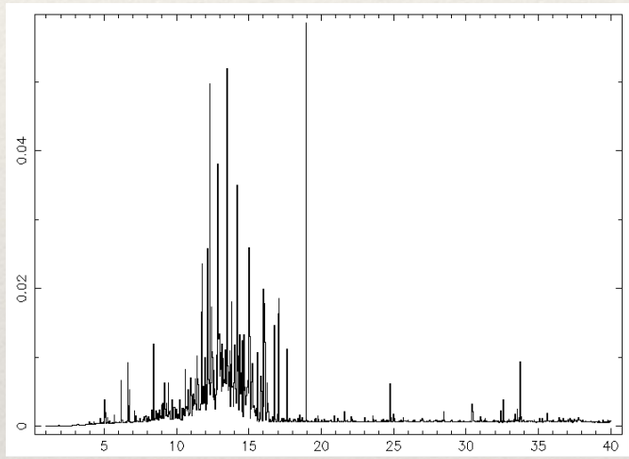
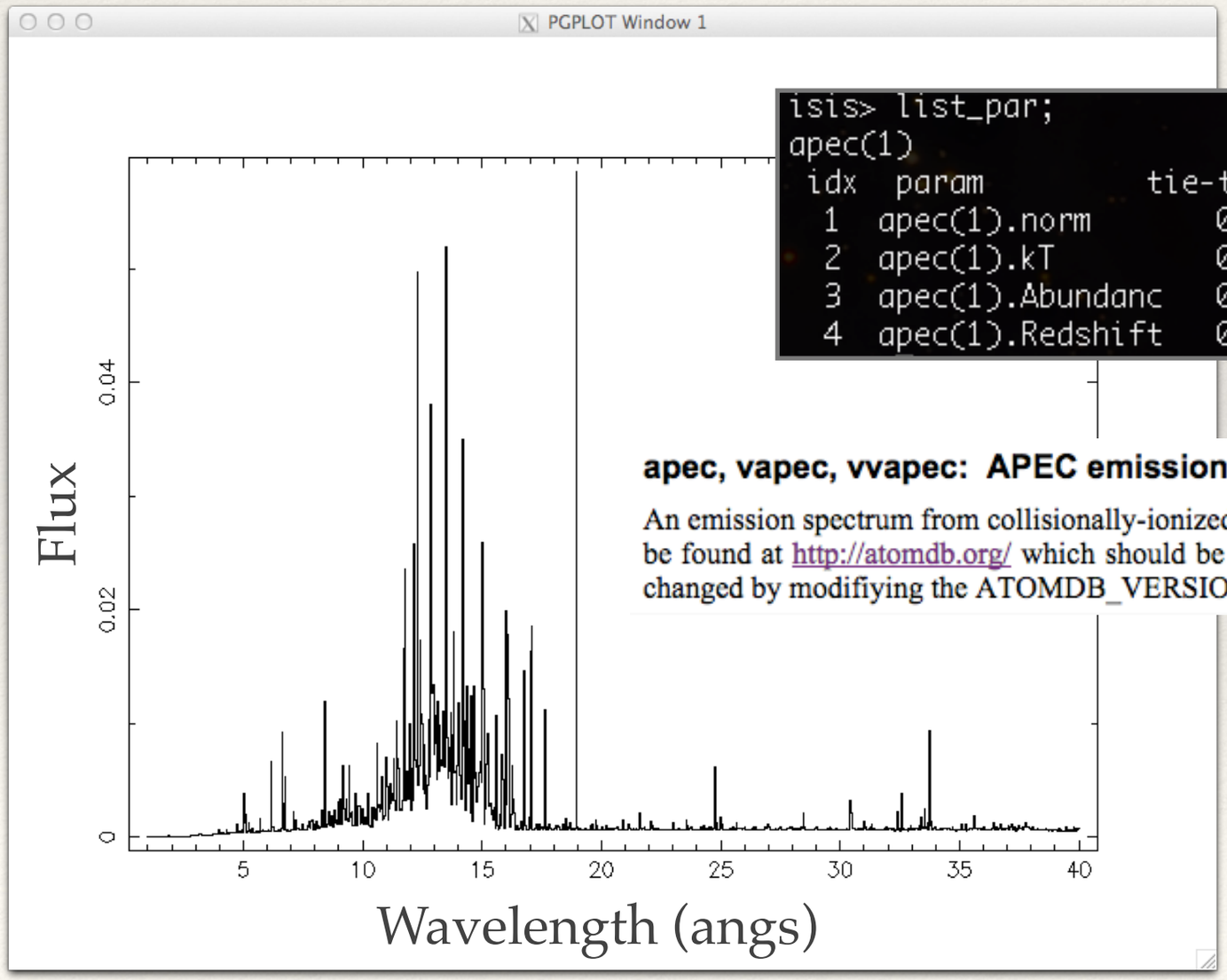
L. Corrales, ISIS Workshop, 18-August-2015

Navigating atomic databases with ISIS



Modeling astrophysical plasmas (apec in XSPEC)

```
isis> (x1, x2) = linear_grid( 1, 40, 1000 );
isis> fit_fun( "apec(1)" );
 Solar Abundance Vector set to angr:  Anders E. & Grevesse N. Geochimica et Cosmochimica Acta 53, 197 (1989)
 Cross Section Table set to bcmc:  Balucinska-Church and McCammon, 1998
isis> y = eval_fun( x1, x2 );
isis> hplot( x1, x2, y );
```



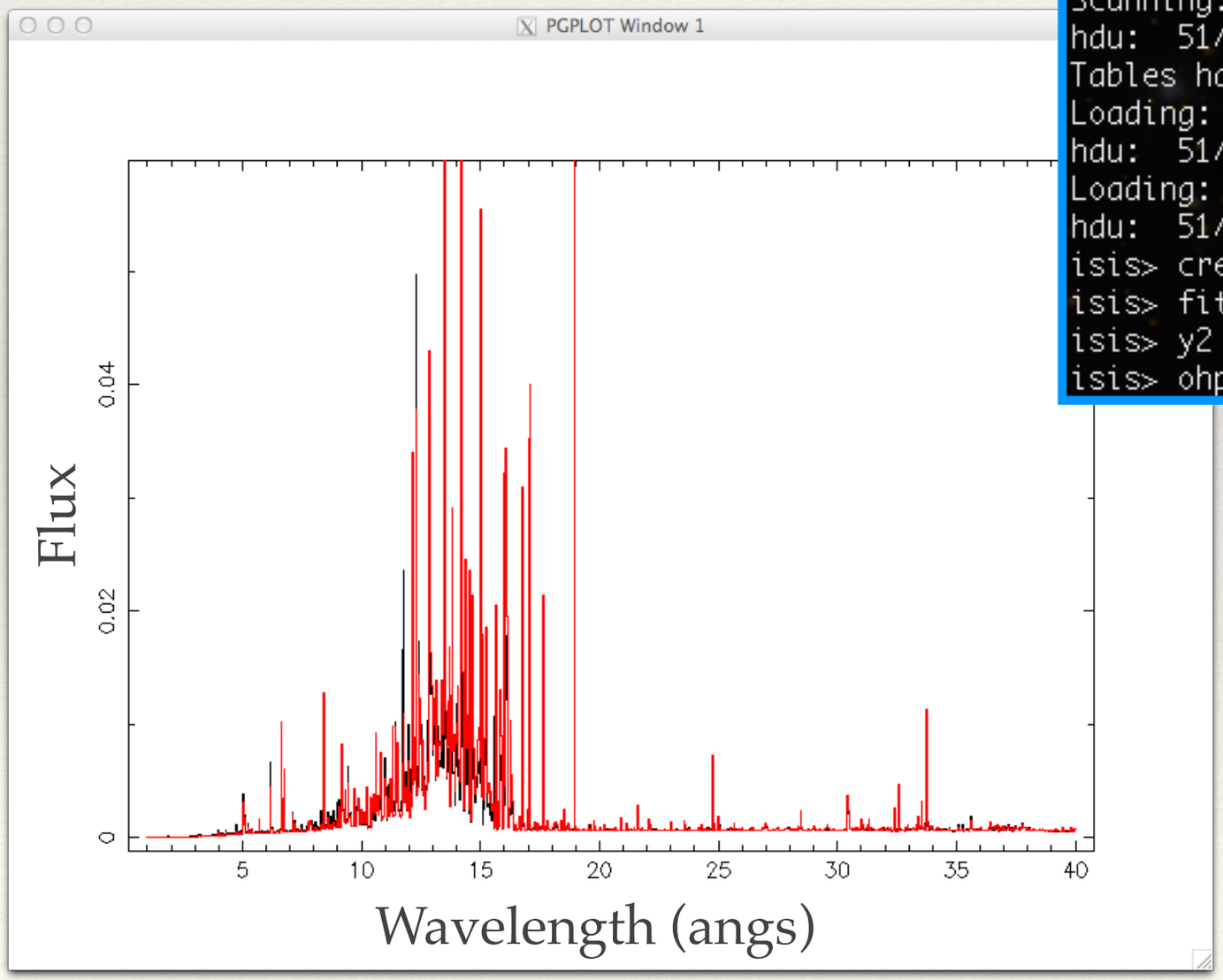

```
isis> list_par;
apec(1)
 idx  param          tie-to  freeze          value          min          max
  1  apec(1).norm        0       0              1            0        1e+10
  2  apec(1).kT          0       0              1        0.008           64  keV
  3  apec(1).Abundanc    0       1              1            0            5
  4  apec(1).Redshift    0       1              0       -0.999           10
```

apec, vapec, vvapec: APEC emission spectrum

An emission spectrum from collisionally-ionized diffuse gas calculated using the ATOMDB code v2.0.2. More information can be found at http://atomdb.org/ which should be consulted by anyone running this model. This default version number can be changed by modifiying the ATOMDB_VERSION string in your Xspec.init file.

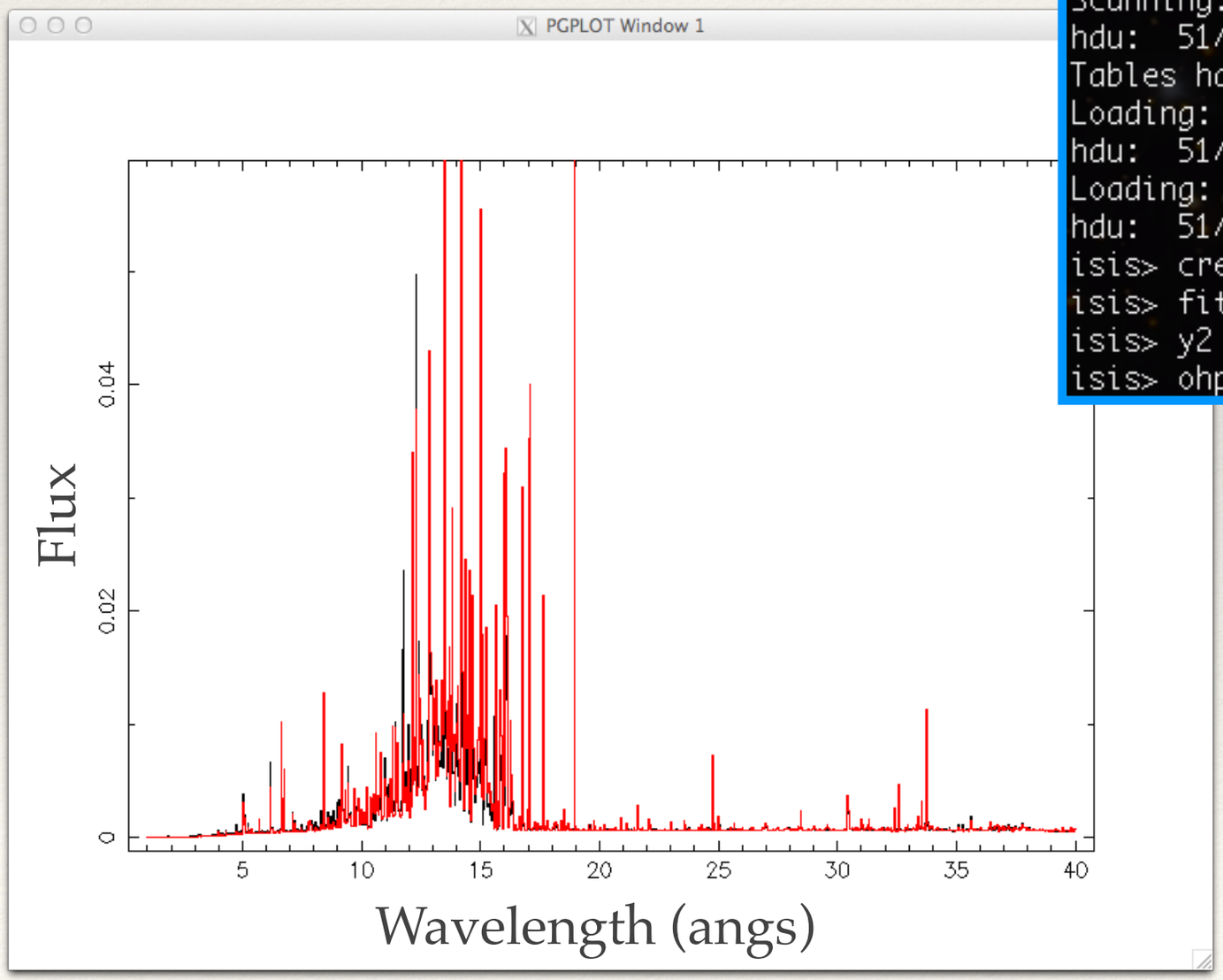
Modeling astrophysical plasmas (ISIS)

```
isis> plasma(aped);
Reading database version 2.0.1 from /nfs/cxc/a1/share/atomdb
Initializing: atomic data
Read 268 energy level files
Read 268 wavelength files
Tables have 914474 lines between 0.6979 and 1e+10 Angstrom
Initializing: emissivity data
Scanning: line emissivity tables [51 hdus]
hdu:  51/51
Tables have 917416 lines between 0.6979 and 1e+10 Angstrom
Loading: line emissivity tables [51 hdus]
hdu:  51/51
Loading: continuum emissivity tables [51 hdus]
hdu:  51/51
isis> create_aped_fun( "myplasma", default_plasma_state );
isis> fit_fun( "myplasma(1)" );
isis> y2 = eval_fun( x1, x2 );
isis> ohplot( x1, x2, y2, 2 );
```



Modeling astrophysical plasmas (ISIS)

```
isis> print( aped );
Reading database version 2.0.1 from /nfs/cxc/a1/share/atomdb
{dir="/nfs/cxc/a1/share/atomdb",
 atomic_data_filemap="filemap",
 abundance="APED/misc/Abundances.fits",
 ion_balance="APED/ionbal/MM98_ionbal.fits",
 line_emissivity="apec_v2.0.1_line.fits",
 continuum_emissivity="apec_v2.0.1_coco.fits"}
```

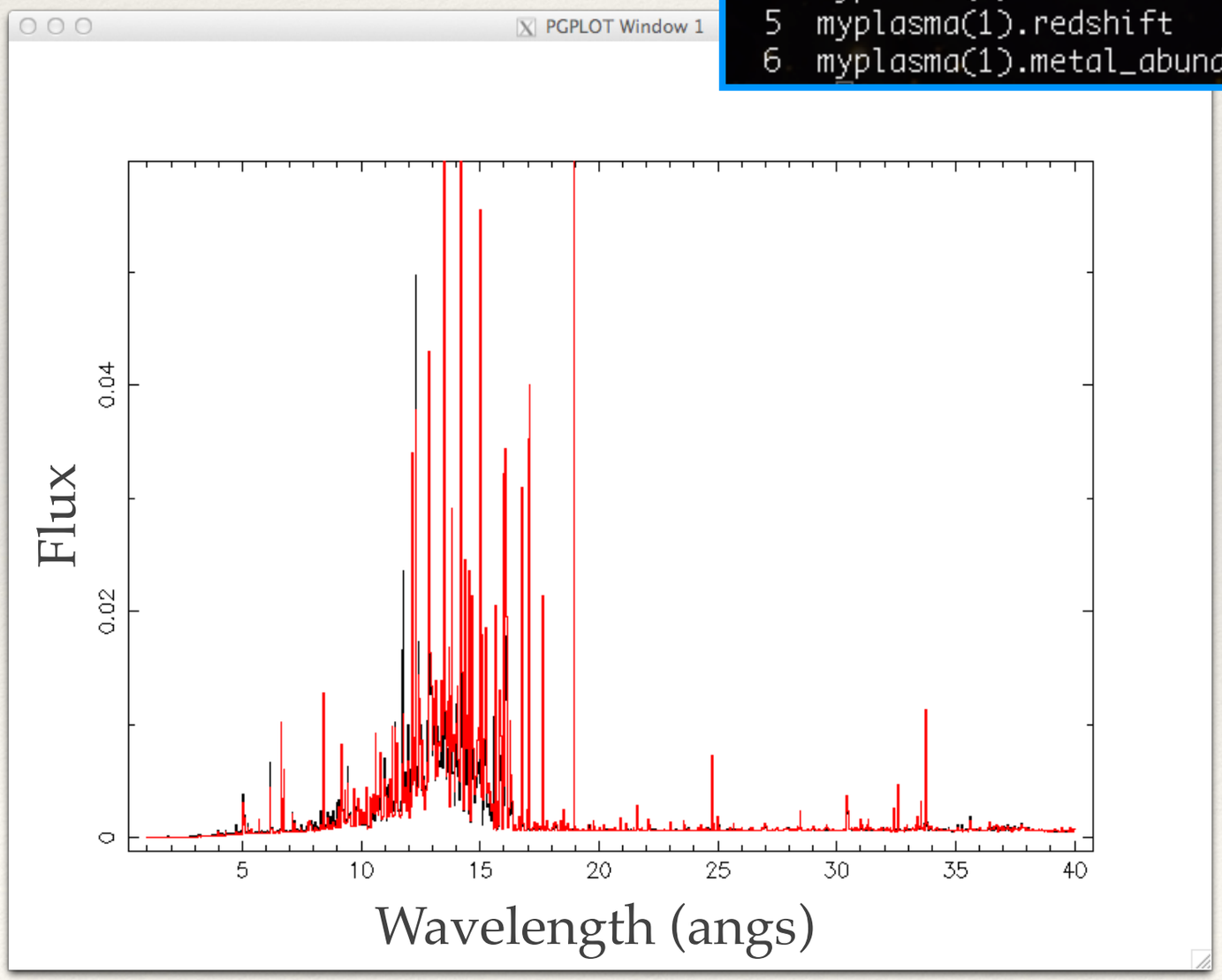



```
isis> plasma(aped);
Reading database version 2.0.1 from /nfs/cxc/a1/share/atomdb
Initializing: atomic data
Read 268 energy level files
Read 268 wavelength files
Tables have 914474 lines between 0.6979 and 1e+10 Angstrom
Initializing: emissivity data
Scanning: line emissivity tables [51 hdus]
hdu:  51/51
Tables have 917416 lines between 0.6979 and 1e+10 Angstrom
Loading: line emissivity tables [51 hdus]
hdu:  51/51
Loading: continuum emissivity tables [51 hdus]
hdu:  51/51
isis> create_aped_fun( "myplasma", default_plasma_state );
isis> fit_fun( "myplasma(1)" );
isis> y2 = eval_fun( x1, x2 );
isis> ohplot( x1, x2, y2, 2 );
```

```
isis> print( default_plasma_state );
{norm=1.0,
 temperature=1e+07,
 density=1.0,
 metal_abund=1.0,
 elem_abund=NULL,
 elem=NULL,
 vturb=0.0,
 redshift=0.0}
```

Modeling astrophysical plasmas (ISIS)

```
isis> list_par;
myplasma(1)
 idx  param                  tie-to  freeze          value          min          max
   1  myplasma(1).norm            0       1              1            0            0
   2  myplasma(1).temperature     0       1          1e+07            0            0
   3  myplasma(1).density         0       1              1            0            0
   4  myplasma(1).vturb           0       1              0            0            0
   5  myplasma(1).redshift        0       1              0            0            0
   6  myplasma(1).metal_abund     0       1              1            0            0
```



Navigating AtomDB with ISIS

SEARCH options

- `wl(min, max);`
- `el_ion(Z[, ion]);`
- `trans(Z[, ion[, upper[, lower]]]);`

returns a boolean array, easy to combine
use where to get a line list:

```
ll = where( wl(18, 20) & el_ion(O) );
```

SCIENCE

- `bl = brightest(n, ll);`
- `line_em(ll, temps[, dens]);`
- `ratio_em(ll[0], ll[1], temps[, dens]);`

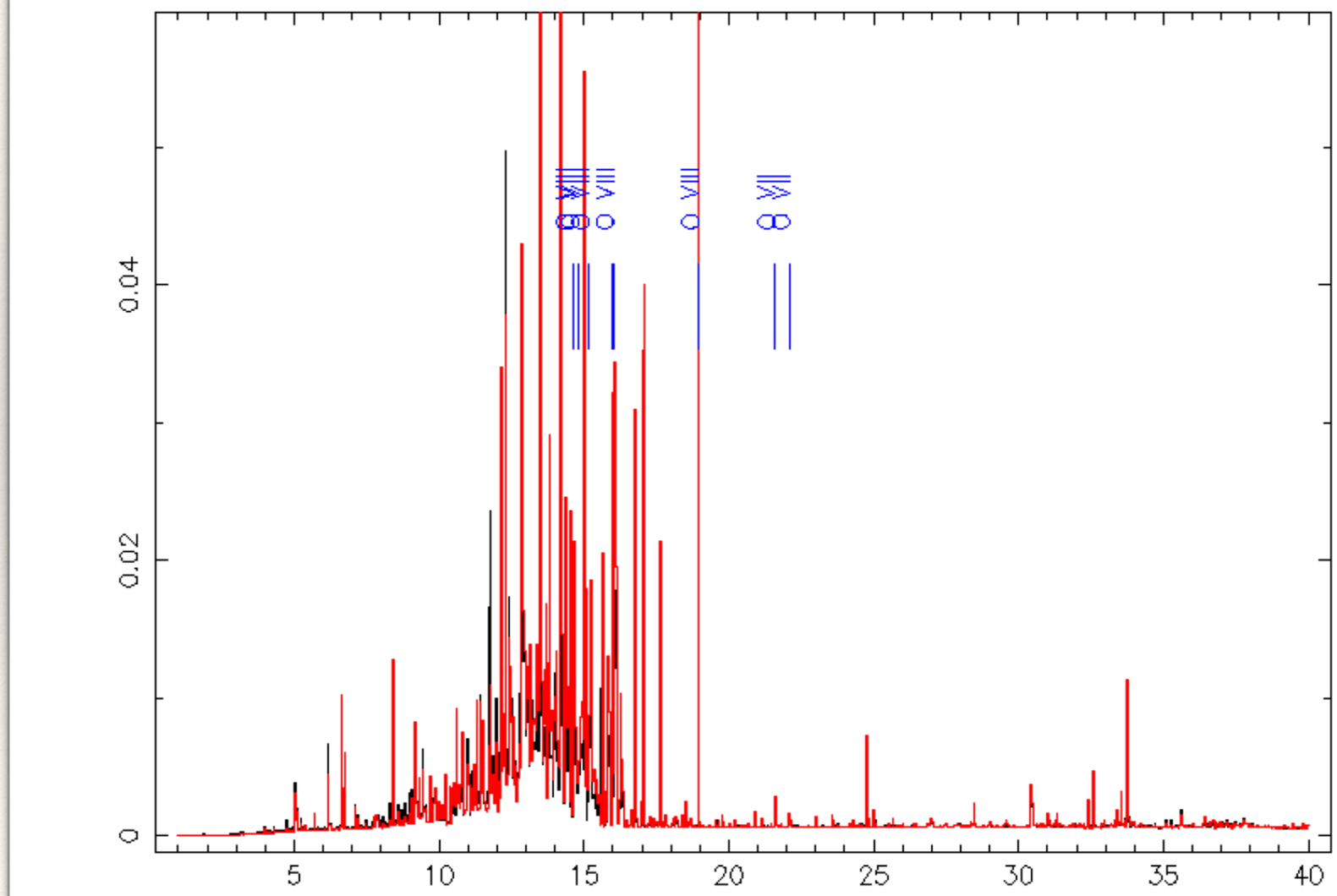
Get database INFO

- `page_group(bl);`
- `plot_group(bl);`
- `s = line_info(bl[0]);`

```
isis> print(s);
{flux=0.04733396397989178,
 lambda=18.96711,
 lambda_err=9.0000001e-05,
 A=2.5658209e+12,
 A_err=1.1754944e-38,
 id=50033,
 ion=8,
 Z=8,
 upper=4,
 lower=1,
 upper_g=4.0,
 lower_g=2.0,
 upper_E=653.77435,
 lower_E=0.0,
 upper_L=1,
 upper_S=0,
 lower_L=0,
 lower_S=0,
 up_name="2p1",
 lo_name="1s1"}
```

```
isis> olines = brightest(10, where( el_ion(0) ) );
isis> page_group( olines );
#  index         ion   lambda    F (ph/cm^2/s)  A(s^-1)  upper lower   label
  50134 *    0 VIII    14.633    6.564e-04   7.951e+10    28     1 6p1 - 1s1
  50091 *    0 VIII    14.821    1.121e-03   1.393e+11    19     1 5p1 - 1s1
  50058 *    0 VIII    15.176    2.431e-03   2.774e+11    12     1 4p1 - 1s1
  50045 *    0 VIII    15.176    1.330e-03   5.562e+11    11     1 4p1 - 1s1
  50041 *    0 VIII    16.006    6.461e-03   6.829e+11     7     1 3p1 - 1s1
  50034 *    0 VIII    16.007    3.902e-03   1.367e+12     6     1 3p1 - 1s1
  50033 *    0 VIII    18.967    4.733e-02   2.566e+12     4     1 2p1 - 1s1
  50031 *    0 VIII    18.973    2.371e-02   2.564e+12     3     1 2p1 - 1s1
  40783 *     0 VII    21.602    2.153e-03   3.430e+12     7     1 1s1 2p1 - 1s2
  49857 *     0 VII    22.098    1.012e-03   9.120e+02     2     1 1s1 2s1 - 1s2
isis> plot_group( olines );
```

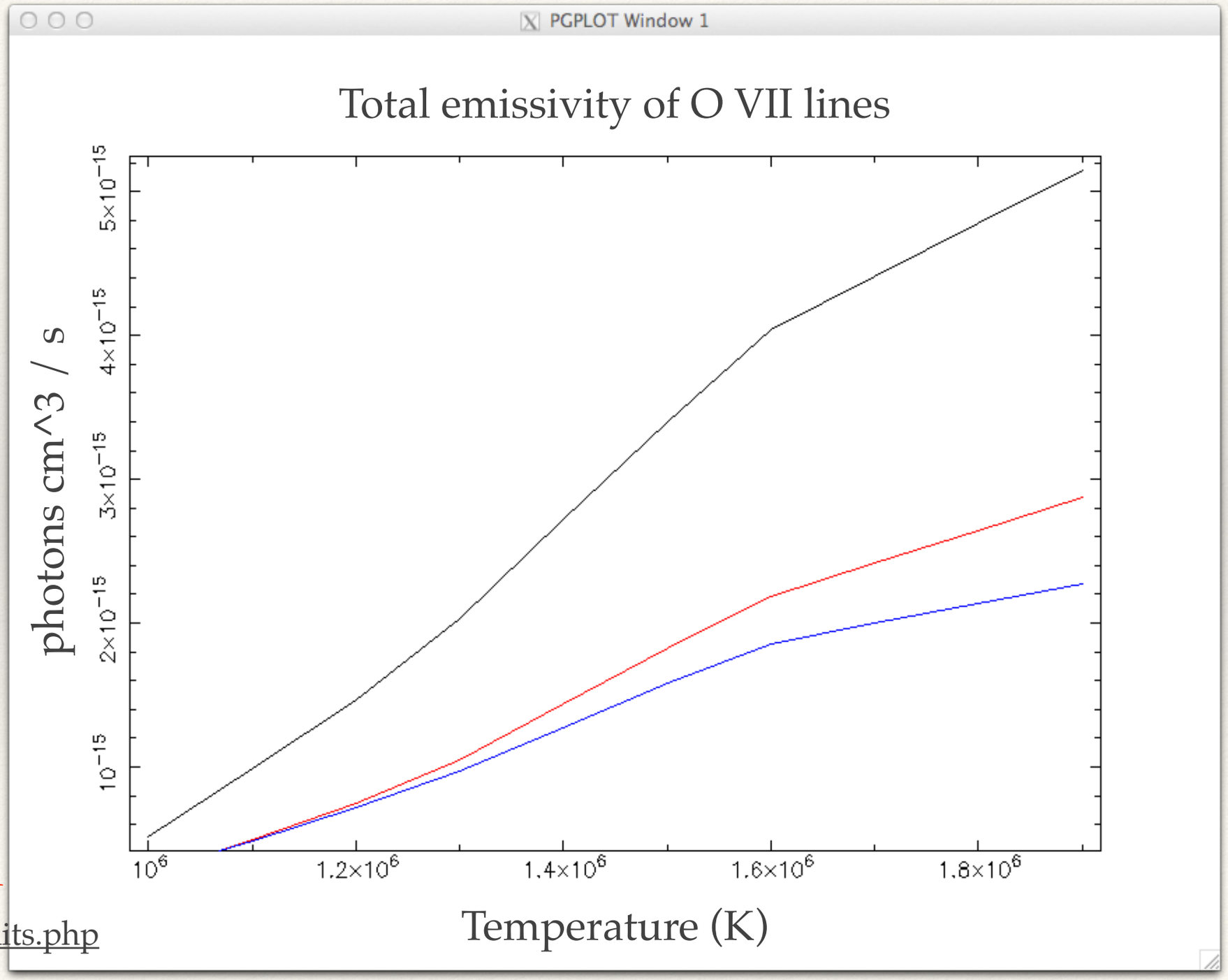
Window 1



```
isis> ovii = brightest(2, where( el_ion(0,7) ) );
isis> temps = [1.e6 : 2.e6 : 1.e5];
isis> emis = line_em( ovii, temps );
```

line_em gives **sum of line-list emissivities**
as a function of temperature (or dens)

```
isis> plot( temps, emis );
isis> oplot( temps, line_em(ovii[0], temps), 2 );
isis> oplot( temps, line_em(ovii[1], temps), 4 );
```

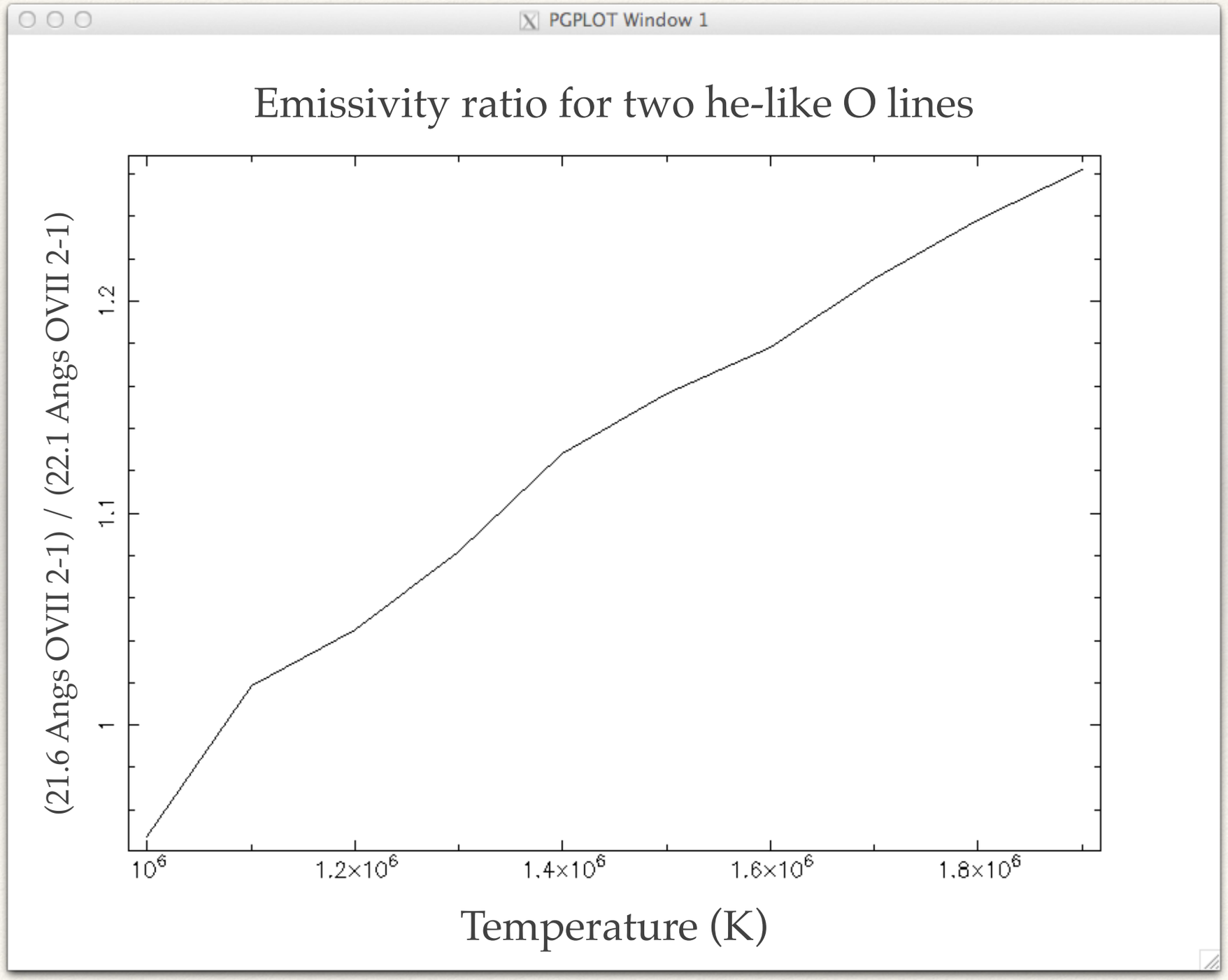


How emissivity is calculated

http://www.atomdb.org/Physics/units.php

```
isis> rat = ratio_em( ovii[0], ovii[1], temps);
isis> plot( temps, rat );
```

ratio_em gives **ratios of emissivities**
as a function of temperature (or dens)
and will keep line-list groups intact



http://space.mit.edu/cxc/analysis/xstardb/

Navigating the *new* XSTAR model databases

Similar philosophy to AtomDB navigation, except:

1. Each model run produces a **separate fits file database**
2. That database file must then be **loaded into a structure**
3. Each **function call acts on the database structure** (instead of environment)

LOAD:

- `db = rd_xstar_output("warmabs_1.fits");`

SEARCH:

- `xstar_wl(db, min, max);`
- `xstar_el_ion(db, Z[, ion]);`
- `xstar_trans(db, Z[, ion[, upper[, lower]]]);`

INFO:

- `xstar_page_group(db, ll[; sort]);`
- `xstar_plot_group(db, ll);`

SCIENCE:

- `xstar_strong(n, db[; wmin, wmax]);`

http://space.mit.edu/cxc/analysis/xstardb/

Navigating the *new* XSTAR model databases

Similar philosophy to AtomDB navigation, except:

1. Each model run produces a **separate fits file database**
2. That database file must then be **loaded into a structure**
3. Each **function call acts on the database structure** (instead of environment)

LOAD:

```
db = rd_xstar_output("warmabs_1.fits");
dbm = xstar_merge(["warmabs_1.fits","warmabs_2.fits"]);
db_grid = xstar_load_grid(filenames);
```

SEARCH:

```
xstar_wl(db, min, max);
xstar_el_ion(db, Z[, ion]);
xstar_trans(db, Z[, ion[, upper[, lower]]]);
```

INFO:

```
xstar_page_group(db, ll[; sort]);
xstar_plot_group(db, ll);
```

SCIENCE:

```
xstar_strong(n, db[; wmin, wmax]);
xstar_run_model_grid(model_info, "path"[; istart]);
xstar_line_prop(db_grid, ll, "luminosity");
xstar_get_grid_par(db_grid, "rlogxi");
```

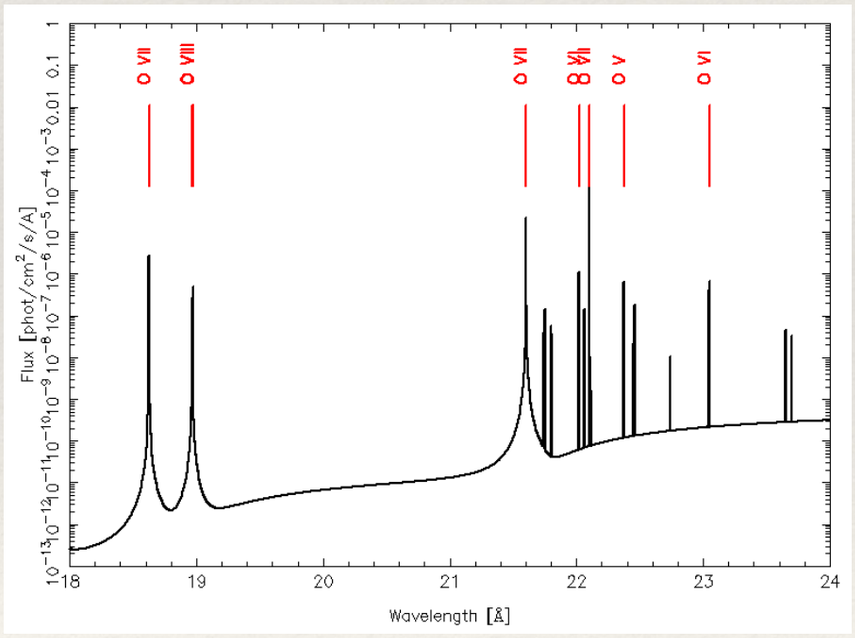
http://space.mit.edu/cxc/analysis/xstardb/

Navigating the *new* XSTAR model databases

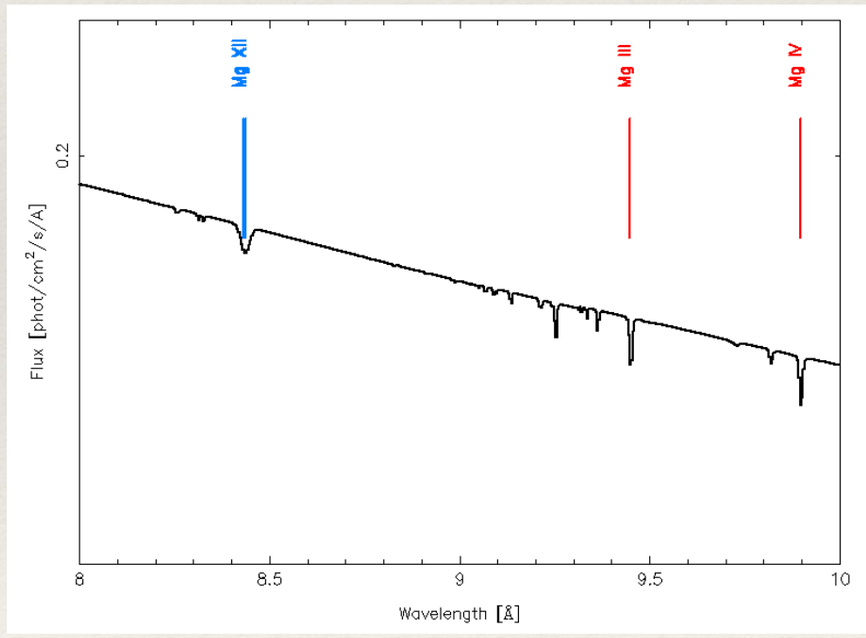
Please see

http://space.mit.edu/cxc/analysis/xstardb/examples.html

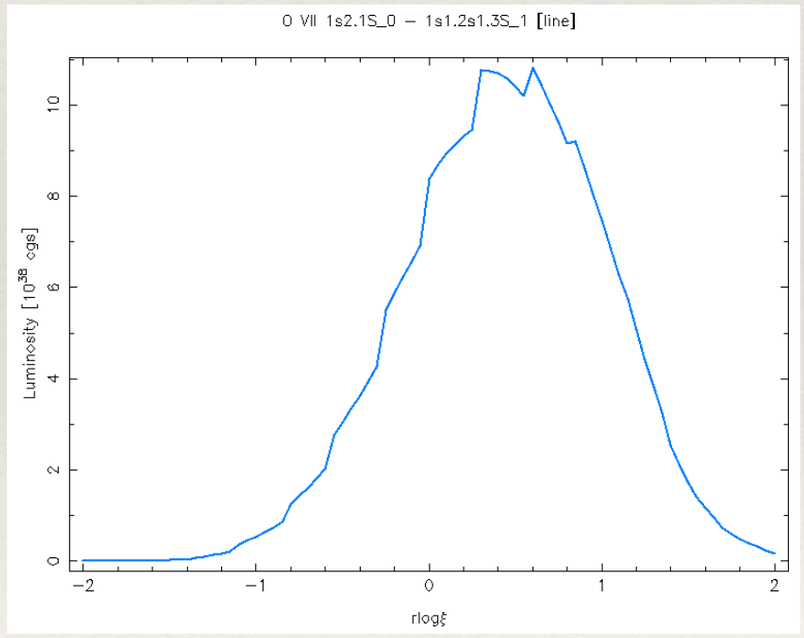
for three essential examples



single photemis model



multiple warmabs models



grid of photemis models
(for line luminosity vs rlogxi)