CLAS Collaboration Meeting – March 10, 2026

Real Run Number Simulations

Raffaella De Vita (Jefferson Lab)

Run-dependent simulations

- Current simulations use a fixed run number and get run-group dependent conditions (e.g. geometry, list of malfunctioning elements, …) reading CCDB tables from different variations
  - Sufficient to account for main or average conditions in a data set
  - High maintenance because of multiplication of CCDB variations, gcards, and yamls
  - Impractical for implementing run-by-run changes such as temporarily dead elements

- **Solution: run simulations with "real run numbers", picking up the corresponding detector configuration and performance (resolutions and efficiency) from CCDB tables**

Run-dependent simulations

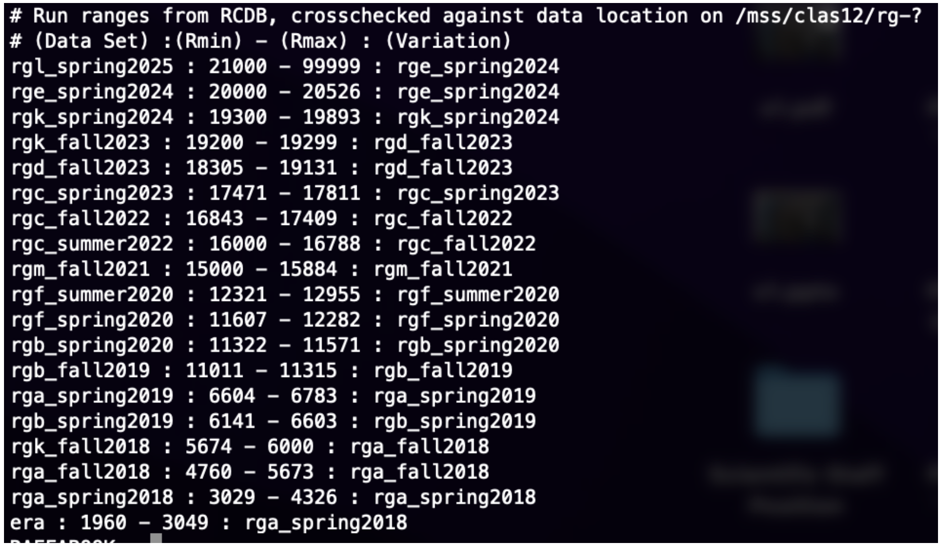
- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Run-dependent simulations

- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Populating CCDB run ranges…

…from run-group variations



```
# Run ranges from RCDB, crosschecked against data location on /mss/clas12/rg-?
# (Data Set) :(Rmin) - (Rmax) : (Variation)
rgl_spring2025 : 21000 - 99999 : rge_spring2024
rge_spring2024 : 20000 - 20526 : rge_spring2024
rgk_spring2024 : 19300 - 19893 : rgk_spring2024
rgk_fall2023 : 19200 - 19299 : rgd_fall2023
rgd_fall2023 : 18305 - 19131 : rgd_fall2023
rgc_spring2023 : 17471 - 17811 : rgc_spring2023
rgc_fall2022 : 16843 - 17409 : rgc_fall2022
rgc_summer2022 : 16000 - 16788 : rgc_fall2022
rgm_fall2021 : 15000 - 15884 : rgm_fall2021
rgf_summer2020 : 12321 - 12955 : rgf_summer2020
rgf_spring2020 : 11607 - 12282 : rgf_spring2020
rgb_spring2020 : 11322 - 11571 : rgb_spring2020
rgb_fall2019 : 11011 - 11315 : rgb_fall2019
rga_spring2019 : 6604 - 6783 : rga_spring2019
rgb_spring2019 : 6141 - 6603 : rgb_spring2019
rgk_fall2018 : 5674 - 6000 : rga_fall2018
rga_fall2018 : 4760 - 5673 : rga_fall2018
rga_spring2018 : 3029 - 4326 : rga_spring2018
era : 1960 - 3049 : rga_spring2018
```

- Defined mapping between **run ranges** and **data sets/variations** based on RCDB
- Developed scripts to crawl CCDB and determine programmatically:
  - The list of tables with entries in run-group variations
  - For each table, the list of such entries
  - Compare entries across variations with contiguous run ranges, and if equal, plan to make a single new entry for the combined run range in default, for better CCDB "readability"
  - Compare to the entries in the default variation for the corresponding run range, to make new entries only if different
  - Provide the list of commands to copy the selected entries to the default variation

(see https://code.jlab.org/devita/realrunnumbers)

Run-dependent simulations

- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Run-dependent simulations

- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Run-dependent simulations

- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Reconstruction - digitization consistency checks

- Eliminate "ad-hoc" conditions used by reconstruction when running on MC events:
  - *For example, "if(runNumber==11) {…}" or "if(event.hasBank("MC::Particle") {…}"*
  - Verified rigorously by reconstructing MC events generated with a "real run number" but the same constants as the predefined MC run 11 and comparing to regular simulations
- Verify the GEMC digitization algorithms match reconstruction algorithms when using "real run number" calibration constants
  - Verified by comparing simulations for run 11 and a few "real run numbers"

Reconstruction - digitization consistency checks

- Removed hardcoded run-number-dependent conditions in reconstruction:
  - Modified DC and ECAL
  - (Removed hardcoded switch to default CCDB variation in BAND)
  - Verified gemc-coatjava chain gives identical results with run=11 and run≠11, when using the same calibration constants and geometry
- Updated gemc digitization to match the reconstruction algorithm when real-run-number calibration constants are used:
  - Necessary because MC constants effectively disable some effects that are accounted for in non-MC constants
  - Added adc-tdc time offset in FTOF and CTOF
  - Updated FTOF time walk corrections
  - Added hpos correction to CTOF
  - Updated HTCC to use gains from CCDB and include ring_time offsets
  - Included TDC jitter for all detectors
  - Found gemc RUN::rf has offset-corrected time while reconstruction assumes RUN::rf has uncorrected times and applies offset in EB: fixed in GEMC, yet to be tested

https://github.com/JeffersonLab/coatjava/pull/905

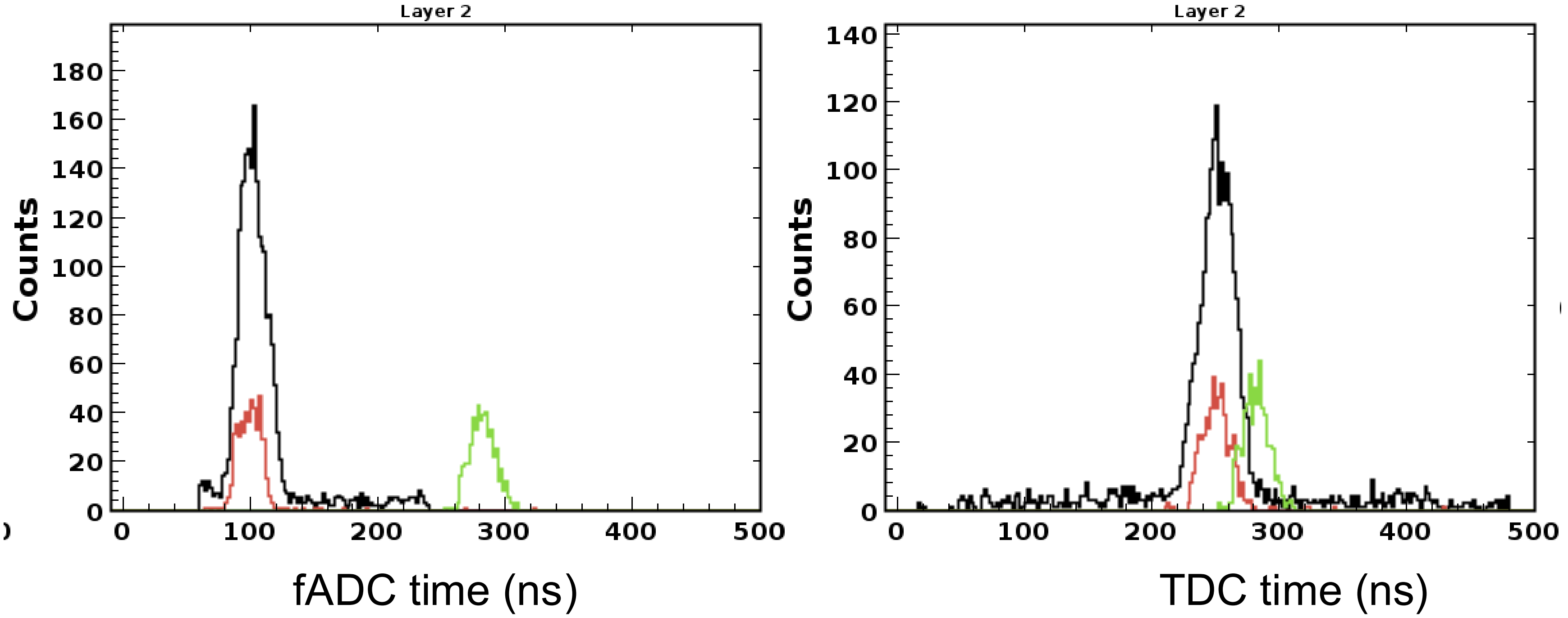
https://github.com/gemc/clas12Tags/pull/62

ECAL

Data: run 6662

Old MC

New MC with constants from run 6662



Run-dependent simulations

- **Requirements:**
  1. CCDB geometry, status, resolutions, and efficiency tables need to be properly populated for the relevant run ranges in the default variation
  2. GEMC should load run-dependent geometry (NEW) and detector response constants (already in place based on CCDB calibration tables)
  3. Reconstruction should load the geometry for the actual run number of the event being processed, reading CCDB geometry tables for that run number and building the detectors' geometry on the fly (https://github.com/JeffersonLab/coatjava/tree/detector-changed-hook)
  4. Reconstruction of MC events should work when using "real run number" calibration constants, i.e. the GEMC digitization algorithm has to be fully consistent with the reconstruction algorithm used for real data

Summary

- All “pieces” are ready!
- Need to agree on the sequence of updates:
  - CCDB first
  - GEMC
  - Coatjava
- Will need new gcards and yamls in clas12-config
- OSG portal to support users providing a list of run numbers

- A few notes:
  - No more need for variations in real data processing
  - No need for special MC time offsets to match signal time to data (relevant for BG merging)
  - Since this requires significant changes to reconstruction, it will not be backported to “old” coatjava versions