=0, N_D03=4, N_D10=4, D02A_startPlate=0, D02B_startPlate=0, $
D03_startPlate=0, D10_startPlate=0, doReduce=F, doExtract=T, $
toExtractFiles=ReducList, doStack=T, doAutoZ=T, doUpdateMaster=F, doTiler=F
)
```

4.2   EXAMPLE 2: Generating new tiling files from an existing catalogue.

Another thing you may commonly wish to do is generate new tiling files without running any of the reduction/analysis. To do this, you simply need to point TAZ to a specific DOCat directory in the **data/observing/** part of the data structure. For example:

```
> DODir<-c('data/observing/run1_2017_12/2017_12_18/DOCats/')

> TAZ(user='ldavies',workingDir='~/DEVILS/TAZ/',verbose=2, $
N_D02A=0,N_D02B=0, N_D03=4, N_D10=4, D02A_startPlate=0, D02B_startPlate=0, $
D03_startPlate=0, D10_startPlate=0, doReduce=F, doExtract=F, $
doStack=F, doAutoZ=F, doUpdateMaster=F, doTiler=T, DODir=DODir)
```

This will generate four new configurations for D03 and four for D10 and put them in the data/observing/run1_2017 _12/2017_12_18/DOCats/Tiling/TileFiles/ directory.

5   TAZ Outputs

TAZ will have generated a large number of output files in different parts of the directory structure. Below is a list of key data products produced by TAZ such that you can easily find things in the data structure:

5.1   Raw Data Meta Information:

/data/raw/*runNum*/*date*/*date*_metaData.Rdata
R data file containing meta data from FITS header of all raw files in the current directory. To view this data, move to the directory in which is is contained, open and R session and type:

```
> load('*date*_metaData.Rdata')

> metaData
```

This will display the filename, ccd, field, RA/DEC centre, exposure time in seconds, the type of data, fibre configuration, date and grating for each file.

5.2   Spliced reduced data for each night/configuration:

/data/reduced/*runName*/*date*/*date*_config_?_reduced.fits
FITS file with the fully reduced and spliced data for a particular fibre configuration.

5.3   Blue arm reduced data for each night/configuration:

/data/reduced/*runName*/*date*/ccd1/*date*_config_?_reduced_blue.fits
FITS file with the fully reduced blue data for a particular fibre configuration.

5.4   Red arm reduced data for each night/configuration:

/data/reduced/*runName*/*date*/ccd2/*date*_config_?_reduced_blue.fits
FITS file with the fully reduced red data for a particular fibre configuration.

5.5 Meta Information for all reduced configurations:

/data/reduced/*runName*/*date*/*date*_metaData.Rdata
Meta information for all reduced data frames. To view this data, move to the directory in which is is contained, open and R session and type:

```
> load('*date*_metaData.Rdata')

> metaData
```

This will display the filename, ccd, field, RA/DEC centre, exposure time in seconds, the type of data, fibre configuration, date and grating for each file.

5.6 Flux calibration meta information for each reduced configuration:

/data/reduced/*runName*/*date*/ccd2/*date*_config_?_reduced_zeroPoints.Rdata
R data file with flux calibration information for each reduced frame. To view this data, move to the directory in which is is contained, open and R session and type:

```
> load('*date*_config_?_reduced_zeroPoints.Rdata')

> zeroPoints
```

This will display:

ZP - Magnitude zero point of the frame calculated from standard stars
ZPMAD - Median average deviation of ZP
ZPRMS - RMS of ZP
ZPNUM - Number of standards used to calculate ZP
FLUXSC - The flux scaling to convert counts to ergs/sec/cm^2/Ang/
ZP_red... - Same values for red and blue arm data individually.

5.7 Individual extracted spectra from corresponding reduced configuration/night:

/data/reduced/*runName*/*date*/ccd2/*date*_config_?_reduced_spec/
Directory containing all individually reduced spectra from the corresponding configuration. This directory will also contain 1D plots if requested in TAZ. Each spectrum is stored as an R data structure. To view this data, move to the directory in which is is contained, open and R session and type (for example):

```
> load('2017_12_18_G2271755.Rdata')

> spec
```

This structure contains various data and meta information for this observation, such a flux, wavelength, sky. It also retains the individual blue and red arms of the spectrum prior to splicing. Note that these are not flux calibrated, but contain the flux calibration information to convert pixels to ergs/sec/cm^2/A

(fluxSc). To view the spectrum simply type:

```
> magplot(spec$wave, spec$flux, type='l')
```

5.8 Individual spectra meta information for each configuration/night:

/data/reduced/*runName*/*date*/ccd2/*date*_config_?_reduced_meta.Rdata
R data file containing the meta information for each spectrum row in the corresponding reduced file.
To view this data, move to the directory in which is is contained, open and R session and type:

```
> load('*date*_config_?_reduced_meta.Rdata')

> meta
```

This will display:

ID - Source ID from input catalogue
RA - Fibre RA
DEC - Fibre DEC
X - Fibre X
Y - Fibre Y
XERR - Fibre X error
YERR- Fibre Y error
THETA - Fibre angle
TYPE - Type of source (Target, sky, standard, guide)
MAG - Magnitude of source from input cat
FIBRE - Fibre Number

6 TAZ data flow and components