HETDEX Data Release 1:
Emission Line eXplorer (ELiXer)
User's Guide

Introduction

The Emission Line eXplorer (ELiXer) is a diagnostic/debugging tool that, along with a limited API, is included as part of the HETDEX data release. ELiXer does not perform detections on its own, but primarily aggregates HETDEX observation data and external photometric catalogs to facilitate the examination of emission line detections and aid in line classification.

Caveats

Runtime Environment

ELiXer was developed and tested against Python 2.7.x under limited desktop and supercomputing environments. It is forward compatible with Python 3.6.x and 3.7.x, but has not been extensively tested with that family of Python kernels. That said, it may be desirable to use a Python 3x kernel if executing on certain TACC clusters (see below).

ELiXer is a single threaded at the application layer, however there are multi-threaded packages included (such as emcee).

ELiXer does support bulk execution on several Texas Advanced Computing Center (TACC) supercomputing clusters (namely maverick (soon to be decommissioned), stampede2, and wrangler) via SLURM batch scheduling.

The base memory footprint is relatively small (~ 200 MB), however, some photometric imaging files are very large and, even with lazy loading, the memory requirements can briefly exceed 15 GB and, depending on the timing of garbage collection, execution with the Python 2.7 kernel may encounter memory allocation issues. In key areas, ELiXer attempts to detect this condition and implements random sleeps and limited retries in simple attempt at remediation. This does not appear to be an issue when using the Python3 kernel. At the time of this writing, testing under Python3 is limited but there are no known compatibility issues (excepting future deprecation warnings).

Line Classification

Classification, where possible, is supplied by the ELiXer tool and consists primarily as two independent approaches. First, ELiXer, in turn, assigns the detected emission line each of two dozen potential classifications (see table below) and examines the spectra for the other lines from the same set (fitting (by least squares) a Gaussian to the position where the other line(s) would be expected, allowing for up

to +/- 2 AA error in the line center to roughly allow for velocity offsets (particularly with Lyα) and measurement error). All combinations are scored by SNR, line width, integrated line flux, and offset from the expected line center. At this time, absorption features are not used. If exactly one solution scores more than 50% of the total score of all solutions and is greater than a minimum acceptable score, it is marked as the probable classification via this method. In all other cases, no classification is assumed.

Second, a Bayesian posterior ratio of the probability that the line is Lyα to the probability that the line is [OII], [noted generally as P(LAE)/P(OII)] is computed based on Leung et al 2016 (Leung, Andrew S, et al 2016, "BAYESIAN REDSHIFT CLASSIFICATION OF EMISSION-LINE GALAXIES WITH PHOTOMETRIC EQUIVALENT WIDTHS", arXiv:1510.07043v2). This does not examine the possibility of any other line classifications and only compares P(Lyα) to P([OII]) on the line flux and estimated equivalent width. The continuum estimate for the equivalent width comes from several sources and, as such, a P(LAE)/P(OII) ration is produced for each source of the continuum estimate. The first continuum estimate comes from the HETDEX extracted 1D spectra, but with a magnitude limit ~22, this is often an upper limit on the continuum. When photometric imaging is available, a second continuum estimate is made using preferentially an r-band filter (or g-band, if r-band is not available). In this case, a 1.5" radius aperture is centered on the RA and Dec of the HETDEX detection and a magnitude is calculated inside that aperture using the Python photutils package. Local sky is subtracted from an annulus 7.5" – 15.0" from the center. The magnitude is calculated using the imaging's reported zero point and assuming a point-source. Due to the non-uniformity of the imaging (different instruments, different calibration techniques, etc) these are not ideal assumptions and the calculated magnitude could have significant error. Lastly, a third continuum estimate is made for up to the top three (nearest, within, by default, 3") photometric catalog reported detections using the catalog's reported magnitude (again, in r-band preferentially) for the detection. No validation is performed on the catalogs and the reported magnitude is taken at face value. For HDR1, the separation to the HETDEX detection does not include an estimate of the spatial extent of the detections and again assumes point-sources. Each of these P(LAE)/P(OII) ratios is computed and reported independently and no attempt is made to combine them. Where no catalog detections or imaging is available or the code does not converge on a ratio, none is provided.

Lyα (1216)	Hβ (4863)	NV (1241)	NaI (4980,5153)
OII (3727)	Hγ (4342)	SiII (1260)	CaII (3935)
OIII (4960,5008)	Hδ (4102)	HeII (1640)	
CII (2326)	Hε (3970)	NeIII (3869,3967)	
CIII (1909)	Hζ (3889)	NeV (3347)	
CIV (1549)	Hη (3835)	NeVI (3427)	

Table 1 ELiXer Emission Lines (to nearest Å)

The photometric imaging catalogs currently used by ELiXer come from a variety of sources and instruments and have not been adapted or calibrated for HETDEX needs. Please see the README under the imaging directory for a description of the imaging catalogs possible limitations (`/work/03946/hetdex/hdr1/imaging/README`).

Installation

ELiXer, itself, requires no installation, but does have a number of dependencies that must be installed in order to run the report generation. Additionally, a Python package exposing a limited API is available for local installation via pip (described later). The recommended setup follows immediately with alternatives after.

Recommended Setup

It is recommended that you add the following line to your `.bashrc` file:

```
export PATH="/home/00115/gebhardt/anaconda2/bin:/home/00115/gebhardt/bin:
/work/03946/hetdex/hdr1/software/elixer/bin:$PATH"
```

This will provide you with a common Python and easy access to ELiXer bash wrappers. If you are only running ELiXer on a TACC cluster, this is the simplest approach.

Alternate Setup

Alternatively, you may use your own Python installation and either run ELiXer from the HDR1 directory as above or copy the ELiXer folder to your own location and run from there. As noted above, ELiXer for HDR1 was tested on Python 2.7.x and support for Python 3.x is not guaranteed.

A pip installable package is available under:

```
/work/03946/hetdex/hdr1/software/elixer/install
```

You may copy the package file (version 1.6.2 at the time of this writing) and run:

```
>pip install --user hetdex-elixer-1.6.2.tar.gz
```

This will install ELiXer under your user local Python site package directory and add the package to Python's search path. Since this is for HETDEX use only, the package is not committed to PyPI and must be installed using the tar.gz file.

Alternately, you can copy the entire `elixer` directory to a location of your choice and add that location to your PATH variable (as in the export statement example above). If you wish to access the package APIs, you will need an additional step in your Python script (shown later), but if you are only going to run the report generation, you may do so without an additional runtime step.

You may need to also install several additional Python packages, though many are typically included in Python (and anaconda) installs.

- Common Python packages (should be included in Python 2.7.15 w/o additional installation): `numpy, matplotlib, pylab, argparse, sys, os, distutils, glob, shutils, socket`
- Less common packages: (install with "`pip install --user xxx`" where xxx is the package name): `astropy, ConfigParser (`or `configparser,` for Python 3.x`), pandas, photutils, scipy,` and `tables`

- One additional HETDEX specific package, `pyhetdex`, is also required: ("`pip install --user --extra-index-url https://gate.mpe.mpg.de/pypi/simple/ pyhetdex`")

A simple check to see if ELiXer is properly installed is to request the help:

```
> python /<path to elixer>/elixer.py --help
```

If the installation is complete, ELiXer will print to screen its basic parameter help information. If one or more packages are missing, you will be presented with an error statement identifying what needs to be installed.

Running ELiXer (generating reports) on TACC

ELiXer is intended to be run in one of two modes ... single instance and batch (via SLURM). If you are running on your own desktop, you must use the single instance mode, but if you are running on a TACC cluster you should normally use the batch or SLURM mode (TACC discourages the execution of high cost tasks from the login nodes).

The basic command line for the two modes are almost identical, with the SLURM mode supporting a few additional parameters related to the SLURM mechanism.

In the single instance mode, ELiXer will serially process each provided detection. In the SLURM mode, ELiXer will spawn multiple instances of itself, dividing the detections across the instances, but still processing serially within any given instance.

Run time varies with server, load, and proximity to the data, but you can generally assume approximately one minute per detection report.

You may launch the ELiXer process with a Python call: e.g.

(single instance version)

```
> python /work/03946/hetdex/hdr1/software/elixer/elixer.py --help
```

-or-

(SLURM version: notice 'selixer.py' instead of 'elixer.py')

```
> python /work/03946/hetdex/hdr1/software/elixer/selixer.py --help
```

-or-

via the bash wrappers if you added the paths to your PATH environment variable (as shown in the Installation section). If you copied the `elixer` directory or otherwise installed locally, you will want to edit the bash wrappers to set the correct path. The wrappers are in the 'bin' directory under the `elixer` directory*.

```
> elixer --help
```

-OR-

```
> selixer –help
```

The "--help" switch will print to screen a simple break out of the command line options. Also NOTE that the SLURM version will NOT spawn multiple instances if the "--help" switch is on the command line (in this case, `elixer` and `selixer` are equivalent (the other case is with the `--merge` switch described later).

* note: There can be environmental issues when running on TACC using the .bash script wrappers. Specifically, you must make sure than the SLURM environment loads the Python configuration you

expect. Typically, errors reporting missing packages when executing under SLURM (but no errors when running with "—help" under the login node) indicate an environment mismatch. If you are not comfortable debugging, it may be simpler in that case to launch ELiXer with the full Python call (noting that on some TACC clusters multiple versions of Python are available and you may need to load a module and/or specify which kernel to use, e.g. :

```
> python2 /work/03946/hetdex/hdr1/software/elixer/selixer.py …
```

-or-

```
> python3 /work/03946/hetdex/hdr1/software/elixer/selixer.py …
```

Common Usage Examples

Although there are many options, ELiXer is anticipated to be used in only a few ways with HETDEX Data Release 1. Essentially, you will either provide an RA, Dec and search radius or a list of detection IDs and ELiXer will produce a report for each.

The following examples will assume the SLURM version invocation using the bash wrapper. If you are not using the bash wrappers, simply replace `selixer` with the Python invocation:

```
python /<path to elixer>/selixer.py
```

EXAMPLE 1

```
> selixer --recover --ra 150.025406 --dec 2.087600 --error 2.0 --name
example1 --tasks 0 --email yourname@utexas.edu
```

Here an --ra and --dec are provide (in decimal degress ... however, hms and dms notations will also work, e.g.: --ra 10h00m6.10s --dec 2d05m15.36s are equivalent).

--recover is a switch that instructs ELiXer to run each detection to completion before starting the next one. This allows ELiXer to be run a second time, using the exact same command, if the first command timed out in the queue and it will resume where it left off and only process detections for which a report does not yet exist.

--error is the radius in which to search from the given RA and Dec and is ALWAYS in arcsecs.

--name is the output directory name under which the results will be written.

--tasks 0 specifies that ELiXer should set the number of instances to spawn based on the cluster on which it is running. You may override this and supply a non-zero value to force ELiXer to use no more than that number of tasks.

--email is entirely optional, but will generate emails to the supplied address when the SLURM job actually begins (when it exits the wait queue) and when it completes or ends via error.

Additional, optional commonly used command line switches:

--time : supply a maximum hh:mm:ss runtime for the SLURM job (if not supplied, ELiXer will calculate a value)

--queue : specify which queue to use to execute the SLRUM job on the cluster (if not supplied, ELiXer will choose a queue)

Example 2

```
> selixer --recover --tasks 0 --email yourname@utexas.edu --dets
detlist --error 2.5 --name example2
```

Here, --dets detlist refers to a file named detlist that contains a list of detectionIDs, one per line.

Obviously, you need to know the detectionIDs in advance (which may well be the case if you are already working with them). This will actually allow detectids (like: 1000000318) OR inputids (in Datevshot_# format, like: 20180123v009_4) OR all shots for a particular observation (like: 20180123v009)

This may also be a comma separated list (no spaces) like:

--dets 1000000318,1000000330,1000000414
or
--dets 20180123v009_4,1000000371

(note that mixed types are allowed)

Output

If you run the single instance of ELiXer, all the output will be immediately under a directory named for the --name switch.

If you run the SLURM version of ELiXer, there will be a series of dispatch_xxx directories (where xxx is a number) under the directory named for the --name switch and under each of those will be a log file and a directory named for the datevshot of the detection. Under this second nested directory will be the output files for the detections.

The output files (excluding files created due to other options that may be supplied on the command line) at a minimum, will consist of a report PDF (named after the detection) and two catalog files, also named after the detection and terminated with *_cat.txt and *_fib.txt.

The PDF reports graphically represent the basic information about the detection including information on the fiber 2D spectra, the fit of the emission line, the full summed/weighted spectra, photometric imaging (if available) and potential catalog matches (if available).

The two catalogs summarize this data in text catalog format with the *_fib.txt file covering each fiber in the HETDEX data and the *_cat.txt catalog focusing on the photometric catalog matches. Each begins with a header than describe the columns.

The catalogs are fragmented under each of the dispatch_xxx directories, but you can combine them all into one catalog (each) by running `selixer --merge` (or `elixer --merge`) at the parent directory of the dispatch_xxx folders.

A description of how to interpret the individual PDF reports is presented later in this document.

Running ELiXer (generating reports) on Your Local Box

You may also wish to run ELiXer directly from your own desktop. This can be useful when running only a few detections and for debugging with ELiXer and/or its API.

The individual runtime may be longer since the data will be remote and accessed via the Internet.

For this option to be effective, you will most likely want to mount the TACC `/work` directory as a remote file system using `sshfs`.

Briefly, on a linux OS, to mount a remote file you would:

1) Create a local directory as a mount point. In order to mimic the TACC structure, it is recommend you (on your local machine) use `/work` or make a soft-link as `/work` and point to the mount point of your choice.
2) Issue the following command:
   > `sshfs yourname@corral.tacc.utexas.edu:/work ~/tacc/work -C -o ServerAliveInterval=30,ServerAliveCountMax=5,noatime`

   Where:
   `yourname@corral.tacc.utexas.edu:/work` = your login to TACC and the remote directory you want to mount. You may substitute wrangler, stampede2, etc for 'corral'.

   `~/tacc/work` = the path to your local mount point. In this case, it is under the user's home directory and a soft-link at /work should be created. (e.g. from the root directory (/) issue this command:

   ```
   sudo ln -s ~/tacc/work /work
   ```

   The other options instruct sshfs to allow compression (-C) and send keep alives every 30 seconds and allow 5 timeout responses before giving up.

To close your mount, issue the following command:

```
fusermount -u ~/tacc/work
```

With the remote file system mounted as above, you can issue ELiXer invocations or use the API with the same paths as if you were running on a TACC cluster.

Using the API

This ELiXer version exposes two limited Python APIs that supports accessing the early photometric catalogs as well as the P(LAE)/P(OII) computation.

If you copied the ELiXer source directory or are referencing the HDR1 source path, you must add that path to your system path so that Python can locate the APIs. To do so, add the following code to the top of your Python script:

```
import sys
sys.path.append('/work/03946/hetdex/hdr1/software/elixer')
import catalogs
import classify
```

Be sure to substitute the correct path to the elixer source code if you copied it locally. If you installed ELiXer with pip, use the following code instead:

```
from elixer import catalogs
from elixer import classify
```

For basic help on each API, you may execute (from within a Python interpreter):

```
help(catalogs)

help(classify)
```

The `catalogs` package provides access to the photometric imaging data and catalogs as well as basic aperture photometry. The `classify` package provides a wrapped interface to the Bayesian P(LAE)/P(OII) ratio computation. Future releases may expand the APIs to expose additional ELiXer functions (such as 1D spectra line identification, etc).

Example iPython Notebooks are provided under the docs folder in the elixer source directory.

Interpreting the Report

As described at the beginning of this document, ELiXer is a diagnostic/debugging tool and is currently intended only to aid in informing expert line identification. Upcoming enhancements in future data releases will provide more authoritative classifications.

Please see the example PDF reports at the end of this document to match up with the following descriptions:

1. (upper right) version information – the time stamp when this file was created and the version of ELiXer used to create it
2. (upper left) Basic HETDEX information.
    - The first row is local identification string (usually of the form DatevShot_relativeID)
    - Obs: DatevShot_HETDEX-ID
    - Entry# (Global HETDEX-ID), Detect ID (local ID within an instance)
    - Primary IFU - identifies the SpecID and IFU Slot ID of the IFU that hosts the highest weighted fiber in the detection
    - RA, Dec - the HETDEX-astrometry weighted centroid position … assumes point source)
    - λ (the observed wavelength center of the detection), FWHM (estimated full width half max of the detection)
    - EstFlux - estimated flux, cgs units
    - EstCont - estimated continuum , cgs units … note, HETDEX mag limit ~ 22, so this is often a ceiling
    - EW_r(LyA) – estimated rest frame equivalent width, assuming Lyα and using the HETDEX estimated flux and continuum
    - S/N – SNR of the detection, χ^2 – Fit to Gaussian
    - P(LAE)/P(OII) – the Bayesian ratio [Leung 2016] using the HETDEX EW_r(LyA) and EstFlux
    - LyA z – the redshift assuming a Lyα identification , OII z – the redshift assuming an OII identification
    - [optional] Possible line identification IF multiple emission lines are matched
3. (top center-left) Cutouts from the HETDEX dataframe(s) in reverse (dark = high count) that cover the wavelength region and fibers of the emission detection. The top most row is a sum of the lower (up to) four rows. Each of the lower (up to) four rows represents one fiber (multiple exposures may be present), but is approximate to 3 fibers tall on the CCD. The first column is from the reduced science frame (green pixels represent masked cosmic ray strikes). The second column is from the pixel flat and the third is a Gaussian smoothed representation of the first column. The text to the left is a (normalized to 1.0) weight of the fiber (the highest weighted fibers are displayed in descending order). The text to the right (you may have to zoom in to read it) shows additional data (the distance in arcsec to the detection center, the X,Y position if viewing in DS9, the date_shot_exposure#, and the IFUSlot_Amp_Fiber#). Each plot is centered on the fiber's (interpolated) fractional pixel corresponding to the detected emission line wavelength. The border is colored to match with panels (4) and (7). Green colored pixels are cosmic ray or other defects.
4. (top center-right) A cutout of the CCD region around the main (highest weighted, most central) fiber (the border is blue, color coded to match). This is intended to provide a view on the CCD to

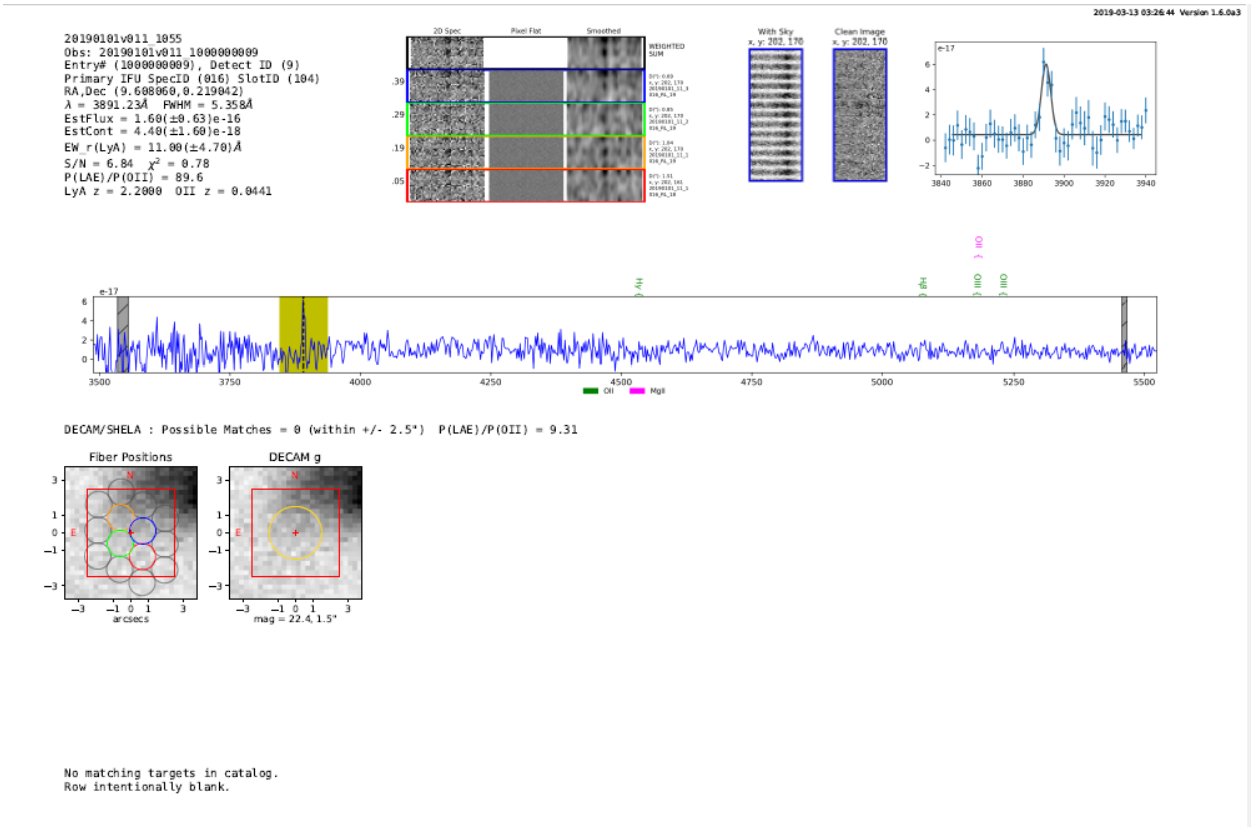
check for non-astrometric scattered light or other anomalies. The left of the two images has not had the sky subtracted where the right image has been sky subtracted. The x,y values under the title are the DS9 pixel coordinated of the center of the frame.

5. (top right) A zoomed scatter plot of the weighted and summed 1D spectrum of the detection over plotted with the best fit Gaussian. In the next frame, this is the same region as that highlighted in gold.
6. (center full width spectrum) A full width plot of the weighted and summed 1D spectrum of the detection. The wavelength positions of known strong skylines are highlighted in gray (they should be subtracted from the data and the highlights only indicate a position, not a presence of residual light). If we assume the peak to be the emission line indicated in the legend beneath the plot, then we could see other lines indicated by labels above the plot, color coded to match the legend below. For example, if we assume the emission line to be Lyman Alpha (red) then other lines we might see are indicated by the labels in red. [optional] Colored highlights (other than the gray and gold already mentioned) may be present over additional, possible emission lines. The color matches the associated line classification below the plot (i.e. red = the additional lines are associated with the main line classified as Lyα).
7. (center imaging) Summary of catalog information, including:
    - [optional, for some catalogs] 3 sigma non-detection estimate of the minimum EW for Lyα and OII based on the depth of the continuum estimate band.
    - [optional, if a magnitude was calculated] P(LAE)/P(OII) Bayesian ratio calculated from the HETDEX estimated flux and the continuum estimated from an aperture magnitude (preferentially of r-band) as described in the Line Classifications section at the beginning of this document.
    - The first cutout shows the fiber positions to scale (colors match those in 3, with gray fibers representing other contributing fibers beyond the top four) overlaid on a stacked image (weighted by exposure time). Fiber circles with circumscribed dashed lines indicate fibers that lie on the periphery of the IFU. The red box shows the size of the input error window (the --error parameter) and indicates celestial north.
    - The remaining cutouts represent [optional] supplemental (different catalog) imaging and then one or more available filters (the cutout title identifies the catalog and/or filter). The small colored boxes mark the positions of catalog identified targets that could be the emission line source (based on angular separation to the target center IF it falls within the red search box). Warning! As noted in the Caveats section at the beginning of the document, spatial extent is NOT yet considered. A gold circle represents the position and extend of the aperture used to calculate a magnitude (not show is the annulus used to calculate local sky … see the Line Classifications section). If a an aperture magnitude is calculated, its value and the radius of the annulus is printed below the cutout.
8. (bottom) Summary of information for up to three catalog targets. The colors match the colored boxes in the previous frame (7).
    - Separation – angular separation (in arcsecs) between the HETDEX position and the reported (center) position in the catalog. Again, spatial extent is NOT considered at this time.

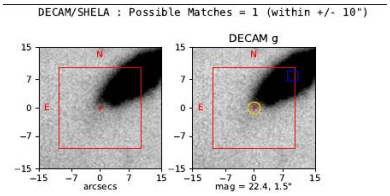
- 1-p(rand) – a limited estimate of the probability that the catalog target is a random match based on the magnitude of the target and the distribution of similar magnitudes in increasing distance annuli from randomly sampled positions. This is a sorting measure only and is predominantly based on angular distance.
- RA, Dec – the reported RA, and Dec from the catalog
- Spec z – if present, the reported spectroscopic redshift
- Photo z – if present, the reported photometric redshift
- Est LyA rest-Ew – the rest frame equivalent width estimate assuming the line is Lyα with the line flux from HETDEX and the continuum estimate from the catalog
- Est OII rest-Ew – the rest frame equivalent width estimate assuming the line is OII with the line flux from HETDEX and the continuum estimate from the catalog
- mag – the catalog reported magnitude with filter name
- P(LAE)/P(OII) - Bayesian ratio calculated from the HETDEX estimated flux and the continuum as reported from the catalog as described in the Line Classifications section at the beginning of this document.

Some Examples

Below is a report that indicates an LAE, but is most likely just in the outskirts of a nearby galaxy. A planned future enhancement will seek to include spatial extent and improve this classification.

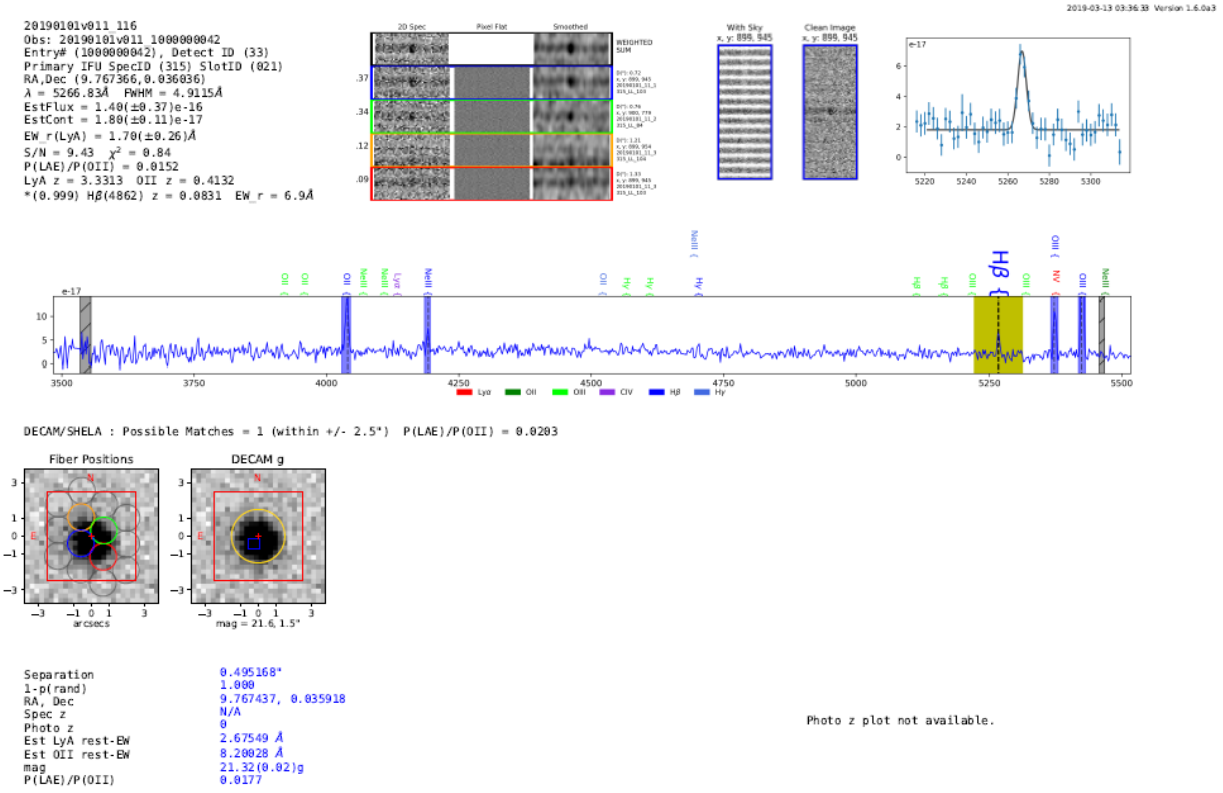


If we re-run with a larger search radius, this becomes obvious (below is a cutout of just the bottom section of the report with a +/- 10” search):



Separation	12.2481"
1-p(rand)	0.994
RA, Dec	9.605460, 0.221236
Spec z	N/A
Photo z	0
Est LyA rest-EW	N/A
Est OII rest-EW	N/A
mag	16.94(0.00)g
P(LAE)/P(OII)	

Below is an example of an object with multiple strong emission lines. This same object may appear in the HDR1 catalog multiple times (once for each of strong emission lines).

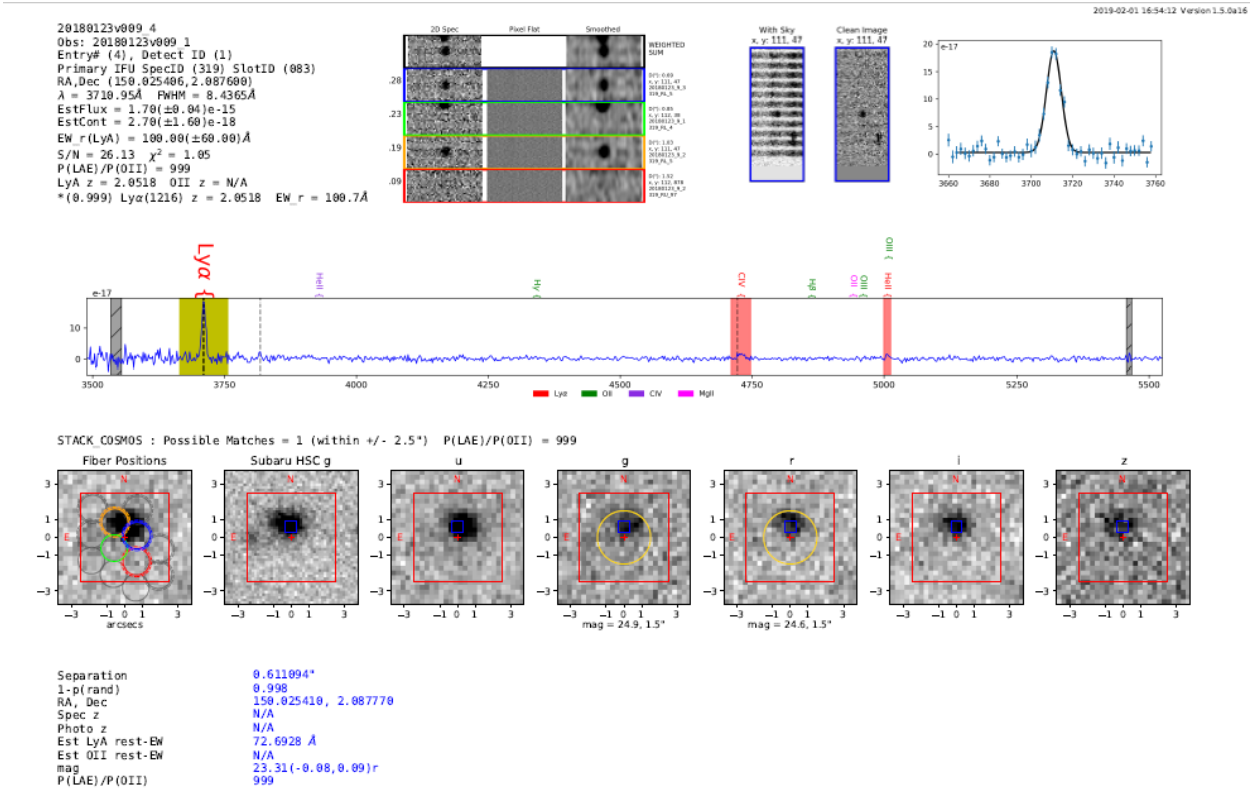


DECAM/SHELA : Possible Matches = 1 (within +/- 2.5") P(LAE)/P(OII) = 0.0203

Separation	0.495168"
1-p(rand)	1.000
RA, Dec	9.767437, 0.035918
Spec z	N/A
Photo z	0
Est LyA rest-EW	2.67549 Å
Est OII rest-EW	8.20028 Å
mag	21.32(0.02)g
P(LAE)/P(OII)	0.0177

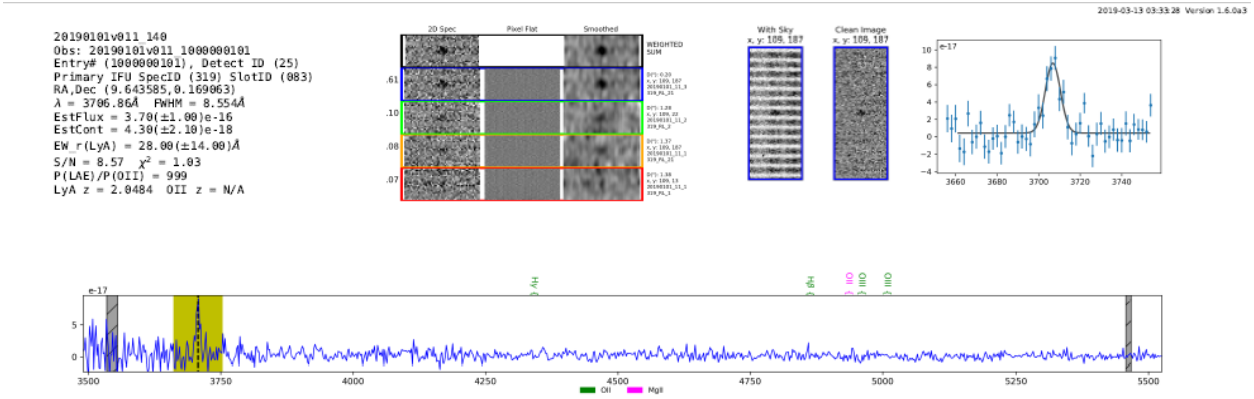
Photo z plot not available.

Here is a large EW LAE. Probably an AGN.

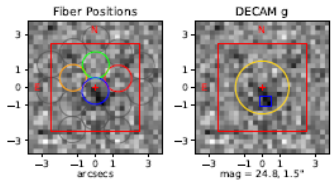


Separation	0.611094"
1-p(rand)	0.998
RA, Dec	150.025410, 2.087770
Spec z	N/A
Photo z	N/A
Est LyA rest-EW	72.6928 Å
Est OII rest-EW	N/A
mag	23.31(-0.08,0.09)r
P(LAE)/P(OII)	999

Here are more typical probable LAEs.

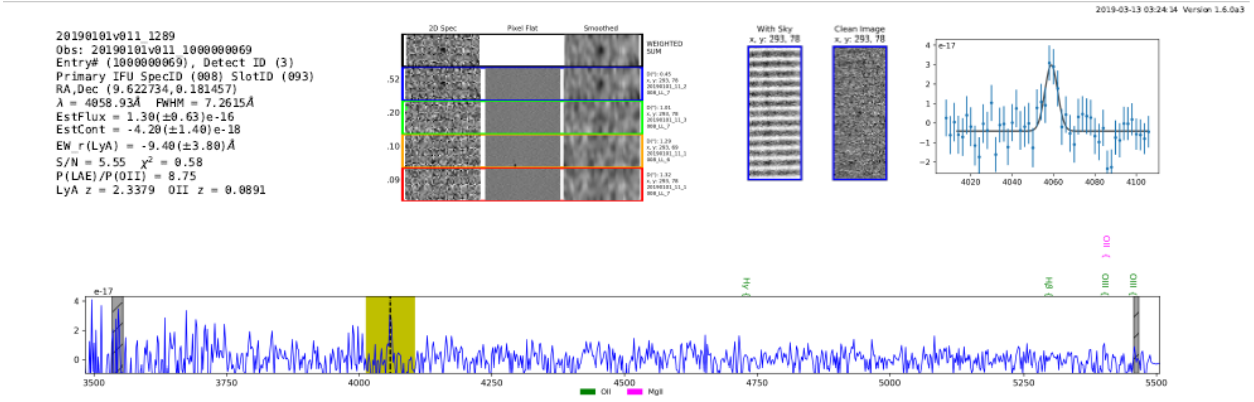


DECAM/SHELA : Possible Matches = 1 (within +/- 2.5") P(LAE)/P(OII) = 999

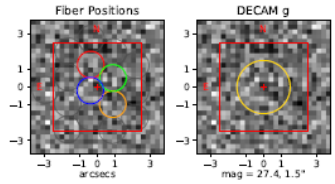


Separation	0.800041"
1-p(rand)	0.986
RA, Dec	9.643534, 0.168847
Spec z	N/A
Photo z	0
Est LyA rest-EW	132.774 Å
Est OII rest-EW	N/A
mag	24.86(0.16)g
P(LAE)/P(OII)	999

Photo z plot not available.



DECAM/SHELA : Possible Matches = 0 (within +/- 2.5") P(LAE)/P(OII) = 999



No matching targets in catalog.
Row intentionally blank.