

# MADANALYSIS 5: status and new developments

**Eric Conte<sup>1</sup> and Benjamin Fuks<sup>2,3</sup>**

<sup>(1)</sup> 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

<sup>(2)</sup> Theory Division, Physics Department, CERN, CH-1211 Geneva 23, Switzerland

<sup>(3)</sup> 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](mailto:eric.conte@iphc.cnrs.fr), [benjamin.fuks@iphc.cnrs.fr](mailto:benjamin.fuks@iphc.cnrs.fr)

**Abstract.** MADANALYSIS 5 is a new PYTHON/C++ package facilitating phenomenological analyses that can be performed in the framework of Monte Carlo simulations of collisions to be produced in high-energy physics experiments. It allows one, by means of a user-friendly interpreter, to perform professional physics analyses in a very simple way. Starting from event samples as generated by any Monte Carlo event generator, large classes of selections can be implemented through intuitive commands, many standard kinematical distributions can be automatically represented by histograms and all results are eventually gathered into detailed HTML and LATEX reports. In this work, we briefly report on the latest developments of the code, focusing on the interface to the FASTJET program dedicated to jet reconstruction.

## 1. Introduction

The TeV energy regime is currently being explored at the Large Hadron Collider, at CERN, and searches for new phenomena are a central topic of the associated experimental program. However, as a consequence of the absence of hints for new physics, the states predicted by any theory beyond the Standard Model (BSM) are bound to lie at higher and higher scales. While experimental analyses mostly focus on popular BSM theories and on simplified extensions of the Standard Model (SM), (re)interpretations of the results in any specific theoretical framework and phenomenological investigations of new types of signatures play key roles in the improvement of our understanding of the TeV scale. To this aim, matrix-element-based Monte Carlo event generators are generally employed, the results being matched to parton showering, hadronization and detector simulation algorithms. Phenomenological analyses can then be performed at several levels, from a purely partonic description of the simulated event final states to a more complex description including hadronization and possibly a fast simulation of the detector response. The MADANALYSIS 5 package [1] offers a unique framework to perform such analyses, in an easy and efficient way. While about to be fully embedded within the Monte Carlo event generators MADGRAPH 5 [2] and aMC@NLO [3], MADANALYSIS 5 can also be used as a standalone program and outside the MADGRAPH environment.

In this paper, we describe some of the novel features of the version 1.1.8 of the code and provide an update of the original user guide. After a brief overview of MADANALYSIS 5 in Section 2, we illustrate how to use it to design a phenomenological analysis in Section 3. Section 4 is dedicated to a recent module added the code, related to jet reconstruction and allowing one to verify the merging procedures usually employed to combine event samples describing the same

physics process but with different final state jet multiplicities. Our conclusions are presented in Section 5.

## 2. Overview

### 2.1. MADANALYSIS 5 *in a nutshell*

MADANALYSIS 5 is a framework designed to facilitate the implementation of phenomenological studies based on Monte Carlo simulations of collisions describing both new physics signal and Standard Model background processes. It allows one, on the one hand, to investigate the characteristics of a particular process, and, on the other hand, to carry out an event selection aiming to extract a given signal from the background. It supports event files encoded under the STDHEP [4], HEPMC [5], LHE [6, 7], LHCO [8] and DELPHES ROOT [9] formats.

Properties of the event final states can be probed by means of a small set of intuitive PYTHON commands (see below) acting on particles modeled by user-defined labels and several observables. They can then be represented by histograms whose layout can be tuned according to the preferences of the user. In the same way, event selection criteria can be implemented via dedicated commands and equally applied to a full event (*i.e.*, to select or reject it under a given condition) or to a particle (*i.e.*, to include it or not in the analysis under a specific condition). MADANALYSIS 5 can also further compute signal over background ratios whose definition can be adjusted by the user.

Flexible, MADANALYSIS 5 also offers sophisticated functionalities allowing one to, *e.g.*, combine particle four-momenta (sums, differences, *etc.*) before the computation of any observable, use the event history (mother-to-daughter relations linking a particle to its decay products) or identify a given particle according to its hardness. All the results are collected into reports available under the HTML, POSTSCRIPT (associated with LATEX) or PDF (associated with PDFLATEX) formats. Interfaced to the FASTJET package [10], MADANALYSIS 5 also includes several jet-clustering algorithms that can be used for object reconstruction, including or not a simplified detector simulation and several (mis)identification efficiencies.

### 2.2. Python console

Once downloaded from the Internet (see Ref. [11]) and unpacked, the MADANALYSIS 5 package does not require any compilation and can be started by issuing

```
bin/ma5
```

from the directory where it has been stored. By default, MADANALYSIS 5 is configured for the analysis of events at the parton level, although events at the hadron or at the reconstructed level can be analyzed by launching the program with the `-H` and `-R` arguments, respectively.

The user interacts with MADANALYSIS 5 through a console implemented in the PYTHON language that invites him/her to define his/her analysis through a set of commands that can also possibly be collected into a script. The MADANALYSIS 5 metalanguage is designed to be simple and concise, as illustrated in the examples presented in the next sections, and the user can be helped in two ways. First, in-line help can be employed by issuing

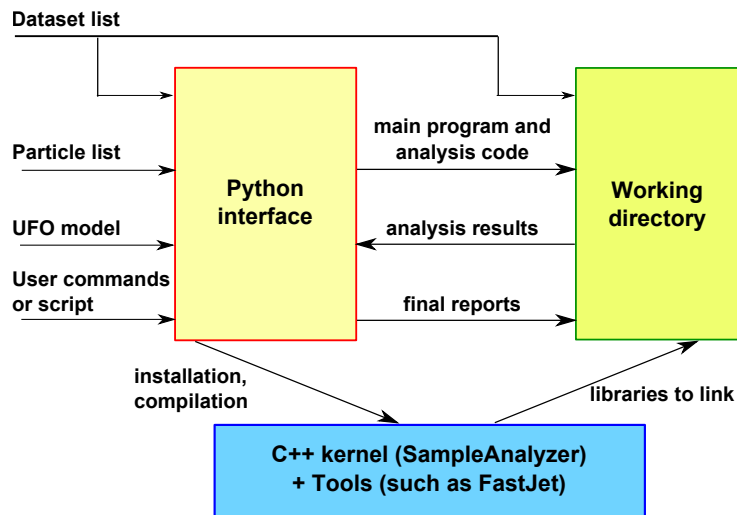
```
ma5> help
```

in the console, which displays the list of available commands. Second, auto-completion has been implemented and can be used via the `tab` key.

### 2.3. Structure of the program

Particle labels can be either directly defined in the PYTHON console or imported from a Universal FeynRules Output (UFO) model containing names and labels for all the particles of a given model [12]. Additionally, the list of event samples to analyze has to be specified.

MADANALYSIS 5 then takes care of automatically exporting the analysis under the form of a C++ code to be linked to a library called SAMPLEANALYZER, shipped with the package, and to any other optional tools such as FASTJET or DELPHES according to the needs of the user. The C++ program is further compiled and executed, and the results are encapsulated into an XML structure stored in a so-called SAF (standing for SAMPLEANALYZER Format) file. The PYTHON console is eventually automatically taking care of rendering the information human-readable, under the form of POSTSCRIPT, PDF and HTML reports including the defined histograms and the computed selection efficiencies. The full code structure is summarized on Figure 1.



**Figure 1.** Logical structure of MADANALYSIS 5

On different footings, users can also directly implement their analysis in the C++ language, bypassing the PYTHON console and using all built-in functions available within the SAMPLEANALYZER library. This way of using MADANALYSIS 5 is called the *expert mode* of running, in contrast to the *normal mode* using the PYTHON console, and allows one to implement any analysis which a user can dream of, the only limitation being his/her imagination and programming skills.

#### 2.4. Requirements

When started, MADANALYSIS 5 firstly checks the presence on the system of required dependencies and secondly verifies which of optional libraries are available. The list of the mandatory packages to run properly the program is the following :

- PYTHON 2.6 or a more recent version (although not a 3.x version),
- the GNU GCC compiler together with the associated Standard Template Libraries,
- the ROOT package version 5.27 or more recent [13].

Some specific functionalities of MADANALYSIS 5 may require the use of other external packages and are thus deactivated if necessary. Those consist of

- the LATEX and PDFLATEX compilers (analysis reports in the POSTSCRIPT and PDF formats, respectively),
- the ZLIB package (reading and writing compressed files),
- the FASTJET package (to switch on the module dedicated to jet reconstruction, see Section 4),

- the DELPHES package (reading ROOT files as outputted by DELPHES 3).
- a few representative event samples useful to learn how to use MADANALYSIS 5.

With the exception of the LATEX and PDFLATEX compilers (that must be installed independently by the user), each optional package that MADANALYSIS 5 may require can be installed automatically by using the command `install` from the PYTHON console. For instance, typing in the commands

```
ma5> install gzip
ma5> install delphes
ma5> install fastjet
ma5> install samples
```

leads to the installation of the ZLIB libraries, the FASTJET and DELPHES 3 programs and some Monte Carlo event samples, respectively.

### 3. Designing an analysis with MadAnalysis 5

In this section, we briefly describe the way to perform a simple analysis with MADANALYSIS 5. More advanced functionalities as well as the expert mode of the code are not discussed and we refer to the manual for more details [1].

#### 3.1. Basic concepts

First, the event samples that must be read and analyzed are specified through the command `import`. By default, all imported samples are collected within a single dataset called `defaultset`, although the user can define as many datasets as necessary by using the command `import` together with the keyword `as`. In general, it is recommended to have one single dataset for each physics process. The wildcard characters `*` and `?` can also be used to import several files simultaneously.

Second, in order to treat the different particle species all along the analysis, MADANALYSIS 5 employs labels referring either to one given type of particle or to several types of particles (the term of multiparticle is used in this case). When MADANALYSIS 5 starts, collections of standard labels for the SM and Minimal Supersymmetric Standard Model particles are loaded. The user has then the possibility to define his/her own labels by the command `define`.

Third, histograms representing either properties of the full event (*e.g.*, the transverse missing energy or the transverse hadronic energy) or properties of a given particle assumed to be part of the event final state content (*e.g.*, its transverse momentum or its multiplicity) can be created by using the command `plot`, followed by the name of the relevant observable (see the manual for more details on the syntax [1]).

Finally, the list of observables that can be represented by histograms can also be used to apply selection requirements either on the events themselves, or to select the final state objects to be considered in the analysis. To this aim, the user can employ two complementary commands, `reject` and `select`, preceding a criterion.

#### 3.2. Illustrative example

We provide, for the sake of the example, a toy analysis based on the events samples provided by installing the optional package SAMPLES. The analysis commands are given by

```
(1) ma5> import samples/zz.lhe.gz as mySignal
(2) ma5> import samples/ttbar_sl_?.lhe.gz as myBkg
(3) ma5> set myBkg.type = background
(4) ma5> define mu = mu+ mu-
(5) ma5> plot MET
```

```

(6) ma5> plot PT(mu) [logY]
(7) ma5> reject (mu) PT < 20
(8) ma5> select 80 < M (mu+ mu-) < 100
(9) ma5> plot M(mu+ mu-)

```

On the first line, a dataset called `mySignal` is created and linked to the event file `zz.lhe.gz`. On the second line, one collects the two samples `ttbar_sl.1.lhe.gz` and `ttbar_sl.2.lhe.gz` into a unique dataset named `myBkg`, which is further defined as containing background events on the third line. In contrast, the type of the `mySignal` dataset is left to its default value, *i.e.*, the related events being considered as signal events.

A new multiparticle label is defined on the fourth line by making use of the command `define` and is associated with both muons and antimuons. This label is then used in the rest of the analysis to define selections and histograms. Two histograms are hence defined on the fifth and sixth lines of the analysis, the first one representing the missing transverse energy distribution extracted from each dataset and the second one describing the transverse momentum spectrum obtained after considering all muons and antimuons of each event. By the commands of the seventh and eighth lines, we demand to consider, in the rest of the analysis, all (anti)muons with a transverse momentum smaller than 20 GeV and to keep events containing at least one muon-antimuon pair with an invariant mass included in the range [80, 100] GeV. A final histogram is created on the last line of the analysis and allows one to study the effect of the previous selections on the invariant mass of all muon-antimuon pairs that can be formed in each event.

### 3.3. Submitting and executing the analysis

The user can execute his/her analysis by typing in, in the PYTHON console, the command

```
submit
```

Behind the scenes, a C++ code corresponding to the implemented analysis is generated, compiled and linked to the `SAMPLEANALYZER` library. The executable is then run over all datasets and the results are displayed via the different reports. The HTML report can always be open directly from the PYTHON interface by means of the command

```
open
```

A representative histogram that can be found in the report associated with the analysis above is shown in Figure 2.

## 4. Interface to FastJet package

In order to enable the functionalities of `MADANALYSIS 5` related to jet clustering, the `FASTJET` package has to be installed on the system of the user, either externally to `MADANALYSIS 5` or within the program (see Section 2.4).

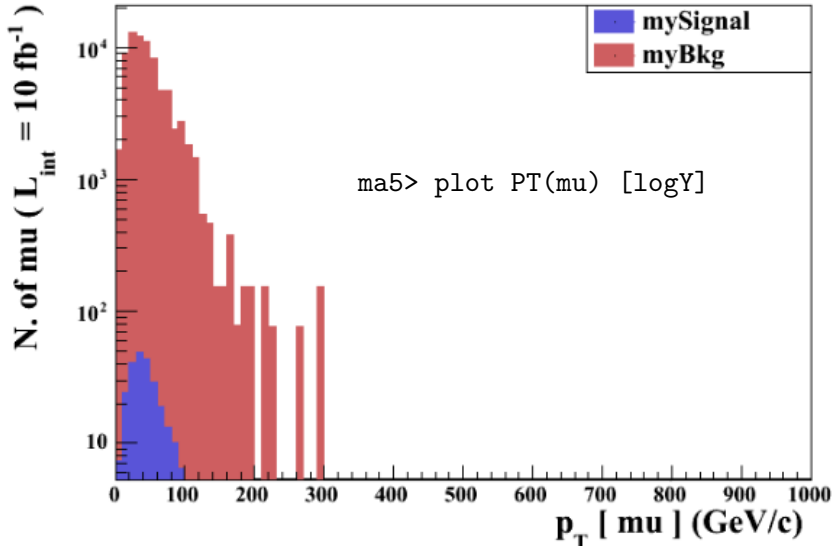
### 4.1. Reconstructing jets in hadron-level events

Starting from an event file at the hadron level (in general encoded in the `STDHEP` or `HEPMC` file format), `MADANALYSIS 5` offers the user a straightforward way to gather the hundreds of hadrons usually present in the event final states into jets by means of any of the algorithms included in the `FASTJET` library. After this procedure, the events are described in terms of jets, electrons, muons, taus, photons and missing transverse energy. While leptonically-decaying taus are accounted for in terms of their decay products (electrons or muons and missing energy), only hadronically-decaying taus are considered as a distinct physics objects (and defined as taus).

In order to enable jet clustering within `MADANALYSIS 5` (assumed to be started in the reconstructed-level mode), it is sufficient to specify the algorithm to be used as the value of the `main.clustering.algorithm` variable. The allowed choices are `kt` (longitudinally-invariant

Dataset	Integral	Entries / events	Mean	RMS	%Underflow	%Overflow
mysignal	256	2.02	48.3857	32.11	0.0	0.0
mybkg	75706	0.493	49.6728	34.55	0.0	0.0

Statistics table



**Figure 2.** Histogram representing the transverse momentum spectra of all (anti)muons for the two datasets denoted by `mySignal` and `myBkg`. The command associated with the histogram generation is superimposed to the figure.

$k_T$  algorithm [14, 15]), `cambridge` (the Cambridge/Aachen algorithm [16, 17]), `antikt` (the anti- $k_T$  algorithm [18]), `genkt` (the generalized  $k_T$  algorithm [10]), `siscone` (the SIScone algorithm [19]), `cdfmidpoint` (the CDF MIDPOINT algorithm [20]), `cdfalgo` (the CDF JETCLU algorithm [21]) and `gridjet` (the GRIDJET algorithm [10]). For instance, employing the anti- $k_T$  algorithm can be enforced by typing

```
ma5> set main.clustering.algo = antikt
```

As soon as a given jet algorithm is selected, new options depending on the algorithm parameters become available. In the (illustrative) example of the anti- $k_T$  algorithm, the variables `main.clustering.radius` and `main.clustering.ptmin`, respectively representing the jet-radius parameter and the jet transverse-momentum threshold, can be set by the user.

We now turn to the description of all other options coming with the jet clustering module of MADANALYSIS 5 and that are related to the identification of the physics objects. After the application of the algorithm, each particle is uniquely identified and cannot contribute to several collections of physics objects. A remark on leptons is however in order here. By default, the jet algorithm distinguishes leptons issued from the hard process from those originating from hadron decays. While the first category of leptons are reconstructed as leptons and included in the relevant collection, leptons of the second category are feeding the jet algorithm and therefore included as parts of the reconstructed jets. This default behavior can be modified by setting the variable `main.clustering.exclusive` to `false`. This forces the jet algorithm to include all visible particles for jet reconstruction, so that a given lepton can consequently be part of the relevant lepton collection and also contribute to a specific jet.

MADANALYSIS also includes a simple algorithm dedicated to the  $b$ -tagging of the reconstructed jets, so that a jet is identified as originating from a  $b$ -quark when it can be matched to a  $b$  (anti)quark, *i.e.*, when it lies within a cone of radius  $R$

around one of the parton-level  $b$  quarks. The size of the cone ( $R$ ) is specified as the value of the `main.clustering.bjet_id.matching_dr` variable, while setting the parameter `main.clustering.bjet_id.exclusive` to `true` indicates the code that one cannot match several jets to one given  $b$  quark. By default, the  $b$ -tagging algorithm is perfect, although (mis)identification efficiencies can also be specified. For example, typing the three commands

```
ma5> set main.clustering.bjet_id. efficiency = 0.60
ma5> set main.clustering.bjet_id.misid_cjet = 0.10
ma5> set main.clustering.bjet_id.misid_ljet = 0.05
```

first sets the  $b$ -tagging efficiency to 60%. Next, the two other commands teach the  $b$ -tagging algorithm that  $c$ -quark-initiated jets and other lighter jets are tagged as  $b$ -jets in respectively 10% and 5% of the cases. If more advanced  $b$ -tagging methods are required by the user, such as implementing a tagging algorithm depending on the jet transverse momentum, he/she is recommended to use instead a program simulating the entire detector response such as DELPHES [9]. Furthermore, similar options are available for hadronically-decaying taus.

#### 4.2. Storing jet-clustered events

Events obtained after jet clustering can be stored in a file whose name is specified by the user as the value of the variable `main.outputfile`. For instance,

```
ma5> set main.outputfile = "myOutput.lhe"
```

allows one to save the reconstructed events in a file named `myOutput.lhe`. The file format is directly specified by the file extension, the suitable choices being either `lhe` (LHE format) or `lhco` (LHCO format). Compressed files can also be used if the optional ZLIB libraries are installed, the related file extensions being in this case `lhe.gz` and `lhco.gz`. The files created by MADANALYSIS 5 are however not strictly following the LHE and LHCO conventions originally defined in Refs. [6, 7, 8].

- Some variables associated with an LHCO event, such as the calorimeter energy deposits, are ignored by MADANALYSIS 5.
- In the case of LHE events, initial state particles are recorded with a status code set to  $-1$ , final state particles with a status code equal to 1 and intermediate particles involved in the hard process with a status code set to 3. Other intermediate particles, such as those generated by parton showering, are omitted from the output file. Each reconstructed physics object, *i.e.*, each of the light jets,  $b$ -tagged jets, electrons, muons, hadronically-decaying taus, photons as well as the missing transverse energy, is identified with a specific code inspired by the Particle Data Group (PDG) particle numbering scheme, chosen to be 21,  $\pm 5$ ,  $\pm 11$ ,  $\pm 13$ ,  $\pm 15$ , 22 and 12, respectively. Finally, the link between an intermediate particle of the hard process and the induced final state objects is saved by means of the associated mother-to-daughter relations as they can be encoded in the LHE format.

We refer to the header of the outputted files for more details.

#### 4.3. Validating multiparton matrix-element merging

In order to achieve an accurate description of the jet properties of a given event sample, two complementary approaches are usually combined. Matrix elements are employed for the description of the hard jets and parton showering for the one of the soft and collinear jets. Consequently, a double counting of the QCD emission is possible as radiation can now emerge from two sources. Several merging procedures have been subsequently developed in the last decades. MADANALYSIS 5 offers a way, based on its interface with FASTJET, to validate the resulting *merged* samples. This relies on a check of the smoothness of the differential jet rate

(DJR) distributions, the DJR observables strongly depending on the merging parameters and procedure as they describe the distribution of the scale at which the jet multiplicity of an event final state switches from  $N$  to  $N + 1$ . The DJR histograms can be included in the MADANALYSIS 5 reports by typing

```
ma5> set main.merging.check = true
```

## 5. Summary

MADANALYSIS 5 is a unique, flexible and user-friendly framework allowing for professional analyses of simulations of collisions to be produced in high-energy physics experiments. It can be used equivalently for the investigation of parton-level, hadron-level or reconstructed events and is therefore able to read many input formats. The program comes with two modes of running, the so-called normal and expert modes. In the first case, the user is invited to define his/her analysis by typing command lines in a PYTHON console. In the second case, he/she can directly implement his/her analysis in C++, using a large class of predefined functions included in the core of the package.

In this paper, we have emphasized new functionalities related to an interface with the FASTJET program and which include the possibility of using different jet algorithms to cluster hadrons into jets, when analyzing hadron-level events. In addition, extra features have been developed and allow to mimic the response of an ideal detector and to output the reconstructed events into a simplified LHCO or LHE format. Moreover, MADANALYSIS 5 is now also capable to generate DJR distributions allowing for the validation of the jet merging procedure traditionally employed in state-of-the-art phenomenological (and also in some experimental) analyses.

## Acknowledgments

The authors are grateful to the ACAT organizers for the very nice conference. They also thank A. Alloul for his numerous checks of the code and for his help during the development and the validation of the later releases. This work has been supported by the French ANR 12 JS05 002 01 BATS@LHC and by the Theory-LHC France initiative of the CNRS/IN2P3.

## References

- [1] Conte E, Fuks B and Serret G 2013 *Comput.Phys.Commun.* **184** 222–256 (*Preprint* 1206.1599)
- [2] Alwall J, Herquet M, Maltoni F, Mattelaer O and Stelzer T 2011 *JHEP* **1106** 128 (*Preprint* 1106.0522)
- [3] Alwall J, Frederix R, Frixione S, Hirschi V, Maltoni F, Mattelaer O, Torrielli P and Zaro M *in preparation*
- [4] <http://cepa.fnal.gov/psm/stdhep/c++/>
- [5] Dobbs M and Hansen J B 2001 *Computer Physics Communications* **134** 41 – 46
- [6] Boos E, Dobbs M, Giele W, Hinchliffe I, Huston J *et al.* 2001 (*Preprint* hep-ph/0109068)
- [7] Alwall J, Ballestrero A, Bartalini P, Belov S, Boos E *et al.* 2007 *Comput.Phys.Commun.* **176** 300–304 (*Preprint* hep-ph/0609017)
- [8] <http://www.jthaler.net/olympicswiki/>
- [9] de Favereau J, Delaere C, Demin P, Giammanco A, Lemaitre V *et al.* 2013 (*Preprint* 1307.6346)
- [10] Cacciari M, Salam G P and Soyez G 2012 *Eur.Phys.J. C* **72** 1896 (*Preprint* 1111.6097)
- [11] <https://launchpad.net/madanalysis5>
- [12] Degrande C, Duhr C, Fuks B, Grellscheid D, Mattelaer O *et al.* 2012 *Comput.Phys.Commun.* **183** 1201–1214 (*Preprint* 1108.2040)
- [13] Brun R and Rademakers F 1997 *Nucl.Instrum.Meth. A* **389** 81–86
- [14] Catani S, Dokshitzer Y L, Seymour M and Webber B 1993 *Nucl.Phys. B* **406** 187–224
- [15] Ellis S D and Soper D E 1993 *Phys. Rev. D* **48**(7) 3160–3166
- [16] Dokshitzer Y, Leder G, Moretti S and Webber B 1997 *Journal of High Energy Physics* **1997** 001
- [17] Wobisch M and Wengler T 1998 (*Preprint* hep-ph/9907280)
- [18] Cacciari M, Salam G P and Soyez G 2008 *JHEP* **0804** 063 (*Preprint* 0802.1189)
- [19] Salam G P and Soyez G 2007 *JHEP* **0705** 086 (*Preprint* 0704.0292)
- [20] Blazey G C, Dittmann J R, Ellis S D, Elvira V D, Frame K *et al.* 2000 47–77 (*Preprint* hep-ex/0005012)
- [21] Abe F *et al.* (CDF Collaboration) 1992 *Phys.Rev. D* **45** 1448–1458