Handling of systematics in CMS analyses

Adinda De Wit, Clemens Lange, Huilin Qu, Keith Ulmer, Lindsey Gray (co-chair), Mario Masciovecchio, Piergiulio Lenzi (co-chair), Pieter David, Sezen Sekmen, Sebastien Wertz

ex officio: Andrea Rizzi, Florencia Canelli, James Letts, Danilo Piparo, Mariarosaria D'Alfonso, Loukas Gouskos

HSF meeting - April 12th, 2022

General Picture

- **An evolving landscape**
- Physics Object Groups (POGs) **producing data/MC corrections and their uncertainties to apply to simulation to bring it in agreement with data**
- A **CrossPOG group** established since long time **to develop and support the NanoAOD data tier** and to **coordinate**, among other things, the **production and dissemination of the corrections/uncertainties**
- An **Analysis Tools Task Force** established at the end of 2021 to **review analysis practices** in Run2 analyses and make recommendations for the future, including the handling of systematics
- Technical application and propagation:
  - In Run 2 so far:
    - Corrections and uncertainties provided with non-uniform format by the POGs
    - Several frameworks, some general purpose ones, all sharing significant effort in the precise handling of systematics
  - Now and in the future:
    - Uniformation of the format for SFs and centrally supported library (correctionlib) for their application
    - Agreed need between coffea and RDF for low-level support for basic mechanics of handling systematics (i.e. making variations as computationally cheap as possible)

What is/isn’t stored in the NanoAOD data tier

- Gen level weights are stored, as they are analysis independent
  - PDF, scale, STXS classification, ISR/FSR
- Anything else that concerns the physics objects, and that is potentially analysis specific, is not stored in the NanoAOD, and it is evaluated by the analyzers
  - Simple example: even something as simple as lepton/jet cleaning is analysis specific → any systematics that depend on this cannot be stored in the NanoAOD
  - Plethora of WPs for the different algorithms, cannot store everything centrally, or it would explode in size

Legacy Run 2 input format for corrections

- No common format until recently
- Format defined by each Physics Object Group (POG)
  - With some proliferation of private correction formats for some analysis specific correction
  - Corrections held in Twikis, suboptimal
  - Versioning left to file naming
- **Each POG also providing the evaluators** (code or examples), documentation mainly twiki based
- **Contact persons** between the POGs and the analysis groups (PAGs) **reviewing** the application of the POG recommendations in each analysis

Application of corrections and systematic uncertainties in analysis with traditional ROOT

- Typical workflow involved running postprocessing code to add to NanoAOD additional branches containing corrections and their uncertainties
- The **main code used for this was, to some extent, maintained centrally,** but still on best effort basis only.
- One could either save entire copies of NanoAOD or just the varied/added branches exploiting friend trees when reading back.
- Different analysis frameworks came up with different ways of dealing with variations and the corresponding bookkeeping
    - Most effort going into systematics changing the event interpretation (e.g. those changing pTs…). **For those, we swap inputs and rerun everything**
- Moving towards new analysis frameworks able to process all at once but those are still new technologies

Ultra legacy run 2 input for corrections

- Standardized json format for all Physics Object Group corrections
  - Human and machine readable
  - Self-documenting
  - Obeying to a defined schema
- Evaluator code supported centrally (correctionlib)
  - C++ and Python bindings
  - Multidimentional lookup
  - Numpy vectorization support

Application of corrections and systematic uncertainties in analysis with RDF/Coffea

- Both frameworks have recent additions which allow handling of systematics
  - Both very generally allow users to add variations and create physics objects with varied features
- **Exploring the feasibility of all-in-one-go systematics**, without the need to save intermediate steps
- Final implementations and user interfaces still out for questions and trial by fire
  - Very likely the case that framework made atop RDF/coffea will design smooth interfaces for users
  - Objective of coffea/RDF to capture and make easily efficient core functionality of systematic variations
    - “How to vary something”, “what variations are there, how many sigma do they represent”
    - “Are these variations correlated with other variations?”

Concluding remarks

- General scheme is to correct the simulation and apply uncertainties to the correction. This is not changing.
- “The How” is changing:
  - Central format and generic evaluator centrally supported
  - Effort towards all-in-one-go processing of the input data without mandatory intermediate steps
    - With advantages in terms of space needs and analysis reproducibility
    - Still needs to be evaluated thoroughly
- Documentation and recommendations from the POG can be difficult to follow from the analyzers
  - Documentation can always be improved…

Backup

Scope

- How does your experiment "prescribe" the calculations of systematic uncertainties?
- Are there central tools to do that?
- How do analysers deal with those prescriptions?
- How much do they have to do manually?
- How do they solve bookkeeping, efficient evaluation?
- How does this tie into a group’s framework?
- Do you put multiple NTuples on disc and rerun everything exchanging the inputs etc?
- Where do the results of those calculations go / how do you save them?