PyNeb_manual_7b

June 2, 2020

[1]:
```
import pyneb as pn
import numpy as np
```

To determine the ionic abundance, one can use the Atom object:

[2]:
```
O3 = pn.Atom('O',3)
Opp_abund = O3.getIonAbundance(int_ratio=3239.4, tem=1.5e4, den=110.,␣
 ↪to_eval='L(5007)+L(4959)', Hbeta=100.0)
print('O++/O = {:5.2e}'.format(Opp_abund))
```

```
O++/O = 2.59e-04
```

One can also use the observations from the Observation object:

[3]:
```
obs = pn.Observation()
obs.readData('observations1.dat', fileFormat='lines_in_rows', err_default=0.05)␣
 ↪# fill obs with data read from observations1.dat
obs.def_EBV(label1="H1r_6563A", label2="H1r_4861A", r_theo=2.85)
obs.correctData(normWave=4861.)
```

[4]:
```
obs.printIntens()
```

```
S4_10.5m      4.076
Ne2_12.8m     4.826
Ne3_15.6m    19.803
S3_18.7m      5.802
O2_3726A     46.576
O2_3729A     21.812
Ne3_3869A    21.722
Ne3_3968A     7.255
S2_4069A      0.950
S2_4076A      0.503
O3_4363A      4.687
H1r_4861A   100.000
O3_5007A    425.599
N2_5755A      0.454
S3_6312A      0.641
O1_6300A      1.428
O1_6364A      0.454
```

```
N2_6548A      5.657
H1r_6563A   285.000
N2_6584A     15.668
S2_6716A      0.995
S2_6731A      1.777
Ar3_7136A     3.882
O2_7319A+     5.106
O2_7330A+     4.034
```

[5]:
```
all_atoms = pn.getAtomDict(atom_list=obs.getUniqueAtoms())
line_ab = {}
ion_ab = {}
temp = 12000.
dens = 1e4
for line in obs.getSortedLines():
    if line.atom != 'H1' and line.atom != 'He1' and line.atom != 'He2':
        line_ab[line.label] = all_atoms[line.atom].getIonAbundance(line.
 ↪corrIntens, temp, dens,
                                                                   to_eval=line.to_eval)
        if line.atom not in ion_ab:
            ion_ab[line.atom] = []
        ion_ab[line.atom].append(line_ab[line.label][0])
for line in sorted(line_ab):
    print('{:10} {:.2f}'.format(line, 12+np.log10(line_ab[line][0])))
```

```
warng _ManageAtomicData: rec data not available for Ar3
warng _ManageAtomicData: atom data not available for H1
warng _ManageAtomicData: coll data not available for H1
warng _ManageAtomicData: rec data not available for Ne3
warng _ManageAtomicData: rec data not available for S2
warng _ManageAtomicData: rec data not available for S3
warng _ManageAtomicData: rec data not available for S4
Ar3_7136A  5.33
H1r_4861A  12.00
H1r_6563A  12.01
N2_5755A   6.36
N2_6548A   6.38
N2_6584A   6.35
Ne2_12.8m  6.77
Ne3_15.6m  7.11
Ne3_3869A  7.07
Ne3_3968A  7.12
O1_6300A   6.16
O1_6364A   6.16
O2_3726A   7.47
O2_3729A   7.48
O2_7319A+  7.32
O2_7330A+  7.29
```

```
O3_4363A   7.89
O3_5007A   7.92
S2_4069A   5.07
S2_4076A   5.29
S2_6716A   5.18
S2_6731A   5.12
S3_18.7m   5.93
S3_6312A   5.82
S4_10.5m   5.17
```

[6]:
```
for ion in sorted(ion_ab):
    print(ion, ion_ab[ion])
```

```
Ar3 [2.1378200554048055e-07]
H1r [1.0, 1.012426284751474]
N2 [2.2736199273983762e-06, 2.3964003294880454e-06, 2.2557784739201715e-06]
Ne2 [5.863763803374603e-06]
Ne3 [1.2834108902936717e-05, 1.1789947985274417e-05, 1.3072908397247668e-05]
O1 [1.448564230567394e-06, 1.4410898953658049e-06]
O2 [2.927851246950385e-05, 3.0189123334478235e-05, 2.0928629352567606e-05,
1.9320429182543967e-05]
O3 [7.76936539188984e-05, 8.284772814263309e-05]
S2 [1.1875081985725224e-07, 1.944136081004211e-07, 1.5018322372094437e-07,
1.3220539652396933e-07]
S3 [8.538520510209691e-07, 6.608571320910761e-07]
S4 [1.4662176292645887e-07]
```

[7]:
```
for atom in ion_ab:
    mean = np.mean(np.asarray(ion_ab[atom]))
    ion_ab[atom] = mean
    print('{:4s}: {:4.2f}'.format(atom, 12+np.log10(mean)))
```

```
Ar3 : 5.33
H1r : 12.00
N2  : 6.36
Ne2 : 6.77
Ne3 : 7.10
O1  : 6.16
O2  : 7.40
O3  : 7.90
S2  : 5.17
S3  : 5.88
S4  : 5.17
```