Table of Contents

Mixing Signal Events into Heavy Ion Background......1

Goal of this page......1

Code and tags......1

Quick Instructions......1

Using cmsDriver.py......1

Writing a custom configuration file......1

Other possibilities you may need......2

Mixing in Gen-Level only......2

Running NoPileUp on hiSignal......3

Configuration Examples......4

Related Links......4

Contact......4

Review status......4

Mixing Signal Events into Heavy Ion Background

Complete: 🟩

Goal of this page

The aim of this page is to provide instructions for generating, simulating and mixing a signal event into a heavy ion background event

Code and tags

The code which does the mixing has been incorporated into the standard MixingModule, which lives in SimGeneral/MixingModule, since release CMSSW_3_3_0. There is no need for additional CVS tags.

Quick Instructions

The configuration file (cfg) for a heavy-ion mixing job can be created in two ways, by either using the cmsDriver.py command, or writing the cfg yourself.

Using cmsDriver.py

You can create a configuration file with the following command:

```
cmsDriver.py Pyquen_DiJet_pt80to120_4TeV_cfi.py \
--himix --scenario CMS.HeavyIons \
-s GEN:hiSignal,SIM,DIGI,L1,DIGI2RAW,RAW2DIGI,RECO \
--conditions FrontierConditions_CMS.GlobalTag,MC_31X_V9::All \
--datatier 'GEN-SIM-RAW-RECO' --eventcontent=FEVTDEBUG \
--processName 'HISIGNAL' \
--filein=inputfile.root --fileout=outputfile.root \
 -n 1 \
--no_exec
```

This configuration mixes dijets into heavy ion background, and runs up to RECO. The signal event is specified in the input cfi file (here `Pyquen_DiJet_pt80to120_4TeV_cfi.py`). The background event into which it should be mixed is specified by the `--filein` parameter. Further details to nore are:

- In the input generator cfi, (`Pyquen_DiJet_pt80to120_4TeV_cfi.py` in the example above), the generator module has to be defined with label "hiSignal", rather than "generator".
- The GEN sequence must be run with the option "hiSignal", by typing it as "GEN:hiSignal" among sequences.
- The option "--himix" must be used.
- Scenario "CMS.HeavyIons" must be specified.
- A process name (of your choice) has to be specified in order not to clash with previous processes that ran on the events.

You can modify all other parameters as you like. See SWGuidePyReleaseValidationOptions for cmsDriver.py details.

Writing a custom configuration file

- Create a PoolSource that opens an input (heavy ion background) file. Make sure you drop the previously produced DIGI, RAW etc. of the background event. You need only the GEN and SIM info from this file:

```
process.source = cms.Source("PoolSource",
    fileNames = cms.untracked.vstring('inputfile.root'),
    inputCommands = cms.untracked.vstring('drop *',
        'keep *_generator_*_*',
        'keep *_g4SimHits_*_*'),
    dropDescendantsOfDroppedBranches = cms.untracked.bool(False)
)
```

- Load a generator particularly configured for mixing, for example:

```
process.load('Configuration.Generator.Pyquen_CMS.GammaJet_pt20_4TeV_cfi')
```

Or, define a generator module from scratch:

```
process.hiSignal = cms.EDFilter('PyquenProducer',....)
```

- Load the standard sequences for your job, including the standard GEN, SIM, DIGI stuff:

```
process.load('Configuration/StandardSequences/Services_cff')
process.load('Configuration/StandardSequences/Generator_cff')
process.load('Configuration/StandardSequences/Sim_cff')
process.load('Configuration/StandardSequences/Digi_cff')
```

and then add the extra sequences needed for hi-mixing:

```
process.load('Configuration/StandardSequences/HiEventMixing_cff')
process.load('SimGeneral/MixingModule/himixGEN_cff')
process.load('SimGeneral/MixingModule/himixSIMExtended_cff')
process.load('SimGeneral/MixingModule/himixDIGI_cff')
```

Make sure you load the hi-mix cff files AFTER you load the standard ones in your cfg!

- Run the standard sequences, except pgen_himix instead of pgen_hi.

```
process.generation_step = cms.Path(process.pgen_himix)
process.simulation_step = cms.Path(process.psim)
process.digitisation_step = cms.Path(process.pdigi)
...
```

Other possibilities you may need

Mixing in Gen-Level only

For some studies it may be sufficient to mix the signal and background events at the generator level, rather than the SIM level as for the cases above. To do this

- Load only standard GEN sequences.

```
process.load('Configuration/StandardSequences/Services_cff')
process.load('Configuration/StandardSequences/Generator_cff')
```

- For mixing, use mixGenNoPU from HiEventMixing, **not** by loading; but **importing**.

```
from SimGeneral.MixingModule.HiEventMixing_cff import *
process.mix = mixGenHI
...
process.pmix = cms.Path(process.mix)
```

Running NoPileUp on hiSignal

You may want to take the already simulated signal events, which have been mixed into HI background and reconstructed, and run RECO on them without the heavy ion background. It is possible to save the RECO output of both mixed and signal-only event in the same file. Here are the instructions to do it:

- For now (CMSSW_3_3_X or 3_4_0_preX) , you need to check out the HEAD of SimGeneral/MixingModule from the CVS.

```
cvs co SimGeneral/MixingModule
scramv1 b
```

- In your configuration file, create a PoolSource that drops the DIGI, RAW, and local RECO of the input file.

```
process.source = cms.Source("PoolSource",
    fileNames = cms.untracked.vstring('inputfile.root'),
    inputCommands = cms.untracked.vstring(
        'drop *_*_*_DIGI',
        'drop *_*_*_RAW',
        'drop *_*_*_RECO'),
    dropDescendantsOfDroppedBranches = cms.untracked.bool(False)
)
```

- Load only standard GEN, SIM, DIGI sequences.

```
process.load('Configuration/StandardSequences/Services_cff')
process.load('Configuration/StandardSequences/Generator_cff')
process.load('Configuration/StandardSequences/Sim_cff')
process.load('Configuration/StandardSequences/Digi_cff')
```

- For mixing, load HiEventMixing and change parameters to use only the signal:

```
process.load("SimGeneral.MixingModule.HiEventMixing_cff")
process.mix.srcGEN = ["hiSignal"]
process.mix.srcSIM = ["hiSignalG4SimHits"]
```

- Run standard DIGI, L1, RAW, RawToDigi sequences.

```
process.p = cms.Path(process.mix*process.doAllDigi*process.L1Emulator*process.DigiToRaw*process.R
```

If you want to keep both Mixed and Signal-only reconstruction objects, then you need new names for the Signal-only RECO objects. This is also easy:

- Load standard RECO for heavy ions, but **AFTER** that, load the extra reco for hiSignal.

```
process.load('Configuration/StandardSequences/ReconstructionHeavyIons_cff')
process.load('RecoHI.Configuration.Reconstruction_hiSignal_cff')
```

This will modify the sequences defined in the standard configuration.

- Run the standard reco sequence:

```
process.reconstruction_step = cms.Path(process.reconstructionHeavyIons)
```

Configuration Examples

The directory UserCode/CmsHi/Utilities/scripts contains some scripts with correctly-formed cmsDriver.py commands for mixing, e.g.

- mixDiJet_reco.sh
- mixZmumuJets_reco.sh
- mixGammaJet_reco.sh
- mixGammaJet.sh (up to RAW only)

Related Links

- SWGuideMixingModule

Contact

- **Hypernews fora**:
    - Software issues: https://hypernews.cern.ch/HyperNews/CMS/get/hiswDevelopment.html, (hn-cms-hiswDevelopment@cernNOSPAMPLEASE.ch)
    - General heavy ions issues https://hypernews.cern.ch/HyperNews/CMS/get/hi.html, (hn-cms-hi@cernNOSPAMPLEASE.ch)
- **Contacts/Developers**: Yetkin Yilmaz, Edward Wenger

Review status

Reviewer/Editor and Date	Comments
YetkinYilmaz - 29 Oct 2009	updated contents as part of documentation review

Responsible: YetkinYilmaz

This topic: CMSPublic > SWGuideHeavyIonEvtMixing
Topic revision: r6 - 2009-10-30 - PhilipAllfrey

Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback