# AGATA Analysis Workshop 2025

## PostPSA Calibration

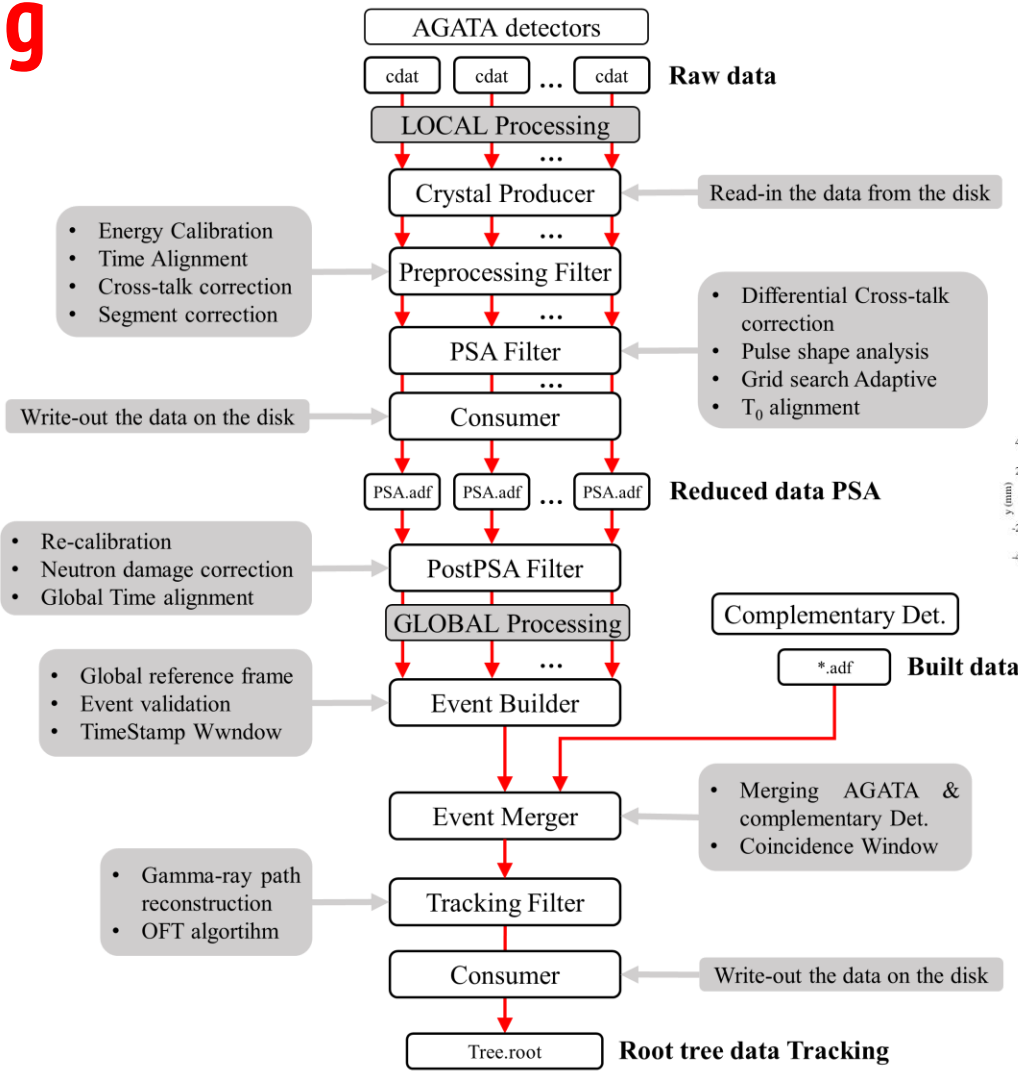
**R.M. Pérez-Vidal**

14/01/2025, Lyon

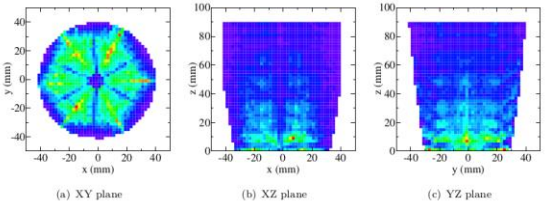
IFIC INSTITUT DE FÍSICA CORPUSCULAR
EXCELENCIA SEVERO OCHOA
VNIVERSITAT DE VALÈNCIA
CSIC CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

# Data Processing

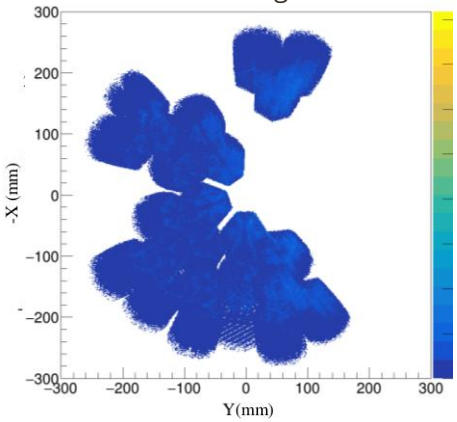
## Narval actors




Pulse Shape Analysis



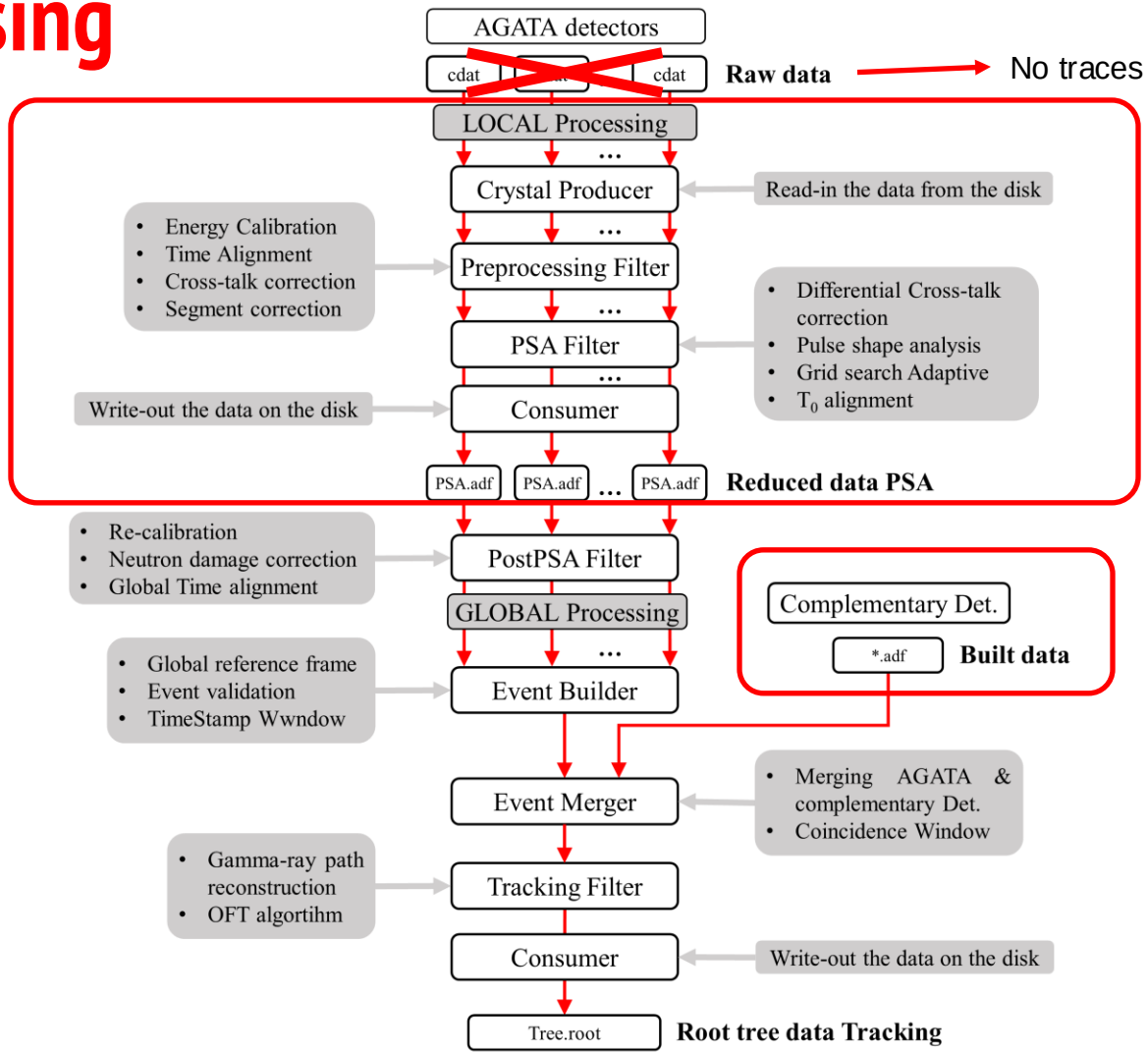

Tracking




First Interaction pattern

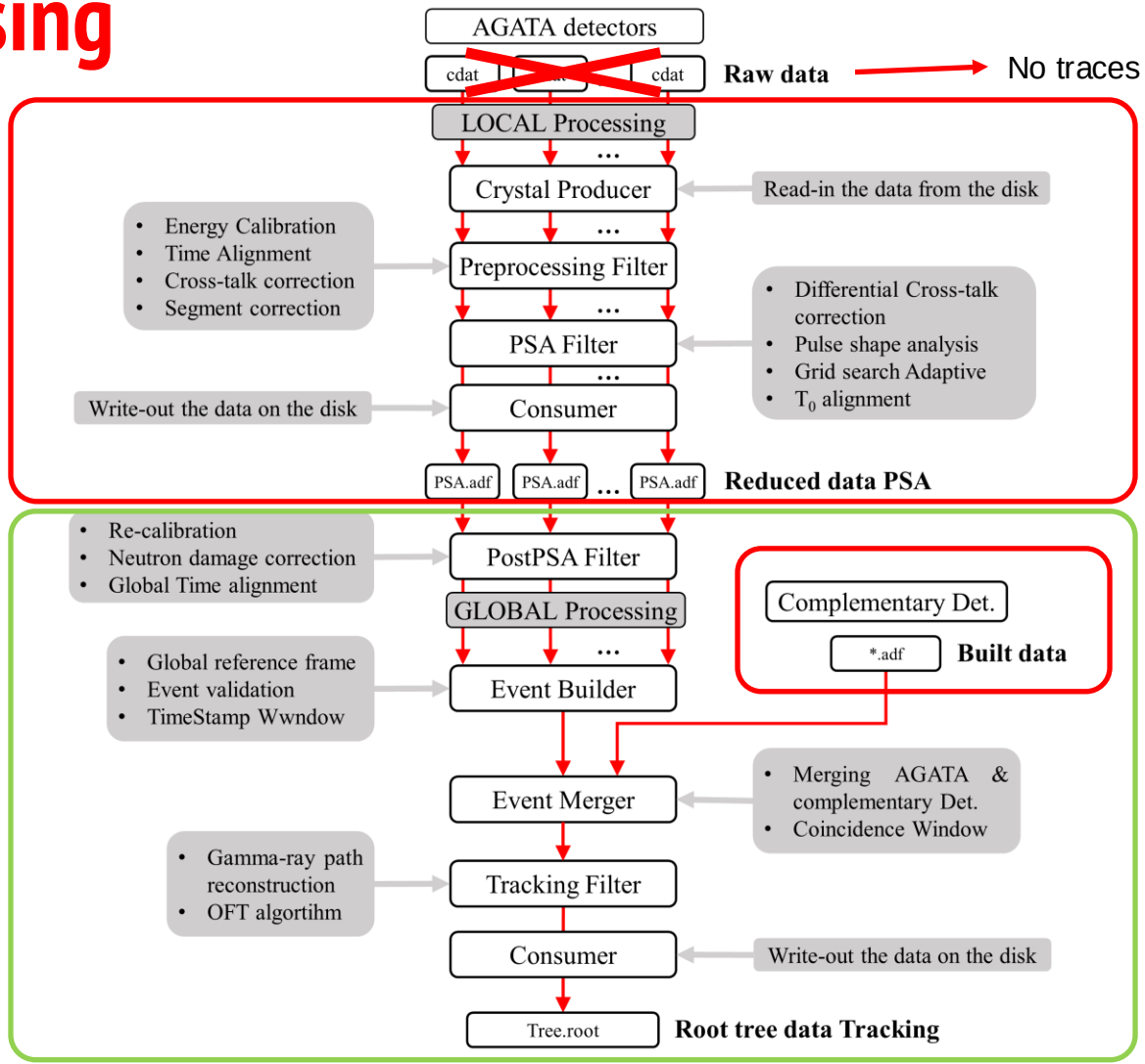
# Data Processing

## Narval actors



# Data Processing

## Narval actors

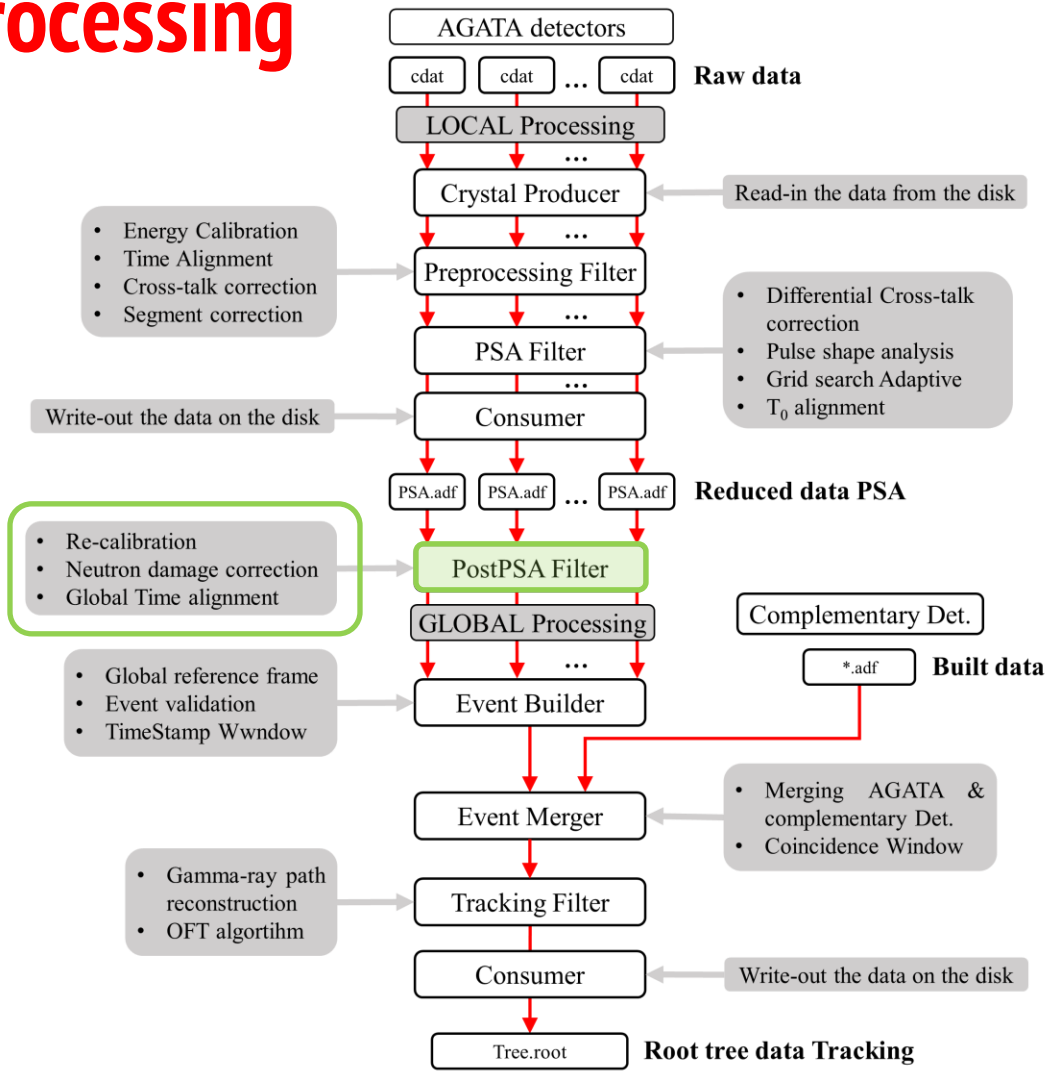



**Online**
Configuration prepared before the experiment

**Nearline/Offline**
Configuration prepared after the experiment

# Local Level Processing

## Narval actors



# Local Level Processing

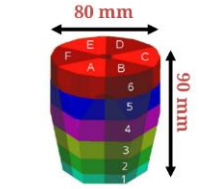
## PostPSA Filter

- Performs:
  - Recovery (partial) of neutron damage using info from PSA: **Trapping.cal**
  - Realibration over time (core): **TimeEvoCC.conf**
  - Final energy calibrations with offset: **RecalEnergy2.cal, gen_conf.py**
  - Force segments to core (optional) : **gen_conf.py**
  - Global Time alignment: **gen_conf.py**
- **Configuration for this actor can be done by users during/after the experiment**
- Generates one file:
  - Post__5-40-16384-UI__Ener.spec → **File in Out/00A e.g.**

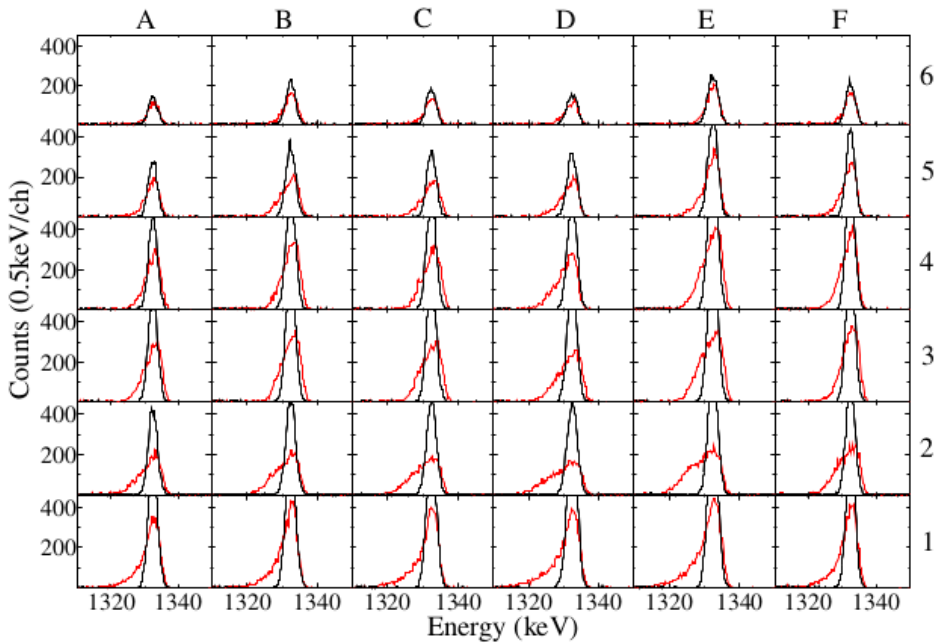
**Files in Conf/00A e.g. gen_conf.py its outside the configuration directory**

# PostPSA Filter

## 1. Neutron Damage correction




- **Fast neutrons** are well known to produce specific lattice defects in germanium crystals which act as **efficient hole traps.**
- Reduction in the charge collection efficiency of the detectors observable by **a low energy tailing** on the energy line shape
- AGATA crystals are n-type HPGe detectors, which are found to be less sensitive to the neutron radiation in terms of central contact signals
- The energy deficit can be **corrected using the position information obtained by the PSA**




*SortPsaHits* program applies to the PSA hits a grid of **correction parameters for the electron and hole trapping**, determining the optimum set parameters for all the detector channels **that minimizes the FWHM and the tail on the left** side of the energy peaks in the spectra.

# PostPSA Filter

## 1. Neutron Damage correction

Replay to generate Psa__0-16-F__Hits.fdat files
**femul key in PSAFilter:** `"WritePSAHits",`

**What is needed:**

- Long 60Co run
- PSA hits file : Data/{crystalID}/**Psa__0-16-F__Hits.fdat**
- Conf File**: Trapping.cal, gen_conf.py**
- Auxiliary files: Pso__2-4-40-2048-UI__Ener.spec
- Programs/scripts:

**femul key in PostPSAFilter:**
`"Trapping Trapping.cal",`
Add it in gen_conf.py

**SortPsaHits:** get the optimum lambdaE lambadaH parameters and generates Pso__2-4-40-2048-UI__Ener.spec

```
SortPsaHits  -f  Psa__0-16-F__Hits.fdat  -best 1300 1350  -bpar 1 10000 0
SortPsaHits -f ../Data/{crystalID}/Psa__0-16-F__Hits.fdat -gain 5 -offs 5000 -fcal Trapping.cal
```

**RecalEnergy:** generate calibration coefficients for the different columms of Trapping.conf

```
RecalEnergy -spe Pso__2-4-40-2048-UI__Ener.spec -num 40 -sub 0 -offs -5000 -gain 5 -noTR -dwa 25 2 | tee Recal_SG_orig.txt
```

**colupdate.py:** add these coefficients to the different columns of Trapping.conf

```
./colupdate.py -c 1 13 Trapping.cal Recal_SG_orig.txt -o Trapping.cal
```

3 steps process

1. RecalEnergy, SortPSAHits (Check) [Optional]
2. SortPSAHits (Check)
3. RecalEnergy, SortPSAHits (Check)

The Trapping.cal file has 36 lines, one per segment:

```
#SG gainSG_orig gainCC_orig lambdaE lambdaH gainSG_corr gainCC_corr
 0          1.          1.   51.6     6.6          1.          1.
 1          1.          1.  269.3     6.6          1.          1.
 2          1.          1.   51.6     6.6          1.          1.
...
 35         1.          1.  104.4     8.9          1.          1.
```

More details in AGATA_LLP_UsersGuide

# PostPSA Filter

## 1. Neutron Damage correction

Replay to generate Psa__0-16-F__Hits.fdat files
**femul key in PSAFilter:** "WritePSAHits",

**What is needed:**

- Long 60Co run
- PSA hits file : Data/{crystalID}/**Psa__0-16-F__Hits.fdat**
- Conf File**: Trapping.cal, gen_conf.py**
- Auxiliary files: Pso__2-4-40-2048-UI__Ener.spec
- Programs/scripts:

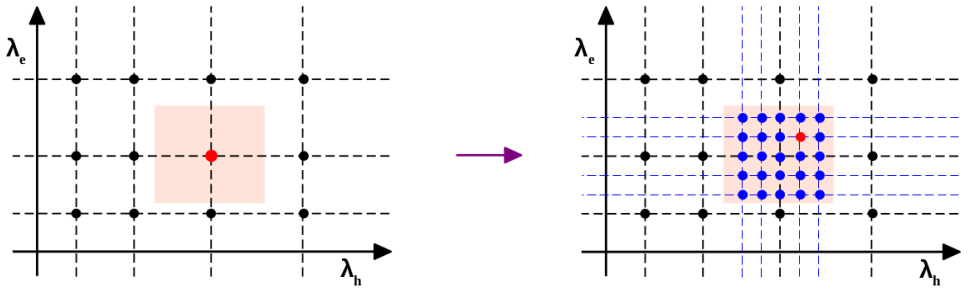
**femul key in PostPSAFilter:**
"Trapping Trapping.cal",
Add it in gen_conf.py

**NEW!**

**Neutron damage correction improvements:**

- For SG+CC, we employ a normalization of the whole grid of FOMs (such that the highest FOM has value 1) before calculating the average
- Implementation of an Adaptive Grid – Search

**"-size" option:** variable grid size, default is kept at nH=100, nE=50
**"-algo" option:** user must choose number of iterations and which spectra to optimize (0=SG, 1=CC or 2=SG+CC). Default values are 1 2
**"zoom" option:** user can specify the magnification factor M between iterations. Default is 0.25




Elia Pilotto Agata Week 2024

Example of command:

```
SortPsaHits -f Psa__0-16-F__Hits.fdat -best 1300 1350 -bpar 1 10000 0 -size 50 30 -algo 3 2
```

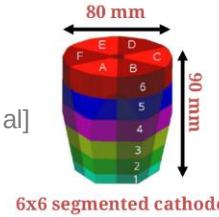
More details in AGATA_LLP_UsersGuide

# PostPSA Filter

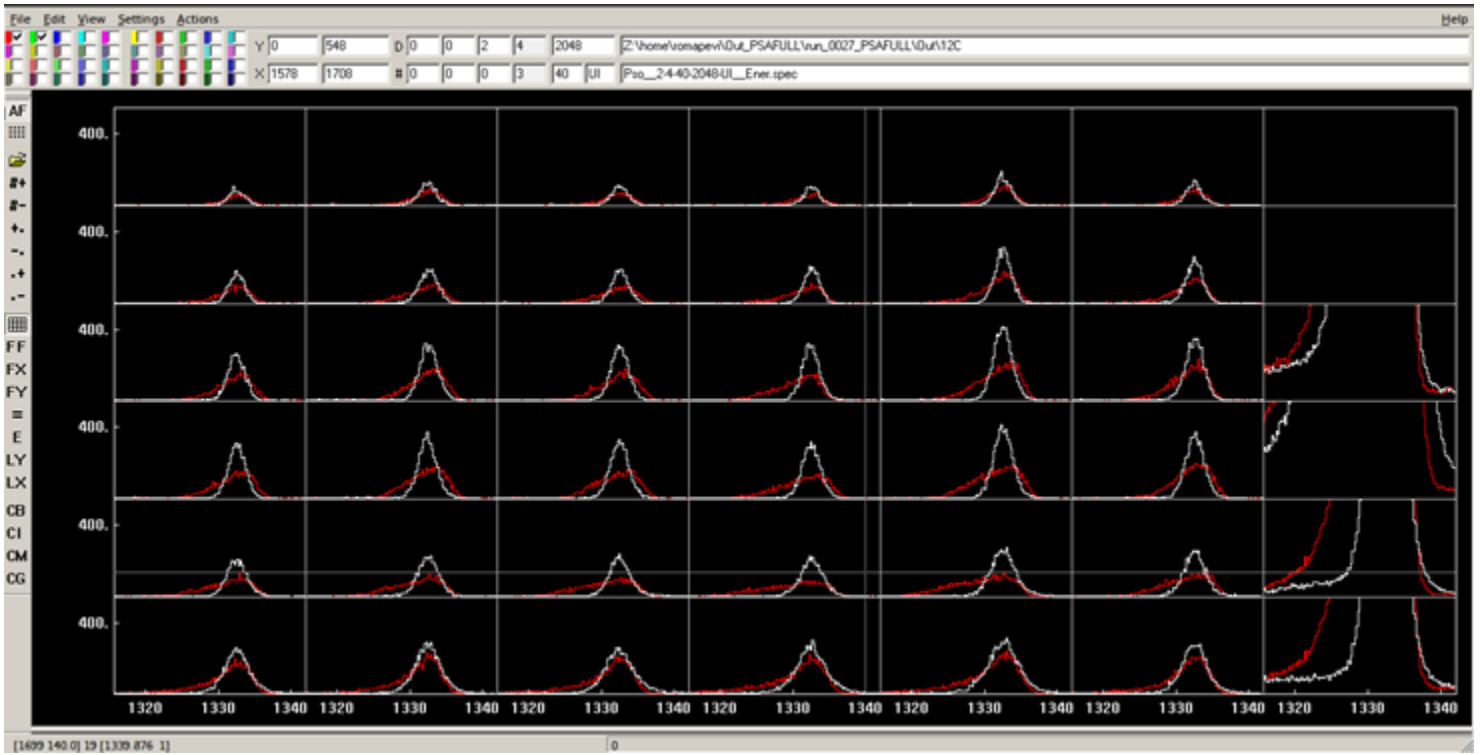
## 1. Neutron Damage correction

3 steps process

1. RecalEnergy, SortPSAHits (Check) [Optional]
2. SortPSAHits (Check)
3. RecalEnergy, SortPSAHits (Check)




Pso__2-4-40-2048-UI__Ener.spec [0][1][all] before (red) and [0][3][all] after (white) the neutron correction




[0-SG,1-CC]
[0-orig,1-orig+recal,2-corr,3-corr+recal]
[0-39]:

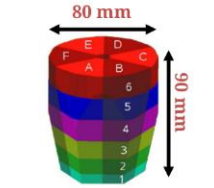
- 0-35 segments
- 36 Or of segments/cores
- 37 Or of segments/cores M=1
- 38 Sum of segments/cores
- 39 Average SumSegs+SumCC

| #SG | gainSG_orig | gainCC_orig | lambdaE | lambdaH | gainSG_corr | gainCC_corr |
|---|---|---|---|---|---|---|
| 0 | 1. | 1. | 51.6 | 6.6 | 1. | 1. |
| 1 | 1. | 1. | 269.3 | 6.6 | 1. | 1. |
| 2 | 1. | 1. | 51.6 | 6.6 | 1. | 1. |
| ... | | | | | | |
| 35 | 1. | 1. | 104.4 | 8.9 | 1. | 1. |

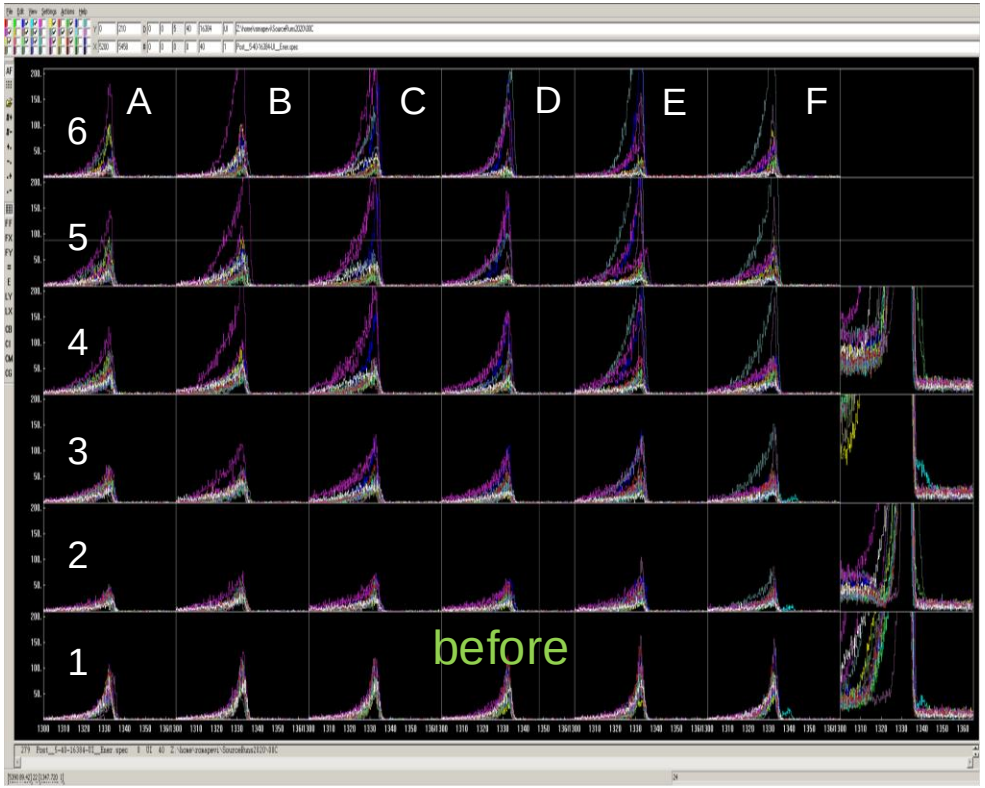
# PostPSA Filter

## 1. Neutron Damage correction

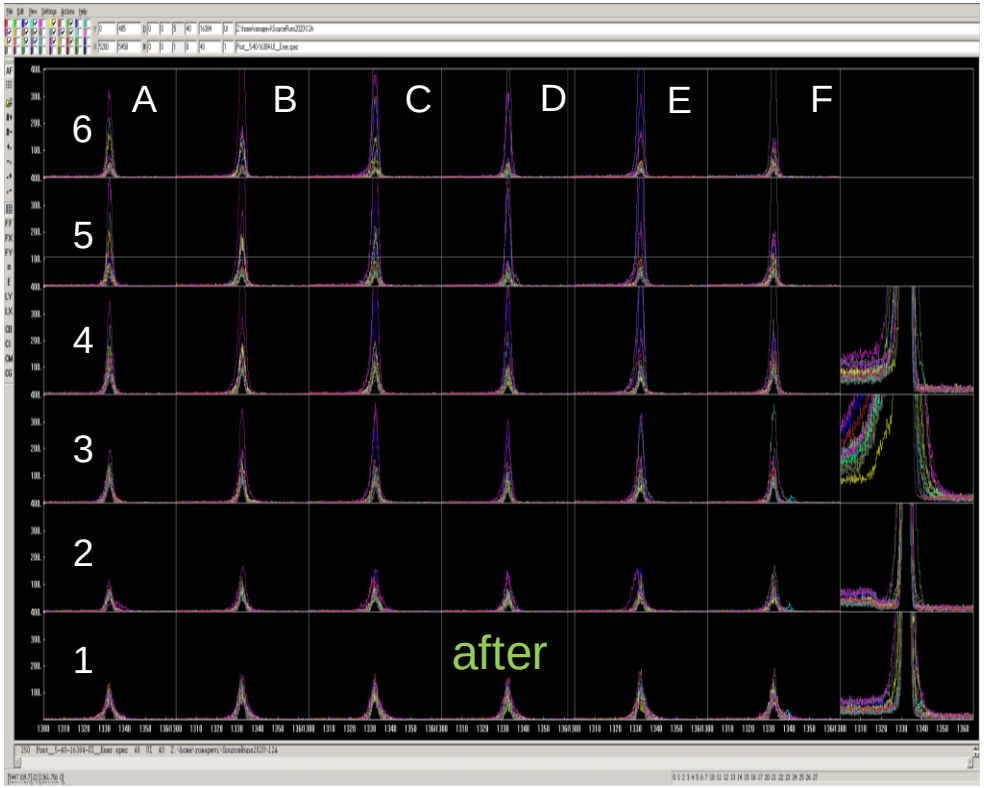
Verification with femul replay




Post__5-40-16384-UI__Ener.spec [0] [0-39]




Post__5-40-16384-UI__Ener.spec [1] [0-39]



# PostPSA Filter

## 2. Recalibration

**The recalibration of segments and cores is a several steps process and one must be careful of the possible redundant calibration done by the PostPSA filter actor:**

0) Recalibration of Segments from file RecalEnergy1

1) Recalibration of Segments and Cores from the Trapping file

**Filling histograms libraries 0 to 3 in PostSpecEner**

2) Correction for energy drifting over time (Core energy)

3) Recalibration of Segments from file RecalEnergy2

4) Core recalibration from RecalCC and Segments recalibration from RecalSG

6) Force segments to core

**Filling histograms library 4 in PostSpecEner**

More details in AGATA_LLP_UsersGuide

# PostPSA Filter

## 2. Recalibration

**The recalibration of segments and cores is a several steps process and one must be careful of the possible redundant calibration done by the PostPSA filter actor:**

~~Recalibration of Segments from file RecalEnergy1~~ ➡ Typically not used

~~Recalibration of Segments and Cores from the Trapping file~~ ➡ performed in the neutron damage correction

**Filling histograms libraries 0 to 3 in PostSpecEner**

0) Correction for energy drifting over time (Core energy) **NEW!**

1) Recalibration of Segments from file RecalEnergy2

2) Core recalibration from RecalCC and Segments recalibration from RecalSG

3) Force segments to core

**Filling histograms library 4 in PostSpecEner**

More details in AGATA_LLP_UsersGuide

# PostPSA Filter

Replay to generate ROOT files
**femul Topology_Global_Tree.conf**

## 2.0 Recalibration: Correction for energy drifting over time

**What is needed:**

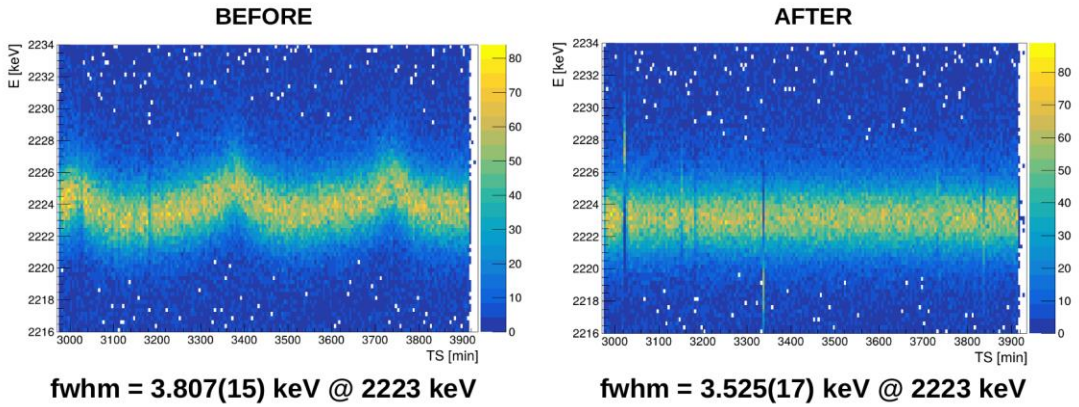
- Long run with high energy transitions
- Root file : Out/Analysis/**Tree_0000.root**
- Conf File**: TimeEvoCC.conf, gen_conf.py**
- Programs/scripts:

**NEW!**

### Cross – correlation Correction Method (CCM)

[Matus Balogh NIM paper]
[GitHub code]

- Create a matrix Core energy versus time and divide in slices
- For each a projection, calculate product between a reference spectrum and the shifted projection
- Perform a scan for each slice




- It has been observed that especially when using the lower energy gain of the core contact gain oscillations of the order of 1/1000 can occur
- Pseudo – periodic with 6h period (AGATA filling)
- More pronounced in detectors placed in the lower part of Agata, pointing upwards
- To be corrected before the recalibration with offset of the core (RecalCC)

The TimeEvoCC.conf file must contain three columns with the following format:

```
#TS_start            TS_stop             gain
6150000000000        6210000000000       0.999948755522396
6210000000000        6270000000000       0.999887585155303
...
```

More details in [AGATA_LLP_UsersGuide] and [Elia Pilotto Agata Week 2024]

**femul key in PostPSAFilter:**
"TimeEvoCCFile TimeEvoCC.conf","",
Add it in gen_conf.py

# PostPSA Filter

## 2.1 Recalibration: RecalEnergy2

3 steps process

Verification with femul replay each step

1. RecalEnergy2 (Check)
2. RecalCC and RecalSG (Check)
3. ForceSegmentstoCore (Check)[Optional]

**What is needed:**

- Long 152Eu run
- PostPSA file : Out/{crystalID}/**Post__5-40-16384-UI__Ener.spec**
- Conf File**: RecalEnergy2.cal, gen_conf.py**
- Auxiliary files: Recal2.txt
- Programs/scripts:

**RecalEnergy:** generate calibration coefficients

- If Trapping.cal is **not** applied:

```
RecalEnergy -spe Out_norecal/{crystalID}/Post__5-40-16384-UI__Ener.spec -sub 0 -num 36 -gain 4 -poly1 -152Eu
```

- If Trapping.cal **is applied**:

```
RecalEnergy -spe Out_norecal/{crystalID}/Post__5-40-16384-UI__Ener.spec -sub 40 -num 36 -gain 4 -poly1 -152Eu
```

| # indx | #spec | #pks | #ok | rEnergy | FW05 | FW01 | Area | Position | Width | Ampli | WTML | WTMR | slope*gain | rChi2% | offs1*g | slope1*g | rChi2% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 40 | 32 | 10 | 1408.15 | 5.032 | 12.662 | 4294 | 5624.78 | 16.5 | 171 | 4.320 | 1.823 | 1.001387 | 1.74 | 0.216 | 1.001157 | 0.42 |
| 1 | 41 | 18 | 8 | 1409.43 | 9.080 | 18.768 | 2425 | 5614.94 | 36.0 | 59 | 2.329 | 1.823 | 1.004054 | 999.99 | 7.040 | 0.996854 | 91.65 |
| 2 | 42 | 24 | 10 | 1407.05 | 7.637 | 17.111 | 3068 | 5611.22 | 29.1 | 85 | 2.861 | 1.823 | 1.003023 | 52.70 | -0.732 | 1.003805 | 41.66 |
| 3 | 43 | 25 | 9 | 1407.38 | 5.256 | 13.774 | 3854 | 5624.43 | 15.7 | 144 | 5.210 | 1.823 | 1.000901 | 29.62 | -1.073 | 1.001038 | 2.44 |
| 4 | 44 | 20 | 9 | 1408.21 | 4.658 | 12.739 | 2799 | 5621.39 | 12.1 | 116 | 6.562 | 1.823 | 1.002036 | 6.53 | 0.213 | 1.001812 | 6.24 |
| 5 | 45 | 14 | 9 | 1408.15 | 4.126 | 10.131 | 1331 | 5623.55 | 14.1 | 65 | 3.909 | 1.823 | 1.001609 | 37.60 | -0.389 | 1.002013 | 38.46 |
| 6 | 46 | 29 | 10 | 1407.69 | 5.470 | 13.087 | 4393 | 5626.02 | 19.5 | 165 | 3.546 | 1.823 | 1.000845 | 9.79 | 0.293 | 1.000532 | 8.18 |
| 7 | 47 | 22 | 10 | 1408.45 | 6.756 | 17.530 | 2612 | 5619.43 | 20.6 | 76 | 4.964 | 1.823 | 1.002556 | 21.86 | 0.213 | 1.002329 | 23.10 |
| 8 | 48 | 22 | 10 | 1406.20 | 7.643 | 17.149 | 3859 | 5615.23 | 29.2 | 107 | 2.873 | 1.823 | 1.001704 | 58.23 | -0.356 | 1.002085 | 61.32 |
| 9 | 49 | 27 | 5 | 1408.70 | 7.257 | 18.956 | 4738 | 3839.23 | 17.0 | 186 | 3.888 | 2.197 | 1.467693 | 919.64 | 3.908 | 1.461898 | 601.93 |

The RecalEnergy2.cal file has 36 lines, one per segment:

```
#segm %d(id)      %f(offset)    %f(egain)
 segm     0         0.216        1.001157
 segm     1         7.040        0.996854
 segm     2        -0.732        1.003805
 ...
 segm    35        -1.083        1.002038
```

**colupdate.py:** add these coefficients to the 3rd and 4th columns of RecalEnergy2.cal

```
./colupdate.py -c 2 14 RecalEnergy2.cal Recal2.txt -o RecalEnergy2.cal
./colupdate.py -c 3 15 RecalEnergy2.cal Recal2.txt -o RecalEnergy2.cal
```

More details in AGATA_LLP_UsersGuide

**femul key in PostPSAFilter:**
"RecalEnergy2 RecalEnergy2.cal",
Add it in gen_conf.py

# PostPSA Filter

## 2.2 Recalibration: RecalCC & RecalSG

3 steps process

Verification with femul replay each step

1. RecalEnergy2 (Check)
2. RecalCC and RecalSG (Check)
3. ForceSegmentstoCore (Check)[Optional]

**What is needed:**

- Long 152Eu run
- PostPSA file : Out/{crystalID}/**Post__5-40-16384-UI__Ener.spec**
- Conf File**:** **gen_conf.py**
- Auxiliary files: Recal2.txt
- Programs/scripts:

**RecalEnergy:** generate calibration coefficients

- For the **core** recalibration with offset

```
RecalEnergy -spe Out_norecal/{crystalID}/Post__5-40-16384-UI__Ener.spec -sub 159 -num 1 -gain 4 -poly1 -152Eu

# indx  #spec #pks #ok   rEnergy     FW05     FW01      Area  Position  Width   Ampli    WTML    WTMR  slope*gain rChi2%   offs1*g   slope1*g rChi2%
#
     0    159   90  10   1408.00    3.048    6.491    214459   5629.83   12.1   15148   2.432   1.869    1.000385   0.32    -0.050   1.000438  0.28
```

- For the s**um of segments** recalibration with ofset **(one gain and offset for all the segments, optional)**

```
RecalEnergy -spe Out_norecal/{crystalID}/Post__5-40-16384-UI__Ener.spec -sub 78 -num 1 -gain 4 -poly1 -152Eu

# indx  #spec #pks #ok   rEnergy     FW05     FW01      Area  Position  Width   Ampli    WTML    WTMR  slope*gain rChi2%   offs1*g   slope1*g rChi2%
#
     0     78   67  10   1408.17    5.674   14.718    179916   5639.15   20.0    6174   2.245   3.643    0.998854   7.87     0.454   0.998371  2.04
```

Add these coefficients to gen_conf.py

**femul key in PostPSAFilter:**

```
"RecalCC offset egain",
"RecalSG offset egain",
```

More details in AGATA_LLP_UsersGuide

# PostPSA Filter

## 2.3 Recalibration: Force Segments to core

3 steps process

Verification with femul replay each step

1. RecalEnergy2 (Check)
2. RecalCC and RecalSG (Check)
3. ForceSegmentstoCore (Check)[Optional]

- Energy of segments scaled in such a way that their sum equals Energy of Core
- Optional/Experiment dependent

**What is needed:**

- Source and In-beam data
- PostPSA file and Track files : Out/{crystalID}/**Post__5-40-16384-UI__Ener.spec**
  Out/Merger/**Track__2-24-16384-UI__EC.spec**
  Out/Merger/**Track__2-15-16384-UI__EE.spec**
- Conf File**: gen_conf.py**

**femul key in PostPSAFilter:**
`"ForceSegmentstoCore",`
Add it in gen_conf.py

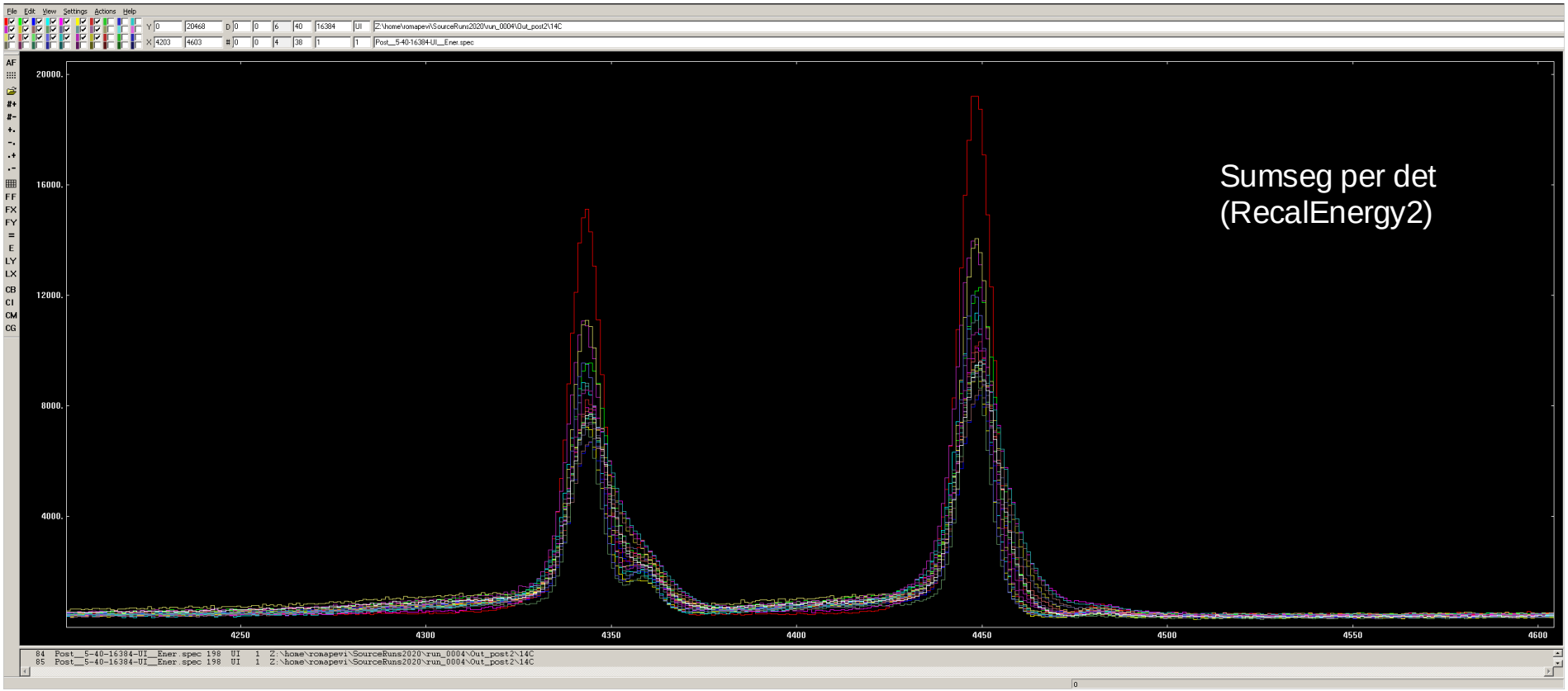
More details in AGATA_LLP_UsersGuide

# PostPSA Filter

Verification with femul replay each step

## 2. Recalibration

Post__5-40-16384-UI__Ener.spec [4] [38]

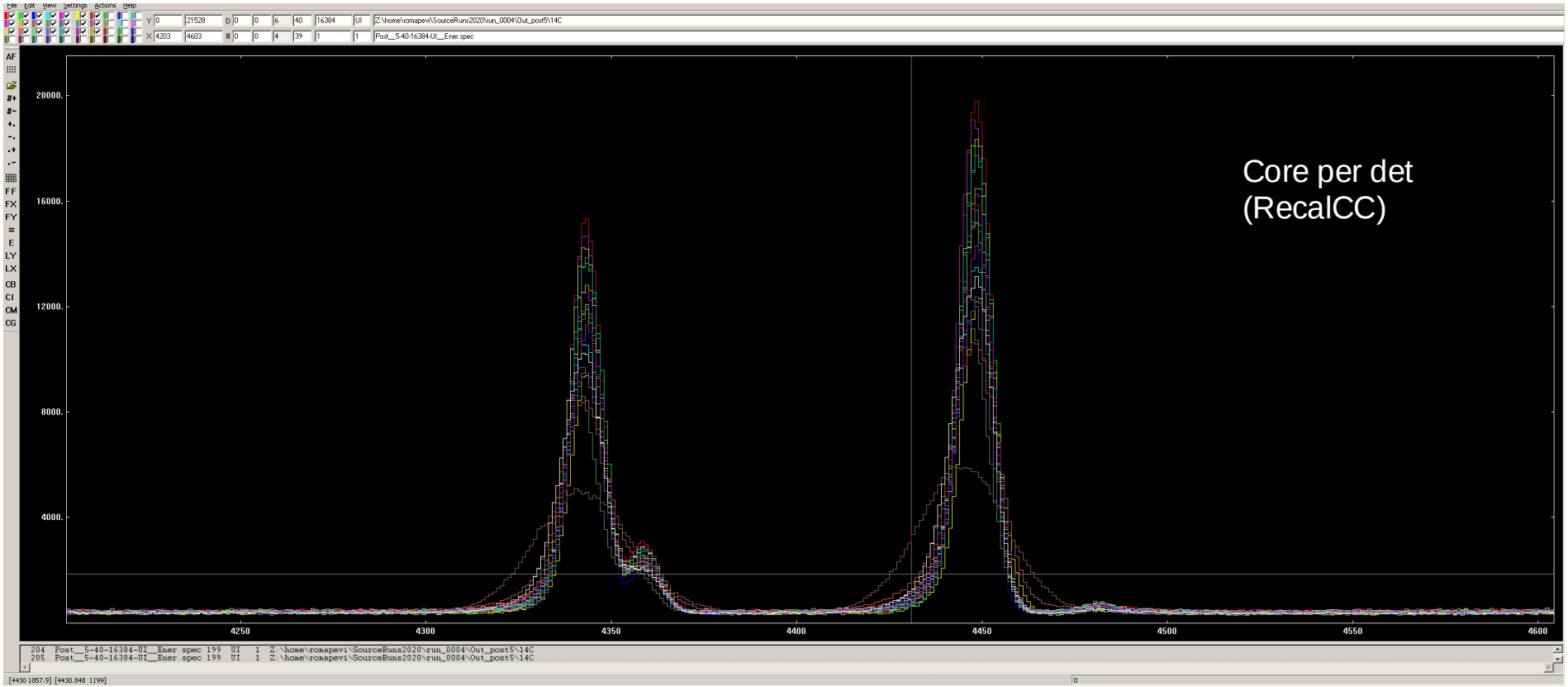


# PostPSA Filter

Verification with femul replay each step

## 2. Recalibration

Post__5-40-16384-UI__Ener.spec [4] [39]

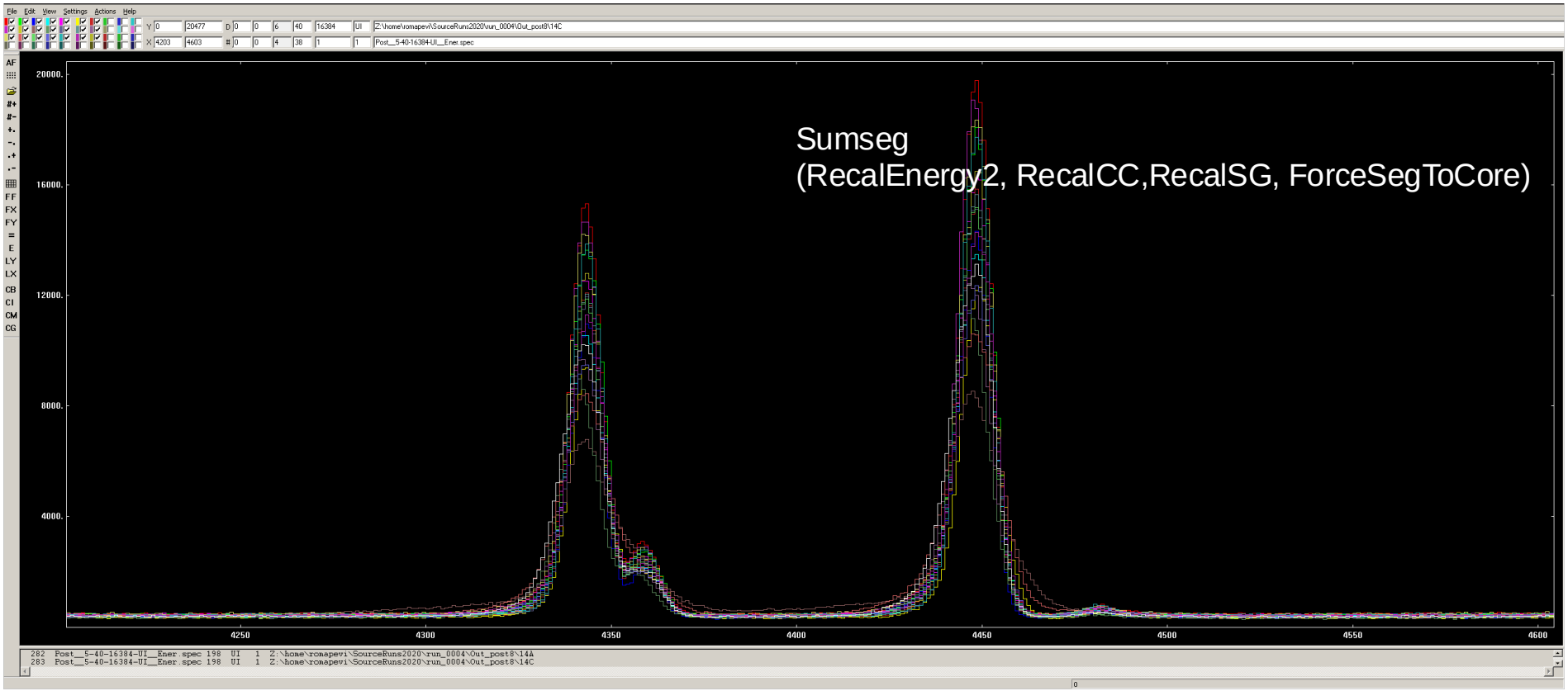


# PostPSA Filter

Verification with femul replay each step

## 2. Recalibration

Post__5-40-16384-UI__Ener.spec [4] [38]

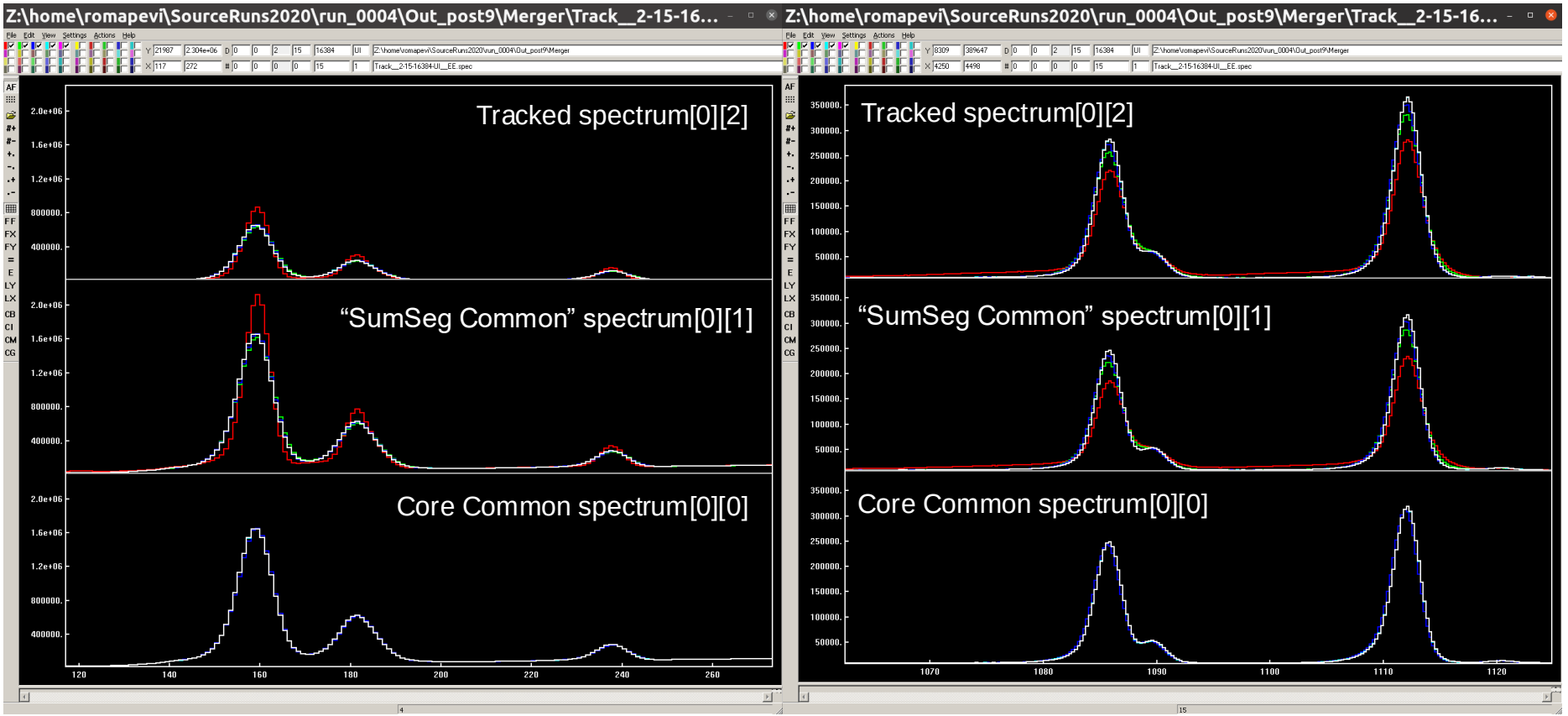


# PostPSA Filter

## 2. Recalibration

No ForceSeg, RecalEnergy2
ForceSeg, RecalEnergy2, RecalCC RecalSG
ForceSeg, RecalEnergy2 no offset, RecalCC RecalSG

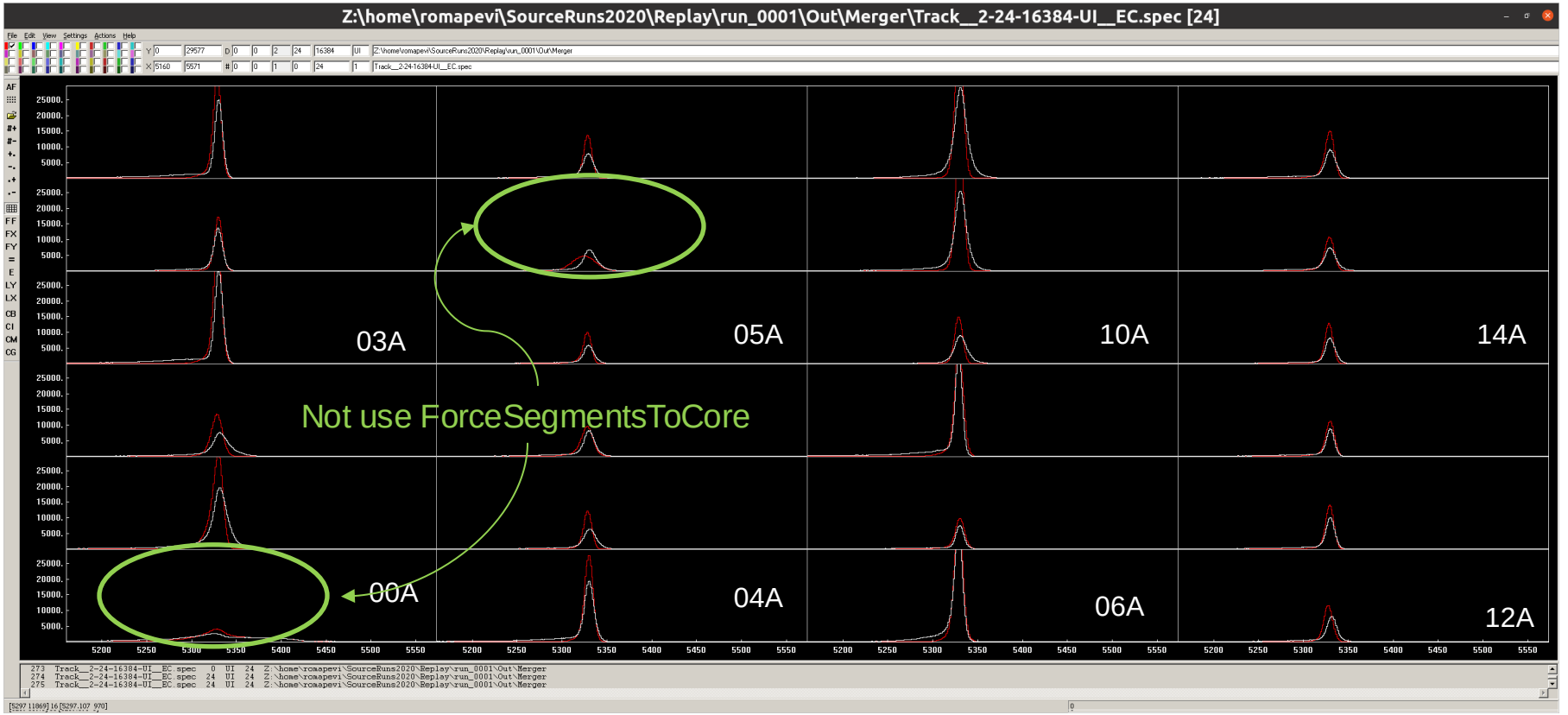
Track__2-15-16384-UI__EE.spec [0][0-2]



# PostPSA Filter

## 2. Recalibration

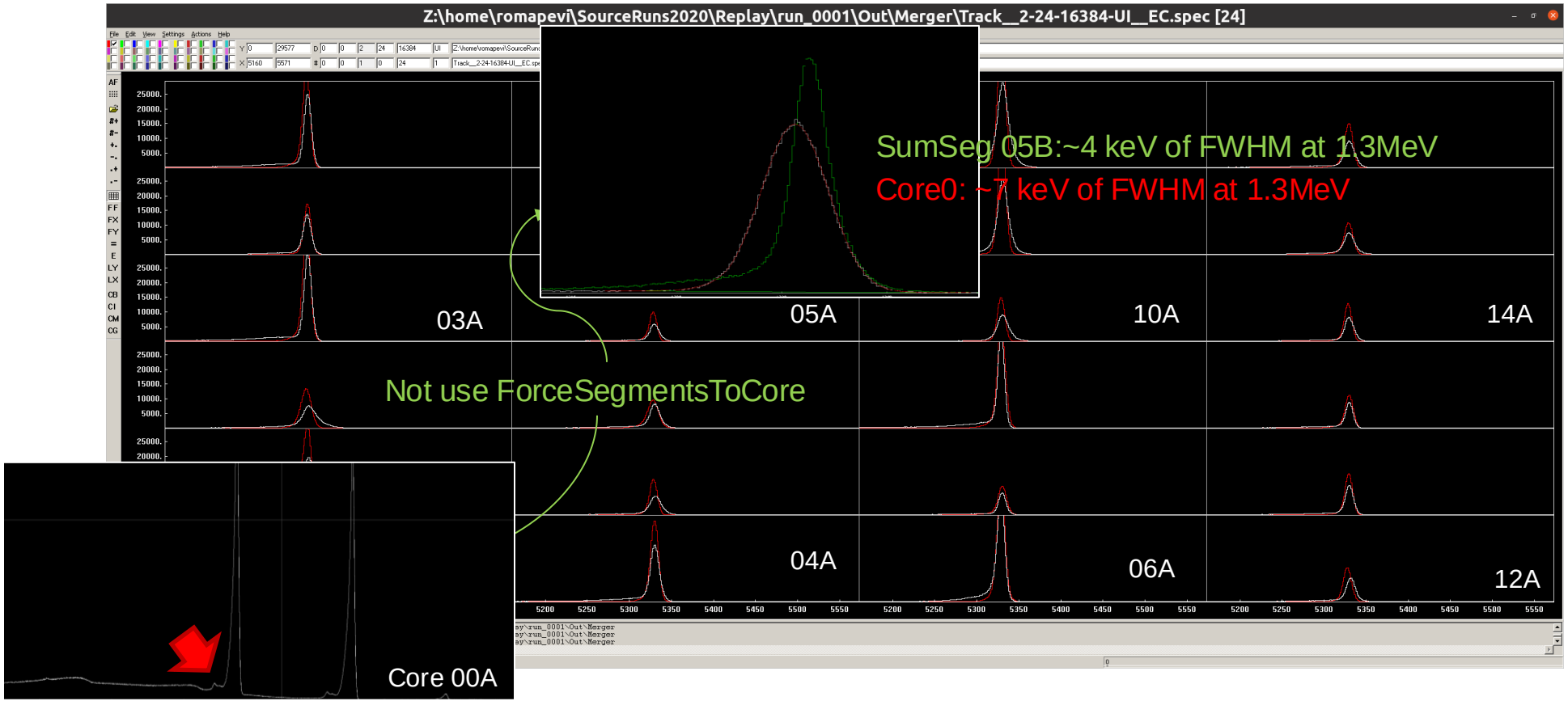
Track__2-24-16384-UI__EC.spec



# PostPSA Filter

## 2. Recalibration

Track__2-24-16384-UI__EC.spec



# PostPSA Filter

## 3. Global time alignment

To be done every time that there is a GTS alignment

2 steps process

1. Replay without coeff
2. Replay with coeff

**What is needed:**

- Any run
- Spectra file : Data/Merger/**Track__35-35-1000-UI__TT.spec**
- Conf File**: gen_conf.py**
- Auxiliary files: recalT.dat
- Programs/scripts:

**RecalEnergy:** generate shift coefficients for the N*N time spectra

```
RecalEnergy -spe ../Out/Merger/Track__${N}-${N}-1000-UI__TT.spec -T 500 -num ${NN} | tee recalT.dat
```

```
# indx  #spec #pks #ok   rEnergy     FW05     FW01      Area  Position  Width   Ampli    WTML    WTMR  shift*gain
#
    0      0    1   1    0.00       0.000    0.000        0     0.00      0.0     0      0.000   0.000       0.000
    1      1    1   1    503.37    19.629   44.382     9422     500.85   19.5     400    2.382   2.171      -0.850
    2      2    1   1    501.04    16.180   36.094    11042     500.47   15.8     574    2.604   1.955      -0.473
    3      3    1   1    502.67    16.667   36.949    10726     499.84   16.4     543    2.558   1.958       0.164
    4      4    1   1    503.28    17.366   40.384     8202     499.65   17.1     388    2.484   2.231       0.350
   ...
```

**solveTT.py:**find the best shift combination.

```
${pathSoftware}/solveTT.py -f recalT.dat -n ${N} -c 13 -p 500
```

```
Shifts that minimize Chi2
0.001
-0.181
0.004
-0.087
0.040
-0.239
...
```

**femul key in PostPSAFilter:**
"TimeShiftCC coeff",

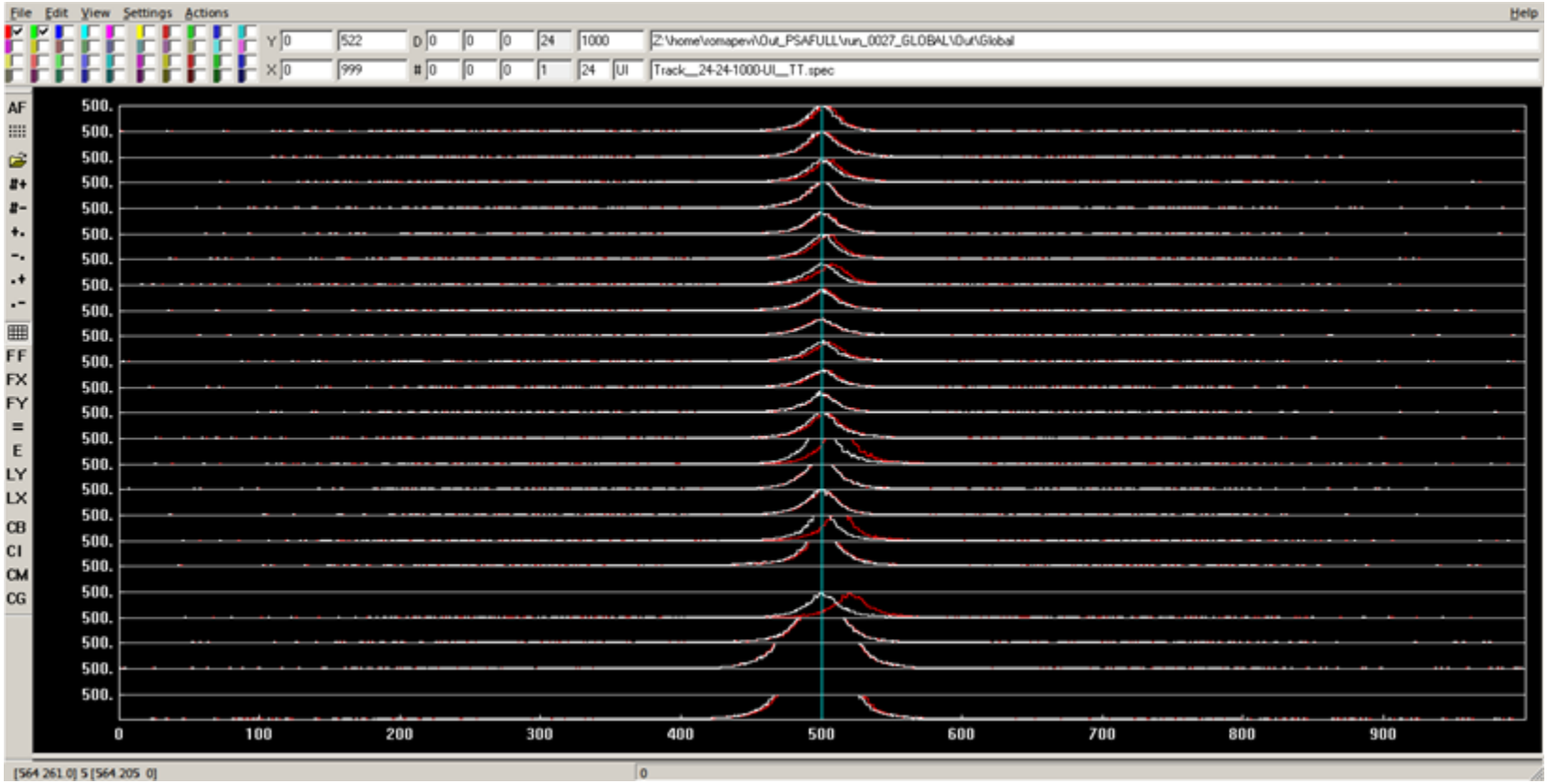
More details in AGATA_LLP_UsersGuide

# PostPSA Filter

Verification with femul replay

## 3. Global Time Alignment

Track__24-24-1000-UI__TT.spec example detector 00B [0][all] red before, white after time alignment

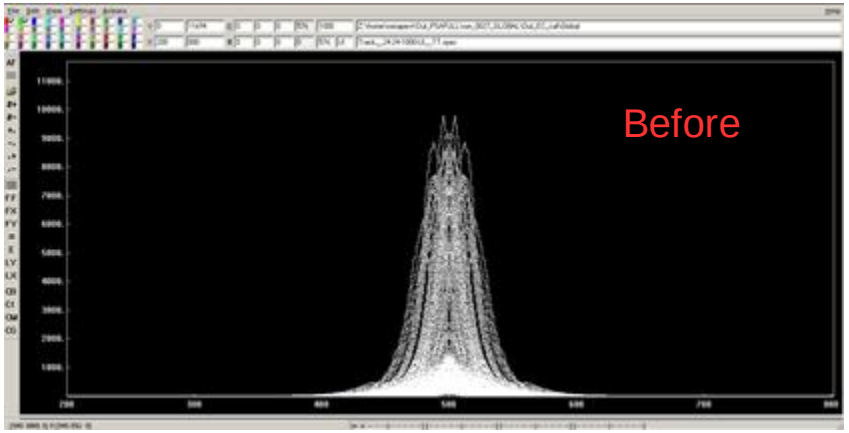


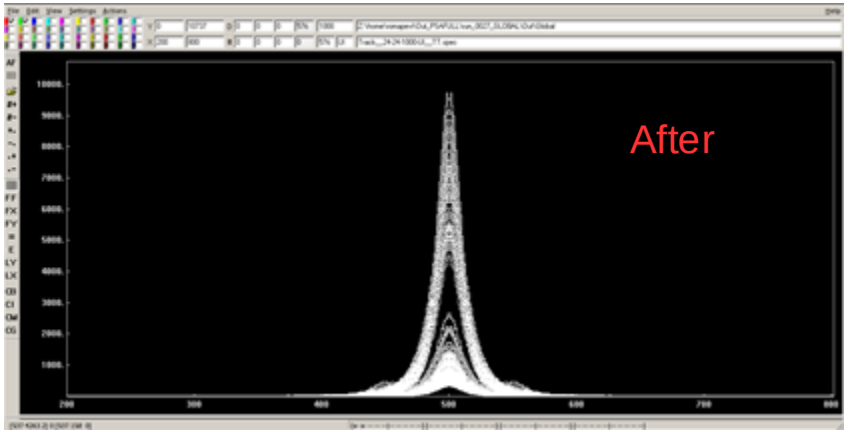
# PostPSA Filter

Verification with femul replay

## 3. Global Time Alignment

Track__24-24-1000-UI__TT.spec all [all][all]



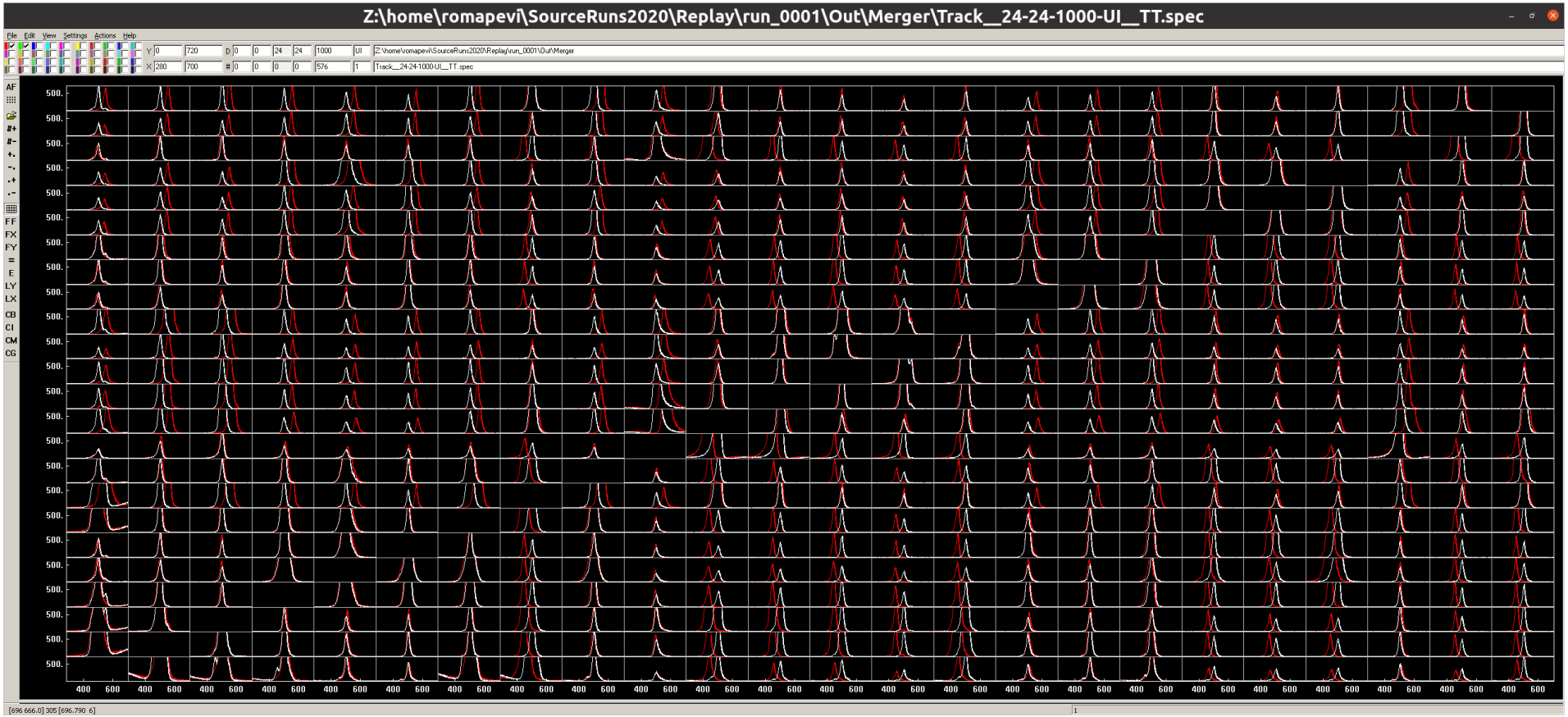



# PostPSA Filter

Verification with femul replay

## 3. Global Time Alignment

. Track__24-24-1000-UI__TT.spec all [all][all] red before, white after time alignment




00A 03A 04A 05A 06A 10A 12A 14A

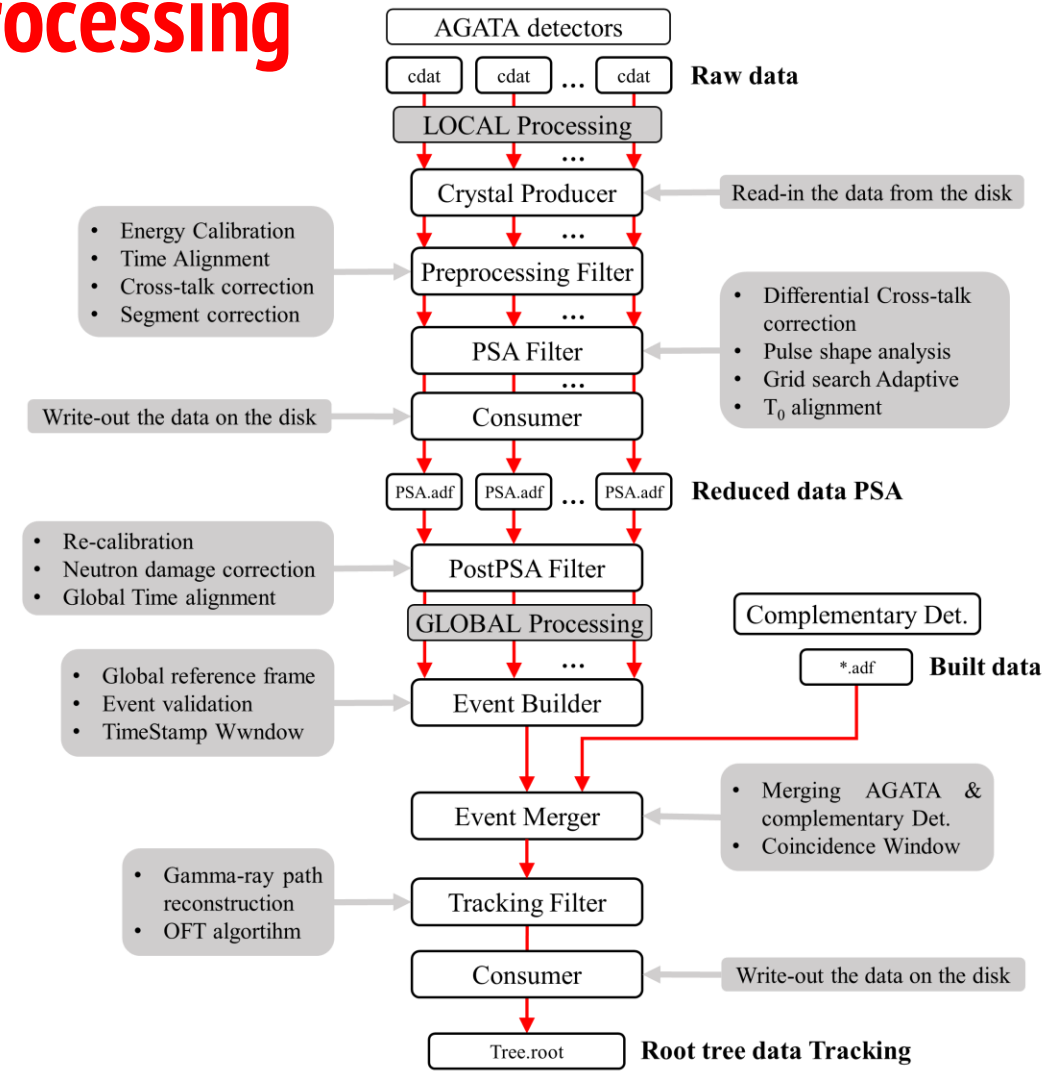
# Local Level Processing

## PostPSA Filter

- Follow the order given here.
- Be careful of the possible redundant calibration done by the PostPSA filter actor.
- The PostPSA is the last chance to have properly calibrated segments.
- The calibration offset can only be set at this level of the analysis.
- ForceSegToCore! final correction, only when the core resolution is good.
- Keep track of the GTS alignment for the Global Time alignment (important to reduce random coincidences)

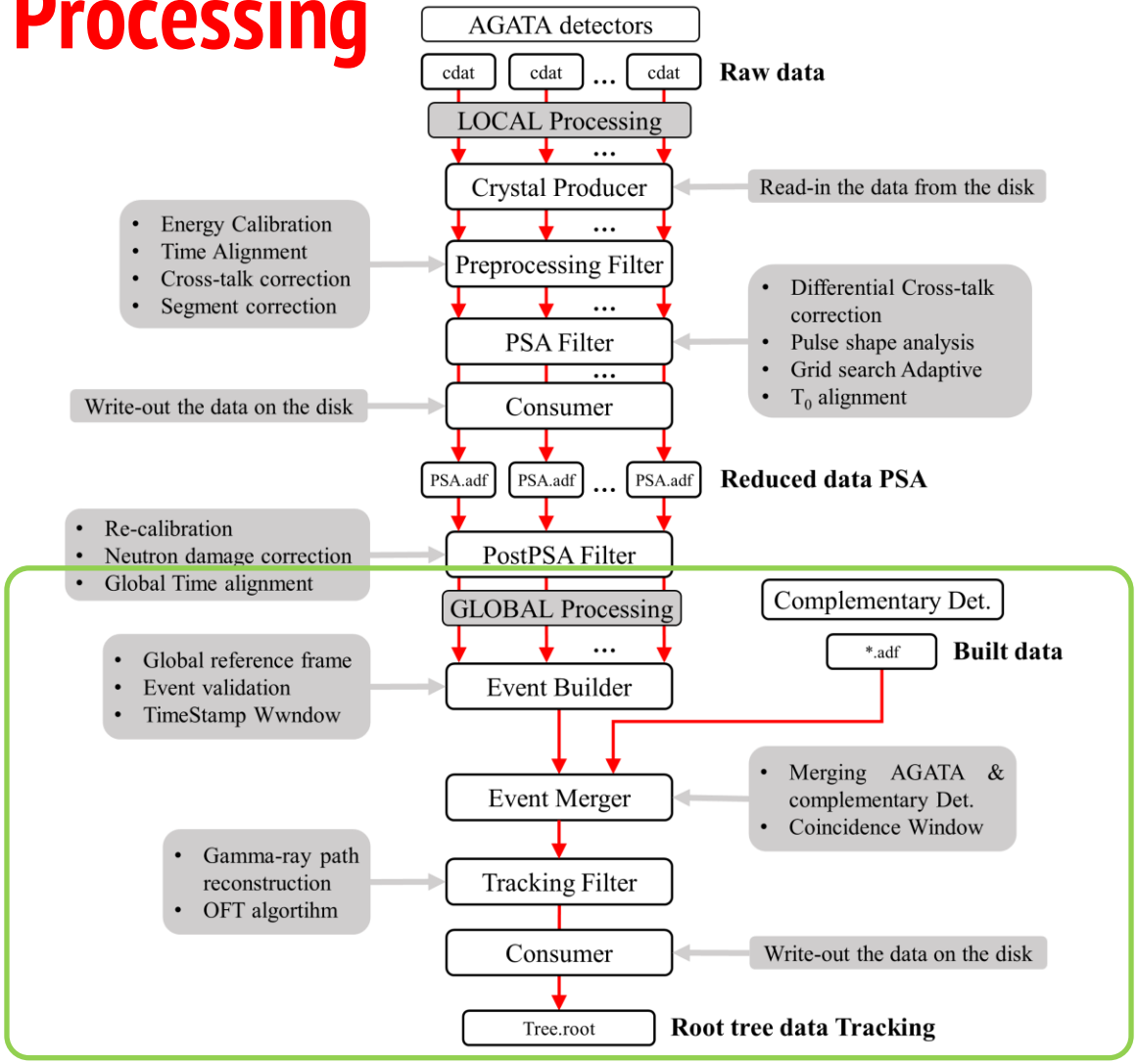
# Local Level Processing

## Narval actors



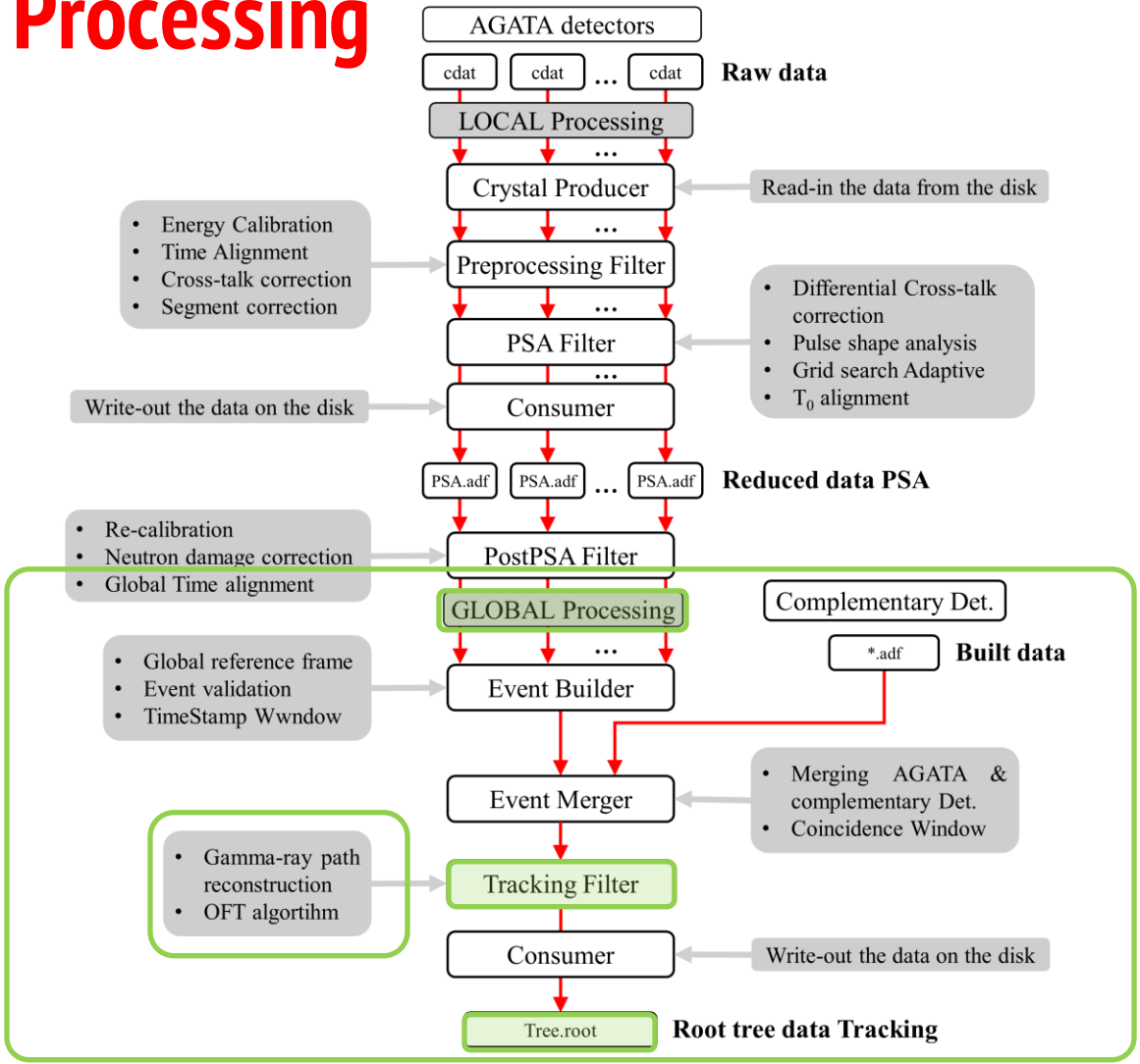
# Global Level Processing

## Narval actors



# Global Level Processing

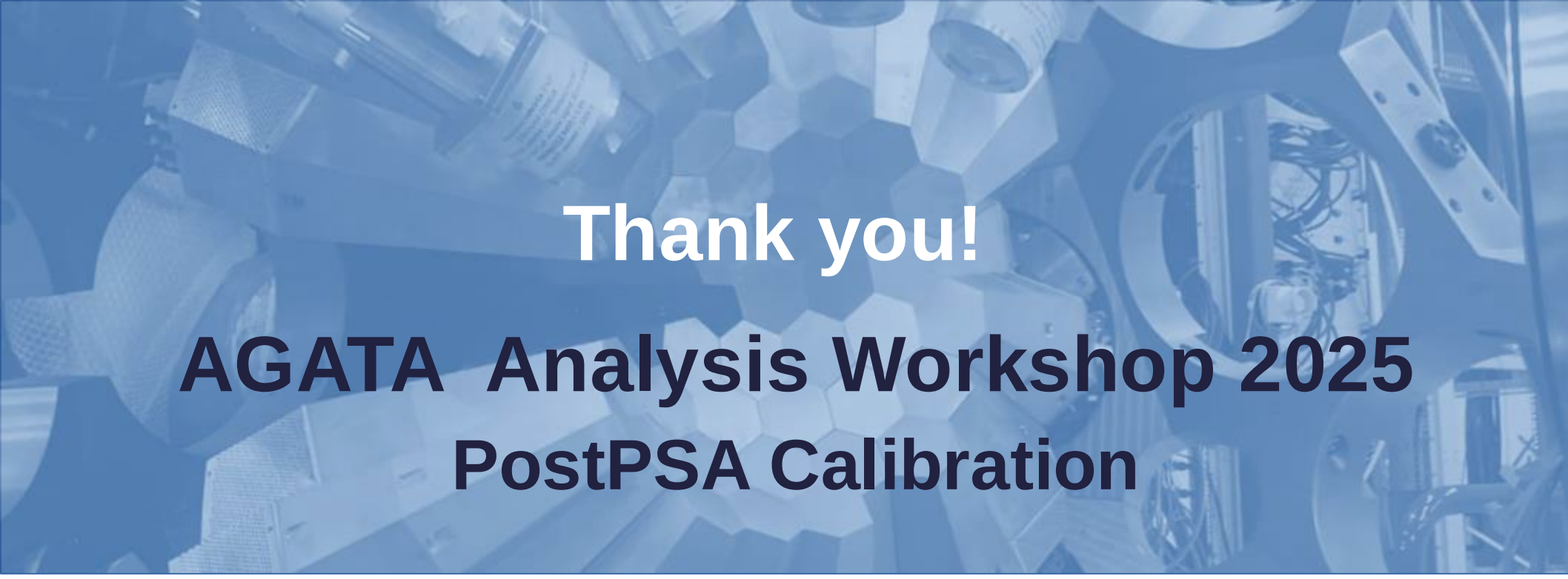
## Narval actors




**Tracking**
Speaker: **Araceli López Martens**
Friday

**Global data replay**
Speaker: **Jérémie Dudouet**
Wednesday

**Selector**
Speaker: **Daniele Brugnara**
Wednesday



# Thank you!

# AGATA Analysis Workshop 2025

## PostPSA Calibration

**R.M. Pérez-Vidal**

14/01/2025, Lyon

# Questions?

IFIC INSTITUT DE FÍSICA CORPUSCULAR
EXCELENCIA SEVERO OCHOA
VNIVERSITAT DE VALÈNCIA
CSIC CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS