```
#!/usr/bin/python2

# numerics
import numpy as np

# imaging
from vigra.impex import readImage, writeImage
from vigra import RGBImage

import matplotlib
matplotlib.use("Qt4Agg")
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import pyplot as plot
from colorsys import rgb_to_hsv, hsv_to_rgb, rgb_to_yiq


imfiles = ['f16.tiff', 'lena.tiff', 'wildflower.png', 'mandrill.tiff', ]

vrgb2hsv = np.vectorize(rgb_to_hsv)
vhsv2rgb = np.vectorize(hsv_to_rgb)
vrgb2yiq = np.vectorize(rgb_to_yiq)


def subsample(img, n):
    return img[0::n, 0::n, :]

def showimg(img):
    # use RGB type
    img = img * 256
    img = img.astype('uint8')

    # correct axes
    img = img.swapaxes(0,1)

    plot.imshow(img)


def show3Dscatter(m,n,k,*args, **kwargs):
    ax = plot.subplot(m,n,k, projection='3d')
    ax.set_title(kwargs.pop('title', 'No Title'))
    ax.set_xlabel(kwargs.pop('xlabel', 'X'))
    ax.set_ylabel(kwargs.pop('ylabel', 'Y'))
    ax.set_zlabel(kwargs.pop('zlabel', 'Z'))
    ax.scatter(*args, **kwargs)


if __name__ == "__main__":

    sample = 16

    for name in imfiles:

        #### scatter plot ####
        plot.figure()

        # read the image from file, subsample and scale to [0.0,1.0]
        img = subsample(readImage(name),8)/256

        # get single channels, shape isn't interesting
        r = img[:,:,0]
        g = img[:,:,1]
        b = img[:,:,2]

        h, s, v = vrgb2hsv(r,g,b)

        r.reshape(r.size)
        g.reshape(g.size)
        b.reshape(b.size)

        h_im = h
        s_im = s
        v_im = v
        h.reshape(h_im.size)
        s.reshape(s_im.size)
        v.reshape(v_im.size)

        # convert RGB to HSV
        h, s, v = vrgb2hsv(r,g,b)

        # create rgba tuples for plotting, 1 == opaque
        c = [(x,y,z,1) for x,y,z in zip(r.flat,g.flat,b.flat)]

        # cylindrical coordinates
        # hue is angle, saturation is radius, value is length
        h_ang = 2*np.pi*h
        x = np.cos(h_ang)*s
        y = np.sin(h_ang)*s
        z = v

        show3Dscatter(1,2,1, r, g, b, c=c, edgecolor='None', xlabel='Red', ylabel='Green',
        zlabel='Blue', title=name+" (RGB)")
        show3Dscatter(1,2,2, x,y,z, c=c, edgecolor='None', xlabel='Hue/Saturation',
        ylabel='Hue/Saturation', zlabel='Value', title=name + " (HSV)")


        #### single image plots ####

        imgs = [img[:,:,0],img[:,:,1],img[:,:,2], s_im, v_im]
        titles = ['red', 'green', 'blue', 'saturation', 'value', 'hue']
        plot.figure()
        for i in range(len(imgs)):
            ax = plot.subplot(2,3,i+1)
            showimg(imgs[i])
            ax.set_title(titles[i])
            plot.gray()

        ax = plot.subplot(2,3,6)
        ax.set_title(titles[5])
        im = np.ndarray((img.shape[0], img.shape[1], 3))
        im[:,:,0],im[:,:,1],im[:,:,2] = vhsv2rgb(h_im, np.ones(imgs[0].shape)*1,
        np.ones(imgs[0].shape)*.7)

        showimg(im)
        break


    plot.show()
```