Analisi dati con Python

Extreme Energy Events
Science inside Schools

LICEO SCIENTIFICO STATALE "Benedetto Croce" di PALERMO

CENTRO FERMI
Museo Storico della Fisica e
Centro Studi e Ricerche Enrico Fermi

Relatori

Mario Lauriano

Giuseppe Bruno

Perché Python?

- **Open Source**
- **Librerie ottimizzate**
- **Analisi dati più efficace (grande mole e grafici più precisi)**


```
Python 3.6.5 (default, Apr  1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> data=np.genfromtxt('TORI-04from2018-04-05to2018-04-06.csv',delimiter=',')
>>> data
array([[ 1.00000000e+00,  3.55270176e+08,  1.82829530e+07, ...,
         1.05321770e+02, -1.00000000e-01,  9.82000000e+02],
       [ 1.00000000e+00,  3.55270176e+08,  2.55473880e+07, ...,
         1.00918465e+02,  7.26400000e-03,  9.82000000e+02],
       [ 1.00000000e+00,  3.55270176e+08,  2.84015920e+07, ...,
         1.21021515e+02,  2.85400000e-03,  9.82000000e+02],
       ...,
       [ 2.70000000e+01,  3.55293331e+08,  1.54089992e+08, ...,
         1.07391197e+02,  3.03000000e-03,  9.82000000e+02],
       [ 2.70000000e+01,  3.55293331e+08,  2.45941266e+08, ...,
         1.01238701e+02,  9.18510000e-02,  9.82000000e+02],
       [ 2.70000000e+01,  3.55293331e+08,  2.52633024e+08, ...,
         1.04087105e+02,  6.69200000e-03,  9.82000000e+02]])
>>>
```

Librerie

- Installazione delle librerie da prompt comandi: pip install nomelibreria (vedi slide seguente.)

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IP[y]:
IPython

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

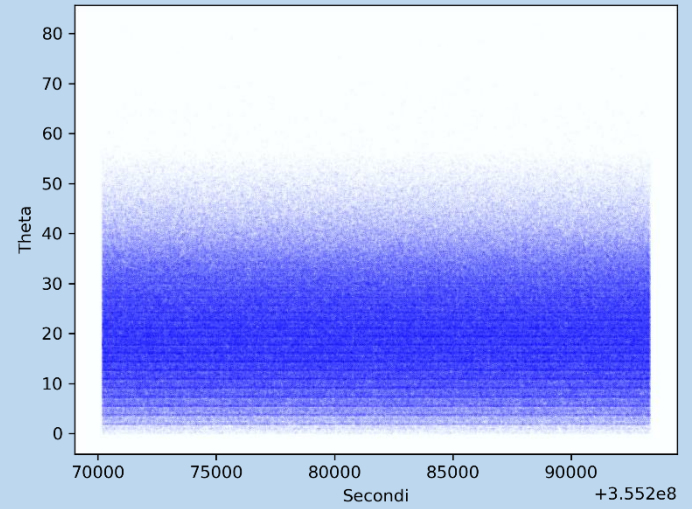
pandas
Data structures & analysis

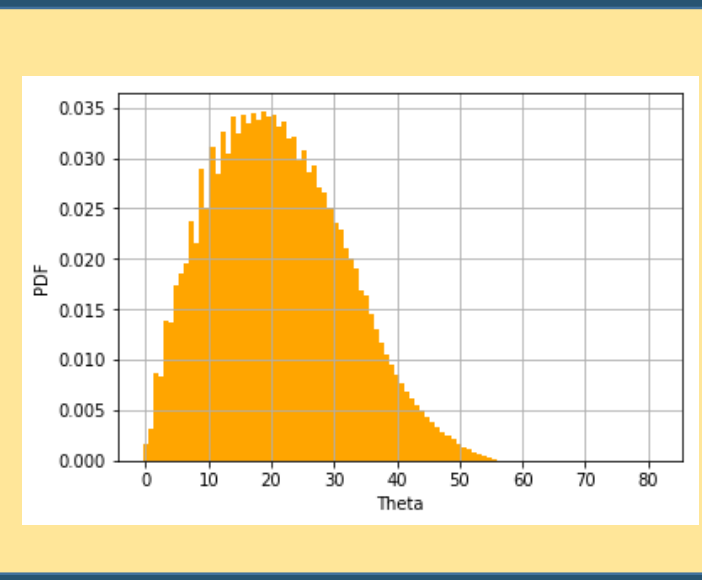
- Le librerie contengono codice compilato scritto in c++. Alcune librerie richiedono il compilatore installato sulla macchina al fine di ottimizzare le prestazioni di calcolo.

Distribuzione angolare

Angolo Theta

```
#Scatter
import numpy as np
import matplotlib.pyplot as plt
data=np.genfromtxt("TORI-04from2018-04-05to2018-04-06.csv",delimiter=",")
x=(data[1:,1]+data[1:,2]*10**(-9)) #secondi + nanosecondi
y=data[1:,3] #Theta
plt.scatter(x,y,0.0005)
plt.xlabel( "Secondi" )
plt.ylabel( "Theta" )
plt.show()
#Istogramma
Nc=100 #numero di intervalli
pdf, bins = np.histogram( y, Nc, density = True )
nextr = len( bins )
dx = bins[1] - bins[0]
plt.xlabel( "Theta" )
plt.ylabel( "PDF" )
plt.grid()
plt.bar( bins[:nextr-1], pdf, dx, color = "orange" )
plt.show()
```





Tempo di volo

```
import scipy.optimize as syt
import numpy as np
def F( x, mu, sigma ):
    ff = 1/(np.sqrt( 2.0 * np.pi )*sigma )*np.exp(-(x-mu)**2.0)/(2.0*sigma**2))
    return ff
imin = 0
imax =999
Nc=1000
pdf, bins = np.histogram( y, Nc, density = True )
nextr = len( bins )
xc = 0.5 * ( bins[0:nextr-1] + bins[1:nextr] )
xdata = xc[imin:imax]
ydata = pdf[imin:imax]
par, corr = syt.curve_fit( F, xdata, ydata )
nextr = len( bins )
dx = bins[1] - bins[0]
plt.xlabel( r"$x$" )
plt.ylabel( "PdF" )
plt.bar( bins[:nextr-1], pdf, dx, color = "g" )
yc = (F( xc, par[0], par[1]))
plt.plot( xc, yc, "r-o" )
plt.xlim(0,8)
plt.title('$\mu=$'+str(par[0])+', $\sigma=$'+str(par[1]), fontsize=10)
plt.show()
```

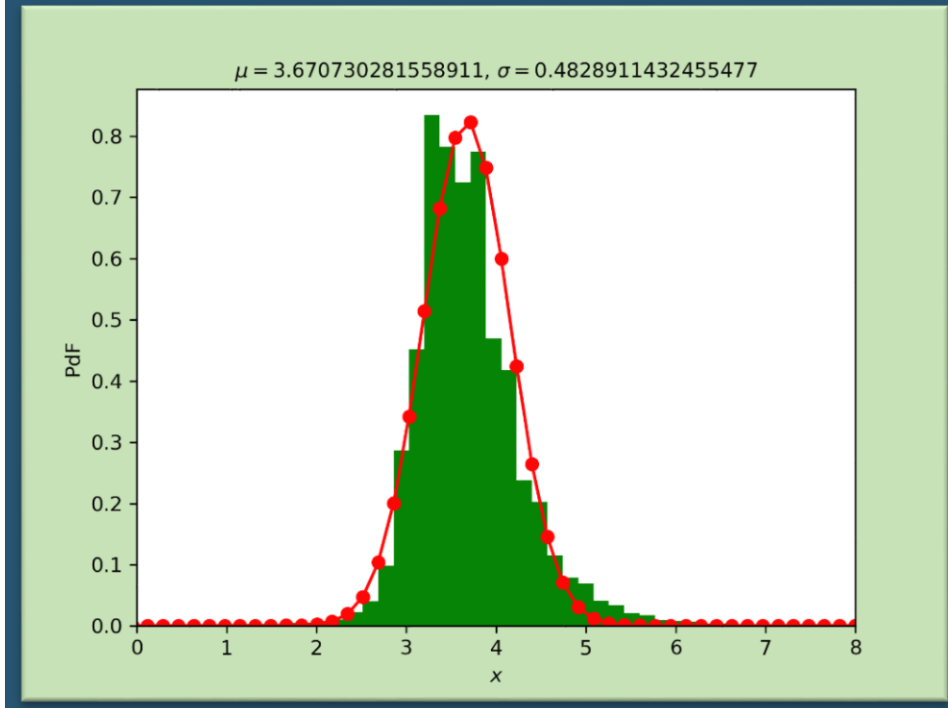
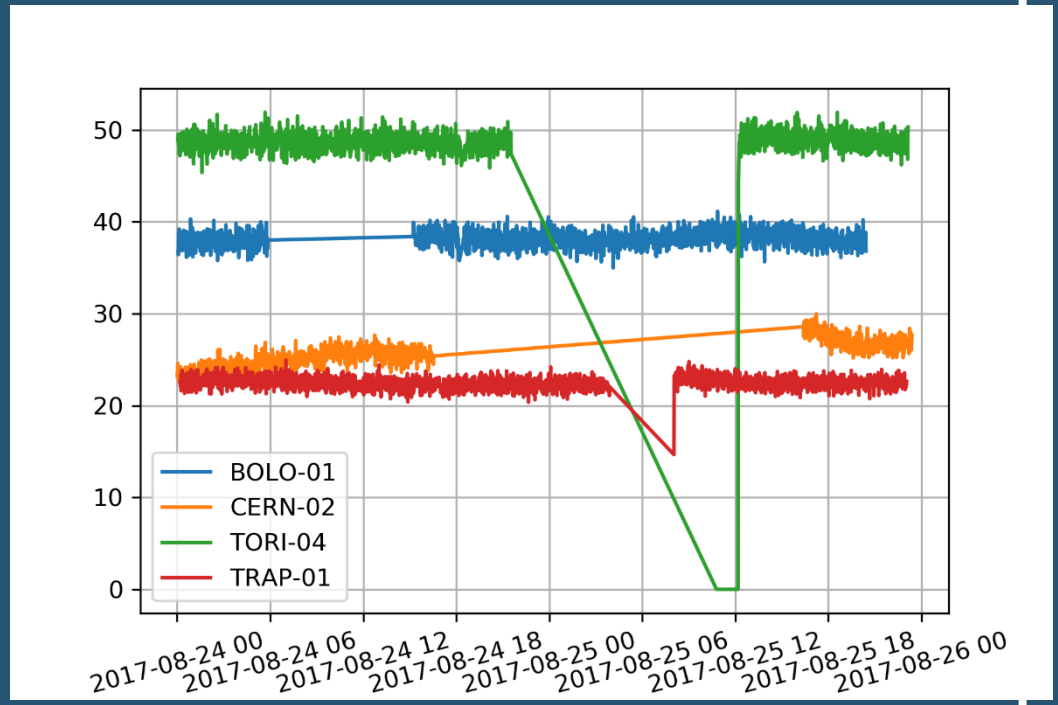


Grafico dei dati con curva di fit gaussiano

Confronti tra osservatori

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as md
import datetime as dt
data_b1=np.genfromtxt("BOLO-01.csv",delimiter=",")
data_c2=np.genfromtxt("CERN-02.csv",delimiter=",")
data_t4=np.genfromtxt("TORI-04.csv",delimiter=",")
data_t1=np.genfromtxt("TRAP-01.csv",delimiter=",")
t0 = 1167609600 #la differenza di secondi tra 1/1/1970 e 1/1/2007
c = 0
data_b1= data_b1[data_b1[:,c].argsort()]
nxi_b1=data_b1.shape[0]
data_b1 = np.unique(data_b1,axis=0)
nx_b1=data_b1.shape[0]
timestamps_b1 = data_b1[:,0] + t0
xts_b1 = md.date2num([dt.datetime.fromtimestamp(s) for s in timestamps_b1])
y_b1 = data_b1[:,4] # estraggo la colonna dei conteggi/minuto RateTrackEvent
[…]
plt.xticks( rotation=15 )
ax=plt.gca()
#xfmt = md.DateFormatter('%Y-%m-%d %H:%M:%S')
xfmt = md.DateFormatter('%Y-%m-%d %H')
ax.xaxis.set_major_formatter(xfmt)
p_b1, =plt.plot(xts_b1,y_b1)
p_c2, =plt.plot(xts_c2,y_c2
p_t4, =plt.plot(xts_t4,y_t4)
p_t1, =plt.plot(xts_t1,y_t1)
plt.legend([p_b1, p_c2, p_t4,p_t1],["BOLO-01","CERN-02","TORI-04","TRAP-01"])
plt.grid()
#plt.savefig('Multiple.png',dpi=300)
plt.show()
```



Download multiplo

```python
import urllib.request
#gmin: primo giorno da scaricare
#gmax: ultimo giorno da scaricare
#tel: nome strumento
#Es. download(1,30,"CERN-02")

def download(gmin,gmax,tel):
    f=open("lista", "a+")
    for i in range(gmin+1,gmax+1):
        if i<10:
            a1="0"+str(i-1)
            a2="0"+str(i)
        elif i==10:
            a1="0"+str(i-1)
            a2=str(i)
        else:
            a1=str(i-1)
            a2=str(i)
        name=tel+'_2017-08-'+a1+'_2017-08-'+a2+'_summary_Trending.csv'
        url = 'https://iatw.cnaf.infn.it/eee/monitor/dqmreport2/'+tel+'/2017-08-'+a2+'/'+name
        urllib.request.urlretrieve(url,name)
        print("Scaricato:"+name)
        f.write(name+ '\n')
    f.close()
```

Cartella contenente 60 files giornalieri

```
21:49           443.873 CERN-02_2017-08-17_2017-08-18_summary_Trending.csv
21:49           449.249 CERN-02_2017-08-18_2017-08-19_summary_Trending.csv
21:49           449.753 CERN-02_2017-08-19_2017-08-20_summary_Trending.csv
21:49           439.505 CERN-02_2017-08-20_2017-08-21_summary_Trending.csv
21:49           442.361 CERN-02_2017-08-21_2017-08-22_summary_Trending.csv
21:49           436.985 CERN-02_2017-08-22_2017-08-23_summary_Trending.csv
21:49           380.537 CERN-02_2017-08-23_2017-08-24_summary_Trending.csv
21:49           224.633 CERN-02_2017-08-24_2017-08-25_summary_Trending.csv
21:49           286.457 CERN-02_2017-08-25_2017-08-26_summary_Trending.csv
21:49           444.377 CERN-02_2017-08-26_2017-08-27_summary_Trending.csv
21:49           447.905 CERN-02_2017-08-27_2017-08-28_summary_Trending.csv
21:49           448.577 CERN-02_2017-08-28_2017-08-29_summary_Trending.csv
20:46    <DIR>          Codice
22:14           320.393 TORI-04_2017-08-01_2017-08-02_summary_Trending.csv
22:14           403.217 TORI-04_2017-08-02_2017-08-03_summary_Trending.csv
22:14           439.001 TORI-04_2017-08-03_2017-08-04_summary_Trending.csv
22:14           452.105 TORI-04_2017-08-04_2017-08-05_summary_Trending.csv
22:14           451.937 TORI-04_2017-08-05_2017-08-06_summary_Trending.csv
22:14           433.289 TORI-04_2017-08-06_2017-08-07_summary_Trending.csv
22:14           423.041 TORI-04_2017-08-07_2017-08-08_summary_Trending.csv
22:14           439.841 TORI-04_2017-08-08_2017-08-09_summary_Trending.csv
```

Unire più file

```
import numpy as np
f=open('lista','r')
flst=f.readlines()
print('Nella lista ci sono '+str(len(flst))+' file')
if (len(flst)>1):
    print('Importo il file ',1,' ->',flst[0][0:-1])
    data=np.genfromtxt(flst[0][0:-1],delimiter=',')
    print('Le dimensioni della matrice dati:',data.shape)
    for i in np.arange(1,len(flst)-1,1):
        print('Importo il file ',i+1,' ->',flst[i][0:-1])
        data=np.vstack((data,np.genfromtxt(flst[i][0:-1],delimiter=',')))
        print('Le dimensioni della matrice dati è diventata:',data.shape)
else:
    print('Importo unico file ->',flst[0:-1])
    data=np.genfromtxt(flst[0:-1],delimiter=',')
    print('Le dimensioni della matrice dati:',data.shape)
print('Sto ordinando la matrice dati in funzione della prima colonna...')
c = 0
data= data[data[:,c].argsort()]
print('...e adesso elimino le righe duplicate...')
nxi=data.shape[0]
data = np.unique(data,axis=0)
nx=data.shape[0]
print('Ho eliminato complessivamente ',nxi-nx,' righe')
print('La nuova matrice dati ha le seguenti dimensioni',data.shape)
```

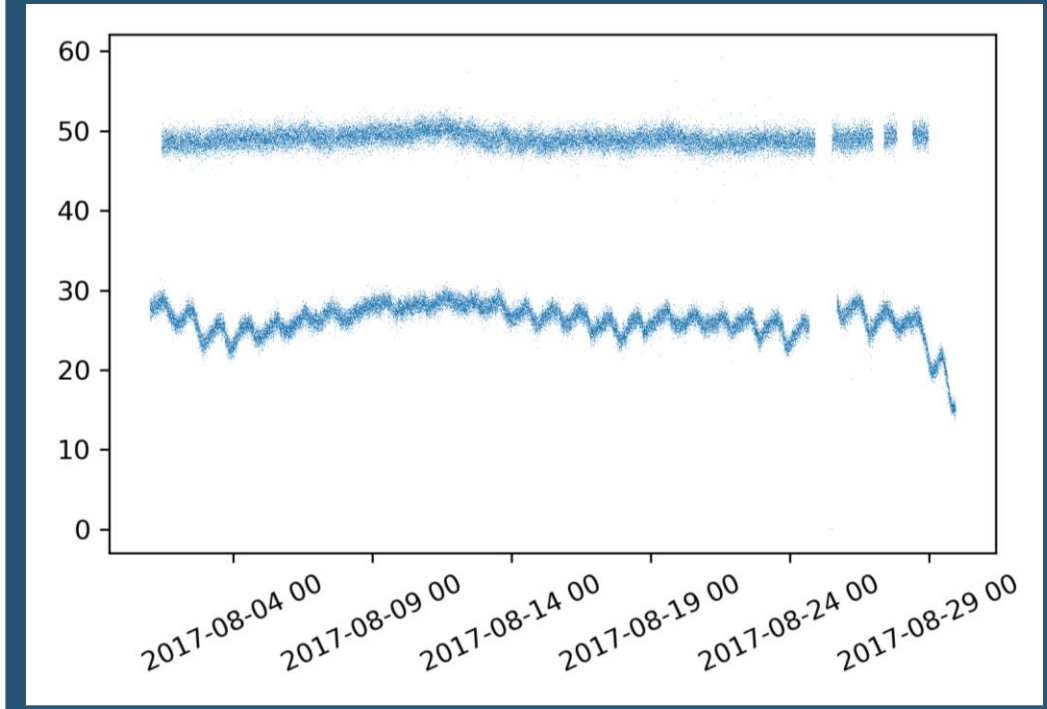
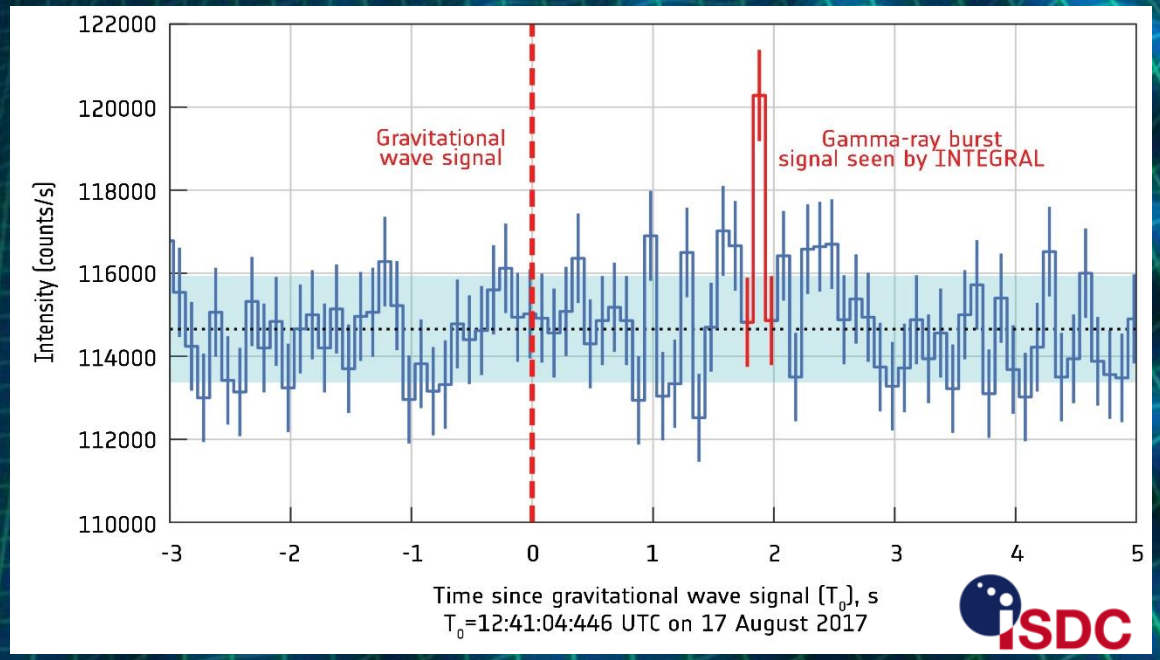


Grafico mensile su TORI-04 e CERN-02 (RateTrackEvent)

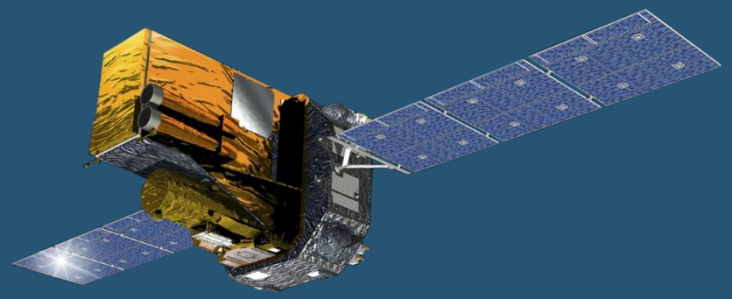
Correlazione tra flusso di raggi cosmici e fusione di due stelle a neutroni

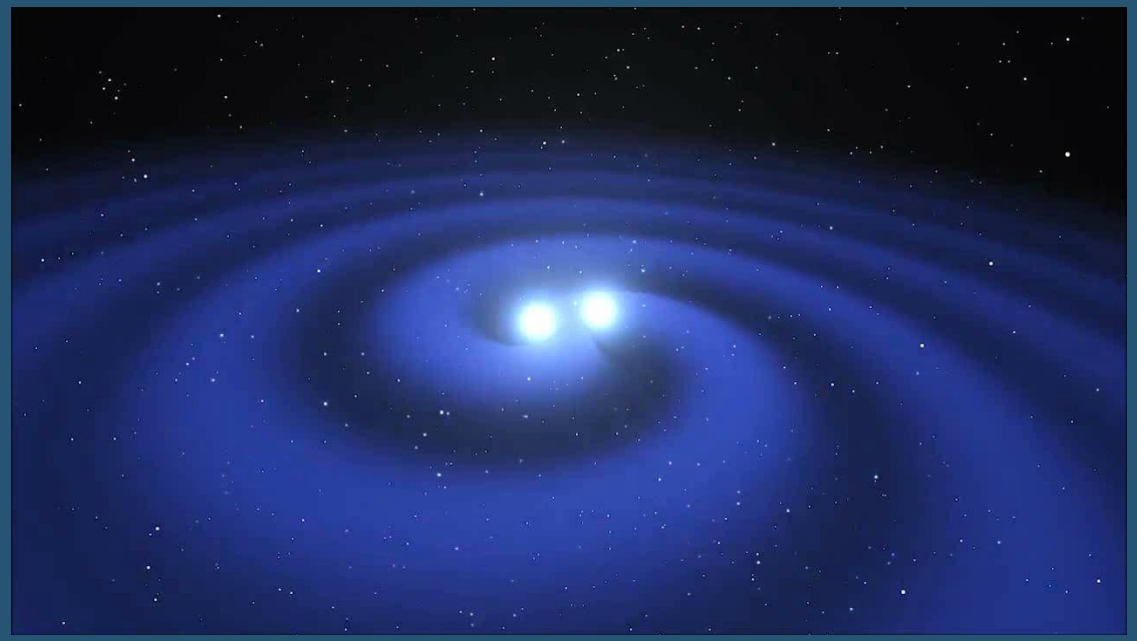
17 agosto 2017



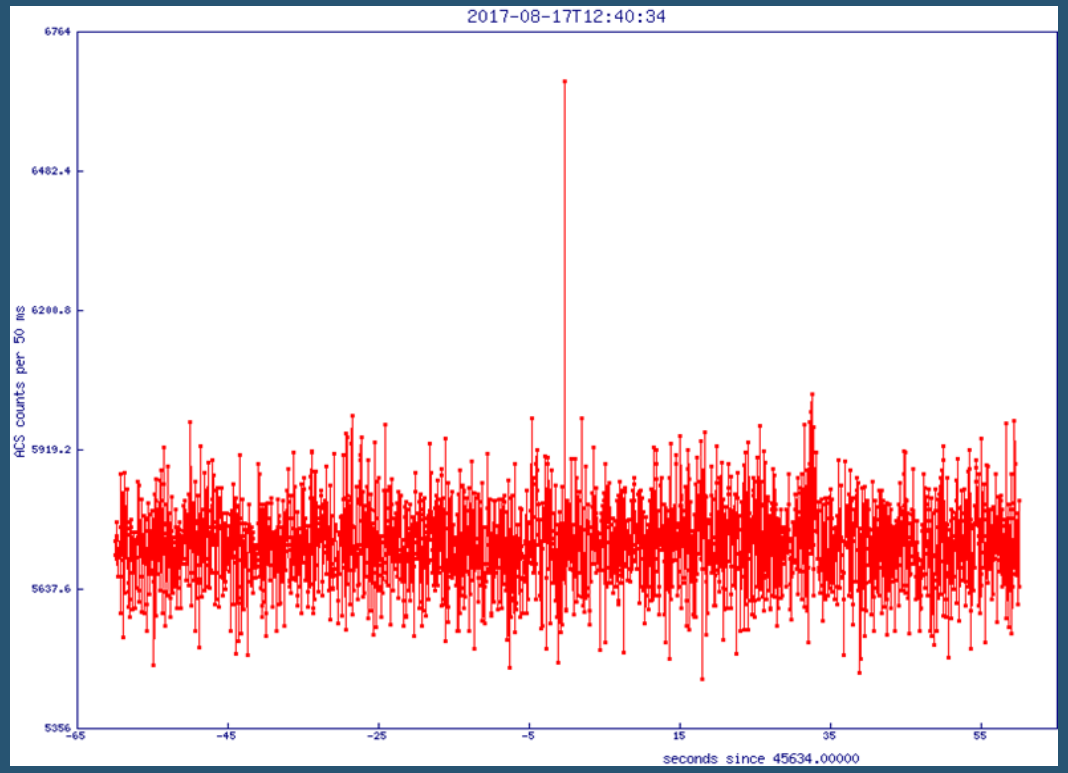
Mario Lauriano
Giuseppe Bruno
Nicolò Pizzo
Andrea Castagna
Matteo Alberto Catalano
Gianmarco Mattana
Prof. Vincenzo Di Matteo

INTEGRAL: **INTE**rnational **G**amma-**R**ay **A**strophysics **L**aboratory





Lampo gamma dopo le onde gravitazionali




```
import numpy as np
f=open('dataintegral.txt','r')
data_integral=np.genfromtxt(f,delimiter=" ")

import matplotlib.pyplot as plt
x=data_integral[:,0]
y=data_integral[:,1]
plt.plot(x,y)
plt.show()
```

Δt tra fotoni e protoni emessi simultaneamente dalla kilonova

$m = 10^{20}\ eV$ $\quad d = 1{,}3 \cdot 10^8\ a.l.$

$m_0 = 10^9$ eV

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} = 10^{11} \quad \Longrightarrow \quad v = c(1 - \gamma^{-2})^{\frac{1}{2}}$$

$$\Delta t = \frac{d}{v} - \frac{d}{c} = \frac{d}{c(1-\gamma^{-2})^{\frac{1}{2}}} = \frac{d}{c}\left[(1 - \gamma^{-2})^{-\frac{1}{2}} - 1\right] \approx \frac{d}{c}\left[1 - \left(-\frac{1}{2}\right)\gamma^{-2} - 1\right] = \frac{d}{c}\,\frac{1}{2}\gamma^{-2}$$

$$\frac{d}{c} = 1{,}3 \cdot 10^6\ anni = (1{,}3 \cdot 10^6) \cdot (3.1 \cdot 10^7 s) = 4{,}03 \cdot 10^{15}\ s$$

$$\Delta t = \left(\frac{4{,}03 \cdot 10^{15}}{2}\ \text{s}\right) \gamma^{-2} = (2.02 \cdot 10^{15}\ \text{s})\ \gamma^{-2}$$

γ^2	Δt (s)
10^{11}	$2{,}02 \cdot 10^7$
10^8	0,202
10^7	20,2
10^6	$2{,}02 \cdot 10^3$
10^5	$2{,}02 \cdot 10^5$

$$m_1 \in (1.36 - 2.26)M_\odot \rightarrow \overline{m_1} = \frac{1.36 + 2.26}{2} = 1.81M_\odot$$

$$m_2 \in (0.86 - 1.36)M_\odot \rightarrow \overline{m_2} = \frac{0.86 + 1.36}{2} = 1.11M_\odot$$

$$m_{tot} = 2.82^{+0.47}_{-0.09}M_\odot$$

$$\overline{m_1} + \overline{m_2} = 2.92M_\odot$$

$$E = (\overline{m_1} + \overline{m_2}) - m_{tot} = 2.92 - 2.82 = 0.1M_\odot$$

$$1M_\odot = 1.989 \times 10^{30}\ kg = 1.116 \times 10^{66}\ eV$$

$$E(eV) = 1.116 \times 10^{66} eV \cdot 0.1M_\odot$$

$$= 1.116 \times 10^{65} eV \quad \textit{Energia emessa nella coalescenza}$$

$$n_{p^+} = \frac{E}{m_{p^+}} = \frac{1.116 \times 10^{65} eV}{9.421 \times 10^{8} eV} = 0.118 \times 10^{57} \quad \textit{Numero di protoni emessi}$$

$$d = 130\ mln\ a.l. = 1.3 \times 10^{8}\ a.l.$$

$$1\ a.l. = 9.461 \times 10^{15} m$$

$$d(m) = 9.461 \times 10^{15} \cdot 1.3 \times 10^{8} = 12.229 \times 10^{23} = 1.229 \times 10^{24} m$$

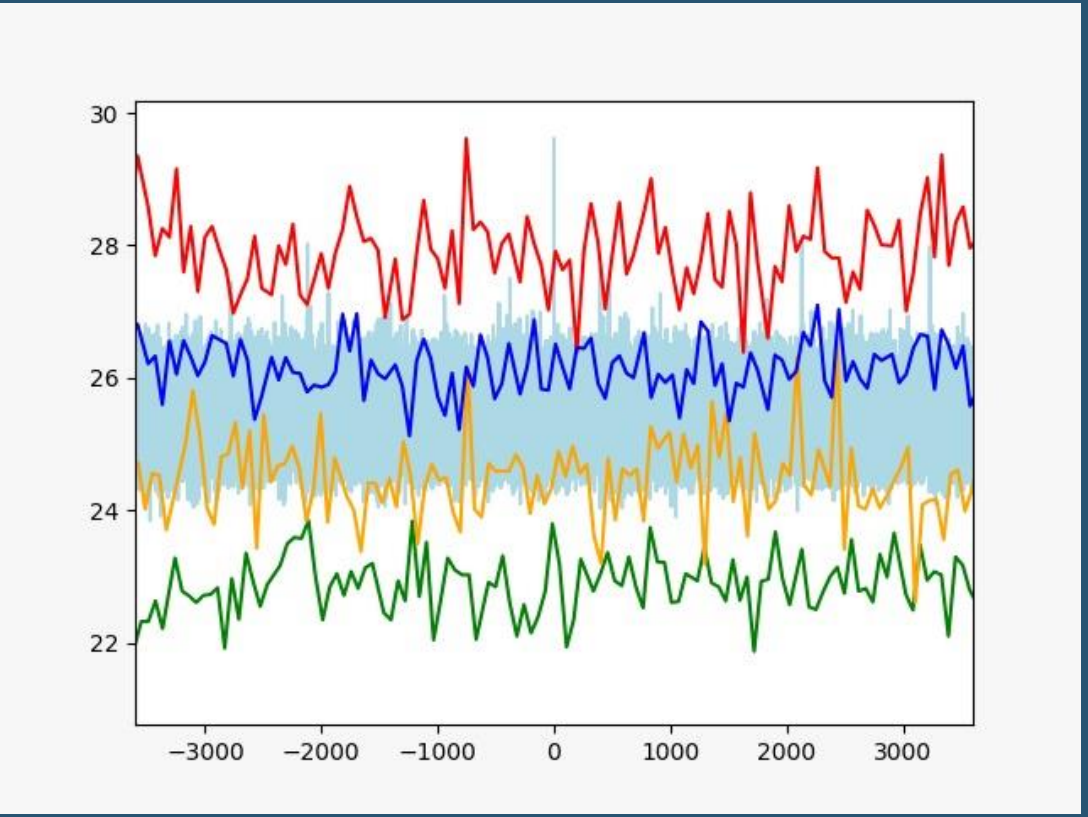
$$\frac{n_{p^+}}{m^2} = \frac{n_{p^+}}{4\pi d^2} = \frac{0.118 \times 10^{57}}{4\pi(1.229 \times 10^{24} m)^2} = \frac{0.118 \times 10^{57}}{1.899 \times 10^{49} m^2}$$

$$= 6.2 \times 10^{6} \frac{n_{p^+}}{m^2} \quad \textit{Numero di protoni per } m^2 \textit{ sulla terra}$$

http://iopscience.iop.org/article/10.3847/2041-8213/aa91c9/meta

Published 2017 October 16

The Astrophysical Journal Letters, Volume 848, Number 2



Confronto tra osservatori:

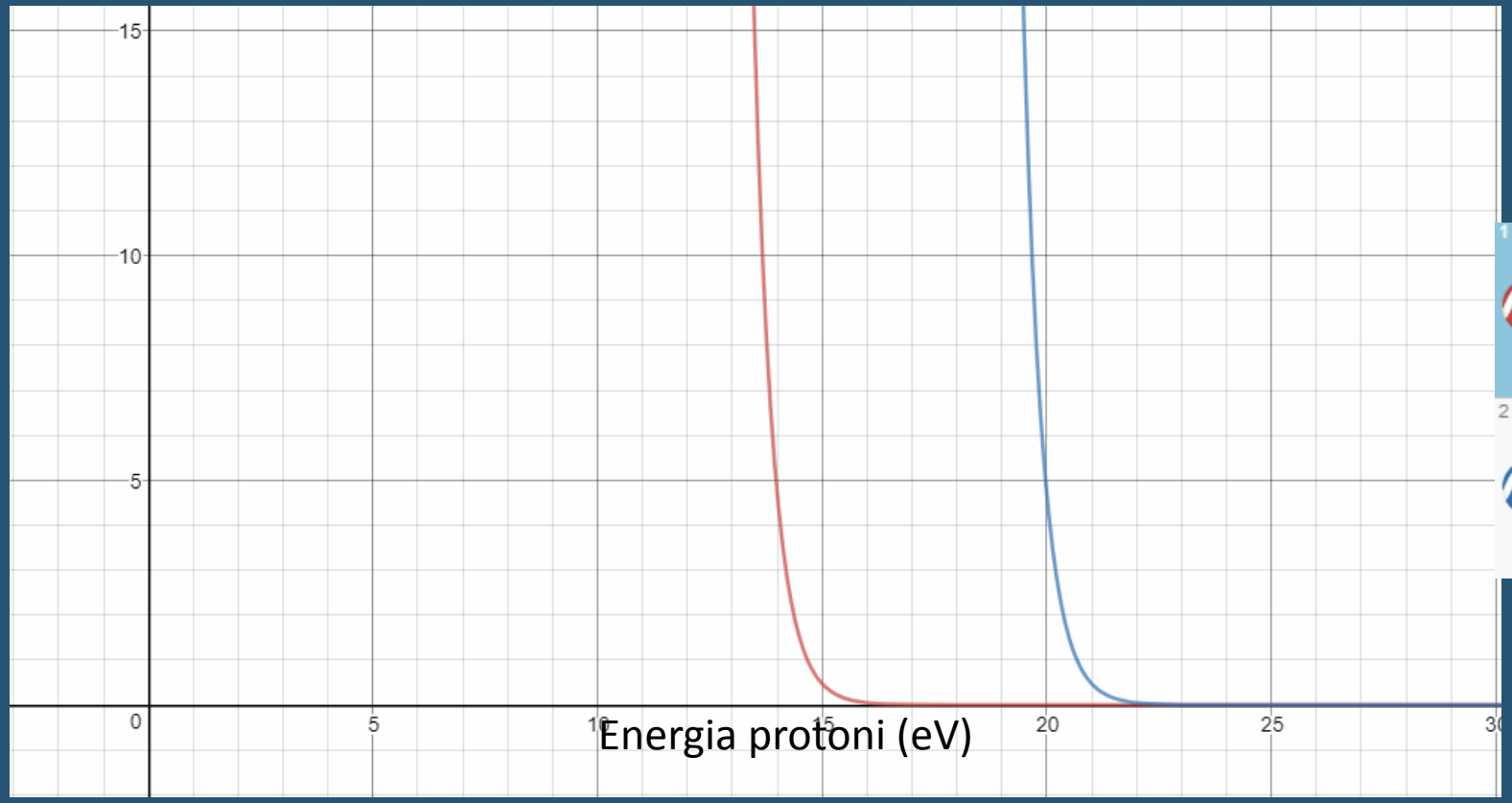
-Backgroung **INTEGRAL**;

-Rosso **CERN-02**;

-Arancione **SALE-02**;

-Verde **TORI-04**;

-Blu sovrapposizione tre osservatori EEE



1 $y = \frac{(1.116 \cdot 10^{65})}{(10^{x}) \cdot (4\pi) \cdot (1.899 \cdot 10^{49})}$

2 $y = \frac{(1.116 \cdot 10^{65})}{(10^{x}) \cdot (4\pi) \cdot (1.899 \cdot 10^{43})}$

$$y = \frac{E\ emessa}{E_{P+}\ 4\pi d^2}$$

Numero di protoni con E = 10^{20} eV : **4.7 x 10^{-6} per m^2**
4.5 per km^2

Attivi solo 7 telescopi su 52 !!

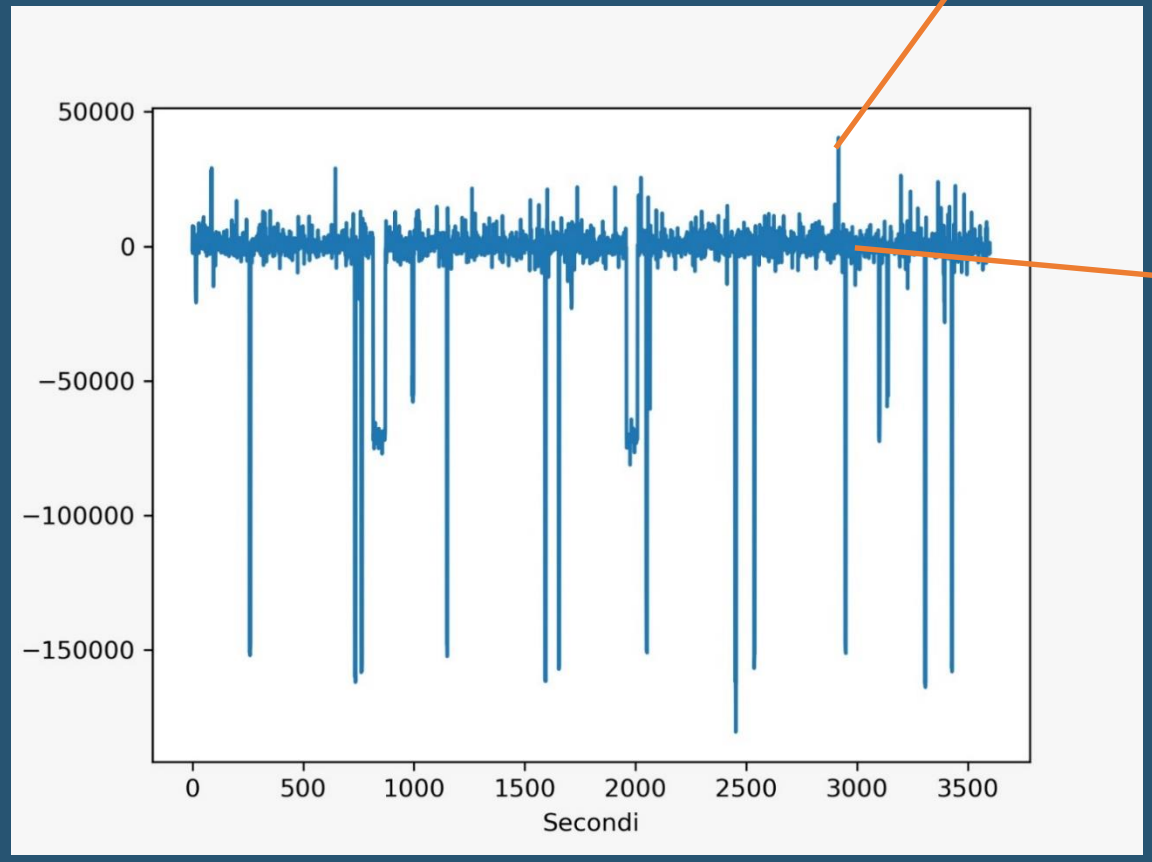
Di cui sei ad alte latitudini

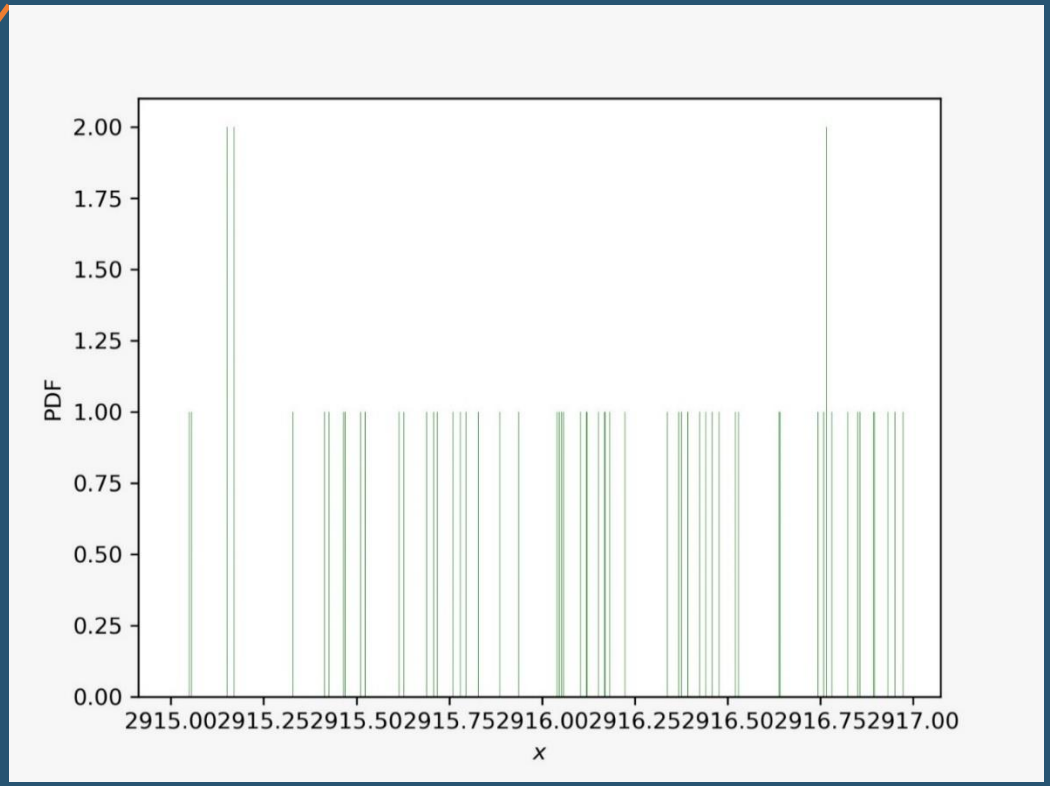
Rilevatore	Altezza della NGC 4993
AREZ-01	17°50’37’’
BOLO-04	16°45’17’’
CERN-01	13°22’30’’
CERN-02	13°22’30’’
SALE-02	21°17’43’’
TORI-03	14°57’43’’
TORI-04	14°57’43’’

Osservatorio di Pino Torinese

Somma di tutti gli eventi (passaggi di muoni) rilevati da sei dei sette osservatori (si esclude Salerno per problemi di importazione dati)

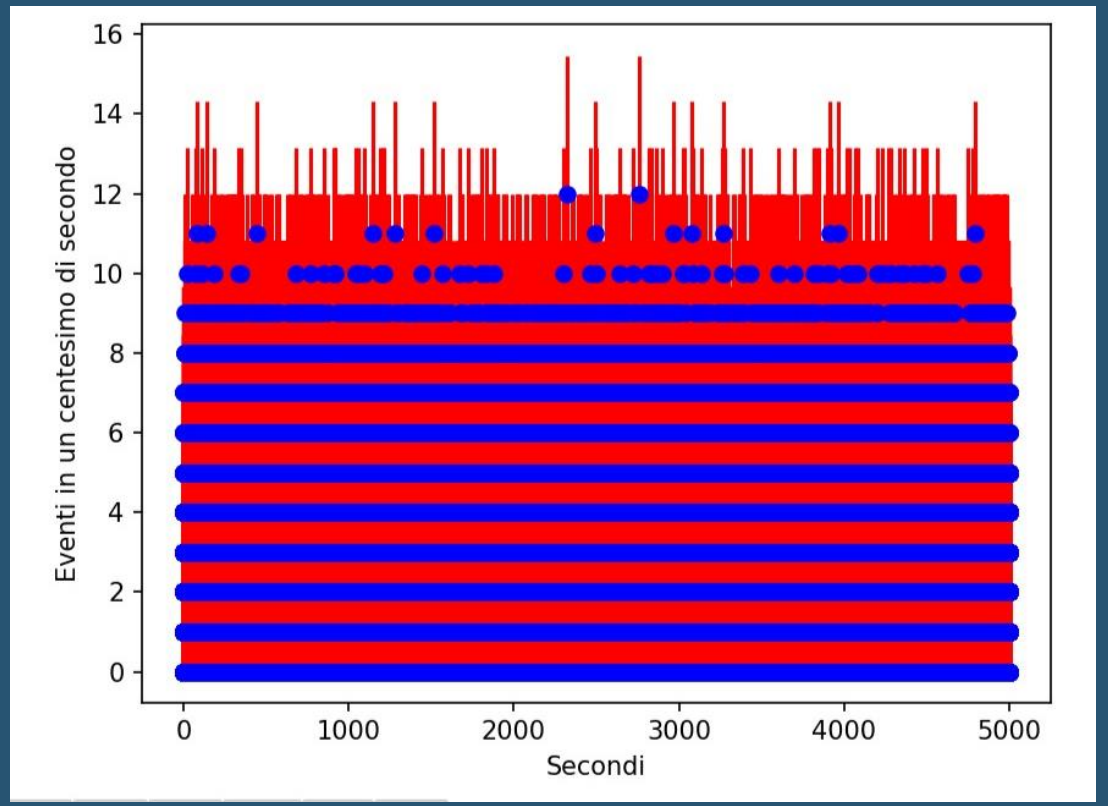
$$\sqrt[3]{\sum_i (y_n - \bar{y}_i)^3}$$

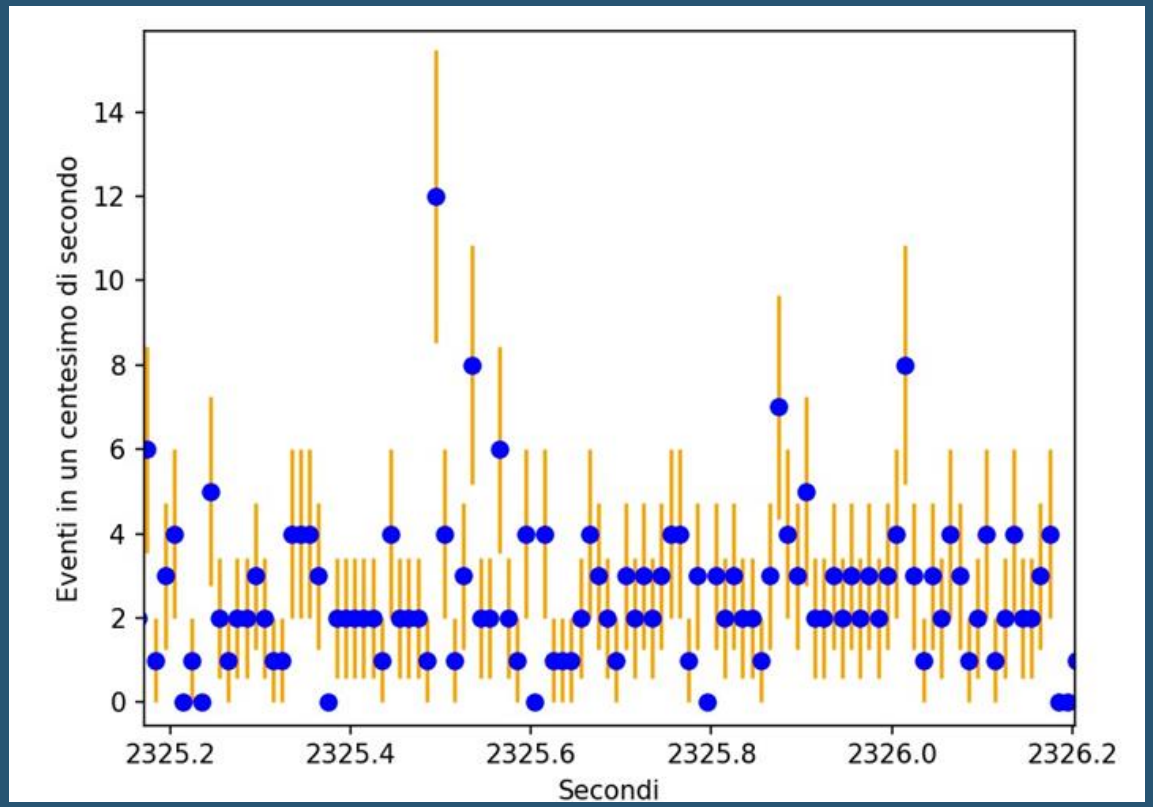




Eventi al microsecondo interni al secondo del picco più alto

Somma degli eventi detetti dai sei rivelatori per ogni centesimo di secondo su un range di 5000 s





Somma degli eventi detetti dai sei rivelatori per ogni centesimo di secondo su un range di 1 s

Lavori in corso:

- **Determinazione della velocità del vento solare in relazione al tipo di brillamento solare. (X9 e X2)**
- **Determinazione della variazione del rapporto del flusso di muoni nel tempo dovuto all'effetto giorno-notte**
- **Confronto tra conteggi/tempo nei casi di Sole attivo e Sole quieto attraverso due diversi telescopi: EEE (muoni) e OULU (neutroni)**

Ringraziamenti:

- **Prof. Vincenzo Di Matteo** – Analisi dati, Liceo Croce
- **Prof.ri La Rosa, Giangalanti, Russo, Cammarata** – Liceo Croce

- **Dott. Ivan Gnesi** – INFN Torino
- **Prof. Giovanni Peres** – Dipartimento di Fisica Palermo (UNIPA)
- **Dott. Carlo Ferrigno** – INTEGRAL Science Data Centre (University of Geneva)
- **Prof. Antonio Maggio** – Osservatorio Astronomico di Palermo (OAPA)
- **Dott.ssa Paola La Rocca** – INFN Catania

"**Remember to look up at the stars and not down at your feet.** Try to make sense of what you see and wonder about what makes the universe exist. **Be curious.** And however difficult life may seem, there is always something you can do and succeed at. It matters that you don't just give up."

-Stephen Hawking