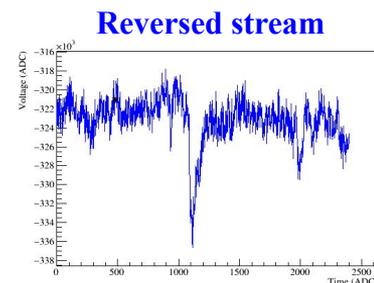# Towards a Kilopixel analysis

Matteo Cappelli,
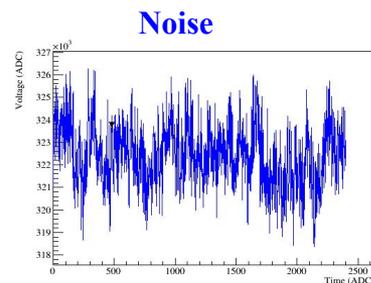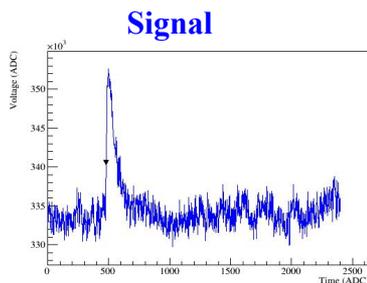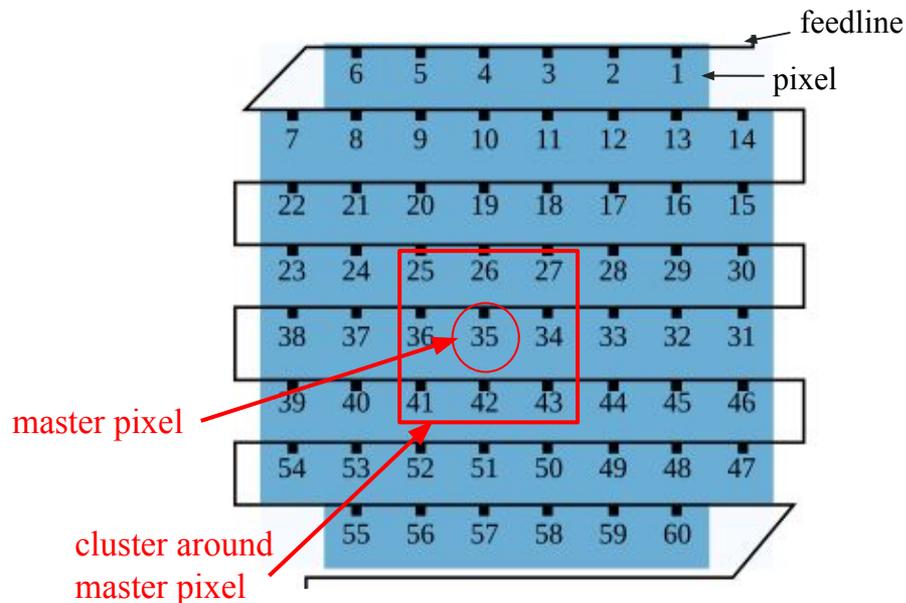BULLKID collaboration meeting, 15/01/2025

- **Step 1: Data acquisition**

Online triggering during data acquisition.

Triggering on one or more pixels (**master pixels**). Three types of triggers (signal, noise and reversed) with two trigger algorithms, <u>threshold trigger</u> and <u>optimum filter trigger</u>

At the same time, when a signal on a master pixel is triggered, a **cluster around it is acquired**

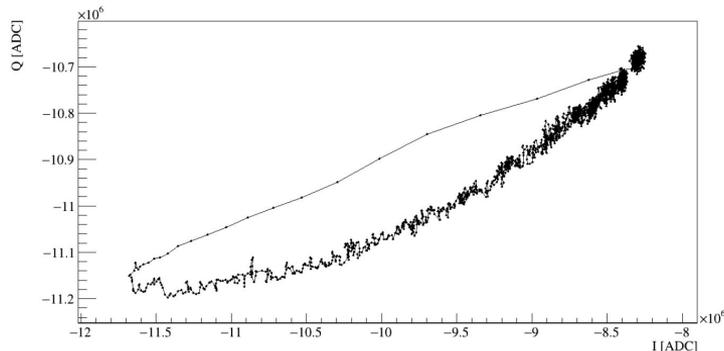**.h5 files** with triggered pulses, being the values of the complex transfer function vs time

feedline

pixel

master pixel

cluster around master pixel

Signal

Noise

Reversed stream

BULLKID analysis pipeline

● **Step 2: Data reading**

Simple conversion from h5 files
to root files used in analysis

> **.h5 files** with triggered pulses, being the
> values of the complex transfer function vs
> time

> **.root** files with all the events, global run
> information and basic event information
> (channel, trigger type, timestamp…)

Analysis software

All the analysis steps performed using the **Diana** software.
- ● **C++/ROOT classes** for BULLKID data processing
- ● **Python modules** for higher level analysis
- ● **Available** for all the collaboration

Documentation:
https://nucleus.roma1.infn.it/dianadoc/index.html

Manual:
https://nucleus.roma1.infn.it/dianadoc/DianaManual/DianaManual.pdf

GitHub:
https://baltig.infn.it/diana/dianasw

Python documentation:
https://nucleus.roma1.infn.it/dianadoc/pydoc/html/index.html
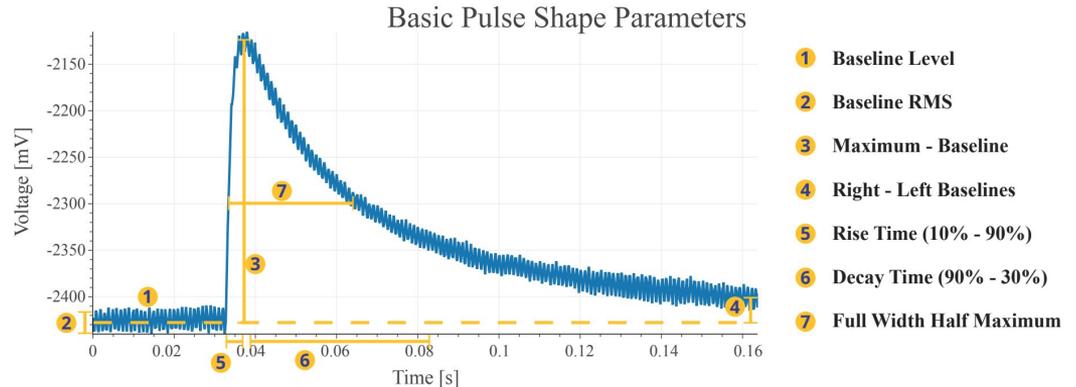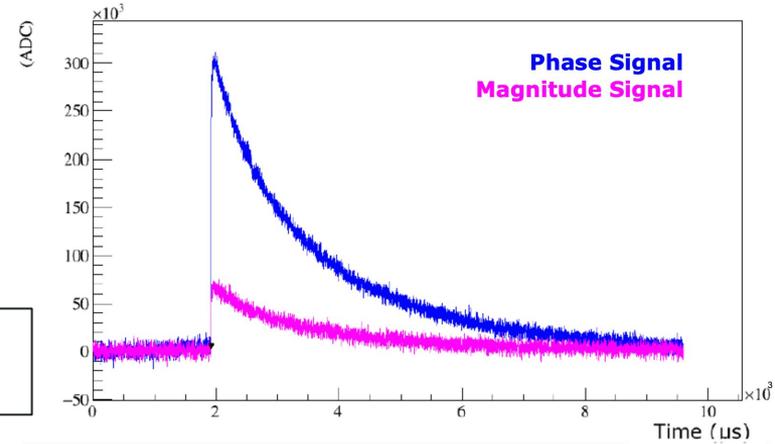
3

BULLKID analysis pipeline

- **Step 3: Data preprocessing**

.root files with all the events, global run information and basic event information (channel, trigger type, timestamp…)

Conversion from events with complex values to real valued pulses

Evaluating magnitude and phase of complex pulses

Evaluating some **preliminary parameters of the events** (baseline quantities, time constants…)

Basic Pulse Shape Parameters

1 Baseline Level

2 Baseline RMS

3 Maximum - Baseline

4 Right - Left Baselines

5 Rise Time (10% - 90%)

6 Decay Time (90% - 30%)
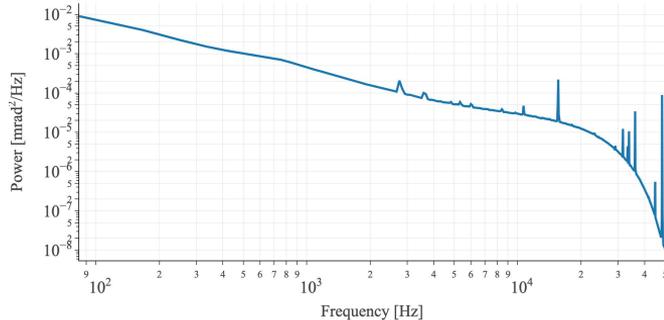
7 Full Width Half Maximum

- **Step 4: Averaging stage**

Build average pulse and noise power spectral density

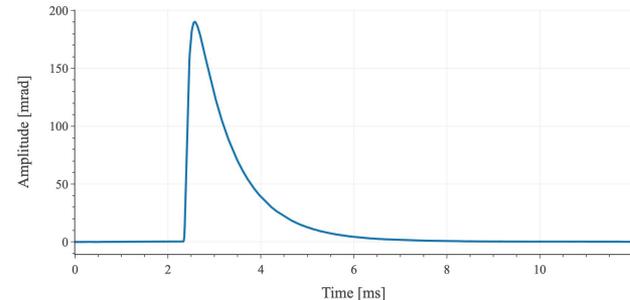Evaluating some **preliminary parameters of the events** (baseline quantities, time constants…)

Evaluate the **noise power spectral density** using noise events

Use pulses from LED source to evaluate the "**average pulse**" on master pixels

Average noise power spectrum

Average pulse

- **Step 5: Amplitude reconstruction**

Estimate the amplitude of pulses using the optimum filter.

Evaluate the **noise power spectral density** using noise events

Use pulses from LED source to evaluate the "**average pulse**" on master pixels

Amplitude reconstruction with **optimum filter**

Optimum Filter

filtered signal

average pulse
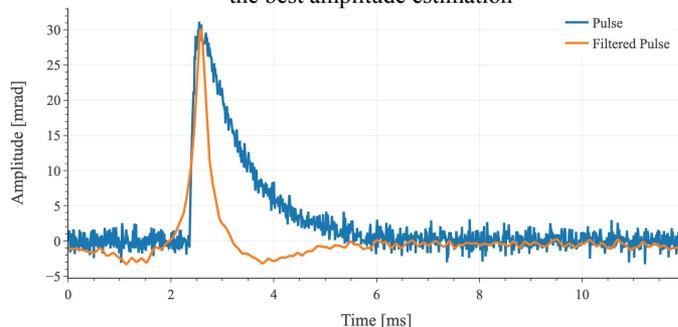
$$\hat{A}(t_0) = \frac{\int_{-\infty}^{\infty} df\, e^{i\omega t_0} \frac{\tilde{s}^*(f)}{N(f)} \tilde{v}(f)}{\int_{-\infty}^{\infty} df\, \frac{|\tilde{s}(f)|^2}{N(f)}}$$

signal to filter

noise power spectral density

Filtering in the frequency domain, maximizes the signal-to-noise ratio.

Maximum of the filtered pulse gives the best amplitude estimation

Amplitude resolution

$$\sigma_{\hat{A}}^2 = \left[\int_{-\infty}^{\infty} df\, \frac{|\tilde{s}(f)|^2}{N(f)} \right]^{-1}$$

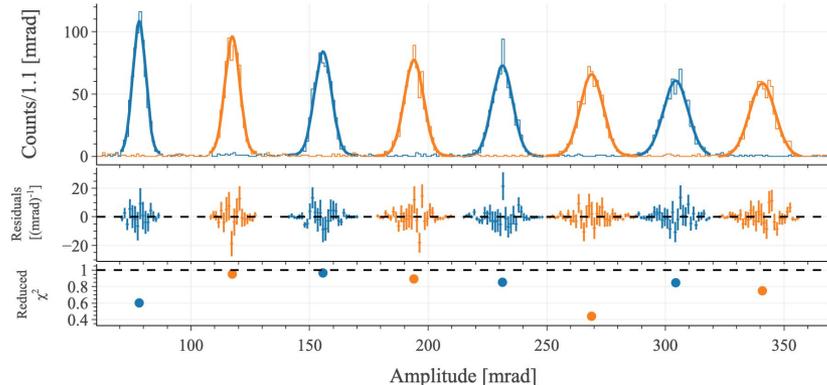| σ^{noise} [mrad]	$\sigma_{\text{OF}}^{\text{noise}}$ [mrad]
1.61	0.95

- ## Step 6: Calibration

Calibrate the master pixels using an optical calibration

> **Amplitude reconstruction** with **optimum filter**

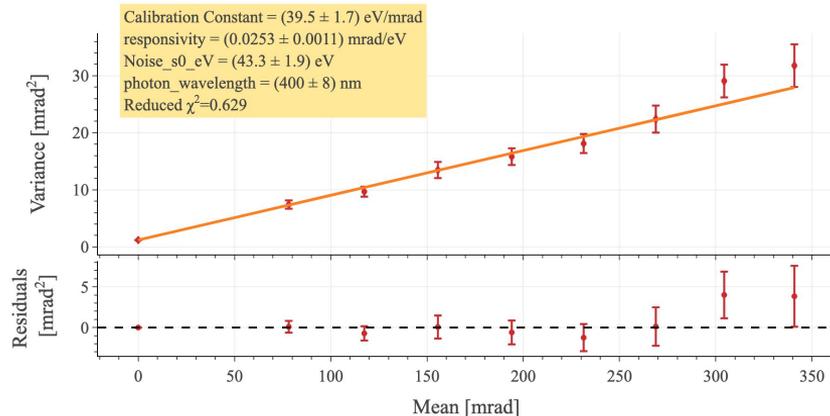> **Calibration** of master pixels using a LED source

Optical calibration (see Giorgio's talk)

Shine the detector with LED pulses in which the number of photons N_{ph} follows a poisson distribution.

$$\mu = r \langle N_{ph} \rangle \epsilon_{ph} \qquad \sigma^2_{LED} = (r \, \epsilon_{ph})^2 \langle N_{ph} \rangle = r \, \epsilon_{ph} \, \mu \longrightarrow \boxed{\sigma^2 = r \, \epsilon_{ph} \, \mu + \sigma_0^2}$$

This relation gives the detector responsivity

Gaussian fit to obtain mean and standard deviation of LED gaussians

Linear fit to obtain the responsivity and baseline resolution

Calibration Constant = (39.5 ± 1.7) eV/mrad
responsivity = (0.0253 ± 0.0011) mrad/eV
Noise_s0_eV = (43.3 ± 1.9) eV
photon_wavelength = (400 ± 8) nm
Reduced χ^2=0.629
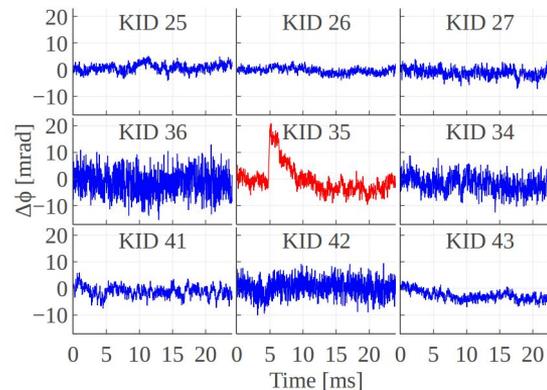
- **Step 7: Coincidence analysis**

Identify events originated by a direct energy release in the master pixels.

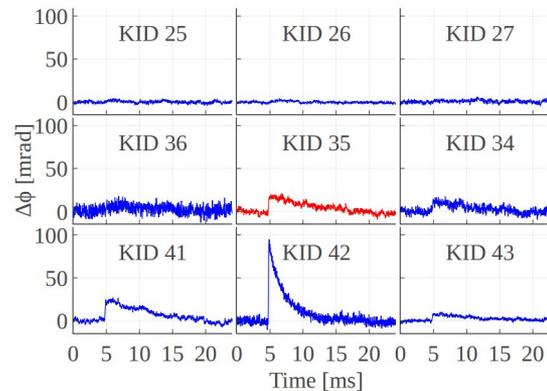| **Calibration** of master pixels using a LED source |
|---|

↓

Analysis on master pixels to **discard leakage events**

Evaluate the **amplitude in neighbour pixels in coincidence** with the master pixel event.

Define proper **cut variables** …more on this in my talk tomorrow.

✅
Keep the event

❌
Discard the event

- **Step 8: Cuts and final energy spectrum**

Cuts are defined on LED events. Cut efficiency is computed on LED events as well.

Analysis on master pixels to **discard leakage events**

↓

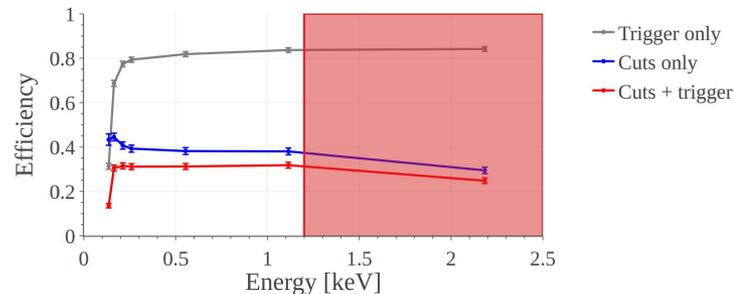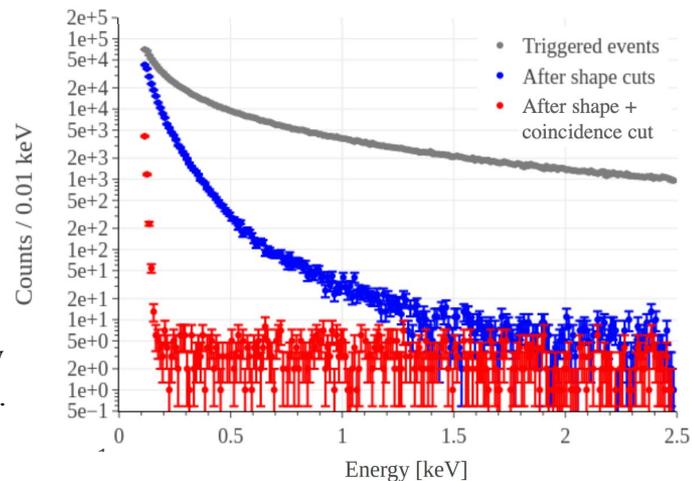**Cut definition** on LED events

↓

Cut implementation on background and reversed events

↓

Correction for cut efficiency, trigger efficiency ad d.r.u. conversion. **Energy spectrum evaluation**

Comparison of background rate with reversed trigger rate shows that **no low-energy excess is observed**.

Cuts (<u>more on this in my talk tomorrow</u>)
- **Pulse shape cuts**: many variables available, ongoing studies
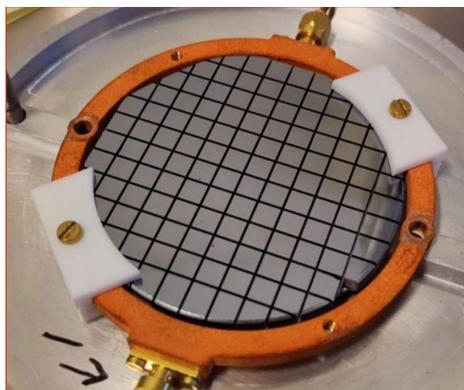- **Coincidence cuts**: efficient in suppressing low energy background
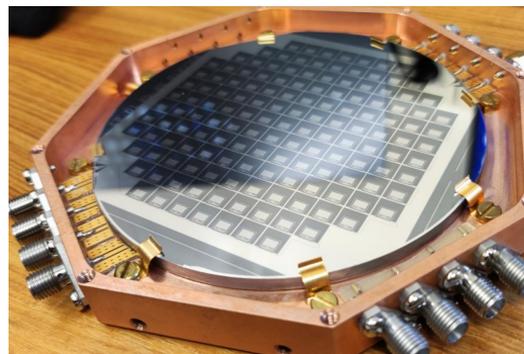
BULLKID future analysis

This standard analysis is implemented on:
-   A single 3 inches wafer
-   Clusters with up to 2 master pixels
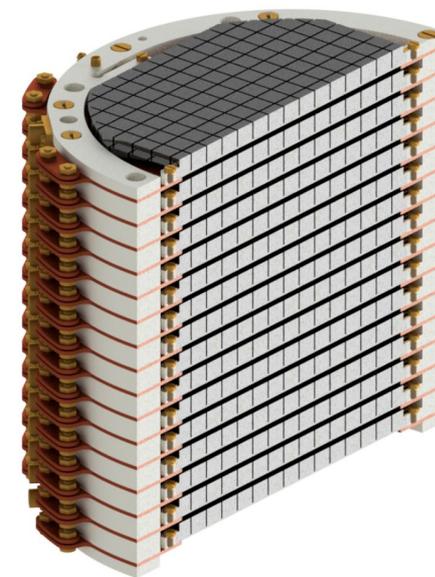-   Clusters with up to 12 neighbours pixels

Need for scaling up!

3 inches wafer, 60 KIDs

100 mm wafer, 145 KIDs
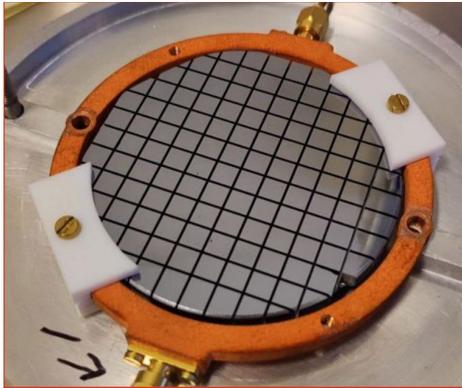

Full stack with 100 mm wafer, 2320 KIDs
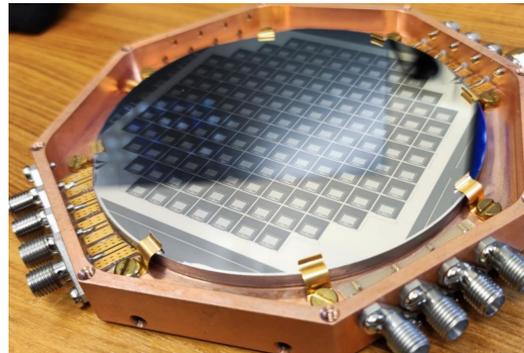
BULLKID future analysis

What we did
- Multi RF line processing
- Speed up of the analysis
- Other detectors compatibility
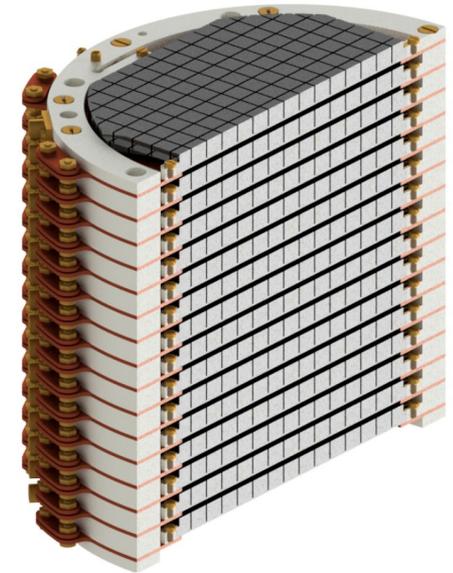- Heavily reduced disk usage

What to do
- Efficient cluster definition and analysis
- Coincidence analysis in the z direction
- Data reduction

3 inches wafer, 60 KIDs

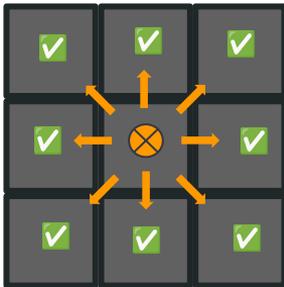
100 mm wafer, 145 KIDs

Full stack with 100 mm wafer, 2320 KIDs

For a single wafer, given N pixels taken as master and N_{neigh} pixels as neighbours for each master, the number of operations scale as $\mathbf{N} \cdot \mathbf{N_{neigh}}$.

→

We need to **minimize N_{neigh}** if we want to exploit all pixels of a wafer, to speed up the analysis, but **keeping the same rejection power**.
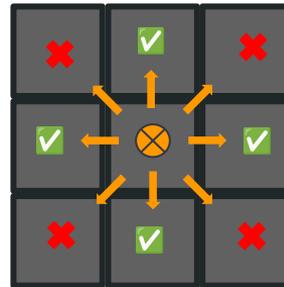
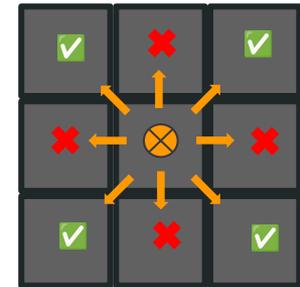Possible choices (rejection power to be evaluated):

All voxels	"+" voxels	"x" voxels

$N_{neigh} = 8$ $N_{neigh} = 4$,   14 % leakage in each $N_{neigh} = 4$,   5 % leakage in each

Or: trigger every pixel independently, then define your master signal(s) in the array via amplitude ordering.

Coincidence analysis in the z direction

Standard coincidence analysis in the x-y wafer plane does not remove events caused by the particles vertically crossing the stacked array (e.g. muons). To be discarded for a dark matter/neutrino search.
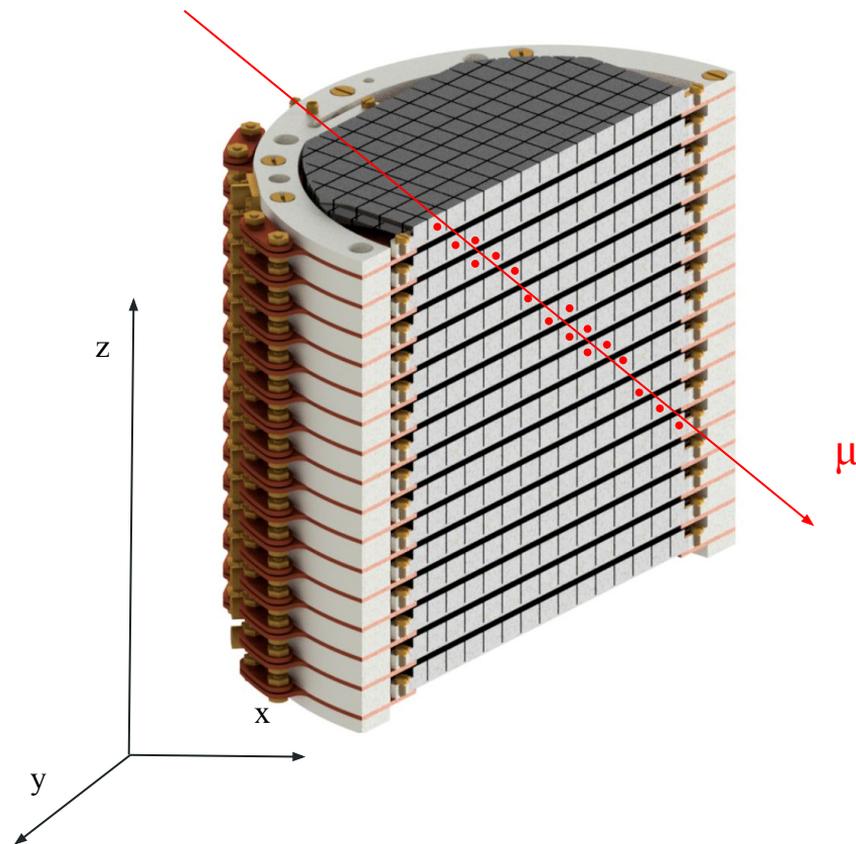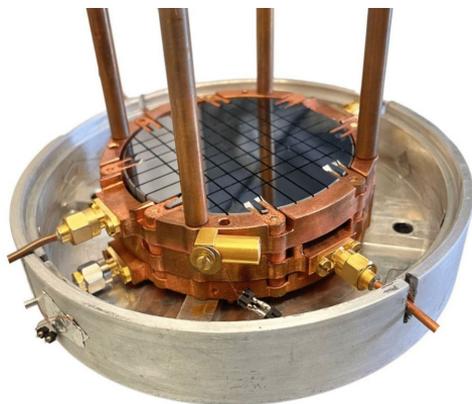
Need for a cluster analysis also with up/down wafers.

Multi wafer analysis ready

The analysis software is ready to process different RF lines, each one coming from a wafer.
<u>Soon test with demonstrator at Sapienza.</u>

Demonstrator@Sapienza: 3 stacked 3 inches wafers

z

x

y

μ

Data reduction

Offline pulse shape analysis means that waveforms must be saved. **Large amounts of data with O(2000) pixels.**
One event is ～8 kB, **～5GB/h files** with a master and 8 neighbors.

Ideas for data reduction:

- Optimum trigger allows for a **preliminary data selection at trigger level, using pulse shape variables** defined by the optimum filter (easy implementation, not efficient at low energies).
- **Preliminary coincidence analysis** at trigger level. Very efficient in removing low energy events, difficult implementation.
- **Not save waveforms**, but only amplitudes and (maybe) some pulse shape variables. Light data, but waveform analysis cannot be performed anymore.

However, a very very low background is expected at LNGS:
~6 counts/day in the entire detector with 1 d.r.u in the region 0-10 keV (assuming flat background with no low energy excess).
Dominated by reversed events, ~10^7 d.r.u. rate measured at threshold.

thanks for your attention!