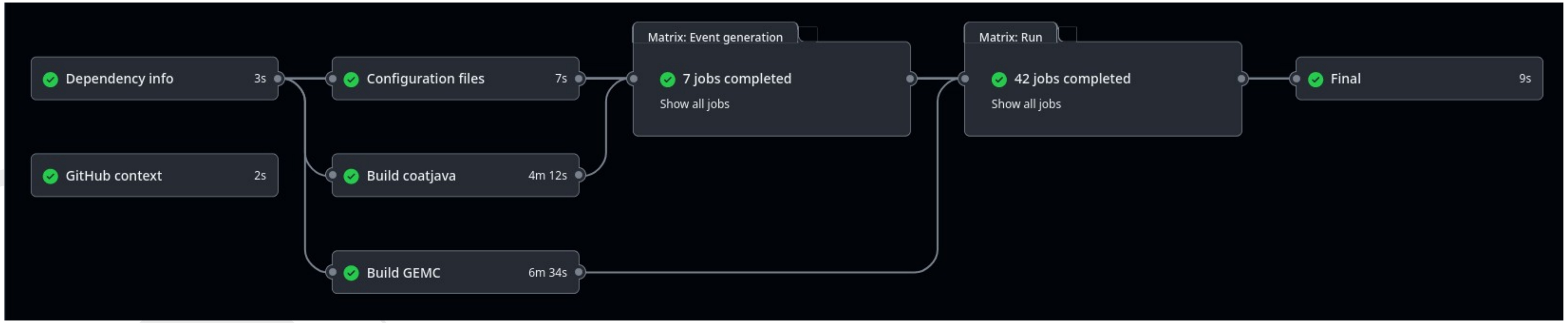
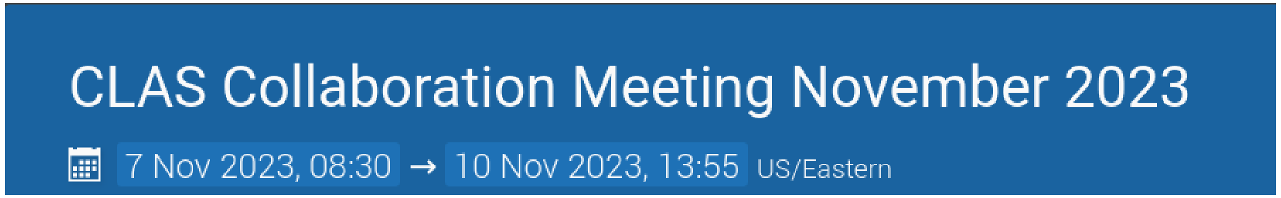
Automatic Validation of Simulation and Reconstruction Software



Christopher Dilks



CLAS Collaboration Meeting November 2023

7 Nov 2023, 08:30 → 10 Nov 2023, 13:55 US/Eastern

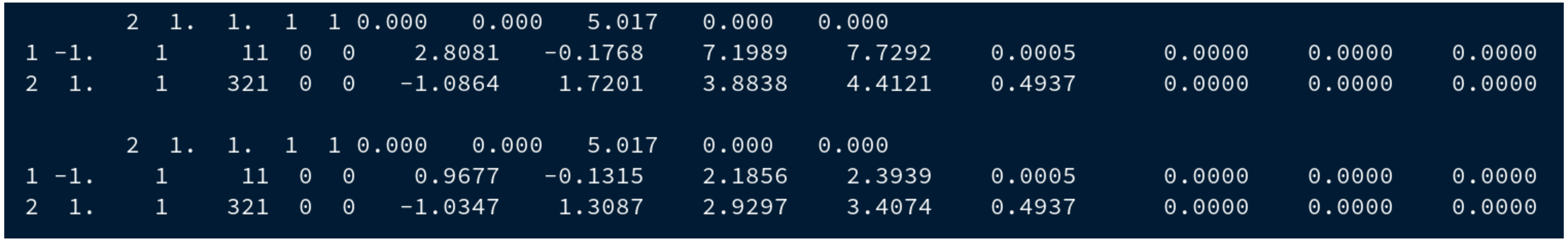
clas12-validation

- Repository: https://github.com/JeffersonLab/clas12-validation
- Purpose: automated testing and validation of the full pipeline:
  - Event generation → Simulation → Reconstruction → Analysis
- The tests run when:
  - Software developers make or propose changes to any relevant code → run a low-statistics “smoke” test
  - Nightly high-statistics test
- Technical details:
  - Uses Github Actions (continuous integration)
  - Reusable and externally configurable workflow
  - Recent nightly runs: https://github.com/JeffersonLab/clas12-validation/actions?query=event%3Aschedule

Event Generation

- Electron + single particle, distributed in FD, CD, or FT
  - Could someday test other generators, but we want to keep it simple
- Tests:
  - FD electron + FD kaon
  - FD electron + CD kaon
  - FD electron + FD neutron
  - FD electron + FD gamma
  - FD electron + CD neutron
  - FT electron + FD kaon
  - FD electron + FT gamma

Sample Lund events for electron + kaon:

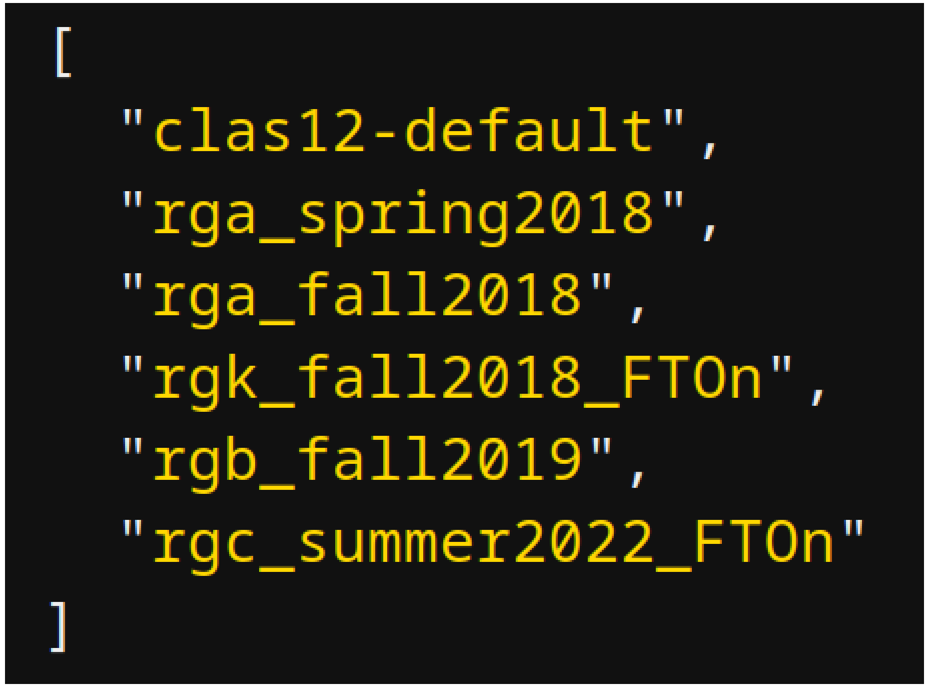


```
       2  1.   1.   1   1 0.000    0.000    5.017    0.000    0.000
1 -1.     1      11  0   0     2.8081    -0.1768     7.1989     7.7292     0.0005       0.0000     0.0000     0.0000
2  1.     1     321  0   0    -1.0864     1.7201     3.8838     4.4121     0.4937       0.0000     0.0000     0.0000

       2  1.   1.   1   1 0.000    0.000    5.017    0.000    0.000
1 -1.     1      11  0   0     0.9677    -0.1315     2.1856     2.3939     0.0005       0.0000     0.0000     0.0000
2  1.     1     321  0   0    -1.0347     1.3087     2.9297     3.4074     0.4937       0.0000     0.0000     0.0000
```

Simulation

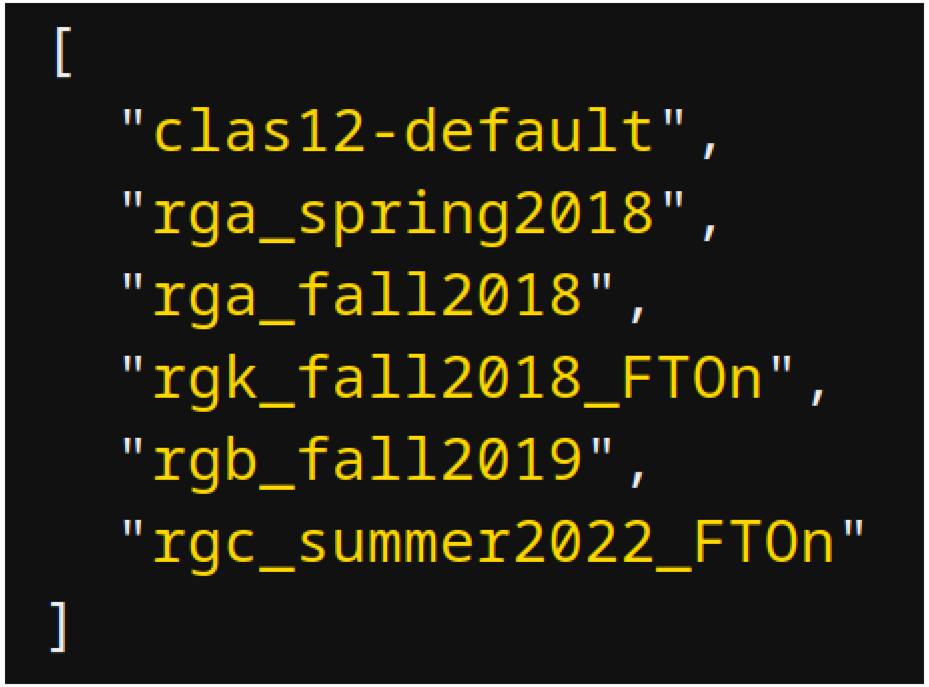
- Builds and runs latest version of **clas12Tags (GEMC)**
  - (or a developer’s version)
  - https://github.com/gemc/clas12Tags
- Uses simulation configurations (gcards) from **clas12-config**
  - https://github.com/JeffersonLab/clas12-config
  - Defaults to use the *latest* version of these specified configurations:



```
[
  "clas12-default",
  "rga_spring2018",
  "rga_fall2018",
  "rgk_fall2018_FTOn",
  "rgb_fall2019",
  "rgc_summer2022_FTOn"
]
```

Reconstruction

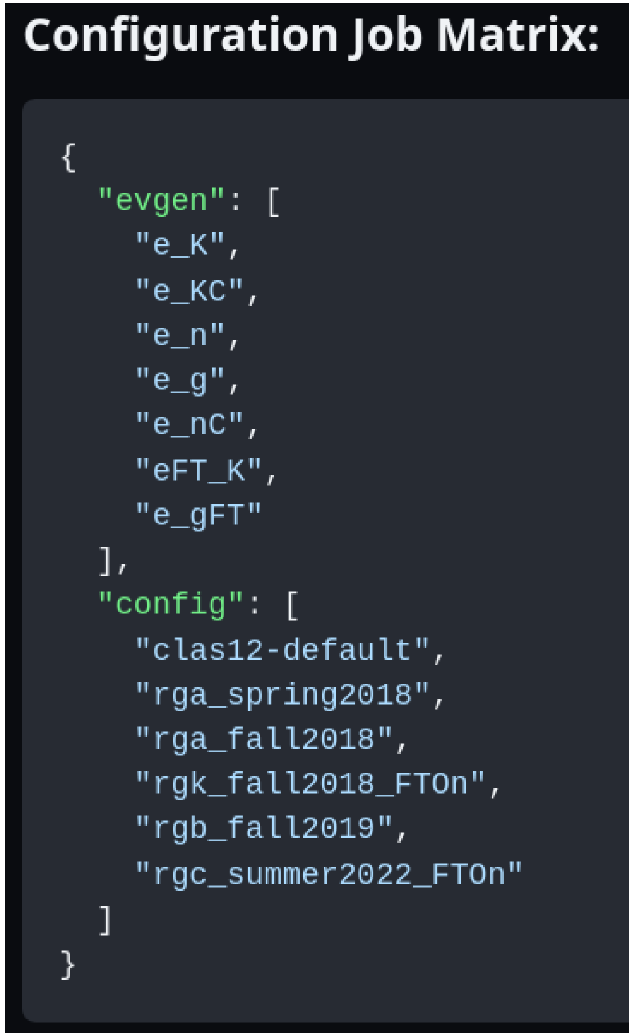
- Builds and runs latest version of **coatjava**
  - (or a developer’s version)
  - https://github.com/JeffersonLab/coatjava
- Uses reconstruction configurations (YAML files) from **clas12-config**
  - https://github.com/JeffersonLab/clas12-config
  - Defaults to use the *latest* version of these specified configurations:



```
[
  "clas12-default",
  "rga_spring2018",
  "rga_fall2018",
  "rgk_fall2018_FTOn",
  "rgb_fall2019",
  "rgc_summer2022_FTOn"
]
```

Validation Job Matrix

- Event generation types and configurations are combined to a full **job matrix**
  - 7 event generations
  - 6 simulation/reconstruction configurations
  - 42 jobs total

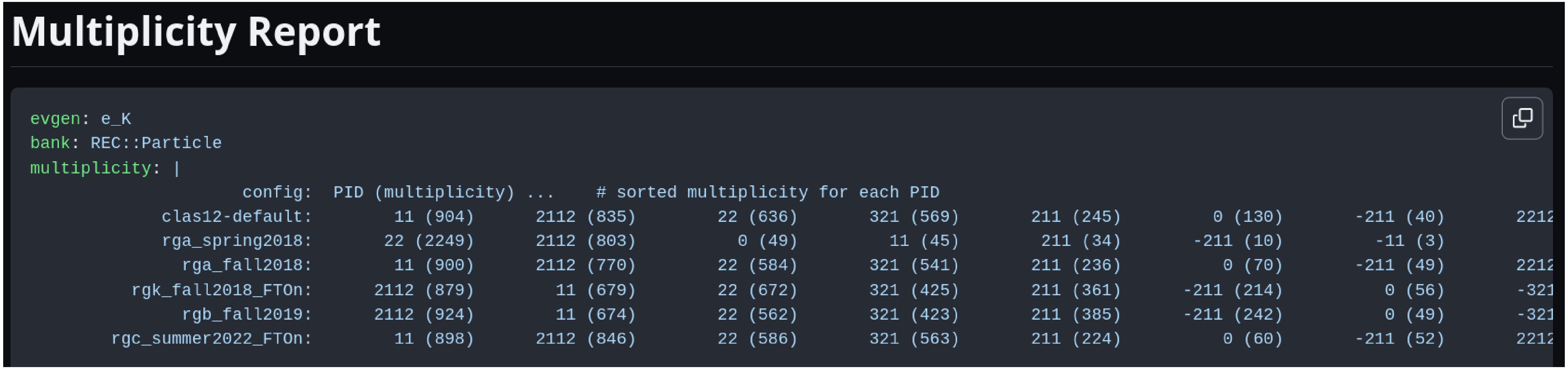


Configuration Job Matrix:

```
{
  "evgen": [
    "e_K",
    "e_KC",
    "e_n",
    "e_g",
    "e_nC",
    "eFT_K",
    "e_gFT"
  ],
  "config": [
    "clas12-default",
    "rga_spring2018",
    "rga_fall2018",
    "rgk_fall2018_FTOn",
    "rgb_fall2019",
    "rgc_summer2022_FTOn"
  ]
}
```

“Analysis”

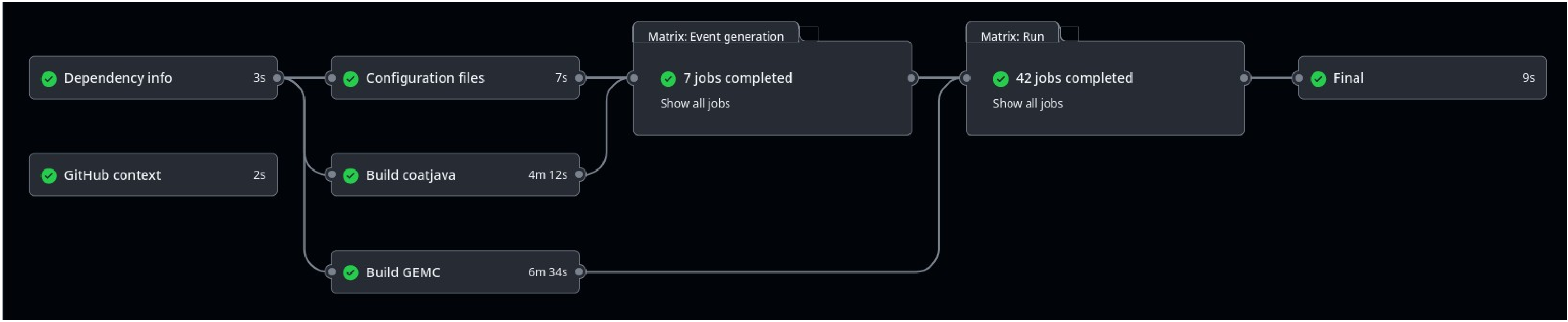
- To be improved…
- Currently just a simple multiplicity report
  - Just to test the reconstruction output is readable
  - Counts how many particles with each Event Builder PID from REC(FT)::Particle
  - Example report at the bottom of https://github.com/JeffersonLab/clas12-validation/actions/runs/6768210914
- Discussion point: how can an “analysis” step be made ***useful*** for automated validation?



Multiplicity Report

```
evgen: e_K
bank: REC::Particle
multiplicity: |
                     config:  PID (multiplicity) ...    # sorted multiplicity for each PID
             clas12-default:        11 (904)      2112 (835)        22 (636)       321 (569)       211 (245)         0 (130)       -211 (40)       2212
             rga_spring2018:       22 (2249)      2112 (803)          0 (49)         11 (45)        211 (34)       -211 (10)         -11 (3)
               rga_fall2018:        11 (900)      2112 (770)        22 (584)       321 (541)       211 (236)          0 (70)       -211 (49)       2212
          rgk_fall2018_FTOn:      2112 (879)        11 (679)        22 (672)       321 (425)       211 (361)      -211 (214)          0 (56)       -321
               rgb_fall2019:      2112 (924)        11 (674)        22 (562)       321 (423)       211 (385)      -211 (242)          0 (49)       -321
        rgc_summer2022_FTOn:        11 (898)      2112 (846)        22 (586)       321 (563)       211 (224)          0 (60)       -211 (52)       2212
```

Technical Details



Get upstream info

Build, get config files

Event generation (7 jobs)

Simulation, Reconstruction, Analysis (7x6=42 jobs)

Summarize, etc.

- Low-stats (10 events) smoke test takes ~30 min total
- Nightly High-stats (1000 events) test takes ~2 hours total
- Yes… the time sounds weird… the limitation seems to be number of parallel jobs GitHub can run
  - Single simulation → reconstruction → analysis job takes about 2 min (1 hour) for low (high) stats

Final Thoughts

- **clas12-validation** has been doing its job for a few weeks
- Improvements:
  - Other event generators?
    - Keep in mind GitHub’s fixed 6 hour time limit (on the workflow run)
  - Analysis
    - A good validator needs teeth!
    - Can we automate the quality assessment of the output files?
    - Anyone have (very simple) analysis code that could do this, or would be willing to contribute to the effort?