

---

# **pyigm Documentation**

***Release 0.1***

**Prochaska, and Associates**

**Nov 12, 2017**



---

## Contents

---

<b>1</b>	<b>Contents:</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>69</b>



pyigm is an astropy-affiliated package developed to provide software useful for research on the Intergalactic Medium (IGM). Its core developers work primarily on absorption-line research with emphasis on the IGM and related areas.



### Overview

## 1.1 Installation

### 1.1.1 Dependencies

pyigm depends on these packages:

- [python](#) versions 2.7, or 3.3 or later
- [numpy](#) version 1.12 or later
- [astropy](#) version 1.3 or later
- [scipy](#) version 0.18 or later
- [matplotlib](#) version 1.4 or later
- [specutils](#) version 0.2 or later
- [linetools](#) version 0.2 or later
- [h5py](#) version 2.5 or later
- PyQt5 – Now the default in conda

We strongly recommend that you use [Anaconda](#) to install them. With Anaconda you can check for the presence and versions of the dependencies with:

```
conda list "^python$|numpy|astropy$|scipy$|matplotlib|pyyaml|specutils"
```

If you're missing any, install them with (for example):

```
conda install astropy matplotlib
```

If their versions are too old, update them with (for example):

```
conda update astropy
```

Specutils can't be installed with conda; instead it needs to be installed using [pip](#):

```
pip install --no-deps specutils
```

This is also true for linetools. If you aren't using Anaconda then all of the dependencies can also be installed with pip. The following packages are required only for MCMC analysis of metallicity PDFs (e.g. Fumagalli+16):

- [corner](#) v0.3 or later
- [mpmath](#) version 0.19 or later
- [emcee](#) version 2.1 or later

### 1.1.2 Installing pyigm

If you plan to play around with the code and possibly contribute changes, then follow the instructions in the section below, *Installing pyigm from Source*. Otherwise simply use:

```
pip install --no-deps git+https://github.com/pyigm/pyigm.git
```

and you're done!

### 1.1.3 Installing pyigm from Source

#### *I just want to play with the code*

Install the development version like this:

```
git clone https://github.com/pyigm/pyigm.git
cd pyigm
python setup.py develop
```

Now you can easily make tweaks to the code, which are immediately applied to your installed version (you'll have to reload the relevant modules to see those changes in an existing Python session, though).

#### *I want to make a code contribution to pyigm*

Fantastic! In that case, follow the [Astropy developer guidelines](#), replacing every instance of *astropy* in those instructions with *pyigm*. This will install a 'fork' of pyigm that you can use to submit code changes to the main repository.

### 1.1.4 Running Tests

If you install pyigm from source, then you can run tests to see whether your installation works correctly. From the source directory run:

```
python setup.py test
```

This takes a couple of minutes to run. If you notice any failures, we'd love you to report them on the [pyigm issue tracker](#).



## 1.1.5 Building Documentation

Only do this if you're a developer! If you want build the documentation, you also need to install Sphinx (version 1.3+) and `astropy_helpers`:

```
conda install sphinx
pip install astropy-helpers
```

If you want the generate inheritance diagrams in the docs then you also need to install graphviz ([MacOSX](#), [Ubuntu](#)), but this isn't required. Once you've installed the dependencies, change to the `/docs` directory under the source directory and run:

```
make html
```

The documentation should now be in `_build/html`.

## 1.2 Miscellaneous

### 1.2.1 Methods

#### DM

This method calculates the Dispersion Measure through the IGM, e.g. [Ioka+03](#). Specifically, we calculate (numerically):

$$\text{DM}_{\text{IGM}} = \frac{3cH_o\Omega_b}{8\pi Gm_p} \int_0^z \frac{(1+z)dz}{[\Omega_m(1+z)^3 + \Omega_\Lambda]^{1/2}}$$

Trivially performed with:

```
from pyigm.fn import tau_eff as pyteff
DM = pyteff.DM(z)
```

## 1.3 The pyigm API

### IGM/CGM Datasets

## 1.4 CGM Classes

### 1.4.1 Notebooks

#### CGM Examples (v2.1)

```
%matplotlib inline
```

```
# import
# imports
try:
    import seaborn as sns; sns.set_style("white")
```

```
except:
    pass

from bokeh.io import output_notebook, show, hplot, output_file
from bokeh.plotting import figure
from bokeh.models import Range1d
output_notebook()

from linetools import utils as ltu

from pyigm.abssys.igmsys import IGMSys
from pyigm.field.galaxy import Galaxy
from pyigm.cgm.cgm import CGM, CGMAbsSys
from pyigm.cgm.cos_halos import COSHalos
from pyigm.cgm import cos_halos as pycsh

from pyigm.cgm.cgmsurvey import CGMAbsSurvey
```

```
/Users/xavier/anaconda/lib/python2.7/site-packages/matplotlib/__init__.py:872:
↳UserWarning: axes.color_cycle is deprecated and replaced with axes.prop_cycle;
↳please use the latter.
    warnings.warn(self.msg_depr % (key, alt_key))
```

## Instantiate CGM

```
radec = (125*u.deg, 45.2*u.deg)
gal = Galaxy(radec, z=0.3)
cgm = CGM(gal)
cgm
```

```
<CGM: 08:20:00 +45:12:00, z=0.3>
```

## Instantiate CGMAbsSys

```
radec_qso = (125*u.deg, 45.203*u.deg)
igmsys = IGMSys(radec_qso, gal.z, [-500, 500]*u.km/u.s)
```

```
cgmabs = CGMAbsSys(gal, igmsys)
cgmabs
```

```
/Users/xavier/local/Python/pyigm/pyigm/cgm/cgm.py:125: UserWarning: cgm.CGMAbsSys:
↳Using WMAP9 cosmology
    warnings.warn('cgm.CGMAbsSys: Using WMAP9 cosmology')
```

```
<CGMAbsSys: J0820+4512_180_11 Galaxy RA/DEC=08:20:00+45:12:00, zgal=0.3, rho=48.7208
↳kpc>
```

## Instantiate CGMAbsSurvey

```
cgmsurvey = CGMAbsSurvey(survey='cos-halos', ref='Tumlinson+13, Werk+13')
```

## COS-Halos

### Load

```
reload(pycsh)
cos_halos = pycsh.COSHalos()
```

```
WARNING: UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS standard.
↳Suggested: 10**-1 nm. [astropy.units.format.utils]
WARNING:astropy:UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS
↳standard. Suggested: 10**-1 nm.
```

```
Be patient, using /Users/xavier/Dropbox/COS-Halos-Data/cos-halos_systems.v1.1.tar.gz
↳to load COS-Halos
Skipping a likely folder: CGM_JSON
```

```
cos_halos.name
```

```
array([u'J0226+0015_268_22', u'J0401-0540_67_24', u'J0803+4332_306_20',
       u'J0910+1014_242_34', u'J0910+1014_34_46', u'J0914+2823_41_27',
       u'J0925+4004_196_22', u'J0928+6025_110_35', u'J0935+0204_15_28',
       u'J0943+0531_106_34', u'J0943+0531_216_61', u'J0943+0531_227_19',
       u'J0950+4831_177_27', u'J1009+0713_170_9', u'J1009+0713_204_17',
       u'J1016+4706_274_6', u'J1016+4706_359_16', u'J1112+3539_236_14',
       u'J1133+0327_110_5', u'J1133+0327_164_21', u'J1157-0022_230_7',
       u'J1220+3853_225_38', u'J1233+4758_94_38', u'J1233-0031_168_7',
       u'J1241+5721_199_6', u'J1241+5721_208_27', u'J1245+3356_236_36',
       u'J1322+4645_349_11', u'J1330+2813_289_28', u'J1342-0053_157_10',
       u'J1342-0053_77_10', u'J1419+4207_132_30', u'J1435+3604_126_21',
       u'J1435+3604_68_12', u'J1437+5045_317_38', u'J1445+3428_232_33',
       u'J1514+3619_287_14', u'J1550+4001_197_23', u'J1550+4001_97_33',
       u'J1555+3628_88_11', u'J1617+0638_253_39', u'J1619+3342_113_40',
       u'J2257+1340_270_40', u'J2345-0059_356_12'],
      dtype='<U17')
```

```
cos_halos.cgm_abs[4].igm_sys
```

```
<IGMSysSystem: IGMSysSystem 09:10:29.8 10:14:13.6, 0.142738, NHI=14.7577, Z/H=-0.7>
```

### Simple output

```
cdict = cos_halos.cgm_abs[10].to_dict()
cdict.keys()
```

```
['cosmo',
 'z',
```

```
'Name',
u'igm_sys',
'RA',
'PA',
'user',
'rho',
'ang_sep',
'DEC',
u'galaxy',
'CreationDate']
```

```
cdict['igm_sys'].keys()
```

```
['vlim',
'CreationDate',
'Name',
'flag_NHI',
u'components',
'Refs',
'NHI',
'RA',
'zem',
'kin',
'sig_NHI',
'zabs',
'DEC',
'user',
'abs_type']
```

```
# Write
import io, json
newdict = ltu.jsonify(cdict)
newdict['igm_sys']['kin'] = None
with io.open('/u/xavier/Desktop/J0943+0531_216_61.json', 'w', encoding='utf-8') as f:
    f.write(unicode(json.dumps(newdict, sort_keys=True, indent=4,
                              separators=(',', ' : '))))
```

## Simple plots

### Impact parameter

```
rho = cos_halos.rho
```

```
p = figure(plot_width=400, plot_height=400, title='COS-Halos Impact Parameters')
hist, edges = np.histogram(rho, range=(25., 155), bins=10)
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:], fill_color='blue')#,
↪alpha=0.5)
p.xaxis.axis_label = 'rho (kpc)'
# Show
show(p)
```

## Stellar Mass

```
Mstar = cos_halos.stellar_mass
```

```
p = figure(plot_width=400, plot_height=400, title='COS-Halos Stellar Masses')
hist, edges = np.histogram(Mstar, range=(9,12), bins=10)
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:], fill_color='blue')#,
↪alpha=0.5)
p.xaxis.axis_label = 'log10 M*'
# Show
show(p)
```

## Metallicity

```
# coming soon
```

## Load spectra

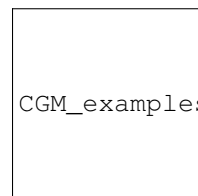
```
#
itup = ('J0928+6025', '110_35')
cgmabs = cos_halos[itup]
print(cgmabs)
```

```
<CGMAbsSys: J0928+6025_110_35 Galaxy RA/DEC=09:28:42.46+60:25:08.7, zgal=0.154,
↪rho=95.2382 kpc>
```

## Galaxy

```
# Not working right now
spec = cos_halos.load_gal_spec(('J0928+6025', '110_35'))
```

```
spec.plot()
```



## QSO

```
qso_spec = cos_halos.load_bg_cos_spec(itup, 1215.6700*u.AA)
```

```
qso_spec.plot()
```



CGM\_examples\_files/CGM\_examples\_35\_0.png

## Velocity plot

```
plt.figure(figsize=(18, 11))
cos_halos.stack_plot(itup, ymnx=(-0.1, 1.3))
```

```
Skipping 977.02 Angstrom. Assuming no coverage
```

```
/Users/xavier/local/Python/pyigm/pyigm/cgm/cos_halos.py:549: UserWarning: File /Users/
↪xavier/Dropbox/COS-Halos-Data//Targets/fitting/J0928+6025_110_35_z0.154_CIII977_
↪slice.fits not found
  warnings.warn("File {s} not found".format(slicedir+slicename))
```

```
<matplotlib.figure.Figure at 0x11a434550>
```



CGM\_examples\_files/CGM\_examples\_37\_3.png

## Ingesting Prochaska+11 CGM dataset [v1.0]

```
# imports
from astropy.table import Table
from astropy.coordinates import SkyCoord

import pyigm
from pyigm.cgm.cgmsurvey import CGMAbsSurvey
```

## Summary

```
ovi_file = pyigm.__path__[0]+'data/CGM/P11/lowovidat.fits'
```

```
ovidat = Table.read(ovi_file)
```

```
ovidat[0:3]
```

```
qso_radec = SkyCoord(ra=ovidat['QSO_RA'], dec=ovidat['QSO_DEC'], unit=(u.hourangle, u.
↪deg))
```

## Dwarfs

```
cgm_dwarf_file = pyigm.__path__[0]+'data/CGM/P11/dwarf_galabs_strct.fits'
```

```
cgm_dwarfs = Table.read(cgm_dwarf_file)
```

```
cgm_dwarfs[0:3]
```

## Funny columns – Renaming

```
cgm_dwarfs.rename_column('DRA', 'rho(kpc)')
cgm_dwarfs.rename_column('DDEC', 'L(L*)')
```

## MAG has HI

```
2 == Ref
3 == z
4 == Ly $\alpha$  EW
5 == sigEW Ly $\alpha$ 
8 == NHI
9 == sigNHI
```

```
cgm_dwarfs[1]['MAG']
```

```
array([ 1.89230003e+01,  1.80701466e+01,  5.00000000e+00,
        2.34109219e-02,  2.22000000e+02,  2.90000000e+01,
        5.60000000e+01,  4.00000000e+00,  1.38000002e+01,
        1.00000000e+00], dtype=float32)
```

## MAGERR has OVI

```
2 == Ref
3 == z
4 == EW 1031
5 == sigEW 1031
8 == NOVI
9 == sigNOVI
```

```
cgm_dwarfs[1]['MAGERR']
```

```
array([ 8.79999995e-02,  6.49999976e-02,  4.00000000e+00,
        2.33999994e-02,  4.30000000e+01,  1.50000000e+01,
        3.00000000e+01,  0.00000000e+00,  1.34799995e+01,
        5.00000007e-02], dtype=float32)
```

## Refs

```
refdict = {1: 'tripp08', 2: 'tc08a', 3: 'ds08', 4: 'dsr+06', 5: 'pss04', 6: 'cm09',
↳9: 'p+11'}
```

## Ingest

```
python ingest_lit.py
```

## Read

```
p11_tarfile = pyigm.__path__[0]+'./data/CGM/P11/P11_sys.tar'
```

```
p11 = CGMAbsSurvey.from_tarball(p11_tarfile, chk_lowz=False)
```

```
WARNING: UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS standard.
↳Suggested: 10**-1 nm. [astropy.units.format.utils]
/Users/xavier/local/Python/linetools/linetools/lists/linelist.py:374: RuntimeWarning:
↳divide by zero encountered in log10
  self._data['log(w*f)'] = np.log10(qm_strength)
/Users/xavier/anaconda/lib/python2.7/site-packages/numpy/ma/core.py:824:
↳RuntimeWarning: invalid value encountered in less_equal
  return umath.less_equal(x, self.critical_value)
```

Loading abundances from Asplund2009

Abundances are relative by number on a logarithmic scale with H=12

Skipping a likely folder: CGM\_JSON

```
/Users/xavier/local/Python/linetools/linetools/isgm/abssystem.py:288: UserWarning:
↳Input AbsComponent with Zion=(8, 6) does not match AbsSystem rules. Not appending
  warnings.warn('Input AbsComponent with Zion={} does not match AbsSystem rules. Not
↳appending'.format(abscomp.Zion))
/Users/xavier/local/Python/linetools/linetools/isgm/abssystem.py:294: UserWarning:
↳Failed velocity overlap
  warnings.warn('Failed velocity overlap')
/Users/xavier/local/Python/linetools/linetools/isgm/abssystem.py:288: UserWarning:
↳Input AbsComponent with Zion=(1, 1) does not match AbsSystem rules. Not appending
  warnings.warn('Input AbsComponent with Zion={} does not match AbsSystem rules. Not
↳appending'.format(abscomp.Zion))
```

```
p11
```

```
<CGM_Survey:  nsys=54, ref=>
<IGMSysSystem: IGMSysSystem 00:29:13.8 13:16:04, 0.0329451, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 00:29:13.8 13:16:04, 0.039311, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 00:57:20 -22:22:56, 0.0233644, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 00:57:20 -22:22:56, 0.045619, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 03:11:55.2 -76:51:51, 0.202643, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 03:11:55.2 -76:51:51, 0.0593531, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 04:07:48.4 -12:11:37, 0.0964516, NHI=0, Z/H=0>
<IGMSysSystem: IGMSysSystem 04:07:48.4 -12:11:37, 0.297609, NHI=0, Z/H=0>
```



```

<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.352, NHI=0, Z/H=0>
<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.153212, NHI=0, Z/H=0>
<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.203022, NHI=0, Z/H=0>
<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.36124, NHI=0, Z/H=0>
<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.167, NHI=0, Z/H=0>
<IGMSys: IGMSys 04:07:48.4 -12:11:37, 0.167043, NHI=0, Z/H=0>
<IGMSys: IGMSys 10:07:26.1 12:48:56, 0.0296659, NHI=0, Z/H=0>
<IGMSys: IGMSys 10:07:26.1 12:48:56, 0.0092207, NHI=0, Z/H=0>
<IGMSys: IGMSys 10:31:54.3 -14:16:51, 0.0508329, NHI=0, Z/H=0>
<IGMSys: IGMSys 11:19:08.7 21:19:18, 0.165951, NHI=0, Z/H=0>
<IGMSys: IGMSys 11:19:08.7 21:19:18, 0.0600208, NHI=0, Z/H=0>
<IGMSys: IGMSys 11:19:08.7 21:19:18, 0.0593876, NHI=0, Z/H=0>
<IGMSys: IGMSys 11:19:08.7 21:19:18, 0.13829, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:14:17.7 14:03:13, 0.06438, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:14:17.7 14:03:13, 0.0646084, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:14:17.7 14:03:13, 0.0519877, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:14:17.7 14:03:13, 0.0511341, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:19:20.9 06:38:38, 0.124102, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:19:20.9 06:38:38, 0.0131789, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:19:20.9 06:38:38, 0.00666757, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:19:20.9 06:38:38, 0.0080957, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:29:06.7 02:03:09, 0.00620912, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:33:25.8 09:31:23, 0.0118122, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:33:25.8 09:31:23, 0.125431, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:33:25.8 09:31:23, 0.0905207, NHI=0, Z/H=0>
<IGMSys: IGMSys 12:33:25.8 09:31:23, 0.206801, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:05:33 -10:33:19, 0.191709, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:05:33 -10:33:19, 0.0935802, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:05:33 -10:33:19, 0.0425621, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:05:33 -10:33:19, 0.145303, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:09:47 08:19:49, 0.127621, NHI=0, Z/H=0>
<IGMSys: IGMSys 13:09:47 08:19:49, 0.0337313, NHI=0, Z/H=0>
<IGMSys: IGMSys 14:29:06.4 01:17:06, 0.0299413, NHI=0, Z/H=0>
<IGMSys: IGMSys 14:29:06.4 01:17:06, 0.0281113, NHI=0, Z/H=0>
<IGMSys: IGMSys 15:55:43 11:11:24, 0.0150682, NHI=0, Z/H=0>
<IGMSys: IGMSys 15:55:43 11:11:24, 0.0395311, NHI=0, Z/H=0>
<IGMSys: IGMSys 15:55:43 11:11:24, 0.0420751, NHI=0, Z/H=0>
<IGMSys: IGMSys 15:55:43 11:11:24, 0.0416811, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.13262, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.0503651, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.155532, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.0807993, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.0788036, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.0809754, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:55:01.5 -09:22:25, 0.0516533, NHI=0, Z/H=0>
<IGMSys: IGMSys 21:58:51.8 -30:13:30, 0.0169383, NHI=0, Z/H=0>

```

p11.rho

[46.32007, 231.08717, 153.91919, 281.05024, 34.785816, 241.08236, 271.56657, 263.85542, 170.30404, 193.11905, 277.85574,

## 1.4.2 Overview

There are a series of classes designed for CGM analyses. We describe each in turn.

### 1.4.3 CGM

This class is intended to provide a physical representation of the CGM. A Galaxy object is required for instantiation. The key attributes are tabulated below. This includes several which may be controversial, e.g. the physical extent of the CGM, and a description of the baryons in ‘phases’. This is the least developed of the classes.

Attribute	Type	Description
galaxy	Galaxy object	Describes the galaxy hosting the CGM
rlim	Quantity	Physical extent of the CGM (e.g. 300 kpc)
phases	dict	Intended to organize the properties of various CGM phases, e.g. mass, metallicity
cgm_abs	list	List of CGMAbsSys classes

#### Instantiation

The only requirement is a Galaxy object.:

```
radec = (125*u.deg, 45.2*u.deg)
gal = Galaxy(radec, z=0.3)
cgm = CGM(gal)
```

### 1.4.4 CGMAbsSys

This class enables an absorption-line analysis of a CGM. This has been the primary approach to CGM analysis to date. This class requires both a Galaxy object and *IGMSys* object for instantiation.

Attribute	Type	Description
galaxy	Galaxy object	Describes the galaxy hosting the CGM
igm_sys	IGMSys	IGM system object to describe absorption on the line-of-sight
cosmo	astropy.cosmology	Cosmological model; defaults to WMAP9
rho	Quantity	Impact parameter from galaxy to sightline

#### Instantiation

One must invoke CGMAbsSys with both a Galaxy and IGMSys object.:

```
radec_qso = (125*u.deg, 45.203*u.deg)
igmsys = IGMSys('CGM', radec_qso, gal.z, [-500,500]*u.km/u.s)
#
radec = (125*u.deg, 45.2*u.deg)
gal = Galaxy(radec, z=0.3)
#
cgmabs = CGMAbsSys(gal, igmsys)
```

### 1.4.5 CGMAbsSurvey

This class organizes a survey of CGMAbsSys objects, e.g. COS-Halos.

Attribute	Type	Description
survey	str	Name of the survey, e.g. COS-Halos
ref	str	References for the survey
cgm_abs	list	list of CGMAbsSys objects
mask	bool array	Mask

## Instantiation

This class requires no input for instantiation. But, it is expected that one will fill the `cgm_abs` list with `CGMAbsSys` objects.

## Properties and Methods

### `nsys`

This property returns the number of `CGMAbsSys` objects in the survey (ignores mask):

```
nsys = cgmsurvey.nsys
```

### `getattr`

This is overloaded to return an array of properties from one of the internal sets of objects in the survey. The order of attribution is `CGMAbsSys` objects, Galaxy objects, and then `IGMSys` objects.

```
rho_array = cgmsurvey.rho # Grabs rho from CGMAbsSys objects
z_array = cgmsurvey.z # Grabs galaxy redshifts
coord = cgmsurvey.coord # Grabs galaxy coordinates
```

## Miscellaneous

There are a few methods related to CGM analysis available.

### `dN/dX`

Calculate  $dN/dX$  given a cosmology and a parameterization of the halos. Code of interest is `cgm.analysis.dndx_rvir`:

```
dNdX = cgm.analysis.dndx_rvir()
```

## COS-Halos

All of the measurements related to the COS-Halos survey [Werk et al. \(2011\)](#), [Tumlinson et al. \(2013\)](#), [Werk et al. \(2013\)](#) [Prochaska et al. \(2017\)](#) are included in the data directory. You can read in the dataset with:

```
from pyigm.cgm import cos_halos as pch
coshalos = pch.COSHalos()
```

See the Notebook in examples for further details.

### 1.4.6 CGMModel

Simple models designed to reproduce observations of the CGM may be generated using the `CGMModel` class. Current implementation is based on the Mathews & Prochaska (2017) halo profile.

See the `CGM_Models` Notebook for examples.

## 1.5 IGMSurvey Class

### 1.5.1 Notebooks

#### Examples for IGMSurvey (v1.2)

```
# imports
from astropy.coordinates import SkyCoord

from pyigm.abssys.igmsys import IGMSystem
from pyigm.abssys.igmsurvey import GenericIGMSurvey
```

#### Simple instantiation

```
gensurvey = GenericIGMSurvey()
gensurvey
```

```
[IGMSurvey: nsys=0, type=Generic, ref=]
```

```
coord = SkyCoord(ra=123.1143*u.deg, dec=-12.4321*u.deg)
gensys1 = IGMSystem('MgII', coord, 1.244, [-300,300.]*u.km/u.s, NHI=16.)
gensys1.name = 'Sys1'
#
coord2 = SkyCoord(ra=223.1143*u.deg, dec=42.4321*u.deg)
gensys2 = IGMSystem('MgII', coord2, 1.744, [-300,300.]*u.km/u.s, NHI=17.)
gensys2.name = 'Sys2'
```

```
gensurvey.add_abs_sys(gensys1)
gensurvey.add_abs_sys(gensys2)
```

```
gensurvey.abs_sys()
```

```
array([<IGMSystem: MgII 08:12:27.432 -12:25:55.56, 1.244, NHI=16, Z/H=0>,
       <IGMSystem: MgII 14:52:27.432 42:25:55.56, 1.744, NHI=17, Z/H=0>],
      dtype=object)
```

#### Parsing

```
gensurvey.NHI
```

```
array([ 16.,  17.])
```

```
gensurvey.zabs
```

```
array([ 1.244,  1.744])
```

## Examples for LLSSurvey (v1.5)

```
# imports
import seaborn as sns; sns.set_style("white")

import imp
import h5py

from bokeh.io import output_notebook, show, hplot, output_file
from bokeh.plotting import figure
from bokeh.models import Range1d

output_notebook()

from linetools import utils as ltu

from pyigm.surveys.llssurvey import LLSSurvey
from pyigm.surveys import llssurvey as llss
from pyigm.abssys.igmsys import AbsSubSystem
from pyigm.abssys.lls import LLSSystem
from pyigm.abssys import lls as pylls
from pyigm.surveys import lls_literature as llit
```

```
/Users/xavier/anaconda/lib/python2.7/site-packages/matplotlib/__init__.py:872:
↳UserWarning: axes.color_cycle is deprecated and replaced with axes.prop_cycle;
↳please use the latter.
  warnings.warn(self.msg_depr % (key, alt_key))
```

## LLS Tree (JXP) – Likely to Deprecate

```
#reload(llsu)
lls_survey = LLSSurvey.from_flist('Lists/lls_metals.lst', tree=os.getenv('LLSTREE'))
```

Read 165 files from Lists/lls\_metals.lst in the tree /u/xavier/LLS/

## HD-LLS (Prochaska+15, ApJS, 221, 22)

### Simple Init

```
# Includes metallicity PDFs from Fumagalli+16
reload(llss)
hdl1s = llss.LLSSurvey.load_HDLLS(load_sys=False)
hdl1s
```

```
HD-LLS: Loading summary file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/
↳HD-LLS_DR1.fits
HD-LLS: Loading ions file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/HD-
↳LLS_ions.json
HD-LLS: Loading metallicity file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-
↳LLS/HD-LLS_DR1_dustnhi.hdf5
```

```
<IGMSurvey: nsys=157, type=LLS, ref=HD-LLS>
```

## With Systems (takes ~2 min)

```
hdl1s = LLSSurvey.load_HDLLS()
hdl1s
```

```
HD-LLS: Loading summary file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/
↳HD-LLS_DR1.fits
HD-LLS: Loading ions file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/HD-
↳LLS_ions.json
HD-LLS: Loading metallicity file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-
↳LLS/HD-LLS_DR1_dustnhi.hdf5
Loading systems from /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/HD-LLS_
↳sys.tar.gz
Skipping a likely folder: SYS
```

```
WARNING: UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS standard.
↳Suggested: 10**-1 nm. [astropy.units.format.utils]
WARNING:astropy:UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS
↳standard. Suggested: 10**-1 nm.
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton03_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton00_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/verner96_tab1.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/verner94_tab6.fits
```

```
WARNING: UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS [astropy.units.core]
WARNING:astropy:UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/EUV_lines.ascii
read_sets: Using set file --
/Users/xavier/local/Python/linetools/linetools/lists/sets/l1list_v1.0.ascii
```

```
<IGMSurvey: nsys=157, type=LLS, ref=HD-LLS>
```

## Spectra

```
hdl1s = LLSSurvey.load_HDLLS(grab_spectra=True)
```

```
HD-LLS: Loading summary file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/
↳HD-LLS_DR1.fits
HD-LLS: Loading ions file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/HD-
↳LLS_ions.json
```

```

HD-LLS: Loading metallicity file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-
↳ LLS/HD-LLS_DR1_dustnhi.hdf5
Loading systems from /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/HD-LLS_
↳ sys.tar.gz
Skipping a likely folder: SYS
HD-LLS: Using files in /Users/xavier/local/Python/pyigm/pyigm/data/LLS/HD-LLS/Spectra/

```

```
hdlls.nsys
```

```
157
```

```
hdlls.NHI[0:10]
```

```
array([ 19.65,  20.05,  17.55,  19.1 ,  20.  ,  19.05,  19.1 ,  19.05,
        19.25,  20.2 ])
```

```
hdlls.name[0:5]
```

```
array([u'J000345-232346.5_z2.187', u'J003454.8+163920_z3.754',
       u'J004049.5-402514_z2.816', u'J010355.3-300946_z2.908',
       u'J010516.8-184642_z2.927'],
      dtype='<U27')
```

```
hdlls._abs_sys[77]._ionN
```

```
CII_clms = hdlls.ions((6,2))
CII_clms[70:80]
```

```
gdCII = CII_clms['flag_N']>0
np.sum(gdCII)
```

```
103
```

```

# NHI
p_NHI = figure(plot_width=400, plot_height=400, title='HD-LLS NHI')#,background_fill="
↳ #E8DDCB")
hist, edges = np.histogram(hdlls.NHI, range=(17.,20.4), density=True, bins=20)
p_NHI.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],fill_color='blue')
p_NHI.xaxis.axis_label = 'N_HI'
# z
p_z = figure(plot_width=400, plot_height=400, title='HD-LLS z')#,background_fill="
↳ #E8DDCB")
hist, edges = np.histogram(hdlls.zabs, range=(2.4, 4.5), density=True, bins=20)
p_z.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],fill_color='red')
p_z.xaxis.axis_label = 'z_LLS'
# Show
show(hplot(p_NHI,p_z))

```

```

print('Mean metallicity of {:s} is {:g}'.format(hdlls._abs_sys[0], hdlls._abs_sys[0].
↳ metallicity.meanZH))

```

```
Mean metallicity of <LLSSystem: 00:03:45 -23:23:46.5, zabs=2.1871, logNHI=19.65, tau_  
↳LL=283.16, [Z/H]=0 dex> is -1.47315
```

## SDSS LLS (Prochaska+10, ApJ, 718, 391)

### All

```
sdss_dr7_all = LLSSurvey.load_SDSS_DR7(sample='all')  
sdss_dr7_all
```

```
SDSS-DR7: Loading LLS file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/SDSS/lls_  
↳dr7_stat_LLS.fits.gz  
SDSS-DR7: Loading QSOs file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/SDSS/lls_  
↳dr7_qsos_sn2050.fits.gz
```

```
<IGMSurvey: nsys=1935, type=LLS, ref=SDSS-DR7, nsightlines=3759>
```

```
sdss_dr7_all.sightlines[0:5]
```

### Stat

```
sdss_dr7_stat = LLSSurvey.load_SDSS_DR7()
```

```
SDSS-DR7: Loading LLS file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/SDSS/lls_  
↳dr7_stat_LLS.fits.gz  
SDSS-DR7: Loading QSOs file /Users/xavier/local/Python/pyigm/pyigm/data/LLS/SDSS/lls_  
↳dr7_qsos_sn2050.fits.gz  
SDSS-DR7: Performing stats (~60s)  
SDSS-DR7: Loaded
```

```
len(sdss_dr7_stat.NHI)
```

```
254
```

### Sky Map

```
all_coord = sdss_dr7_all.coord  
stat_coord = sdss_dr7_stat.coord
```

```
%matplotlib inline
```

```
plt.figure(figsize=(12, 7))  
plt.clf()  
# Setup  
ax = plt.axes(projection='mollweide')  
ax = plt.axes()  
ax.set_xlabel('RA')
```



```

ax.set_ylabel('DEC')
ax.set_xticklabels(np.arange(30, 331, 30))
ax.grid(True)
# All
plt.scatter((all_coord.ra.value-180.)*np.pi/180., all_coord.dec.value*np.pi/180.,
            marker='o', s=3., lw=0.5, edgecolors='gray', facecolors='none',
            label='all')
# Stat
plt.scatter((stat_coord.ra.value-180.)*np.pi/180., stat_coord.dec.value*np.pi/180.,
            marker='o', s=3., lw=0.5, edgecolors='blue', facecolors='blue',
            label='stat')
# Legend
legend = plt.legend(loc='upper right', scatterpoints=1, borderpad=0.2,
                   handletextpad=0.1, fontsize='large')
plt.show()

```



LLSSurvey\_examples\_files/LLSSurvey\_examples\_33\_0.png

$z \sim 2.5$  HST (O'Meara et al. 2013, ApJ, 765, 137)

## ACS

```

acs = LLSSurvey.load_HST_ACS()
acs

```

HST-ACS: Loaded

```
<IGMSurvey: nsys=34, type=LLS, ref=HST-ACS, nsightlines=18>
```

## WFC3

```

wfc3 = LLSSurvey.load_HST_WFC3()
wfc3

```

HST-WFC3: Loaded

```
<IGMSurvey: nsys=91, type=LLS, ref=HST-WFC3, nsightlines=53>
```

## Combined

```

HST_LLS = wfc3 + acs
HST_LLS

```

```
<IGMSurvey: nsys=125, type=LLS, ref=HST-WFC3,HST-ACS, nsightlines=71>
```

---

$z \sim 3$  **MagE** (Fumagalli et al. 2013, ApJ, 775, 78)

## Load

```
z3mage = LLSurvey.load_mage_z3()  
z3mage
```

```
<IGMSurvey: nsys=60, type=LLS, ref=z3_MagE, nsightlines=105>
```

```
z3mage_NC = LLSurvey.load_mage_z3(sample='non-color')  
z3mage_NC
```

```
<IGMSurvey: nsys=32, type=LLS, ref=z3_MagE, nsightlines=61>
```

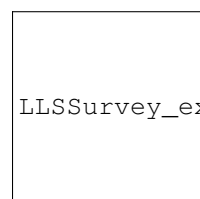
## g(z) plot

```
zeval, gz = z3mage.calculate_gz()
```

```
zeval[4000], gz[4000]
```

```
(2.8705998897560931, 67)
```

```
plt.clf()  
plt.plot(zeval, gz)  
plt.xlabel('z')  
plt.ylabel('g(z)')  
plt.show()
```



LLSSurvey\_examples\_files/LLSSurvey\_examples\_50\_0.png

## Literature

```
from pyigm.surveys.lls_literature import log_sum
```

```
reload(llit)  
zonak04 = llit.zonak2004()  
print(zonak04)  
zonak04._ionN
```

```
<LLSSystem: 16:34:28.9897 +70:31:32.422, zabs=1.0414, logNHI=17.23, tau_LL=1.07654,
↪[Z/H]=0 dex>
```

```
reload(llit)
jenkins05 = llit.jenkins2005()
jenkins05._ionN
```

```
No error for N I
No error for O I
No error for O I
WARNING: Using 1393.7550 Angstrom for your input 1393.7600 Angstrom
WARNING: Using 1402.7700 Angstrom for your input 1402.7730 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5010 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.1910 Angstrom
```

```
reload(llit)
tripp05 = llit.tripp2005()
tripp05._ionN
```

```
reload(llit)
peroux06a = llit.peroux06a()
print(peroux06a)
peroux06a._ionN
```

```
<LLSSystem: 01:34:05.75 +00:51:09.4, zabs=0.842, logNHI=19.93, tau_LL=539.55, [Z/H]=0
↪dex>
```

```
reload(llit)
peroux06b = llit.peroux06b()
print(peroux06b)
peroux06b._ionN
```

```
<LLSSystem: 13:23:23.78 -00:21:55.2, zabs=0.716, logNHI=20.21, tau_LL=1028.09, [Z/
↪H]=0 dex>
```

```
reload(llit)
meiring06 = llit.meiring06()
print(meiring06)
meiring06._ionN
```

```
<LLSSystem: 11:07:36.6552 +00:03:28.62, zabs=0.9542, logNHI=20.26, tau_LL=1153.54, [Z/
↪H]=0 dex>
```

```
reload(llit)
meiring07 = llit.meiring07()
for imeiring07 in meiring07:
    print(imeiring07)
    print(imeiring07._ionN)
```

```
<LLSSystem: 03:54:05.9 -27:24:25.7, zabs=1.4051, logNHI=20.18, tau_LL=959.471, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
---
2      0.04  24      1 13.25
```

```

 2      0.03  25      1 12.82
 1      0.01  12      1 12.7
 2      0.01  26      1 15.03
 2      0.0   12      2 14.39
<LLSSystem: 08:26:01.5 -22:30:26.2, zabs=0.911, logNHI=19.04, tau_LL=69.5076, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.04  20      1 11.42
 1      0.02  12      1 12.06
 2      0.12  26      1 13.43
 2      0.0   12      2 13.71
<LLSSystem: 10:09:30.4 -00:26:19.1, zabs=0.8426, logNHI=20.2, tau_LL=1004.69, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.03  26      1 14.37
 2      0.04  25      1 12.28
 1      0.05  12      1 11.8
 3      0.03  13      1 12.74
 2      0.0   12      2 13.87
<LLSSystem: 10:09:30.4 -00:26:19.1, zabs=0.8866, logNHI=19.48, tau_LL=191.44, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.06  26      1 14.33
 2      0.04  20      1 12.1
 1      0.02  12      1 12.43
 3      0.02  13      1 13.0
 2      0.0   12      2 14.32
<LLSSystem: 10:10:33.4 -00:47:24.5, zabs=1.327, logNHI=19.81, tau_LL=409.291, [Z/H]=0.
↪dex>
ion  Z  sig_logN  flag_N  logN
-----
 2  13      0.0      2 13.72
 3  13      0.02     1 13.29
 2  14      0.15     1 14.86
 1  12      0.02     1 12.49
 2  12      0.0      2 14.26
 2  26      0.02     1 14.5
<LLSSystem: 12:24:14.3 +00:37:09, zabs=1.2665, logNHI=20, tau_LL=633.916, [Z/H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.04  26      1 14.36
 3      0.14  13      1 12.74
 1      0.08  12      1 12.21
 2      0.0   13      2 13.56
 2      0.0   12      2 14.25
<LLSSystem: 23:31:21.8 +00:38:07.4, zabs=1.1414, logNHI=20, tau_LL=633.916, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.05  26      1 14.44
 3      0.14  13      1 12.86
 1      0.03  12      1 12.47
 2      0.0   13      2 13.33
 2      0.0   12      2 14.39

```

```

reload(llit)
meiring08 = llit.meiring08()
for imeiring08 in meiring08:
    print(imeiring08)
    print(imeiring08._ionN)

```

```

<LLSSystem: 10:37:44.4 +00:28:09.2, zabs=1.4244, logNHI=20.04, tau_LL=695.076, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.11  25      1 12.57
  2      0.0   13      2 13.6
  3      0.03  13      1 13.15
  2      0.05  14      1 15.05
  2      0.0   12      2 14.44
  2      0.09  26      1 14.96
<LLSSystem: 10:54:40.98 -00:20:48.4, zabs=0.8301, logNHI=18.95, tau_LL=56.4979, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.01  26      1 14.35
  2      0.1   25      1 12.31
  1      0.02  12      1 12.54
  3      0.04  13      1 13.65
  2      0.0   12      2 14.3
<LLSSystem: 10:54:40.98 -00:20:48.4, zabs=0.9514, logNHI=19.28, tau_LL=120.79, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  1      0.07  12      1 12.04
  2      0.08  26      1 13.49
  2      0.0   12      2 13.59
<LLSSystem: 12:15:49.81 -00:34:32.1, zabs=1.5543, logNHI=19.56, tau_LL=230.161, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.11  26      1 14.35
  2      0.0   12      2 14.25
<LLSSystem: 12:20:37 -00:40:32.4, zabs=0.9746, logNHI=20.2, tau_LL=1004.69, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.01  26      1 14.35
  1      0.05  12      1 12.1
  3      0.12  13      1 12.68
  2      0.0   12      2 14.37
<LLSSystem: 12:28:36.8 +10:18:41.7, zabs=0.9376, logNHI=19.41, tau_LL=162.942, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  1      0.05  12      1 12.32
  2      0.02  26      1 14.55
  2      0.0   12      2 14.36
<LLSSystem: 13:30:07.7 -20:56:16.4, zabs=0.8526, logNHI=19.4, tau_LL=159.233, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.07  26      1 13.76

```

```

1      0.09 12      1 12.37
3      0.15 13      1 12.52
2      0.0  12      2 14.12
<LLSSystem: 14:36:45.03 -00:51:50.6, zabs=0.7377, logNHI=20.08, tau_LL=762.135, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
2      0.01 20      1 12.79
2      0.03 25      1 13.05
1      0.0  12      2 12.9
2      0.02 26      1 14.92
2      0.0  12      2 14.26
<LLSSystem: 14:55:08.14 -00:45:07.5, zabs=1.0929, logNHI=20.08, tau_LL=762.135, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
2      0.17 24      1 12.75
2      0.1  25      1 12.2
2      0.0  13      2 13.71
3      0.01 13      1 13.67
2      0.12 14      1 14.84
1      0.02 12      1 12.51
2      0.0  12      2 14.45
2      0.02 26      1 14.61

```

```

reload(llit)
nestor08 = llit.nestor08()
print(nestor08)
nestor08._ionN

```

```

<LLSSystem: 21:51:45.8304 +21:30:13.5, zabs=1.0023, logNHI=19.3, tau_LL=126.483, [Z/
↪H]=0 dex>

```

```

reload(llit)
meiring09 = llit.meiring09()
for imeiring09 in meiring09:
    print(imeiring09)
    print(imeiring09._ionN)

```

```

<LLSSystem: 00:05:20.21 +05:24:10.8, zabs=0.8514, logNHI=19.08, tau_LL=76.2135, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
2      0.02 26      1 13.75
1      0.04 12      1 12.24
3      0.06 13      1 13.13
2      0.0  12      2 14.13
<LLSSystem: 00:12:10.89 -01:22:07.5, zabs=1.3862, logNHI=20.26, tau_LL=1153.54, [Z/
↪H]=0 dex>
ion  Z  sig_logN  flag_N  logN
----
2 13      0.0      2 13.07
3 13      0.04     1 12.83
2 14      0.04     1 14.45
1 12      0.05     1 11.75
2 12      0.0      2 13.81

```

```

 2  26      0.01      1 14.25
<LLSSystem: 00:21:27.88 +01:04:20.1, zabs=1.3259, logNHI=20.04, tau_LL=695.076, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.04  26      1 14.68
 2      0.0   14      2 14.86
 1      0.06  12      1 12.26
 2      0.0   13      2 13.68
 2      0.0   12      2 14.5
<LLSSystem: 04:27:07.32 -13:02:53.6, zabs=1.408, logNHI=19.04, tau_LL=69.5076, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.04  14      1 13.56
 2      0.02  26      1 13.33
 2      0.03  13      1 12.21
 2      0.0   12      2 13.25
<LLSSystem: 16:31:45.24 +11:56:02.9, zabs=0.9004, logNHI=19.7, tau_LL=317.711, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.1   20      1 12.22
 1      0.05  12      1 12.44
 2      0.03  26      1 14.17
 2      0.0   12      2 14.04
<LLSSystem: 20:51:45.87 +19:50:06.3, zabs=1.1157, logNHI=20, tau_LL=633.916, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.1   24      1 13.08
 2      0.04  25      1 13.24
 2      0.0   13      2 13.77
 3      0.02  13      1 13.52
 2      0.12  14      1 15.31
 1      0.02  12      1 12.67
 2      0.04  20      1 12.59
 2      0.0   12      2 14.44
 2      0.02  26      1 15.0
<LLSSystem: 23:52:53.51 -00:28:51.3, zabs=0.873, logNHI=19.18, tau_LL=95.9471, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 1      0.14  12      1 11.84
 2      0.02  26      1 13.47
 2      0.0   12      2 14.05
<LLSSystem: 23:52:53.51 -00:28:51.3, zabs=1.0318, logNHI=19.81, tau_LL=409.291, [Z/
↪H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
 2      0.26  24      1 12.94
 3      0.05  13      1 13.41
 2      0.06  14      1 15.49
 1      0.03  12      1 12.6
 2      0.0   12      2 14.58
 2      0.01  26      1 14.88
<LLSSystem: 23:52:53.51 -00:28:51.3, zabs=1.2467, logNHI=19.6, tau_LL=252.367, [Z/
↪H]=0 dex>

```

```

ion sig_logN  Z  flag_N  logN
-----
  2      0.09  26      1 14.19
  3      0.03  13      1 13.43
  1      0.07  12      1 12.32
  2      0.02  13      1 13.42
  2      0.0   12      2 14.39

```

```

reload(llit)
dessauges09 = llit.dessauges09()
for ills in dessauges09:
    print(ills)
    print(ills._ionN)

```

```

<LLSSystem: 00:12:10.9 -01:22:08, zabs=1.3861, logNHI=20.26, tau_LL=1153.54, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.04  26      1 14.32
  2      0.0   30      3 10.87
<LLSSystem: 00:21:33.3 +00:43:00, zabs=0.5203, logNHI=19.54, tau_LL=219.802, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.05  26      1 13.17
<LLSSystem: 00:21:33.3 +00:43:00, zabs=0.9424, logNHI=19.38, tau_LL=152.066, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.14  26      1 14.62
  2      0.0   30      3 11.6
<LLSSystem: 01:57:33.8 -00:48:24, zabs=1.4157, logNHI=19.9, tau_LL=503.538, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.08  26      1 14.57
  2      0.11  30      1 12.1
<LLSSystem: 02:18:57.3 +08:17:28, zabs=1.7687, logNHI=20.2, tau_LL=1004.69, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.1   26      1 14.48
  2      0.16  30      1 12.02
<LLSSystem: 03:54:05.6 -27:24:20, zabs=1.4054, logNHI=20.18, tau_LL=959.471, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.15  26      1 15.1
  2      0.15  30      1 12.86
<LLSSystem: 04:27:07.3 -13:02:54, zabs=1.408, logNHI=19.04, tau_LL=69.5076, [Z/H]=0.0
↳dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.04  26      1 13.45
  2      0.0   30      3 10.83
<LLSSystem: 10:09:30.5 -00:26:18, zabs=0.8428, logNHI=20.2, tau_LL=1004.69, [Z/H]=0.0
↳dex>

```



```

ion sig_logN  Z  flag_N  logN
-----
  2      0.06  26      1  14.48
  2      0.0   30      3  11.42
<LLSSystem: 10:09:30.5 -00:26:18, zabs=0.8865, logNHI=19.48, tau_LL=191.44, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.09  26      1  14.37
  2      0.15  30      1  12.35
<LLSSystem: 10:28:37.1 -01:00:28, zabs=0.6321, logNHI=19.95, tau_LL=564.979, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.08  26      1  15.06
  2      0.0   30      3  12.38
<LLSSystem: 10:28:37.1 -01:00:28, zabs=0.7089, logNHI=20.04, tau_LL=695.076, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.07  26      1  15.1
  2      0.0   30      3  12.49
<LLSSystem: 10:39:21.9 -27:19:16, zabs=2.1395, logNHI=19.6, tau_LL=252.367, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.1   26      1  14.7
  2      0.1   30      1  12.18
<LLSSystem: 10:54:41 -00:20:48, zabs=0.9513, logNHI=19.28, tau_LL=120.79, [Z/H]=0 dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.02  26      1  13.71
  2      0.0   30      3  11.25
<LLSSystem: 13:30:07.8 -20:56:17, zabs=0.8514, logNHI=19.4, tau_LL=159.233, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.04  26      1  13.9
  2      0.0   30      3  11.57
<LLSSystem: 15:25:10.6 +00:26:33, zabs=0.5674, logNHI=19.78, tau_LL=381.972, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.1   26      1  14.19
<LLSSystem: 20:51:12.7 +19:50:07, zabs=1.1161, logNHI=20, tau_LL=633.916, [Z/H]=0 dex>
ion  Z  sig_logN  flag_N  logN
-----
  2  26      0.0      2  15.22
  2  30      0.09     1  12.96
<LLSSystem: 23:52:53.5 -00:28:52, zabs=0.873, logNHI=19.18, tau_LL=95.9471, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-----
  2      0.11  26      1  13.5
  2      0.0   30      3  12.03
<LLSSystem: 23:52:53.5 -00:28:52, zabs=1.0318, logNHI=19.81, tau_LL=409.291, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN

```

```

-- -- -- -- --
  2      0.14  26      1 14.96
  2      0.0  30      3 11.94
<LLSSystem: 23:52:53.5 -00:28:52, zabs=1.2468, logNHI=19.6, tau_LL=252.367, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-- -- -- -- --
  2      0.24  26      1 14.28
  2      0.0  30      3 11.1

```

```

reload(llit)
tumlinson11 = llit.tumlinson11()
print(tumlinson11)
tumlinson11._ionN

```

```

<LLSSystem: 10:09:02.06 +07:13:43.8, zabs=0.3558, logNHI=18.4, tau_LL=15.9233, [Z/
↪H]=0 dex>

```

```

reload(llit)
kacprzak12 = llit.kacprzak12()
print(kacprzak12)
kacprzak12._ionN

```

```

<LLSSystem: 13:19:56.2209 +27:28:08.271, zabs=1.0023, logNHI=18.3, tau_LL=12.6483, [Z/
↪H]=0 dex>

```

```

reload(llit)
battisti12 = llit.battisti12()
for ills in battisti12:
    print(ills)
    print(ills._ionN)

```

```

<LLSSystem: 09:25:54.7 +40:04:14.1, zabs=0.2477, logNHI=19.5, tau_LL=200.462, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-- -- -- -- --
  2      0.0  15      3 13.5
  2      0.06 14      1 14.62
  2      0.0   7      2 14.95
  3      0.0  14      2 13.74
  2      0.0  16      3 14.72
  1      0.09   8      1 15.95
  3      0.0   6      2 14.17
  1      0.04   7      1 14.75
  3      0.0  26      3 14.23
  4      0.11  14      1 13.54
  2      0.0  28      3 14.41
  2      0.0   6      2 15.18
  2      0.09  26      1 14.22
<LLSSystem: 09:28:37.98 +60:25:21, zabs=0.1538, logNHI=19.3, tau_LL=126.483, [Z/H]=0.
↪dex>
ion sig_logN  Z  flag_N  logN
-- -- -- -- --
  4      0.0   6      2 14.1
  2      0.0  15      3 14.17
  2      0.0  14      2 14.39

```

```

2      0.0   7      2 14.85
3      0.0  14      2 13.77
2      0.0  16      3 14.65
1      0.0   8      2 15.08
1     0.06  12      1 12.7
2      0.0  22      3 11.94
2     0.06  20      1 12.81
1     0.09   7      1 14.1
3     0.11  26      1 14.59
2      0.0  12      2 13.99
4     0.12  14      1 13.86
2      0.0  28      3 13.67
2      0.0   6      2 14.91
2     0.08  26      1 14.9
<LLSSystem: 10:01:02.55 +59:44:14.3, zabs=0.3035, logNHI=19.3, tau_LL=126.483, [Z/
↪H]=0 dex>
ion sig_logN   Z   flag_N   logN
-----
2      0.07  15      1 12.81
2      0.03  14      1 14.73
2      0.0   7      2 14.95
3      0.0  14      2 14.0
2      0.0  16      3 14.53
3     0.09  26      1 14.14
1     0.02   8      1 15.64
3      0.0   6      2 14.52
1     0.17   7      1 13.65
2     0.04  26      1 14.3
2      0.0  28      3 14.23
2      0.0   6      2 15.06
6     0.05   8      1 14.34
<LLSSystem: 14:35:11.53 +36:04:37.2, zabs=0.2026, logNHI=19.8, tau_LL=399.974, [Z/
↪H]=0 dex>
ion sig_logN   Z   flag_N   logN
-----
2      0.0  15      3 12.79
2      0.0  14      2 14.11
3      0.0  14      2 13.41
2     0.12  16      1 14.6
1      0.0   8      2 15.58
3      0.0   6      2 14.33
1     0.14   7      1 14.6
3      0.0  26      3 13.69
4     0.13  14      1 13.2
2      0.0  28      3 13.99
2      0.0   6      2 14.52
2     0.08  26      1 14.2
<LLSSystem: 15:53:04.92 +35:48:28.6, zabs=0.083, logNHI=19.5, tau_LL=200.462, [Z/H]=0.
↪dex>
ion sig_logN   Z   flag_N   logN
-----
4     0.08   6      1 13.99
2     0.05  14      1 14.22
2     0.07   7      1 14.16
3      0.0  14      2 13.3
2      0.0  16      3 14.24
1      0.0   8      2 14.56
1      0.0   7      3 13.74

```

```
4      0.08  14      1 13.59
2      0.0   28      3 13.92
2      0.0   6       2 14.35
2      0.07  26      1 14.01
```

## Load them All

```
lls_lit = llit.load_lls_lit()
lls_lit
```

```
No error for N I
No error for O I
No error for O I
WARNING: Using 1393.7550 Angstrom for your input 1393.7600 Angstrom
WARNING: Using 1402.7700 Angstrom for your input 1402.7730 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5010 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.1910 Angstrom
```

```
<IGMSurvey: nsys=58, type=LLS, ref=Zon04,Jen05,Tri05,Prx06a,Prx06b,Mei06,Mei07,Mei08,
↪Nes08,Mei09,DZ09,Tum11,Kcz12,Bat12>
```

```
lls_lit.ref
```

```
u'Zon04,Jen05,Tri05,Prx06a,Prx06b,Mei06,Mei07,Mei08,Nes08,Mei09,DZ09,Tum11,Kcz12,Bat12
↪'
```

## Plot

```
# NHI
p_NHI = figure(plot_width=400, plot_height=400, title='Lit-LLS NHI')#,background_fill=
↪"#E8DDCB")
hist, edges = np.histogram(lls_lit.NHI, range=(17.,20.4), density=True, bins=20)
p_NHI.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],fill_color='blue')
p_NHI.xaxis.axis_label = 'N_HI'
# z
p_z = figure(plot_width=400, plot_height=400, title='Lit-LLS z')#,background_fill="
↪#E8DDCB")
hist, edges = np.histogram(lls_lit.zabs, range=(0., 4.5), bins=20)
p_z.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],fill_color='red')
p_z.xaxis.axis_label = 'z_LLS'
# Show
show(hplot(p_NHI,p_z))
```

### 1.5.2 IGMSurvey

This Class is designed to organize and analyze a survey of absorption systems (defined as AbsSystem objects).

By definition, an IGMSurvey is a unique collection of AbsSystem objects. It is specified by the number of systems and the references.

### 1.5.3 Instantiation

The AbsSystem Class may be instantiated in a few ways. The default sets the properties listed above:

```
gensurvey = GenericIGMSurvey()
```

More commonly, one will instantiate with one or more IGMSystem objects:

```
coord = SkyCoord(ra=123.1143*u.deg, dec=-12.4321*u.deg)
gensys = IGMSystem('MgII', coord, 1.244, [-300,300.]*u.km/u.s, NHI=16.)
gensys.name = 'Sys1'
#
coord2 = SkyCoord(ra=223.1143*u.deg, dec=42.4321*u.deg)
gensys2 = IGMSystem('MgII', coord2, 1.744, [-300,300.]*u.km/u.s, NHI=17.)
gensys2.name = 'Sys2'
#
gensurvey.add_abs_sys(gensys1)
gensurvey.add_abs_sys(gensys2)
```

### 1.5.4 Attributes/Properties

Variable	Type	Description
nsys	int	Number of systems in the survey
ref	str	References for the survey

### 1.5.5 Sub Classes

#### LLS

Subclass for an LLS survey. There are many published surveys that can be read in. Several require access to the Internet which will then generate a file on your drive for future use. There is also a method to handle the .dat and .lst files used by JXP. See [Examples for LLSSurvey \(v1.5\)](#) for more.

Here is a Table describing the various samples that may be accessed.

Survey	Call	Reference(s)	Description
SDSS	LLSSurvey.load_SDSS_DR7()	<a href="#">Prochaska+10</a>	tau>2 survey
z2_HST	LLSSurvey.load_HST_ACS() LLSSurvey.load_HST_WFC3()	<a href="#">Omeara+11</a> <a href="#">Omeara+11</a>	tau>2 with HST/ACS tau>1 with HST/WFC3
z3_MagE	LLSSurvey.load_mage_z3()	<a href="#">Fumagalli+13</a>	tau>2 with Magellan/MagE
HD-LLS	LLSSurvey.load_HDLLS()	<a href="#">Prochaska+15</a> <a href="#">Fumagalli+16</a>	Ionic column densities and metallicity PDF
Literature	lls_literature.load_lls_lit()	See <a href="#">Fumagalli+16</a>	Ionic column densities

Below are additional options for a few of these.

#### HD-LLS DR1

The standard call loads the ionic column densities and metallicity PDFs. One call access the spectra with:

```
hdl1s = LLSSurvey.load_HDL1S(grab_spectra=True)
```

This will grab 154Mb of data from the internet, and place them within pyigm/data/LLS/HD-LLS.

## DLAs

Subclass for dlasurvey. Presently handles the .dat and .lst files used by JXP. See DLASurvey\_examples for more.

Here is a Table describing the various DLA surveys that may be loaded and manipulated.

Survey	Call	Reference(s)	Description
HST16	DLASurvey.load_HST16()	<a href="#">Neeleman+16</a>	Blind survey of HST UV spectra
H100	DLASurvey.load_H100()	<a href="#">Neeleman+13</a>	100 unbiased HIRES spectra
SDSS_DR5	DLASurvey.load_SDSS_DR5()	<a href="#">PW09</a>	DR5
P03	DLASurvey.load_P03()	<a href="#">P03</a>	Compilation by Peroux et al.
G09	DLASurvey.load_G09()	<a href="#">G09</a>	Compilation by Guimaraes et al.
GGG	DLASurvey.load_GGG()	<a href="#">GGG</a>	Giant Gemini GMOS survey
XQ100	DLASurvey.load_XQ100()	<a href="#">XQ100</a>	Survey of XQ-100 spectra

See [DLASurvey\\_examples](#) for usage example.

## 1.5.6 Plots

## 1.5.7 Methods

### $g(z)$

Provide the sightlines Table is filled and has keys Z\_START and Z\_END, this method will generate a selection function  $g(z)$  curve:

```
# LLS
z3mage = LLSSurvey.load_mage_z3()
zeval, gz = z3mage.calculate_gz()
# DLA
sdss = DLASurvey.load_SDSS_DR5()
zeval, gz = sdss.calculate_gz()
```

### $f(N,X)$

Calculate the NHI frequency distribution in bins of NHI and z. e.g.,

```
fN, fN_lo, fN_hi = sdss_stat.calculate_fn([20.3, 20.5, 21., 21.5, 22.], [2, 2.5], ↪log=True)
```

Setting log=True returns log10 values for f(N) and its error.

### $I(X)$

Calculate the incidence per unit dX in binned redshift intervals. Default is over all NHI values. Here is an example:

```
lX, lX_lo, lX_hi = sdss_stat.calculate_lox([2., 2.5, 3])
```

This calculates lX and its error in the intervals z=[2,2.5] and z=[2.5,3].

## rhoHI

Similar to the last two methods but now for the HI mass density. Here is an example:

```
zbins = [2., 2.5, 3.]
NHImnx = (20.3, 23.)
rho, rho_lo, rho_hi = sdss_stat.calculate_rhoHI(zbins, NHImnx)
```

rho will have units of Solar mass per Mpc<sup>3</sup>.

## 1.5.8 Output

Core classes

## 1.6 CUBA Class

### 1.6.1 Notebooks

Examples for the CUBA Class (v1.0)

Designed to work with the CUBA output from Haardt & Madau

```
# import
# imports
try:
    import seaborn as sns; sns.set_style("white")
except:
    pass

from pyigm.euvb import cuba as pycuba # Location may move
```

```
/Users/xavier/anaconda/lib/python2.7/site-packages/matplotlib/__init__.py:872:
↳ UserWarning: axes.color_cycle is deprecated and replaced with axes.prop_cycle;
↳ please use the latter.
    warnings.warn(self.msg_depr % (key, alt_key))
```

### Initialize

```
reload(pycuba)
cuba = pycuba.CUBA()
cuba
```

```
read_cuba: Using CUBA file -- /Users/xavier/local/Python/pyigm/pyigm/data/euvb/cuba_
↳ uvbapr2011_q1g01.hiz.out
```

```
[CUBA: cuba_file=/Users/xavier/local/Python/pyigm/pyigm/data/euvb/cuba_uvapr2011_
↳ q1g01.hiz.out]
```

## Data in Attributes

```
# Redshifts
cuba.z[0:10]
```

```
array([ 0.      ,  0.04912,  0.1006 ,  0.1547 ,  0.2114 ,  0.2709 ,
        0.3333 ,  0.3988 ,  0.4675 ,  0.5396 ])
```

```
# Wavelengths
cuba.wave[0:10]
```

[1155000, 1101000, 1050000, 1000000, 953600, 908900, 866400, 825800, 787100, 750300] Å

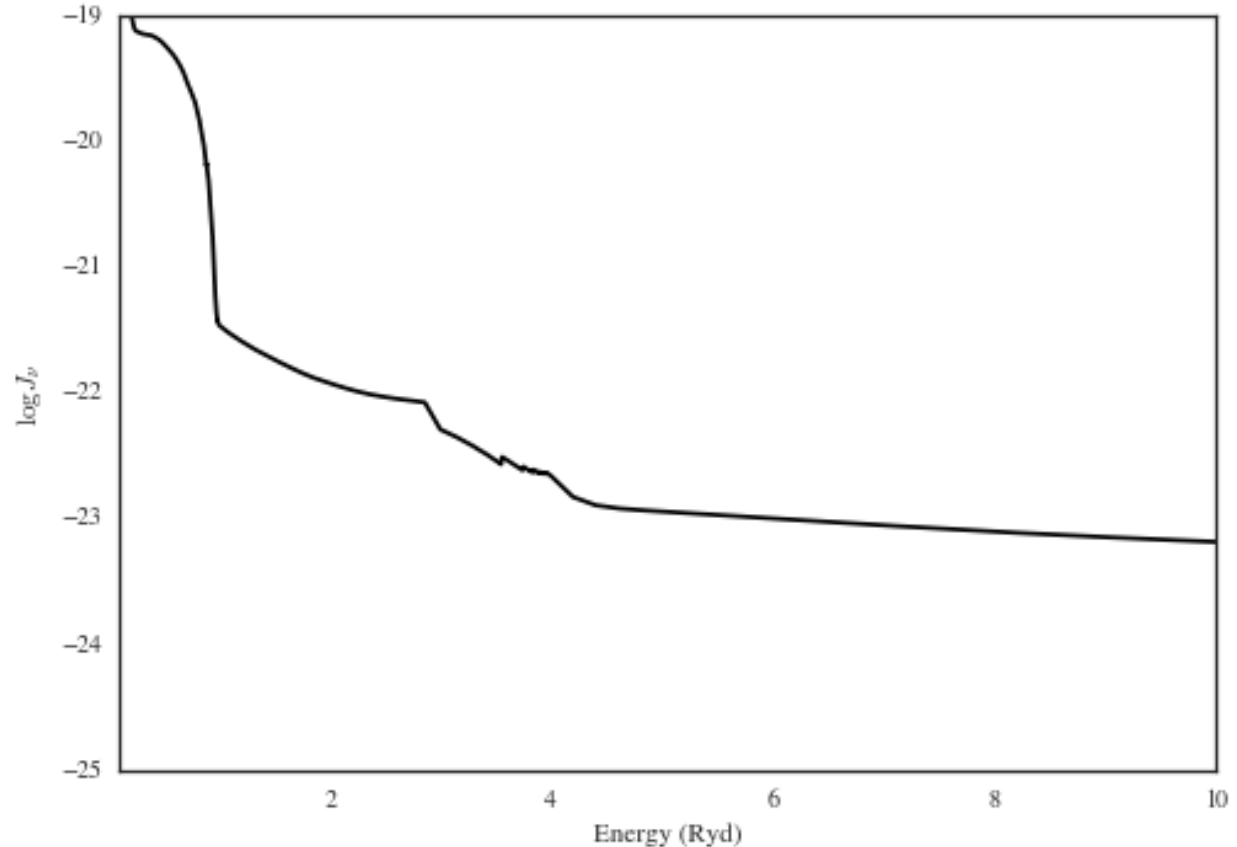
```
# Jnu
cuba.Jnu[0:10,0]
```

$[1.054 \times 10^{-20}, 1.035 \times 10^{-20}, 1.017 \times 10^{-20}, 1 \times 10^{-20}, 9.83 \times 10^{-21}, 9.666 \times 10^{-21}, 9.508 \times 10^{-21}, 9.355 \times 10^{-21}, 9.207 \times 10^{-21}]$

## Quick Plot

```
%matplotlib inline
```

```
cuba.plot(2.1,xlim=(0.1,10))
```





## Interpolate on redshift

```
jnu = cuba.zinterp_jnu(2.1)
jnu[0:5]
```

$[1.3653469 \times 10^{-19}, 1.3393469 \times 10^{-19}, 1.3139388 \times 10^{-19}, 1.2889388 \times 10^{-19}, 1.2649388 \times 10^{-19}] \frac{\text{erg}}{\text{s cm}^2}$

## Phi (Photon flux)

```
phi = cuba.phi(2.1) #,min_energy=50*u.eV
print('Ionizing flux = {:g}'.format(phi))
```

```
cuba.phi: Assuming minimum energy = 13.6057 eV
Ionizing flux = 95459.4 1 / (cm2 s)
```

## Matches JFH IDL code to 1%

```
print(phi)
```

```
95459.3988508 1 / (cm2 s)
```

## 1.6.2 Overview

This Class is designed to interface with the somewhat cumbersome output of CUBA. It may then be used to perform EUVB calculations.

## 1.6.3 Instantiation

The CUBA Class is instantiated with a CUBA file. If one is not provided, a default file from within the package is loaded.:

```
from pyigm.euvb import cuba as pycuba
cuba = pycuba.CUBA()
```

## 1.6.4 Attributes/Properties

Variable	Type	Description
z	ndarray	Redshifts of the evaluation
wave	Quantity array	Wavelengths of the evaluation
Jnu	Quantity array	Intensity of the EUVB

## 1.6.5 Plots

One can generate a quick plot at a given redshift with:

```
cuba.plot(z)
```

## 1.6.6 Methods

### zinterp

The intensity may be evaluated at any redshift with this method:

```
jnu = cuba.zinterp_jnu(z)
```

### phi

The ionizing photon flux may be calculated with this method, again at a given redshift:

```
phi = cuba.phi(z)
```

## 1.6.7 Output

A future method will be provided to generate a file that may be used with [Cloudy](#).

## 1.7 fN Classes

### 1.7.1 Notebooks

### 1.7.2 Overview

There are two classes designed for  $f(N)$  calculations. While the authors have reservations on the validity of  $f(N)$  as a description of the IGM, it still offers good value for a number of calculations.

### 1.7.3 FNModel

The FNModel Class may take several forms as described below.

Type	Description	Reference
Hspline	Hermite spline which is constrained to monotonically decrease	<a href="#">Prochaska+14</a>
Gamma	Power-law + exponential	<a href="#">Inoue+14</a>

### Instantiation

Here are some examples of instantiation:

```
from pyigm.fN.fnmodel import FNModel
fN_P14 = FNModel('Hspline', zmnx=(2., 5.))
fN_I14 = FNModel('Gamma')
```

And the default model is currently the P14 formulation:

```
fN_default = FNModel.default_model()
```

## Attributes/Properties

Variable	Type	Description
zmnx	tuple	min/max redshift for using this model to evaluate f(N)

## Methods

### l(X)

Calculate  $\ell(X)$ , the incidence of absorption per absorption path length  $dX$  in a given interval of column density:

$$\ell(X) = \int_{N_{\min}}^{N_{\max}} f(N, X) dN$$

Easily evaluated with:

```
lX = fN_default.calculate_lox(z, Nmin, Nmax)
```

### tau\_eff^LL

Calculate the effective optical depth to Lyman limit photons

$$\tau_{\text{eff}}^{\text{LL}}(z_{912}, z_q) = \int_{z_{912}}^{z_q} \int_0^{\infty} f(N_{\text{HI}}, z) [1 - \exp[-N_{\text{HI}} \sigma_{\text{ph}}(z')]] dN_{\text{HI}} dz'$$

Easily evaluated over across a redshift interval:

```
from pyigm.fN import tau_eff as pyteff
zval,teff_LL = pyteff.lyman_limit(fN_default, 0.5, 2.45)
```

## MFP

Evaluate the mean free path to ionizing photons at a given redshift:

```
mfp = fN_default.mfp(z)
```

## rho HI

Evaluate the mass density in HI atoms at a specific redshift, integrating over a  $N_{\text{HI}}$  interval:

$$\rho_{\text{HI}} = \frac{m_p H_0}{c} \int_{N_{\min}}^{N_{\max}} f(N_{\text{HI}}, X) N_{\text{HI}} dN_{\text{HI}}$$

As such:

```
rho_HI = fN_default.calculate_rhoHI(z, (Nmin, Nmax))
```

Most useful for DLA calculations.

### 1.7.4 fN Constraint

The FNConstraint class organizes constraints on  $f(N)$ . It is likely to be used mostly to read in existing data:

```
pyigm_path = imp.find_module('pyigm')[1]
fn_file = pyigm_path+'/data/fN/fN_constraints_z2.5_vanilla.fits'
fn_cs = FNConstraint.from_fitsfile(fn_file)
```

### 1.7.5 Mock Forest

One may generate mock spectra of the Ly $\alpha$  forest with the mockforest module. One inputs a wavelength array, source redshift, and FNModel and the mock spectrum is generated:

```
mock_spec, HI_comps, _ = pyigm.mockforest.mk_mock(wave, zem, fN_model)
```

### 1.7.6 Measured teff

The estimated teff values from observational studies are encoded in the lyman\_alpha\_obs() method. These draw upon the Kirkman+07, Kirkman+05 and Becker+13 studies. Here is an example call:

```
from pyigm.fN import tau_eff as pyteff
teff = pyteff.lyman_alpha_obs(1.)
# Array
teff = pyteff.lyman_alpha_obs(np.linspace(1., 4., 100))
```

A float or ndarray is returned depending on the type of input. The code raises an error if the input redshift exceeds zmax which is currently set to 4.9.

## 1.8 IGMGalaxyField Class

### 1.8.1 Notebooks

### 1.8.2 Overview

This Class is designed to enable analysis between galaxies and absorbers in a chosen field. In particular, cross-correlation analysis. Although, in many ways this is the starting point for essentially any analysis of galaxies with the IGM.

### 1.8.3 Instantiation

Instantiation only requires the field coordinates, presumably near the center.:

```
field = ('PG1407+265', 212.349634*u.deg, 26.3058650*u.deg)
lfield = igmf.IgmGalaxyField((field[1], field[2]), name=field[0], verbose=True)
```

### 1.8.4 Attributes/Properties

Variable	Type	Description
name	str	Given name for the field
cosmo	astropy.cosmology	Given name for the field
igm	??	A means of conveniently storing IGM system info
targets	astropy.Table	Table of target info
galaxies	astropy.Table	Table of galaxy info
observing	astropy.Table	Table of info on observing the galaxies
selection	??	Object to enable calculation of the galaxy selection function

### 1.8.5 Methods

#### Impact Parameter

Calculate  $\rho$ , the projected impact parameter from a given object at a given redshift to the line-of-sight (LOS) coordinate. By default, the projected impact parameter is calculated in co-moving coordinates and the LOS is assumed to be the field coordinate.:

```
rho = lfield.calc_rhoimpact(obj)
```

#### Observed Galaxies

Generate a table of the observed galaxies in the field within a given angular radius of the field coord.:

```
targ, dates, idx = lfield.get_observed(5.*u.arcmin)
```

#### Unobserved Galaxies

Generate a table of the unobserved galaxies in the field within a given angular radius of the field coord.:

```
need_targ = lfield.get_unobserved(5.*u.arcmin)
```

#### Associated Galaxies

Generate a table of the galaxies “associated” to a given LOS.:

```
close_gal, rho = lfield.get_associated_galaxies(0.13, R=300*u.kpc)
```

#### Mask Date

Returns a list of the date(s) when a given mask was observed.:

```
dates = lfield.get_mask_obsdate('PG1407_may_mid2')
```

## Clean Duplicates

Returns a version of a table (e.g. targets, galaxies) without duplicates. The table has to have columns for sky coordinates (e.g. RA, DEC) and the duplication criteria is based on an angular tolerance (usually small; default is  $tol = 1 \text{ u.arcsec}$ ). Currently, the duplication conflict is solved by only keeping the first entry but we expect other methods will be available in the future.:

```
targets = lfield.targets
clean_targets = lfield.clean_duplicates(targets, tol=1*u.arcsec, method='first')
```

## 1.9 IGMSysystem

### 1.9.1 Notebooks

#### Examples for the LLSSysystem Class (v1.1.1)

```
%matplotlib inline
```

```
# imports
try:
    import seaborn as sns; sns.set_style("white")
except:
    pass

from linetools.spectra.xspectrum1d import XSpectrum1D
from pyigm.abssys import utils as igmau
from pyigm.abssys.lls import LLSSysystem
```

#### Simple Instantiate

```
lls = LLSSysystem((0.*u.deg, 0.*u.deg), 2.5, None, NHI=17.9)
lls
```

```
[LLSSysystem: 00:00:00 +00:00:00, zabs=2.5, logNHI=17.9, tau_LL=5.03537, [Z/H]=0 dex]
```

#### Generate/plot a model

```
# Fill HI lines
lls.fill_lls_lines()
```

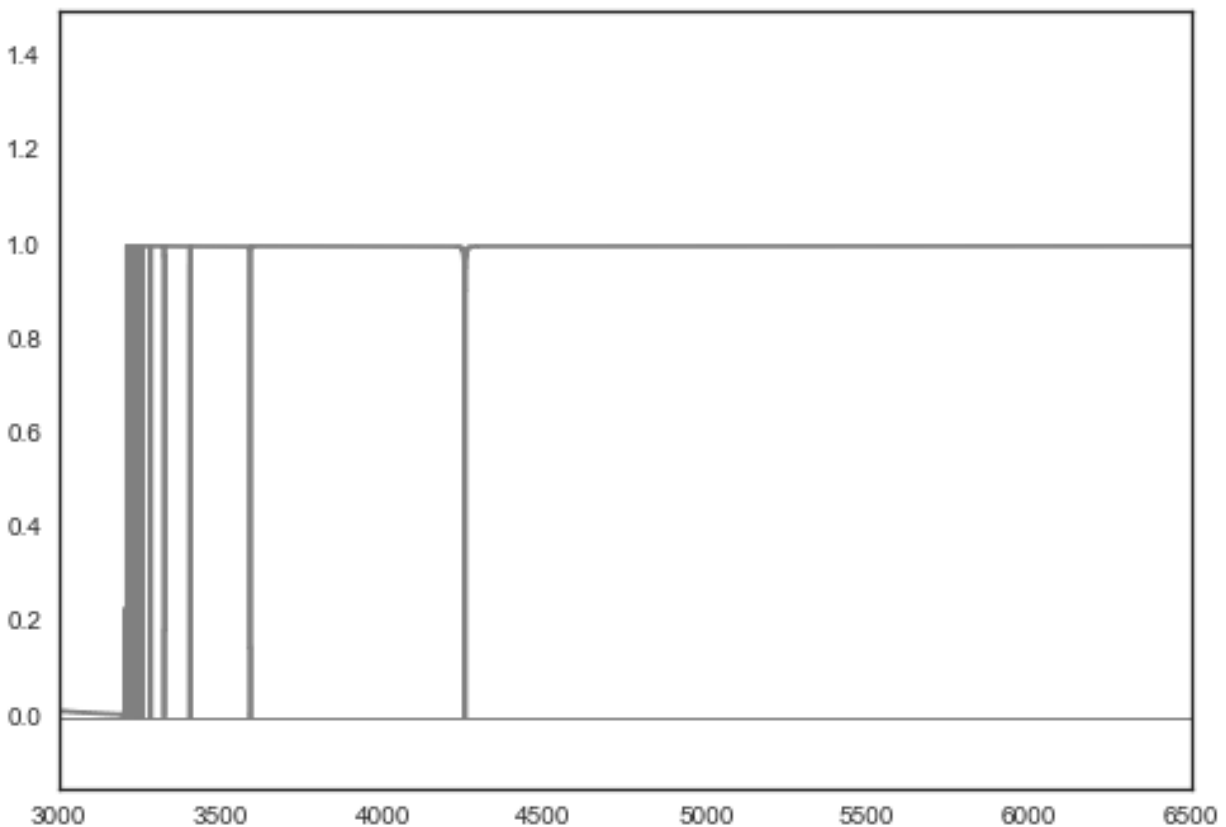
```
WARNING: UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS standard.
↳ Suggested: 10**-1 nm. [astropy.units.format.utils]
WARNING: astropy:UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS
↳ standard. Suggested: 10**-1 nm.
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton03_table2.fits.gz
read_sets: Using set file --
/Users/xavier/local/Python/linetools/linetools/lists/sets/llist_v0.4.ascii
```

```
# Generate a spectrum
wave = np.arange(3000., 6500)
npix = len(wave)
spec = XSpectrum1D.from_tuple((wave*u.AA,np.ones(npix)))
```

```
# Model
model = lls.flux_model(spec)
model.plot()
```

```
/Users/xavier/local/Python/linetools/linetools/analysis/voigt.py:177: UserWarning:
↳Using a sub-grid wavelength array because the input array is too coarse.
  warnings.warn('Using a sub-grid wavelength array because the input array is too
↳coarse.')
/Users/xavier/local/Python/linetools/linetools/analysis/voigt.py:178: UserWarning:
↳Will return values rebinned to the input array.
  warnings.warn('Will return values rebinned to the input array.')
/Users/xavier/local/Python/linetools/linetools/analysis/voigt.py:213: UserWarning:
↳Rebinned tau back to your input array. Reconsider input
  warnings.warn('Rebinned tau back to your input array. Reconsider input')
```



### From a JXP .dat file

(use of these .dat files is no longer recommended)

```
datfil = 'Data/UM184.z2929.dat'
lls = LLSSystem.from_datfile(dat_file=datfil, tree=os.getenv('LLSTREE'))
lls
```

```
[LLSSystem: 23:50:57.87 -00:52:09.9, zabs=2.93012, logNHI=18.5, tau_LL=20.0462, [Z/
↪H]=0 dex]
```

```
print(lls.nsub)
lls.subsys['A']._datdict
```

```
1
```

```
{u'Fe_H': 0.0,
 u'NH': 0.0,
 u'NHI': 18.5,
 u'NHsig': array([ 1.1,  0.4]),
 u'NHsig': array([ 0.,  0.]),
 u'U': 0.0,
 u'Usig': array([ 0.,  0.]),
 u'VPFIT_file': u'',
 u'alpha_H': 0.0,
 u'bsig': 0.0,
 u'bval': 0.0,
 u'clm_file': u'Abund/UM184.z2929_MAGE.clm',
 u'flg_Fe': 0,
 u'flg_alpha': 0,
 u'flg_low': 0,
 u'logx': 0.0,
 u'sig_Fe_H': 0.0,
 u'sig_a_H': 0.0,
 u'sigx': array([ 0.,  0.]),
 u'zabs': 2.93}
```

```
lls._datdict
```

```
OrderedDict([(u'QSO name', u'UM184'),
              (u'RA (2000)', u'23:50:57.87'),
              (u'DEC (2000)', u'-00:52:09.9'),
              (u'QSO zem', u'3.021000'),
              (u'flg_QSOMag', u'0'),
              (u'QSO Mag', u'0.000000'),
              (u'QSO Survey', u'0'),
              (u'Survey', u'0.00000'),
              (u'Referenc', u''),
              (u'SDSS plt,fib,mjd', u'0      0      0'),
              (u'zabs', u'2.93012'),
              (u'NHI tot', u'18.5000'),
              (u'NHI sig', u'1.1000  0.4000'),
              (u'NH tot', u'0.0000'),
              (u'NH sig', u'0.0000  0.0000'),
              (u'vmn,vmx (2f7)', u'0.0    0.0'),
              (u'fdelv', u'0.00'),
              (u'fmm', u'0.00'),
              (u'fedg', u'0.00'),
              (u'ftpk', u'0.00'),
              (u'flg_M/H', u'0')])
```



```
(u'[M/H] ave', u'0.00'),
(u'sig[M/H]', u'0.00'),
(u'flg_D/H', u'0'),
(u'[D/H]', u'0.000000'),
(u'N subsys', u'1'),
(u'Cloudy Grid File', u'dum.fil'),
(u'System A', u'A'),
(u'A zabs', u'2.930000'),
(u'A NHI', u'18.5000'),
(u'A NHIsig', u'1.1000 0.4000'),
(u'A NH', u'0.0000'),
(u'A NHsig', u'0.0000 0.0000'),
(u'A log x', u'0.0000'),
(u'A sigx', u'0.0000 0.0000'),
(u'A b', u'0.0000'),
(u'A bsig', u'0.0000'),
(u'A Abund file', u'Abund/UM184.z2929_MAGE.clm'),
(u'A U', u'0.00'),
(u'A Usig', u'0.0000 0.0000'),
(u'A flg_low', u'0'),
(u'A flg_alpha', u'0'),
(u'A [alpha/H]', u'0.000000'),
(u'A sig[a/H]', u'0.000000'),
(u'A flg_Fe', u'0'),
(u'A [Fe/H]', u'0.000000'),
(u'A sig[Fe/H]', u'0.000000'),
(u'A VPFIT file', u''))]
```

## Load up ions

```
lls.get_ions(use_Nfile=True)
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton03_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton00_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/verner94_tab6.fits
```

```
WARNING: UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS [astropy.units.core]
WARNING:astropy:UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/EUV_lines.ascii
read_sets: Using set file --
/Users/xavier/local/Python/linetools/linetools/lists/sets/llist_v0.4.ascii
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
```

```
lls._ionN
```

## Examples for the DLASystem Class (v1.0)

```
%matplotlib inline
```

```
# imports
try:
    import seaborn as sns; sns.set_style("white")
except:
    pass

from linetools.spectra.xspectrum1d import XSpectrum1D
from pyigm.abssys.dla import DLASystem, DLASurvey
```

### Simple Instantiate

```
dla = DLASystem((0.*u.deg, 0.*u.deg), 2.5, None, NHI=20.55)
dla
```

```
<DLASystem: 00:00:00 00:00:00, 2.5, NHI=20.55, Z/H=0>
```

### From a JXP .dat file

(use of these .dat files is no longer recommended)

```
dla = DLASystem.from_datfile('Data/PH957.z2309.dat', tree=os.environ.get('DLA'))
dla
```

```
<DLASystem: 01:03:11.38 13:16:16.7, 2.309, NHI=21.37, Z/H=-1.41>
```

```
dla._datdict
```

```
OrderedDict([(u'QSO name', u'PH957'),
              (u'RA (2000)', u'01:03:11.38'),
              (u'DEC (2000)', u'+13:16:16.7'),
              (u'QSO zem', u'2.690000'),
              (u'flg_QSOMag', u'2'),
              (u'QSO Mag', u'16.600000'),
              (u'zabs', u'2.309000000'),
              (u'NHI', u'21.370'),
              (u'sig(NHI)', u'0.080 0.000'),
              (u'Abund file', u'Abund/PH957.z2309.clm'),
              (u'flg_Fe', u'1'),
              (u'[Fe/H]', u'-1.849'),
              (u'sig(Fe)', u'0.042'),
              (u'flg_Zn', u'1'),
              (u'[Zn/H]', u'-1.554'),
              (u'sig(Zn)', u'0.021'),
              (u'flg_Alpha', u'4'),
              (u'[Alpha/H]', u'-1.410'),
              (u'sig(Alpha)', u'0.010'),
              (u'flg_low', u'1'),
              (u'hi res file', u'/HIRES/RedData/PH957/PH957_f.fits'),
```

```
(u'low_wav', u'1741.5490'),
(u'low_vmn,vmx (2f7)', u'-25.0 80.0'),
(u'fdelv', u'56.00'),
(u'fmm', u'0.36'),
(u'fedg', u'0.36'),
(u'ftpk', u'0.56'),
(u'flg_CII', u'9'),
(u'N(CII*)', u'13.590'),
(u'sig(CII*)', u'0.050'),
(u'flg_civ', u'1'),
(u'civ hi res file', u'/HIRES/RedData/PH957/PH957_f.fits'),
(u'civ_wav', u'1548.1950'),
(u'civ_vmn,vmx (2f7)', u'-60.0 100.0'),
(u'civ fdelv', u'104.01'),
(u'civ fmm', u'0.23'),
(u'civ fedg', u'0.69'),
(u'civ ftpk', u'-0.62'),
(u'civ fdv', u'-1.86'),
(u'civ frto', u'1.93'),
(u'civ fnmm', u'-0.04'),
(u'civ ftvm', u'10.89'),
(u'E(B-V)', u'0.035'),
(u'Filt 124816', u'0'),
(u'Slit 0nls2y', u'0'),
(u'QSO Survey', u'3'),
(u'Survey Mag', u'16.60'),
(u'References', u'W94, PW99, P01, WPRR08'),
(u'flg_mtl', u'4'),
(u'[M/H]', u'-1.410'),
(u'sig([M/H])', u'0.081'),
(u'SDSS plt,fib,mjd', u'0, 0, 0'),
(u'VPFIT FILE', u''),
(u'flg_CI', u'0'),
(u'N(CI)', u''),
(u'sig(CI)', u''),
(u'flg_H2', u'0'),
(u'N(H2)', u''),
(u'sig(H2)', u'')])
```

## Load up ions

```
dla.tree
```

```
"/u/xavier/DLA/"
```

```
clm_fil = dla.tree+dla._datdict['Abund file']
print(clm_fil)
dla.get_ions(use_Nfile=True)
```

```
WARNING: UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS standard.
↳Suggested: 10**-1 nm. [astropy.units.format.utils]
WARNING:astropy:UnitsWarning: The unit 'Angstrom' has been deprecated in the FITS_
↳standard. Suggested: 10**-1 nm.
```

```
/u/xavier/DLA/Abund/PH957.z2309.clm
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton03_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/morton00_table2.fits.gz
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/verner94_tab6.fits
```

```
WARNING: UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS [astropy.units.core]
WARNING:astropy:UnitsWarning: '0.1nm' did not parse as fits unit: Numeric factor not
↳supported by FITS
```

```
linetools.lists.parse: Reading linelist ---
/Users/xavier/local/Python/linetools/linetools/data/lines/EUV_lines.ascii
read_sets: Using set file --
/Users/xavier/local/Python/linetools/linetools/lists/sets/l1list_v0.4.ascii
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1454.84 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1741.55 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=2062.66 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

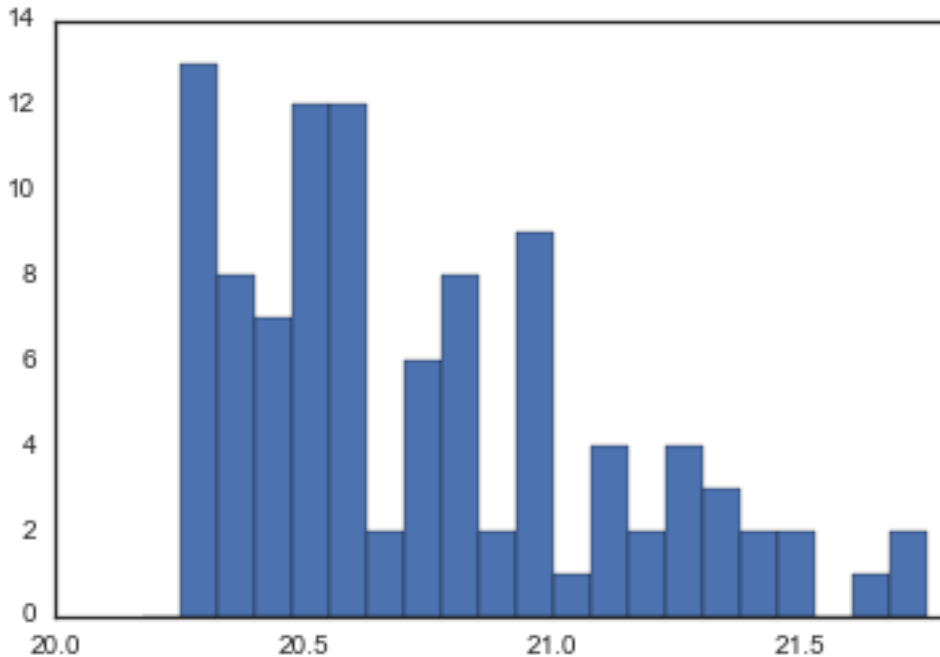
```
dla._ionN
```

## Survey

```
dlas = DLASurvey.default_sample()
```

```
Read 100 files from Lists/Neeleman13.lst in the tree /u/xavier/DLA/
```

```
xdb.xhist(dlas.NHI)
```



```
dlas.fill_ions(use_Nfile=True)
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=988.773 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1548.19 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1253.81 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1467.26 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abssystem.py:158: UserWarning:
↳Input AbsComponent with does not match AbsSystem rules. Not appending
```

```
warnings.warn('Input AbsComponent with does not match AbsSystem rules. Not appending
↳')
```

```
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1526.71 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=2056.26 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=2382.77 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1048.22 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1751.92 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1709.6 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
```

```
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1304.37 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1393.76 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1560.31 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=2062.23 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=2344.21 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5020 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1134.41 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1854.72 Angstrom does not match component rules. Not
↳appending
```

```
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1123.1350 Angstrom for your input 1125.4478 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.2080 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1703.41 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1317.22 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
```

```
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5020 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1123.1350 Angstrom for your input 1125.4478 Angstrom
WARNING: Using 1277.1900 Angstrom for your input 1277.2450 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1334.53 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1199.55 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1134.98 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1302.17 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1039.23 Angstrom does not match component rules. Not_
↳ appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
```



```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1260.42 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.2080 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1608.45 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1611.2 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1550.77 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```
WARNING: Using 1682.1274 Angstrom for your input 1682.1500 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
WARNING: Using 1910.9538 Angstrom for your input 1910.9380 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.2080 Angstrom
WARNING: Using 1277.1900 Angstrom for your input 1277.2450 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1910.61 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1190.42 Angstrom does not match component rules. Not
↳appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
```

```

WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1355.5977 Angstrom for your input 1347.2400 Angstrom
WARNING: Using 1400.4400 Angstrom for your input 1400.4000 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.6000 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom

```

```

/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1200.71 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1656.93 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))

```

```

WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom

```

```

/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1240.39 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1370.13 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))

```

```

WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1123.1350 Angstrom for your input 1125.4478 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1277.1900 Angstrom for your input 1277.2450 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom

```

```

/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳Input absline with wrest=1142.37 Angstrom does not match component rules. Not
↳appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↳appending'.format(absline.wrest))

```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1328.83 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
```

```
WARNING: Using 1012.4950 Angstrom for your input 1012.5020 Angstrom
WARNING: Using 1062.6640 Angstrom for your input 1062.6620 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1467.2590 Angstrom for your input 1466.2120 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1193.29 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1066.66 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1112.05 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1144.94 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1081.87 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1143.23 Angstrom does not match component rules. Not
↪appending
  warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
```

```
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1355.5977 Angstrom for your input 1347.2400 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1190.2030 Angstrom for your input 1190.2080 Angstrom
```

```

WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1560.6820 Angstrom for your input 1574.5503 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1910.6123 Angstrom for your input 1910.7500 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1703.4119 Angstrom for your input 1703.4050 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom

```

```

/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1121.97 Angstrom does not match component rules. Not
↪appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1096.88 Angstrom does not match component rules. Not
↪appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1063.18 Angstrom does not match component rules. Not
↪appending
    warnings.warn('Input absline with wrest={:g} does not match component rules. Not
↪appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↪Input absline with wrest=1063.97 Angstrom does not match component rules. Not
↪appending

```

```
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1133.67 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5020 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
```

```
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1055.26 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1200.22 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=950.885 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=948.686 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
/Users/xavier/local/Python/linetools/linetools/isgm/abscomponent.py:189: UserWarning:
↳ Input absline with wrest=1259.52 Angstrom does not match component rules. Not_
↳ appending
warnings.warn('Input absline with wrest={:g} does not match component rules. Not_
↳ appending'.format(absline.wrest))
```

```
WARNING: Using 2056.2569 Angstrom for your input 2056.2539 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1012.4950 Angstrom for your input 1012.5020 Angstrom
WARNING: Using 1083.9937 Angstrom for your input 1083.9900 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
```

```
WARNING: Using 1190.2030 Angstrom for your input 1190.2080 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
WARNING: Using 1532.5330 Angstrom for your input 1533.4312 Angstrom
WARNING: Using 1122.5240 Angstrom for your input 1122.5260 Angstrom
WARNING: Using 1250.5780 Angstrom for your input 1250.5840 Angstrom
WARNING: Using 1253.8050 Angstrom for your input 1253.8110 Angstrom
```

```
CII_clms = dlas.ions((6,4))
CII_clms
```

## 1.9.2 Overview

The IGMSystem class is designed to organize and analyze an absorption system. This is generally constructed of one or more (AbsComponent). The base class is abstract, i.e. one must instantiate one of its flavors (e.g. HILyman, MgII, LLS, DLA).

By definition, an IGMSystem is a unique collection of absorption components. It is specified by:

Property	Variable	Type	Description
RA, DEC	radec	tuple or coord	RA,DEC in deg or astropy.coordinate
Redshift	z	float	absorption redshift
Velocity limits	vlim	Quantity array	-/+ velocity limits of the system

## 1.9.3 Instantiation

The IGMSystem Class may be instantiated in a few ways. The default sets the properties listed above:

```
gensys = GenericIGMSystem((15.23*u.deg, -23.22*u.deg), 1.244, [-500, 500]*u.km/u.s,
↳NHI=16.)
```

More commonly, one will instantiate with one or more AbsComponent objects:

```
# HI Lya, Lyb
radec = SkyCoord(ra=123.1143*u.deg, dec=-12.4321*u.deg)
lya = AbsLine(1215.670*u.AA, z=2.9239)
lya.analy['vlim'] = [-300., 300.]*u.km/u.s
lyb = AbsLine(1025.7222*u.AA)
lyb.analy['vlim'] = [-300., 300.]*u.km/u.s
lyb.setz(lya.attrib['z'])
abscomp = AbsComponent.from_abslines([lya, lyb])
abscomp.coord = radec
# Finish
HISys = HISystem.from_components([abscomp])
```

## 1.9.4 Attributes

## 1.9.5 Sub Classes

### Generic

A catch-all subclass for IGMSystem.

## HI

A subclass for HI systems. All of the components are required to have  $Z_{\text{ion}}=(1,1)$ .

## LLS

An IGMSystem class for LLS. One is required to set the NHI keyword upon instantiation. See the *Examples for the LLSSystem Class (v1.1.1)* notebook for more.

## DLA

An IGMSystem class for DLA. One is required to set the NHI keyword upon instantiation and it must equal or exceed 20.3. See the *Examples for the DLASystem Class (v1.0)* notebook for more.

## 1.9.6 Plots

## 1.9.7 Methods

## 1.9.8 Output

## 1.10 Metallicity

### 1.10.1 Notebooks

### 1.10.2 Overview

pyigm contains (or will contain) a set of classes and methods to enable analysis of the metallicity of gas in the IGM/CGM.

### 1.10.3 MetallicityPDF

The MetallicityPDF Class holds the PDF of metallicity values for a system (which need not be specified). By default, the analysis is performed in log10 space. Here is a simple instantiation:

```
ZH = np.linspace(-5, 0., 25)
pdf = np.exp(-(ZH+1.5)**2/0.2**2)
# Class
mpdf = MetallicityPDF(ZH, pdf)
# Mean
print('Mean of the PDF is {:g}'.format(mpdf.meanZH))
```

### Attributes/Properties

Variable	Type	Description
meanZH	float	Weighted Mean of $[Z/H]$
medianZH	float	Median of the cumulative PDF (log space)



## Methods

One can calculate the bounds of a confidence interval with:

```
mpdf.confidence_limits(0.68)
```

One can combine two MetallicityPDF's with math. The resultant PDF is **not** normalized.:

```
sum_pdf = mpdf + mpdf2
sum_pdf.normalize()
```

## Plots

One can generate a quick histogram plot of the PDF with:

```
mpdf.hist_plot()
```

## IGM/CGM Surveys

### GUIs

## 1.11 IGMGuesses GUI

### 1.11.1 Overview

IGMGuesses is a GUI for an straightforward identification of absorption features in the spectra of background sources (e.g. QSOs, GRBs, stars, galaxies, etc). IGMGuesses is not meant to provide an optimal Voigt profile fitting of absorption features, but to aid with the identification of absorption lines and provide reasonable first guesses on the main parameters of absorption profiles ( $z$ ,  $N$ ,  $b$ ); these initial guesses should be then used for subsequent Voigt profile fitting. Still, IGMGuesses does provide an easy handling of blends, which can become a problem for high-redshift or high-S/N data.

### 1.11.2 Running IGMGuesses

IGMGuesses should be called from terminal using the script *pyigm\_igmguesses*:

```
>pyigm_igmguesses in_file [-h] [-o OUT_FILE] [-p PREVIOUS_FILE] [--fwhm FWHM]
                        [--n_max_tuple N_MAX_TUPLE] [--min_strength MIN_STRENGTH]
                        [--min_ew MIN_EW]
```

Where *in\_file* is the input spectrum. This file is usually a fits table, with columns for wavelength, flux and flux uncertainty. IGMGuesses works on a continuum-normalized spectrum, thus it has the extra requirement that the *in\_file* also contains an estimation of the continuum. An easy way to make sure that *in\_file* has a compatible format is by using the output file of linetools' *lt\_continuumfit* script (see <https://github.com/linetools/linetools>).

### Optional arguments

<b>-h, --help</b>	Show the help message and exit
-------------------	--------------------------------



**-o OUT\_FILE, --out\_file OUT\_FILE** Output JSON file with absorption model IGM\_model.json

**-p PREVIOUS\_FILE, --previous\_file PREVIOUS\_FILE** Input JSON file with absorption model (if any)

**-fwhm FWHM** FWHM Gaussian smoothing for fitting (in pixels) 3 **-n\_max\_tuple N\_MAX\_TUPLE** Maximum number of transitions per ion species 5

to display

**--min\_strength MIN\_STRENGTH** Minimum strength for transitions to be 0 displayed; choose values between (0,14.7)

**--min\_ew MIN\_EW** Minimum EW (in AA) for transitions to be shown/stored within a component. This is useful 0.005 for getting rid of very weak transitions from the model

**--vlim VLIM** Velocity limit (in km/s) for the display 500.

**--external\_model EXTERNAL\_MODEL** Name of an external spectrum model fits file

**--scale SCALE** Scaling of screen 1.

The number of transitions available for some ions can be excessive for many, especially low-redshift spectra (e.g.  $\text{Cl}$ ,  $\text{Cl}^{**}$ ), so using the default argument “`-n_max_tuple 5`” is a decent starting option. Feel free however to try different values depending on your scientific needs. If an external model is given, you can toggle displaying/hiding it using the keystroke ‘E’ (for external).

### 1.11.3 Component definition

We remind the user that IGMGuesses works on a “absorption component” basis as given by linetools AbsComponent object. Thus, absorption features seen in a spectrum are the result of the superposition of single or multiple absorption components. An absorption component is described by the tuple (ion, N, b, z), where ion is the ion species (e.g.  $\text{HI}$ ,  $\text{CIV}$ ,  $\text{CIII}$ ,  $\text{SiII}$ ), N is the column density, b is the Doppler parameter, z is the redshift. Be aware that this definition may be different from the convention used in other softwares.

In order for IGMGuesses to estimate (N, b, z) one may also need to specify a rest-frame velocity window associated to the component (but note that more generally one could use different velocity windows for individual absorption lines in a given component). This velocity window can also be used to figure out which components are blended or unblended, in the expected subsequent Voigt profile fitting.

### 1.11.4 Line reliability

Because the identification of absorption lines in a given spectrum is not always 100% certain, in IGMGuesses we have incorporated three levels of reliability for a component identification, defined as follows.

- **Certain (flag a):** These include components with multiple transitions where at least two of them are available and visible in the spectrum, and showing the expected ratios and kinematic structure. Common absorption seen at  $z=0$  fall in this category, as well as strong  $\text{HI}$  showing multiple Lyman series transitions. Use ‘P’ to toggle on/off colorful display of components and this will appear in green.
- **Probable (flag b):** These include components from single transition ions that are at the same redshift (within a reasonable velocity window) from another certain component (e.g.  $\text{CIII}$  at the same redshift than a certain  $\text{HI}$ ). Another case where this category should apply is when we have components from ions with multiple transitions but that for some reason only 1 transition is clearly seen (e.g. due to heavy blends, poor S/N, wavelength coverage, etc). Examples of these could be weak  $\text{HI}$  where only  $\text{HI Ly}\alpha$  is visible, or a  $\text{OVI}$  component where one of the transition is blended with something else thus not certain. Use ‘P’ to toggle on/off colorful display of components and this will appear in blue.

- **Uncertain (flag c):** These correspond to those components that based on the user experience are likely to be an incorrect identification. Hopefully components identified in this category will be later replaced by a better identification. These could include an unphysical narrow line, artifacts, etc. Use 'P' to toggle on/off colorful display of components and this will appear in red.
- **Unknown (not implemented yet):** This category is for those absorption features that cannot be explained with current information.

### 1.11.5 Line identification algorithm

Although the users are free to use IGMGuesses as they please, here we provide a simple algorithm for a systematic absorption line identification which has empirically proven to be very efficient.

1. Identify all absorption components available at redshift  $z = 0$ , and assign them to the corresponding reliability category (see above). Depending on the (RA, Dec) of the QSO also inspect  $dv$  close to known structures (e.g.  $dv = -200 \text{ km/s}$  for sightlines close to Andromeda galaxy).
2. Identify all absorption components available at redshift  $z = z_{\text{qso}}$ , and assign them to the corresponding reliability category (see above).
3. Identify HI components showing at least two transitions (e.g. Ly-alpha and Ly-beta, Ly-beta and Ly-gamma, etc), starting at  $z=z_{\text{qso}}$  until  $z=0$ , and assign them to the 'certain' category. This classification includes the full Lyman series transitions of the component available in the spectrum.
4. Identify all possible metal absorption components within a reasonable rest-frame velocity window ( $dv$ ) from each HI redshift found in the previous step and assign them to the corresponding reliability category (see above).
5. Assume all the unidentified absorption features to be HI Ly $\alpha$  starting from  $z=z_{\text{qso}}$  down to  $z=0$ , and assign them to the 'possible' category. Then repeat step (iv).

### 1.11.6 Basic IGMGuesses usage

Once IGMGuesses is launched from terminal, a GUI will appear with four main panels, these are:

- **Velocity Windows (left):** This is the main graphic display where different transitions of different ion are shown at a given redshift (see top). Transitions of the same ion are grouped by the same color sorted by strength. The model is plotted on top of the spectrum as a brown line. The spectrum  $\pm 1$  sigma uncertainty is plotted around the zero level in arbitrary units (re-scaled for clarity). Residuals are plotted around the zero level in the same units as the uncertainty, which helps to assess the statistical significance of absorption/emission features. You can add/remove columns or rows by using the keystrokes 'C'/'c' or 'K'/'k', respectively. You can also go to the previous/next "page" using the keystrokes '-/'=''. Other navigation/display options are available (by pressing '?' within IGMGuesses the full option list will be generated in the terminal)
- **Component Widget (top right):** This is the widget that displays and controls the main parameter for modeling the currently selected absorption component, i.e. (N, b, z). You can slightly increase/decrease the current values using the keystrokes 'N'/'n', 'V'/'v' and '<'/'>', respectively. You can also modify them directly by hand just editing the respective values on the widget itself. After done with editing, make sure you press the "Update" button. Reliability flags must be chosen from the available list (see above for definitions). A string comment can also be entered.
- **Components List (middle right):** This widget displays and allows to navigate between currently defined components. The name of the component includes the redshift and ion species. By selecting a component from this list, its parameters can be modified from the *Component Widget* but in order to move to the corresponding redshift one needs to manually navigate to it (by using the keystroke ^ or the Space Bar; see below)

- **Line List** (*bottom right*): This widget displays the current parent LineList (see `linetools` LineList Class for further details) where ion transitions are selected from. By selecting/unselecting them you can control which transitions are displayed in the *Velocity Windows*. Built-in LineList can be loaded by keystrokes ‘H’ (HI Lyman series), ‘T’ (Common Strong IGM transitions), ‘F’ (Full list of ISM known transitions). Of course, depending on redshift some transitions may or may not be available in the current spectrum; in order to select those available at the current redshift of interest from the current LineList you can use keystroke ‘U’ (update). By doing ‘U’ it will also restrict the subset of lines satisfying *n\_max\_tuple* and *min\_strength* as given in the initialization (see Table above).
- **Adding a component**: Click on the *Velocity Window* associated to the relevant ion transition and define the rest-frame velocity limits by pressing ‘A’ twice (one for each limit). IGMGuesses will use the pixel information within those limits to fit a Voigt profile (convolved with a Gaussian of FWHM as given in the initialization; see above) in order to estimate the (N,b,z) parameters. It only uses the pixels in the selected transition of the given ion for guessing the model parameters. Thus, it is recommended to select a transition that is not saturated, blended or in a poor S/N spectral region, when possible. Still, a model of the component will be displayed encompassing *all* the transitions. Another convenient way to add a component is by using the single keystroke ‘a’, which tells IGMGuesses to use the same velocity window as the previously selected component.

– **Removing a component**: Use ‘D’ to remove the *closest* (in velocity) component to

the cursor position in a given velocity window panel. You can also remove a component regardless of whether is being displayed in the velocity window panels by selecting it from the *Components List* widget and then pressing the keystroke ‘d’.

– **Selecting a component**: Use ‘S’ to select a component from the velocity window

panel at the cursor position (you may need to click on the white area of the velocity panel to make sure IGMGuesses recognizes you are selecting from the velocity panel). You can also select a component directly from the *Components List* widget.

– **Editing a component**: Use the keystrokes N’/’n’, V’/’v’ and ‘</>’, to slightly

increase/decrease the values of (N, b, z), respectively, for the currently selected component. Alternatively, you can modify these values by hand directly from the *Component Widget*. The lower/upper velocity limits of a given component can be modified by using the keystrokes ‘1’/’2’, respectively. Use ‘R’ if you wish to refit the data using the current velocity limits.

There are two ways that you can navigate in redshift space:

- **Panning**: Use the Space Bar to set the redshift given by the the rest-frame velocity of the cursor.
- **By hand**: Use ‘^’ to select the redshift by hand. A pop-up widget should appear.

The superposition of all absorption components identified provides the overall model for the given spectrum. Such model is by default displayed as a brown line. Other options for model visualization include:

- **Colorful display**: use ‘P’ to toggle on/off colorful display in which each component of the model is plotted by different colors depending on their assigned reliability (see above).
- **Show/hide labels**: use ‘L’ to toggle on/off the label of each identified component.

The label is composed by the ion transition, its redshift, and its reliability flag (see above)

- **Show/hide model**: use ‘M’ to toggle on/off the model being displayed.
- **Show/hide external model**: use ‘E’ to toggle on/off the model being displayed.

## 1.12 XFitDLA GUI

### 1.12.1 Overview

XFitDLA GUI is a GUI for the by-hand fit to one or more DLAs in an input spectrum. One can modify both the continuum and DLA properties and write the solution to disk as a JSON file.

### 1.12.2 Running XFitDLA GUI

XFitDLA GUI should be called from the terminal using the script *pyigm\_fitdla*. Here is the usage:

```
usage: pyigm_fitdla [-h] [--out_file OUT_FILE] [--smooth SMOOTH]
                  [--dla_fit_file DLA_FIT_FILE] [--conti_file CONTI_FILE]
                  [--zdla ZDLA] [--NHI NHI]
                  in_file zqso

Parser for FitDLA GUI (v1.0)

positional arguments:
  in_file              Spectral file
  zqso                Use QSO template with zqso

optional arguments:
  -h, --help          show this help message and exit
  --out_file OUT_FILE Output DLA Fit file
  --smooth SMOOTH     Smoothing (pixels)
  --dla_fit_file DLA_FIT_FILE
                      Input DLA Fit file
  --conti_file CONTI_FILE
                      Input continuum spectrum
  --zdla ZDLA         Input DLA redshift
  --NHI NHI           Input DLA NHI
```

There are 2 required arguments: (1) *in\_file* which is the input spectrum file; (2) *zqso* which is an estimate of the quasar (or other source) emission redshift.

#### Optional arguments

Argument	Description	Default
-h, -help	Show the help message and exit	
-o OUT_FILE, -out_file OUT_FILE	Output JSON file with model	DLA_fit.json
-smooth SMOOTH	FWHM Gaussian smoothing for fitting (in pixels)	3
-dla_fit_file DLA_FIT_FILE	Input .JSON file containing a previous DLA fit	
-conti_file CONTI_FILE	Input continuum spectrum file	
-zdla	Input redshift for an initial DLA model	
-NHI	Input NHI value for an initial DLA model	

### 1.12.3 Usage

When the DLA gui launches, you should begin by left clicking in the plot window. This will activate plotting and key strokes (see below for the help message).

From there, navigate to the DLA of interest, add a model, refine it and the continuum, etc.

The Write button saves the current model to disk as a JSON file.

## 1.12.4 Keystrokes

Here is a brief description of the key strokes that control the DLA GUI (also displayed when launching the GUI):

```
i,o      : zoom in/out x limits
I,O      : zoom in/out x limits (larger re-scale)
Y        : zoom out y limits
y        : guess y limits
W        : Show original zoom
t,b      : set y top/bottom limit
l,r      : set left/right x limit
a,m,d    : Add/modify/delete continuum knot
A        : Add a new DLA
g        : Move nearest Lyman line to cursor and reset z
N/n      : Increase/decrease NHI
V/v      : Increase/decrease bvalue
Z/z      : Increase/decrease zabs
D        : Delete DLA
$        : Toggle displaying metal lines
6,7,8,9  : Add forest lines
?        : Print these help notes
Q        : Quit the GUI
```

## 1.13 XFitLLS GUI

### 1.13.1 Overview

XFitLLSGUI is a GUI for the by-hand fit to one or more LLSs in an input spectrum. One can modify both the continuum and LLS properties and write the solution to disk as a JSON file.

### 1.13.2 Running XFitLLSGUI

XFitLLSGUI should be called from the terminal using the script *pyigm\_fitlls*. Here is the usage:

```
wolverine> pyigm_fitlls -h
usage: pyigm_fitlls [-h] [--out_file OUT_FILE] [--smooth SMOOTH]
                  [--lls_fit_file LLS_FIT_FILE]
                  in_file zqso

Parser for FitLLSGUI (v1.0)

positional arguments:
  in_file              Spectral file
  zqso                Use Telfer template with zqso

optional arguments:
  -h, --help          show this help message and exit
  --out_file OUT_FILE  Output LLS Fit file
  --smooth SMOOTH      Smoothing (pixels)
  --lls_fit_file LLS_FIT_FILE
                        Input LLS Fit file
```

There are 2 required arguments: (1) *in\_file* which is the input spectrum file; (2) *zqso* which is an estimate of the quasar (or other source) emission redshift.

### Optional arguments

Argument	Description	Default
-h, -help	Show the help message and exit	
-o OUT_FILE, -out_file OUT_FILE	Output JSON file with model	DLA_fit.json
-smooth SMOOTH	FWHM Gaussian smoothing for fitting (in pixels)	3
-lls_fit_file LLS_FIT_FILE	Input .JSON file containing a previous DLA fit	

### 1.13.3 Usage

When the LLS gui launches, you should begin by left clicking in the plot window. This will activate plotting and key strokes (see below for the help message). You should then refine the continuum (“C”, “1”, “2”).

From there, navigate to the LLS of interest, add a model (“A” or “F”), refine it and the continuum, etc.

The Write button saves the current model to disk as a JSON file.

### 1.13.4 Keystrokes

Here is a brief description of the key strokes that control the DLA GUI (also displayed when launching the GUI):

i,o	: zoom in/out x limits
I,O	: zoom in/out x limits (larger re-scale)
Y	: zoom out y limits
y	: guess y limits
W	: Show original zoom
t,b	: set y top/bottom limit
l,r	: set left/right x limit
C	: Set the continuum normalization to the cursor
1,2	: Tilt the continuum
A	: Add a new LLS at cursor
F	: Add a new LLS automatically near the cursor (probably)
g	: Move nearest Lyman line to cursor and reset z
N/n	: Increase/decrease NHI
V/v	: Increase/decrease bvalue
Z/z	: Increase/decrease zabs
D	: Delete LLS
\$	: Toggle displaying metal lines
6,7,8,9	: Add forest lines
?	: Print these help notes
Q	: Quit the GUI

### Misc

## 1.14 Continua

### 1.14.1 Notebooks

### 1.14.2 Overview

Observational analysis of the IGM generally requires the continuum normalization of the background source observed. pyigm provides a few methods useful for such analysis.

### 1.14.3 Quasars

It is common practice to adopt a quasar template as a first guess for the continuum of these sources. pyigm provides two methods.

#### Telfer

One may generate a quasar continuum using the Telfer+05 template (radio-quiet).

```
from pyigm.continuum import quasar as pyicq
telfer = pyicq.get_telfer_spec(3.)
```

The resultant object is a XSpectrum1D object of the model continuum. One may specify whether to include an average IGM opacity model (see [fN Classes](#) for a description of the fN model). Multiprocessing is employed for speed.

```
telfer = pyicq.get_telfer_spec(3., igm=True, nproc=4)
```

#### WFC3

Individual quasars without apparent, significant Lyman limit absorption that were observed with the HST/WFC3 instrument may provide *real* continua. These may be selected randomly from the full set or by index.

```
wfc3, _ = pyicq.wfc3_continuum(wfc3_indx=0, zqso=2.)
wfc3, idx = pyicq.wfc3_continuum(zqso=2.5)
```

Again, the returned object is a XSpectrum1D object.

## 1.15 pyigm Scripts

There are a number of scripts, many of which are GUIs, provided with pyigm. As regards the GUIs we warn again that Mac users will need to set their matplotlib to something other than MacOSX. See [backends](#).

### 1.15.1 pyigm\_igmguesses

GUI for examining IGM sightlines. See *ref:igmguesses* for details.

### 1.15.2 pyigm\_mkigmsys

Generate a simple JSON file for an IGM System. Useful for things like the GUI `lt_xabssys` in *linetools*.

Here is the usage:

```
usage: pyigm_mkigmsys [-h] [--NHI NHI] [--jcoord JCOORD] [--zem ZEM]
                    [--sigNHI SIGNHI] [--vlim VLIM]
                    itype zabs outfile

Show contents of a JSON file, assuming one of several formats. (v1.1)

positional arguments:
  itype                Type of IGMSys: dla, lls, mgii
  zabs                 Absorption redshift
  outfile              Name of JSON file to create

optional arguments:
  -h, --help            show this help message and exit
  --NHI NHI             log10 NHI value
  --jcoord JCOORD       Coordinates in JXXXXXXXXX.X+XXXXXX.X format
  --zem ZEM             Emission redshift
  --sigNHI SIGNHI       Error in NHI
  --vlim VLIM           Velocity limits in format ###,###
```

And an example:

```
pyigm_mkigmsys dla 3.0 tmp.json --zem=4. --vlim=-232,300 --NHI=20.5 --sigNHI=0.2
```

### 1.15.3 pyigm\_showjson

Simple script to give a brief summary of stuff in a JSON file. Best to use on an IGMSightline file.



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## A

AbsSystem, [16](#)

## C

CGM, [5](#)

continuum, [67](#)

CUBA, [35](#)

## F

fNclass, [38](#)

## I

igmfield, [40](#)

IGMGuesses, [60](#), [64](#), [65](#)

IGMSystem, [42](#)

## M

metallicity, [59](#)

misc, [5](#)