Data & Analysis Preservation: status update

Maxim Potekhin
Nuclear and Particle Physics Software Group

Brookhaven™
National Laboratory

PHENIX DAP Meeting
01/13/2022

Overview

- Website updates
  - Links to multiple conference Zenodo uploads/keywords added (thanks Gabor)
  - 73 conferences total, now going back to 2014
  - Substantial improvement of the direct photon page (next slides)
- HEPData
  - Multiple items progressing/catching up with work done throughout 2021
  - Spreadsheet updated:

  https://docs.google.com/spreadsheets/d/1rABxzuM-h9Rukz08ut_m8xnMo0B_J1LKre8bM7B7264/edit#gid=0
- Direct Photons – REANA adaptation, work in progress

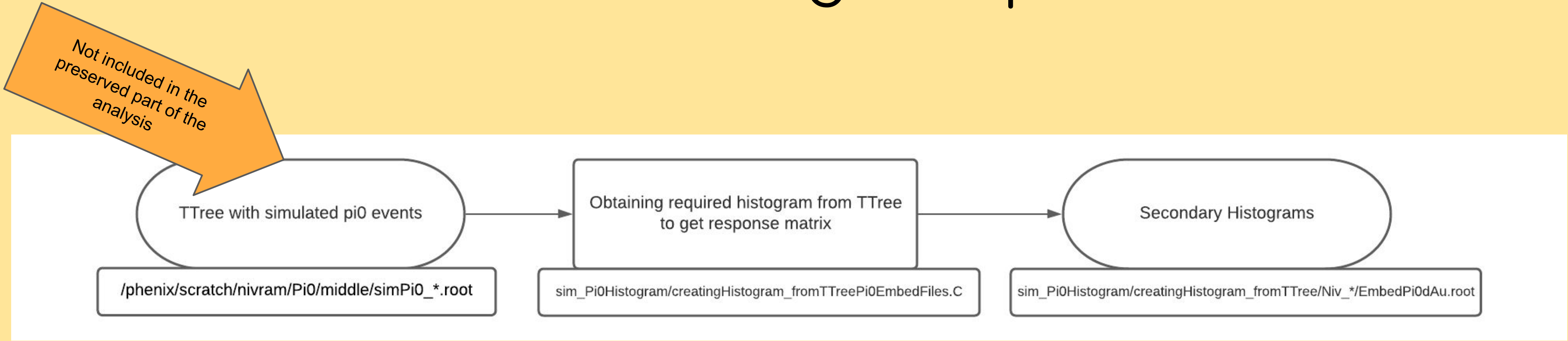
Designing a Workflow for Direct Photon Analysis

- Monolithic vs modular
- Monolithic – “fire and forget”. Easier to operate. Usefulness?
- Modular – more manual operations and flexibility
- Serial workflows are easy to code, but there is a component which involves parallel execution of embedding jobs (for performance reasons), originally written for Condor

The Parallel Part

- Spent time studying the syntax of CWL (one of the workflow languages supported in REANA which allows arbitrary graphs to be implemented).
- Note – the parallel step does not depend explicitly on any components of the overall workflow designated for preservation – see next slide
- Its result is a single file (after merge)

The embedding component



Not included in the preserved part of the analysis

TTree with simulated pi0 events

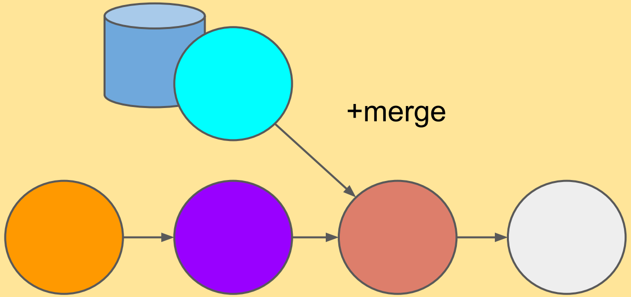
/phenix/scratch/nivram/Pi0/middle/simPi0_*.root

Obtaining required histogram from TTree to get response matrix

sim_Pi0Histogram/creatingHistogram_fromTTreePi0EmbedFiles.C

Secondary Histograms

sim_Pi0Histogram/creatingHistogram_fromTTree/Niv_*/EmbedPi0dAu.root



- 60 input files (preserved in our designated storage area)
- Run using Condor in the original software
- Can be run in REANA in different ways (CWL or otherwise)

An example of a simple workflow in CWL

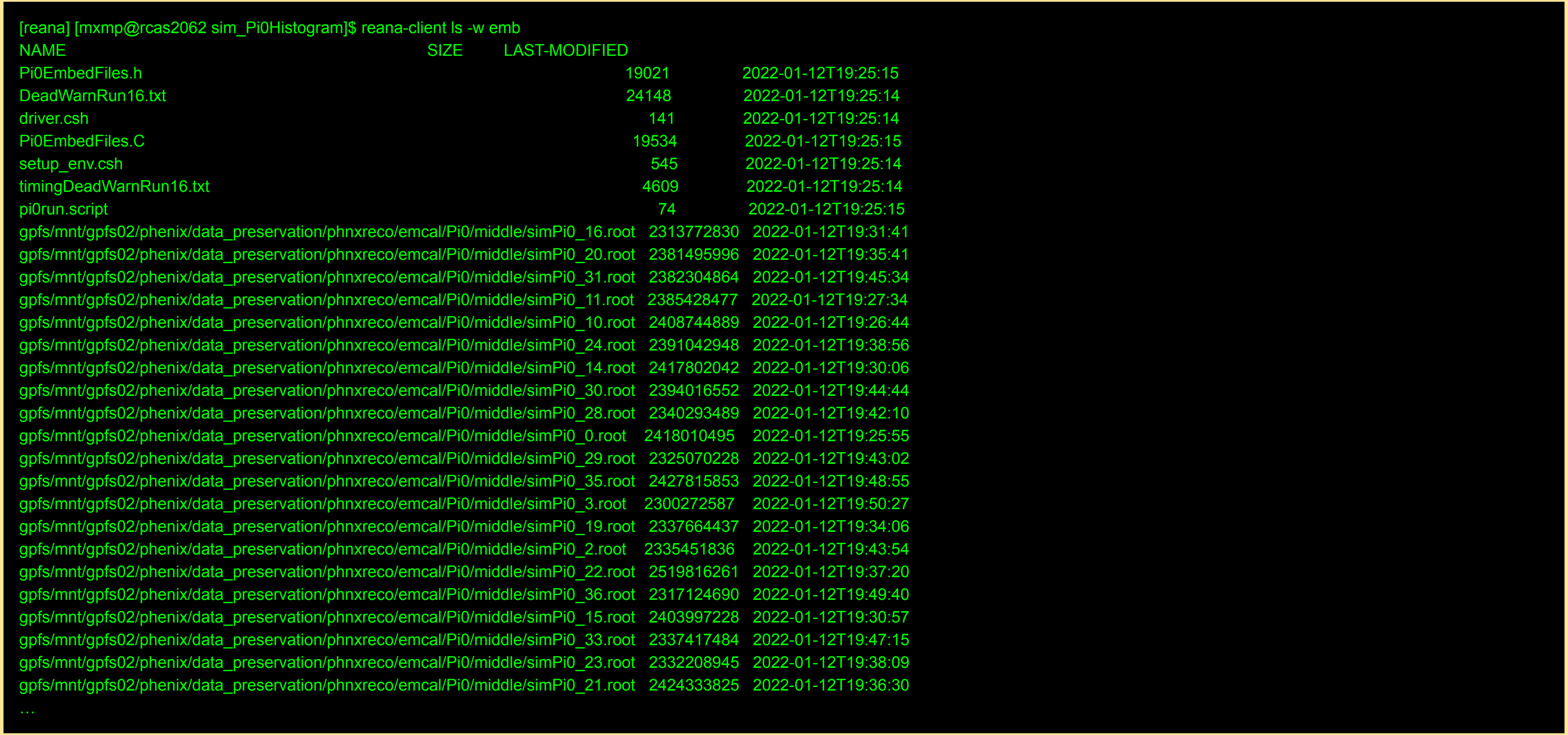
```
cwlVersion: v1.0
class: Workflow

requirements:
 ScatterFeatureRequirement: {}
 SubworkflowFeatureRequirement: {}
inputs:
  message_array: string[]
steps:
  subworkflow:
    run:
      class: Workflow
      inputs:
        message: string
      outputs: []
      steps:
        echo:
          run: scatter-tool-mod.cwl
          in:
            message: message
          out: [echo_out]
        wc:
          run: wc-tool.cwl
          in:
            input_file: echo/echo_out
          out: []
    scatter: message
    in:
      message: message_array
    out: []
outputs: []
```

Additional configuration elements

Conclusions regarding parallel workflows

- Complexity introduced by the CWL machinery does not pay off in this analysis
- Example: executables need to be pre-staged in the CWL env
- Unless actively used, it is hard to correctly deploy and maintain… solution –
- Unroll the Condor loop and execute jobs sequentially in REANA – at least for now
- Will be slower but still acceptable for demo and moderate use purposes: 9 min per job times 60
- In general, decided to stick with modular approach (with relatively complex modules)

Large scale upload, testing with REANA

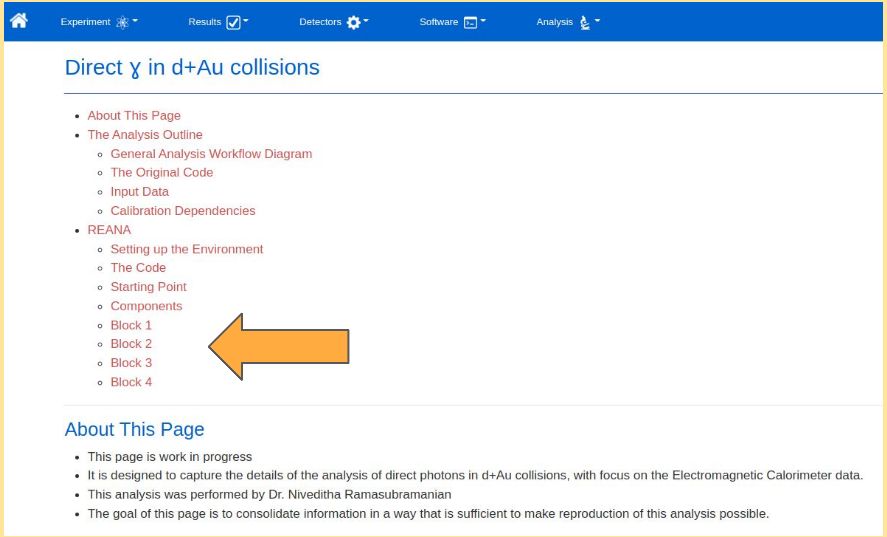


```
[reana] [mxmp@rcas2062 sim_Pi0Histogram]$ reana-client ls -w emb
NAME                                                        SIZE        LAST-MODIFIED
Pi0EmbedFiles.h                                                                 19021          2022-01-12T19:25:15
DeadWarnRun16.txt                                                               24148          2022-01-12T19:25:14
driver.csh                                                                        141          2022-01-12T19:25:14
Pi0EmbedFiles.C                                                                 19534          2022-01-12T19:25:15
setup_env.csh                                                                     545          2022-01-12T19:25:14
timingDeadWarnRun16.txt                                                          4609          2022-01-12T19:25:14
pi0run.script                                                                      74          2022-01-12T19:25:15
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_16.root  2313772830  2022-01-12T19:31:41
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_20.root  2381495996  2022-01-12T19:35:41
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_31.root  2382304864  2022-01-12T19:45:34
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_11.root  2385428477  2022-01-12T19:27:34
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_10.root  2408744889  2022-01-12T19:26:44
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_24.root  2391042948  2022-01-12T19:38:56
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_14.root  2417802042  2022-01-12T19:30:06
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_30.root  2394016552  2022-01-12T19:44:44
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_28.root  2340293489  2022-01-12T19:42:10
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_0.root  2418010495  2022-01-12T19:25:55
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_29.root  2325070228  2022-01-12T19:43:02
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_35.root  2427815853  2022-01-12T19:48:55
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_3.root  2300272587  2022-01-12T19:50:27
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_19.root  2337664437  2022-01-12T19:34:06
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_2.root  2335451836  2022-01-12T19:43:54
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_22.root  2519816261  2022-01-12T19:37:20
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_36.root  2317124690  2022-01-12T19:49:40
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_15.root  2403997228  2022-01-12T19:30:57
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_33.root  2337417484  2022-01-12T19:47:15
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_23.root  2332208945  2022-01-12T19:38:09
gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_21.root  2424333825  2022-01-12T19:36:30
…
```

Status

- Created a separate REANA workflow for the multi-input (previously parallel) part
- Some legacy scripting needed rewriting
- Scaling it up to multiple input files, each file ~4.5GB
- 260GB takes about 50 min to upload to the REANA cluster (sequentially)
- Updated the “direct gamma” web page to reflect these developments

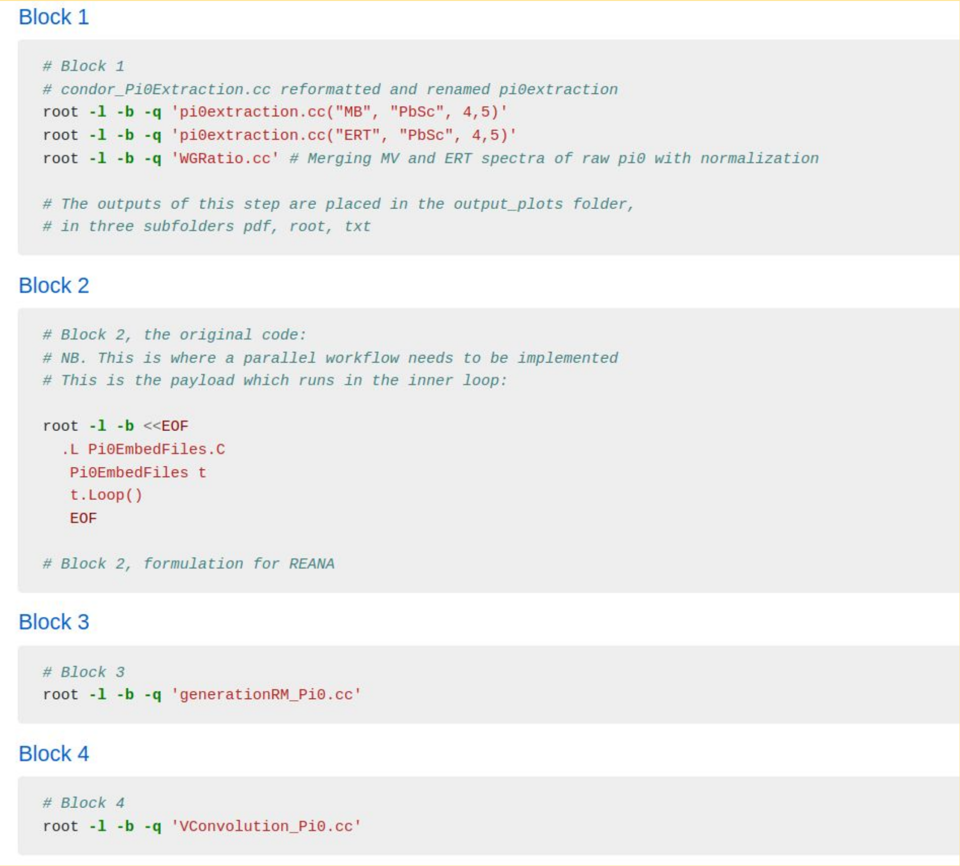
Webpage updates



Direct ɣ in d+Au collisions

About This Page

- This page is work in progress
- It is designed to capture the details of the analysis of direct photons in d+Au collisions, with focus on the Electromagnetic Calorimeter data.
- This analysis was performed by Dr. Niveditha Ramasubramanian
- The goal of this page is to consolidate information in a way that is sufficient to make reproduction of this analysis possible.



Block 1

```
# Block 1
# condor_Pi0Extraction.cc reformatted and renamed pi0extraction
root -l -b -q 'pi0extraction.cc("MB", "PbSc", 4,5)'
root -l -b -q 'pi0extraction.cc("ERT", "PbSc", 4,5)'
root -l -b -q 'WGRatio.cc' # Merging MV and ERT spectra of raw pi0 with normalization

# The outputs of this step are placed in the output_plots folder,
# in three subfolders pdf, root, txt
```

Block 2

```
# Block 2, the original code:
# NB. This is where a parallel workflow needs to be implemented
# This is the payload which runs in the inner loop:

root -l -b <<EOF
  .L Pi0EmbedFiles.C
   Pi0EmbedFiles t
   t.Loop()
   EOF

# Block 2, formulation for REANA
```

Block 3

```
# Block 3
root -l -b -q 'generationRM_Pi0.cc'
```

Block 4

```
# Block 4
root -l -b -q 'VConvolution_Pi0.cc'
```

…work in progress

Plans for REANA/Direct Photon

- Continue developing the remaining REANA components
- Have a run-through in 1–3 months
- Wrap it up