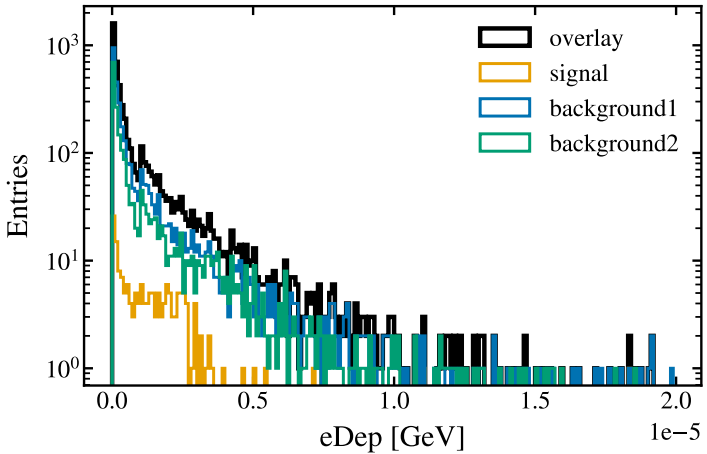
The Background Overlay Algorithm in Key4hep

Juan Miguel Carceller

CERN

July 10, 2024

Background Overlay



Background Overlay: Summary

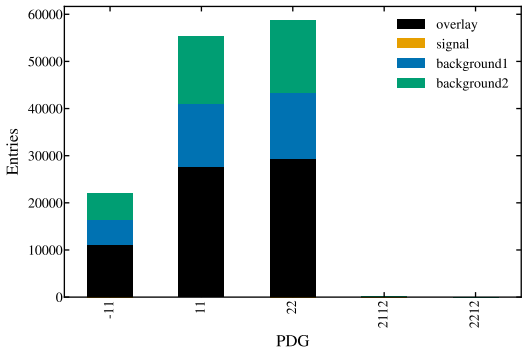
- We have a set of background events that we want to overlay on signal events
- After simulation (after `ddsim`)
- Takes 3 inputs:
  - **MCParticles**: from background and signal are all overlayed
  - **SimTrackerHits**: are overlayed if they are in a certain time window
  - **SimCalorimeterHits**: are overlayed only if they have contributions in a certain time window. If a signal hit and a background hit have the same cellID, they are combined into a single hit

Background Overlay: Summary

- For MCParticles and SimTrackerHits, there is an `isOverlay()` method that tells us if they come from background or signal
- Relations in the new objects point to the new objects: a SimTrackerHit from signal will point to the corresponding MCParticle in the overlayed collection, the same for background
  - Consistent set of collections without any relations to the original signal or background objects
- Pull request in key4hep/k4Reco (the starting point was the `OverlayTiming` processor from iLCSoft)
- How does it work? The event loop in Gaudi processes each signal event, then background events are read from files

MCParticles

- Setup: signal (2 electrons) file and two background files (IDEA)



PDG	signal	background1	background2	sum	overlay
-11	0	5359	5688	11047	11047
11	2	13341	14352	27695	27695
22	0	14081	15282	29363	29363
2112	0	30	45	75	75
2212	0	3	4	7	7

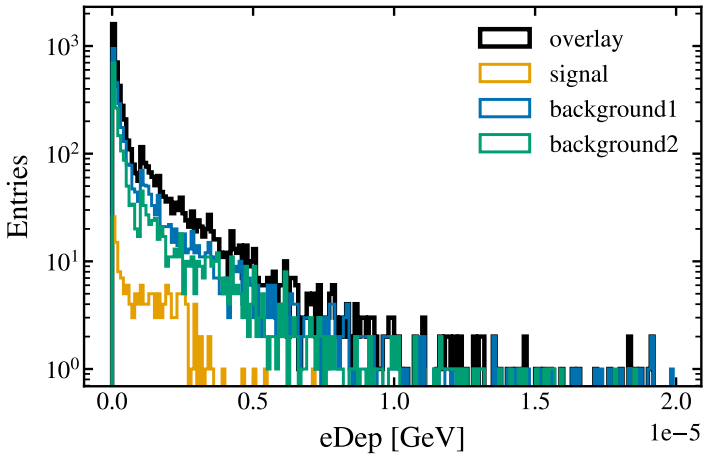
- Good news: The sum and what is found in the Overlay file are the same

MCParticles: Other checks

- Number of parent and daughter MCParticles is the same as originally for each MCParticle
- Daughters and parents point to the same MCParticles they pointed originally

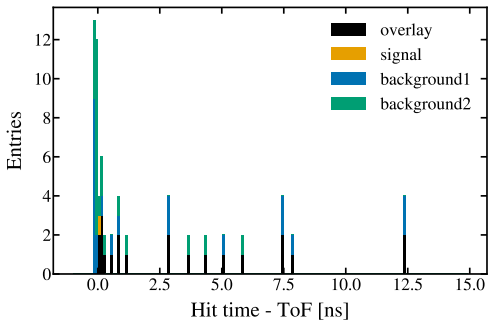
SimTrackerHits

- All SimTrackerHits are overlayed if inside the integration window



SimTrackerHits

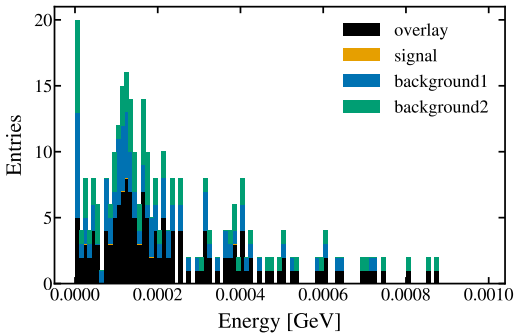
- Let's check the integration window (in this case from 0 to more than 20)
- There aren't any hits below 0
- Stacked histogram



signal	background1	background2	sum	overlay
1	22	23	46	21

SimCalorimeterHit

- Stacked histogram



signal	background1	background2	sum	overlay
0	86	78	164	149

SimCalorimiterHit

- To be checked:
  - Unique cellIDs for Overlay hits?
    - Checked with toy data
  - Are the CaloHitContributions the expected ones?
    - Also checked with toy data

Steering file

```
from k4FWCore import ApplicationMgr
from k4FWCore import IOSvc
from Configurables import EventDataSvc
from Configurables import OverlayTiming
from Configurables import UniqueIDGenSvc

id_service = UniqueIDGenSvc("UniqueIDGenSvc")
eds = EventDataSvc("EventDataSvc")

iosvc = IOSvc()
iosvc.input = "input.root"
iosvc.output = "output_overlay.root"

overlay = OverlayTiming()
overlay.MCParticles = ["MCParticle"]
overlay.SimTrackerHits = ["VertexBarrelCollection", "VertexEndcapCollection"]
overlay.SimCalorimeterHits = ["HCalRingCollection"]
overlay.CaloHitContributions = ["CaloHitContributionsCollection"]
overlay.OutputSimTrackerHits = ["NewVertexBarrelCollection", "NewVertexEndcapCollection"]
overlay.OutputSimCalorimeterHits = ["NewHCalRingCollection"]
overlay.OutputCaloHitContributions = ["NewCaloHitCollection"]
# overlay.StartBackgroundEventIndex = 0
overlay.BackgroundFileNames = [
    ["/Overlay/background1.root"],
    ["/Overlay/background2.root"],
]
overlay.TimeWindows = {"MCParticle": [0, 23.5], "VertexBarrelCollection": [0, 23.5], "VertexEndcapCollection": [0, 23.5], "HCalRingCollection": [0, 23.5]}

ApplicationMgr(TopAlg=[overlay],
               EvtSel="NONE",
               EvtMax=10,
               ExtSvc=[eds],
               OutputLevel=INFO,
               )
```

Issues and current status

- Implementation was not trivial because initially there wasn't support for reading an arbitrary number of collections
- In addition, there was a bug in Podio (#583 and #632) that made relations unusable in cloned objects
- Current status: in a PR but also needs k4FWCore/201
  - A few details about randomisation of the background files may need to be sorted out
  - Implement caching of background events if they are reused since most time seems to be spent on reading background events (to be checked)
  - Possibility of adding histograms with information about the overlay process

Summary

- Background Overlay algorithm ready to be tested (and reviewed)
- Looks OK so far, still need to check some details
- Plan to use this in the CLD reconstruction
  - Some non-trivial questions to solve: Overlay in Key4hep will always produce new collections (unlike in iLCSoft, where they can be modified in place)
  - Overlay will have to always run so that switching it off doesn't mean changing the names of the collections - it should be easy to switch on and off

Backup

CaloHitContributions: Fake data

- Merging:
  - Signal: 500 hits with cellID from 0 to 499
  - Background1: 100 hits with cellID from 0 to 99
  - Background2: 200 hits with cellID from 400 to 599
  - All with one contribution for each hit