

Работа с цифрови астрономически изображение с помощта на Python и astropy

Целта на упражнението е да можем да отваряме и променяме астрономически изображения с помощта на езика Python. Използваме jupyter notebook за интерактивна работа

- В част едно ще отворим един кадър. Данните ще са запазени в променливите data, header. Можем да променяме ключови думи, да правим аритметика или директно да променяме пискелни координати.
- В втората част ще се опитаме да генерираме изкуствено изображение, използвайки случайни числа за позициите и стойностите на пикселите на звездите. Използваме Гаусова функция за формата на звездата, добавяме и произволен сигнал за фон.
- В част три ще търсим координатите на звездите от оригинално изображение, чрез проста функция find_stars.

Част 1

```
In [52]: from __future__ import division#only python2
import os
from astropy.io import fits
import numpy as np

moiat_file = 'new-image.fits'
data, header = fits.getdata(moiat_file, header=True, update=True)
header['DATE-OBS'] = '2018-08-06T20:33:10'

type(moiat_file)
type(header)
```

```
Out[52]: astropy.io.fits.header.Header
```

Част 2

```
In [36]: ## test
data = np.zeros((400,400))
print data
hdu = fits.PrimaryHDU(data=data)
hdu.writeto('generatedbyus.fits',overwrite=True)
moiat_file = 'generatedbyus.fits'
new_data, header = fits.getdata(moiat_file,header=True, update=True)
print(new_data)

[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
```

Можем да варираме параметрита за да създадем различни изображения. Всеки път резултатът ще презаписва ново изображение. По кои белези можем да забележим че изображението е симулирано?

```
In [49]: #Vary those

#create our own image, x_len by y_len pixels
x_len = 1000
y_len = 1000
number_sim_stars = 100
#pixels, seeing measure, play with those
r = 1.1
#actual star spread
r_star = 25

#create initial background
data = np.zeros((x_len, y_len), dtype=int)

for i in range(x_len):
    for j in range(y_len):
        data[i,j] = np.random.randint(2000,2100)#create background

for n in range(number_sim_stars):
    x0,y0 = np.random.randint(r_star,x_len-r_star),np.random.randint(r_star,y_len-r_star)#do not go outside image
    signal = np.random.randint(10000,50000)#create central signal peak7
    r_scaled = r*signal/50000#scale star by its intensity
    for i in range(x0-r_star, x0+r_star+1):
        for y in range(y0-r_star, y0+r_star+1):
            current_r = ((i-x0)**2 + (y-y0)**2)**0.5
            if current_r <= r_star:
                #some kind of gaussian shape
                data[i,y] += signal*np.exp(-(current_r)**2/(2*r_scaled**2))

hdu = fits.PrimaryHDU(data=data)
hdu.writeto('my-just-generated.fits', overwrite=True)
#os.remove('my-just-generated.fits')
```

Част 3 - търсене на звезди в полето

Функцията `find_stars` търси сигнал над даден праг (`threshold`). Подаваме му масив `img` от вече прочетено изображение.

Какви пропуски можем да имаме? Защо намираме по-малко звезди отколкото имаме?

```
In [51]: def find_stars(img, threshold, star_rad = 25):
    found_stars = []#create a list for the new-found positions
    brightestpixel=img.max() # find the brightest pixel on the image

    while brightestpixel > threshold:#set a mininum value of the brightest point

        starindex=np.where(img==brightestpixel)#find the indices of the brigthest pixel

        x_position=starindex[1]#turn these into coordinates of the point
        y_position=starindex[0]

        found_stars.append((x_position[0], y_position[0]))#add coordinates to the array

        delta_r = star_rad
        xmin = starindex[0][0]-delta_r#use parameters to find the indices of the square to be removed
        ymin = starindex[1][0]-delta_r
        xmax = starindex[0][0]+delta_r
        ymax = starindex[1][0]+delta_r

        img[xmin:xmax,ymin:ymax]=threshold-1# substitute those values with zeros

        brightestpixel=img.max()
    return np.array(found_stars)

moiat_file = 'my-just-generated.fits'
data, header = fits.getdata(moiat_file,header=True)

star_coords = find_stars(data,2100)
print len(star_coords)
```

92