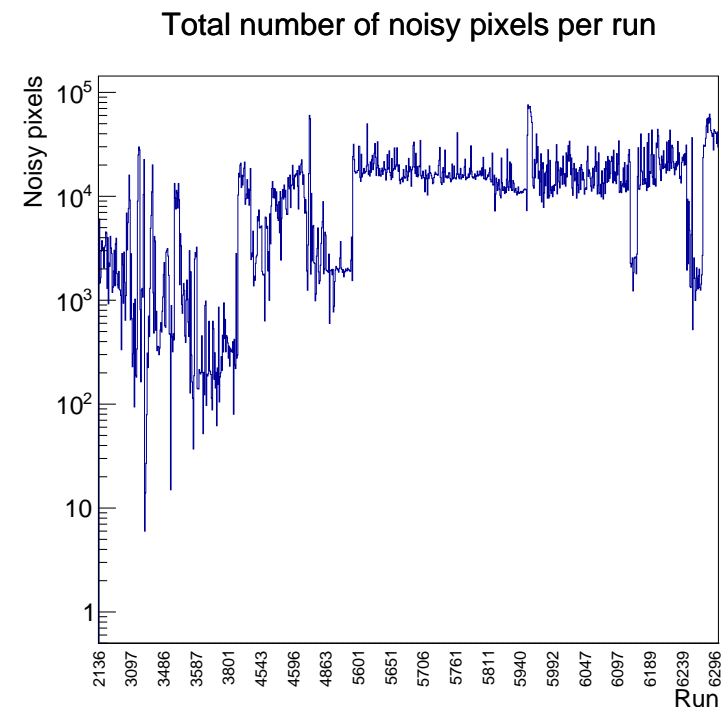


# "pixelquality" in the CDB

Urs Langenegger

2025/07/04

- Introduction
- "Results"
- Coding (how to)



# CDB - "pixelquality"

---

- Pixel "quality" used as guinea pig for run-dependent "conditions"
  - ▷ therefore in the CDB - "conditions" database
  - ▷ "quality" captures
    - noise
    - suspect/bad
    - turned off/masked
    - possibly other aspects
  - ▷ criteria are algorithm dependent (can be eternally improved/changed)
- CDB record to describe this in Beam 2025
  - ▷ `calPixelQualityLM` with
    - link flag
    - column flag
    - pixel flag
  - where flag is char/enum with values
    - NotFound = -1,
    - Good = 0,
    - Noisy = 1,
    - Suspect = 2,
    - DeclaredBad = 3,
    - TurnedOff = 9
- Attempted balance between access speed and record size

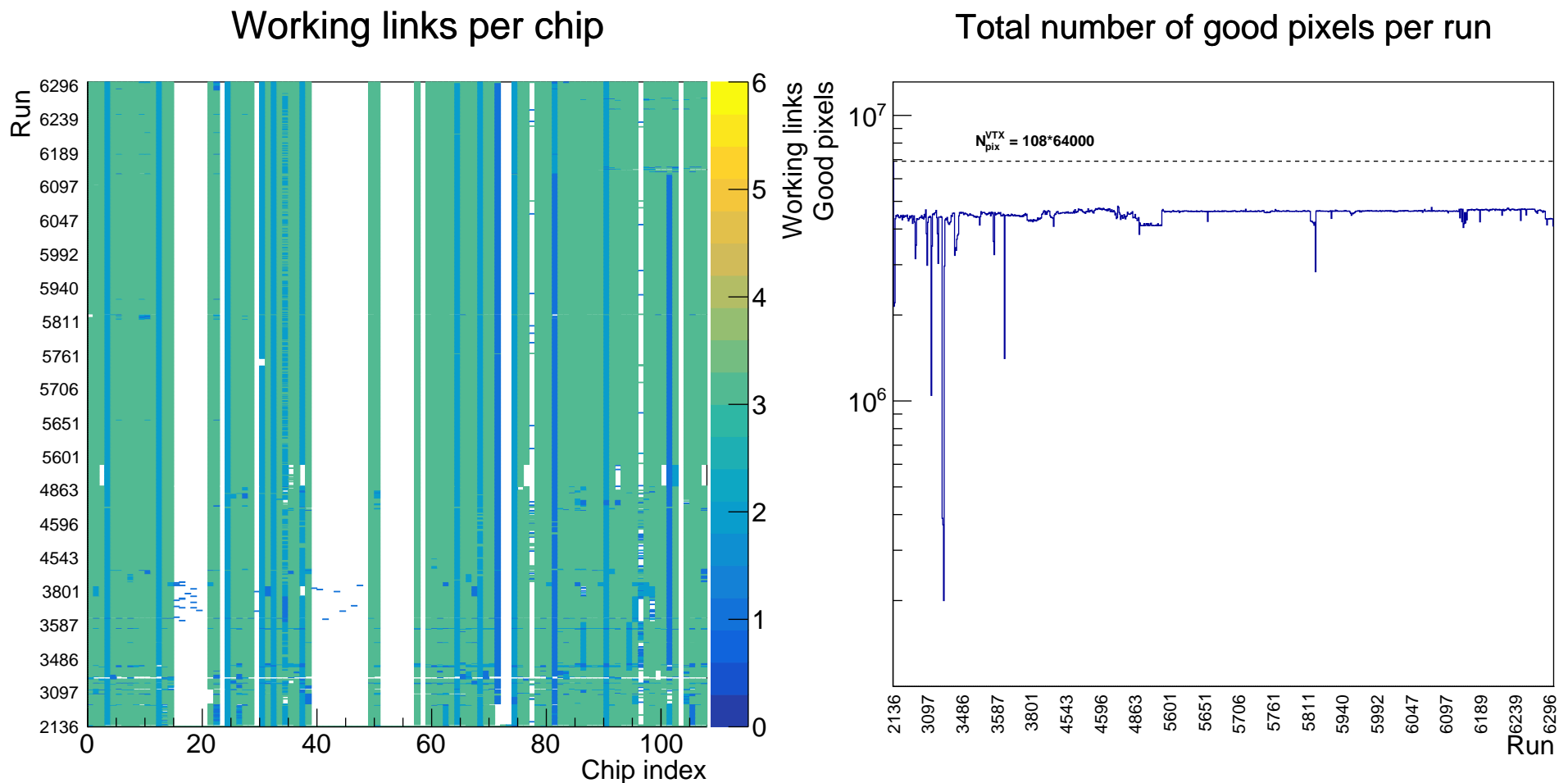
# Storage of pixelqualityLM records

---

- "pixelqualitylm" produced for global tag "datav6.2=2025Beam"
  - ▷ "v6.2" indicates that it will NOT work with (mu3eTrirec) v6.1
  - ▷ for each "significant" run with class "Beam"
- Availability
  - ▷ pc11740.psi.ch/cdb
    - (note: do not display the IOVs, you will need a log of patience)
    - use it via the REST interface (see below)
  - ▷ merlin:/data/experiment/mu3e/code/cdb
    - you (likely) should mirror that to your local platform!
    - < 0.5 GB
    - use it via the JSON interface (see below)

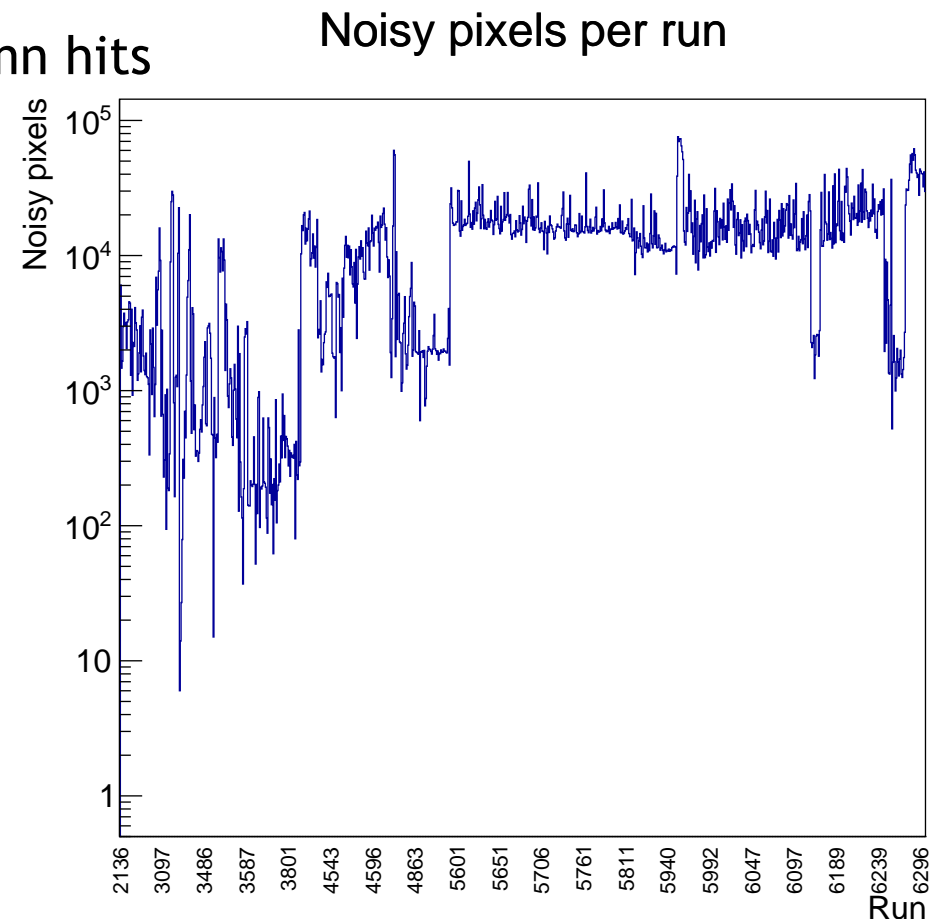
# "Results"

- The most obvious bugs have been fixed, but . . .
  - ▷ there are still many more
  - ▷ consider this a start (with our flights leaving tomorrow at 7am)



# Pixel quality determination

- Pixel quality determination (same disclaimer as previous slide)
  - ▷ Dead/problematic links determined with (per-chip) average link hits
    - not counting 8 columns at sensor edges
    - "dead" links have  $< 10\%$  of average link hit multiplicity (except if link average hits  $< 10$ )
  - ▷ Dead columns with average column hits
    - same as for links
  - ▷ Pixels are labeled "noisy" if their hit count is  $> 10 \times \sqrt{\langle N_{hit} \rangle}$ 
    - average over entire sensor
    - hence biased by (noisy) edges
      - i.e. underestimates noisy pixels (because average is too high)
    - column with  $> 40$  noisy pixels
      - mark entire column as noisy! (saving payload space)
- This is just a start!



# Coding How-to (reco level)

---

- Validated on merlin (with the HEAD of dev as of 2025/07/04 16:00)

```
git clone git@bitbucket.org:mu3e/mu3e
cd mu3e
git submodule update --init --recursive
cd modules/mu3eUtil
git pull origin dev
git checkout dev
git merge origin/ursl-pixelquality

cd ../../
git merge origin/ursl-pixelquality

mkdir _build && cd _build
cmake ..
make -j40

cd ../run
../_build/mu3eTrirec/mu3eTrirec -n 1000000 --conf ./trirec_twolayer_beam.conf \
--cdb.dbconn=/data/experiment/mu3e/code/cdb --cdb.globalTag=datav6.2=2025Beam \
/data/experiment/mu3e/data/2025/trirec/250621/run06267-sorted.root \
--output ./run06267-trirec.root
^C
```

- This is very naive example code
  - ▶ can be ported to any context
- You should closely look at the pixelqualityLM payload using

```
../_build/modules/mu3eUtil/cdb/test/cdbPrintPayload \
/data/experiment/mu3e/code/cdb/payloads/tag_pixelqualitylm_datav6.2=2025Beam_iov_6267
```

# Coding How-to (pixelqualityLM analysis)

---

- Assuming you followed the instructions on the previous slide

- ▷ Using default JSON CDB and all runs in tag

```
merlin>cd ../_build  
merlin>modules/mu3eUtil/cdb/test/anaPixelQuality
```

- ▷ Using REST CDB and runs in certification file

```
merlin>modules/mu3eUtil/cdb/test/anaPixelQuality -db rest \  
-r ../modules/mu3eUtil/cdb/certification/2025/2025-Beam-v1-significant.run
```

- This will result in pdf files

- ▷ modify ad libitum

# Conclusions

---

- Vertex detector "pixel quality" analysis

- ▷ All significant Beam data runs
- ▷ All payloads stored in
  - JSON directory on [merlin:/data/experiment/mu3e/code/cdb](https://merlin:/data/experiment/mu3e/code/cdb)
  - MongoDB/REST server [pc11740.psi.ch](https://pc11740.psi.ch)

⇒ Coding examples provided

- ▷ how to enable "bad-pixel" masking in reco
- ▷ how to study "pixelquality" characteristics over the full beam period
- ▷ how to look at the "pixelquality" payload

- This is not (yet!) an assessment of the VTX hardware performance

- ▷ (much?) more validation required

- Of note

- ▷ New global tag "datav6.2=2025Beam"
- ▷ All this does NOT work with the CDB in "mu3e/install" (see above)!
- ▷ Does not work with (mu3eTrirec) v6.1 and old minalyzer

- I will read emails again on Aug 4, 2025 :-)