SUSY Signal Production In ATLAS and CMS

Zach Marshall (LBNL)
HSF Generators Meeting
14 March 2019

Basics

- ATLAS and CMS use MadGraph5_aMC@NLO+Pythia8 for their SUSY signal event generation
- This work began to be able to *compare:*
  - Process definitions
  - MadGraph5_aMC@NLO settings (param and run cards)
  - Pythia8 settings (tunes)
- Such a comparison lets us understand whether our signal samples differ in meaningful ways
- It allows us the possibility to harmonize definitions (or cross-check and look for bugs)
- It allows us the possibility to share things down the line
- Comparisons here are by code inspection! Mistakes are my own!

Setups

- CMS software is public; SUSY files here
  - Documentation is in the code, and pretty clear (thanks!)
  - Comments here based primarily on go go, sq sq, t1 t1, C1 N2, and “Full Higgsino” production (not complete, but indicative)
- ATLAS software is becoming public, but not as clear to run for the public
  - Documentation is on (private) TWikis. Good luck figuring it out yourself :(
- Generally seems that:
  - ATLAS tries to do everything in the production system after the input cards are defined (including LHE production and potentially grid pack making)
  - CMS is more willing to have things prepared by hand, including grid packs
  - CMS don’t store LHE files (only gridpacks); in ATLAS depends on the point

Processes

- LO matrix elements with 0/1/2 extra jets √
- Intermediate off-shell particles differ
  - $ (no on-shell) vs / (none at all) syntax
- CMS always include b in p and j, ATLAS sometimes
  - Tested in ATLAS and found to have a tiny effect
- I don't see any other meaningful differences
  - Subtle differences in which particles are explicitly removed (CMS are more thorough)

Run Cards

- PDFs very subtly different (both use LHAPDF)
  - CMS NNPDF30_lo_as_0130; ATLAS NNPDF23_lo_as_0130_qed
  - CMS updating to NNPDF31, looking at a positive-definite NNLO PDF for the future
- Matching differs
  - CMS: MLM Durham-kT; xqcut=30 (default, then tuned)
  - ATLAS: CKKW-L; ktdurham=m/4; dparameter=0.4
  - CMS seem to check this much more thoroughly (or recommend to)
- Cuts differ
  - CMS use auto_ptj_mjj
  - ptj=30 (CMS) vs 20 (ATLAS); drjj=0.01 (CMS, but then set to 0) vs 0 (ATLAS); draj=0.1 (CMS) vs 0.4 (ATLAS). Generally ATLAS using more defaults.
- Difference in maxjetflavor
  - CMS=5; ATLAS=4; expected from b jets in p / j
- We both try to enable use of systematics during running

Param Cards

- Both experiments work from ‘template’ param cards that are modified according to the masses of interest
- Off shell masses differ (100 TeV vs 450 TeV) but irrelevant
- Widths set differently, but largely irrelevant (provided decays are prompt or long-lived as desired)
- Difference in mixing matrices and other parameters
    - CMS sets them rarely (only Higgsinos?)
    - ATLAS does tend to set them whenever they might be relevant
    - This should be a very small effect for the most part, but matters when looking for example at edges in electroweak models

MadSpin

- ATLAS rely on MadSpin heavily for stop and electroweakino decays (especially off-shell)
- In extreme off-shell scenarios (small mass splittings), it's not obvious that Pythia will do the right thing with branching fractions (at least this is a potential difference)
- CMS don't use MadSpin for decays, but set the minimum W/Z mass to 0.1 GeV and cross-check with decays in MadGraph

Pythia8 / Showering

- Both experiments use Pythia8 for showering; settings differ, tune differs
- CMS:

```
JetMatching:setMad = off
JetMatching:scheme = 1
JetMatching:merge = on
JetMatching:jetAlgorithm = 2
JetMatching:etaJetMax = 5.
JetMatching:coneRadius = 1.
JetMatching:slowJetPower = 1
JetMatching:qCut = $QCUT.
JetMatching:nQmatch = 5
JetMatching:nJetMax = 2
JetMatching:doShowerKt = off
```

- ATLAS:

```
Merging:doKTMerging = on
Merging:Dparameter = 0.4
Merging:mayRemoveDecayProducts = on
Merging:nJetMax = 2
Merging:nQuarksMerge = 4
Merging:Process = guess
Merging:TMS = $ktdurham
```

Sharing

- Long term, if we want to share, we have some work to do
- Need to harmonize / agree upon all settings, at least up to LHE generation (more if we share further down the chain)
  - Harmonizing does not mandate sharing, but sharing prevents later divergence
- Settle on a format to share (gridpack, LHE, HepMC)
  - CMS not storing LHE files right now; neither storing HepMC in a common format
- Need to settle on a mechanism for sharing
  - My favorite is sharing rucio datasets if we both use rucio; this avoids the problem of disk allocations. Need to figure out rights.
- In my view, we should focus on harmonization for now, but it would be great if this group could help explore the technical issues so they're solved if/when we get there!
- Many big Run 2 productions are already done. We could migrate signals continuously to a harmonized setup, but if we started today this would likely not hit all published analyses before Moriond 2020 at the earliest.

Bonus: LLPs

- CMS LLP code lives here , ATLAS here
- R-hadrons hadronic interaction code is common and public
  - Checking on configuration of the interaction model
- Both groups use G4hMultipleScattering and G4hIonisation
- Still checking on the production of R-hadrons (e.g. masses)

Summary

- Beginning to look into settings in ATLAS and CMS SUSY signal generation
- Goals: identify differences, explore impact of differences, consider harmonization, consider sharing of data
- Some differences (of course), but many similarities
    - Differences in processes seem straightforward and trivial
    - Differences in run cards mostly choice of matching setup
    - Differences in Pythia8 setup to do with matching and tune
- Next: Check with others to see if I've identified the differences correctly, ask about **versions** in use, then look into what impact these differences have (if any) on the output!
    - If differences are due to ATLAS or CMS standards, harmonization may be very difficult.
    - Also: look into long-lived particle setups