MadAnalysis 5

A new framework for collider phenomenology

Guillaume Serret

In collaboration with Eric Conte and Benjamin Fuks

[Conte, Fuks, Serret (arXiv:1206.1599 [hep-ph])]

Tools 2012

June 21, 2012.

Outline

Particle physics phenomenology

Tools chain

1. New physics models building

   Monte Carlo model builder program
   such as FeynRules

2. Event generation
   - **Parton-level** phenomenology

   Monte Carlo event generator
   such as MadGraph5

3. Parton showering & hadronization
   - **Hadron-level** phenomenology

   Parton showering tool
   such as Pythia or Herwig

4. Fast detector simulation
   - **Reconstructed-level** phenomenology

   Fast detector simulation program
   such as Delphes

Motivations for a new framework

- **Several formats for different phenomenological analyses**
  - **Parton** level,
  - **Hadron** level,
  - **Reconstructed** level.
- **Writing a flexible analysis**
  - **Reading** event files.
  - Applying **selection cuts**.
  - Producing of **histograms** and **cut-flow charts**.
  - **Extracting** the $^{\text{Signal}}/_{\text{Background}}$ ratio.

- **A unique framework** :
  - To work at **any sophistication level** with the associated **file format**.
  - To **build analyses** in an **user-friendly** way.
  - **Flexible**.
  - **Fast**.

Motivations for a new framework

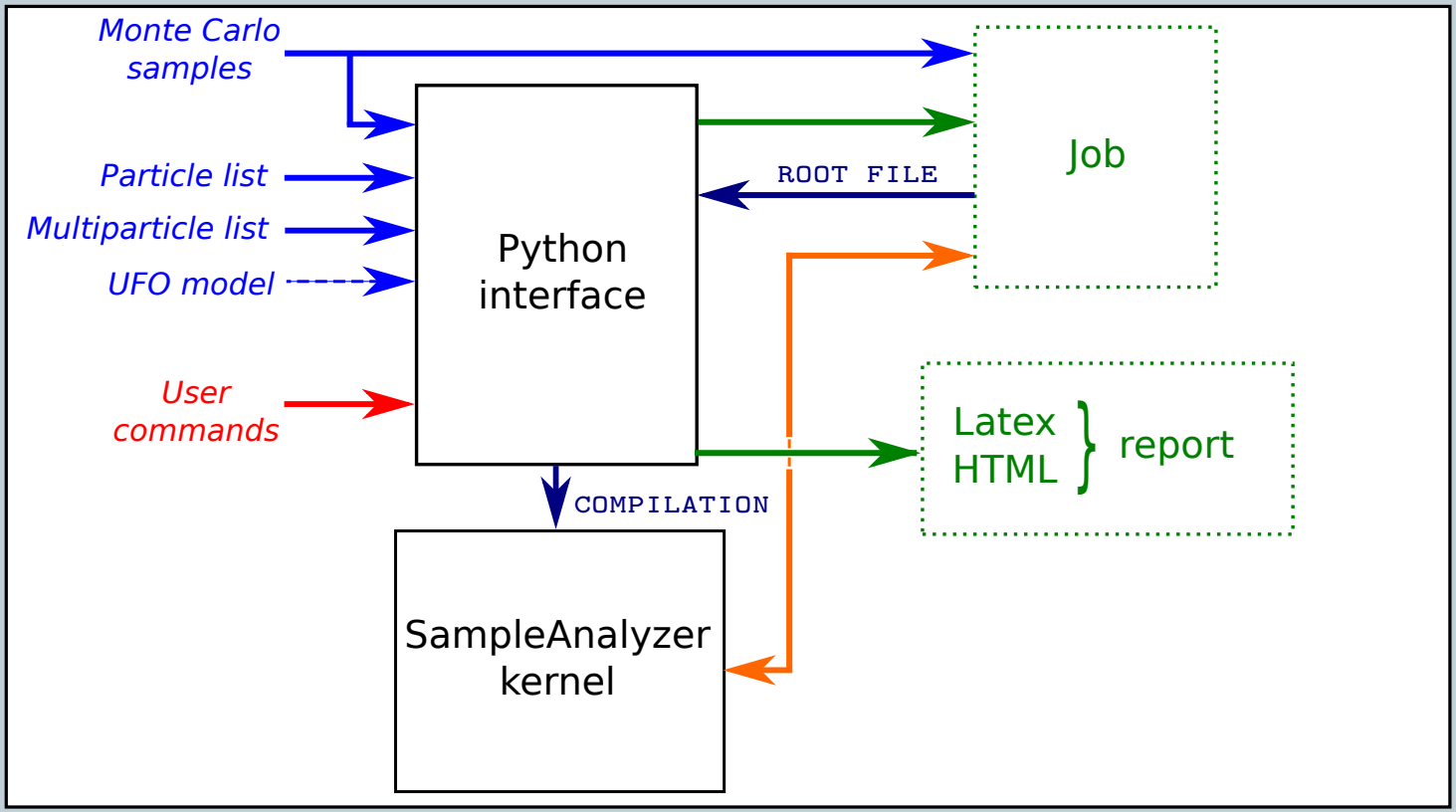
MadAnalysis 5

- **A new unique framework** :
  - To work at **any sophistication level**,
    - **partonic** level,
    - **hadronic** level,

    } LHE, StdHep, HepMC.
    - **reconstructed** level.

    } LHCO.
  - To **build analyses** in an **user-friendly** way : **normal mode**.
  - **Flexible** : **expert mode**.
  - **Fast** : t<1 min for analysing 100 000 events after hadronization.

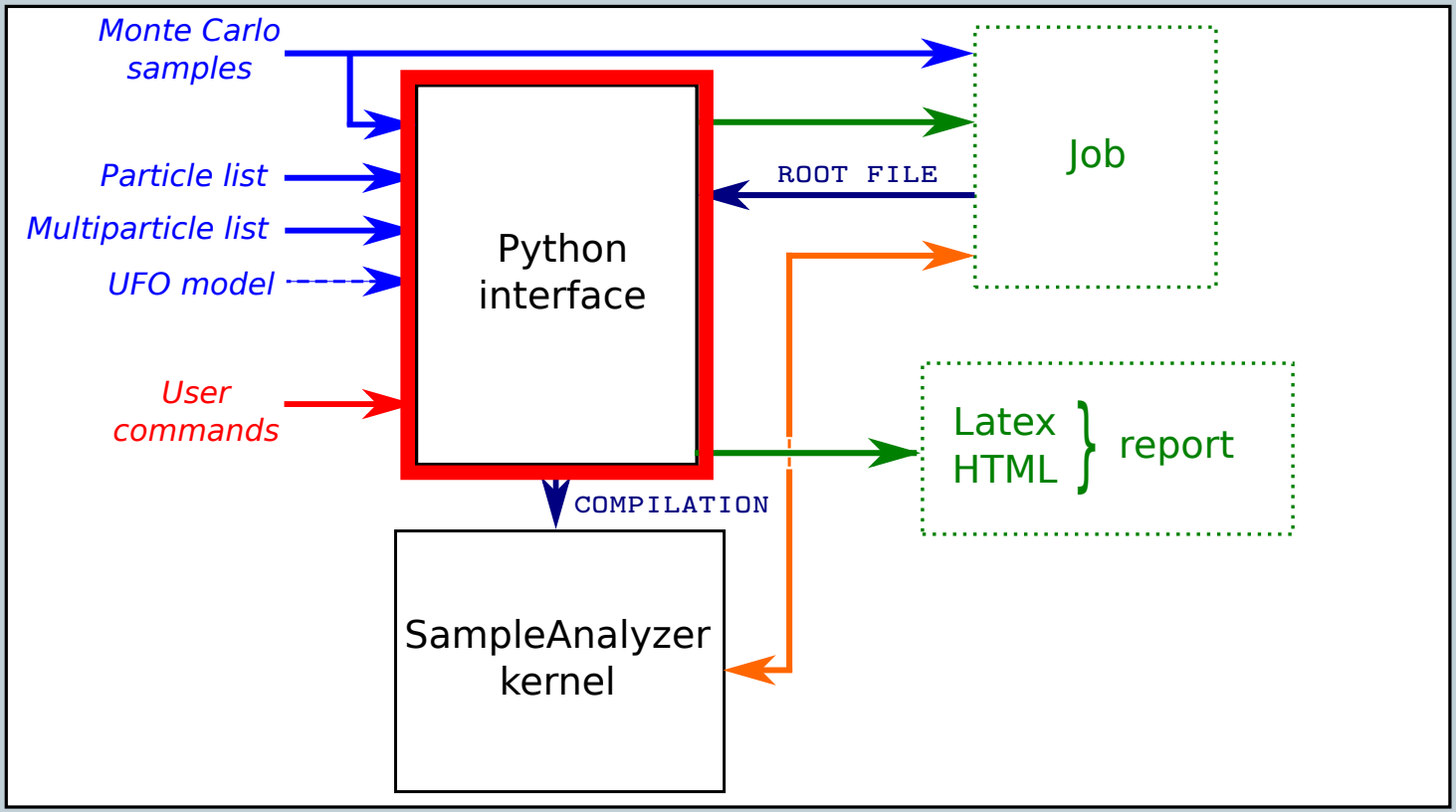
MadAnalysis 5 scheme



Two modules

- Python **command line interface** : **interactive commands**.
- SampleAnalyzer : C++/ROOT **kernel**.

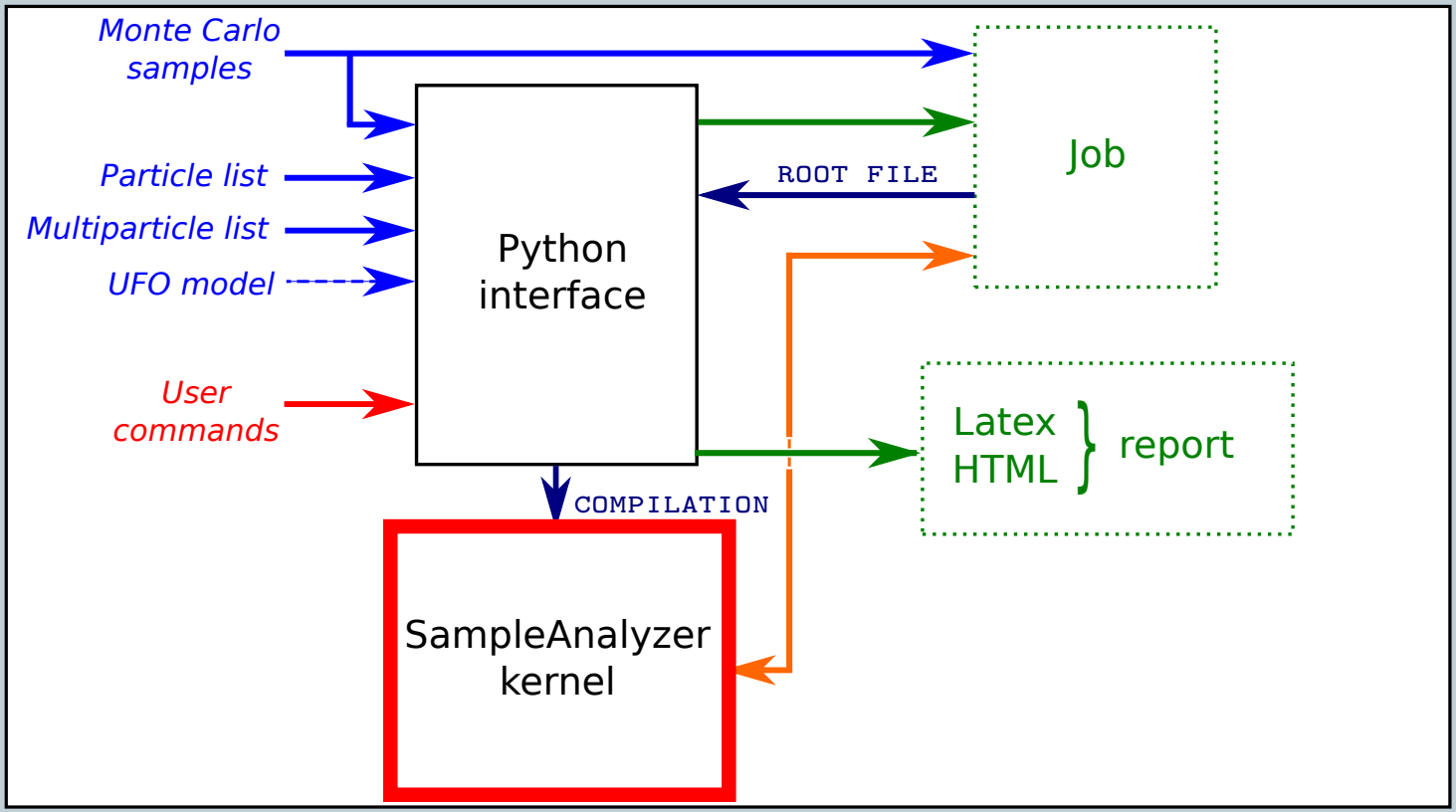
MadAnalysis 5 scheme



Two modules

- Python **command line interface** : **interactive commands**.
- SampleAnalyzer : C++/ROOT **kernel**.

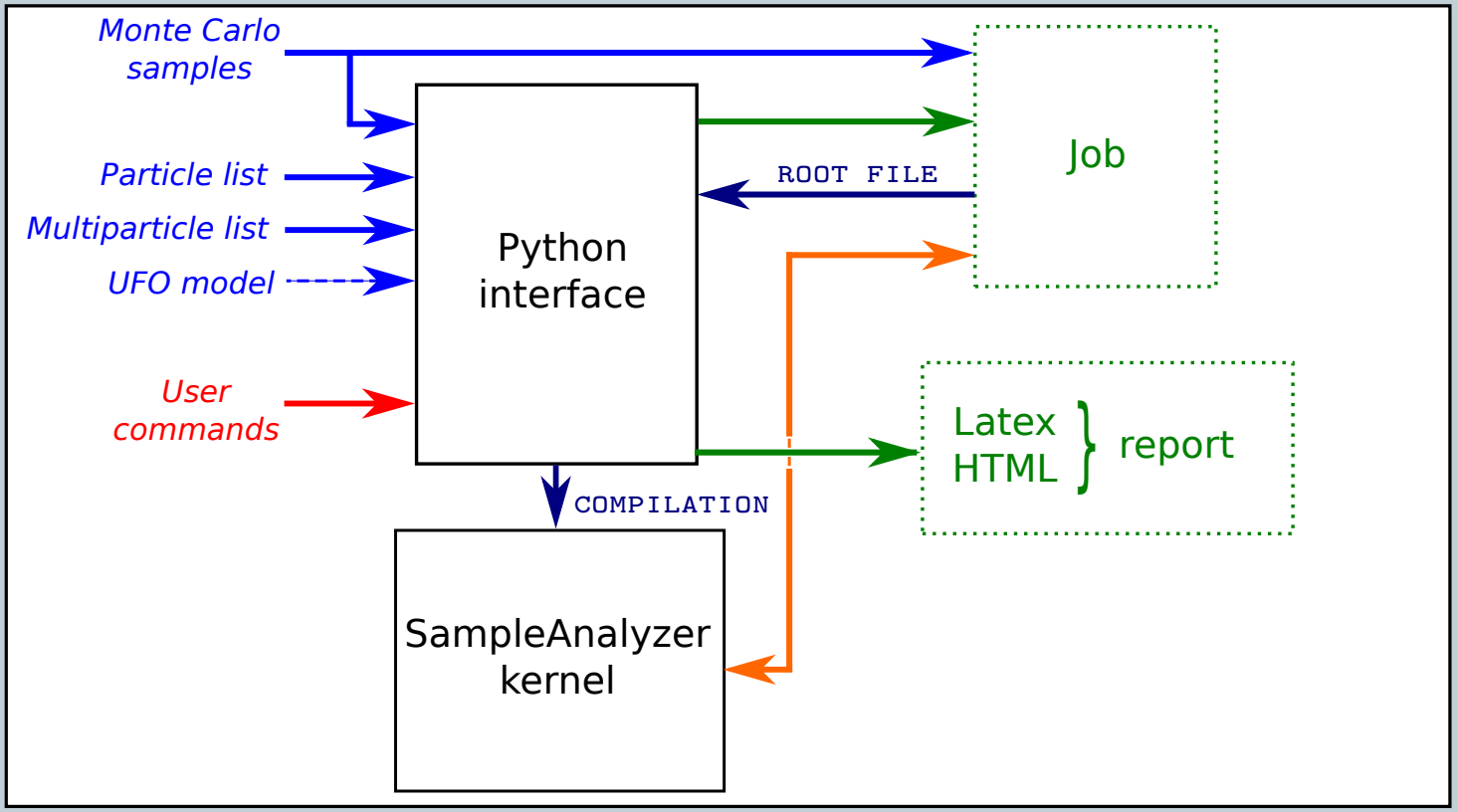
MadAnalysis 5 scheme



Two modules

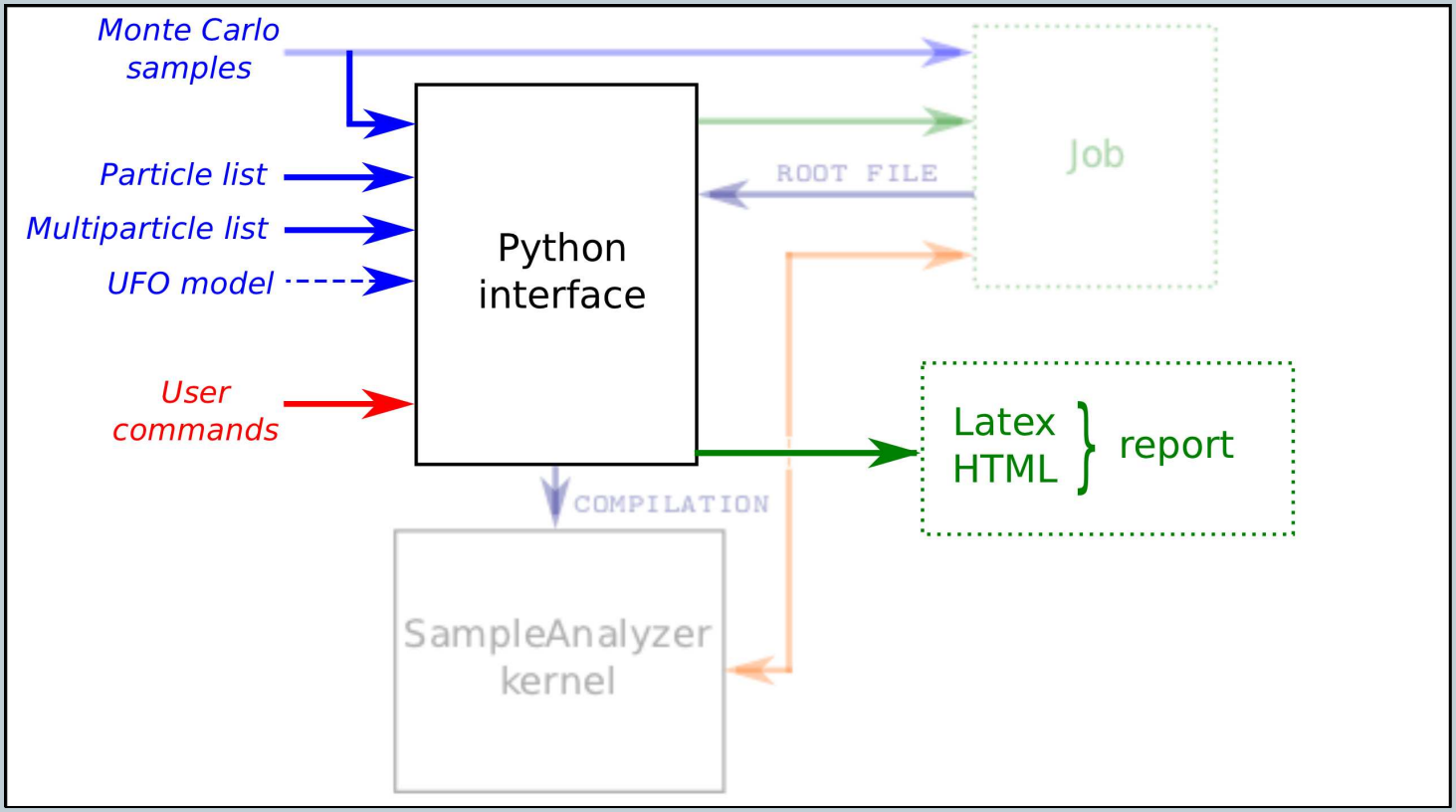
- Python **command line interface** : **interactive commands**.
- SampleAnalyzer : C++/ROOT **kernel**.

MadAnalysis 5 scheme



Two modes of running :

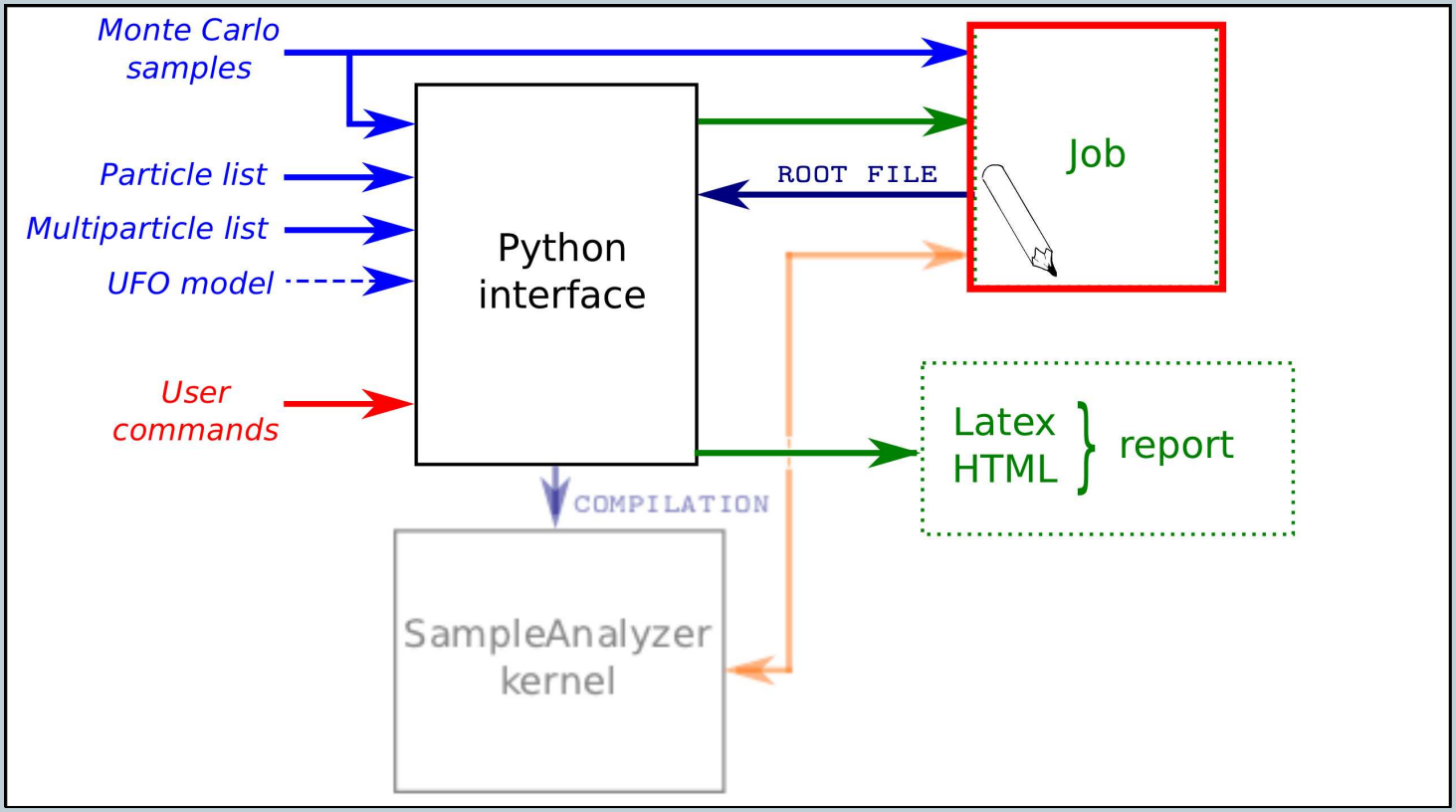
MadAnalysis 5 scheme



Two modes of running : **Normal**

* Use of the Python **interface**.
* Processing performed **behind the scene**.
* **Human readable output** : HTML, LaTeX.

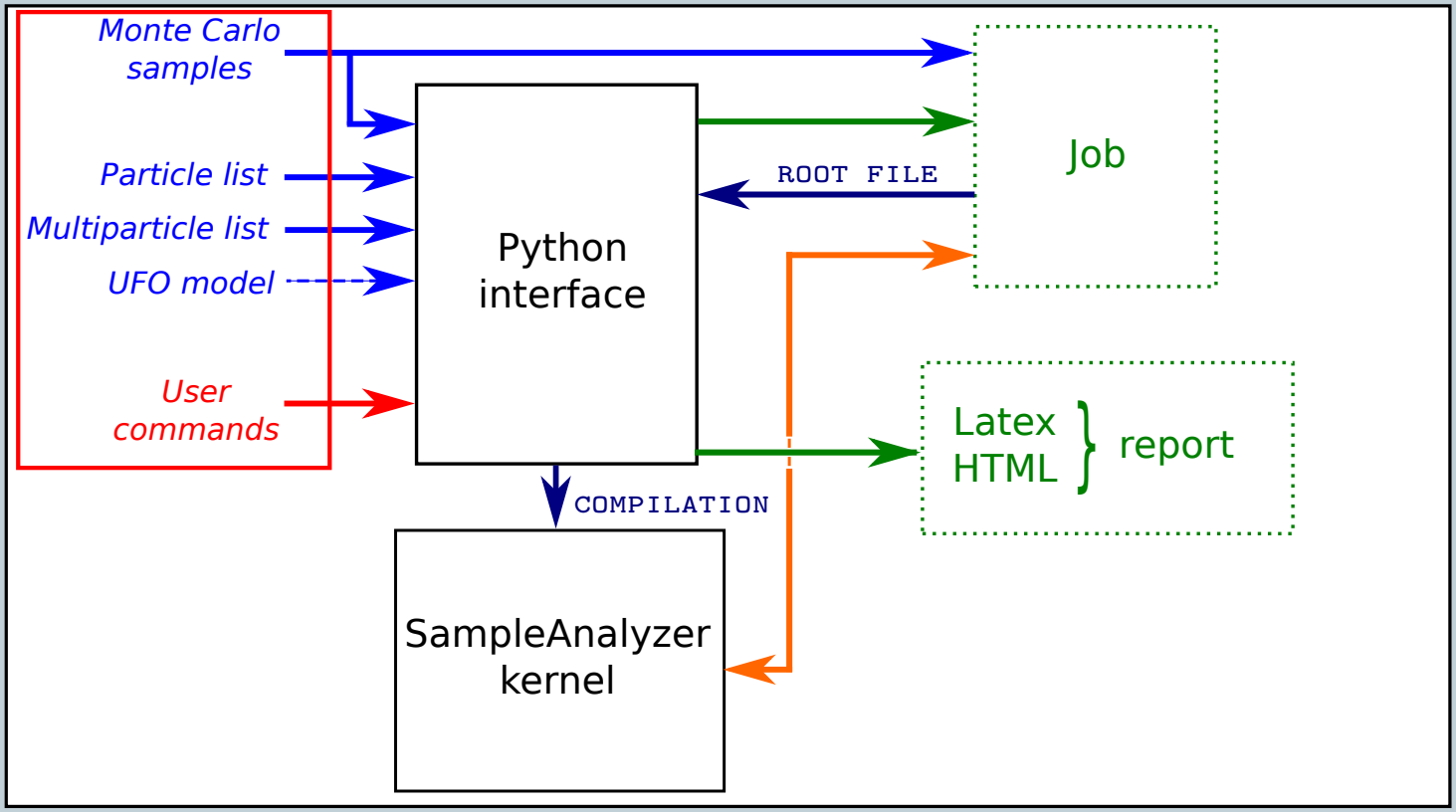
MadAnalysis 5 scheme



Two modes of running : **Expert**

* C++ **&** ROOT **skills required** .
* **Analysis template** is produced by the **interface**.
* **Developer-friendly** mode.

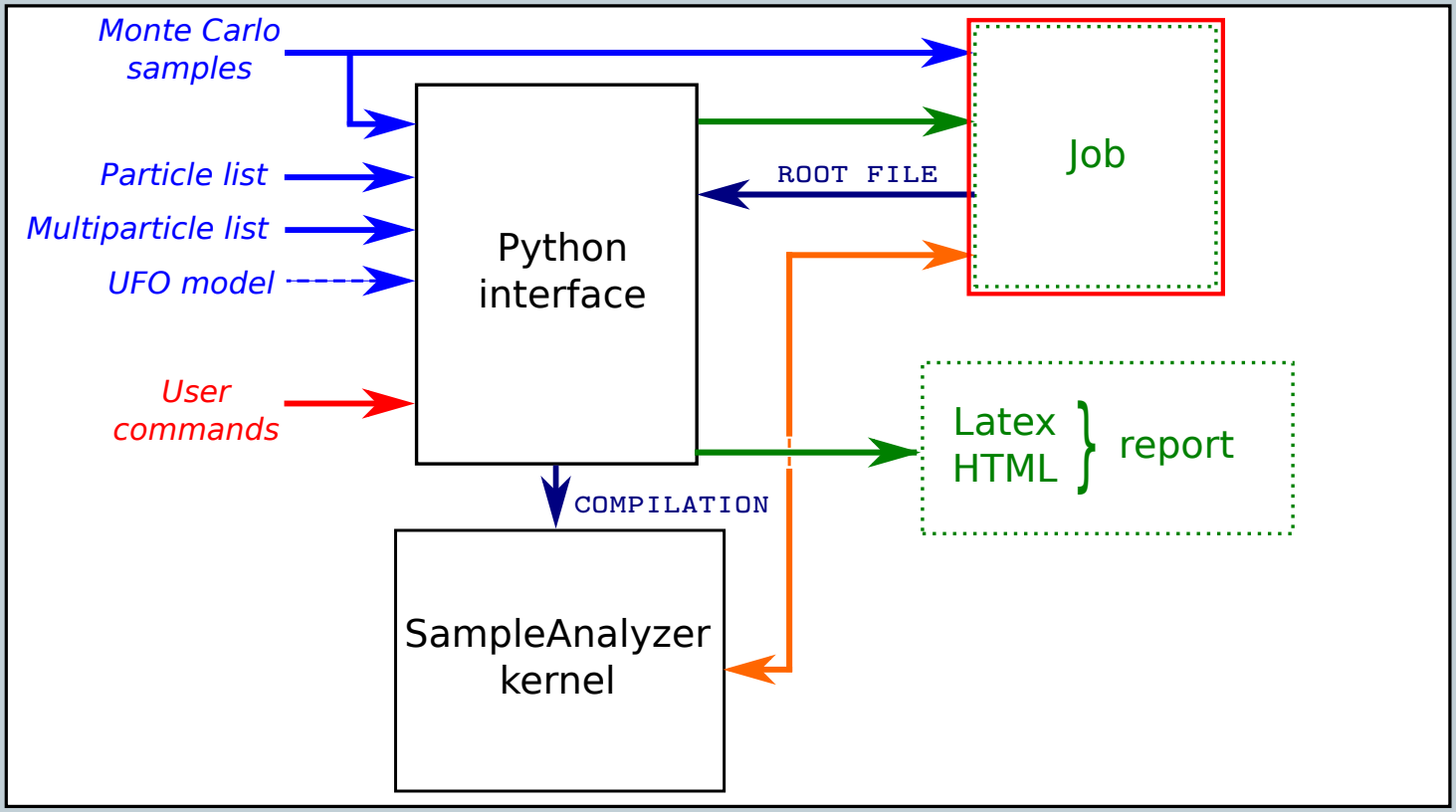
MadAnalysis 5 scheme



Inputs

- **Event files** (Monte Carlo samples) ⇔ **datasets**.
- **Particles** & **multiparticles** definition.
- **User commands**.

MadAnalysis 5 scheme



Job

- **Translation** in C++ of **user commands** by Python interface.
- Uses the SampleAnalyzer kernel.
- Generation of **results** → **report**.

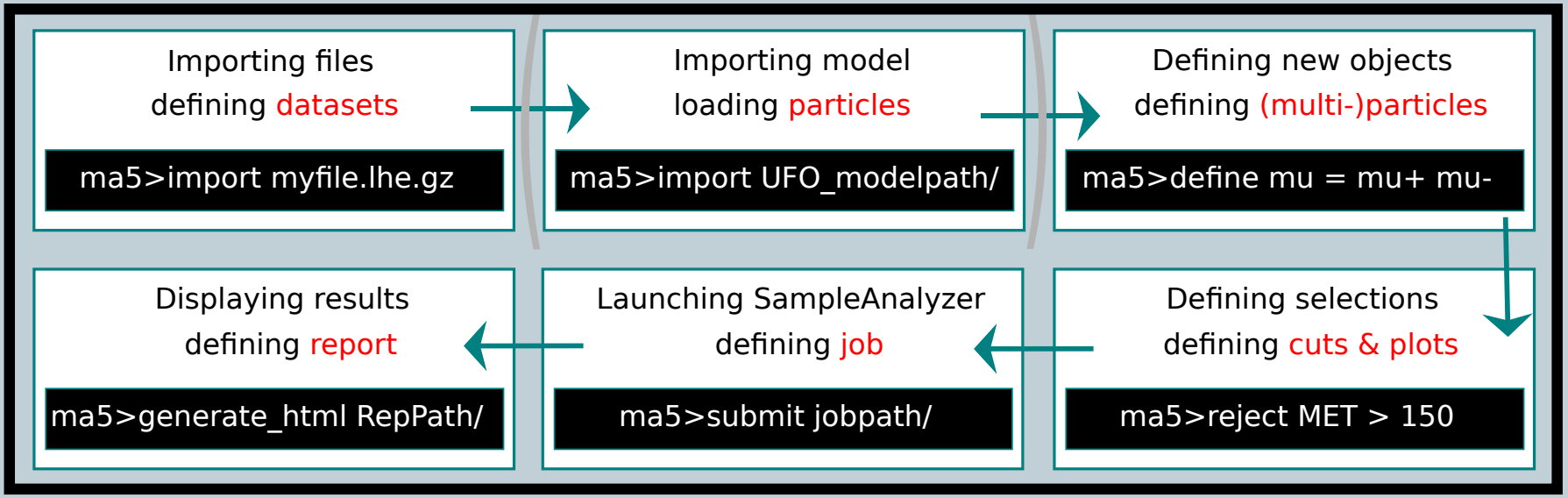
MadAnalysis 5 concepts

Command line interface

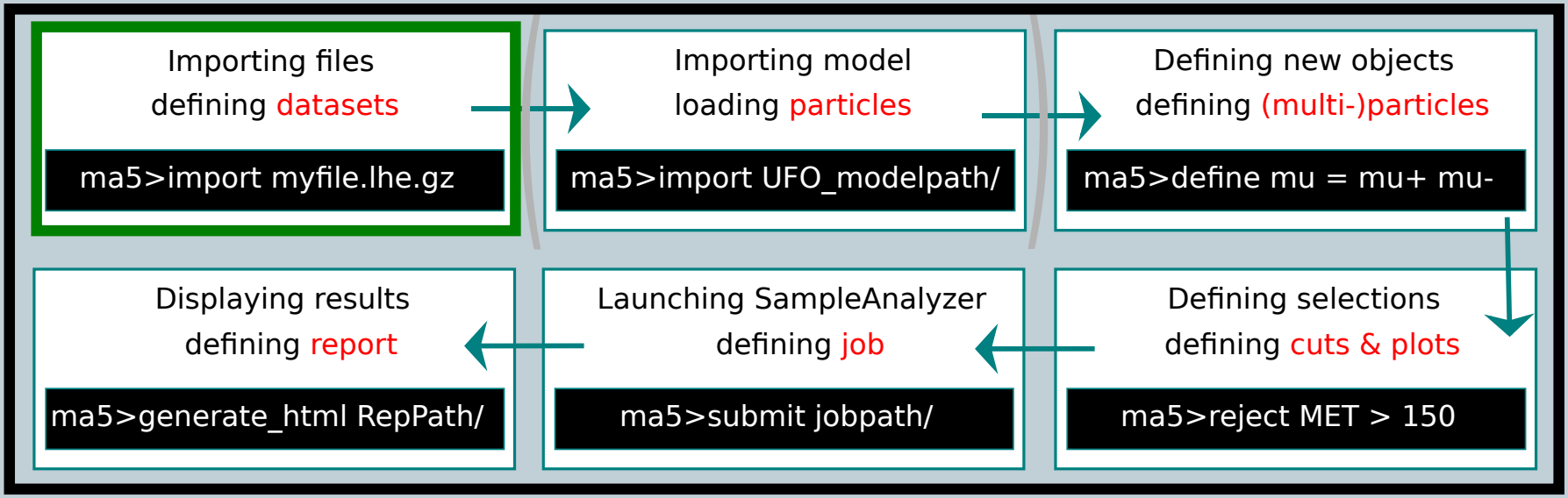
- **In-line help**.
- **Auto-completion**.

```
ma5> help <command>
```

MadAnalysis 5 concepts



MadAnalysis 5 concepts

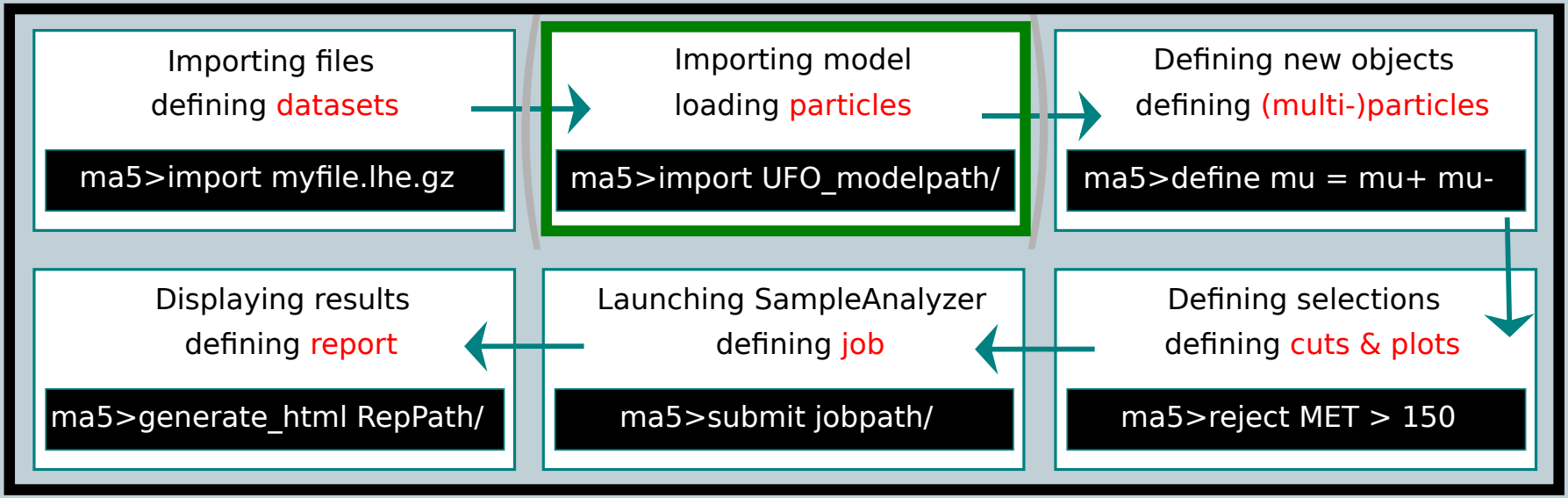


Datasets

- **Regroup** event files that can be treated in the **same way** during the analysis.

```
import ttbar1.lhe.gz as ttbar
import ttbar2.lhe.gz as ttbar
import tW1.lhe.gz as singletop
import tW2.lhe.gz as singletop
```

MadAnalysis 5 concepts

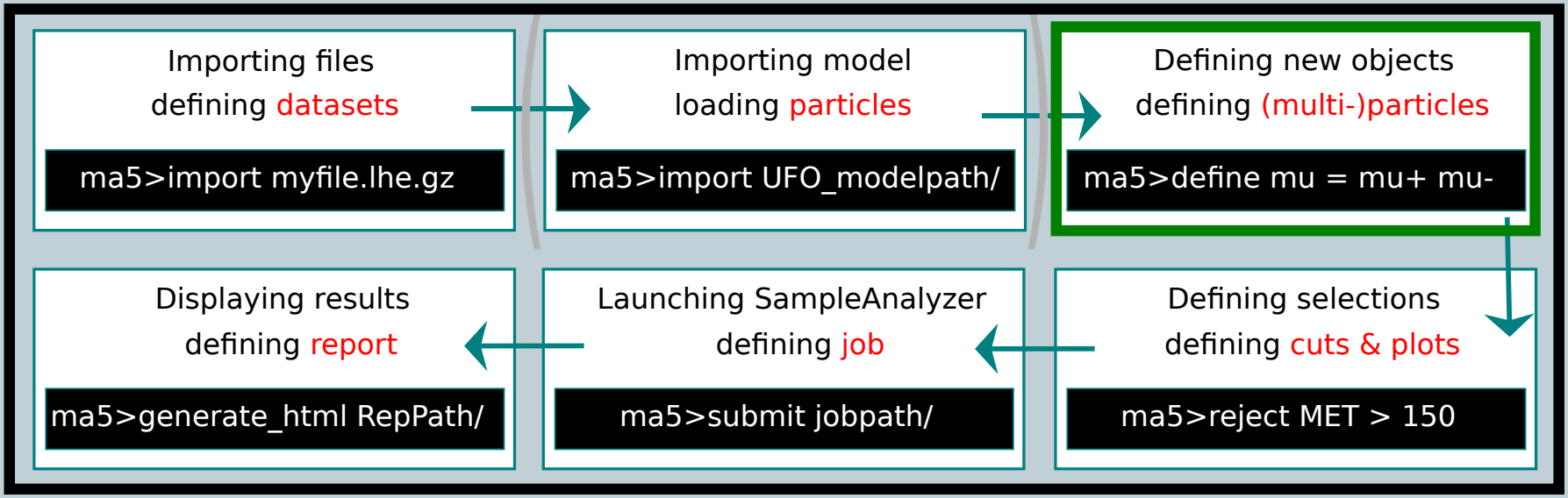


Particles

- **User-friendly** way to refer to **PDG-Id** : `mu- = -13`,
- **MSSM & SM labels** : **predefined**.
- Possibility to `import` all **new particles** from a **UFO model**.

```
define myMu- = 13
remove myMu-
```

MadAnalysis 5 concepts

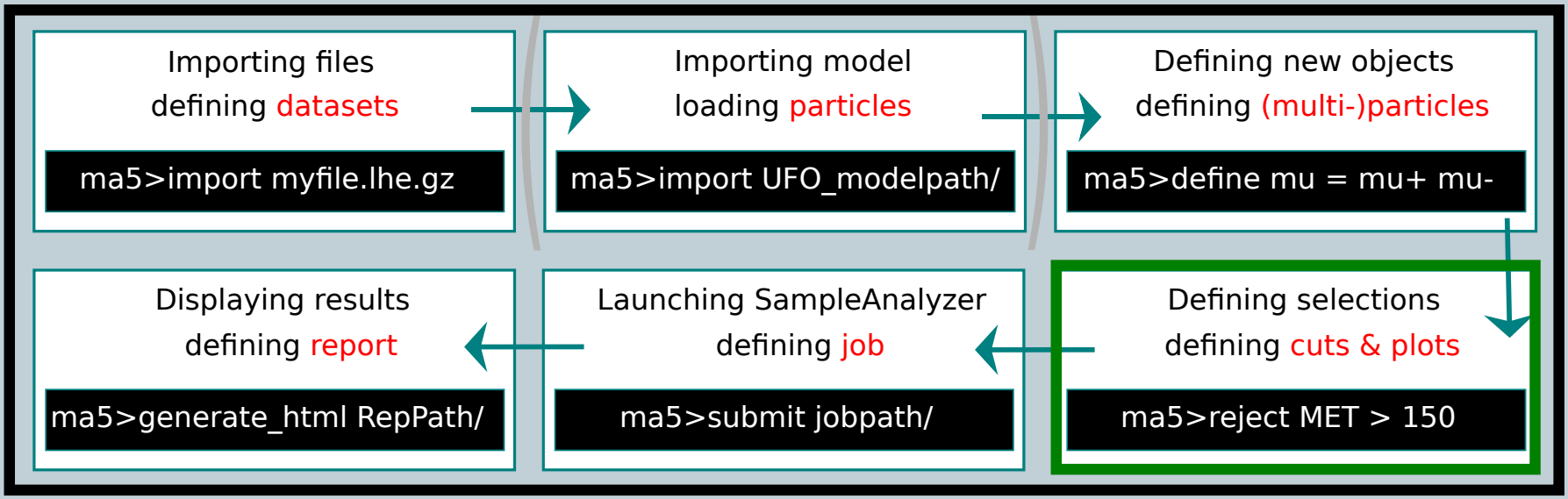


Multiparticles

- **One label** referring to **several** particles.

```
define mu = mu+ mu-
define jet = u u~ d d~ c c~
define jet = jet s s~ g
remove jet
```

MadAnalysis 5 concepts

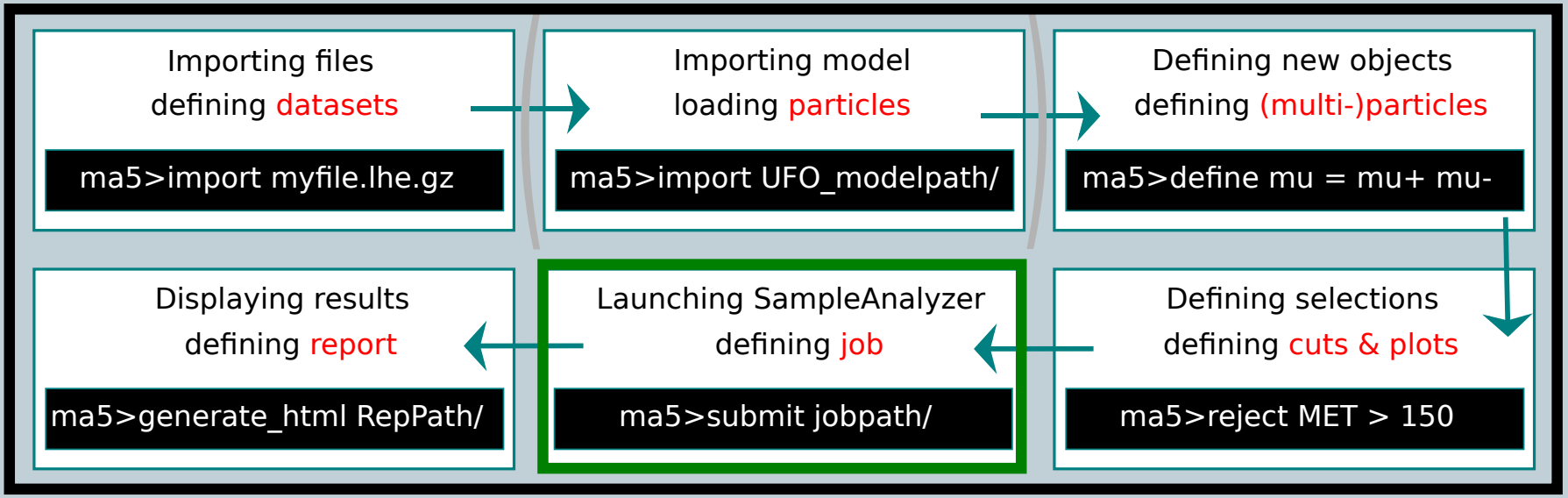


Cuts & Plots

- **Selecting/rejecting** events.
- **Selecting/rejecting** particles.
- Producing **histograms** and **cut-flow charts**.

```
reject MET > 150
select (mu) PT > 50
plot MET
plot PT(mu[1])
```

MadAnalysis 5 concepts

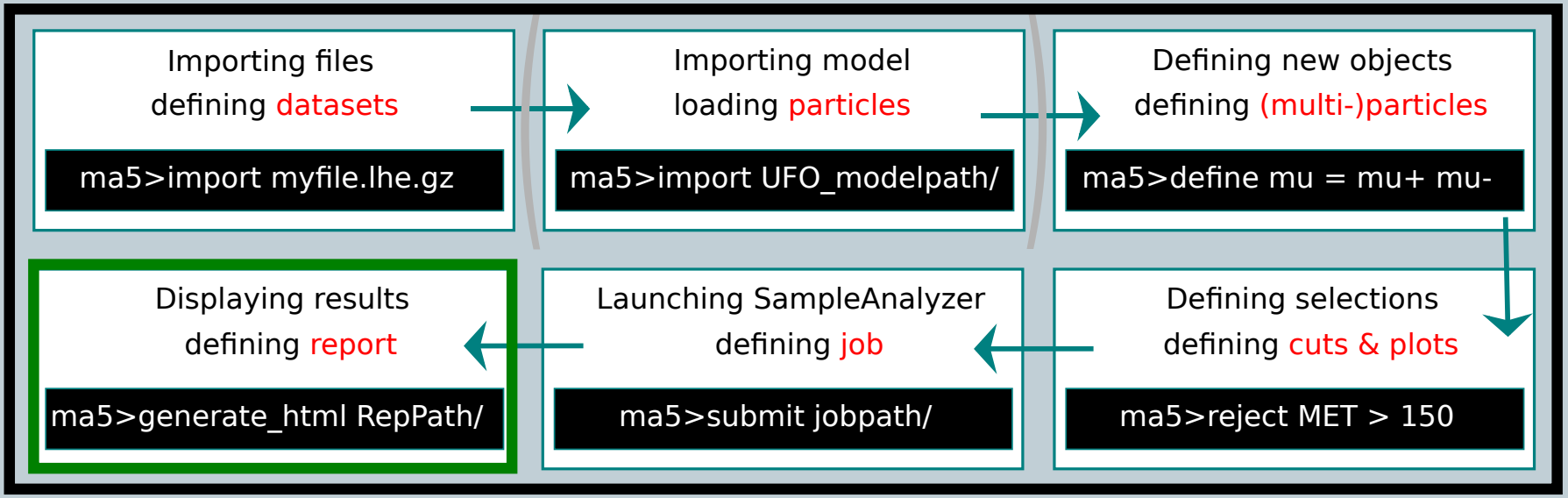


Job

- **Translation** in C++ of the analysis for SampleAnalyzer

```
submit jobpath/
```

MadAnalysis 5 concepts



Report

- **Human readable** way of displaying **all information**.

```
generate_html Path/
generate_latex Path/
generate_pdflatex Path/
```

Detailed examples

- **Installing** MadAnalysis 5 : **"nothing"** to do.
- **Launching** MadAnalysis 5 : `./bin/ma5`

Level option :

- `-P or --partonlevel (default)`
- `-H or --hadronlevel`
- `-R or --recolevel`

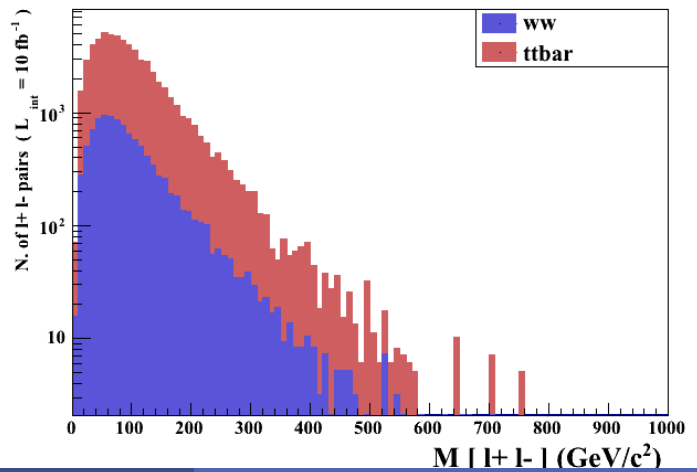
* Checking **dependencies**.
* **First time** : **Behind the scene** compilation.
* **SM** & **MSSM** particle labels **loaded**.
* Two **special multiparticles** :
  * **invisible** (**partonic**, **hadronic** levels)
  * **hadronic** (**partonic**, **hadronic** levels)

Detailed examples : partonic level

- **Plot kinematics distribution related to particle species**

* **Available observables** <OBS>
  BETA, DELTAR, E, ET, ETA, GAMMA, M, MT, P, PHI, PT, PX, PY, PZ, R, THETA, Y.
* **Scalar & vectorial sums/differences available**
  "dv","vd","d","ds","sd","r" prefixes : vd<OBS>(mu+ mu-)
* **Particle ordering (several ordering variables)**
  PT(mu+[1]), PT(mu+[2]), ... , PT(mu+[-2]) PT(mu+[-1]).
* **Particle history**
  PT(mu+ < Z), PT(l+ << t) with possible combinations : PT(mu+[1] < Z[1]).

```
ma5>import ttbar.lhe.gz as ttbar
ma5>import ww.lhe.gz as ww
ma5>plot M(l+ l-)
```

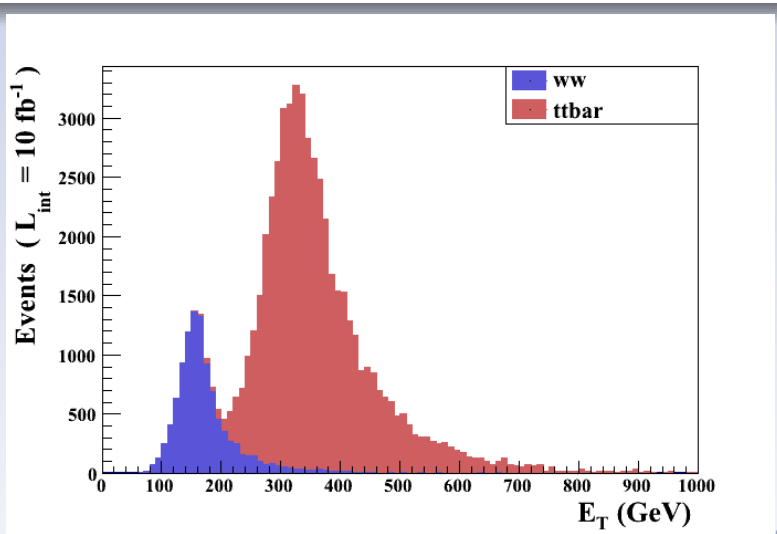


Detailed examples : partonic level

- **Plot kinematics distribution related to event**

* **Missing and visible energy**
  `MET, TET`
* **Missing and visible hadronic energy**
  `THT, MHT`
* **Center-of-mass energy (partonic)**
  `SQRTS`

```
ma5>plot TET
```



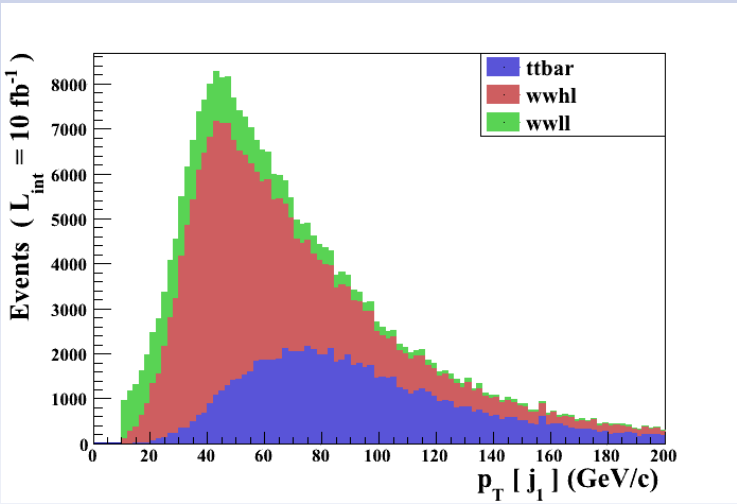
Detailed examples : reconstructed level

- **Checking particle/multiparticle content**

```
ma5>display_multiparticles
 e l l+ l- mu mu_isol ta
ma5>display e
 The multiparticle 'e' is defined by the PDG-ids -11 11.
```

- **Importing samples and producing histograms.**

```
ma5>import ttbar.lhco.gz as ttbar
ma5>import wwhl.lhco.gz as wwhl
ma5>import wwll.lhco.gz as wwll
ma5>plot PT(j[1])
ma5>plot M(l[1] j[1])
ma5>plot MET
```



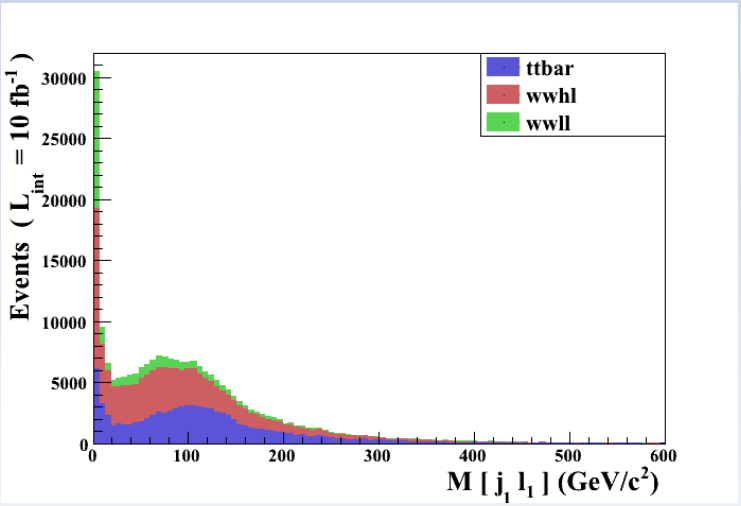
Detailed examples : reconstructed level

- **Checking particle/multiparticle content**

```
ma5>display_multiparticles
 e l l+ l- mu mu_isol ta
ma5>display e
 The multiparticle 'e' is defined by the PDG-ids -11 11.
```

- **Importing samples and producing histograms.**

```
ma5>import ttbar.lhco.gz as ttbar
ma5>import wwhl.lhco.gz as wwhl
ma5>import wwll.lhco.gz as wwll
ma5>plot PT(j[1])
ma5>plot M(l[1] j[1])
ma5>plot MET
```



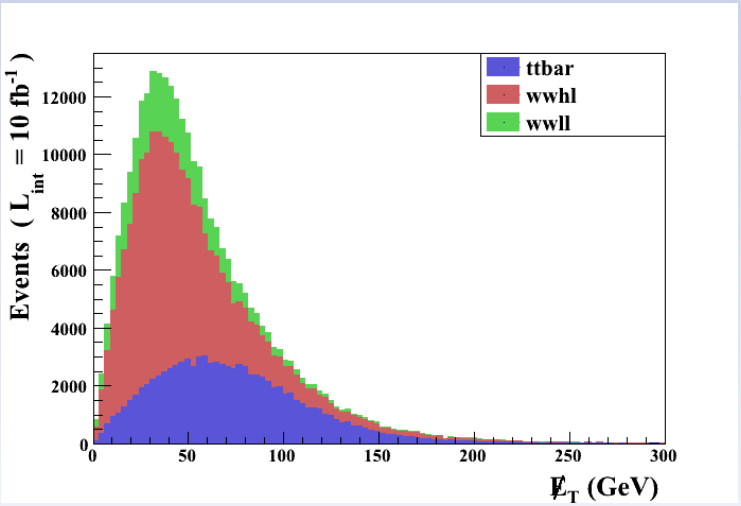
Detailed examples : reconstructed level

- **Checking particle/multiparticle content**

```
ma5>display_multiparticles
 e l l+ l- mu mu_isol ta
ma5>display e
 The multiparticle 'e' is defined by the PDG-ids -11 11.
```

- **Importing samples and producing histograms.**

```
ma5>import ttbar.lhco.gz as ttbar
ma5>import wwhl.lhco.gz as wwhl
ma5>import wwll.lhco.gz as wwll
ma5>plot PT(j[1])
ma5>plot M(l[1] j[1])
ma5>plot MET
```

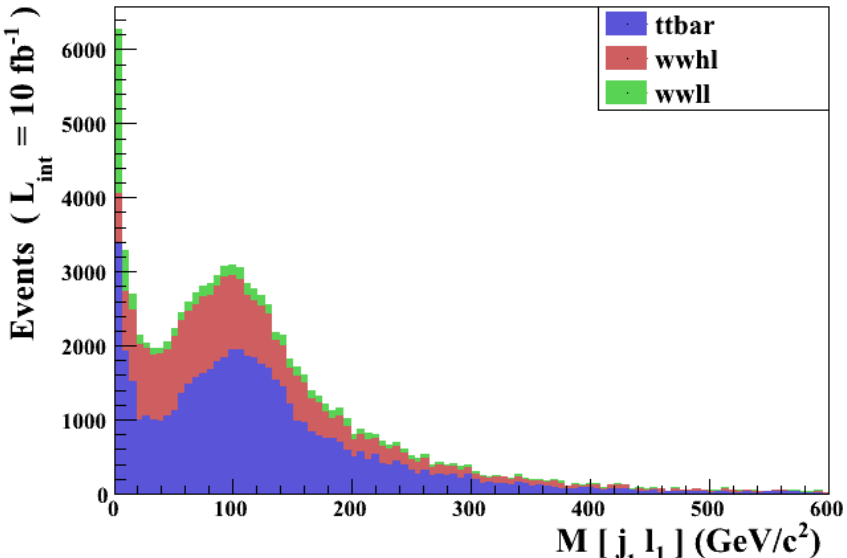


Detailed examples : **reconstructed** level

- **Apply cuts.**

* Two **keywords** : `select` & `reject`.
* Set datasets **type** as `signal` or `background`.
* User can **implement his formula** for $^{\text{Signal}}/_{\text{Background}}$ **ratio calculation**
* **Several formulas are implemented**
  `"B/(B+S)"`, `"B/S"`, `"B/sqrt(B+S)"`, `"S/(S+B)"`, `"S/B"`, `"S/sqrt(S+B)"`
  with the **associated error automatically calculated**.

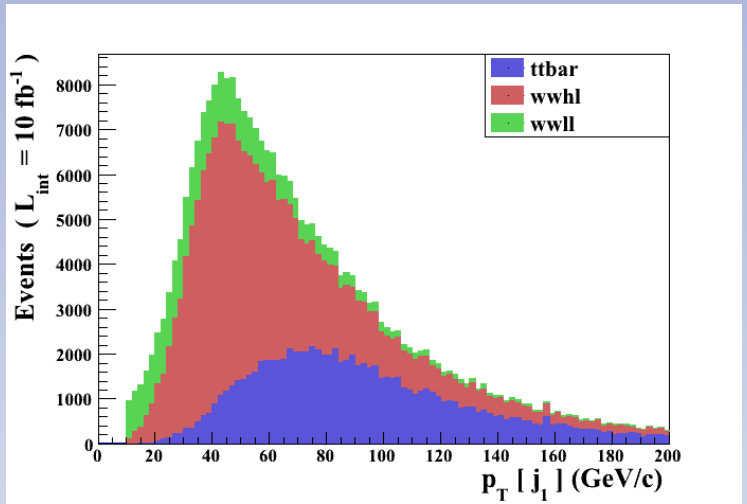
```
ma5>set ttbar.type = signal
ma5>set wwhl.type = background
ma5>set wwll.type = background
ma5>set main.SBratio = "S/B"
ma5>select (j) PT > 50
ma5>reject MET < 50
ma5>plot M(l[1] j[1])
```



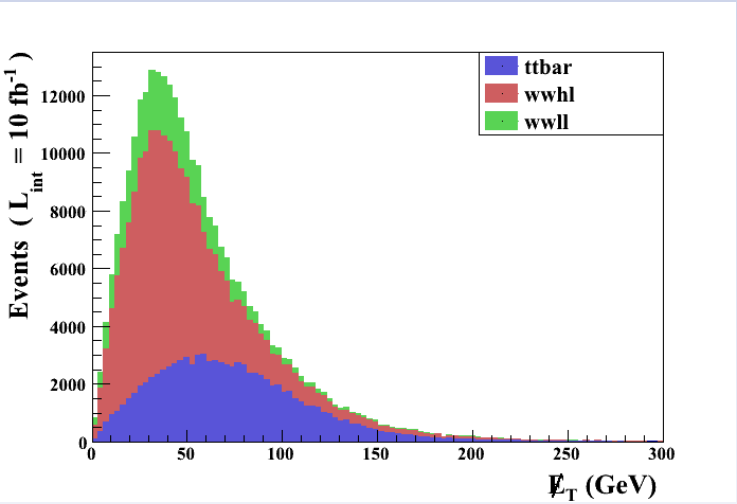
Detailed examples : **reconstructed** level

- Recall of the example :

* Jet p_T distribution.



* Event missing energy distribution.



Detailed examples : **reconstructed** level

- Recall of the example :

```
ma5>set main.SBratio = "S/B"
ma5>select (j[1]) PT > 50
ma5>reject MET > 50
```

- **Results**.

Cuts	Signal (S)	Background (B)	S vs B
Initial (no cut)	92200	197900	0.466
cut 1	92200	197900	0.466
cut 2	63516 +/- 140	72869 +/- 214	0.87164 +/- 0.00321

Formula for **uncertainty** on **S-B comparison** :
1./(B2)*sqrt(B**2*ES**2+S**2*EB**2)**.

Detailed examples

- **Available options.**

* **Datasets layout** : `set dataset.<OPT> = value`
  `backcolor, backstyle, linecolor, linestyle, linewidth, title`
* **Histograms layout** : `set selection[i].<OPT> = value`
  `logY, logX, nbins, stacking_method, ...`
* **General** : `set main.<OPT> = value`
  `Lumi, normalize, SBratio, ...`

- **More options** : see the manual.

Developer-friendly mode

C++ & ROOT skills **required**.

- **Empty analysis template.**

```
./bin/ma5 --expert
```

∗ **Data & sample format** :
Everything well detailed in **manual**.

- **Parton** level **format**.
- **Hadron** level **format**.
- **Reconstructed** level **format**.

Common methods (P = particle object) :
`P.e()`, `P.et()`, `P.gamma()`, `P.theta()`, `P.px()`, ...

Developer-friendly mode

C++ & ROOT skills **required**.

- **Empty analysis template.**

```
./bin/ma5 --expert
```

∗ **Physics services** : C++ `pointer PHYSICS`
Methods & **functions** aiming to facilitate writing of an analysis.

- `PHYSICS->IsHadronic(Pvt)`
- `PHYSICS->IsInvisible(prt)`
- `PHYSICS->IsFinalState(prt)`
- `PHYSICS->ToRestFrame(prt)`
- `PHYSICS->sort(prt_vector,obs)`
- ...

Developer-friendly mode

C++ & ROOT skills **required**.

- **Empty analysis template.**

```
./bin/ma5 --expert
```

∗ **So the user can** :

- **Define** new **sophisticated observables**.
- **Add** more **sophisticated cuts**.
- **Include two- or three-dimensional histograms.**

Conclusions

- A **unique** framework for **different level** of analyses.
  - **Parton** level
  - **Hadron** level
  - **Reconstructed** level
- Will be soon **integrated** to MadGraph5 **framework**.
- **Very user-friendly**
  - Python **interface** with **intuitive commands**.
- **Two ways** of **using** the program
  - **Normal mode**.
    - **No particular programming skills required**.
    - **Synthetic**.
  - **Expert mode**.
    - Requiring programming skills (C++, ROOT).
    - Limited to **user's imagination**.

Perspectives

- **Weighted events** (in particular **NLO negative weights**).
- **Matching** plot.
- **Interface** with FastJet.
- On the way to **fast detector simulation**.
  - **"Fast-sim" meeting** @ CERN (11-12/06).
  - Supply a **flexible fast-simulation of collider detectors**.
  - **Functionnalities** not provided by Delphes or PGS programs : **fakes**, **pile-up**, **trigger structure**, ...

Thanks for your attention.

Want to try it ?

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

Comments, remarks, suggestions :

ma5team@iphc.cnrs.fr