New features of MadAnalysis 5 for analysis design and reinterpretation

Eric Conte[1], **Béranger Dumont**[2,3], **Benjamin Fuks**[4,5] **and Thibaut Schmitt**[5]

(1) Groupe de Recherche de Physique des Hautes Énergies (GRPHE), Université de Haute-Alsace, IUT Colmar, 34 rue du Grillenbreit BP 50568, 68008 Colmar Cedex, France
(2) Center for Theoretical Physics of the Universe, Institute for Basic Science (IBS), Daejeon 305-811, Korea
(3) LPSC, Université Grenoble-Alpes, CNRS/IN2P3, 53 Avenue des Martyrs, F-38026 Grenoble, France
(4) CERN, PH-TH, CH-1211 Geneva 23, Switzerland
(5) Institut Pluridisciplinaire Hubert Curien/Département Recherches Subatomiques, Université de Strasbourg/CNRS-IN2P3, 23 Rue du Loess, F-67037 Strasbourg, France

E-mail: eric.conte@iphc.cnrs.fr, beranger.dumont@lpsc.in2p3.fr, fuks@cern.ch, thibaut.schmitt@iphc.cnrs.fr

Abstract. We present MadAnalysis 5, an analysis package dedicated to phenomenological studies of simulated collisions occurring in high-energy physics experiments. Within this framework, users are invited, through a user-friendly Python interpreter, to implement physics analyses in a very simple manner. A C++ code is then automatically generated, compiled and executed. Very recently, the expert mode of the program has been extended so that analyses with multiple signal/control regions can be handled. Additional observables have also been included, and an interface to several fast detector simulation packages has been developed, one of them being a tune of the Delphes 3 software. As a result, a recasting of existing ATLAS and CMS analyses can be achieved straightforwardly.

1. MadAnalysis 5 in a nutshell

While both LHC experiments are currently pushing limits on particles beyond the Standard Model to higher and higher scales, tremendous progress has been made in the development of Monte Carlo event generators and satellite programs, in particular with respect to precision predictions, new physics implementations and event analysis. The public MadAnalysis 5 package [1, 2] addresses this last aspect and provides a framework for analyzing events generated either at the parton-level, after hadronization or after detector simulation. The program hence allows one to efficiently design and recast LHC analyses. The user can in this way investigate any physics model and determine the LHC sensitivity to its signatures by either conceiving a novel analysis or recasting existing ATLAS and CMS studies. MadAnalysis 5 starts by reading event samples as generated by any Monte Carlo event generator that satisfies community-endorsed output formats. Next, it applies selection cuts and computes differential distributions as requested by the user. In its normal mode of running, the results are represented under the form of histograms and cut-flow tables that are collected within Html and LaTeX reports. In the expert mode of the program, they are presented in text files compliant with the Saf format, an XML-inspired format internally used by MADANALYSIS 5 that has been described in Ref. [2]. Moreover, events can be preprocessed. For instance, the application of a jet-clustering algorithm or the simulation of a detector response can be achieved effortlessly before the execution of the analysis itself, and preprocessed events can be possibly saved in output files.

2. Main concepts

We recall in this section the main concepts of the MADANALYSIS 5 package. We refer to Refs. [1, 2] for more details.

- **Universal**: One can in the same way design new phenomenological analyses and recast existing LHC studies, and a given analysis can be equally applied on parton-level, hadron-level or reconstructed-level events stored under the STDHEP [3], HEPMC [4], LHE [5, 6], LHCO [7] or ROOT [8] format.
- **User friendly**: In the normal mode of running, the user designs her/his analysis by interacting with a PYTHON console through a metalanguage designed to be intuitive. The syntax is moreover easy to handle thanks to tab completion and in-line help.
- **Efficient**: The PYTHON module of the program takes care of exporting the analysis encoded with the MADANALYSIS 5 metalanguage to an optimized C++ program readily to be compiled and executed.
- **Flexible**: MADANALYSIS 5 is shipped with a series of basic (transverse momentum, *etc.*) and more sophisticated (α_T, M_{T2}, *etc.*) built-in observables that can be used in analysis design. Operations on particle four-momenta are also available. For more complicated selections, the user can directly write his/her analysis in C++ (the so-called expert mode of the tool) and implement the desired observables.
- **Interfacing**: MADANALYSIS 5 is interfaced to ZLIB [9], ROOT [8], FASTJET [10] (and FASTJET-CONTRIB) and DELPHES [11]. The installation of these packages can be performed automatically from the PYTHON console. The program also contains the so-called MA5TUNE of DELPHES [12] (see Section 4).

3. Analysis design with the MADANALYSIS 5 metalanguage

From the PYTHON console, the user is invited to describe its selection and configure MADANALYSIS 5 through intuitive and human-understandable commands that inherit from the PYTHON syntax. As an exhaustive review of this metalanguage is beyond the scope of this paper, we focus on a detailed illustrative example and refer to Ref. [1] for more information. We consider a simple analysis defined by the following commands:

```
[0] ma5> import DrellYan*.hep as dy
[1] ma5> import ttbar*.hepmc.gz as tt
[2] ma5> set detector.fastsim.package = fastjet
[3] ma5> set detector.fastsim.algo = kt
[4] ma5> plot MET
[5] ma5> define mu = mu+ mu-
[6] ma5> select (mu) PT > 25
[7] ma5> plot M(mu+ mu-)
[8] ma5> set main.outputfile = output.lhco
[9] ma5> submit
```

The first two commands (lines 0 and 1) enable the import of the event samples to be considered. We recall that MADANALYSIS 5 can read samples provided in the STDHEP, HEPMC, LHE, LHCO or ROOT format, while compressed files can be used if the ZLIB package is available. The next two lines (lines 2 and 3) indicate how to use the interface to the FASTJET program and select the k_T algorithm for jet reconstruction (see also Section 4). The computation of two differential distributions, namely the missing transverse energy and the muon-pair invariant mass spectra, are then requested after enforcing the selection of muons with a transverse momentum of at least 25 GeV (lines 4 to 7). The next command (line 8) requires MadAnalysis 5 to save the reconstructed events in the Lhco format, the Root and Lhe formats being also available. The final command (line 9) requests the program to operate. A C++ code corresponding to the analysis is generated and executed. The results (histograms and selection efficiencies) are collected by a report generator based on Root (other graphical components will be available in the next release of MadAnalysis 5) and further exported into Html and LaTeX reports.

4. Detector fast-simulation packages

The MadAnalysis 5 package offers the option to install, configure and launch the fast simulation of a detector response on simulated events given as input in order to mimic instrumental and experimental effects and increase the realism of the simulation. Three possible choice are available, a very simple (but often sufficient for, *e.g.*, generator-level based phenomenology) and fast simulation based on FastJet, and two more sophisticated and slower simulations based on Delphes 3.

- MadAnalysis 5 is shipped with its own detector simulation based on FastJet. All the jet-clustering algorithms included in FastJet can be applied to hadron-level events, and the user can further switch on several detector effects like reconstruction efficiencies, resolution, misidentification rates, *etc.*
- MadAnalysis 5 is interfaced to Delphes 3, which offers a realistic description of the ATLAS and CMS detectors. Default configuration can be used, but the user is also invited to tune the Delphes setup according to his/her needs. Pile-up effects can be included.
- A modified version of Delphes 3, the so-called Delphes-MA5tune, can be employed. Based on Delphes, it includes the calculation of new observables and optimizes the size of the output file. The main improvements address lepton and photon isolation. Unlike the official package, isolation variables are saved in the output file so that isolation requirements can be implemented at the analysis level instead of at the detector simulation level.

5. Recasting existing LHC analyses with the MadAnalysis 5 expert mode

The *expert mode* of the program offers the possibility to write an analysis directly in C++, using the MadAnalysis 5 framework. It is developer-friendly, and, in addition, a large collection of methods dedicated to common high-energy physics issues are available, as well as various services allowing one, for instance, to produce cut-flow charts and/or histograms. A detailed manual can be found in Ref. [2]. With version 1.1.10 onwards, one can also recast any cut-based ATLAS and CMS analysis possibly containing multiple subanalyses or regions. This allows reinterpretation of experimental results under the approximation (compared to a real experimental setup) of using a fast, simple, detector simulation software. Our framework offers hence a way to work out the implications of the LHC for any new physics model, derive limits in a realistic-enough manner, point out possible loopholes in the current searches and help to design future analyses. It could also be used to improve the content and realism of fast detector simulation packages.

The adopted strategy consists in first applying the Delphes-MA5tune fast detector simulation on signal events stored under the StdHep or HepMc format. Secondly, a reimplementation of a public ATLAS or CMS analysis is executed to derive the number of selected signal events from the input samples. A Python script finally extracts limits by confronting the numbers associated with the more sensitive signal region to the Standard Model expectation provided in the experimental publication. In the case where a user would need a more sophisticated procedure, he/she is welcome to extract directly the limits from

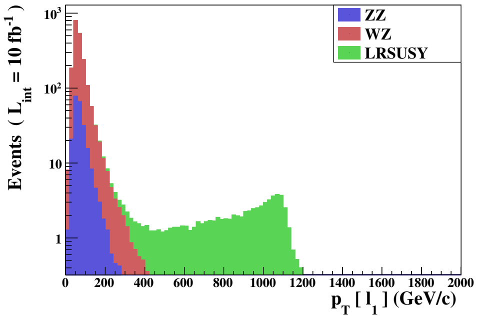

Figure 1. Transverse-momentum distribution of the hardest lepton for the LRSUSY signal and the diboson background.

Cut-flow chart

- How to compare signal (S) and background (B): S/B.
- Associated uncertainty: 1./(B**2)*sqrt(B**2*ES**2+S**2*EB**2).

Cuts	Signal (S)	Background (B)	S vs B
Initial (no cut)	387.033 +/- 0.584	2029.72 +/- 1.44	0.190683 +/- 0.000318
Cut 1	84.95 +/- 8.14	2029.72 +/- 1.44	0.04185 +/- 0.00401
Cut 2	81.56 +/- 8.02	26.09 +/- 5.08	3.126 +/- 0.681
Cut 3	76.19 +/- 7.82	19.45 +/- 4.39	3.92 +/- 0.97

Signal and Background comparison

Figure 2. Snippet of the MadAnalysis 5 report with the cut-flow chart associated with the selection described in the text. The figure of merit consists of the ratio of the number of signal to the number of background events.

the MadAnalysis output independently. Since the validation of reimplemented analyses can sometimes be complicated (see, *e.g.*, Refs. [12, 13]), validated codes can be shared through our public analysis database [12] and submitted to Inspire [14], that additionally assigns each submission a DOI [15].

6. Selection of results

In order to demonstrate the performances of the program, we focus on two examples. A first analysis is related to the search for a multileptonic signature of left-right symmetric supersymmetry (LRSUSY) in simulated LHC collisions at a center-of-mass energy of 13 TeV [16]. A second example is devoted to the recast of the CMS-SUS-13-012 analysis focusing on squark and gluino searches in multijet events in collisions at a center-of-mass energy of 8 TeV [17, 18].

6.1. Prospective phenomenology in the left-right symmetric supersymmetric context

LRSUSY predicts additional gaugino and higgsino states that can potentially lead to signals in multileptonic events to be produced at the LHC. In the present example, we perform a simple selection in order to extract the LRSUSY signal from the Standard Model WZ and ZZ background. In the MadAnalysis 5 metalanguage, the analysis reads

```
ma5> define l = e+ e- mu+ mu-
ma5> select N(l) >= 3
ma5> plot PT(l[1]) 200 0 2000 [ logY ]
ma5> select PT(l) > 200
ma5> select MET > 50
```

Events containing at least three charged leptons are selected. The transverse momentum (p_T) distribution of the hardest lepton is then calculated, the results being shown in Figure 1. The signal exhibiting a harder spectrum, the p_T of the hardest lepton is constrained to be larger than 200 GeV, while at least 50 GeV of missing transverse energy (assumed to be carried by the lightest supersymmetric state that is stable and escapes detection) is demanded. The corresponding cut-flow table is presented on Figure 2, that shows that the selection yields a figure of merit of about 4σ.

6.2. Example of recast analysis

A recasting of the CMS-SUS-12-012 analysis [17] has been recently performed, the validated MadAnalysis 5 code being available in Ref. [18]. This validation has been achieved by

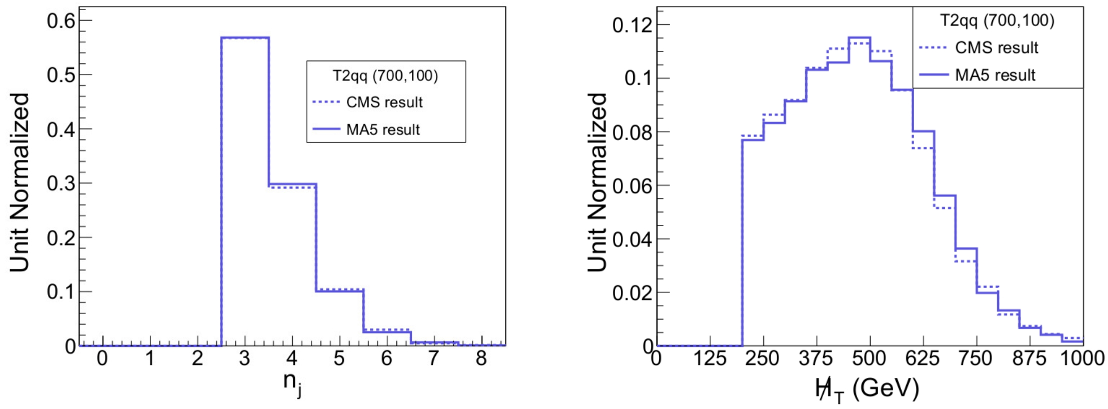

Figure 3. Comparison between CMS and MA5 for the distribution in the jet multiplicity (left) and in the $\not{H}_T$ variable (right) for a specific signal scenario and after the baseline selection described in Ref. [17].

comparing CMS and MadAnalysis 5 (MA5) results, both at the cut-flow level and for various observables. Two examples are shown in Figure 3, where we present distributions for the jet multiplicity and the missing transverse hadronic energy $\not{H}_T$ in the case of the *T2qq* simplified model [19]. We confront MA5 predictions to CMS official results and observe a good agreement.

7. Summary

The public package MadAnalysis 5 offers a simple and efficient way to design and recast LHC analyses. The user can either benefit from its Python console and dedicated metalanguage to implement his/her analysis, or make use of the expert mode of the program which provides a flexible developer-friendly C++ environment. We have also discussed detector simulation within the MadAnalysis 5 framework, presenting the Delphes-MA5tune simulation of a detector response, and introduced our recently initiated database of reimplementations of LHC analyses.

Acknowledgments

We thank the organizers for setting up this nice conference. This work has been partially supported by the French ANR projects 12-JS05-002-01 BATS@LHC and 12-BS05-0006 DMAstroLHC, by the Theory-LHC France initiative of the CNRS/IN2P3 and by the *Investissement d'avenir, Labex ENIGMASS*.

References

[1] Conte E, Fuks B and Serret G 2013 *Comput.Phys.Commun.* **184** 222–256 (*Preprint* `1206.1599`)
[2] Conte E, Dumont B, Fuks B and Wymant C 2014 (*Preprint* `1405.3982`)
[3] URL `http://cepa.fnal.gov/psm/stdhep/c++/`
[4] Dobbs M and Hansen J B 2001 *Comput.Phys.Commun.* **134** 41–46
[5] Boos E, Dobbs M, Giele W, Hinchliffe I, Huston J *et al.*
[6] Alwall J, Ballestrero A, Bartalini P, Belov S, Boos E *et al.* 2007 *Comput.Phys.Commun.* **176** 300–304 (*Preprint* `hep-ph/0609017`)
[7] URL `http://www.jthaler.net/olympicswiki/`
[8] Brun R and Rademakers F 1997 *Nucl.Instrum.Meth.* **A389** 81–86
[9] http://www.zlib.net
[10] Cacciari M, Salam G P and Soyez G 2012 *Eur.Phys.J.* **C72** 1896 (*Preprint* `1111.6097`)
[11] de Favereau J *et al.* (DELPHES 3) 2014 *JHEP* **1402** 057 (*Preprint* `1307.6346`)
[12] Dumont B, Fuks B, Kraml S, Bein S, Chalons G *et al.* 2014 (*Preprint* `1407.3278`)
[13] Kraml S, Allanach B, Mangano M, Prosper H, Sekmen S *et al.* 2012 *Eur.Phys.J.* **C72** 1976 (*Preprint* `1203.2489`)

[14] http://inspirehep.net, see also http://www.projecthepinspire.net
[15] http://www.datacite.org
[16] Alloul A, Frank M, Fuks B and Rausch de Traubenberg M 2013 *JHEP* **1310** 033 (*Preprint* 1307.5073)
[17] Chatrchyan S *et al.* (CMS Collaboration) 2014 *JHEP* **1406** 055 (*Preprint* 1402.4770)
[18] Bein S and Sengupta D 10.7484/INSPIREHEP.DATA.83GG.U5BW
[19] Chatrchyan S *et al.* (CMS Collaboration) 2013 *Phys.Rev.* **D88** 052017 (*Preprint* 1301.2175)