AOD ⇒ AOD Correction

Alfio Lazzaro
Università degli Studi and INFN, Milano
CERN, Geneva
alfio.lazzaro@mi.infn.it

Atlas Milano
January 30th, 2009

Introduction

- Goals:
  - Fast (ATLFASTII) to Full MC corrections
  - MC to data corrections
- Extract performances from data and MC
  - From their comparison we can extract the corrections
- Based on AOD files, applying corrections and making new AODs

MCReact

InsituPerformance

- In order:
  - First: Extract corrections

THEN

  - Apply corrections

Connection: my work

Extract Corrections

- Extracted from performances:
  - Efficiency
  - Resolution and scale factor
- In principle each object should have his corrections
  - Electrons, photons, pions, muons, jets...
- InsituPerformance provides a nice framework to do that:

  https://twiki.cern.ch/twiki/bin/view/AtlasProtected/InsituPerformance
- See talk by Matthias for more details:

  http://indico.cern.ch/getFile.py/access?contribId=9&sessionId=0&resId=1&materialId=slides&confId=51164
- Note that it extracts the performances directly from data

The database

- InsituPerformance package gives objects as output (essentially matrix and functions) which store the performances
  - for example the efficiency can be stored in a matrix as function of η, ϕ, p_T
- The objects are stored in ROOT files
- ROOT files can be linked in COOL database
- Very easy to access to the ROOT files and retrieve the performances
- For the moment I used directly the ROOT files, not using the COOL database

Apply Corrections

- Read the performances for two samples
  - For example Fast and Full simulations (same considerations can be applied in case of data)
- From the comparison extract the corrections and apply them to AOD objects
- The corrected objects saved in a new AOD file in a new container with different version, in order to keep the original objects (new, introduced in R14)
- MCReact (Marc) does that in case of MCfast to MCfull:

  https://twiki.cern.ch/twiki/bin/view/Atlas/MonteCarloReact
- Last update by Marc in this talk (25/11/2008):

  http://indico.cern.ch/materialDisplay.py?contribId=3&materialId=slides&confId=45962

MCReact: input data

- The performance from the experts for each object are provided as ASCII files with a specific format, and they are part of the package itself
  - the informations are saved in matrices (a matrix in each file)
    - For example, for electrons we have files for Tight, Medium, Loose criteria for efficiency, resolution and scale factors (of course for fast and full simulation)
  - MCReact has a list of experts for each object (see Marc's talk)
- Needs to replace those files with what InsituPerformance provides
  - Require some changes in the MCReact code, but overall needs a general agreement between the informations provided by InsituPerformance and what MCReact requires

Insitu ==> MCReact

- MCReact has been developed in general for MC to MC corrections
  - Next step is to move to MC to data corrections
- It could not be trivial
  - for example, resolutions and scale factors in MCReact are based on MCtruth informations (which are used to extract **matrices**). InsituPerformance provides **smearing functions** extracted from data, that you can directly use to correct MC
- Different procedure for MCfast=>MCfull and MC=>data
  - Needs some discussion between the experts of performance of MC and data (different experts)

Tests

- There are (not yet!) official performance files from InsituPerformance
- Based on 14.5.0 I did my personal ROOT files for the efficiency performance of electrons (using CVS Head code), using samples of $Z \rightarrow e^+e^-$ (just an exercise, nothing serious)
- I modified the MCReact code to read these files and apply the corrections
  - Same procedure can be considered for the other objects
- Not yet available a procedure to extract the resolutions and scale factors (see talk in egamma meeting last week:

  http://indico.cern.ch/materialDisplay.py?contribId=14&materialId=slides&confId=48800)

Conclusion

- First step is to have an official template of the corrections based InsituPerformance (defined as official one) package
  - More input from performance sub-groups
  - Milano (Attilio) is involved for the pions and muons resolution performance for Insitu
- All discussion in a wiki page and in the PAT HN
- Once we have an idea of the performance and corrections and how they are implemented, next step should be easy, i.e. modified MCReact to read those corrections
- For the moment I tried the code using Athena
  - Decide do not have a standalone version
- We need a general treatment of the systematics
  - In principle all corrections should have their errors, which we can use to extract a systematic effect of the corrections