Sistemas de Monitoramento para o *Trigger* de Elétrons e Fótons no Experimento ATLAS/LHC

Edmar E. P. de Souza [1], Eduardo F. de Simas Filho [1], João V. F. Pinto [2], José M. de Seixas [2], Micael V. de Araújo [2], Paulo C. M. A. Farias [1]

[1]Laboratório de Sistemas Digitais, PPGEE/UFBA,
Universidade Federal da Bahia
[2]Laboratório de Processamento de Sinais, COPPE/POLI,
Universidade Federal do Rio de Janeiro

13 de julho de 2021

Sumário

Projeto: Física Experimental de Altas Energias e Tecnologias Associadas

- Colaboração ATLAS-Brasil: UFRJ, USP, UFBA, UFJF, UERJ e UFRN;
- Colaboração internacional com: Sorbonne Université e Universität Bern;
- Colaboração internacional com: Indiana University (EUA) e UNLP (ARG) e o grupo de Trigger do ATLAS (TrigEgamma Signature);
- Projeto CAPES/CONFECUB e bolsas FAPESB e FAPEMIG; Swiss National Science Foundation e CERN;
- Possíveis outras fontes de financiamento: CNPq, FAPESB, FAPERJ
- Uso de técnicas de monitoramento e avaliação de qualidade de dados em HEP;
- Inserção de HEP em fórum de inovação tecnológica;
- Empresas atuantes no CERN que surgiram desse projeto.
- Cronograma: Desenvolvimentos para atendimento às exigências da Run3;

Sistema de Filtragem Online

Porque um sistema de filtragem no ATLAS?

- Frequência de colisões em 40 MHz. Um evento de colisão apresenta $\approx$ 1,7 MB. A taxa de saída de dados total (sem *trigger*) seria de $\approx$ 70 TBytes/s.
- Nem tudo pode ser armazenado: Deve ser eficiente na física de interesse e ainda rejeitar a maioria dos eventos não relevantes.
- O que não é selecionado pelo trigger $\rightarrow$ **LOST FOREVER**.

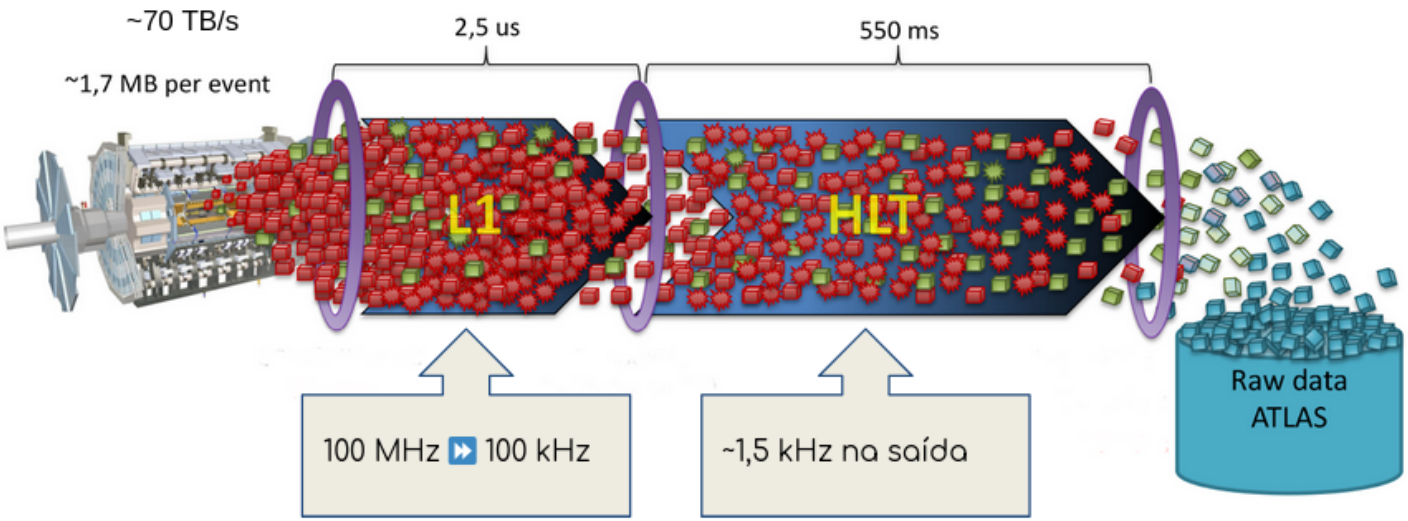
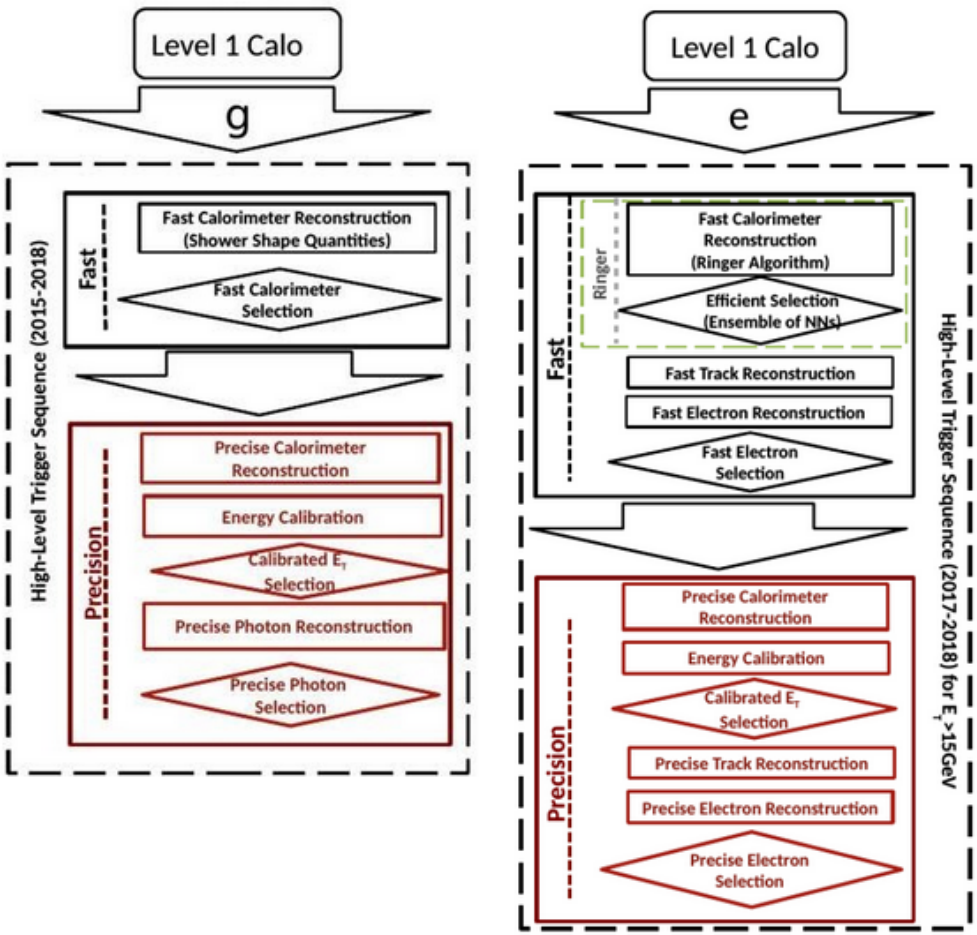


Diagrama das etapas do sistema de filtragem online no ATLAS.

Trigger de Elétrons e Fótons no ATLAS



- **Fast**: que utiliza os detectores com a resposta mais rápida para realizar uma seleção eficiente com uma baixa latência;
- **Precision**: nesta etapa são utilizados algoritmos mais complexos para tomar a decisão se um candidato é ou não aprovado;
- Algoritmos de hipóteses na etapa rápida operam com elevada eficiência de detecção (preservar física de interesse), e objetivam reduzir o quanto for possível aprovação de *background*.
- Na etapa precisa algoritmos de reconstrução baseados no ambiente offline, atuam para manter a eficiência de *trigger* elevada.

Desafios para a Run3

Condições mais acentuadas do que na Run2:

- Aumento de centro de massa de 13TeV para 14TeV;
- Aumento da luminosidade de pico: $2 \times 10^{34} cm^{-2} s^{-1}$ para $2.2 \times 10^{34} cm^{-2} s^{-1}$;
- Espera-se aumento na ocupação dos detectores e o consequente aumento do efeito de empilhamento de sinais;

Configurações do *Trigger* e/γ:

- Inúmeras cadeias de trigger são configuradas no *Menu* para diferentes propósitos do programa de física o ATLAS;
  - Cadeias de elétrons e fótons em cortes distintos de energia;
  - Cadeias para estudos de j/ψ
  - Cadeias com *triggers* combinados;
- Cadeias de trigger com diferentes *workpoints*;

Antes de serem certificados para análise física, os dados coletados devem ser examinados para garantir que estejam livres de qualquer problema proveniente de *hardware* ou *software*, que podem comprometer sua integridade.

Monitoramento no ATLAS

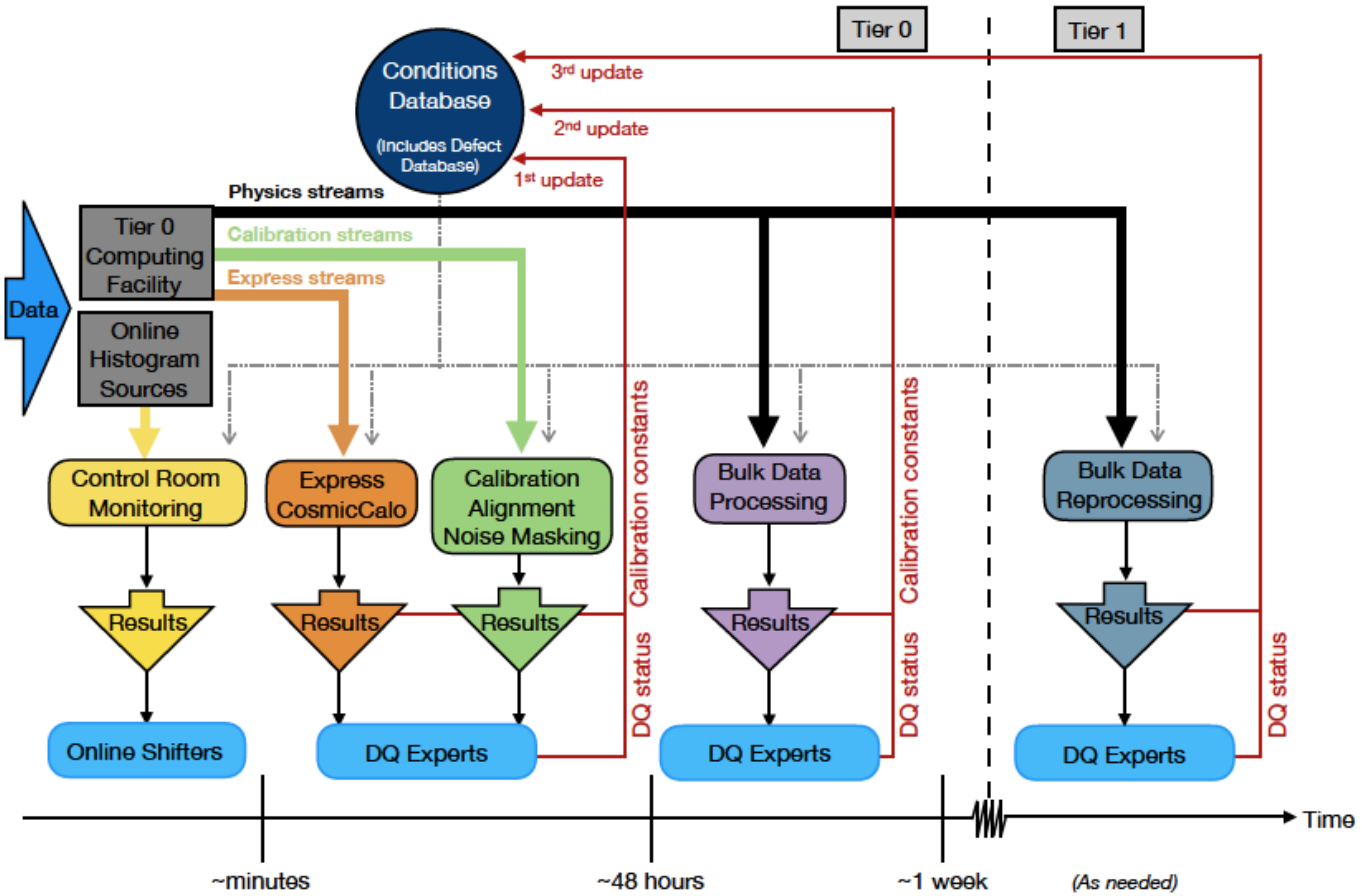
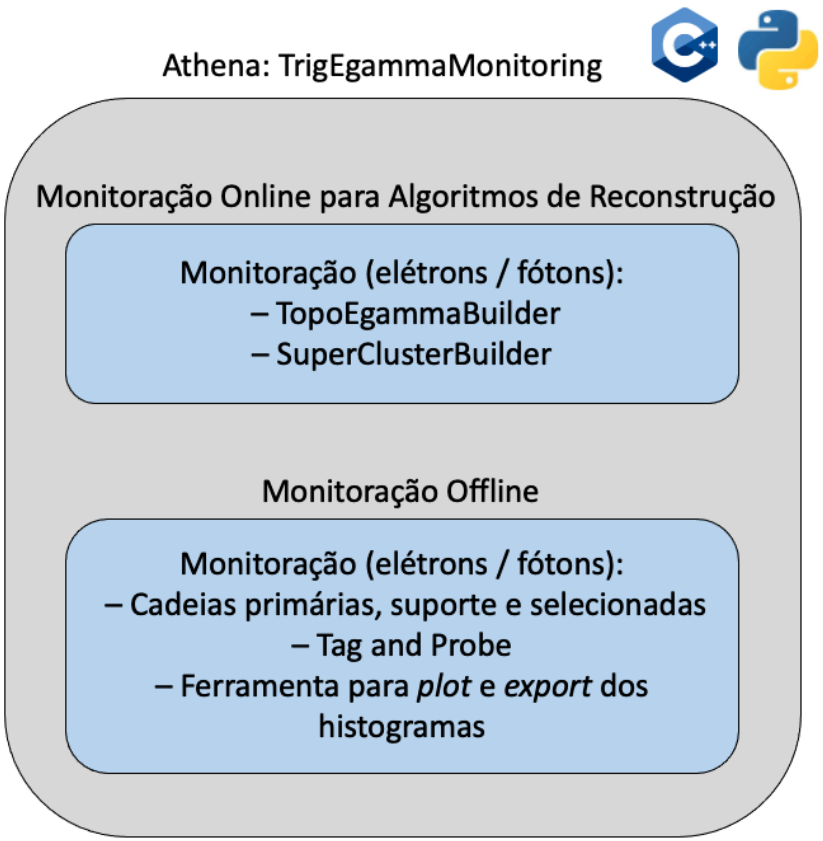


Diagrama esquemático que ilustra o fluxo de trabalho de operações da Run2 para DQ no ATLAS.

Framework TrigEgammaMonitoring

- Um ambiente de monitoramento dos algoritmos do HLT foi desenvolvido para a operação na *Run3* do LHC.



Esquemático do pacote TrigEgammaMonitoring e seus módulos integrados ao Athena.

Monitoramento Online

```
//================================================================

bool TrigEgammaFastElectronHypoTool::decide_cb ( const ITrigEgammaFastE

{

  if ( m_acceptAll ) {
    ATH_MSG_DEBUG( "AcceptAll property is set: taking all events" );
    return true;
  } else {
     ATH_MSG_DEBUG( "AcceptAll property not set: applying selection" );
  }

  auto cutCounter = Monitored::Scalar<int>( "CutCounter", -1 );
  auto ptCalo     = Monitored::Scalar( "PtCalo", -999. );
  auto ptTrack    = Monitored::Scalar( "PtTrack", -999. );
  auto dEtaCalo   = Monitored::Scalar( "CaloTrackEta", -1. );
  auto dPhiCalo   = Monitored::Scalar( "CaloTrackPhi", -1. );
  auto eToverPt   = Monitored::Scalar( "CaloTrackEoverP", -1. );
  auto caloEta    = Monitored::Scalar( "CaloEta", -100. );
  auto caloPhi    = Monitored::Scalar( "CaloPhi", -100. );
  auto trk_d0    = Monitored::Scalar( "d0 value", -1. );
  auto monitorIt  = Monitored::Group( m_monTool,
                                      cutCounter,
                                      ptCalo, ptTrack,
                                      dEtaCalo, dPhiCalo,
                                      eToverPt,
                                      caloEta, caloPhi, trk_d0);
```

```
  #
  # Monitoring code
  #
  def addMonitoring(self):

    from AthenaMonitoringKernel.GenericMonitoringTool import GenericMon
    monTool = GenericMonitoringTool("MonTool"+self.__name)
    monTool.defineHistogram('CutCounter', type='TH1I', path='EXPERT', t
    monTool.defineHistogram('CaloTrackdEta', type='TH1F', path='EXPERT',
    monTool.defineHistogram('CaloTrackdPhi', type='TH1F', path='EXPERT',
    monTool.defineHistogram('CaloTrackEoverP', type='TH1F', path='EXPER
    monTool.defineHistogram('PtTrack', type='TH1F', path='EXPERT', titl
    monTool.defineHistogram('PtCalo', type='TH1F', path='EXPERT', title
    monTool.defineHistogram('CaloEta', type='TH1F', path='EXPERT', titl
    monTool.defineHistogram('CaloPhi', type='TH1F', path='EXPERT', titl
    if self.tool().DoRinger:
      monTool.defineHistogram('NNOutput',type='TH1F', path='EXPERT',tit

    monTool.HistPath = 'FastElectronHypo/'+self.__name
    self.tool().MonTool = monTool



def _IncTool(name, cpart, tool=None):
  config = TrigEgammaFastElectronHypoToolConfig(name, cpart, tool
  config.compile()
  return config.tool()
```

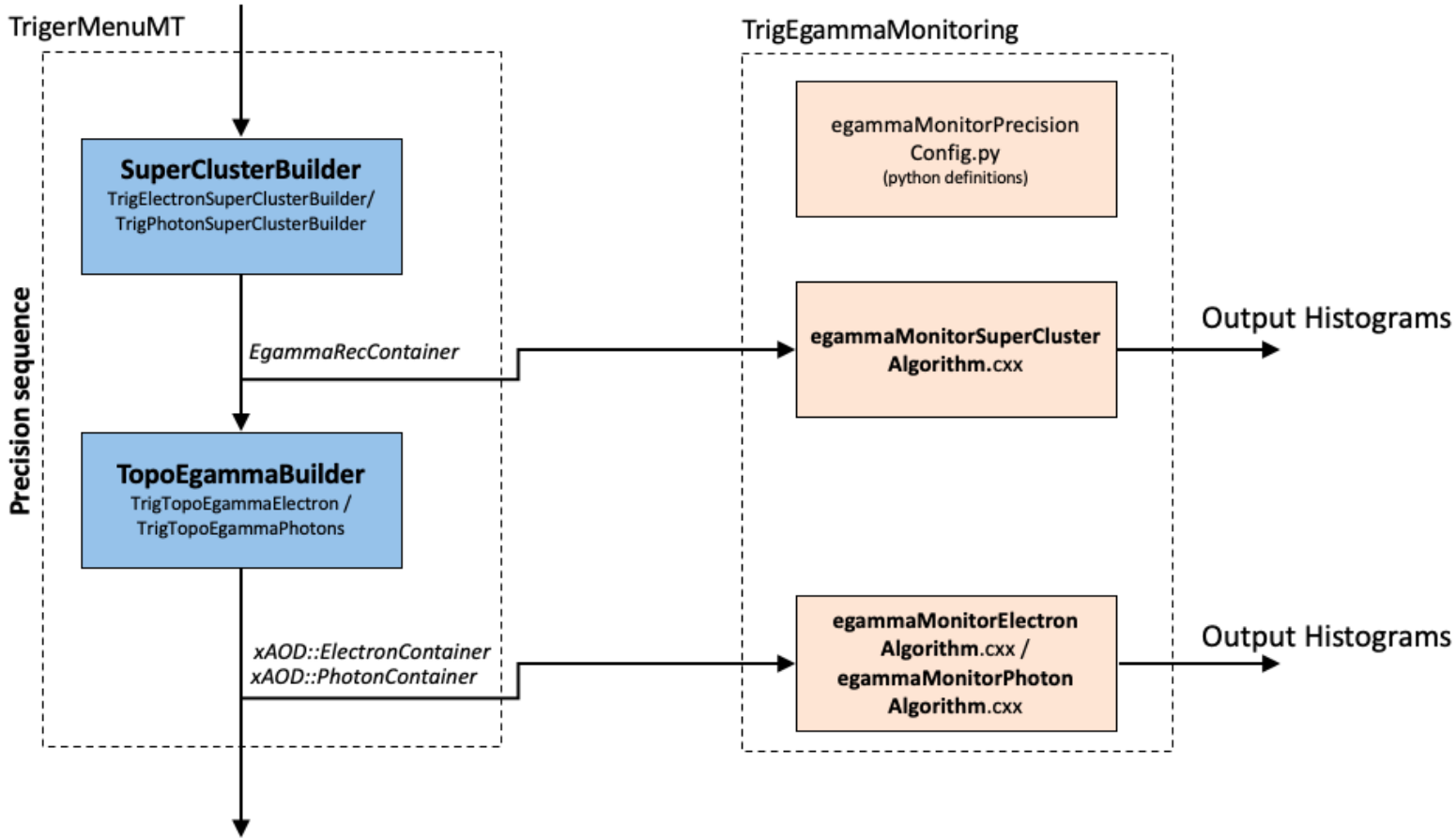
Monitoramento Online:

- São monitoradas as variáveis que descrevem o formato do chuveiro, variáveis de decisão dos algoritmos de hipóteses e saídas dos algoritmos de reconstrução (etapa precisa);
- Histogramas produzidos são disponibilizados na sala de controle do experimento;
- Desta forma, o monitoramento online permite problemas relacionados a hardware ou software sejam capturados durante a tomada de dados;

O monitoramento online é feito de dois modos no HLT:

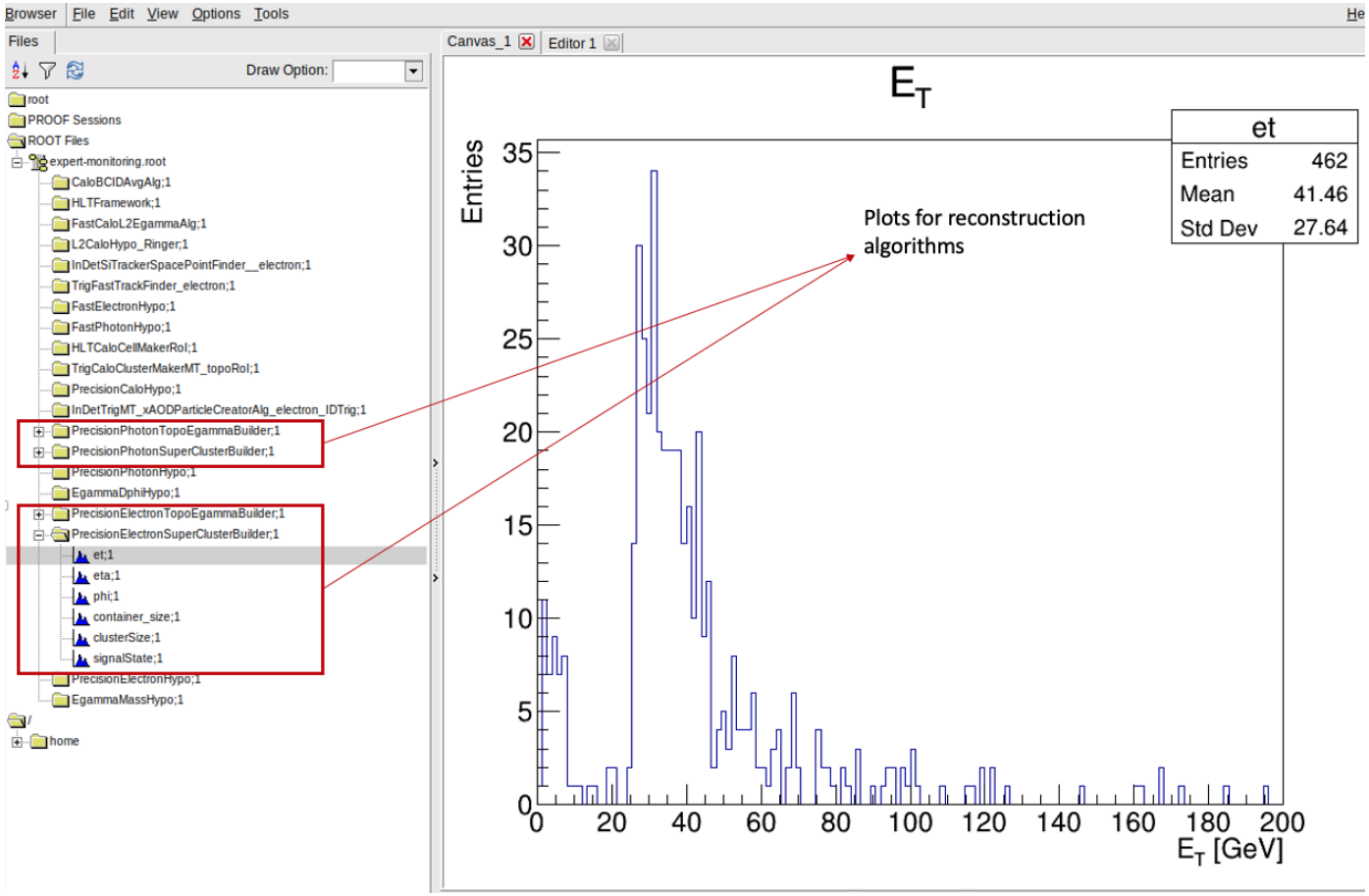
- Os valores das variáveis são coletados durante a execução do algoritmo;
- Algoritmos de monitoramento coletam a saída dos algoritmos de reconstrução e monitoram variáveis selecionadas;

Monitoramento Online dos Algoritmos de Reconstrução



Fluxo de operação da monitoramento dos principais algoritmos de reconstrução da etapa precisa.

Saídas do Monitoramento Online



Exemplo de um arquivo ROOT de saída, com os histogramas produzidos pelo monitoramento online nas diversas etapas do trigger.

Monitoramento Offline

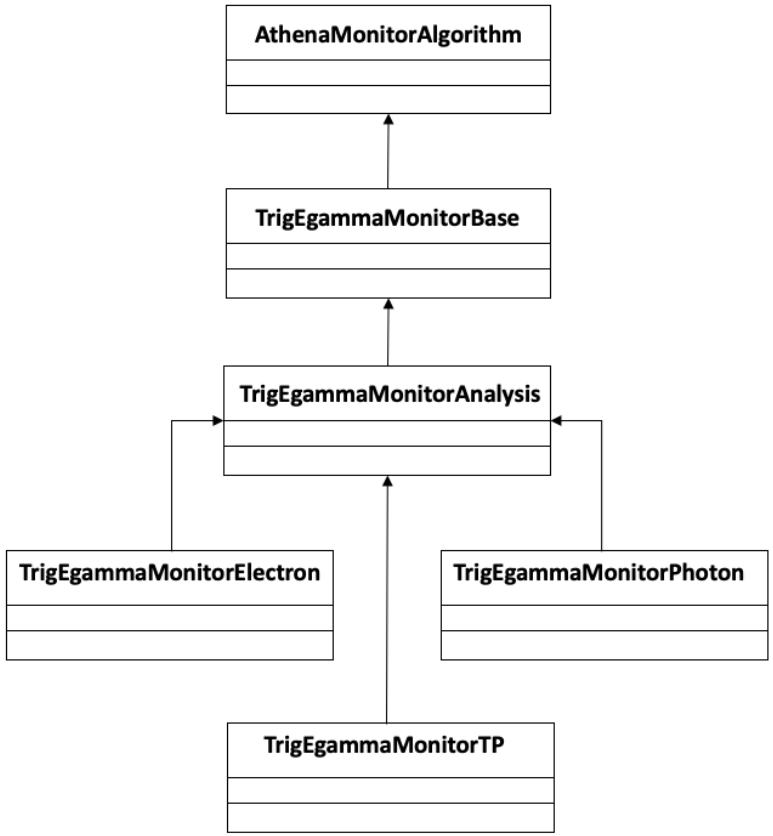
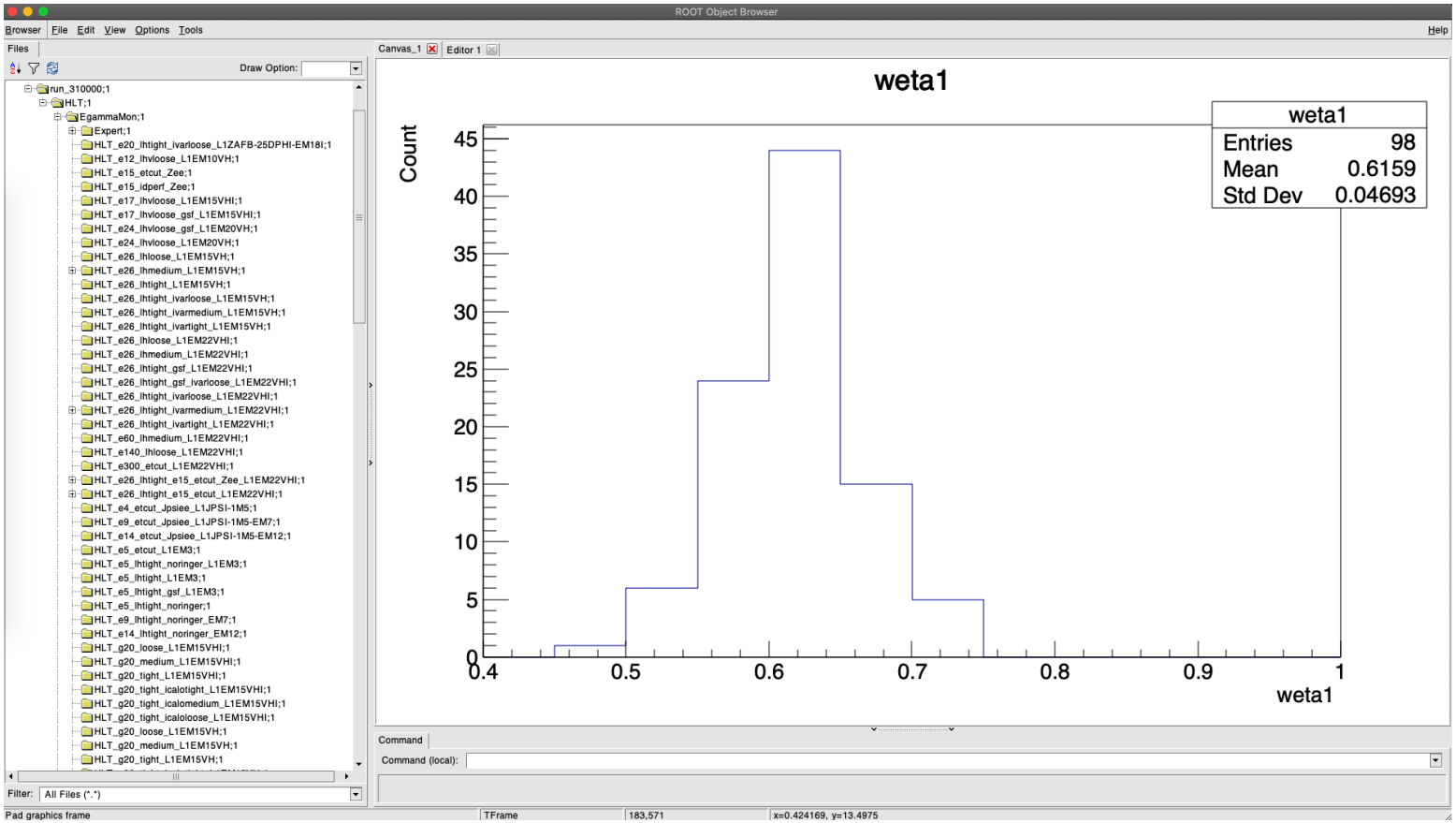


Diagrama de relacionamento dos algoritmos desenvolvidos no framework de monitoramento.

Monitoramento Offline:

- Projetado para realizar a monitoramento de diferentes cadeias e dar suporte a avaliação dos dados coletados durante as colisões ;
- Possui módulos distintos para monitorar cadeias de elétrons e fótons;
- Faz a interface com o *webdisplays* utilizados para qualidade de dados;
- Os algoritmos desenvolvidos funcionam com base em heranças, de propriedades das classes base do Athena.

Monitoramento Offline

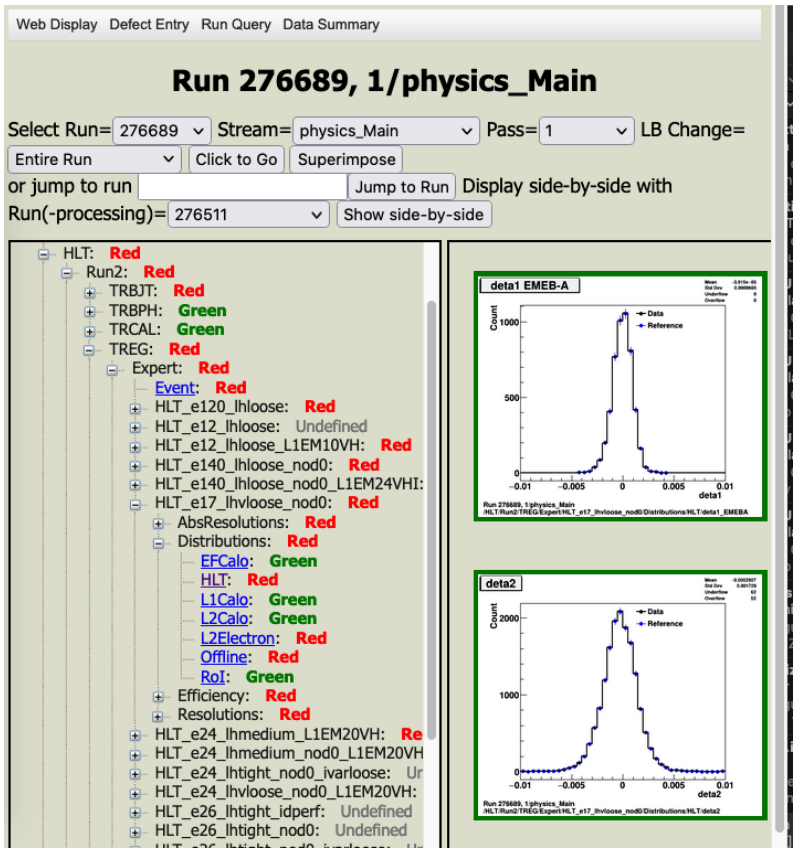


Exemplo de um arquivo ROOT de saída, com os histogramas produzidos pelo monitoramento offline para diferentes cadeias selecionadas.

Suporte na Avaliação de Qualidade de Dados

O framework desenvolvido da suporte ao processo de avaliação da qualidade de dados:

- Reprocessamentos são realizados periodicamente;
- Webdisplays com as cadeias selecionadas para análise são utilizados para avaliação;
- Distribuições são avaliadas para verificar se há algum problema no reprocessamento (novas releases do Athena, condições de operação, modificações em cadeias, etc);
- Os dados experimentais avaliados nos reprocessamentos, caso não apresentem problemas, são adotados como referência, em comparações posteriores nas próximas tomadas de dados;



Exemplo de um webdisplay gerado para análise de DQ em reprocessamento.

Considerações Finais

- O Framework está implementado no Athena, disponível para os propósitos de monitoramento do Trigger e/y;
- Os algoritmos de monitoramento tem sido atualizados com propriedades adicionais requeridas para a Run3 (inserção de novas cadeias, manutenção, etc);
- Uma maior campanha de reprocessamento tem previsão para iniciar em 08/2021;
- Projeto, implementação e manutenção sob responsabilidade do grupo ATLAS-Brasil inseridos no contexto Trigger e/y.

Obrigado!

Perguntas?