MadAnalysis 5 for experts

Benjamin Fuks

LPTHE / Sorbonne Université

More information:
Conte & Fuks, arXiv:1808.00480, IJMPA 33 (2018) 1830027

The second MadAnalysis 5 workshop on LHC recasting @ Korea

KIAS (Seoul, Korea) - 14 February 2020

Outline

MADANALYSIS 5 in a nutshell

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

- What is MADANALYSIS 5?
  - A framework for phenomenological analyses
  - Any level of sophistication: partonic, hadronic, detector, reconstructed
  - Several input format: STDHEP, HEPMC, LHE, LHCO, ROOT (from DELPHES)
  - User-friendly, flexible and fast
  - Interfaced to other HEP packages (detector simulation, jet clustering, …)

MADANALYSIS 5 in a nutshell

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

✦ What is MADANALYSIS 5?

- ✤ A framework for phenomenological analyses
- ✤ Any level of sophistication: partonic, hadronic, detector, reconstructed
- ✤ Several input format: STDHEP, HEPMC, LHE, LHCO, ROOT (from DELPHES)
- ✤ User-friendly, flexible and fast
- ✤ Interfaced to other HEP packages (detector simulation, jet clustering, …)

✦ Normal pythonic mode

- ✤ Intuitive commands typed in a PYTHON interface
- ✤ Analysis performed behind the scenes (a C++ black box)
- ✤ Human readable output: HTML and LaTeX

MadAnalysis 5 in a nutshell

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

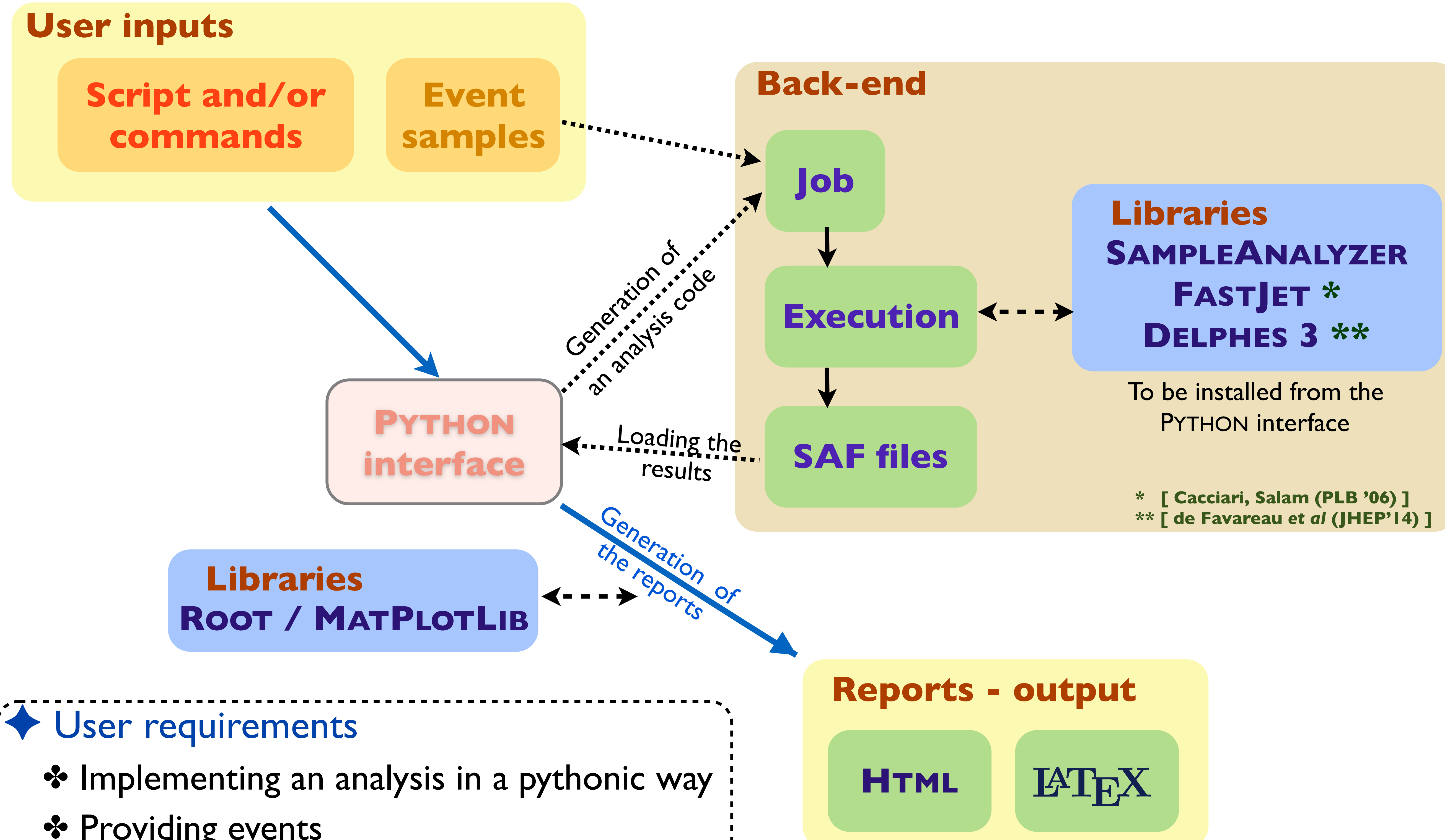
✦ What is MadAnalysis 5?
- ✤ A framework for phenomenological analyses
- ✤ Any level of sophistication: partonic, hadronic, detector, reconstructed
- ✤ Several input format: StdHep, HepMC, Lhe, Lhco, Root (from Delphes)
- ✤ User-friendly, flexible and fast
- ✤ Interfaced to other HEP packages (detector simulation, jet clustering, …)

✦ Normal pythonic mode
- ✤ Intuitive commands typed in a Python interface
- ✤ Analysis performed behind the scenes (a C++ black box)
- ✤ Human readable output: Html and LaTeX

✦ Expert mode
- ✤ C++ programming within the SampleAnalyzer framework (the MA5 core)

The (normal) user-friendly mode

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

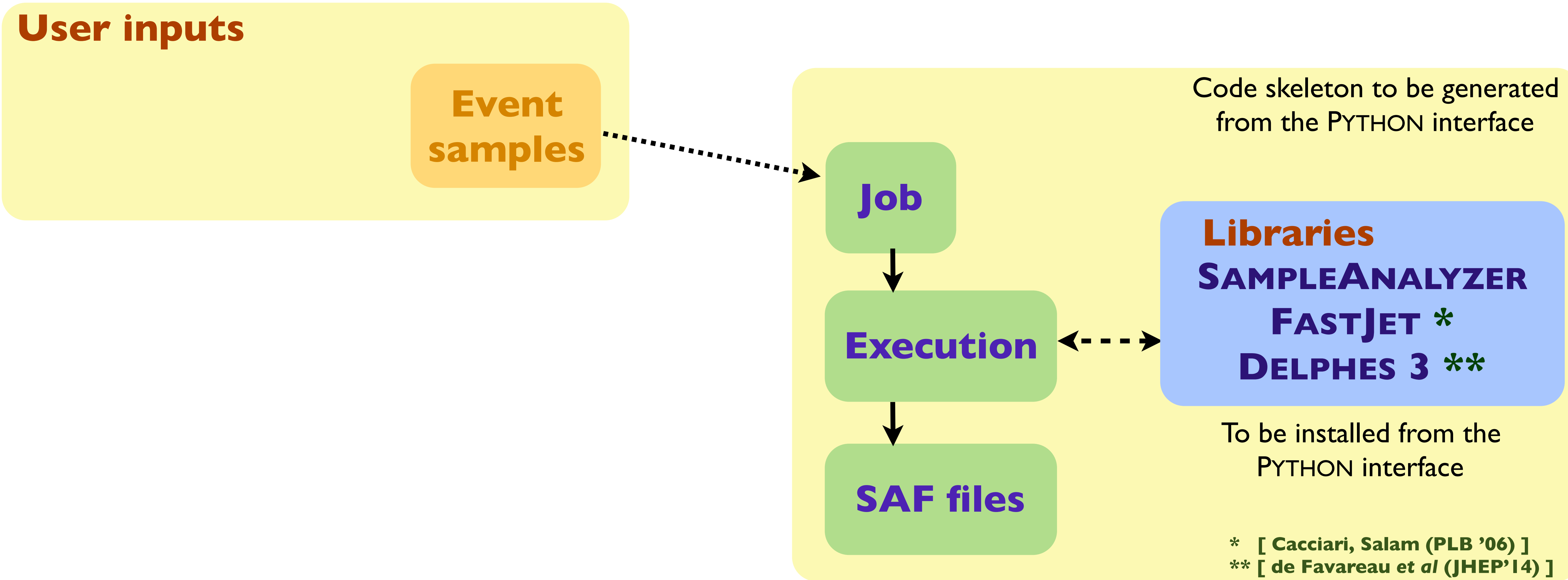


✦ User requirements

- Implementing an analysis in a pythonic way
- Providing events

The expert (developer-friendly) mode

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]



✦ User requirements

- ✤ Implementing a C++ analysis: a skeleton can be automatically generated
- ✤ Providing events
- ✤ Treatment of the output by the user

The expert mode of MadAnalysis 5

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

- MadAnalysis 5 without its Python interface
  - More freedom in the observables (only some can be called from the Python console)
  - Complicated cuts can be implemented
  - More suitable for large numbers of events (using several cores)

The expert mode of MADANALYSIS 5

[Conte, BF, Serret (CPC '13); Conte, Dumont, BF, Wymant (EPJC '14); Conte & BF (IJMPA`18)]

✦ MADANALYSIS 5 without its PYTHON interface

- More freedom in the observables (only some can be called from the PYTHON console)
- Complicated cuts can be implemented
- More suitable for large numbers of events (using several cores)

✦ The expert mode is developer-friendly

The analysis is a C++ class

The SAMPLEANALYZER internal data format

- ★ Readers for LHE, STDHEP, HEPMC, LHCO and DELPHES
- ★ Many classes and methods for particle and object properties
- ★ Specific methods for histograms and cuts

Services

- ★ Physics observables (transverse variables, object identification, isolation)
- ★ Streamers
- ★ Exceptions

Interfaces

- ★ FASTJET
- ★ DELPHES 3
- ★ ROOT

Scripts

- ★ Compilation
- ★ Linking
- ★ Analysis skeleton generator

Main features: signal regions, histograms & cuts

[Conte, Dumont, BF, Wymant (EPJC '14)]

✦ Main features (recast of most cut-based LHC analyses)

- ✤ Support for multiple sub-analyses (signal and control regions)
- ✤ New ready-to-use observables (M_{T2}, M_{T2W}, *etc.*)
- ✤ New optimised handling of cuts and histograms

Main features: signal regions, histograms & cuts

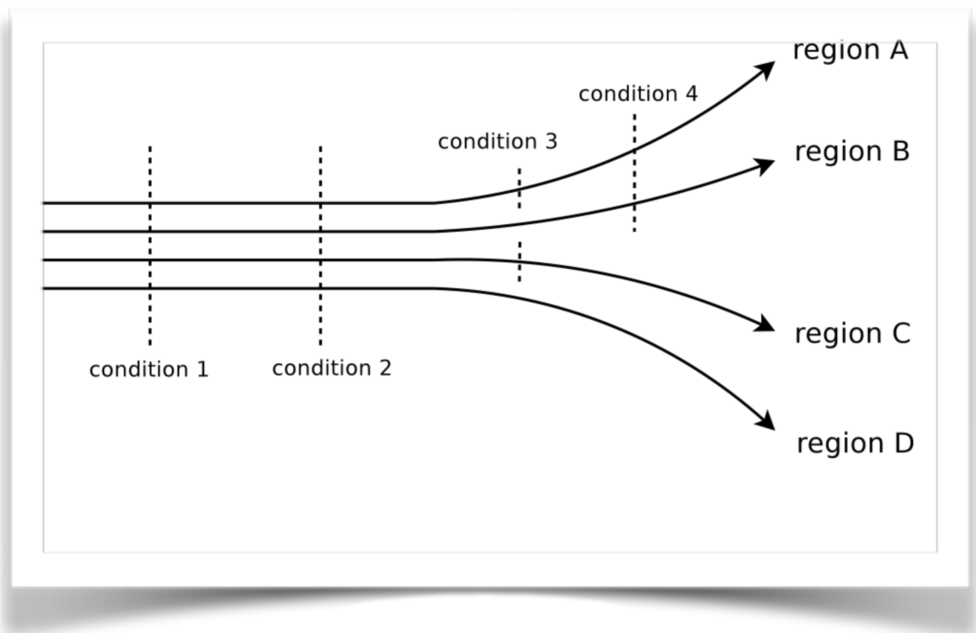
[Conte, Dumont, BF, Wymant (EPJC '14)]

✦ Main features (recast of most cut-based LHC analyses)

- ✤ Support for multiple sub-analyses (signal and control regions)
- ✤ New ready-to-use observables (M_{T2}, M_{T2W}, *etc.*)
- ✤ New optimised handling of cuts and histograms

✦ Handling cuts and histograms

- ✤ Naive approach not efficient (see cut #4 for instance)




```
count the event in region D
if (condition 3)
{
  count the event in region C
  if (condition 4)
  {
    count the event in region A
  }
}
if (condition 4)
{
  count the event in region B
}
```

- ✤ A more efficient algorithm has been implemented
  - ★ Each cut condition is only evaluated once
  - ★ It is applied to all <u>surviving</u> regions simultaneously
- ✤ Similar treatment for histograms

Getting started: understanding a paper

Reimplementing ATLAS SUSY-2018-32

✦ Goal of the lecture

- ✤ Detailing step-by-step the implementation of a search in MADANALYSIS 5
- ✤ From coding to validation

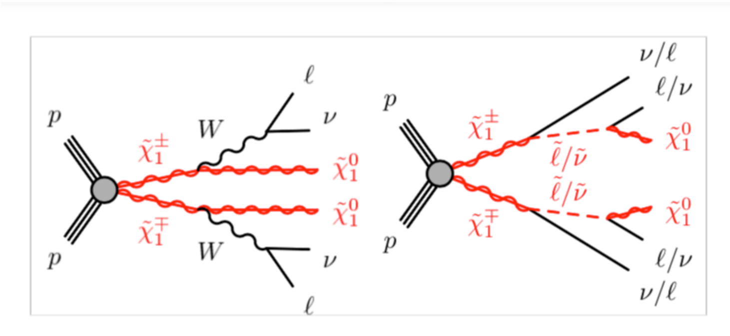
Reimplementing ATLAS SUSY-2018-32

✦ Goal of the lecture

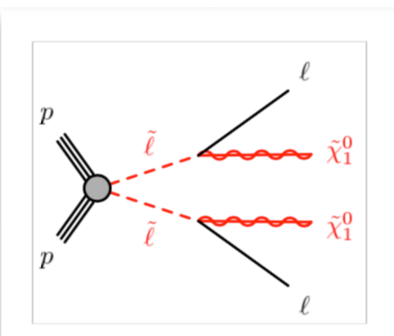
- ✤ Detailing step-by-step the implementation of a search in MadAnalysis 5
- ✤ From coding to validation

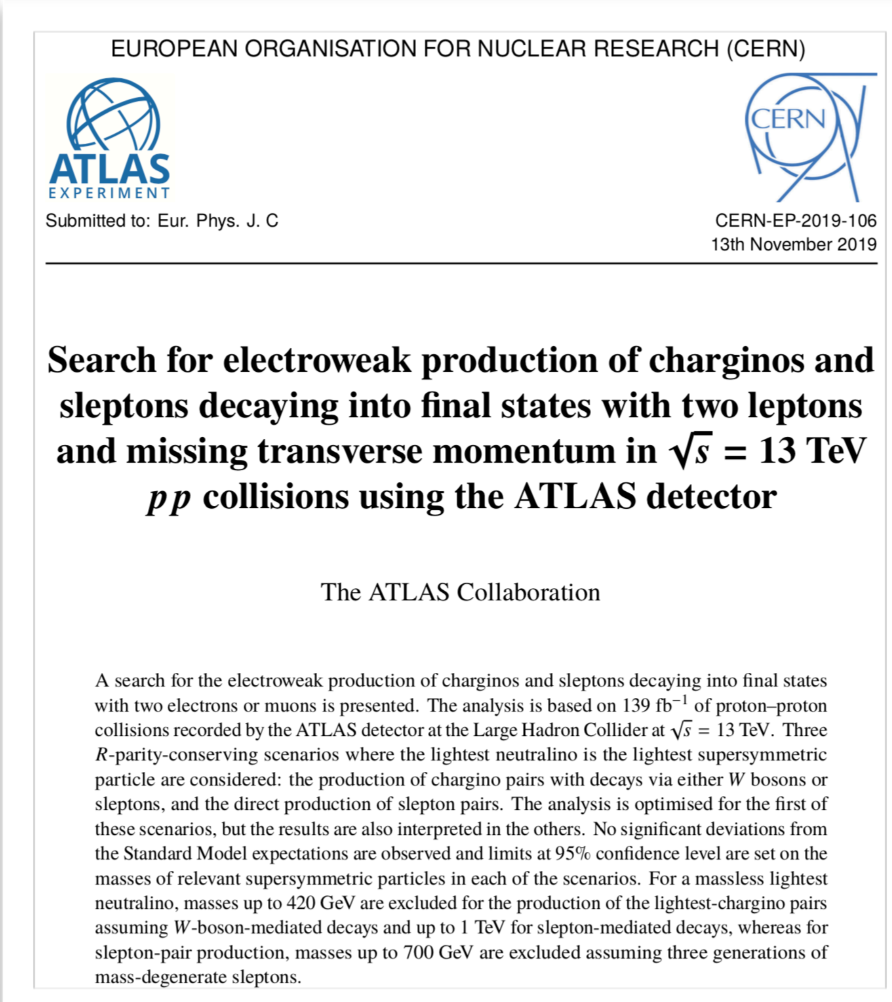
✦ Example: ATLAS SUSY-2018-32

- ✤ Searching for sleptons and electroweakinos
- ✤ Luminosity: 139 fb^{-1} of 13 TeV LHC data
- ✤ Signature: dilepton plus missing energy
  - ★ Chargino pair-production and decay



  - ★ Slepton pair-production and decay





EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

ATLAS EXPERIMENT

CERN

Submitted to: Eur. Phys. J. C

CERN-EP-2019-106
13th November 2019

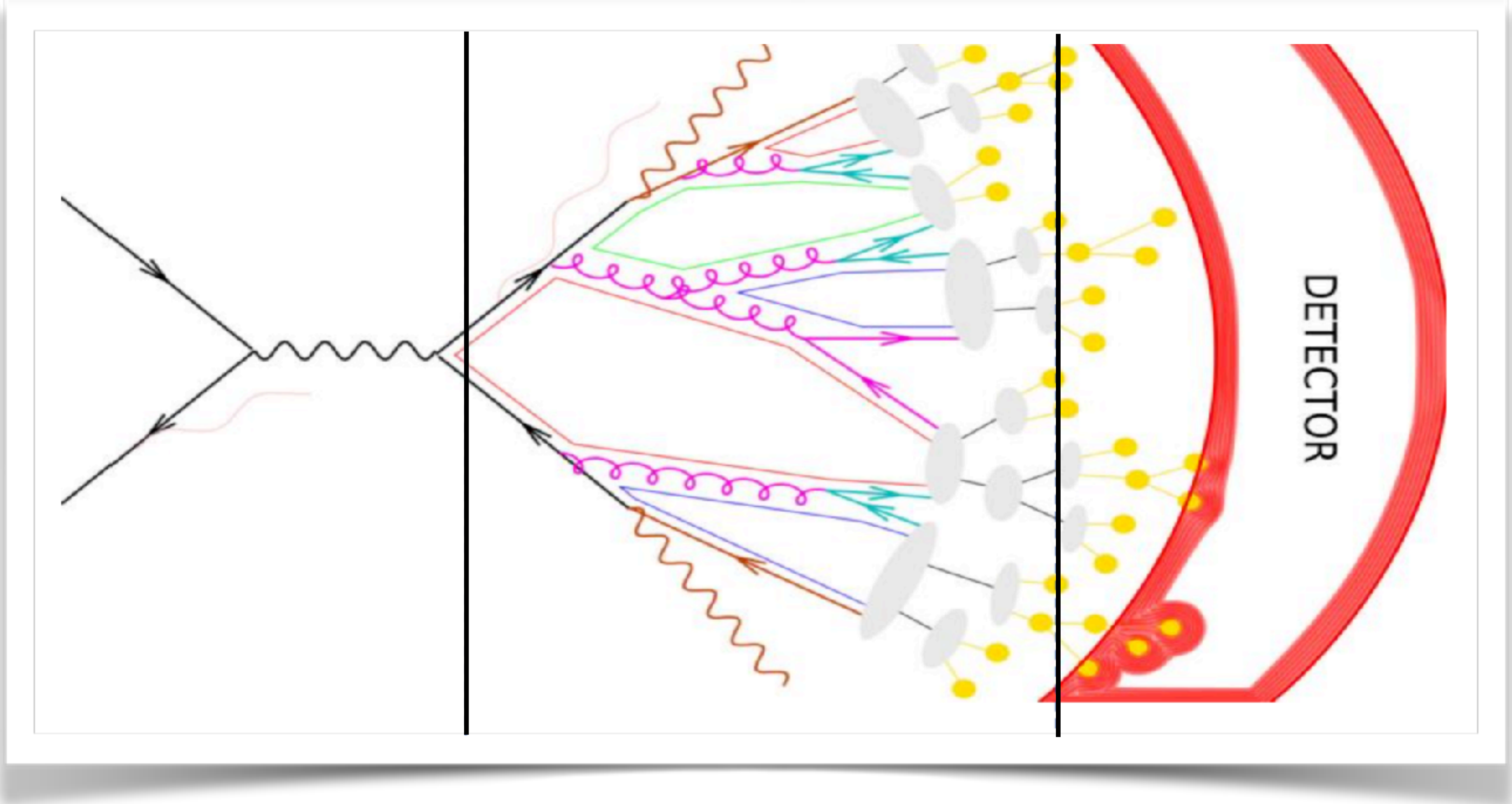
Search for electroweak production of charginos and sleptons decaying into final states with two leptons and missing transverse momentum in $\sqrt{s}$ = 13 TeV pp collisions using the ATLAS detector

The ATLAS Collaboration

A search for the electroweak production of charginos and sleptons decaying into final states with two electrons or muons is presented. The analysis is based on 139 fb^{-1} of proton–proton collisions recorded by the ATLAS detector at the Large Hadron Collider at $\sqrt{s}$ = 13 TeV. Three R-parity-conserving scenarios where the lightest neutralino is the lightest supersymmetric particle are considered: the production of chargino pairs with decays via either W bosons or sleptons, and the direct production of slepton pairs. The analysis is optimised for the first of these scenarios, but the results are also interpreted in the others. No significant deviations from the Standard Model expectations are observed and limits at 95% confidence level are set on the masses of relevant supersymmetric particles in each of the scenarios. For a massless lightest neutralino, masses up to 420 GeV are excluded for the production of the lightest-chargino pairs assuming W-boson-mediated decays and up to 1 TeV for slepton-mediated decays, whereas for slepton-pair production, masses up to 700 GeV are excluded assuming three generations of mass-degenerate sleptons.

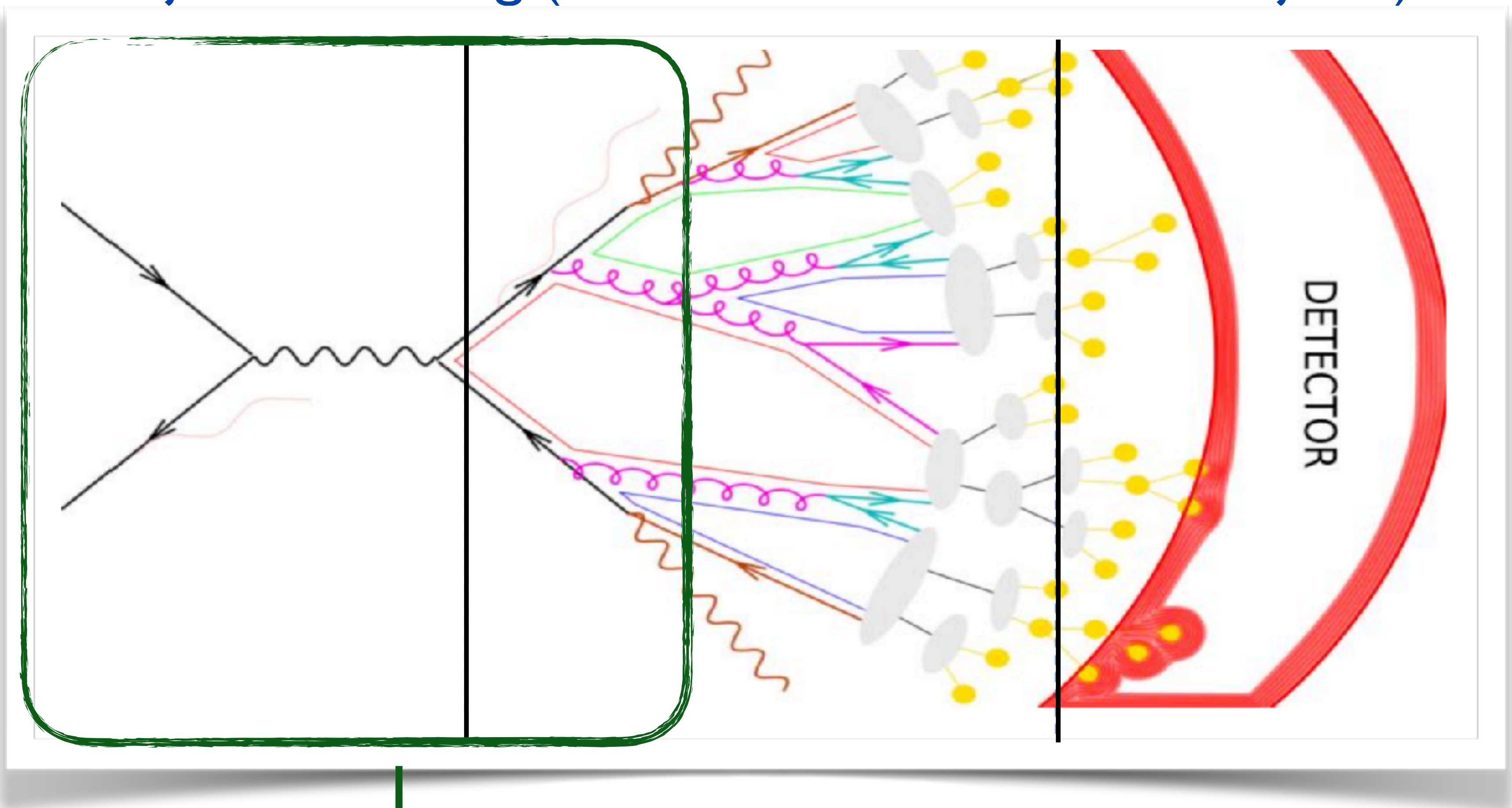
Event reconstruction

✦ Object clustering (into a smaller number of objects)



Event reconstruction

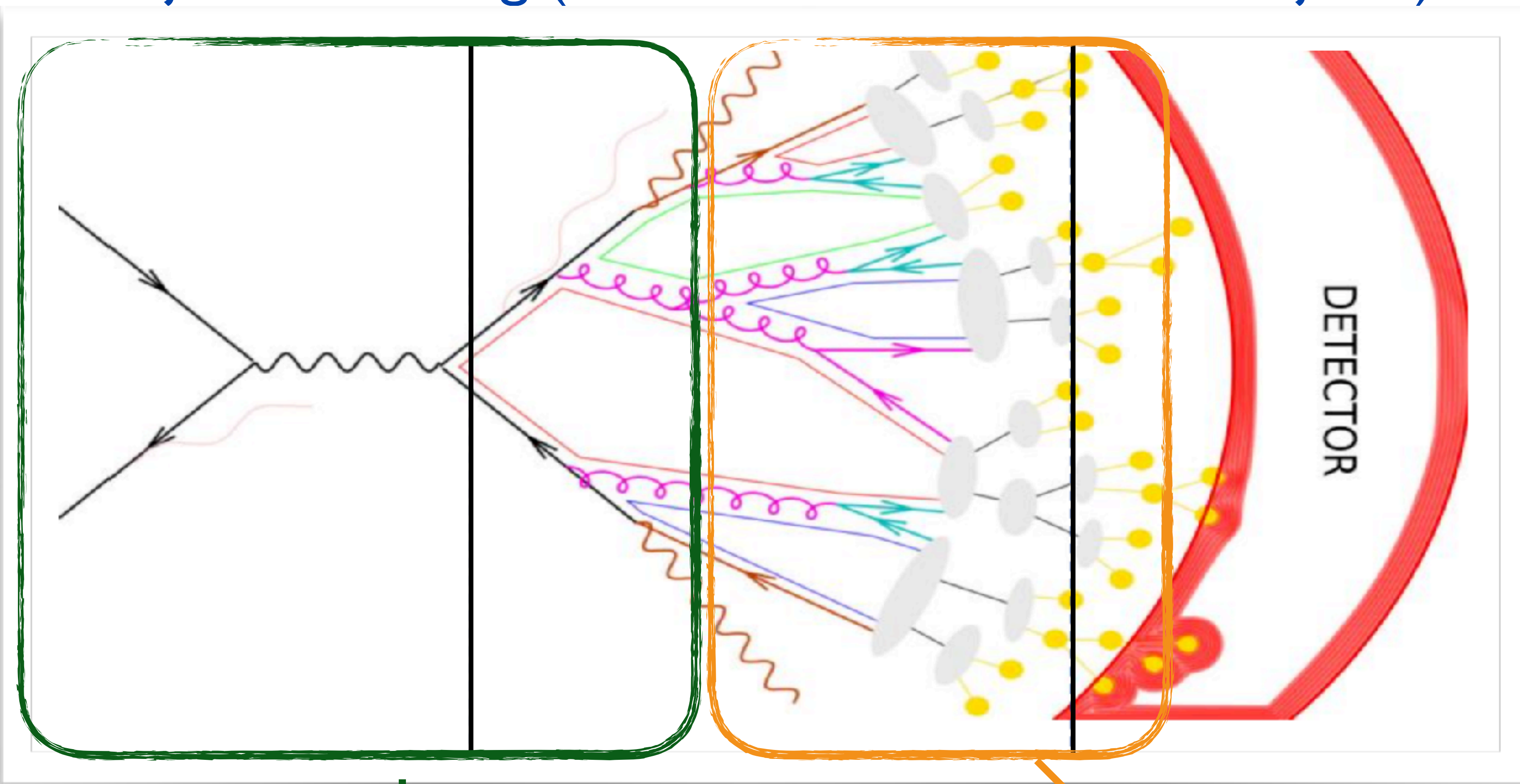
✦ Object clustering (into a smaller number of objects)



✤ Hard scattering / parton showers
- ★ No real need for clustering
  - → However possible
- ★ "Stable" tops, gluinos, *etc.*

Event reconstruction

✦ Object clustering (into a smaller number of objects)



✤ Hard scattering / parton showers

★ No real need for clustering

→ However possible

★ "Stable" tops, gluinos, *etc.*

✤ Hadron level

★ Hundreds hadrons → small number of jets

★ Reconstruction yields few objects:

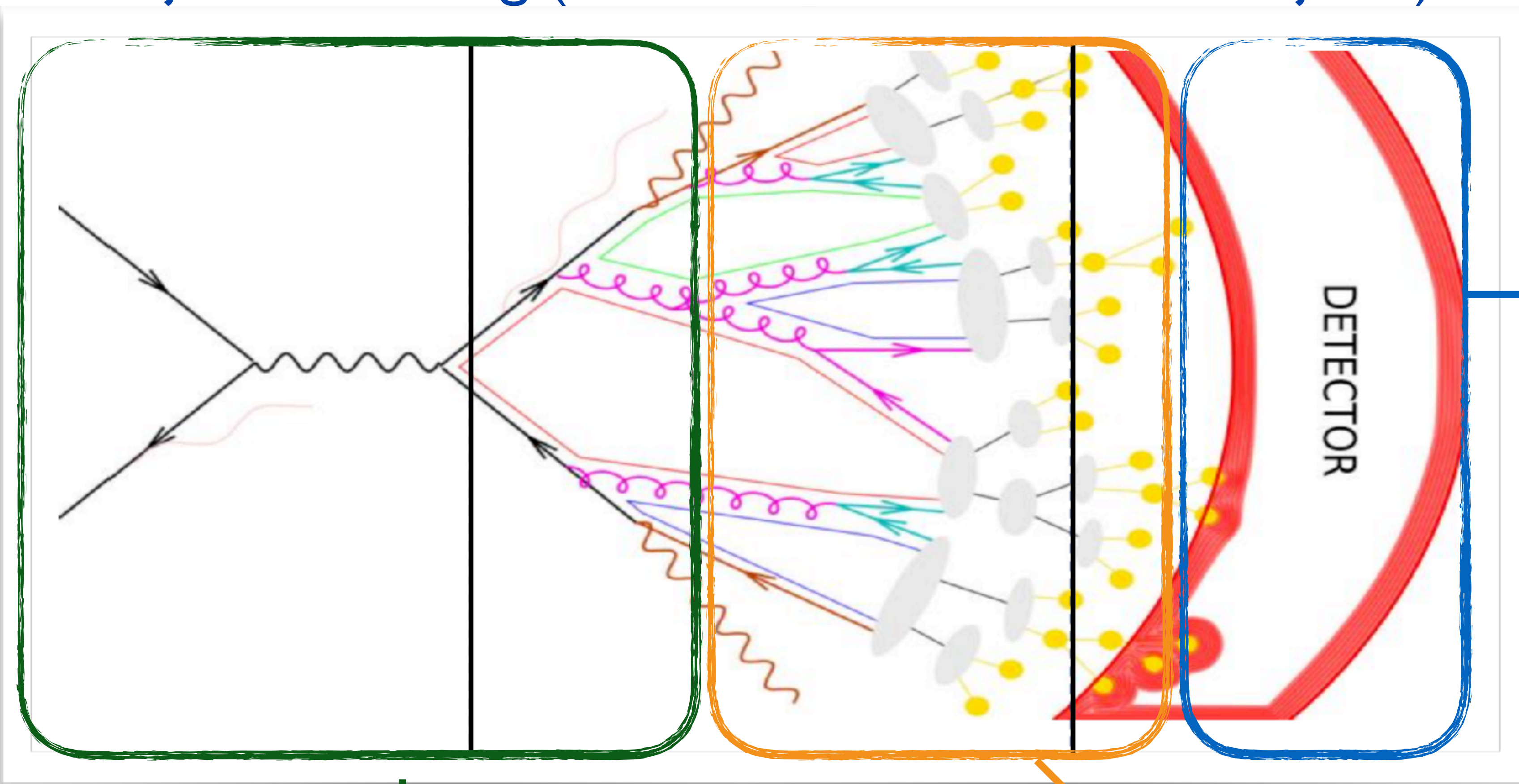
❖ Electrons, muons, photons

❖ Hadronic taus, jets

❖ Missing transverse energy

Event reconstruction

✦ Object clustering (into a smaller number of objects)



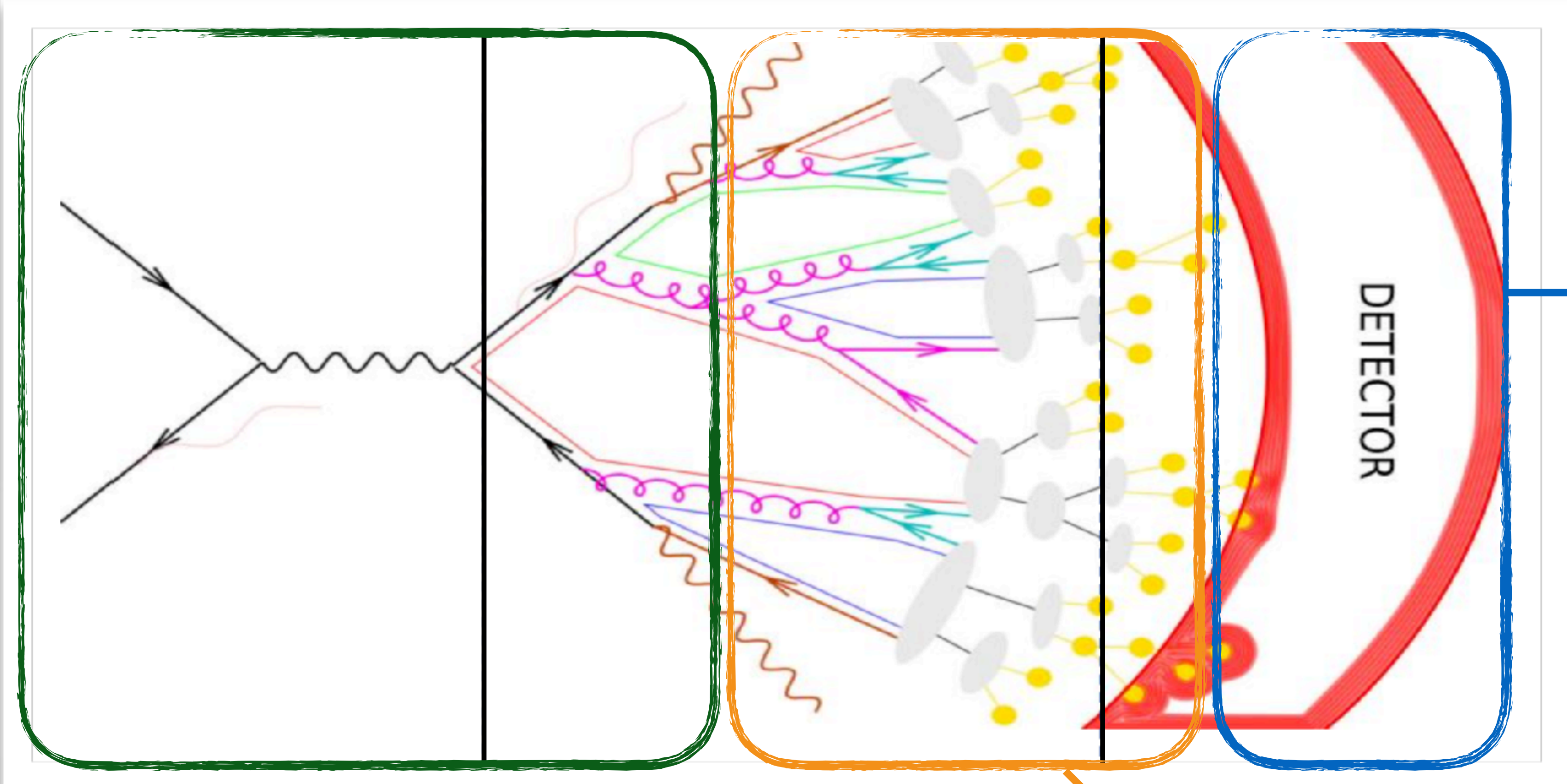
✤ Detector level
- ★ 100s of hits/tracks
- ★ Reconstruction yields:
  - ❖ Electrons, muons, γ's
  - ❖ Hadronic taus, jets
  - ❖ Missing energy

✤ Hard scattering / parton showers
- ★ No real need for clustering
  - → However possible
- ★ "Stable" tops, gluinos, *etc.*

✤ Hadron level
- ★ Hundreds hadrons → small number of jets
- ★ Reconstruction yields few objects:
  - ❖ Electrons, muons, photons
  - ❖ Hadronic taus, jets
  - ❖ Missing transverse energy

Event reconstruction

✦ Object clustering (into a smaller number of objects)



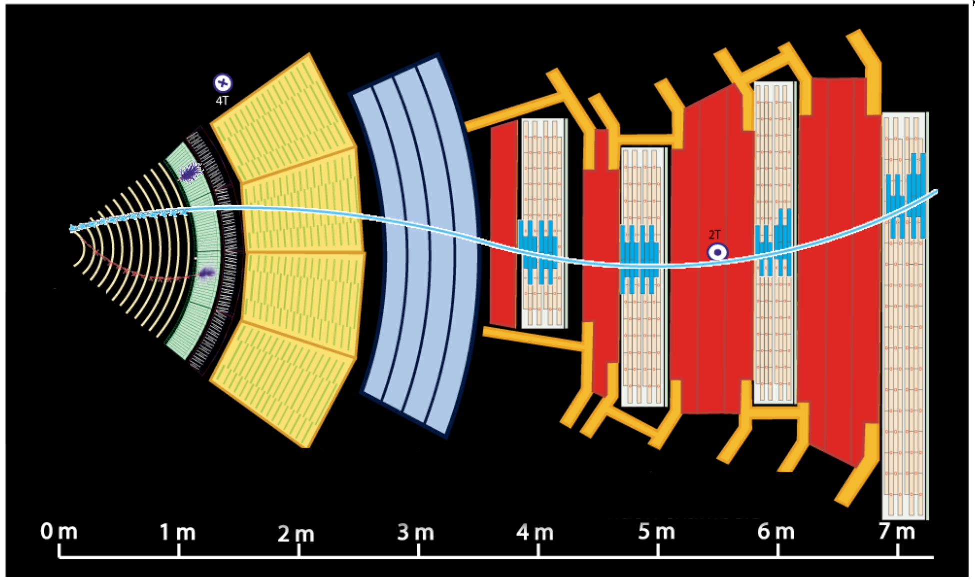
✤ Detector level
★ 100s of hits/tracks
★ Reconstruction yields:
❖ Electrons, muons, γ's
❖ Hadronic taus, jets
❖ Missing energy

✤ Hard scattering / parton showers
★ No real need for clustering
→ However possible
★ "Stable" tops, gluinos, *etc.*

✤ Hadron level
★ Hundreds hadrons → small number of jets
★ Reconstruction yields few objects:
❖ Electrons, muons, photons
❖ Hadronic taus, jets
❖ Missing transverse energy

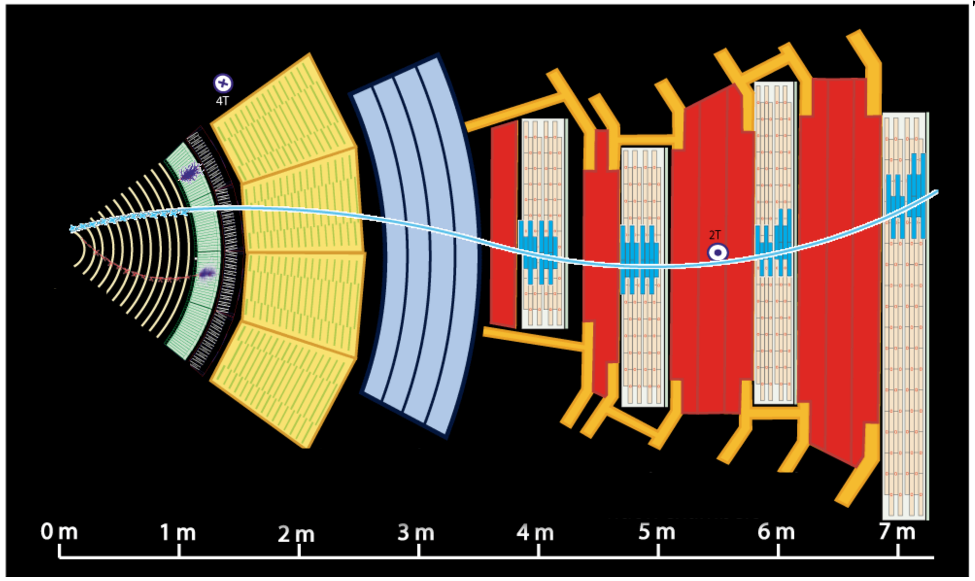
Reconstructed objects in a nutshell

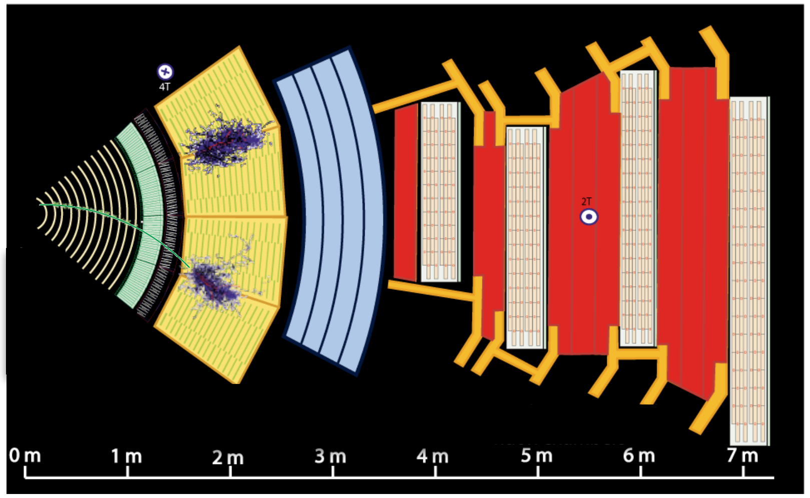
- ✦ Directly observable objects
  - ✤ Truely stable: electrons, γ's
  - ✤ Stable on detector scales: muons
- ✦ Other unstable particles
  - ✤ Observable through decays
  - ✤ SM and BSM



Reconstructed objects in a nutshell

- ✦ Directly observable objects
  - ✤ Truely stable: electrons, γ's
  - ✤ Stable on detector scales: muons
- ✦ Other unstable particles
  - ✤ Observable through decays
  - ✤ SM and BSM





- ✦ Jets
  - ✤ Charged and neutral hadrons
  - ✤ Matched with an initial parton

Reconstructed objects in a nutshell

✦ Directly observable objects

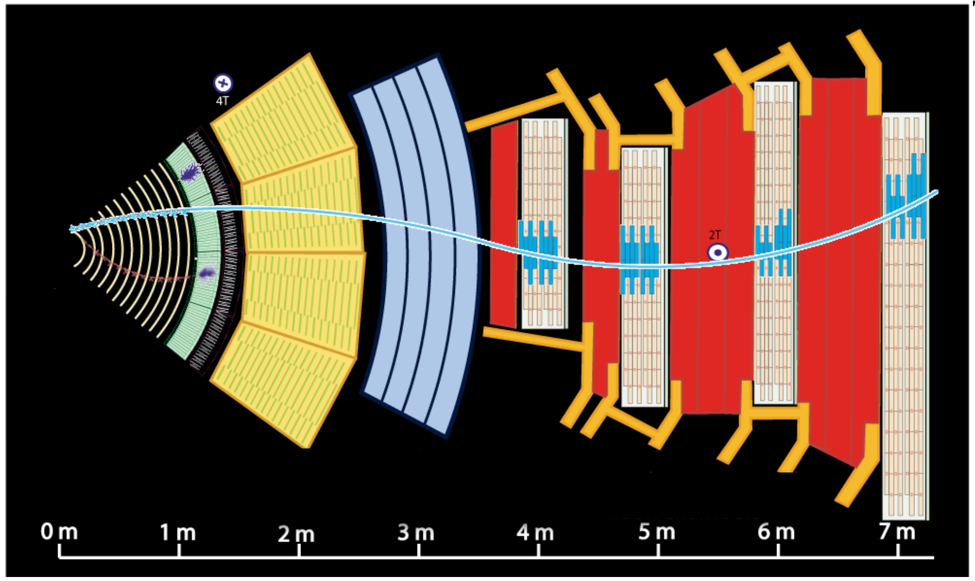
✤ Truely stable: electrons, γ's

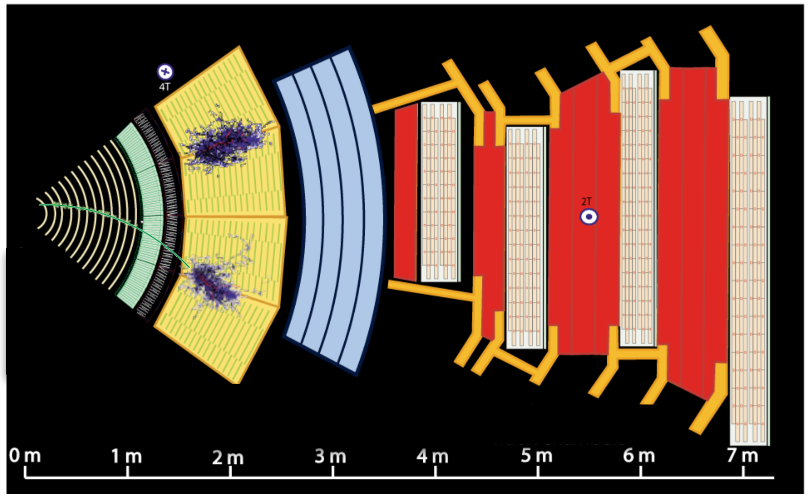
✤ Stable on detector scales: muons

✦ Other unstable particles

✤ Observable through decays

✤ SM and BSM





✦ Jets

✤ Charged and neutral hadrons

✤ Matched with an initial parton

✦ The invisible

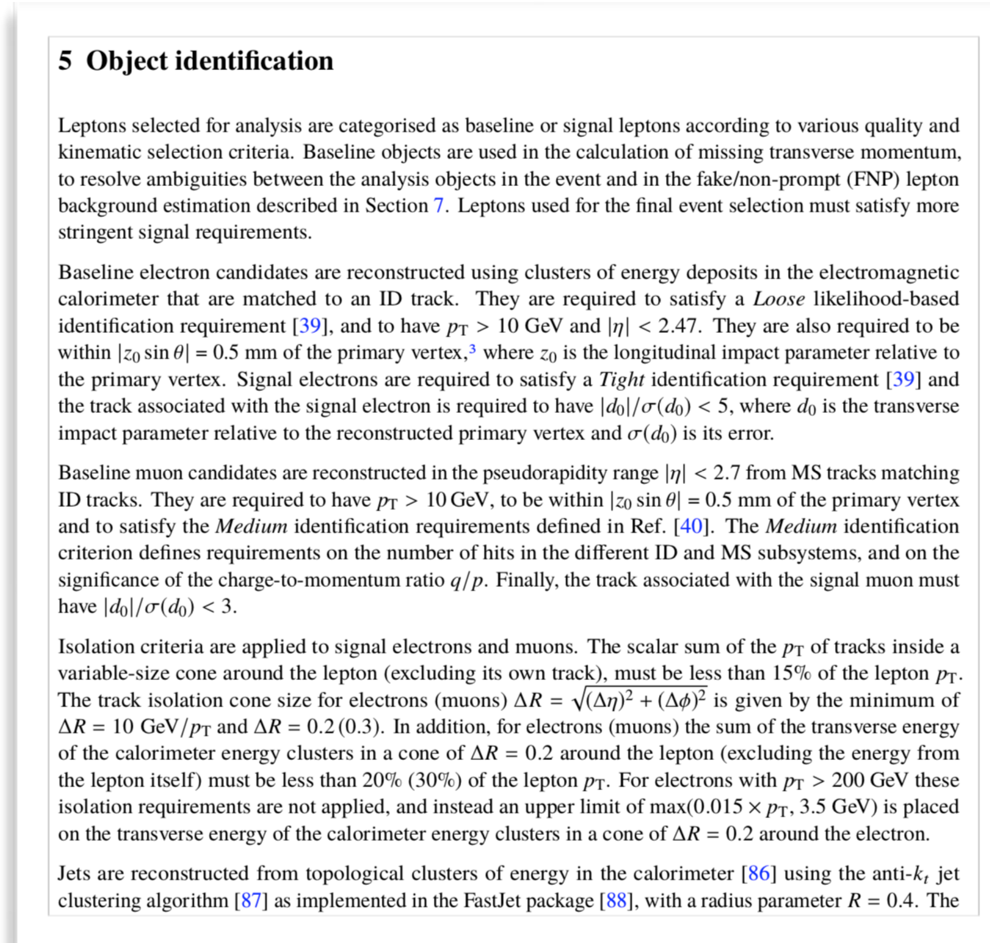
✤ Energy-momentum conservation

→ reconstructing the invisible (neutrinos & more)

Object identification → collections

✦ Dedicated section in the experimental paper

- ✤ Quality of the objects
- ✤ Properties of the objects
- ✤ References



5 Object identification

Leptons selected for analysis are categorised as baseline or signal leptons according to various quality and kinematic selection criteria. Baseline objects are used in the calculation of missing transverse momentum, to resolve ambiguities between the analysis objects in the event and in the fake/non-prompt (FNP) lepton background estimation described in Section 7. Leptons used for the final event selection must satisfy more stringent signal requirements.

Baseline electron candidates are reconstructed using clusters of energy deposits in the electromagnetic calorimeter that are matched to an ID track. They are required to satisfy a *Loose* likelihood-based identification requirement [39], and to have $p_T > 10$ GeV and $|\eta| < 2.47$. They are also required to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex,[3] where z_0 is the longitudinal impact parameter relative to the primary vertex. Signal electrons are required to satisfy a *Tight* identification requirement [39] and the track associated with the signal electron is required to have $|d_0|/\sigma(d_0) < 5$, where d_0 is the transverse impact parameter relative to the reconstructed primary vertex and $\sigma(d_0)$ is its error.

Baseline muon candidates are reconstructed in the pseudorapidity range $|\eta| < 2.7$ from MS tracks matching ID tracks. They are required to have $p_T > 10$ GeV, to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex and to satisfy the *Medium* identification requirements defined in Ref. [40]. The *Medium* identification criterion defines requirements on the number of hits in the different ID and MS subsystems, and on the significance of the charge-to-momentum ratio q/p. Finally, the track associated with the signal muon must have $|d_0|/\sigma(d_0) < 3$.

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10$ GeV$/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\text{ GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The

Object identification → collections

- ✦ Dedicated section in the experimental paper
  - ✤ Quality of the objects
  - ✤ Properties of the objects
  - ✤ References
- ✦ Two classes of objects
  - ✤ Leptons: electrons, muons
    - ★ Two categories:
      - → baseline (loose) leptons
      - → signal (tighter) leptons
    - ★ Isolation requirements
  - ✤ Jets
    - ★ anti-k_T, R=0.4
    - ★ Quality: centrality, hardness
    - ★ b-tagging (MV2C10)

5 Object identification

Leptons selected for analysis are categorised as baseline or signal leptons according to various quality and kinematic selection criteria. Baseline objects are used in the calculation of missing transverse momentum, to resolve ambiguities between the analysis objects in the event and in the fake/non-prompt (FNP) lepton background estimation described in Section 7. Leptons used for the final event selection must satisfy more stringent signal requirements.

Baseline electron candidates are reconstructed using clusters of energy deposits in the electromagnetic calorimeter that are matched to an ID track. They are required to satisfy a *Loose* likelihood-based identification requirement [39], and to have $p_T > 10$ GeV and $|\eta| < 2.47$. They are also required to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex,[3] where z_0 is the longitudinal impact parameter relative to the primary vertex. Signal electrons are required to satisfy a *Tight* identification requirement [39] and the track associated with the signal electron is required to have $|d_0|/\sigma(d_0) < 5$, where d_0 is the transverse impact parameter relative to the reconstructed primary vertex and $\sigma(d_0)$ is its error.

Baseline muon candidates are reconstructed in the pseudorapidity range $|\eta| < 2.7$ from MS tracks matching ID tracks. They are required to have $p_T > 10$ GeV, to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex and to satisfy the *Medium* identification requirements defined in Ref. [40]. The *Medium* identification criterion defines requirements on the number of hits in the different ID and MS subsystems, and on the significance of the charge-to-momentum ratio q/p. Finally, the track associated with the signal muon must have $|d_0|/\sigma(d_0) < 3$.

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The

Object identification → collections

- Dedicated section in the experimental paper
  - Quality of the objects
  - Properties of the objects
  - References
- Two classes of objects
  - Leptons: electrons, muons
    - Two categories:
      - → baseline (loose) leptons
      - → signal (tighter) leptons
    - Isolation requirements
  - Jets
    - anti-k_T, R=0.4
    - Quality: centrality, hardness
    - *b*-tagging (MV2C10)

5 Object identification

Leptons selected for analysis are categorised as baseline or signal leptons according to various quality and kinematic selection criteria. Baseline objects are used in the calculation of missing transverse momentum, to resolve ambiguities between the analysis objects in the event and in the fake/non-prompt (FNP) lepton background estimation described in Section 7. Leptons used for the final event selection must satisfy more stringent signal requirements.

Baseline electron candidates are reconstructed using clusters of energy deposits in the electromagnetic calorimeter that are matched to an ID track. They are required to satisfy a *Loose* likelihood-based identification requirement [39], and to have $p_{\mathrm{T}} > 10$ GeV and $|\eta| < 2.47$. They are also required to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex,[3] where z_0 is the longitudinal impact parameter relative to the primary vertex. Signal electrons are required to satisfy a *Tight* identification requirement [39] and the track associated with the signal electron is required to have $|d_0|/\sigma(d_0) < 5$, where d_0 is the transverse impact parameter relative to the reconstructed primary vertex and $\sigma(d_0)$ is its error.

Baseline muon candidates are reconstructed in the pseudorapidity range $|\eta| < 2.7$ from MS tracks matching ID tracks. They are required to have $p_{\mathrm{T}} > 10$ GeV, to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex and to satisfy the *Medium* identification requirements defined in Ref. [40]. The *Medium* identification criterion defines requirements on the number of hits in the different ID and MS subsystems, and on the significance of the charge-to-momentum ratio q/p. Finally, the track associated with the signal muon must have $|d_0|/\sigma(d_0) < 3$.

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_{T} of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_{T}. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10~\mathrm{GeV}/p_{\mathrm{T}}$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_{T}. For electrons with $p_{\mathrm{T}} > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_{\mathrm{T}}, 3.5~\mathrm{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The

- Extra features
  - Overlap removal
    - jets-electrons
    - jets-muons
    - electrons-muons
  - Missing energy → inclusively reconstructed from the visible sector

Signal region definitions

- Dedicated section in the experimental paper
- Preselection
  - Two opposite-charge leptons
    - Hard leptons ($p_T > 25$ GeV)
    - Heavy system ($m_{ll} > 100$ GeV)
  - *b*-veto
  - Missing energy requirements
    - *MET* > 100 GeV
    - *MET* significance > 10
      → approximated by $MET/\sqrt{H_T}$

6 Search strategy

Events are required to have exactly two oppositely charged signal leptons ℓ_1 and ℓ_2, both with $p_{\mathrm{T}} > 25$ GeV. To remove contributions from low-mass resonances and to ensure good modelling of the SM background in all relevant regions, the invariant mass of the two leptons must be $m_{\ell_1\ell_2} > 100$ GeV. Events are further required to have no reconstructed b-jets, to suppress contributions from processes with top quarks. Selected events must also satisfy $E_{\mathrm{T}}^{\mathrm{miss}} > 110$ GeV and $E_{\mathrm{T}}^{\mathrm{miss}}$ significance > 10.

The stransverse mass m_{T2} [96, 97] is a kinematic variable used to bound the masses of a pair of particles that are assumed to have each decayed into one visible and one invisible particle. It is defined as

$$m_{\mathrm{T2}}(\mathbf{p}_{\mathrm{T},1}, \mathbf{p}_{\mathrm{T},2}, \mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}) = \min_{\mathbf{q}_{\mathrm{T},1}+\mathbf{q}_{\mathrm{T},2}=\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}} \left\{\max[\, m_{\mathrm{T}}(\mathbf{p}_{\mathrm{T},1}, \mathbf{q}_{\mathrm{T},1}), m_{\mathrm{T}}(\mathbf{p}_{\mathrm{T},2}, \mathbf{q}_{\mathrm{T},2})\,]\right\},$$

where m_{T} indicates the transverse mass,[5] $\mathbf{p}_{\mathrm{T},1}$ and $\mathbf{p}_{\mathrm{T},2}$ are the transverse-momentum vectors of the two leptons, and $\mathbf{q}_{\mathrm{T},1}$ and $\mathbf{q}_{\mathrm{T},2}$ are vectors with $\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}} = \mathbf{q}_{\mathrm{T},1} + \mathbf{q}_{\mathrm{T},2}$. The minimisation is performed over all the possible decompositions of $\mathbf{p}_{\mathrm{T}}^{\mathrm{miss}}$. For $t\bar{t}$ or WW decays, assuming an ideal detector with perfect momentum resolution, $m_{\mathrm{T2}}(\mathbf{p}_{\mathrm{T},\ell_1}, \mathbf{p}_{\mathrm{T},\ell_2}, \mathbf{p}_{\mathrm{T}}^{\mathrm{miss}})$ has a kinematic endpoint at the mass of the W boson [97]. Signal models with significant mass splittings between the $\tilde{\chi}_1^{\pm}$ and the $\tilde{\chi}_1^0$ feature m_{T2} distributions that extend beyond the kinematic endpoint expected from the dominant SM backgrounds. Therefore, events are required to have

Signal region definitions

- ✦ Dedicated section in the experimental paper
- ✦ Preselection
  - ✤ Two opposite-charge leptons
    - ★ Hard leptons ($p_T > 25$ GeV)
    - ★ Heavy system ($m_{ll} > 100$ GeV)
  - ✤ *b*-veto
  - ✤ Missing energy requirements
    - ★ *MET* > 100 GeV
    - ★ *MET* significance > 10
      - → approximated by $MET/\sqrt{H_T}$

6 Search strategy

Events are required to have exactly two oppositely charged signal leptons ℓ_1 and ℓ_2, both with $p_T > 25$ GeV. To remove contributions from low-mass resonances and to ensure good modelling of the SM background in all relevant regions, the invariant mass of the two leptons must be $m_{\ell_1\ell_2} > 100$ GeV. Events are further required to have no reconstructed *b*-jets, to suppress contributions from processes with top quarks. Selected events must also satisfy $E_T^{miss} > 110$ GeV and E_T^{miss} significance > 10.

The stransverse mass m_{T2} [96, 97] is a kinematic variable used to bound the masses of a pair of particles that are assumed to have each decayed into one visible and one invisible particle. It is defined as

$$m_{T2}(\mathbf{p}_{T,1}, \mathbf{p}_{T,2}, \mathbf{p}_T^{miss}) = \min_{\mathbf{q}_{T,1}+\mathbf{q}_{T,2}=\mathbf{p}_T^{miss}} \left\{ \max[\, m_T(\mathbf{p}_{T,1}, \mathbf{q}_{T,1}), m_T(\mathbf{p}_{T,2}, \mathbf{q}_{T,2}) \,] \right\},$$

where m_T indicates the transverse mass,[5] $\mathbf{p}_{T,1}$ and $\mathbf{p}_{T,2}$ are the transverse-momentum vectors of the two leptons, and $\mathbf{q}_{T,1}$ and $\mathbf{q}_{T,2}$ are vectors with $\mathbf{p}_T^{miss} = \mathbf{q}_{T,1} + \mathbf{q}_{T,2}$. The minimisation is performed over all the possible decompositions of $\mathbf{p}_T^{miss}$. For $t\bar{t}$ or WW decays, assuming an ideal detector with perfect momentum resolution, $m_{T2}(\mathbf{p}_{T,\ell_1}, \mathbf{p}_{T,\ell_2}, \mathbf{p}_T^{miss})$ has a kinematic endpoint at the mass of the W boson [97]. Signal models with significant mass splittings between the $\tilde{\chi}_1^\pm$ and the $\tilde{\chi}_1^0$ feature m_{T2} distributions that extend beyond the kinematic endpoint expected from the dominant SM backgrounds. Therefore, events are required to have

- ✦ Signal regions
  - ✤ Two classes of regions
    - ★ DF: different lepton flavours ($e\mu$)
    - ★ SF: same lepton flavour (ee or $\mu\mu$)
      - → off-Z requirements (reducing SM backgrounds)
  - ✤ 0 or 1 light jet
  - ✤ Binning in M_{T2}
    - ★ 9 exclusive bins (tracking of specific endpoints)
    - ★ 2 inclusive bins
    - ★ 2 regions tracking lower M_{T2} endpoints

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	= 0	= 1	= 0	= 1
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	= 0			
Binned SRs				
m_{T2} [GeV]	∈[100,105) ∈[105,110) ∈[110,120) ∈[120,140) ∈[140,160) ∈[160,180) ∈[180,220) ∈[220,260) ∈[260,∞)			
Inclusive SRs				
m_{T2} [GeV]	∈[100,∞) ∈[160,∞) ∈[100,120) ∈[120,160)			

Creating a blank analysis

Adding a new analysis to the local PAD

- ✦ Assumption 1: MadAnalysis 5 v1.8 installed
  - ✤ See https://launchpad.net/madanalysis5

Adding a new analysis to the local PAD

- ✦ Assumption 1: MadAnalysis 5 v1.8 installed
  - ✤ See https://launchpad.net/madanalysis5
- ✦ Assumption 2: recasting tools installed
  - ✤ Install Root externally
  - ✤ Delphes 3 and the PAD:

```
ma5>install PAD
```

```
ma5>install delphes
```

```
MA5: Checking mandatory packages:
MA5:      - Python                        [OK]
MA5:      - GNU GCC g++                   [OK]
MA5:      - GNU Make                      [OK]
MA5: Checking optional packages devoted to data processing:
MA5:      - Zlib                          [OK]
MA5:      - FastJet                       [DISABLED]
MA5:      - Root                          [OK]
MA5:      - Delphes                       [OK]
MA5:      - Delphes-MA5tune               [DISABLED]
MA5: Checking the MadAnalysis 5 core library:
MA5:   => MadAnalysis libraries found.
MA5:   => MadAnalysis test program works.
MA5: Reading user settings ...
MA5: Checking optional packages devoted to reinterpretation:
MA5:      - SciPy                         [OK]
MA5:      - PAD                           [OK]
MA5:      - PADForMA5tune                 [DISABLED]
MA5:      - PADForSFS                     [DISABLED]
MA5:      - pyhf                          [DISABLED]
```

Adding a new analysis to the local PAD

- ✦ Assumption 1: MadAnalysis 5 v1.8 installed
  - ✤ See https://launchpad.net/madanalysis5
- ✦ Assumption 2: recasting tools installed
  - ✤ Install Root externally
  - ✤ Delphes 3 and the PAD:

```
ma5>install PAD
```

```
ma5>install delphes
```

```
MA5: Checking mandatory packages:
MA5:       - Python                     [OK]
MA5:       - GNU GCC g++                [OK]
MA5:       - GNU Make                   [OK]
MA5: Checking optional packages devoted to data processing:
MA5:       - Zlib                       [OK]
MA5:       - FastJet                    [DISABLED]
MA5:       - Root                       [OK]
MA5:       - Delphes                    [OK]
MA5:       - Delphes-MA5tune            [DISABLED]
MA5: Checking the MadAnalysis 5 core library:
MA5:    => MadAnalysis libraries found.
MA5:    => MadAnalysis test program works.
MA5: Reading user settings ...
MA5: Checking optional packages devoted to reinterpretation:
MA5:       - SciPy                      [OK]
MA5:       - PAD                        [OK]
MA5:       - PADForMA5tune              [DISABLED]
MA5:       - PADForSFS                  [DISABLED]
MA5:       - pyhf                       [DISABLED]
```

- ✦ The *newAnalyzer.py* script → analysis skeleton generation
  - ✤ Located in *tools/PAD/Build/SampleAnalyzer*
  - ✤ Adds an empty analysis in *tools/PAD/Build/SampleAnalyzer/User/Analyzer/*
  - ✤ Links the new analysis into the main executable (*tools/PAD/Build/Main/main.cpp*)

```
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5$] cd tools/PAD/Build/SampleAnalyzer/
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer$] ./newAnalyzer.py atlas_susy_2018_32 atlas_susy_2018_32
        A new class called 'atlas_susy_2018_32' will be created.
        Done !
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer$]
```

Location of the analysis files

✦ Modifications in *tools/PAD/Build/SampleAnalyzer/User/Analyzer*

```
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer$] cd User/Analyzer/
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer/User/Analyzer$] ls
analysisList.bak          atlas_exot_2015_03.o      atlas_susy_2016_07.cpp        cms_b2g_12_022.cpp        cms_exo_16_010.h          cms_sus_16_039.info
analysisList.h            atlas_exot_2016_25.cpp    atlas_susy_2016_07.h          cms_b2g_12_022.h          cms_exo_16_010.info       cms_sus_16_039.o
atlas_conf_2016_086.cpp   atlas_exot_2016_25.h      atlas_susy_2016_07.info       cms_b2g_12_022.info       cms_exo_16_010.o          cms_sus_16_052.cpp
atlas_conf_2016_086.h     atlas_exot_2016_25.info   atlas_susy_2016_07.o          cms_b2g_12_022.o          cms_exo_16_012.cpp        cms_sus_16_052.h
atlas_conf_2016_086.info  atlas_exot_2016_25.o      atlas_susy_2018_031.cpp       cms_b2g_14_004.cpp        cms_exo_16_012.h          cms_sus_16_052.info
atlas_conf_2016_086.o     atlas_exot_2016_27.cpp    atlas_susy_2018_031.h         cms_b2g_14_004.h          cms_exo_16_012.info       cms_sus_16_052.o
atlas_conf_2019_040.cpp   atlas_exot_2016_27.h      atlas_susy_2018_031.info      cms_b2g_14_004.info       cms_exo_16_012.o          cms_sus_17_001.cpp
atlas_conf_2019_040.h     atlas_exot_2016_27.info   atlas_susy_2018_031.o         cms_b2g_14_004.o          cms_exo_16_022.cpp        cms_sus_17_001.h
atlas_conf_2019_040.info  atlas_exot_2016_27.o      atlas_susy_2018_031_SRA.json  cms_exo_12_047.cpp        cms_exo_16_022.h          cms_sus_17_001.info
atlas_conf_2019_040.o     atlas_exot_2016_32.cpp    atlas_susy_2018_031_SRB.json  cms_exo_12_047.h          cms_exo_16_022.info       cms_sus_17_001.o
atlas_exot_2014_06.cpp    atlas_exot_2016_32.h      atlas_susy_2018_031_SRC.json  cms_exo_12_047.info       cms_exo_16_022.o          cms_top_17_009.cpp
atlas_exot_2014_06.h      atlas_exot_2016_32.info   atlas_susy_2018_32.cpp        cms_exo_12_047.o          cms_sus_16_033.cpp        cms_top_17_009.h
atlas_exot_2014_06.info   atlas_exot_2016_32.o      atlas_susy_2018_32.h          cms_exo_12_048.cpp        cms_sus_16_033.h          cms_top_17_009.info
atlas_exot_2014_06.o      atlas_susy_2015_06.cpp    cms_b2g_12_012.cpp            cms_exo_12_048.h          cms_sus_16_033.info       cms_top_17_009.o
atlas_exot_2015_03.cpp    atlas_susy_2015_06.h      cms_b2g_12_012.h              cms_exo_12_048.info       cms_sus_16_033.o
atlas_exot_2015_03.h      atlas_susy_2015_06.info   cms_b2g_12_012.info           cms_exo_12_048.o          cms_sus_16_039.cpp
atlas_exot_2015_03.info   atlas_susy_2015_06.o      cms_b2g_12_012.o              cms_exo_16_010.cpp        cms_sus_16_039.h
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer/User/Analyzer$]
```

- ✤ The info file to be created later
- ✤ The source file to be implemented

Location of the analysis files

✦ Modifications in *tools/PAD/Build/SampleAnalyzer/User/Analyzer*

```
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer$] cd User/Analyzer/
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer/User/Analyzer$] ls
analysisList.bak          atlas_exot_2015_03.o      atlas_susy_2016_07.cpp        cms_b2g_12_022.cpp    cms_exo_16_010.h      cms_sus_16_039.info
analysisList.h            atlas_exot_2016_25.cpp    atlas_susy_2016_07.h          cms_b2g_12_022.h      cms_exo_16_010.info   cms_sus_16_039.o
atlas_conf_2016_086.cpp   atlas_exot_2016_25.h      atlas_susy_2016_07.info       cms_b2g_12_022.info   cms_exo_16_010.o      cms_sus_16_052.cpp
atlas_conf_2016_086.h     atlas_exot_2016_25.info   atlas_susy_2016_07.o          cms_b2g_12_022.o      cms_exo_16_012.cpp    cms_sus_16_052.h
atlas_conf_2016_086.info  atlas_exot_2016_25.o      atlas_susy_2018_031.cpp       cms_b2g_14_004.cpp    cms_exo_16_012.h      cms_sus_16_052.info
atlas_conf_2016_086.o     atlas_exot_2016_27.cpp    atlas_susy_2018_031.h         cms_b2g_14_004.h      cms_exo_16_012.info   cms_sus_16_052.o
atlas_conf_2019_040.cpp   atlas_exot_2016_27.h      atlas_susy_2018_031.info      cms_b2g_14_004.info   cms_exo_16_012.o      cms_sus_17_001.cpp
atlas_conf_2019_040.h     atlas_exot_2016_27.info   atlas_susy_2018_031.o         cms_b2g_14_004.o      cms_exo_16_022.cpp    cms_sus_17_001.h
atlas_conf_2019_040.info  atlas_exot_2016_27.o      atlas_susy_2018_031_SRA.json  cms_exo_12_047.cpp    cms_exo_16_022.h      cms_sus_17_001.info
atlas_conf_2019_040.o     atlas_exot_2016_32.cpp    atlas_susy_2018_031_SRB.json  cms_exo_12_047.h      cms_exo_16_022.info   cms_sus_17_001.o
atlas_exot_2014_06.cpp    atlas_exot_2016_32.h      atlas_susy_2018_031_SRC.json  cms_exo_12_047.info   cms_exo_16_022.o      cms_top_17_009.cpp
atlas_exot_2014_06.h      atlas_exot_2016_32.info   atlas_susy_2018_32.cpp        cms_exo_12_047.o      cms_sus_16_033.cpp    cms_top_17_009.h
atlas_exot_2014_06.info   atlas_exot_2016_32.o      atlas_susy_2018_32.h          cms_exo_12_048.cpp    cms_sus_16_033.h      cms_top_17_009.info
atlas_exot_2014_06.o      atlas_susy_2015_06.cpp    cms_b2g_12_012.cpp            cms_exo_12_048.h      cms_sus_16_033.info   cms_top_17_009.o
atlas_exot_2015_03.cpp    atlas_susy_2015_06.h      cms_b2g_12_012.h              cms_exo_12_048.info   cms_sus_16_033.o
atlas_exot_2015_03.h      atlas_susy_2015_06.info   cms_b2g_12_012.info           cms_exo_12_048.o      cms_sus_16_039.cpp
atlas_exot_2015_03.info   atlas_susy_2015_06.o      cms_b2g_12_012.o              cms_exo_16_010.cpp    cms_sus_16_039.h
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer/User/Analyzer$]
```

- ✤ The info file to be created later
- ✤ The source file to be implemented

✦ Re-compiling the PAD with the new analysis

```
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build/SampleAnalyzer/User/Analyzer$] cd ../../../
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build$] source setup.sh
------------------------------------------------------------
    Your environment is properly configured for MA5
------------------------------------------------------------
[fuks@Benjamins-MacBook-Pro-8 ~/madanalysis5/tools/PAD/Build$] make
-e ------------------------------------------------------------
-e               Building MadAnalysis Job
-e ------------------------------------------------------------
-e ------------------------------------------------------------
-e                     Compilation
-e ------------------------------------------------------------
```

- ✤ Configuring the system
  → *source setup.sh*
- ✤ Updating the executable
  → *make*
- ✤ Crucial to verify the code
  → *to be done many times*

The structure of the analysis file

- Editing the *atlas_susy_2018-32.cpp* file
  - Removal of all comments
  - Three main methods

```
#include "SampleAnalyzer/User/Analyzer/atlas_susy_2018_32.h"
using namespace MA5;
using namespace std;

// -----------------------------------------------------------------------------
// Initialize
// function called one time at the beginning of the analysis
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Initialize(const MA5::Configuration& cfg, const std::map<std::string,std::string>& parameters)
{
  return true;
}

// -----------------------------------------------------------------------------
// Finalize
// function called one time at the end of the analysis
// -----------------------------------------------------------------------------
void atlas_susy_2018_32::Finalize(const SampleFormat& summary, const std::vector<SampleFormat>& files)
{
}

// -----------------------------------------------------------------------------
// Execute
// function called each time one event is read
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Execute(SampleFormat& sample, const EventFormat& event)
{
  return true;
}
```

The structure of the analysis file

✦ Editing the *atlas_susy_2018-32.cpp* file

- ✤ Removal of all comments
- ✤ Three main methods
  - ★ *Initialize*: declarations
    - → signal regions
    - → cuts
    - → histograms

```
#include "SampleAnalyzer/User/Analyzer/atlas_susy_2018_32.h"
using namespace MA5;
using namespace std;

// -----------------------------------------------------------------------------
// Initialize
// function called one time at the beginning of the analysis
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Initialize(const MA5::Configuration& cfg, const std::map<std::string,std::string>& parameters)
{
  return true;
}

// -----------------------------------------------------------------------------
// Finalize
// function called one time at the end of the analysis
// -----------------------------------------------------------------------------
void atlas_susy_2018_32::Finalize(const SampleFormat& summary, const std::vector<SampleFormat>& files)
{
}

// -----------------------------------------------------------------------------
// Execute
// function called each time one event is read
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Execute(SampleFormat& sample, const EventFormat& event)
{
  return true;
}
```

The structure of the analysis file

- ✦ Editing the *atlas_susy_2018-32.cpp* file
  - ✤ Removal of all comments
  - ✤ Three main methods
    - ★ *Initialize*: declarations
      - → signal regions
      - → cuts
      - → histograms
    - ★ *Execute*: the analysis itself

```
#include "SampleAnalyzer/User/Analyzer/atlas_susy_2018_32.h"
using namespace MA5;
using namespace std;

// -----------------------------------------------------------------------------
// Initialize
// function called one time at the beginning of the analysis
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Initialize(const MA5::Configuration& cfg, const std::map<std::string,std::string>& parameters)
{
  return true;
}

// -----------------------------------------------------------------------------
// Finalize
// function called one time at the end of the analysis
// -----------------------------------------------------------------------------
void atlas_susy_2018_32::Finalize(const SampleFormat& summary, const std::vector<SampleFormat>& files)
{
}

// -----------------------------------------------------------------------------
// Execute
// function called each time one event is read
// -----------------------------------------------------------------------------
bool atlas_susy_2018_32::Execute(SampleFormat& sample, const EventFormat& event)
{
  return true;
}
```

The structure of the analysis file

✦ Editing the *atlas_susy_2018-32.cpp* file

- Removal of all comments
- Three main methods
  - ★ *Initialize*: declarations
    - → signal regions
    - → cuts
    - → histograms
  - ★ *Execute*: the analysis itself
  - ★ *Finalize*: if needed

```
#include "SampleAnalyzer/User/Analyzer/atlas_susy_2018_32.h"
using namespace MA5;
using namespace std;

// -----------------------------------------------------------------------
// Initialize
// function called one time at the beginning of the analysis
// -----------------------------------------------------------------------
bool atlas_susy_2018_32::Initialize(const MA5::Configuration& cfg, const std::map<std::string,std::string>& parameters)
{
  return true;
}

// -----------------------------------------------------------------------
// Finalize
// function called one time at the end of the analysis
// -----------------------------------------------------------------------
void atlas_susy_2018_32::Finalize(const SampleFormat& summary, const std::vector<SampleFormat>& files)
{
}

// -----------------------------------------------------------------------
// Execute
// function called each time one event is read
// -----------------------------------------------------------------------
bool atlas_susy_2018_32::Execute(SampleFormat& sample, const EventFormat& event)
{
  return true;
}
```

Declarations:

Cuts, histograms and signal regions

The Initialize method

✦ Implementation of the *Initialize* method

The Initialize method

✦ Implementation of the *Initialize* method

- ✤ Adding an electronic signature
  - ★ Not mandatory
  - ★ Useful: contact info, papers, *etc.*

```
// Information on the analysis, authors, ...
INFO << "   <><><><><><><><><><><><><><><><><><><><><><><><><><><>" << endmsg;
INFO << "   <>  Analysis: ATLAS-SUSY-2018-32, arXiv:1908.08215  <>" << endmsg;
INFO << "   <>            (Ewkinos/sleptons: dilepton)          <>" << endmsg;
INFO << "   <>  Recaster: B. Fuks                               <>" << endmsg;
INFO << "   <>  Contact:  fuks@lpthe.jussieu.fr                 <>" << endmsg;
INFO << "   <>  Based on MadAnalysis 5 v1.8                     <>" << endmsg;
INFO << "   <>  DOI: XX.YYYY/ZZZ                                <>" << endmsg;
INFO << "   <><><><><><><><><><><><><><><><><><><><><><><><><><><>" << endmsg;
```

The Initialize method

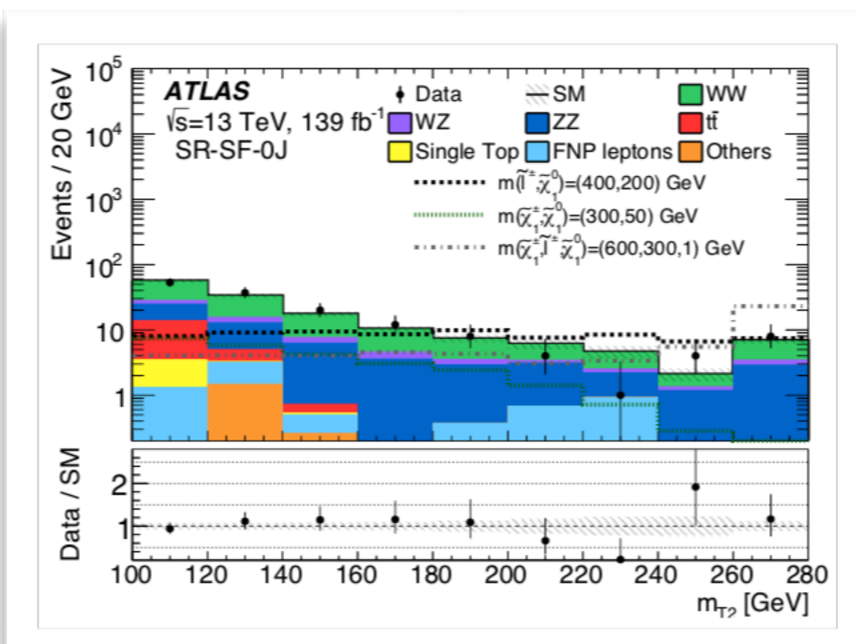
✦ Implementation of the *Initialize* method

- Adding an electronic signature
  - ★ Not mandatory
  - ★ Useful: contact info, papers, *etc.*

```
// Information on the analysis, authors, ...
INFO << "   <><><><><><><><><><><><><><><><><><><><><><><><><><><>" << endmsg;
INFO << "   <>  Analysis: ATLAS-SUSY-2018-32, arXiv:1908.08215   <>" << endmsg;
INFO << "   <>            (Ewkinos/sleptons: dilepton)           <>" << endmsg;
INFO << "   <>  Recaster: B. Fuks                                <>" << endmsg;
INFO << "   <>  Contact:  fuks@lpthe.jussieu.fr                  <>" << endmsg;
INFO << "   <>  Based on MadAnalysis 5 v1.8                      <>" << endmsg;
INFO << "   <>  DOI: XX.YYYY/ZZZ                                 <>" << endmsg;
INFO << "   <><><><><><><><><><><><><><><><><><><><><><><><><><><>" << endmsg;
```

- Declaration of the 52 analysis signal regions
- Declaration of the $\mathcal{O}(20)$ analysis cuts
- Declaration of 4 histograms
  - ★ M_{T2} distributions (fig. 6 of the ATLAS paper)
  - ★ 3 signal curves to reproduce

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	= 0	= 1	= 0	= 1
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	= 0			
Binned SRs				
m_{T2} [GeV]	∈[100,105) ∈[105,110) ∈[110,120) ∈[120,140) ∈[140,160) ∈[160,180) ∈[180,220) ∈[220,260) ∈[260,∞)			
Inclusive SRs				
m_{T2} [GeV]	∈[100,∞) ∈[160,∞) ∈[100,120) ∈[120,160)			



Declarations of all signal regions

✦ Regions, cuts and histogram declarations via the *Manager*

Declarations of all signal regions

✦ Regions, cuts and histogram declarations via the *Manager*

✦ Signal regions: *AddRegionSelection*

- ✤ One mandatory argument
  - ★ User-defined name (user-friendliness!)
- ✤ To be done 52 times

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	$=0$	$=1$	$=0$	$=1$
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	$=0$			
Binned SRs				
m_{T2} [GeV]	$\in$[100,105) $\in$[105,110) $\in$[110,120) $\in$[120,140) $\in$[140,160) $\in$[160,180) $\in$[180,220) $\in$[220,260) $\in$[260,∞)			
Inclusive SRs				
m_{T2} [GeV]	$\in$[100,∞) $\in$[160,∞) $\in$[100,120) $\in$[120,160)			

```
// Declaration of the signal regions
Manager()->AddRegionSelection("SR-DF-0J-MT2_100_105");
Manager()->AddRegionSelection("SR-DF-0J-MT2_105_110");
Manager()->AddRegionSelection("SR-DF-0J-MT2_110_120");
Manager()->AddRegionSelection("SR-DF-0J-MT2_120_140");
Manager()->AddRegionSelection("SR-DF-0J-MT2_140_160");
Manager()->AddRegionSelection("SR-DF-0J-MT2_160_180");
Manager()->AddRegionSelection("SR-DF-0J-MT2_180_220");
Manager()->AddRegionSelection("SR-DF-0J-MT2_220_260");
Manager()->AddRegionSelection("SR-DF-0J-MT2_260_inf");
Manager()->AddRegionSelection("SR-DF-0J-MT2_100_inf");
Manager()->AddRegionSelection("SR-DF-0J-MT2_160_inf");
Manager()->AddRegionSelection("SR-DF-0J-MT2_100_120");
Manager()->AddRegionSelection("SR-DF-0J-MT2_120_160");

Manager()->AddRegionSelection("SR-DF-1J-MT2_100_105");
Manager()->AddRegionSelection("SR-DF-1J-MT2_105_110");
Manager()->AddRegionSelection("SR-DF-1J-MT2_110_120");
Manager()->AddRegionSelection("SR-DF-1J-MT2_120_140");
Manager()->AddRegionSelection("SR-DF-1J-MT2_140_160");
Manager()->AddRegionSelection("SR-DF-1J-MT2_160_180");
Manager()->AddRegionSelection("SR-DF-1J-MT2_180_220");
Manager()->AddRegionSelection("SR-DF-1J-MT2_220_260");
Manager()->AddRegionSelection("SR-DF-1J-MT2_260_inf");
Manager()->AddRegionSelection("SR-DF-1J-MT2_100_inf");
Manager()->AddRegionSelection("SR-DF-1J-MT2_160_inf");
Manager()->AddRegionSelection("SR-DF-1J-MT2_100_120");
Manager()->AddRegionSelection("SR-DF-1J-MT2_120_160");
```

Declarations of all cuts

✦ Regions, cuts and histogram declarations via the *Manager*

Declarations of all cuts

✦ Regions, cuts and histogram declarations via the *Manager*

✦ Cuts: *AddCut*

- ✤ One mandatory argument
  - ★ User-defined name (user-friendliness!)
- ✤ One optional argument
  - ★ Linking cuts to regions
- ✤ Selection and preselection cuts similar
  - ★ Preselection cuts linked to all regions
  - ★ Selection cuts linked to some regions

```
// Declaration of the preselection cuts
Manager()->AddCut("OS dilepton");
Manager()->AddCut("Mll > 100 GeV");
Manager()->AddCut("MET > 110 GeV");
Manager()->AddCut("MET sign. > 10");
Manager()->AddCut("b veto");
```

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	= 0	= 1	= 0	= 1
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	= 0			
Binned SRs				
m_{T2} [GeV]	∈[100,105)			
	∈[105,110)			
	∈[110,120)			
	∈[120,140)			
	∈[140,160)			
	∈[160,180)			
	∈[180,220)			
	∈[220,260)			
	∈[260,∞)			
Inclusive SRs				
m_{T2} [GeV]	∈[100,∞)			
	∈[160,∞)			
	∈[100,120)			
	∈[120,160)			

```
// Njet cuts
std::string ZeroJet[] = {  "SR-DF-0J-MT2_100_105", "SR-DF-0J-MT2_105_110",
    "SR-DF-0J-MT2_110_120", "SR-DF-0J-MT2_120_140", "SR-DF-0J-MT2_140_160",
    "SR-DF-0J-MT2_160_180", "SR-DF-0J-MT2_180_220", "SR-DF-0J-MT2_220_260",
    "SR-DF-0J-MT2_260_inf", "SR-DF-0J-MT2_100_inf", "SR-DF-0J-MT2_160_inf",
    "SR-DF-0J-MT2_100_120", "SR-DF-0J-MT2_120_160", "SR-SF-0J-MT2_100_105",
    "SR-SF-0J-MT2_105_110", "SR-SF-0J-MT2_110_120", "SR-SF-0J-MT2_120_140",
    "SR-SF-0J-MT2_140_160", "SR-SF-0J-MT2_160_180", "SR-SF-0J-MT2_180_220",
    "SR-SF-0J-MT2_220_260", "SR-SF-0J-MT2_260_inf", "SR-SF-0J-MT2_100_inf",
    "SR-SF-0J-MT2_160_inf", "SR-SF-0J-MT2_100_120", "SR-SF-0J-MT2_120_160" };
std::string OneJet[]  = {  "SR-DF-1J-MT2_100_105", "SR-DF-1J-MT2_105_110",
    "SR-DF-1J-MT2_110_120", "SR-DF-1J-MT2_120_140", "SR-DF-1J-MT2_140_160",
    "SR-DF-1J-MT2_160_180", "SR-DF-1J-MT2_180_220", "SR-DF-1J-MT2_220_260",
    "SR-DF-1J-MT2_260_inf", "SR-DF-1J-MT2_100_inf", "SR-DF-1J-MT2_160_inf",
    "SR-DF-1J-MT2_100_120", "SR-DF-1J-MT2_120_160", "SR-SF-1J-MT2_100_105",
    "SR-SF-1J-MT2_105_110", "SR-SF-1J-MT2_110_120", "SR-SF-1J-MT2_120_140",
    "SR-SF-1J-MT2_140_160", "SR-SF-1J-MT2_160_180", "SR-SF-1J-MT2_180_220",
    "SR-SF-1J-MT2_220_260", "SR-SF-1J-MT2_260_inf", "SR-SF-1J-MT2_100_inf",
    "SR-SF-1J-MT2_160_inf", "SR-SF-1J-MT2_100_120", "SR-SF-1J-MT2_120_160" };
Manager()->AddCut("Nj=0", ZeroJet);
Manager()->AddCut("Nj=1", OneJet);
```

Declarations of histograms

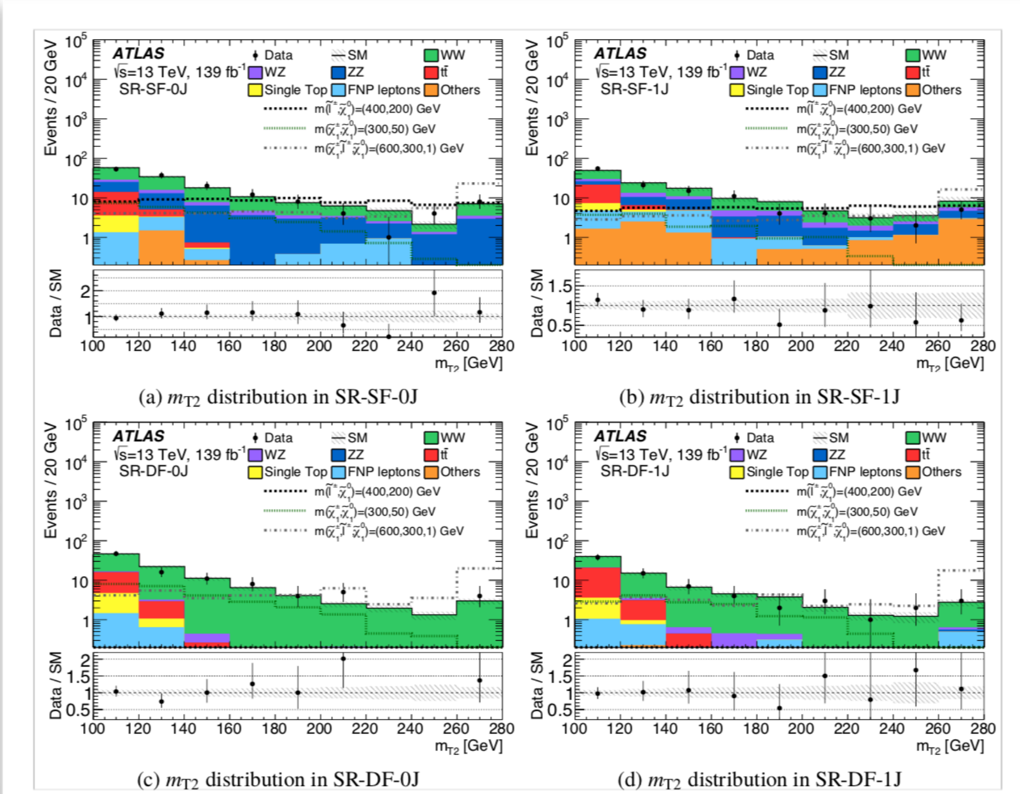
✦ Regions, cuts and histogram declarations via the *Manager*

Declarations of histograms

✦ Regions, cuts and histogram declarations via the *Manager*

✦ Histograms: *AddHisto*

- Four mandatory argument
  - ★ User-defined name (user-friendliness!)
  - ★ Number of bins, minimum, maximum
- One optional argument (as for cuts)
  - ★ Linking histograms to regions



(a) m_{T2} distribution in SR-SF-0J
(b) m_{T2} distribution in SR-SF-1J
(c) m_{T2} distribution in SR-DF-0J
(d) m_{T2} distribution in SR-DF-1J

```
// Histogram declaration
std::string SF0J[]    = {  "SR-SF-0J-MT2_100_105", "SR-SF-0J-MT2_105_110",
    "SR-SF-0J-MT2_110_120", "SR-SF-0J-MT2_120_140", "SR-SF-0J-MT2_140_160",
    "SR-SF-0J-MT2_160_180", "SR-SF-0J-MT2_180_220", "SR-SF-0J-MT2_220_260",
    "SR-SF-0J-MT2_260_inf", "SR-SF-0J-MT2_100_inf", "SR-SF-0J-MT2_160_inf",
    "SR-SF-0J-MT2_100_120", "SR-SF-0J-MT2_120_160" };
std::string SF1J[]    = {  "SR-SF-1J-MT2_100_105",
    "SR-SF-1J-MT2_105_110", "SR-SF-1J-MT2_110_120", "SR-SF-1J-MT2_120_140",
    "SR-SF-1J-MT2_140_160", "SR-SF-1J-MT2_160_180", "SR-SF-1J-MT2_180_220",
    "SR-SF-1J-MT2_220_260", "SR-SF-1J-MT2_260_inf", "SR-SF-1J-MT2_100_inf",
    "SR-SF-1J-MT2_160_inf", "SR-SF-1J-MT2_100_120", "SR-SF-1J-MT2_120_160" };
std::string DF0J[]    = {  "SR-DF-0J-MT2_100_105", "SR-DF-0J-MT2_105_110",
    "SR-DF-0J-MT2_110_120", "SR-DF-0J-MT2_120_140", "SR-DF-0J-MT2_140_160",
    "SR-DF-0J-MT2_160_180", "SR-DF-0J-MT2_180_220", "SR-DF-0J-MT2_220_260",
    "SR-DF-0J-MT2_260_inf", "SR-DF-0J-MT2_100_inf", "SR-DF-0J-MT2_160_inf",
    "SR-DF-0J-MT2_100_120", "SR-DF-0J-MT2_120_160" };
std::string DF1J[]    = {  "SR-DF-1J-MT2_100_105",
    "SR-DF-1J-MT2_105_110", "SR-DF-1J-MT2_110_120", "SR-DF-1J-MT2_120_140",
    "SR-DF-1J-MT2_140_160", "SR-DF-1J-MT2_160_180", "SR-DF-1J-MT2_180_220",
    "SR-DF-1J-MT2_220_260", "SR-DF-1J-MT2_260_inf", "SR-DF-1J-MT2_100_inf",
    "SR-DF-1J-MT2_160_inf", "SR-DF-1J-MT2_100_120", "SR-DF-1J-MT2_120_160" };
Manager()->AddHisto("MT2-SF-0J",9, 100., 280., SF0J);
Manager()->AddHisto("MT2-SF-1J",9, 100., 280., SF1J);
Manager()->AddHisto("MT2-DF-0J",9, 100., 280., DF0J);
Manager()->AddHisto("MT2-DF-1J",9, 100., 280., DF1J);
```

The *Execute method:* objects

Weight management

- Event weight initialisation mandatory
  - Correct cutflow management
    - The cutflow table of each region is incremented after each cut
    - Initialisation as an 'event passing all cuts so far' required
  - Correct weights added when filling histograms
  - First thing to be done in *Execute* (before anything else)

```
// Event weight
double myWeight;
if(Configuration().IsNoEventWeight()) myWeight=1.;
else if(event.mc()->weight()!=0.) myWeight=event.mc()->weight();
else { return false; }
Manager()->InitializeForNewEvent(myWeight);

// Security for empty events
if (event.rec()==0) return true;
```

To be included in any single analysis

Weight management

✦ Event weight initialisation mandatory

- Correct cutflow management
  - The cutflow table of each region is incremented after each cut
  - Initialisation as an 'event passing all cuts so far' required
- Correct weights added when filling histograms
- First thing to be done in *Execute* (before anything else)

```
// Event weight
double myWeight;
if(Configuration().IsNoEventWeight()) myWeight=1.;
else if(event.mc()->weight()!=0.) myWeight=event.mc()->weight();
else { return false; }
Manager()->InitializeForNewEvent(myWeight);

// Security for empty events
if (event.rec()==0) return true;
```

To be included in any single analysis

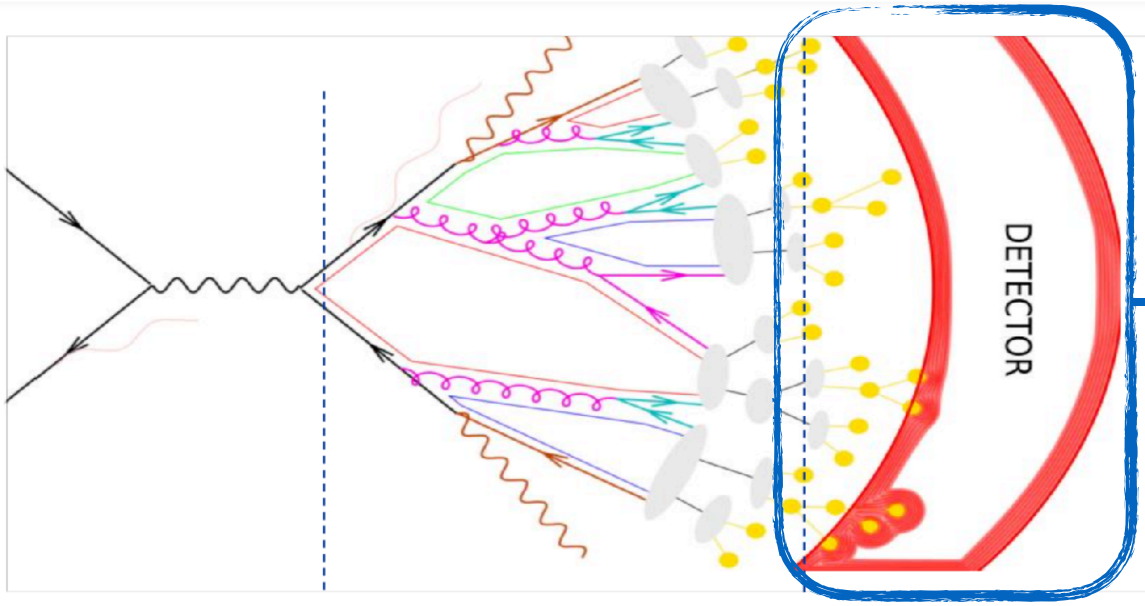
✦ Implementation details

- Weights read from the Monte Carlo information (if available)
  - If not, $w = 1$
- In case of $w = 0$: moving to the next event
- Initialisation: resetting the single event cutflows
  - The initial number of events (prior to any cut) in each region is incremented

Objects: generalities

✦ Object definitions from *event.rec()*

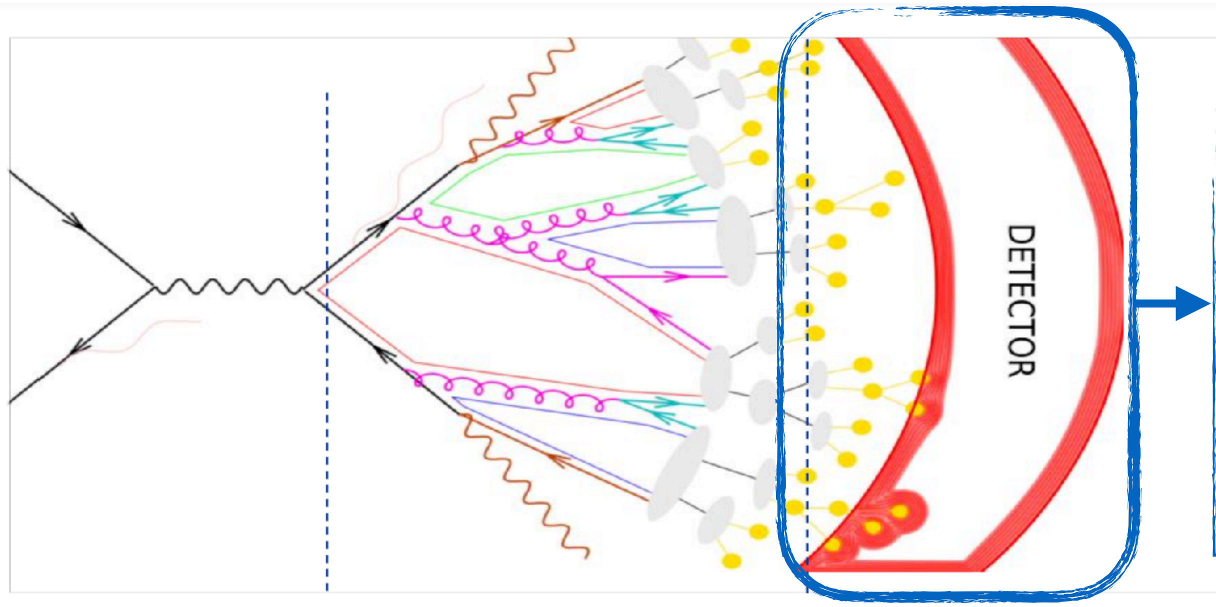
- Separate collection for each class of reconstructed objects
  - ★ Electrons, muons, jets, photons, taus, *etc.*
  - ★ All objects of a given class are included



- Detector level
  - ★ 100s of hits/tracks
  - ★ Reconstruction yields:
    - ❖ Electrons, muons, γ's
    - ❖ Hadronic taus, jets
    - ❖ Missing energy

Objects: generalities

- Object definitions from *event.rec()*
  - Separate collection for each class of reconstructed objects
    - Electrons, muons, jets, photons, taus, *etc.*
    - All objects of a given class are included

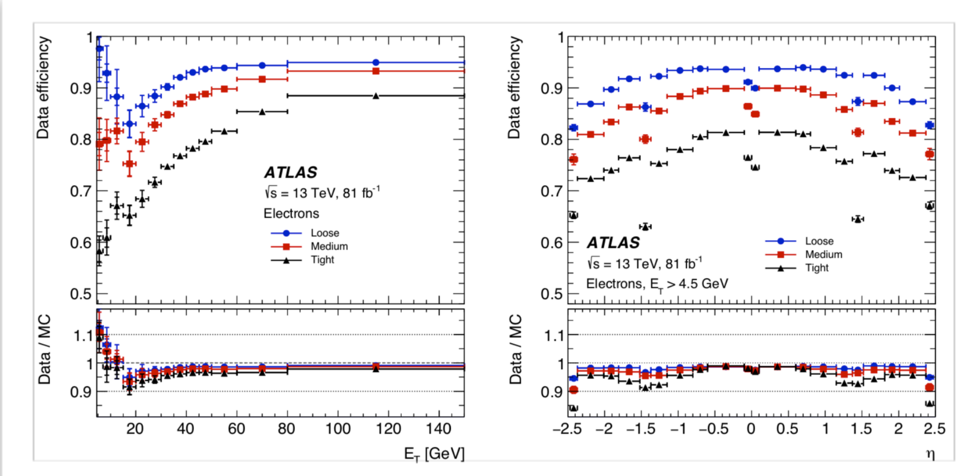


- Detector level
  - 100s of hits/tracks
  - Reconstruction yields:
    - Electrons, muons, γ's
    - Hadronic taus, jets
    - Missing energy

- Extra requirements to be implemented → analysis-relevant objects
  - p_T, η, …
  - MadAnalysis 5 treatment for electrons/jets: overlap removal

Objects: the electron example

- ✦ Two classes of electrons
  - ✤ Baseline (loose) electrons: not needed
  - ✤ Signal (tighter) leptons:
    - ★ Hard and central: p_T > 10 GeV, $|\eta|$<2.47
    - ★ Displacement ignored
    - ★ Tight reconstruction (DELPHES card)

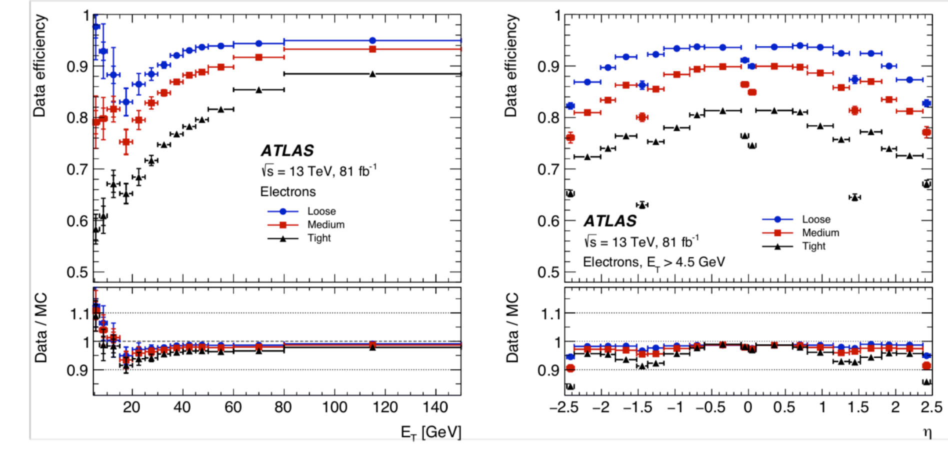


Baseline electron candidates are reconstructed using clusters of energy deposits in the electromagnetic calorimeter that are matched to an ID track. They are required to satisfy a *Loose* likelihood-based identification requirement [39], and to have $p_{\mathrm{T}} > 10$ GeV and $|\eta| < 2.47$. They are also required to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex,[3] where z_0 is the longitudinal impact parameter relative to the primary vertex. Signal electrons are required to satisfy a *Tight* identification requirement [39] and the track associated with the signal electron is required to have $|d_0|/\sigma(d_0) < 5$, where d_0 is the transverse impact parameter relative to the reconstructed primary vertex and $\sigma(d_0)$ is its error.

Objects: the electron example

✦ Two classes of electrons

- ✤ Baseline (loose) electrons: not needed
- ✤ Signal (tighter) leptons:
  - ★ Hard and central: p_T > 10 GeV, $|\eta|$<2.47
  - ★ Displacement ignored
  - ★ Tight reconstruction (DELPHES card)



Baseline electron candidates are reconstructed using clusters of energy deposits in the electromagnetic calorimeter that are matched to an ID track. They are required to satisfy a *Loose* likelihood-based identification requirement [39], and to have $p_T > 10$ GeV and $|\eta| < 2.47$. They are also required to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex,[3] where z_0 is the longitudinal impact parameter relative to the primary vertex. Signal electrons are required to satisfy a *Tight* identification requirement [39] and the track associated with the signal electron is required to have $|d_0|/\sigma(d_0) < 5$, where d_0 is the transverse impact parameter relative to the reconstructed primary vertex and $\sigma(d_0)$ is its error.

✦ Implementation:

```
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Signal leptons
  if(eta<2.47 && pt>10.) { SignalElectrons.push_back(Lep);}
}
```

Objects: electron isolation

✦ Signal electrons are isolated

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_{T} of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_{T}. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \mathrm{GeV}/p_{\mathrm{T}}$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_{T}. For electrons with $p_{\mathrm{T}} > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_{\mathrm{T}}, 3.5\ \mathrm{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

Objects: electron isolation

✦ Signal electrons are isolated

> Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

Objects: electron isolation

✦ Signal electrons are isolated

> Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

- ✤ Tracking isolation
  - ★ Variable cone size

Objects: electron isolation

✦ Signal electrons are isolated

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

✤ Tracking isolation
- ★ Variable cone size
- ★ Low activity around the electron

Objects: electron isolation

✦ Signal electrons are isolated

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

✤ Tracking isolation
- ★ Variable cone size
- ★ Low activity around the electron

✤ Calorimetric isolation

Objects: electron isolation

✦ Signal electrons are isolated

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10$ GeV$/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\text{ GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

- ✤ Tracking isolation
  - ★ Variable cone size
  - ★ Low activity around the electron
- ✤ Calorimetric isolation
- ✤ Very hard objects: special

Objects: electron isolation

✦ Signal electrons are isolated

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \mathrm{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \mathrm{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

✦ Use of built-in dedicated methods

```
// Electrons
std::vector<const RecLeptonFormat*> SignalElectrons;
for(unsigned int i=0; i<event.rec()->electrons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->electrons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.2);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.20);
  if(pt>200.) { iso = (iso_all<std::max(0.015, 3.5/pt)); }

  // Signal leptons
  if(eta<2.47 && pt>10. && iso) { SignalElectrons.push_back(Lep);}
}
```

- ✤ Tracking isolation
  - ★ Variable cone size
  - ★ Low activity around the electron
- ✤ Calorimetric isolation
- ✤ Very hard objects: special
- ✤ Signal electrons are saved

Similar treatment for muons

Baseline muon candidates are reconstructed in the pseudorapidity range $|\eta| < 2.7$ from MS tracks matching ID tracks. They are required to have $p_T > 10$ GeV, to be within $|z_0 \sin\theta| = 0.5$ mm of the primary vertex and to satisfy the *Medium* identification requirements defined in Ref. [40]. The *Medium* identification criterion defines requirements on the number of hits in the different ID and MS subsystems, and on the significance of the charge-to-momentum ratio q/p. Finally, the track associated with the signal muon must have $|d_0|/\sigma(d_0) < 3$.

Isolation criteria are applied to signal electrons and muons. The scalar sum of the p_T of tracks inside a variable-size cone around the lepton (excluding its own track), must be less than 15% of the lepton p_T. The track isolation cone size for electrons (muons) $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is given by the minimum of $\Delta R = 10\ \text{GeV}/p_T$ and $\Delta R = 0.2\,(0.3)$. In addition, for electrons (muons) the sum of the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the lepton (excluding the energy from the lepton itself) must be less than 20% (30%) of the lepton p_T. For electrons with $p_T > 200$ GeV these isolation requirements are not applied, and instead an upper limit of $\max(0.015 \times p_T, 3.5\ \text{GeV})$ is placed on the transverse energy of the calorimeter energy clusters in a cone of $\Delta R = 0.2$ around the electron.

```
std::vector<const RecLeptonFormat*> SignalMuons;
for(unsigned int i=0; i<event.rec()->muons().size(); i++)
{
  const RecLeptonFormat *Lep = &(event.rec()->muons()[i]);

  // Kinematics
  double eta = Lep->abseta();
  double pt  = Lep->pt();

  // Isolation
  double iso_dR = std::min(10./pt,0.3);
  double iso_tracks = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), iso_dR, 0., IsolationEFlow::TRACK_COMPONENT);
  double iso_all    = PHYSICS->Isol->eflow->relIsolation(Lep,
    event.rec(), 0.2, 0., IsolationEFlow::ALL_COMPONENTS);
  bool iso = (iso_tracks<0.15 && iso_all<0.30);

  // Signal leptons
  if(eta<2.7 && pt>10. && iso) { SignalMuons.push_back(Lep);}
}
```

- **Muon definitions**
  - p_T and η requirements
  - Isolation
  - Displacement ignored

Jet definition: generalities

✦ Jets are reconstructed at the DELPHES level

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The reconstructed jets are then calibrated by the application of a jet energy scale derived from 13 TeV data and simulation [89]. Only jet candidates with $p_T > 20$ GeV and $|\eta| < 2.4$ are considered,[4] although jets with $|\eta| < 4.9$ are included in the missing transverse momentum calculation and are considered when applying the procedure to remove reconstruction ambiguities, which is described later in this Section.

To reduce the effects of pile-up, for jets with $|\eta| \leq 2.5$ and $p_T < 120$ GeV a significant fraction of the tracks associated with each jet are required to have an origin compatible with the primary vertex, as defined by the jet vertex tagger [90]. This requirement reduces jets from pile-up to 1%, with an efficiency for pure hard-scatter jets of about 90%. For jets with $|\eta| > 2.5$ and $p_T < 60$ GeV, pile-up suppression is achieved through the forward jet vertex tagger [91], which exploits topological correlations between jet pairs. Finally, events containing a jet that does not satisfy the jet-quality requirements [92, 93] are rejected to remove events impacted by detector noise and non-collision backgrounds.

Jet definition: generalities

✦ Jets are reconstructed at the DELPHES level

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The reconstructed jets are then calibrated by the application of a jet energy scale derived from 13 TeV data and simulation [89]. Only jet candidates with $p_T > 20$ GeV and $|\eta| < 2.4$ are considered,[4] although jets with $|\eta| < 4.9$ are included in the missing transverse momentum calculation and are considered when applying the procedure to remove reconstruction ambiguities, which is described later in this Section.

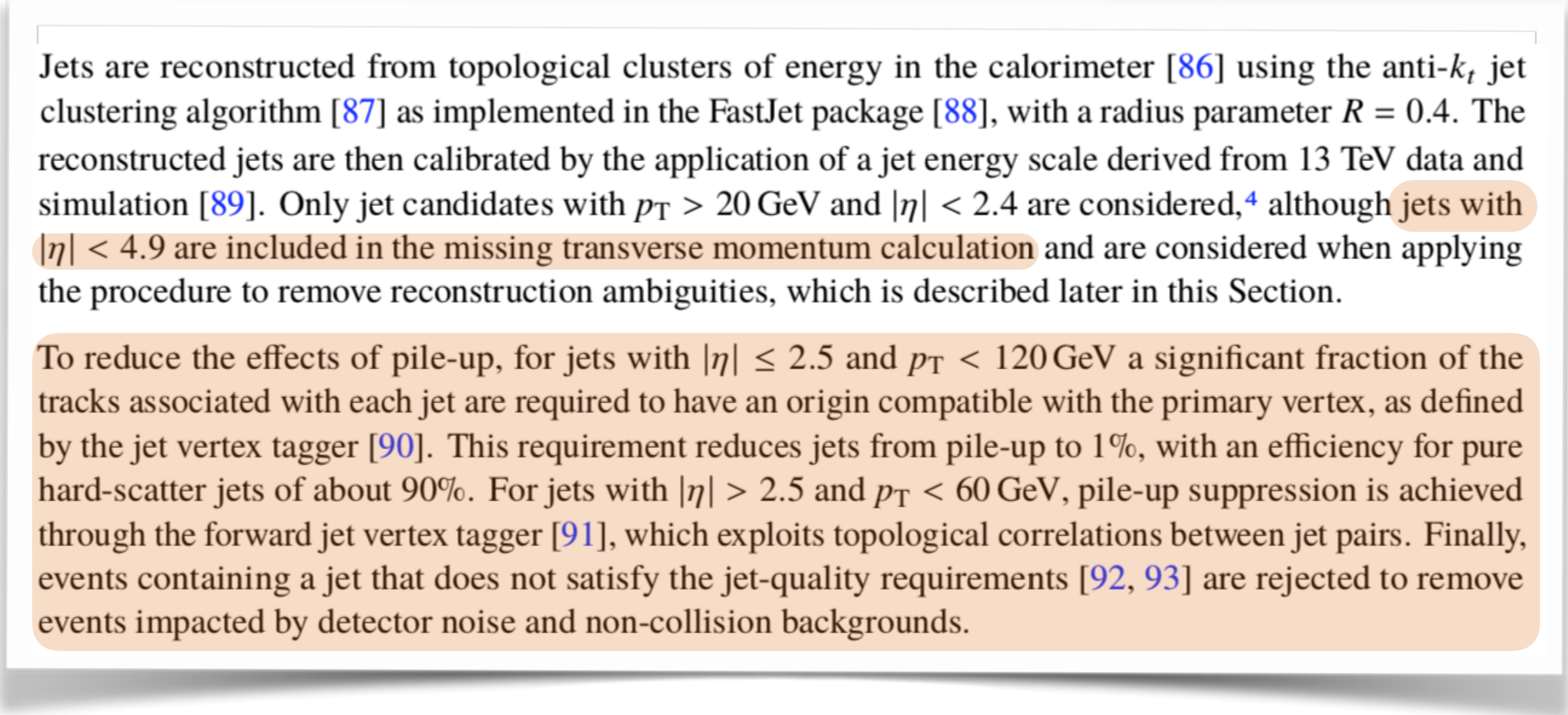
To reduce the effects of pile-up, for jets with $|\eta| \leq 2.5$ and $p_T < 120$ GeV a significant fraction of the tracks associated with each jet are required to have an origin compatible with the primary vertex, as defined by the jet vertex tagger [90]. This requirement reduces jets from pile-up to 1%, with an efficiency for pure hard-scatter jets of about 90%. For jets with $|\eta| > 2.5$ and $p_T < 60$ GeV, pile-up suppression is achieved through the forward jet vertex tagger [91], which exploits topological correlations between jet pairs. Finally, events containing a jet that does not satisfy the jet-quality requirements [92, 93] are rejected to remove events impacted by detector noise and non-collision backgrounds.

✦ Jets candidates

- ✤ p_T and η requirements

Jet definition: generalities

✦ Jets are reconstructed at the DELPHES level



Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The reconstructed jets are then calibrated by the application of a jet energy scale derived from 13 TeV data and simulation [89]. Only jet candidates with $p_T > 20$ GeV and $|\eta| < 2.4$ are considered,[4] although jets with $|\eta| < 4.9$ are included in the missing transverse momentum calculation and are considered when applying the procedure to remove reconstruction ambiguities, which is described later in this Section.

To reduce the effects of pile-up, for jets with $|\eta| \leq 2.5$ and $p_T < 120$ GeV a significant fraction of the tracks associated with each jet are required to have an origin compatible with the primary vertex, as defined by the jet vertex tagger [90]. This requirement reduces jets from pile-up to 1%, with an efficiency for pure hard-scatter jets of about 90%. For jets with $|\eta| > 2.5$ and $p_T < 60$ GeV, pile-up suppression is achieved through the forward jet vertex tagger [91], which exploits topological correlations between jet pairs. Finally, events containing a jet that does not satisfy the jet-quality requirements [92, 93] are rejected to remove events impacted by detector noise and non-collision backgrounds.

✦ Jets candidates

- ✤ p_T and η requirements

✦ Missing energy and pile-up

- ✤ *MET* automatically handled with DELPHES
- ✤ Pile-up ignored (removed from the DELPHES card)

Jet definition: implementation

✦ Straightforward: as for electrons and muons

Jets are reconstructed from topological clusters of energy in the calorimeter [86] using the anti-k_t jet clustering algorithm [87] as implemented in the FastJet package [88], with a radius parameter $R = 0.4$. The reconstructed jets are then calibrated by the application of a jet energy scale derived from 13 TeV data and simulation [89]. Only jet candidates with $p_{\mathrm{T}} > 20$ GeV and $|\eta| < 2.4$ are considered,[4] although jets with $|\eta| < 4.9$ are included in the missing transverse momentum calculation and are considered when applying the procedure to remove reconstruction ambiguities, which is described later in this Section.

```
std::vector <const RecJetFormat*> Jets;
for(unsigned int i=0; i<event.rec()->jets().size(); i++)
{
  const RecJetFormat *Jet = &(event.rec()->jets()[i]);
  double eta = Jet->abseta();
  double pt  = Jet->pt();
  if(pt>20. && eta<2.4) { Jets.push_back(Jet); }
}
```

Object overlap removal - mandatory

- In MADANALYSIS 5 electrons are also part of the jet collection
  - The double-counting should be removed → reorganisation of the code
    1. Electron candidates → into the *SignalElectrons* collection
    2. Muon candidates → into the *SignalMuons* collection
    3. Jet candidates → into the *Jets* collection
    4. Overlap removal (dedicated *JetCleaning* method)

```
Jets = PHYSICS->Isol->JetCleaning(Jets, SignalElectrons, 0.2);
```

Object overlap removal - analysis dependent

✦ The ATLAS analysis includes an overlap removal procedure

To avoid the double counting of analysis baseline objects, a procedure to remove reconstruction ambiguities is applied as follows:

- jet candidates within $\Delta R' = \sqrt{\Delta y^2 + \Delta\phi^2} = 0.2$ of an electron candidate are removed;
- jets with fewer than three tracks that lie within $\Delta R' = 0.4$ of a muon candidate are removed;
- electrons and muons within $\Delta R' = 0.4$ of the remaining jets are discarded, to reject leptons from the decay of b- or c-hadrons;
- electron candidates are rejected if they are found to share an ID track with a muon.

Object overlap removal - analysis dependent

✦ The ATLAS analysis includes an overlap removal procedure

To avoid the double counting of analysis baseline objects, a procedure to remove reconstruction ambiguities is applied as follows:

- jet candidates within $\Delta R' = \sqrt{\Delta y^2 + \Delta\phi^2} = 0.2$ of an electron candidate are removed;
- jets with fewer than three tracks that lie within $\Delta R' = 0.4$ of a muon candidate are removed;
- electrons and muons within $\Delta R' = 0.4$ of the remaining jets are discarded, to reject leptons from the decay of *b*- or *c*-hadrons;
- electron candidates are rejected if they are found to share an ID track with a muon.

✤ Jet-electron removal: already done (inherent to MADANALYSIS 5)

Object overlap removal - analysis dependent

✦ The ATLAS analysis includes an overlap removal procedure

To avoid the double counting of analysis baseline objects, a procedure to remove reconstruction ambiguities is applied as follows:

- jet candidates within $\Delta R' = \sqrt{\Delta y^2 + \Delta\phi^2} = 0.2$ of an electron candidate are removed;
- jets with fewer than three tracks that lie within $\Delta R' = 0.4$ of a muon candidate are removed;
- electrons and muons within $\Delta R' = 0.4$ of the remaining jets are discarded, to reject leptons from the decay of *b*- or *c*-hadrons;
- electron candidates are rejected if they are found to share an ID track with a muon.

✤ Jet-electron removal: already done (inherent to MADANALYSIS 5)

✤ Jet-muon removal to be implemented

```
Jets = PHYSICS->Isol->JetCleaning(Jets, SignalMuons, 0.4);
```

Object overlap removal - analysis dependent

✦ The ATLAS analysis includes an overlap removal procedure

To avoid the double counting of analysis baseline objects, a procedure to remove reconstruction ambiguities is applied as follows:

- jet candidates within $\Delta R' = \sqrt{\Delta y^2 + \Delta\phi^2} = 0.2$ of an electron candidate are removed;
- jets with fewer than three tracks that lie within $\Delta R' = 0.4$ of a muon candidate are removed;
- electrons and muons within $\Delta R' = 0.4$ of the remaining jets are discarded, to reject leptons from the decay of b- or c-hadrons;
- electron candidates are rejected if they are found to share an ID track with a muon.

✤ Jet-electron removal: already done (inherent to MADANALYSIS 5)

✤ Jet-muon removal to be implemented

```
Jets = PHYSICS->Isol->JetCleaning(Jets, SignalMuons, 0.4);
```

✤ Electron/muon removal (if within a jet) → dedicated method (to be coded)

```
SignalElectrons = Removal(SignalElectrons, Jets, 0.4);
SignalMuons     = Removal(SignalMuons, Jets, 0.4);
```

Object overlap removal - analysis dependent

✦ The ATLAS analysis includes an overlap removal procedure

To avoid the double counting of analysis baseline objects, a procedure to remove reconstruction ambiguities is applied as follows:

- jet candidates within $\Delta R' = \sqrt{\Delta y^2 + \Delta\phi^2} = 0.2$ of an electron candidate are removed;
- jets with fewer than three tracks that lie within $\Delta R' = 0.4$ of a muon candidate are removed;
- electrons and muons within $\Delta R' = 0.4$ of the remaining jets are discarded, to reject leptons from the decay of b- or c-hadrons;
- electron candidates are rejected if they are found to share an ID track with a muon.

✤ Jet-electron removal: already done (inherent to MADANALYSIS 5)

✤ Jet-muon removal to be implemented

```
Jets = PHYSICS->Isol->JetCleaning(Jets, SignalMuons, 0.4);
```

✤ Electron/muon removal (if within a jet) → dedicated method (to be coded)

```
SignalElectrons = Removal(SignalElectrons, Jets, 0.4);
SignalMuons     = Removal(SignalMuons, Jets, 0.4);
```

✤ Electron-muon removal to be implemented

```
SignalElectrons = Removal(SignalElectrons, SignalMuons, 0.2);
```

Analysis-dependent removal method

✦ The ATLAS analysis includes an overlap removal procedure

```
template<typename T1, typename T2> std::vector<const T1*>
  Removal(std::vector<const T1*> &v1, std::vector<const T2*> &v2, const MAdouble64 &drmin)
{
  // Determining with objects should be removed
  std::vector<bool> mask(v1.size(),false);
  for (MAuint32 j=0;j<v1.size();j++)
    for (MAuint32 i=0;i<v2.size();i++)
      if (v2[i]->dr(v1[j]) < drmin)
      {
        mask[j]=true;
        break;
      }

  // Building the cleaned container
  std::vector<const T1*> cleaned_v1;
  for (MAuint32 i=0;i<v1.size();i++)
    if (!mask[i]) cleaned_v1.push_back(v1[i]);

  return cleaned_v1;
}
```

Analysis-dependent removal method

✦ The ATLAS analysis includes an overlap removal procedure

```
template<typename T1, typename T2> std::vector<const T1*>
  Removal(std::vector<const T1*> &v1, std::vector<const T2*> &v2, const MAdouble64 &drmin)
{
  // Determining with objects should be removed
  std::vector<bool> mask(v1.size(),false);
  for (MAuint32 j=0;j<v1.size();j++)
    for (MAuint32 i=0;i<v2.size();i++)
      if (v2[i]->dr(v1[j]) < drmin)
      {
        mask[j]=true;
        break;
      }

  // Building the cleaned container
  std::vector<const T1*> cleaned_v1;
  for (MAuint32 i=0;i<v1.size();i++)
    if (!mask[i]) cleaned_v1.push_back(v1[i]);

  return cleaned_v1;
}
```

✤ If a v_1 element is too close to a v_2 element, the v_1 element is discarded

✤ To be implemented if necessary (no general method shipped with MA5)

b-tagging, hadronic activity and missing energy

✦ *b*-tagging efficiencies/mistagging rates at the DELPHES level

The MV2C10 boosted decision tree algorithm [41] identifies jets containing b-hadrons ('b-jets') by using quantities such as the impact parameters of associated tracks, and well-reconstructed secondary vertices. A selection that provides 85% efficiency for tagging b-jets in simulated $t\bar{t}$ events is used. The corresponding rejection factors against jets originating from c-quarks, from τ-leptons, and from light quarks and gluons in the same sample at this working point are 2.7, 6.1 and 25, respectively.

✤ Only the number of *b*-jets is needed (cf. *b*-veto)

b-tagging, hadronic activity and missing energy

✦ *b*-tagging efficiencies/mistagging rates at the DELPHES level

The MV2C10 boosted decision tree algorithm [41] identifies jets containing b-hadrons ('b-jets') by using quantities such as the impact parameters of associated tracks, and well-reconstructed secondary vertices. A selection that provides 85% efficiency for tagging b-jets in simulated $t\bar{t}$ events is used. The corresponding rejection factors against jets originating from c-quarks, from τ-leptons, and from light quarks and gluons in the same sample at this working point are 2.7, 6.1 and 25, respectively.

✤ Only the number of *b*-jets is needed (cf. *b*-veto)

✦ Hadronic activity

✤ Sum of the p_T of all remaining jets

b-tagging, hadronic activity and missing energy

✦ *b*-tagging efficiencies/mistagging rates at the DELPHES level

The MV2C10 boosted decision tree algorithm [41] identifies jets containing b-hadrons ('b-jets') by using quantities such as the impact parameters of associated tracks, and well-reconstructed secondary vertices. A selection that provides 85% efficiency for tagging b-jets in simulated $t\bar{t}$ events is used. The corresponding rejection factors against jets originating from c-quarks, from τ-leptons, and from light quarks and gluons in the same sample at this working point are 2.7, 6.1 and 25, respectively.

- Only the number of *b*-jets is needed (cf. *b*-veto)

✦ Hadronic activity

- Sum of the p_T of all remaining jets

✦ Implemented after the overlap removal

```
unsigned int nb = 0;
double HT= 0.;
for (unsigned int i=0; i<Jets.size(); i++)
{
  if(Jets[i]->btag()) { nb++; }
  HT += Jets[i]->pt();
}
```

b-tagging, hadronic activity and missing energy

✦ *b*-tagging efficiencies/mistagging rates at the DELPHES level

> The MV2C10 boosted decision tree algorithm [41] identifies jets containing b-hadrons ('b-jets') by using quantities such as the impact parameters of associated tracks, and well-reconstructed secondary vertices. A selection that provides 85% efficiency for tagging b-jets in simulated $t\bar{t}$ events is used. The corresponding rejection factors against jets originating from c-quarks, from τ-leptons, and from light quarks and gluons in the same sample at this working point are 2.7, 6.1 and 25, respectively.

- Only the number of *b*-jets is needed (cf. *b*-veto)

✦ Hadronic activity

- Sum of the p_T of all remaining jets

✦ Implemented after the overlap removal

```
unsigned int nb = 0;
double HT= 0.;
for (unsigned int i=0; i<Jets.size(); i++)
{
  if(Jets[i]->btag()) { nb++; }
  HT += Jets[i]->pt();
}
```

✦ Missing transverse energy

```
double MET = event.rec()->MET().pt();
```

- *event.rec()->MET()* is a four-momentum

Object implementation: summary

✦ Starting from the *event.rec()* collections

- ✤ Selection of electron, muon and jet candidates
- ✤ Overlap removal (mandatory + analysis dependent)

Object implementation: summary

- Starting from the *event.rec()* collections
  - Selection of electron, muon and jet candidates
  - Overlap removal (mandatory + analysis dependent)

- Creation of the signal lepton collection
  - Merging electrons and muons
  - Ordering in p_T

```
std::vector<const RecLeptonFormat*> SignalLeptons;
for(unsigned int i=0; i<SignalMuons.size(); i++)
  { SignalLeptons.push_back(SignalMuons[i]); }
for(unsigned int i=0; i<SignalElectrons.size(); i++)
  { SignalLeptons.push_back(SignalElectrons[i]); }
SORTER->sort(SignalLeptons, PTordering);
```

The *Execute method:*

cuts and histograms

Cuts and histograms

- Cuts implemented in a the same way as for their declaration
  - Ensures the correct treatment of the cutflows
  - Ensures an efficient running of the code (the *return* statement)
    - The code stops if the event is failing all cuts applied so far (for all regions)

```
if(!Manager()->ApplyCut(cut1_cond,"2_leptons")) return true;
```

Cut condition

Declared cut name

Cuts and histograms

✦ Cuts implemented in a the same way as for their declaration

- Ensures the correct treatment of the cutflows
- Ensures an efficient running of the code (the *return* statement)
  - ★ The code stops if the event is failing all cuts applied so far (for all regions)

```
if(!Manager()->ApplyCut(cut1_cond,"2_leptons")) return true;
```

Cut condition

Declared cut name

✦ Filling a declared histograms is straightforward

- Linking a value to the histogram name

```
Manager()->FillHisto("histogram name", value);
```

Histogram name

Observable value

Preselection

Reminder: 5 preselection cuts

- 2 OS leptons
- M_{ll} > 100 GeV
- *MET* > 110 GeV
- $MET/\sqrt{H_T}$ > 10 $\sqrt{\text{GeV}}$
- *b*-jet veto

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	= 0	= 1	= 0	= 1
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	= 0			

```
// Declaration of the preselection cuts
Manager()->AddCut("OS dilepton");
Manager()->AddCut("Mll > 100 GeV");
Manager()->AddCut("MET > 110 GeV");
Manager()->AddCut("MET sign. > 10");
Manager()->AddCut("b veto");
```

Preselection

✦ Reminder: 5 preselection cuts

- 2 OS leptons
- M_{ll} > 100 GeV
- *MET* > 110 GeV
- $MET/\sqrt{H_T}$ > 10 $\sqrt{\text{GeV}}$
- *b*-jet veto

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	= 0	= 1	= 0	= 1
$m_{\ell_1\ell_2}$ [GeV]	> 100		> 121.2	
E_T^{miss} [GeV]	> 110			
E_T^{miss} significance	> 10			
$n_{b\text{-tagged jets}}$	= 0			

```
// Declaration of the preselection cuts
Manager()->AddCut("OS dilepton");
Manager()->AddCut("Mll > 100 GeV");
Manager()->AddCut("MET > 110 GeV");
Manager()->AddCut("MET sign. > 10");
Manager()->AddCut("b veto");
```

✦ Straightforward implementation

```
// Preselection
bool OS = false;
if(SignalLeptons.size()>=2)
  OS = (SignalLeptons[0]->charge()!=SignalLeptons[1]->charge());
if(!Manager()->ApplyCut(SignalLeptons.size()==2 && OS,"OS dilepton")) return true;

double mll = (SignalLeptons[0]->momentum() + SignalLeptons[1]->momentum()).M();
if(!Manager()->ApplyCut(mll>100.,"Mll > 100 GeV")) return true;

if(!Manager()->ApplyCut(MET>110.,"MET > 110 GeV")) return true;

if(!Manager()->ApplyCut(MET/sqrt(HT)>10.,"MET sign. > 10")) return true;

if(!Manager()->ApplyCut(nb==0,"b veto")) return true;
```

- The ordering MUST follow the cut declaration one

  → ordered cut flows

Signal region definitions

- ✦ Many useful methods are implemented (no need for reinventing the wheel)
  - ✤ Please check the appendix of the IJMPA paper (or the reference cards)

Signal region (SR)	SR-DF-0J	SR-DF-1J	SR-SF-0J	SR-SF-1J
$n_{\text{non-}b\text{-tagged jets}}$	$= 0$	$= 1$	$= 0$	$= 1$
$m_{\ell_1\ell_2}$ [GeV]	>100		>121.2	
E_T^{miss} [GeV]	>110			
E_T^{miss} significance	>10			
$n_{b\text{-tagged jets}}$	$= 0$			

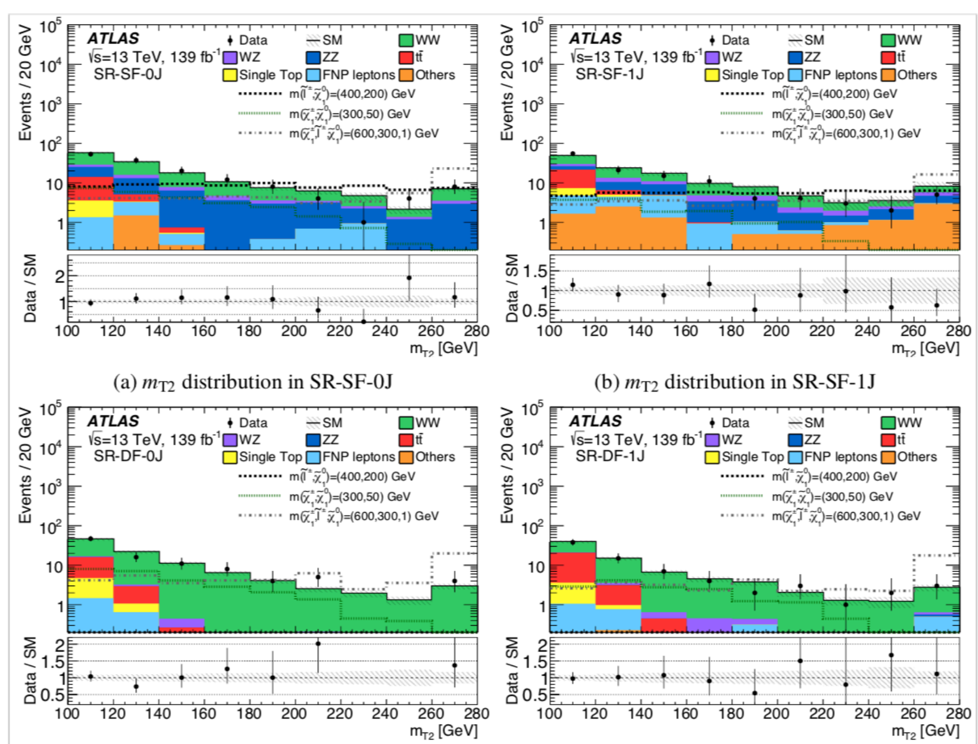
m_{T2} [GeV]	∈[100,105)
	∈[105,110)
	∈[110,120)
	∈[120,140)
	∈[140,160)
	∈[160,180)
	∈[180,220)
	∈[220,260)
	∈[260,∞)
Inclusive SRs	
m_{T2} [GeV]	∈[100,∞)
	∈[160,∞)
	∈[100,120)
	∈[120,160)

```
// Njets
unsigned int nj = Jets.size();
if(!Manager()->ApplyCut(nj==0,"Nj=0")) return true;
if(!Manager()->ApplyCut(nj==1,"Nj=1")) return true;

// SF vs DF
bool SF = (SignalLeptons[0]->isMuon()==SignalLeptons[1]->isMuon());
if(!Manager()->ApplyCut(!SF,"DF dilepton")) return true;
if(!Manager()->ApplyCut( SF,"SF dilepton")) return true;
if(!Manager()->ApplyCut(mll>121.2,"Mll > 121.2 GeV")) return true;

// MT2
double mt2 = PHYSICS->Transverse->MT2(SignalLeptons[0],SignalLeptons[1],event.rec()->MET(),0.);
if(!Manager()->ApplyCut(mt2>100. && mt2<105.,"MT2 in [100, 105] GeV")) return true;
if(!Manager()->ApplyCut(mt2>105. && mt2<110.,"MT2 in [105, 110] GeV")) return true;
if(!Manager()->ApplyCut(mt2>110. && mt2<120.,"MT2 in [110, 120] GeV")) return true;
if(!Manager()->ApplyCut(mt2>120. && mt2<140.,"MT2 in [120, 140] GeV")) return true;
if(!Manager()->ApplyCut(mt2>140. && mt2<160.,"MT2 in [140, 160] GeV")) return true;
if(!Manager()->ApplyCut(mt2>160. && mt2<180.,"MT2 in [160, 180] GeV")) return true;
if(!Manager()->ApplyCut(mt2>180. && mt2<220.,"MT2 in [180, 220] GeV")) return true;
if(!Manager()->ApplyCut(mt2>220. && mt2<260.,"MT2 in [220, 260] GeV")) return true;
if(!Manager()->ApplyCut(mt2>260.            ,"MT2 in [260, inf[ GeV")) return true;
if(!Manager()->ApplyCut(mt2>100.            ,"MT2 in [100, inf[ GeV")) return true;
if(!Manager()->ApplyCut(mt2>160.            ,"MT2 in [160, inf[ GeV")) return true;
if(!Manager()->ApplyCut(mt2>100. && mt2<120.,"MT2 in [100, 120] GeV")) return true;
if(!Manager()->ApplyCut(mt2>120. && mt2<160.,"MT2 in [120, 160] GeV")) return true;
```

Histograms

- Many useful methods are implemented (no need for reinventing the wheel)
  - Please check the appendix of the IJMPA paper (or the reference cards)
  - Histograms must be filled prior to the region specific cuts

```
// Histograms
double mt2 = PHYSICS->Transverse->MT2(SignalLeptons[0],SignalLeptons[1],event.rec()->MET(),0.);
Manager()->FillHisto("MT2-SF-0J",mt2);
Manager()->FillHisto("MT2-SF-1J",mt2);
Manager()->FillHisto("MT2-DF-0J",mt2);
Manager()->FillHisto("MT2-DF-1J",mt2);
```



(a) m_{T2} distribution in SR-SF-0J

(b) m_{T2} distribution in SR-SF-1J

(c) m_{T2} distribution in SR-DF-0J

(d) m_{T2} distribution in SR-DF-1J

Summary

Summary

✦ MADANALYSIS 5:

- ✤ A framework for collider phenomenology (parton, hadron & reco level)
- ✤ User-friendly by means of its PYTHON interface
- ✤ Flexible thanks to its C++ kernel
- ✤ Interfaced to several HEP packages

http://launchpad.net/madanalysis5

✦ Two modes of the code

- ✤ PYTHONIC: intuitive commands typed in a PYTHON interface
  - ★ Analyses performed behind the scenes
  - ★ Human readable reports as output
- ✤ C++: programming in the SAMPLEANALYZER framework (the MADANALYSIS 5 core)

✦ The LHC legacy

http://madanalysis.irmp.ucl.ac.be/wiki/PublicAnalysisDatabase

- ✤ Crucial to be able to reinterpret the LHC results in any theoretical context
- ✤ This talk presents how to implement a new analysis in the PAD from A to Z
- ✤ Do not hesitate to compile your code to verify it is bug-free

✦ Next steps

- ✤ Detector simulation and validation (for this analyses, see HEPDATA)