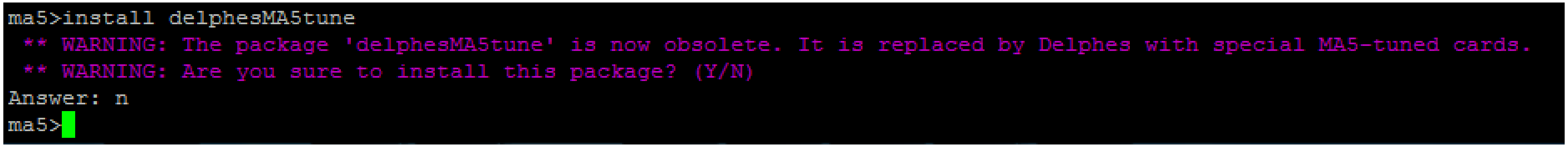
Delphes-based detector simulation in MadAnalysis 1.2

Update : 2015/07/11 - 21:00

General

Package Delphes-MA5tune is obsolete now.
Conserved temporary for backward-compatibility.



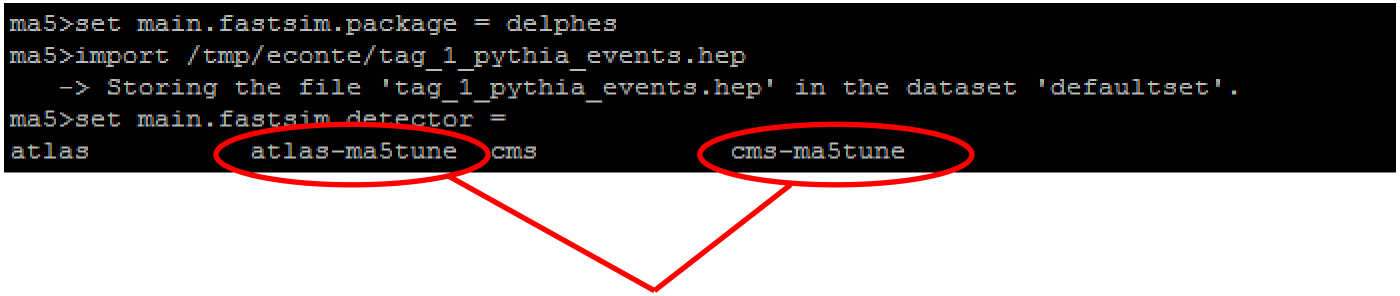
```
ma5>install delphesMA5tune
 ** WARNING: The package 'delphesMA5tune' is now obsolete. It is replaced by Delphes with special MA5-tuned cards.
 ** WARNING: Are you sure to install this package? (Y/N)
Answer: n
ma5>
```

Advice: please do not use DelphesMA5tune

→ Former «DelphesMA5-tune» package is replaced by Delphes with special cards.

How to launch MA5 with these specials cards?

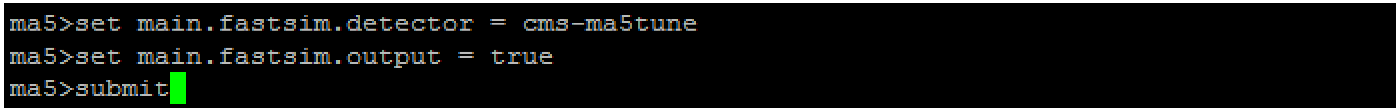
./bin/ma5 -R



```
ma5>set main.fastsim.package = delphes
ma5>import /tmp/econte/tag_1_pythia_events.hep
   -> Storing the file 'tag_1_pythia_events.hep' in the dataset 'defaultset'.
ma5>set main.fastsim.detector =
atlas          atlas-ma5tune  cms            cms-ma5tune
```

MA5tune detectors

Remaining instructions are indentical to the last release



```
ma5>set main.fastsim.detector = cms-ma5tune
ma5>set main.fastsim.output = true
ma5>submit
```

NB: INFO messages coming from Delphes are now vetoed by MA5

Data format comparison: Delphes card vs Delphes-MA5tune card

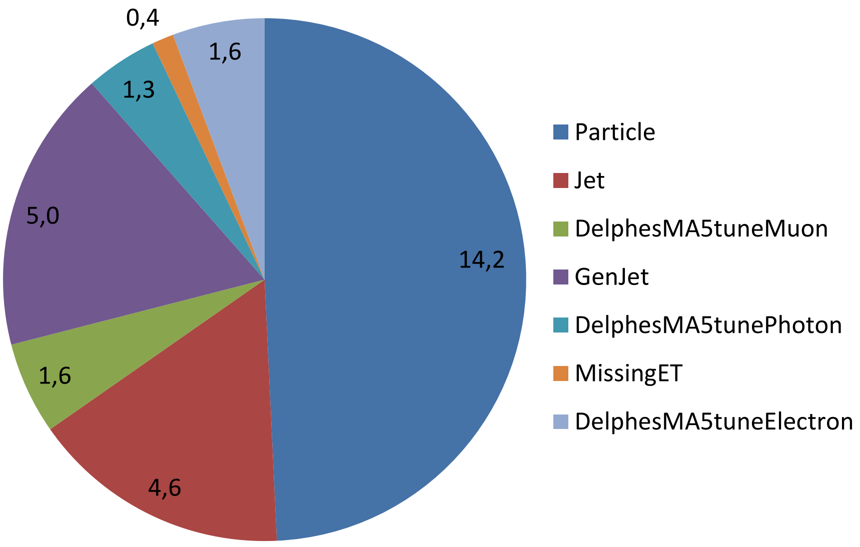
	Typical CMS Delphes Card	CMS MA5tune Card
Gen info	Particle	Particle
	GenJet	GenJet
Primary objects	Track	Track
	Tower	Tower
	EFlowTrack	EFlowTrack
	EFlowPhoton	EFlowPhoton
	EFlowNeutralHadron	EFlowNeutralHadron
Final objects	MissingET	MissingET
	ScalarHT	
	Jet	JetMA5
	Muon	MuonMA5
	Electron	ElectronMA5
	Photon	PhotonMA5

Data format comparison: old DelphesMA5tune vs new Delphes+MA5card

Test sample: drell-yan (+2 jets)

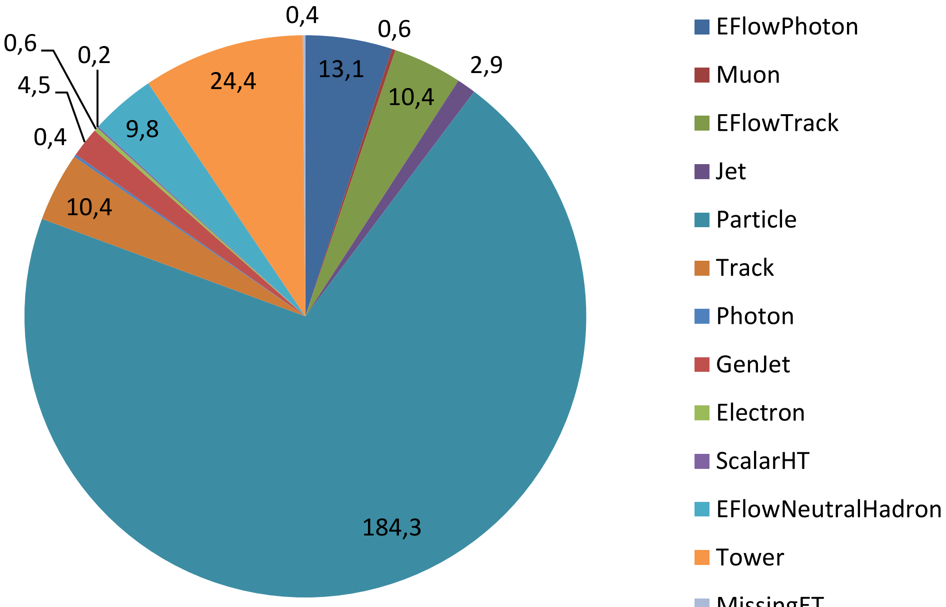
DelphesMA5tune

28.7 MB



Delphes with MA5tune card

261.9 MB



x9

Data format comparison: old DelphesMA5tune vs new Delphes+MA5card

Test sample: drell-yan (+2 jets)

DelphesMA5tune → **Delphes with MA5tune card**

x9

Solutions for MadAnalysis 1.3?

→ Designing with Delphes people 2 modules

- Zero-suppression for towers
- Filtering generated particles with a set of criteria

Isolation for people in a hurry

New isolation functions are contained in the service PHYSICS. 4 algorithms are available.

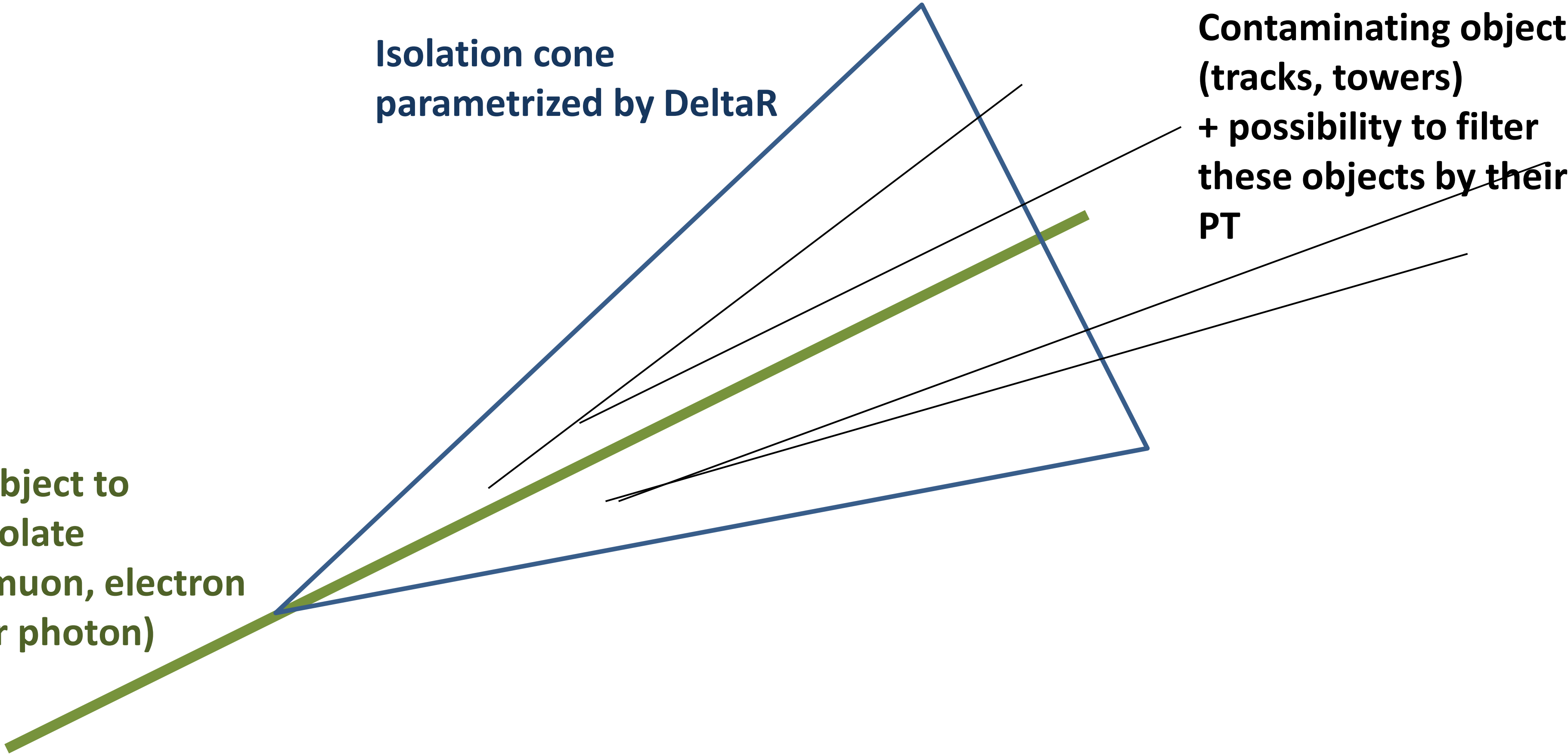
`PHYSICS->Isol->`	*Common functions related to isolation*
`PHYSICS->Isol->tracker->`	*Functions related to isolation based on tracker*
`PHYSICS->Isol->calorimeter->`	*Functions related to isolation based on calorimeter*
`PHYSICS->Isol->combined->`	*Functions related to isolation based on tracker+calorimeter*
`PHYSICS->Isol->eflow->`	*Functions related to isolation based on particle flow algorithm*

Same structure

Extra functions

Not provided by the original Delphes package

Same structure for algorithm = native Delphes isolation functions



Isolation cone parametrized by DeltaR

Contaminating objects (tracks, towers) + possibility to filter these objects by their PT

Object to isolate (muon, electron or photon)

Algorithms could provide 2 variables:

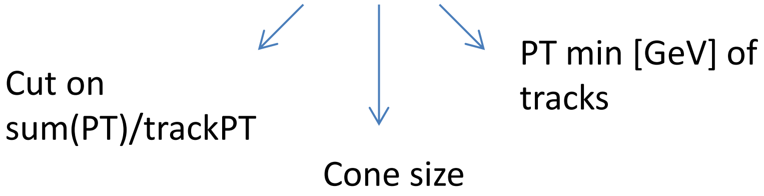
- Absolute variable: Scalar sum of the contaminating object PT
- Relative variable: absolute variable / PT of the object to isolate

all-in-one functions for getting isolated objects

Example of the branch: `PHYSICS->Isol->tracker->`

muons

```
std::vector<const RecLeptonFormat*> isolated =
PHYSICS->Isol->tracker->getRelIsolated (event.rec()->muons(),
                                        event.rec(),
                                        1,0.5,1);
```



Cut on sum(PT)/trackPT

Cone size

PT min [GeV] of tracks

electrons

```
std::vector<const RecLeptonFormat*> isolated =
PHYSICS->Isol->tracker->getRelIsolated (event.rec()->electrons(),
                                        event.rec(),
                                        1,0.5,1);
```

photons

```
std::vector<const RecPhotonFormat*> isolated =
PHYSICS->Isol->tracker->getRelIsolated (event.rec()->photons(),
                                        event.rec(),
                                        1,0.5,1);
```

Cleaning the jet collection

Reminder: there are some overlap between electron/photon collection
and jet collection

```
std::vector<const RecJetFormat*> cleaned_jets =
PHYSICS->Isol->JetCleaning(event.rec()->jets(), isolated_electrons, 0.1, 1);

cleaned_jets =
PHYSICS->Isol->JetCleaning(cleaned_jets, isolated_muons,   0.1, 1);

cleaned_jets =
PHYSICS->Isol->JetCleaning(cleaned_jets, isolated_photons, 0.1, 1);
```

Filter jets with PT min

Matching criterion
(DR max between objects)

More details related to isolation

Accessing absolute isolation value

Example of the branch: `PHYSICS->Isol->tracker->`

```
double value=
PHYSICS->Isol->tracker->sumIsolation(myMuon,event.rec(),0.5,1);
```

PT min [GeV] of tracks

Cone size

```
double value=
PHYSICS->Isol->tracker->sumIsolation(myElectron,event.rec(),0.5,1);
```

```
double value=
PHYSICS->Isol->tracker->sumIsolation(myPhoton,event.rec(),0.5,1);
```

Accessing relative isolation value

Example of the branch: `PHYSICS->Isol->tracker->`

```
double value=
PHYSICS->Isol->tracker->relIsolation(myMuon,event.rec(),0.5,1);
```

PT min [GeV] of tracks

Cone size

```
double value=
PHYSICS->Isol->tracker->relIsolation(myElectron,event.rec(),0.5,1);
```

```
double value=
PHYSICS->Isol->tracker->relIsolation(myPhoton,event.rec(),0.5,1);
```

Additional functions for Eflow algorithms

Example of the branch: `PHYSICS->Isol->eflow->`

```
double value=
PHYSICS->Isol->eflow->sumIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::TRACK_COMPONENT);

double value=
PHYSICS->Isol->eflow->sumIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::NEUTRAL_COMPONENT);

double value=
PHYSICS->Isol->eflow->sumIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::PHOTON_COMPONENT);

double value=
PHYSICS->Isol->eflow->relIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::TRACK_COMPONENT);

double value=
PHYSICS->Isol->eflow->relIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::NEUTRAL_COMPONENT);

double value=
PHYSICS->Isol->eflow->relIsolation(myMuon,event.rec(),0.5,1,
                                  IsolationEFlow::PHOTON_COMPONENT);
```