```
#############################################
#
# Correction of emission lines for MW dust extinction and intrinsic
# dust extinction
# Then, SFR estimate
# from [OII], H-beta and H-alpha
#
#############################################
#
# Reading fluxes and correcting for the MW dust extinction
#
g1,o2a,o2b,hb,ha = loadtxt('Gal_emi.txt',unpack=True,usecols=(0,1,2,3,4))
g2,av_mw = loadtxt('sample_dust_mw.txt',unpack=True,usecols=(0,2))
z,ko2_mw,khb_mw,kha_mw=loadtxt('Gal_ext.txt',unpack=True,usecols=(0,1,2,3))
#
o2 = o2a+o2b                    # Total [OII] flux
#
rv = 3.1                        # constant A_V = E(B-V)*R_V
ao2_mw = ko2_mw*av_mw/rv
ahb_mw = khb_mw*av_mw/rv
aha_mw = kha_mw*av_mw/rv
#
o2c = o2*10**(0.4*ao2_mw)       # o2 MW dust correction
hbc = hb*10**(0.4*ahb_mw)       # hb MW dust correction
hac = ha*10**(0.4*aha_mw)       # ha MW dust correction
#############################################
#
# CALCULATE INTRINSIC DUST EXTINCTION
#
#
ko2 = 4.56127                          # Extinction at [OII]
khb = 3.63035                          # Extinction at H-beta
kha = 2.46790                          # Extinction at H-alpha
#
av=[]
o2=[]
hb=[]
ha=[]
#
for i in range(0,len(hac)):
    if hac[i] > 0 and hbc[i]>0:
        avi = 2.5*rv/(kha-khb)*log10(2.86/(hac[i]/hbc[i]))
        av.append(avi)
        o2i = o2c[i]*10**(0.4*ko2*avi/rv)          # Corrected [OII]
        hbi = hbc[i]*10**(0.4*khb*avi/rv)          # Corrected H-beta
        hai = hac[i]*10**(0.4*kha*avi/rv)          # Corrected H-alpha
        o2.append(o2i)
        hb.append(hbi)
        ha.append(hai)
    else:
        avi = 1.48          # Mean A_V calculated from sample with H-a & H-b
        av.append(avi)
        o2i = o2c[i]*10**(0.4*ko2*avi/rv)          # Corrected [OII]
        hbi = hbc[i]*10**(0.4*khb*avi/rv)          # Corrected H-beta
        hai = hac[i]*10**(0.4*kha*avi/rv)          # Corrected H-alpha
        o2.append(o2i)
        hb.append(hbi)
        ha.append(hai)
av_mean = np.sum(av)/np.size(av)
print ('Average intrinsic extinction A_V = %f' % av_mean)
```

```
##############################################
# To change from list to array
#
o2 = array(o2)
hb = array(hb)
ha = array(ha)
###############################################
#
# Finds best linear correlation [OII] vs. H-alpha (remove null values)
# Fluxes are in units of 1e-17 erg/s/cm^2
#
lo2=[]
lha=[]
for i in range(0,len(ha)):
    if ha[i]>0 and o2[i] > 0:
        lo2.append(log10(o2[i]))
        lha.append(log10(ha[i]))
#
lo2 = array(lo2)
lha = array(lha)
#
# Polynomial linear fit between log([OII]) and log(H-alpha) and plot
# Then comparison with the x = y line
#
plot (lo2,lha,'ro')
x=arange(1.5,4,.1)
plot (x,x)                                          # line with x = y
coeffs = numpy.polyfit(lo2,lha,1)
plot(x,coeffs[0]*x+coeffs[1])
xlabel('log ([OII]) [x10^(-17) erg/s/cm^2]')
ylabel('log (H-alpha) [x10^(-17) erg/s/cm^2]')
corr = numpy.corrcoef(lo2,lha)[0,1]
print ("Slope:\t\t %2.5f" % coeffs[0])          # Print slope
print ("Intercept:\t %2.5f\n" % coeffs[1])      # PRint intercept
print ('Correlation log [OII] vs. log H-alpha = %f' % corr)
################################################
#
# SFR CALCULATION FROM H-alpha, H-beta AND [OII]
# Need first to run 'cosmology.py' to get Luminosity Distance of galaxies
# This is in Ld.txt, in cm, 1 Mpc = 3.085677581e+24 cm
#
Ld=loadtxt('Ld.txt',unpack=True)
#
Lo2 = 4*pi*Ld**2*o2/1e17        # Luminosity [OII]
Lha = 4*pi*Ld**2*ha/1e17        # Luminosity H-beta
Lhb = 4*pi*Ld**2*hb/1e17        # Luminosity H-alpha
#
#
cha = 4.39e-42                  # constant for Ha
co2 = cha                       # constant [OII] = constant H-alpha
chb = 12.6e-42                  # constant for H-beta
#
# SFR estimate from [OII], H-beta & H-alpha lines
#
sfr_o2 = co2*Lo2
sfr_hb = chb*Lhb
sfr_ha = cha*Lha
#
print ('SFR_[OII] (M_sun/yr) =', sfr_o2)        # print SFR from [OII]
print ('SFR_H-alpha (M_sun/yr) =', sfr_ha)      # print SFR from H-alpha
```

```
##############################################
#
# Plot SFRs in new display
#
plt.figure("SFR [OII] vs H-alpha")
plt.plot (log10(sfr_o2),log10(sfr_ha), 'b^')
xlabel('log (SFR_[OII]) [M_sun/yr]')
ylabel('log (SFR_H-alpha [M_sun/yr])')
x=(-1,1.5)
plot (x,x)
```