

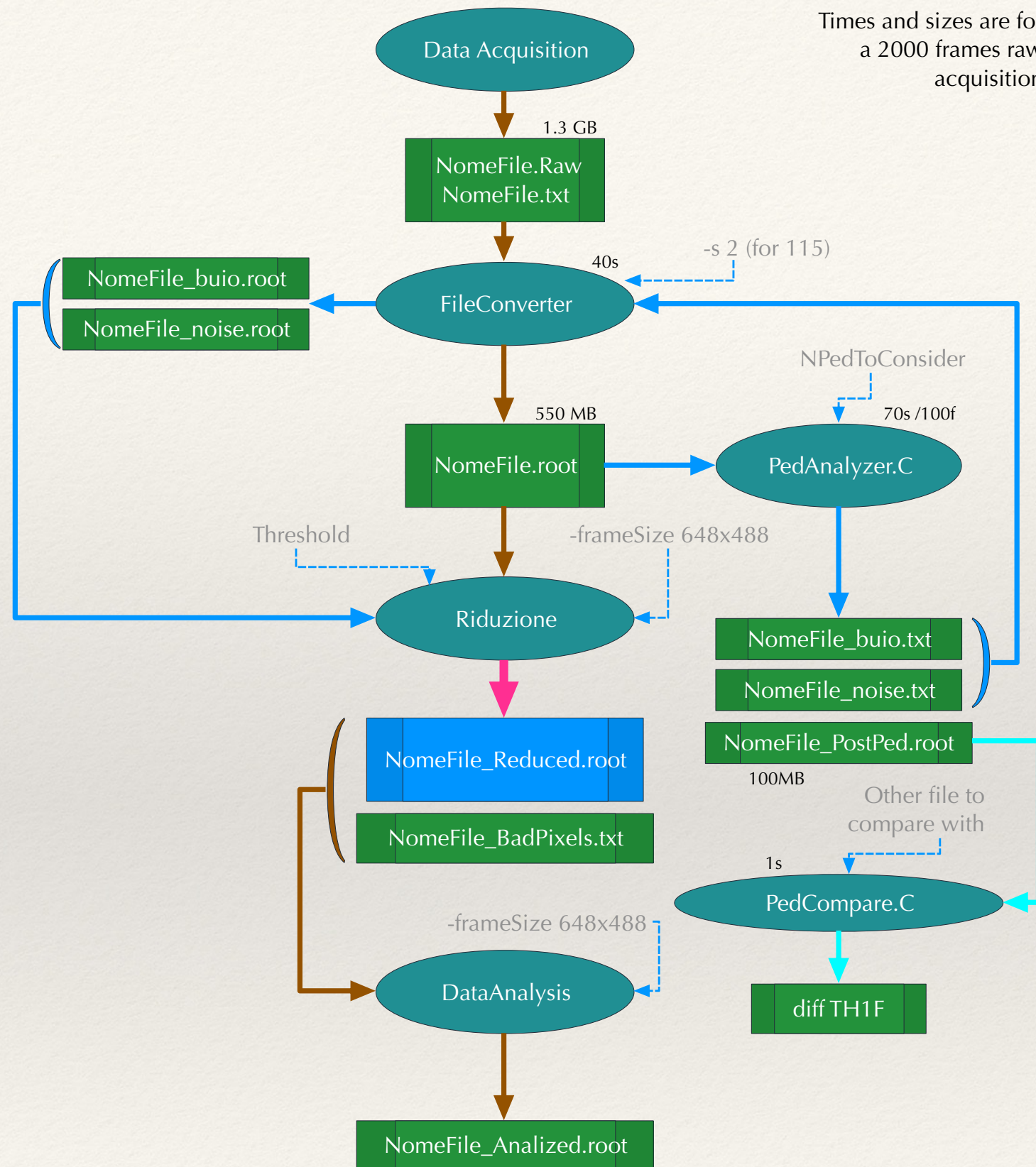
Francesco - Rosa

Aggiornamento sulla simulazione del CMOS

08-06-2018

Work-flow analisi DATI

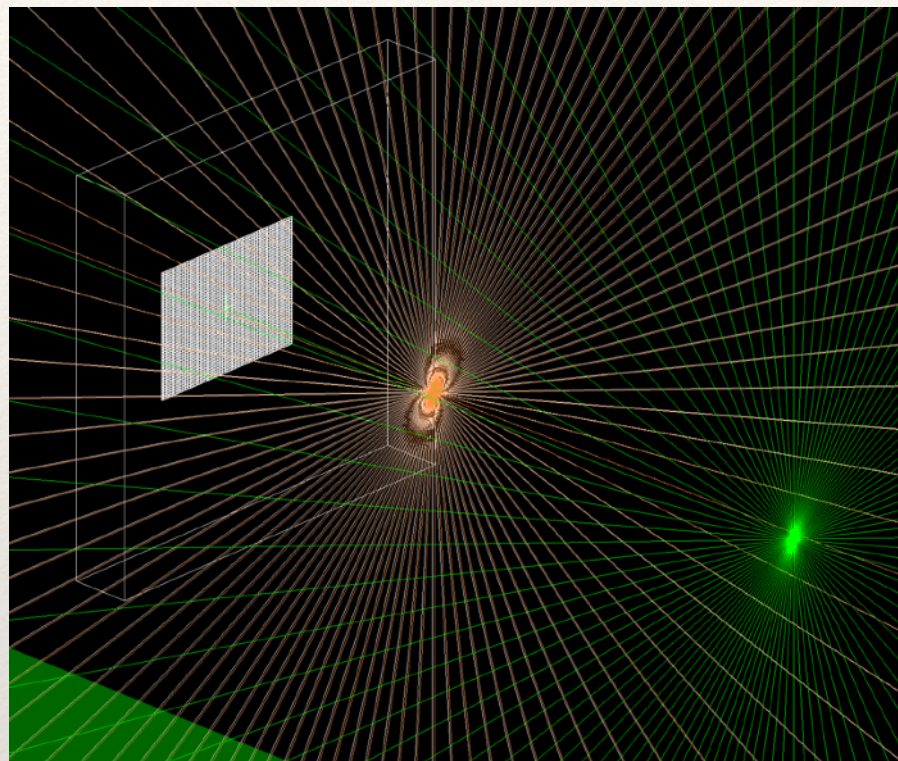
Times and sizes are for
a 2000 frames raw
acquisition



Simulazione Monte Carlo

- ❖ Geometria:
 - ❖ CMOS (sensore MT9V011 o MT9V115)
 - ❖ Volume e materiale sorgente
 - ❖ Eventuale assorbitore / collimatore di fronte la sorgente
 - ❖ Eventuale filtro davanti il sensore
- ❖ Particelle primarie:
 - ❖ Sr-Y puntiforme
 - ❖ Sr-Y estesa
 - ❖ Y estesa con TBR (agar-agar)
 - ❖ Gallio liquido esteso
 - ❖ (virtualmente qualunque isotopo desiderato)

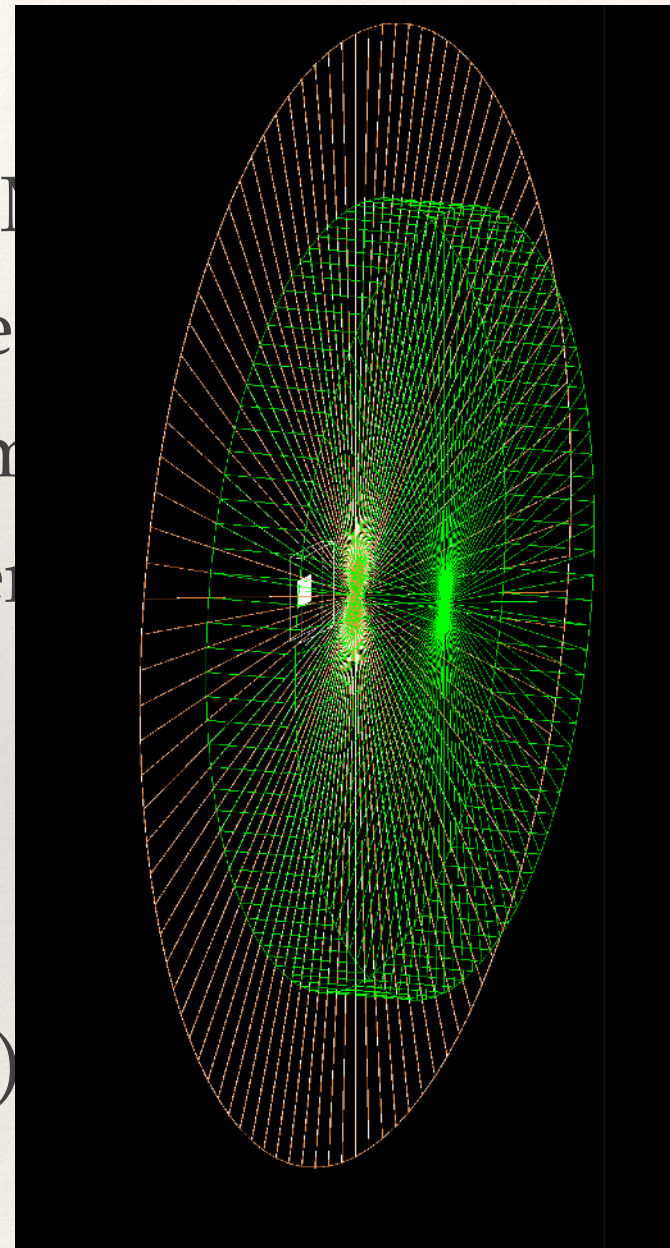
Simulazione Monte Carlo



011 o I
rgente
/collim
ti il ser

Partecine primarie.

- ❖ Sr-Y
- ❖ Sr-Y
- ❖ Y es
- ❖ Gal
- ❖ (vir

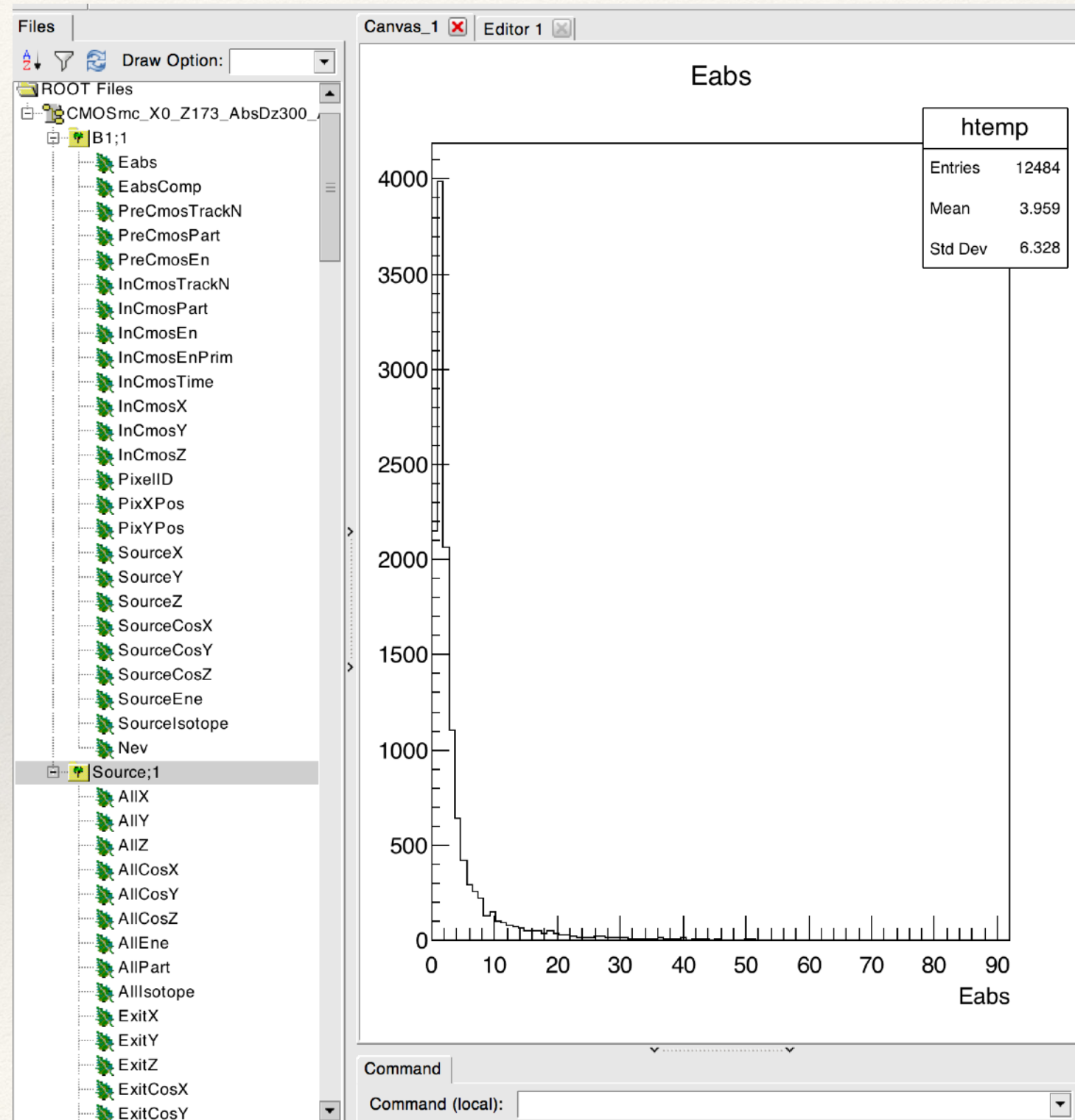


orgent

otopo desiderato)

Simulazione Monte Carlo

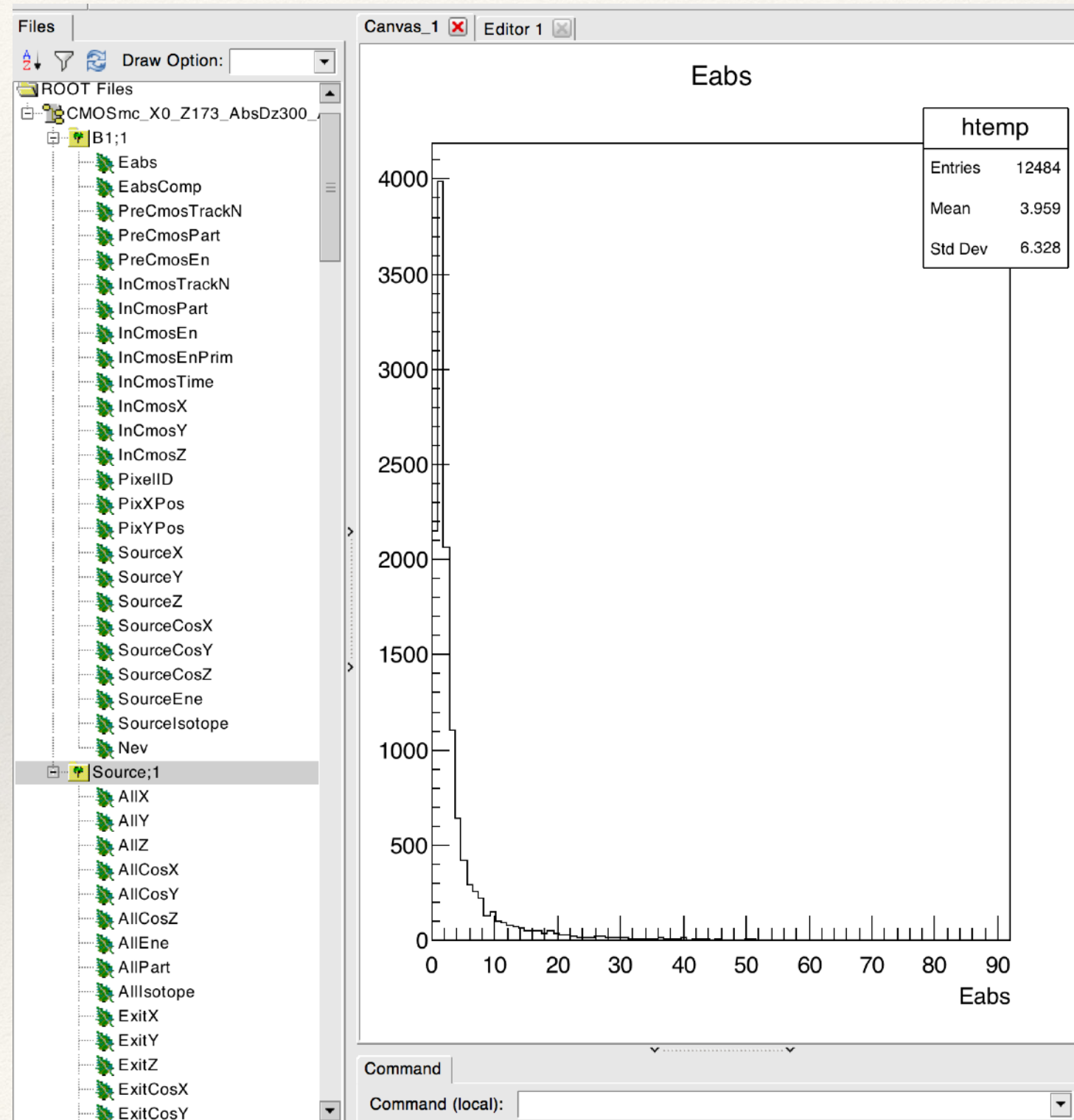
- ❖ Output:
 - ❖ File root nel quale vengono salvate particella primaria per particella primaria (i.e. 1 isotopo radioattivo) le informazioni



Simulazione Monte Carlo

- ❖ Output:
 - ❖ File root nel quale vengono salvate particella primaria per particella primaria (i.e. 1 isotopo radioattivo) le informazioni

Per poter dare in pasto alla stessa catena di analisi dei DATI la simulazione, il file .root creato da Geant va prima “**digerito**” per creare una struttura di **frame**



Simulazione Monte Carlo

❖ *AnaCmosMC.C/h*

- ❖ Classe (in git repo “*CodiceCMOS*”) per analizzare il file MC e creare un file contenente i vari Frame
- ❖ Chiede in input:
 - ❖ Attività della sorgente utilizzata (non “emissività”!)
 - ❖ Durata del frame
 - ❖ Fattore di conversione ADC/KeV
 - ❖ (opzionale): file di dati di noise per aggiungere un rumore gaussiano al MC
- ❖ Funzionamento:
 - ❖ Dall’attività della sorgente valuta a quanti frame equivale la quantità di primari simulati, e quante storie di primari devono (in media) finire in ogni frame
 - ❖ Genera quindi un numero poissoniano che usa come numero di “storie di primari” da inserire nel dato frame

```
class AnaCmosMC {
public:
    TFile *FrameFile;
    TFile *ImageFile;

    TTree      *fChain;    //pointer to t
    Int_t      fCurrent;   //current Tree
    // Double_t ConvFactor=10*1e3;

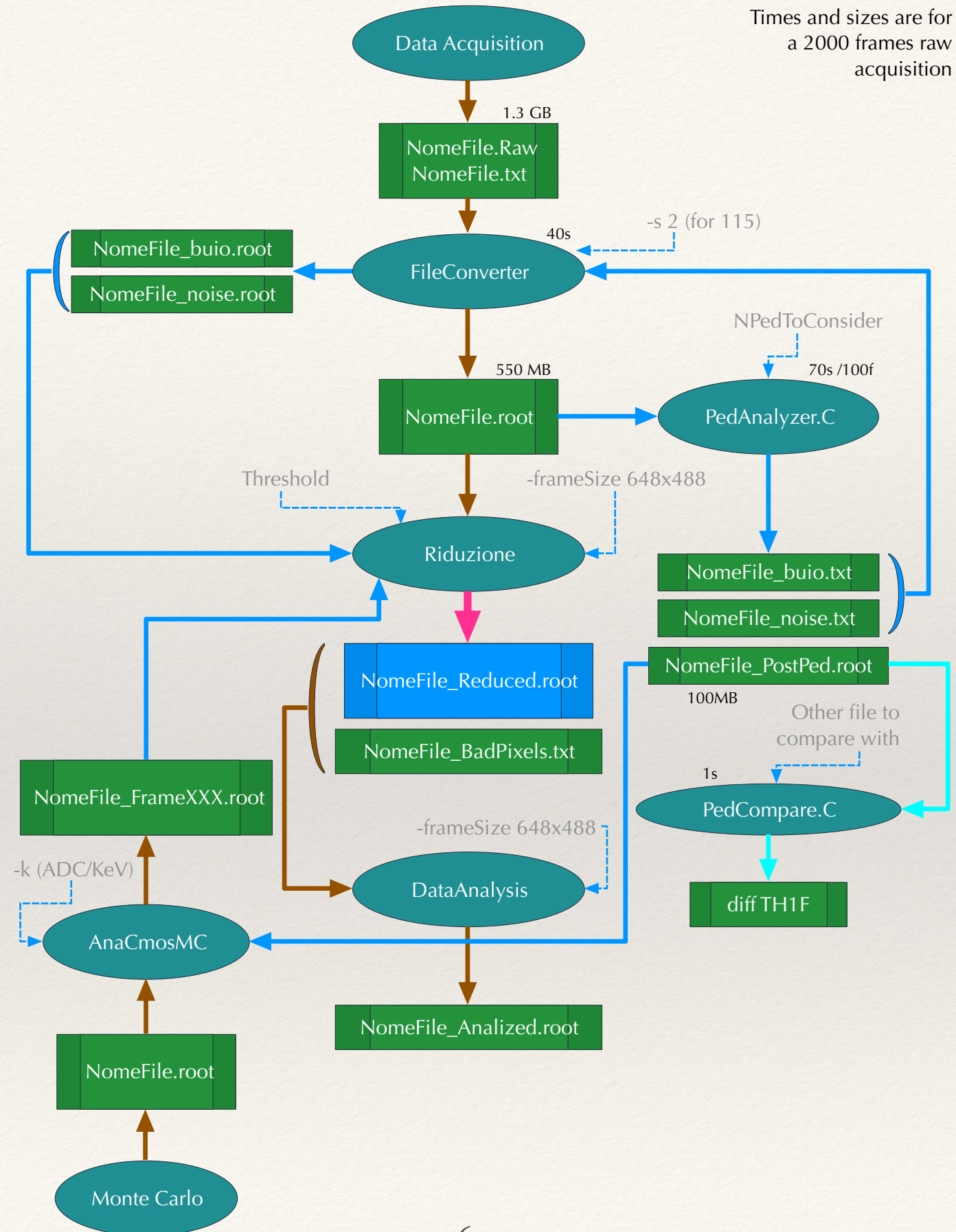
    TVectorT<double> *fVectorNoise;

    bool NoiseFlag=kTRUE;

    // TH2F* cluster;
    // Fixed size dimensions of array or coll
    // Declaration of leaf types
    Double_t      Eabs;
    vector<double> *EabsComp;
    Double_t      PreCmosTrackN;
    vector<double> *PreCmosPart;
    vector<double> *PreCmosEn;
    Double_t      InCmosTrackN;
    vector<double> *InCmosPart;
    vector<double> *InCmosEn;
    vector<double> *InCmosEnPrim;
    vector<float> *InCmosTime;
    vector<double> *InCmosX;
    vector<double> *InCmosY;
    vector<double> *InCmosZ;
    vector<double> *PixelID;
    vector<double> *PixXPos;
    vector<double> *PixYPos;
    Double_t      SourceX;
    Double_t      SourceY;
    Double_t      SourceZ;
    vector<double> *SourceCosX;
    vector<double> *SourceCosY;
    vector<double> *SourceCosZ;
    vector<double> *SourceEne;
    vector<double> *SourceIsotope;
    Int_t         Nev;

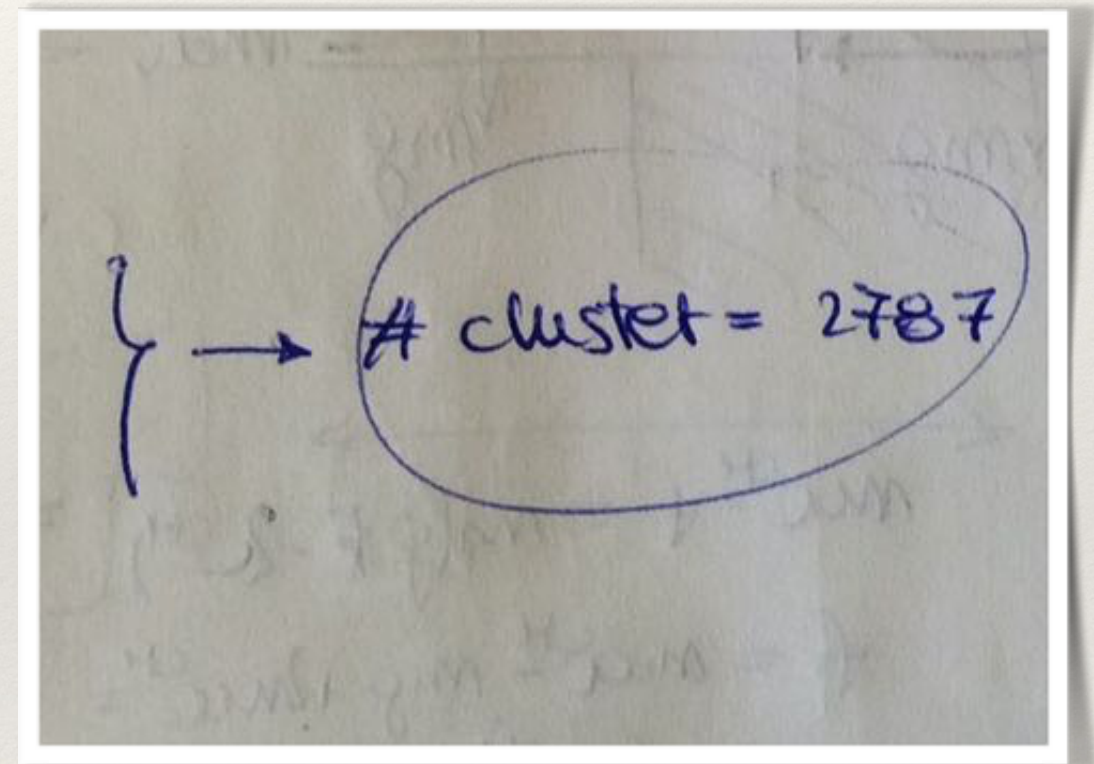
    // List of branches
    TBranch      *b_Eabs;    //!
    TBranch      *b_EabsComp; //!
    TBranch      *b_PreCmosTrackN; //!
    TBranch      *b_PreCmosPart; //!
    TBranch      *b_PreCmosEn; //!
    TBranch      *b_InCmosTrackN; //!
    TBranch      *b_InCmosPart; //!
    TBranch      *b_InCmosEn; //!
    TBranch      *b_InCmosEnPrim; //!
    TBranch      *b_InCmosTime; //!
    TBranch      *b_InCmosX; //!
    TBranch      *b_InCmosY; //!
    TBranch      *b_InCmosZ; //!
    TBranch      *b_PixelID; //!
    TBranch      *b_PixXPos; //!
    TBranch      *b_PixYPos; //!
    TBranch      *b_SourceX; //!
    TBranch      *b_SourceY; //!
    TBranch      *b_SourceZ; //!
    TBranch      *b_SourceCosX; //!
    TBranch      *b_SourceCosY; //!
    TBranch      *b_SourceCosZ; //!
    TBranch      *b_SourceEne; //!
    TBranch      *b_SourceIsotope; //!
    TBranch      *b_Nev;    //!
```

Work-flow analisi MONTE CARLO



Accordo DATI-MC

- ❖ Abbiamo iniziato il confronto su di una singola misura (campagna “stabilità in Temperatura”):
 - ❖ Sensore MT9V115
 - ❖ Sorgente di Sr estesa (Rm)
 - ❖ Distanza sorgente-sensore 1.73mm
 - ❖ $T=22^{\circ}\text{C}$
 - ❖ 3000 frames da 200ms
- ❖ Strategia:
 - ❖ Cercare il fattore di conversione ($k=\text{ADC}/\text{KeV}$) da applicare al MC che riproduca al meglio #cluster e spettri (a partire da Cluster3_3)

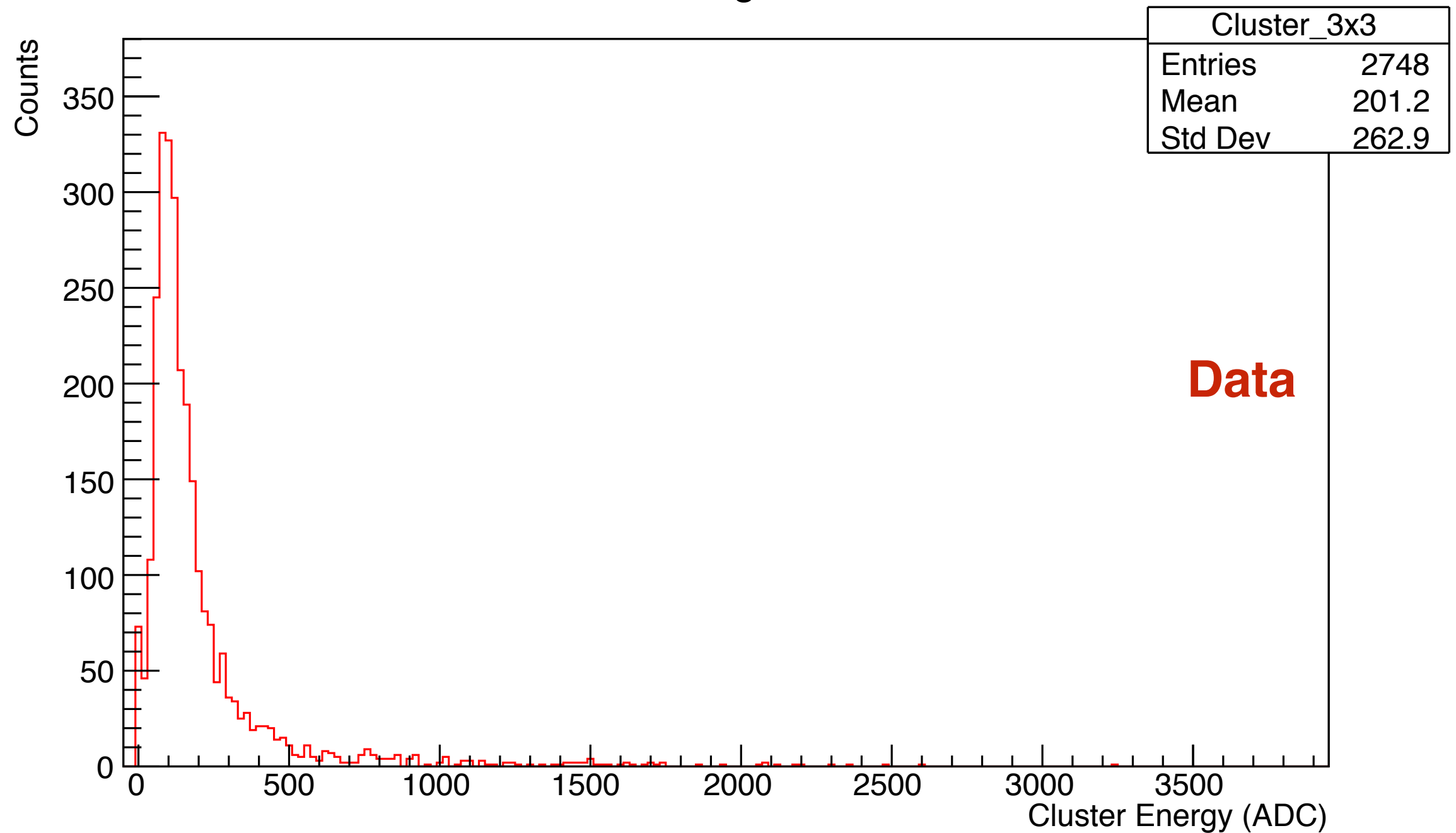


Accordo DATI-MC

- ❖ Nel cercare di accordare rate e spettro, abbiamo imparato che:
 - ❖ Altro “manico” fondamentale è lo **spessore della zona attiva** del CMOS
 - ❖ Se è troppo sottile (era nel MC originale 2.5um), anche “beccando” la rate gli spettri restano insanabilmente diversi
 - ❖ Giocare sul K sposta significativamente gli spettri, ma più lentamente il numero di cluster ricostruiti
- ❖ Il **migliore accordo** si è ottenuto per **DZ=4.5um** (tesi di Stefano, e misure di Leonello, anche se per altro sensore) e **K=80ADC/KeV**

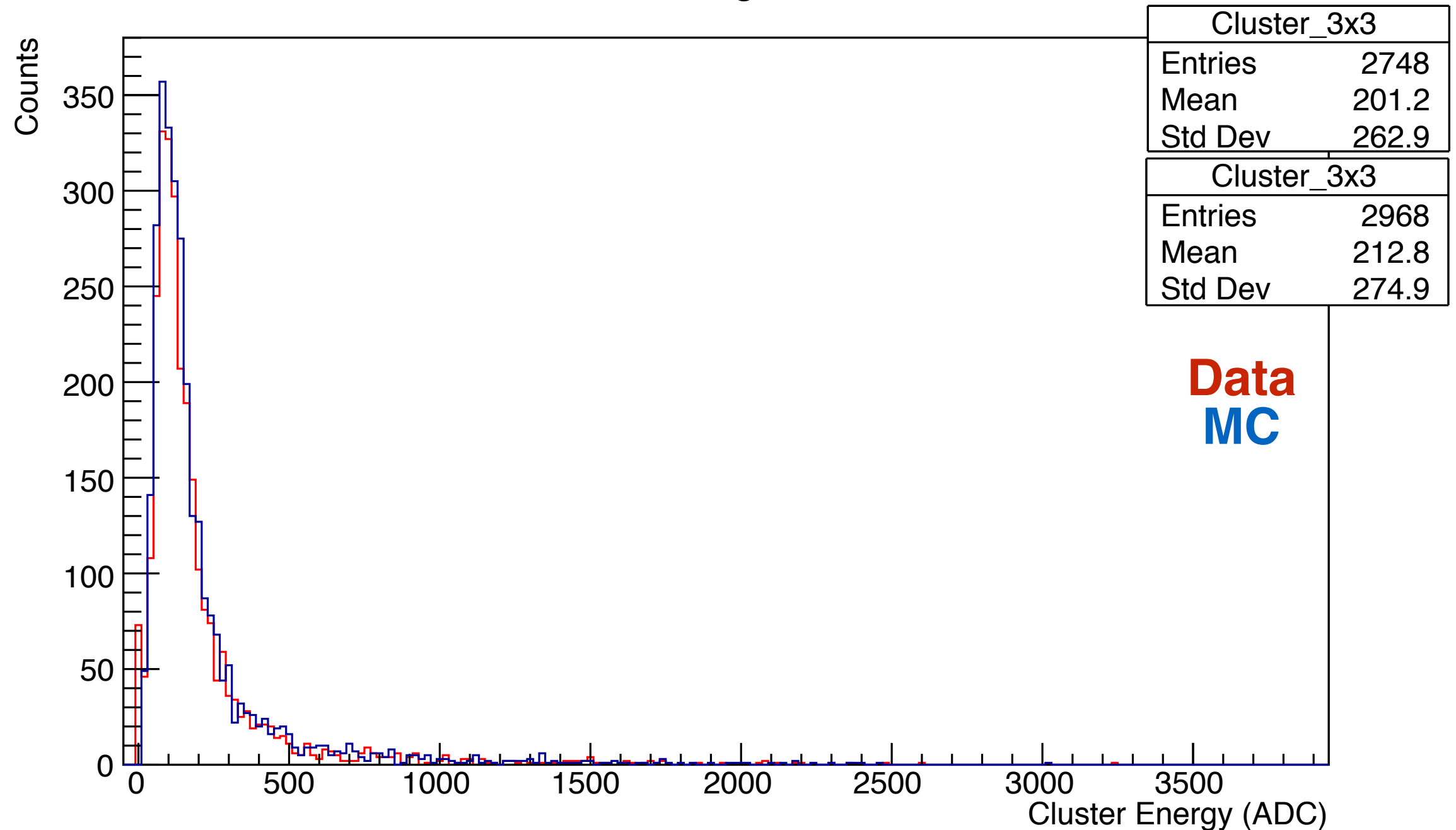
Accordo DATI-MC

Cluster signal 3x3

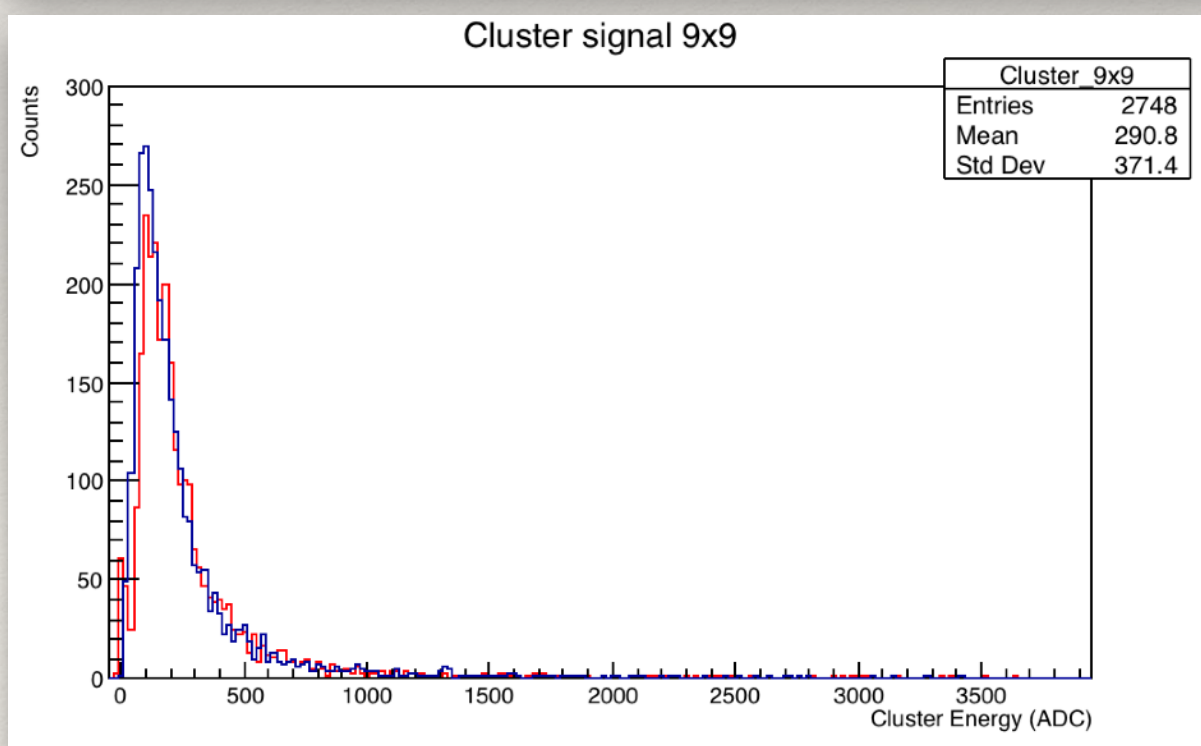
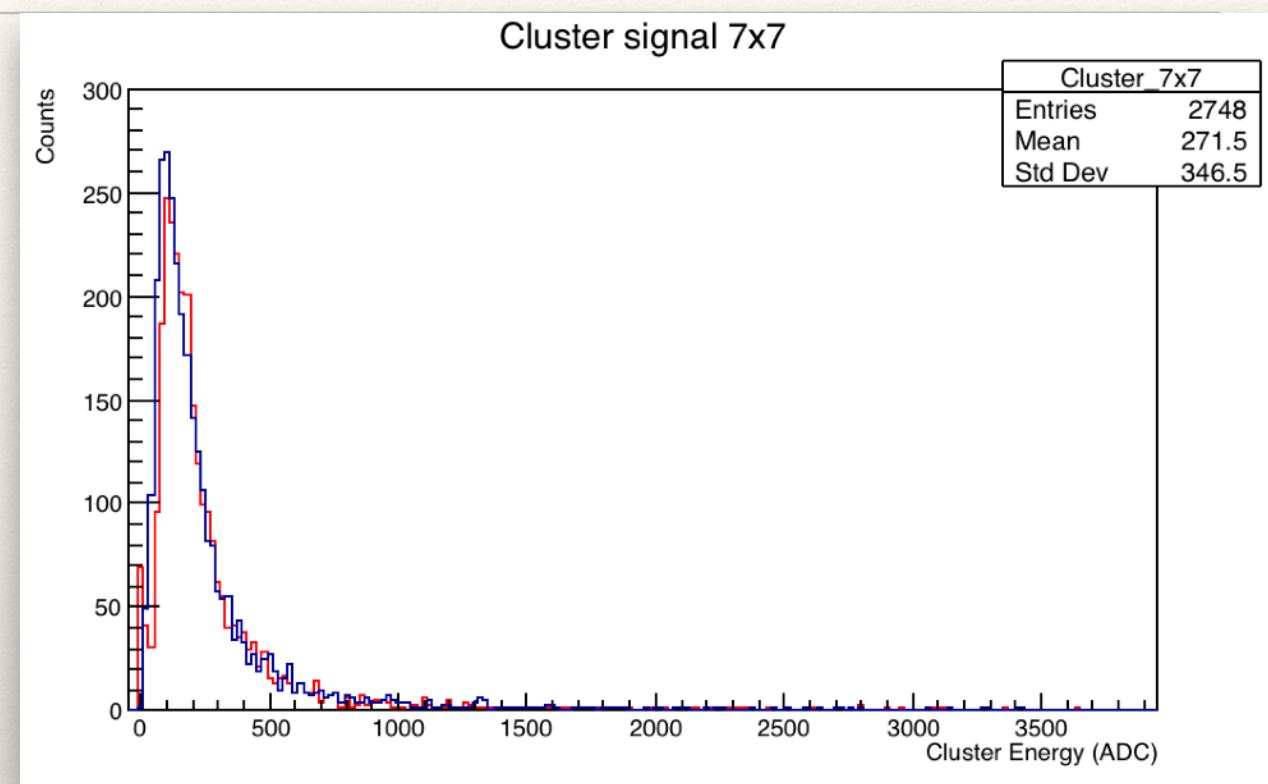
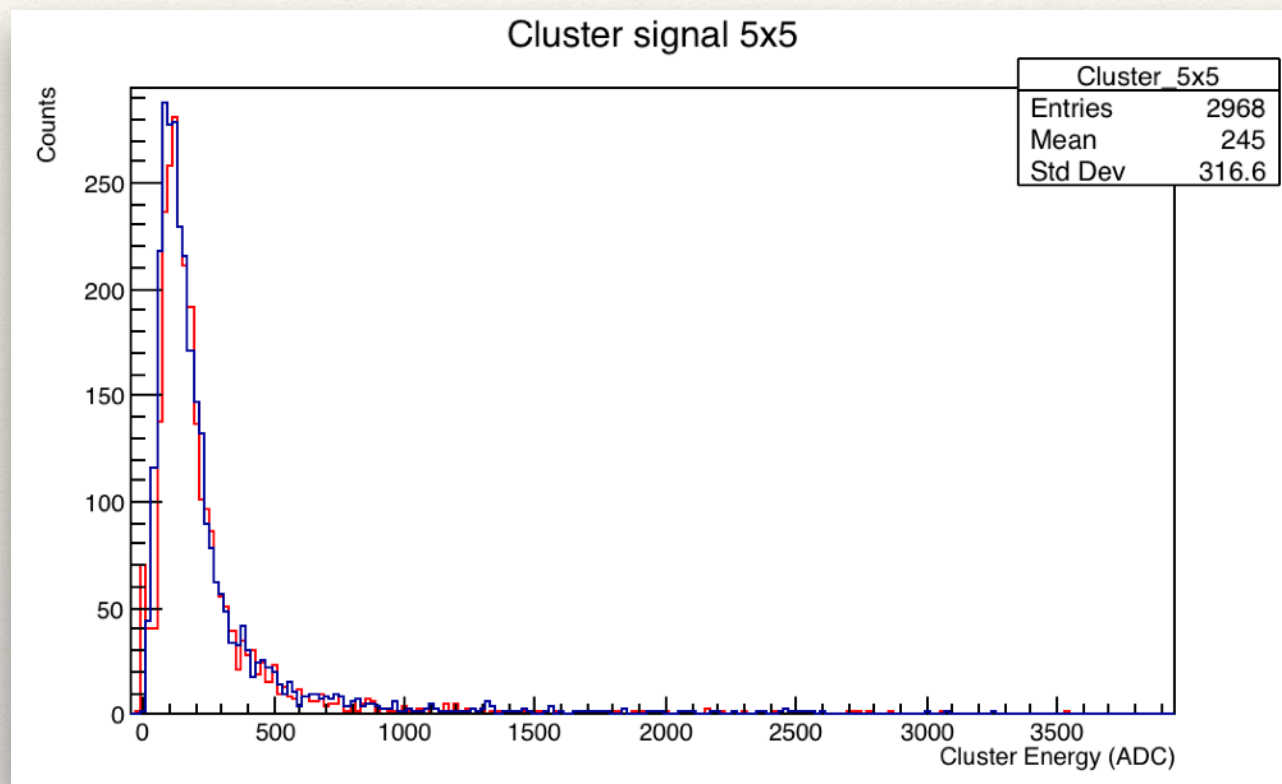


Accordo DATI-MC

Cluster signal 3x3



Accordo DATI-MC



L'accordo (ottenuto matchando il cluster 3x3) si mantiene abbastanza buono all'interno della stessa simulazione per i vari istogrammi

Accordo DATI-MC

Accordo DATI-MC

- ❖ Riassumendo:
 - ❖ Concentrandoci su una configurazione Dati/MC abbiamo trovato una parametrizzazione ($DZ=4.5\mu\text{m}$ & $K=80\text{ADC/KeV}$) che ci permette di accordare in maniera soddisfacente Rate & Spettri del MC con i dati...

Accordo DATI-MC

- ❖ Riassumendo:
 - ❖ Concentrandoci su una configurazione Dati/MC abbiamo trovato una parametrizzazione ($DZ=4.5\mu\text{m}$ & $K=80\text{ADC/KeV}$) che ci permette di accordare in maniera soddisfacente Rate & Spettri del MC con i dati...



Accordo DATI-MC

- ❖ Riassumendo:

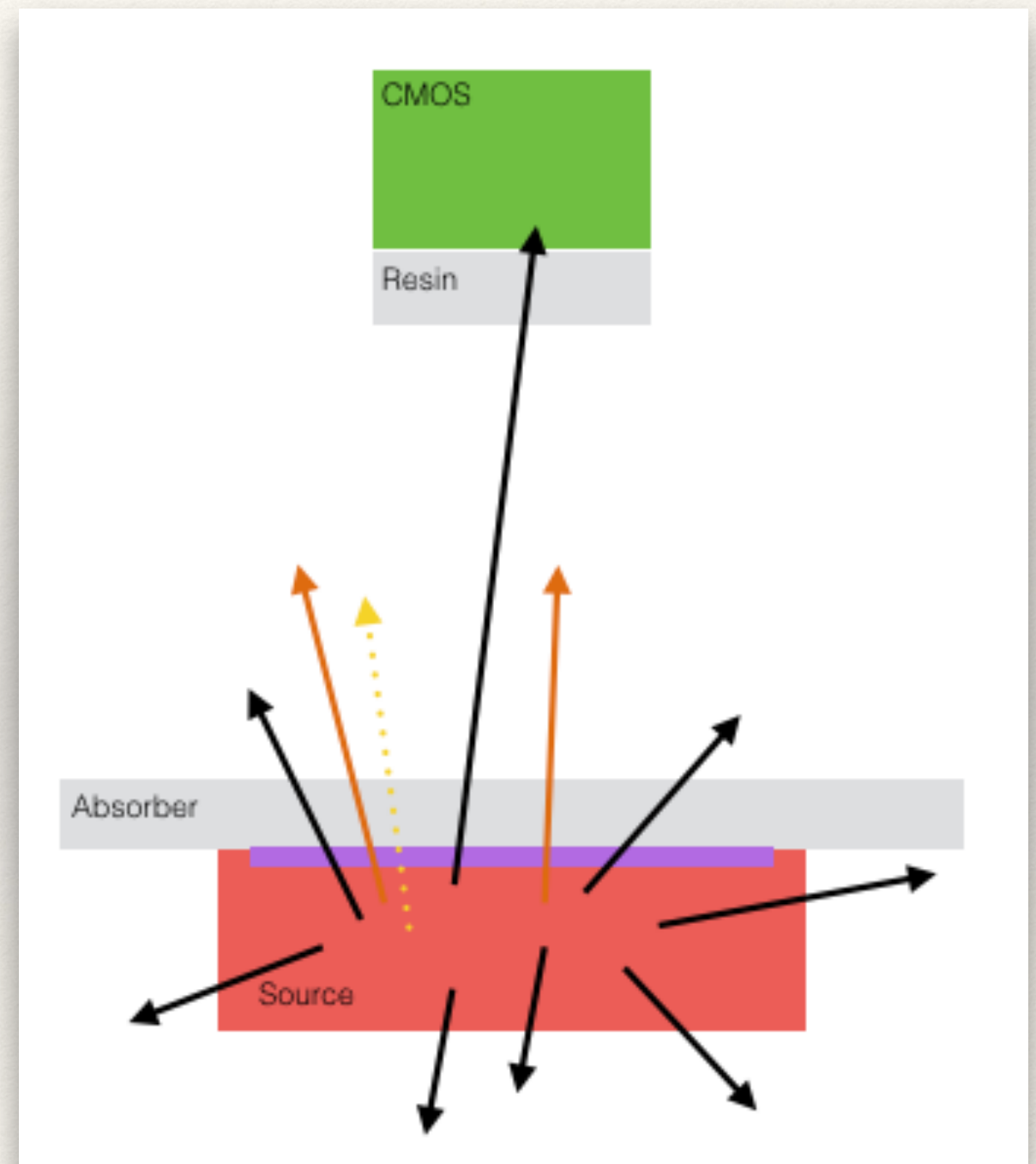
- ❖ Concentrandoci su una configurazione Dati/MC abbiamo trovato una parametrizzazione ($DZ=4.5\mu\text{m}$ & $K=80\text{ADC/KeV}$) che ci permette di accordare in maniera soddisfacente Rate & Spettri del MC con i dati...



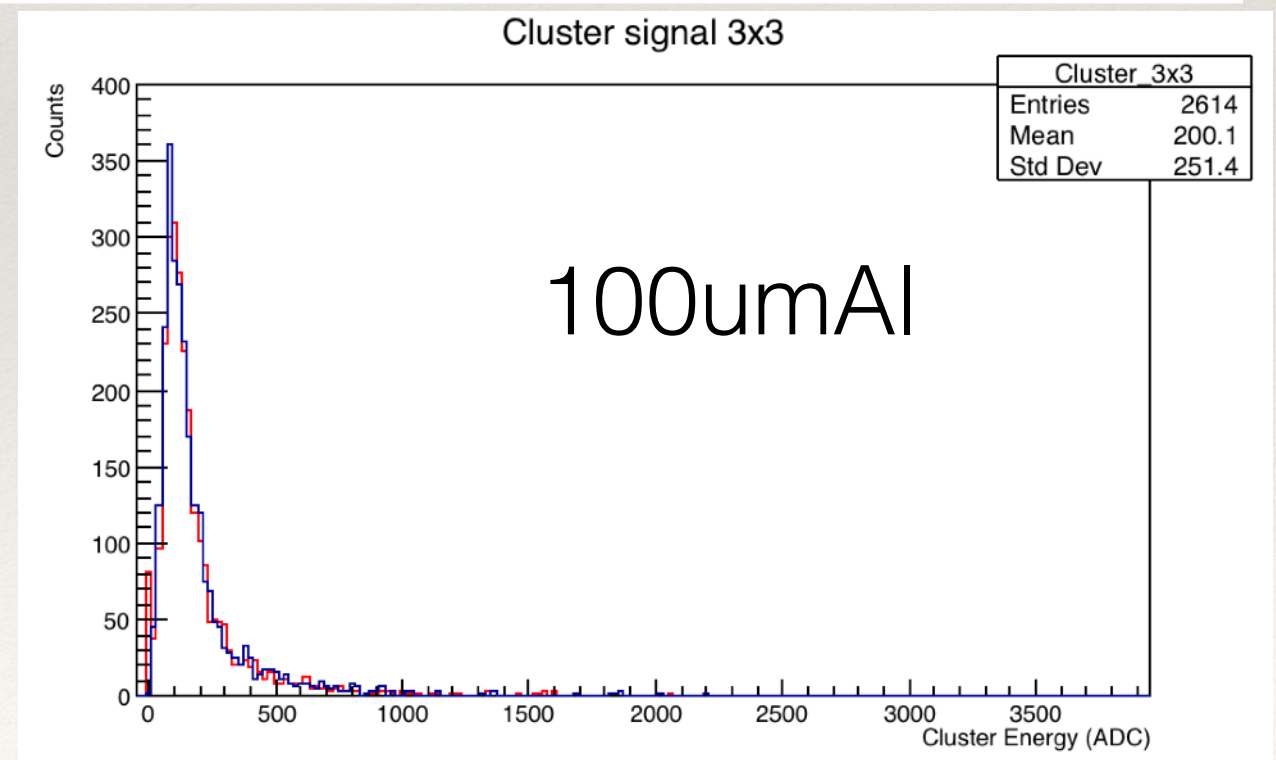
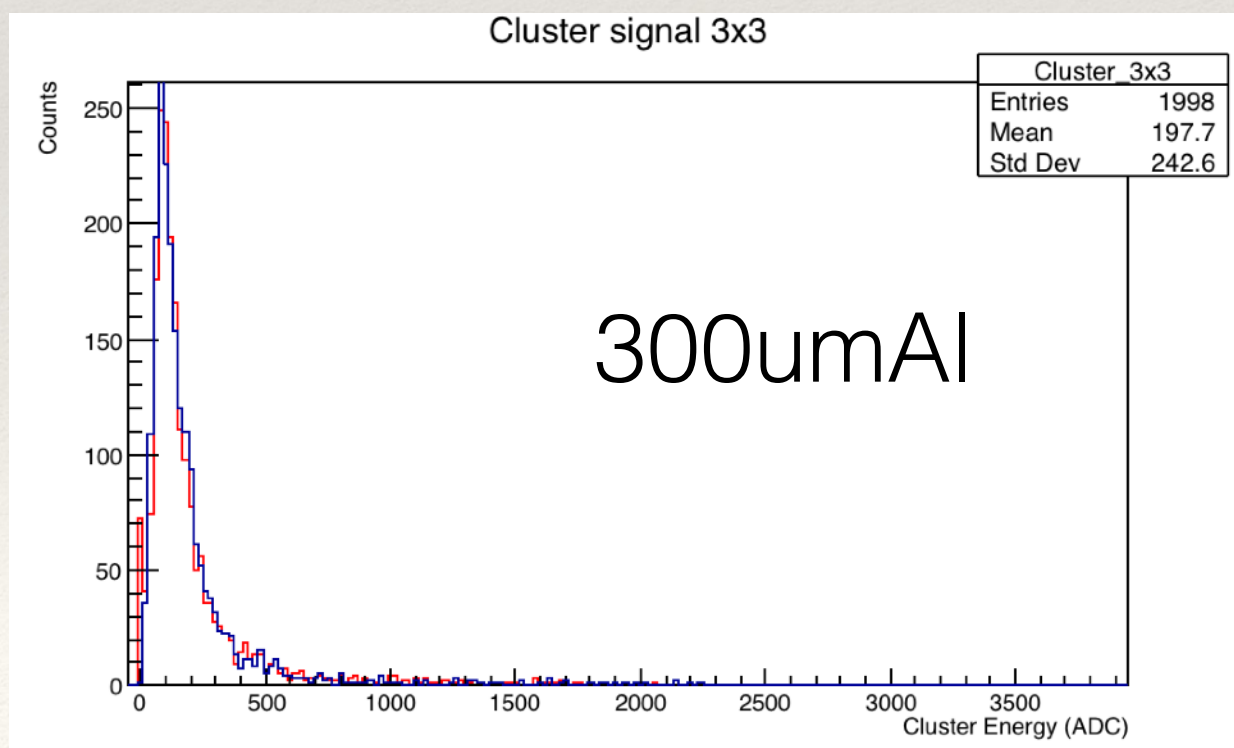
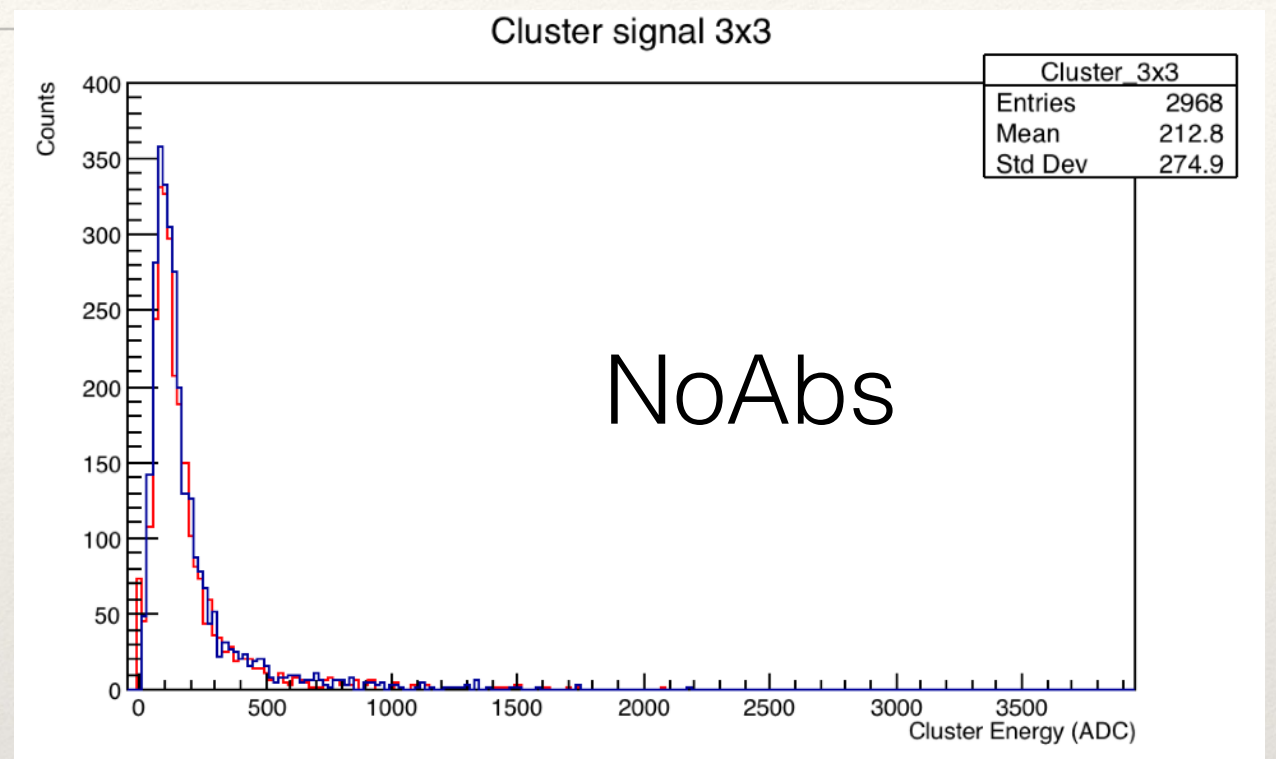
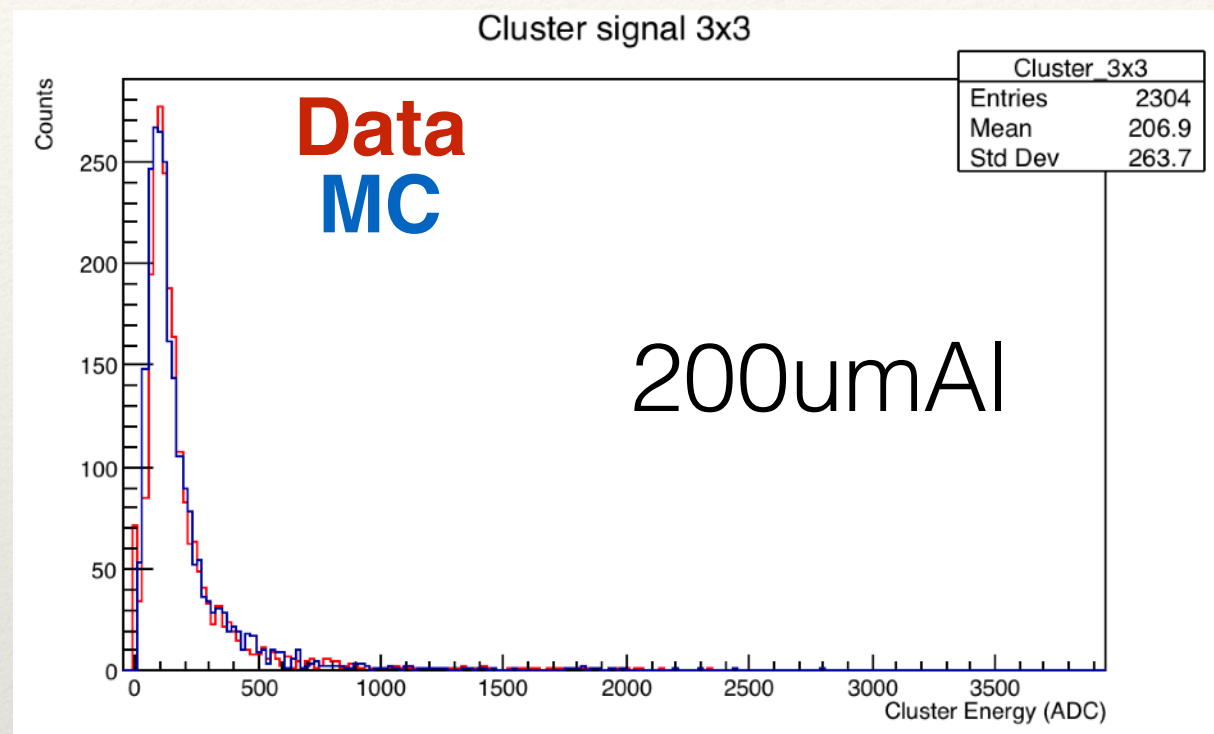
- ❖ Per verificarlo, abbiamo chiesto a Rosa di effettuare qualche altra misura inserendo degli spessori di assorbitorre fra sorgente e CMOS

Accordo DATI-MC

- ❖ Configurazione misura:
 - ❖ Sensore MT9V115
 - ❖ Sorgente Sr extended RM
 - ❖ Assorbitori:
 - ❖ Niente
 - ❖ 100um Al
 - ❖ 200um Al
 - ❖ 300um Al
 - ❖ 1.1mm ABS
- ❖ Simulazione delle stesse configurazioni
(DZ=4.5um & K=80 ADC/KeV)

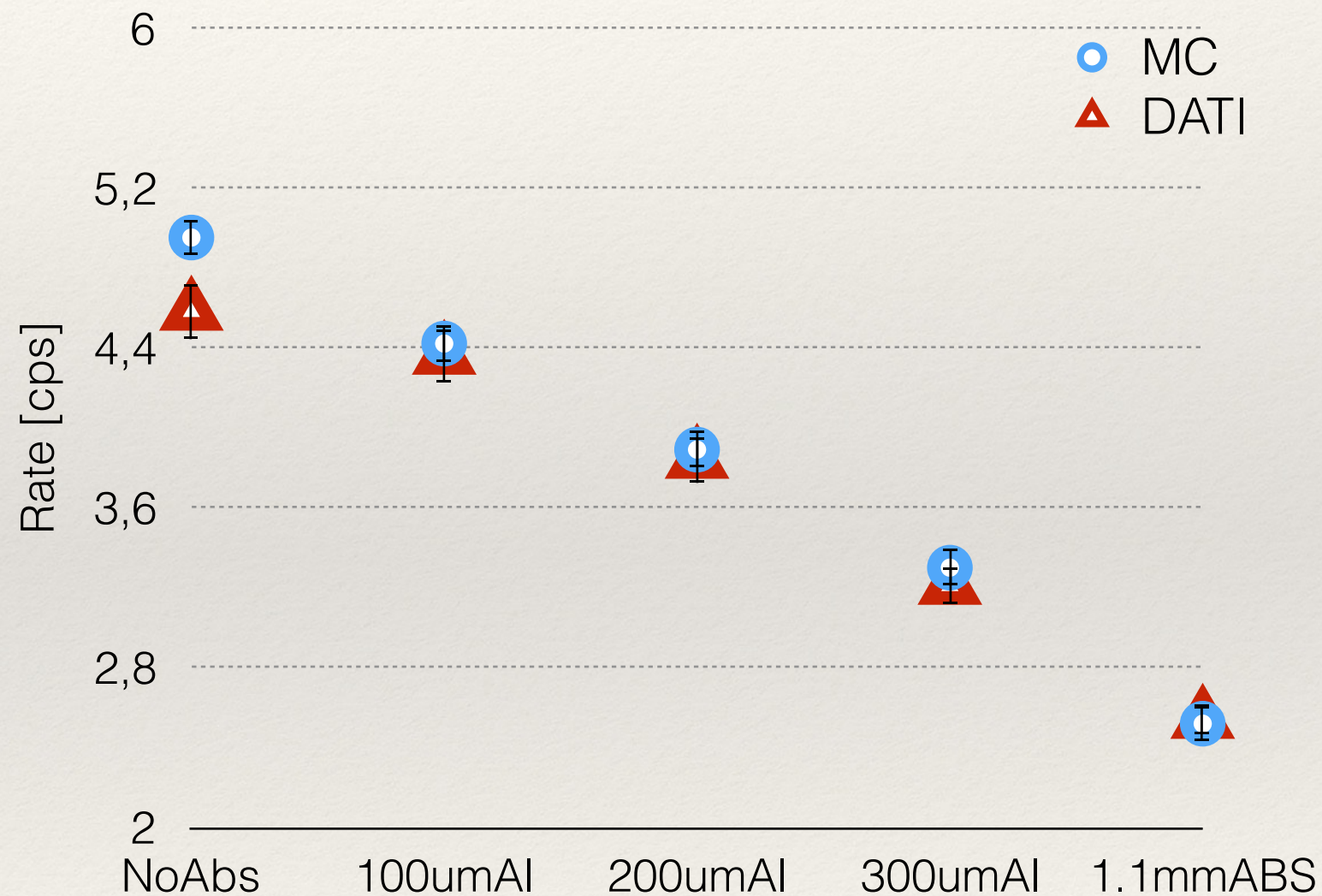


Accordo DATI-MC



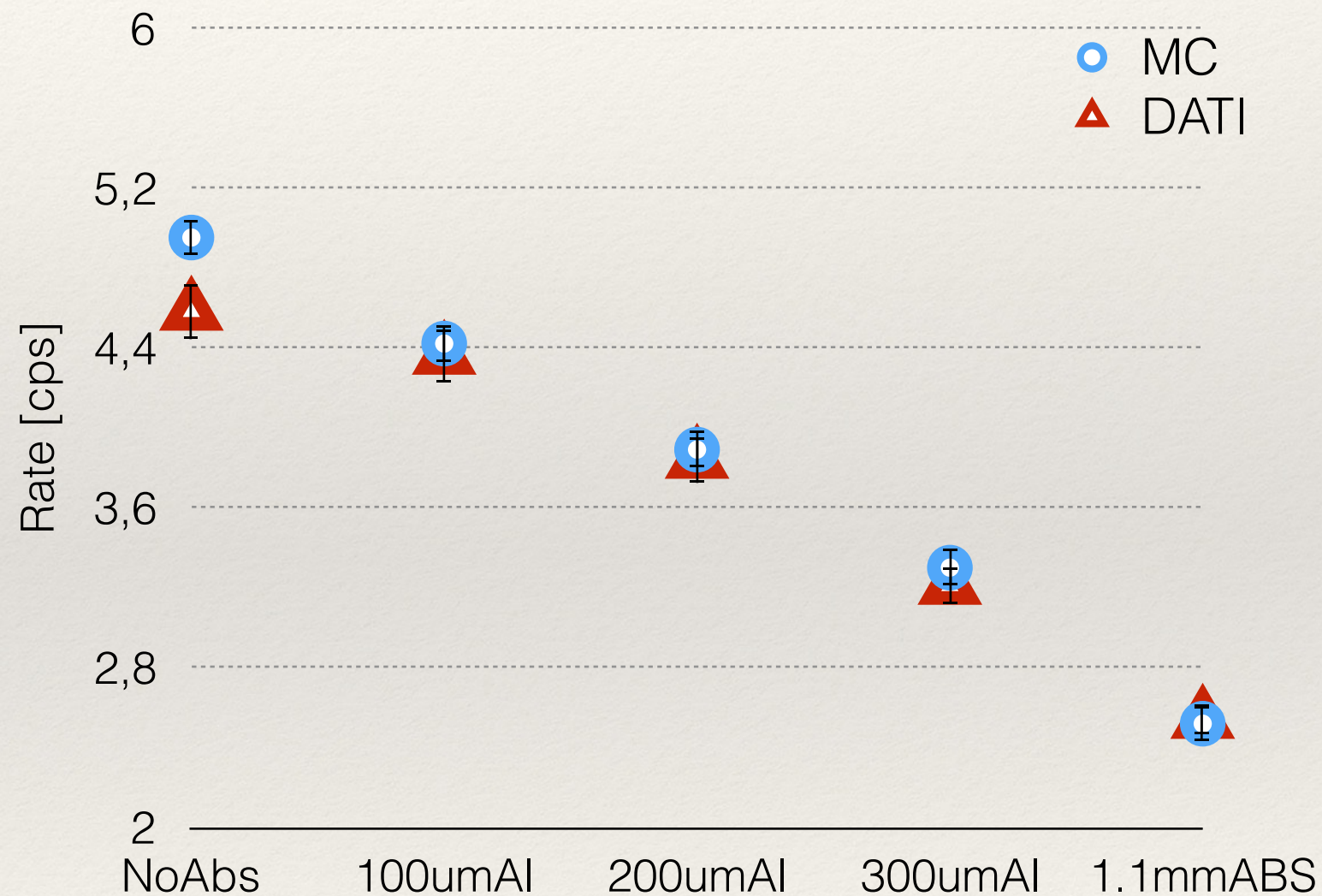
Accordo DATI-MC

DT	MONTE CARLO		DATI	
	Nseed	RateSeed	Nseed	RateSeed
600				
—	2968	4,95	2748,0	4,58
100umAl	2650	4,42	2614,0	4,36
200umAl	2335	3,89	2304,0	3,84
300umAl	1980	3,30	1924,0	3,21
1.1mm ABS	1509	2,52	1522,0	2,54

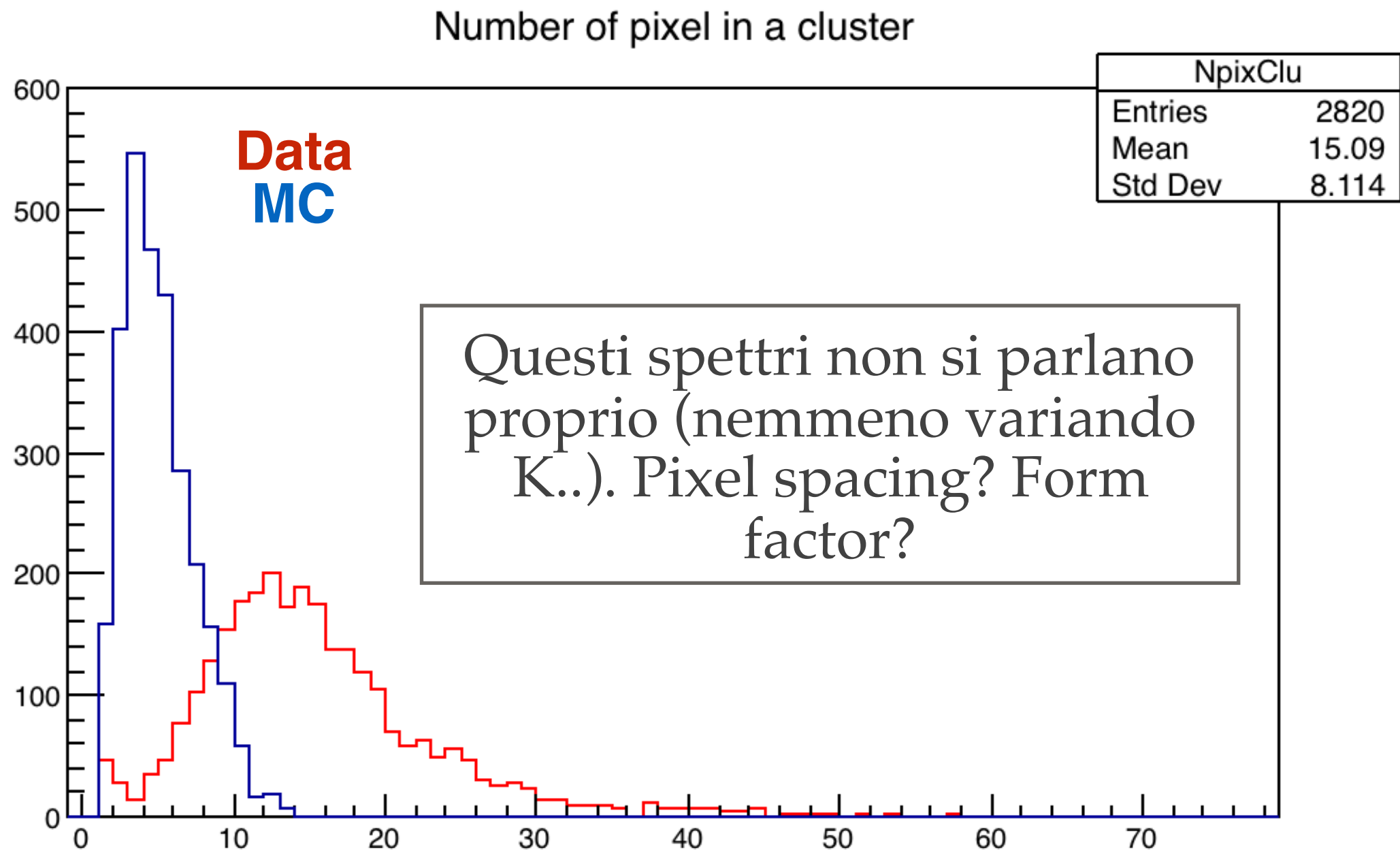


Accordo DATI-MC

DT	MONTE CARLO		DATI	
	Nseed	RateSeed	Nseed	RateSeed
600				
—	2968	4,95	2748,0	4,58
100umAl	2650	4,42	2614,0	4,36
200umAl	2335	3,89	2304,0	3,84
300umAl	1980	3,30	1924,0	3,21
1.1mm ABS	1509	2,52	1522,0	2,54



Cose da capire



Accordo DATI-MC

- ❖ Riassumendo:
 - ❖ Concentrandoci su una configurazione Dati/MC abbiamo trovato una parametrizzazione ($DZ=4.5\mu\text{m}$ & $K=80\text{ADC/KeV}$) che ci permette di **accordare in maniera soddisfacente Rate & Spettri** del MC con i dati anche su altre configurazioni misurate e simulate
 - ❖ Se riflettiamo un secondo su tutto il mistero che aleggia attorno a questo sensore (dimensioni dei pixel, spessore, spazio morto, dimensioni totali..) **sa quasi di miracolo!**
- ❖ Restano ancora alcune cose da capire, possiamo provare ad arrovelarci, ma *ad un certo punto dovremo anche alzare le mani e dirci soddisfatti*