Eur. Phys. J. C (2014) 74:2714
DOI 10.1140/epjc/s10052-014-2714-9

THE EUROPEAN PHYSICAL JOURNAL C

Special Article - Tools for Experiment and Theory

MCPLOTS: a particle physics resource based on volunteer computing

A. Karneyeu[1], **L. Mijovic**[2,3], **S. Prestel**[2,4], **P. Z. Skands**[5,a]

[1] Institute for Nuclear Research, Moscow, Russia
[2] Deutsches Elektronen-Synchrotron, DESY, 22603 Hamburg, Germany
[3] Irfu/SPP, CEA-Saclay, bat 141, 91191 Gif-sur-Yvette Cedex, France
[4] Department of Astronomy and Theoretical Physics, Lund University, Lund, Sweden
[5] European Organization for Nuclear Research, CERN, 1211 Geneva 23, Switzerland

Received: 1 July 2013 / Accepted: 3 January 2014 / Published online: 5 February 2014

Abstract The **mcplots.cern.ch** web site (MCPLOTS) provides a simple online repository of plots made with high-energy-physics event generators, comparing them to a wide variety of experimental data. The repository is based on the HEPDATA online database of experimental results and on the RIVET Monte Carlo analysis tool. The repository is continually updated and relies on computing power donated by volunteers, via the LHC@HOME 2.0 platform.

1 Introduction

Computer simulations of high-energy interactions are used to provide an explicit theoretical reference for a wide range of particle-physics measurements. In particular, Monte Carlo (MC) event generators [1–3] enable a comparison between theory and experimental data down to the level of individual particles. An exact calculation taking all relevant dynamics into account would require a solution to infinite-order perturbation theory coupled to non-perturbative QCD—a long-standing and unsolved problem. In the absence of such a solution, MC generators apply a divide-and-conquer strategy, factorizing the problem into many simpler pieces, and treating each one at a level of approximation dictated by our understanding of the corresponding parts of the underlying fundamental theory.

A central question, when a disagreement is found between simulation and data, is thus whether the discrepancy is within the intrinsic uncertainty allowed by the inaccuracy of the calculation, or not. This accuracy depends both on the sophistication of the simulation itself, driven by the development and implementation of new theoretical ideas, but it also depends crucially on the available constraints on the free parameters of the model. Using existing data to constrain these is referred to as "tuning". Useful discussions of tuning can be found, e.g., in [1,2,4–10].

Typically, experimental studies include comparisons of specific models and tunes to the data in their publications. Such comparisons are useful both as immediate tests of commonly used models, and to illustrate the current amount of theoretical uncertainty surrounding a particular distribution. They also provide a set of well-defined theoretical reference curves that can be useful as benchmarks for future studies. However, many physics distributions, in particular those that are infrared (IR) sensitive[1] often represent a complicated cocktail of physics effects. The conclusions that can be drawn from comparisons on individual distributions are therefore limited. They also gradually become outdated, as new models and tunes supersede the old ones. In the long term, the real aim is not to study one distribution in detail, for which a simple fit would in principle suffice, but to study the degree of simultaneous agreement or disagreement over many, mutually complementary, distributions. This is also a prerequisite to extend the concept of tuning to more rigorous consistency tests of the underlying physics model, for instance as proposed in [8].

The effort involved in making simultaneous comparisons to large numbers of experimental distributions, and to keep those up to date, previously meant that this kind of exercise was restricted to a small set of people, mostly Monte Carlo authors and dedicated experimental tuning groups. The aim with the **mcplots.cern.ch** (MCPLOTS) web site is to provide a

[a] e-mail: peter.skands@cern.ch

[1] IR sensitive observables change value when adding an infinitely soft particle or when splitting an existing particle into two collinear ones. Such variables have larger sensitivity to non-perturbative physics than IR safe ones, see, e.g., [2,11]. Note also that we here use the word "IR" as a catchall for both the soft and collinear limits.

simple browsable repository of such comparisons so that anyone can quickly get an idea of how well a particular model describes various data sets.[2] Simultaneously, we also aim to make all generated data, parameter cards, source codes, experimental references, etc, freely and directly available in as useful forms as possible, for anyone who wishes to reproduce, re-plot, or otherwise make use of the results and tools that we have developed.

The MCPLOTS web site is now at a mature and stable stage. It has been online since Dec 2010 and is nearing a trillion generated events in total (900 billion as of June 2013). This paper is intended to give an overview of what is available on the site, and how to use it. In particular, Sect. 2 contains a brief "user guide" for the site, explaining its features and contents in simple terms, how to navigate through the site, and how to extract plots and information about how they were made from it. As a reference for further additions and updates, and for the benefit of future developers, Sects. 3–7 then describe the more concrete details of the technical structure and implementation of the site, which the ordinary user would not need to be familiar with.

Section 3 describes the architecture of the site, and the thinking behind it. It currently relies on the following basic prerequisites,

- The HEPDATA database [13] of experimental results.
- The RIVET Monte Carlo analysis tool [14] which contains a large code library for comparing the output of MC generators to distributions from HEPDATA. RIVET in turn relies on the HEPMC universal event-record format [15], on the FASTJET package for jet clustering [16,17], and on the LHAPDF library for parton densities [18,19].
- Monte Carlo event generators. Currently implemented generators include ALPGEN [20], EPOS [21] HERWIG++ [22], PHOJET [23], PYTHIA 6 [24], PYTHIA 8 [25], SHERPA [26], and VINCIA [27]. Some of these in turn use the Les Houches Event File (LHEF) format [28,29] to pass parton-level information back and forth.
- The LHC@HOME 2.0 framework for volunteer cloud computing [30–33]. LHC@HOME 2.0 in turn relies on CERNVM [34,35] (a Virtual-Machine computing environment based on Scientific Linux), on the COPILOT job submission system [36,35], and on the BOINC middleware for volunteer computing [37,38].

The basic procedure to include a new measurement on MCPLOTS is, first, to provide the relevant experimental data points to HEPDATA, second to provide a RIVET routine for the corresponding analysis, and lastly to provide a very small additional amount of information to MCPLOTS, essentially specifying the placement of the observable in the MCPLOTS menus and summarizing the cuts applied in a LaTeX string, as e.g. exemplified by the already existing analyses on the site. This is described in more detail in Sect. 4.

To update MCPLOTS with a new version of an existing generator, the first step is to check whether it is already available in the standard CERN Generator Services (GENSER) repository [39], and if not announce the new version to the GENSER team. The MCPLOTS steering scripts should then be updated to run jobs for the new version, as described in Sect. 5.

To add a new generator to MCPLOTS, the first step is to check that it can run within CERNVM. CERNVM provides a standardized Scientific-Linux environment that should be appropriate for most high-energy physics (HEP) applications, including several commonly used auxiliary packages such as the GNU Scientific Library (GSL), the C++ BOOST libraries, and many others. A standalone version of CERNVM can be downloaded from [34] for testing purposes. To the extent that dependencies require additional packages to be installed, these should be communicated to the MCPLOTS and CERNVM development teams. The code should then be provided to the GENSER team for inclusion in the standard CERN generator repository. The complete procedure is described in more detail in Sect. 6.

The main benefit of using CERNVM is that this has enabled MCPLOTS to draw on significant computing resources made available by volunteers, via the BOINC and LHC@HOME 2.0 projects. Through the intermediary of CERNVM, generator authors can concentrate on developing code that is compatible with the Scientific Linux operating system, a fairly standard environment in our field. This code can then be run on essentially any user platform by encapsulating it within CERNVM and distributing it via the BOINC middleware. (The latter also enabled us to access a large existing BOINC volunteer-computing community.) The resulting "TEST4THEORY" project [33,40] was the first application developed for the LHC@HOME 2.0 framework, see [30,33], and it represents the world's first virtualization-based volunteer cloud. A brief summary of it is given in Sect. 7.

2 User guide

In this section, we describe the graphical interface on the MCPLOTS web site and how to navigate through it. Care has been taken to design it so as to make all content accessible through a few clicks, in a hopefully intuitive manner.

2.1 The main menu

The main menu, shown in Fig. 1, is always located at the top left-hand corner of the page. The *Front Page* link is just

[2] Note: this idea was first raised in the now defunct JETWEB project [12]. The CERN Generator Services (GENSER) project also maintains a set of web pages with basic generator validation plots/calculations.

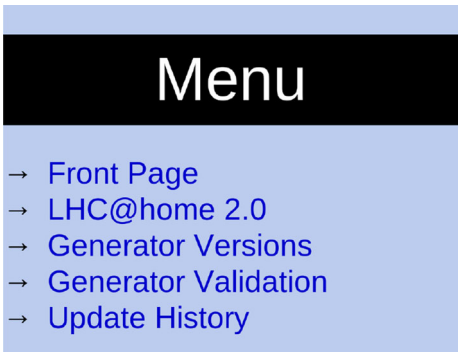


Fig. 1 The main MCPLOTS menu

a "home" button that takes you back to the starting page for MCPLOTS, and the *LHC@home 2.0* one takes you to the external LHC@HOME 2.0 web pages, where you can connect your computer to the volunteer cloud that generates events for MCPLOTS.

The *Generator Versions* link opens a configuration page that allows you to select which generator versions you want to see results for on the site. The default is simply to use the most recently implemented ones, but if your analysis, for instance, uses an older version of a particular generator, you can select among any of the previously implemented versions on the site by choosing that specific version on the *Generator Versions* page. All displayed content on the site will thereafter reflect your choice, as you can verify by checking the explicit version numbers written at the bottom of each plot. You can return to the *Generator Versions* page at any time to modify your choice. After making your choice, click on the *Front Page* button to exit the *Generator Versions* view.

The *Generator Validation* link changes the page layout and content from the *Front Page* one, to one in which different generator versions can be compared both globally, via χ^2 values, and individually on each distribution. This view will be discussed in more detail in Sect. 2.4.

The *Update History* link simply takes you to a page on which you can see what the most recent changes and additions to the site were, and its previous history.

As an experimental social feature, we have added a "like" button to the bottom of the front page, which you can use to express if you are happy with the MCPLOTS site.

2.2 The analysis filter

Immediately below the main menu, we have collected a few options to control and organize which analyses you want to see displayed on the site, under the subheading *Analysis Filter*, illustrated in Fig. 2.

At the time of writing, the main choice you have here is between *ALL pp/ppbar* (for hadron collisions) and *ALL ee* (for fragmentation in electron-positron collisions). The default is *ALL pp/ppbar*, so if you are interested in seeing all hadron-collider analyses, you will not have to make any changes here. Using the *Specific Analysis* dropdown menu, you can also select to see only the plots from one particular RIVET analysis. The latter currently requires that you know the RIVET ID of the analysis you are interested in. The ID is typically formed from the experiment name, the year, and the inSPIRE ID (or SPIRES ID, for older analyses) of the paper containing the original analysis, as illustrated in Fig. 2 (the numbers beginning with "I" are inSPIRE codes, while ones beginning with "S" are SPIRES ones). You can also find this information in the RIVET user manual [14] and/or on the http://rivet.hepforge.org/ rivet web pages.

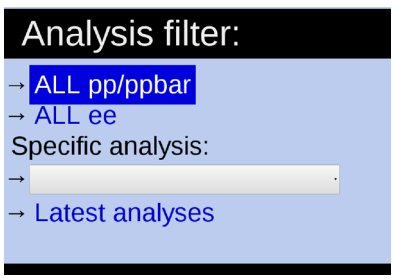
Finally, if you click on *Latest Analyses*, only those analyses that were added in the last update of the site will be shown. This can be useful to get a quick overview of what is new on the site, for instance to check for new distributions that are relevant to you and that you may not have been able to see on your last visit to the site. More options may of course be added in the future, in particular as the number of observables added to the pp/ppbar set grows.

2.3 Selecting observables

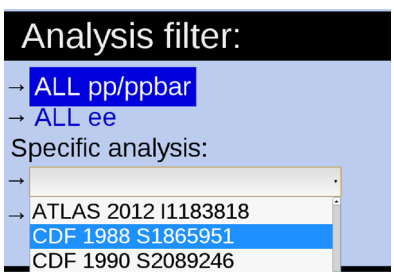
Below the *Analysis Filter*, the list of processes and observables for the selected set of analyses is shown. This is illustrated in Fig. 3.

Clicking on any blue link below one of the process headers (e.g. below the "Jets" header), Fig. 3, or any blue link in the shaded drop-down menus, Fig. 3, will open the plot page for that observable in the right-hand part of the page.

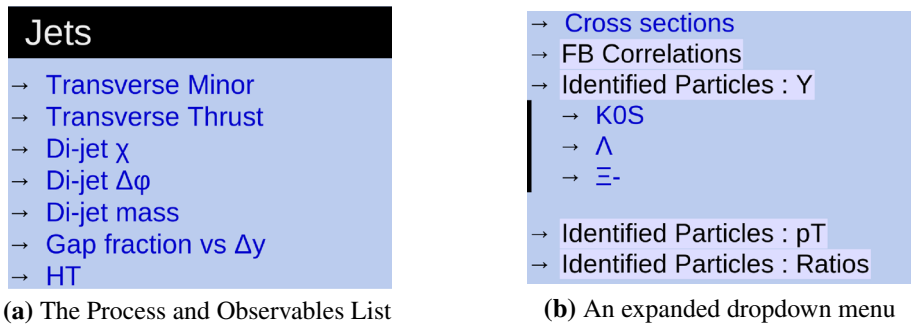
Fig. 2 The analysis filter submenu; normal view (*left*) and after clicking on the *Specific Analysis* dropdown menu (*right*)



(a) Analysis Filter: Normal View



(b) Analysis Filter: Dropdown Menu



(a) The Process and Observables List **(b)** An expanded dropdown menu

Fig. 3 Illustrations of the process and observables list; normal view (*left*) and after clicking on a shaded dropdown menu (*right*), in this case *Identified Particles: Y*

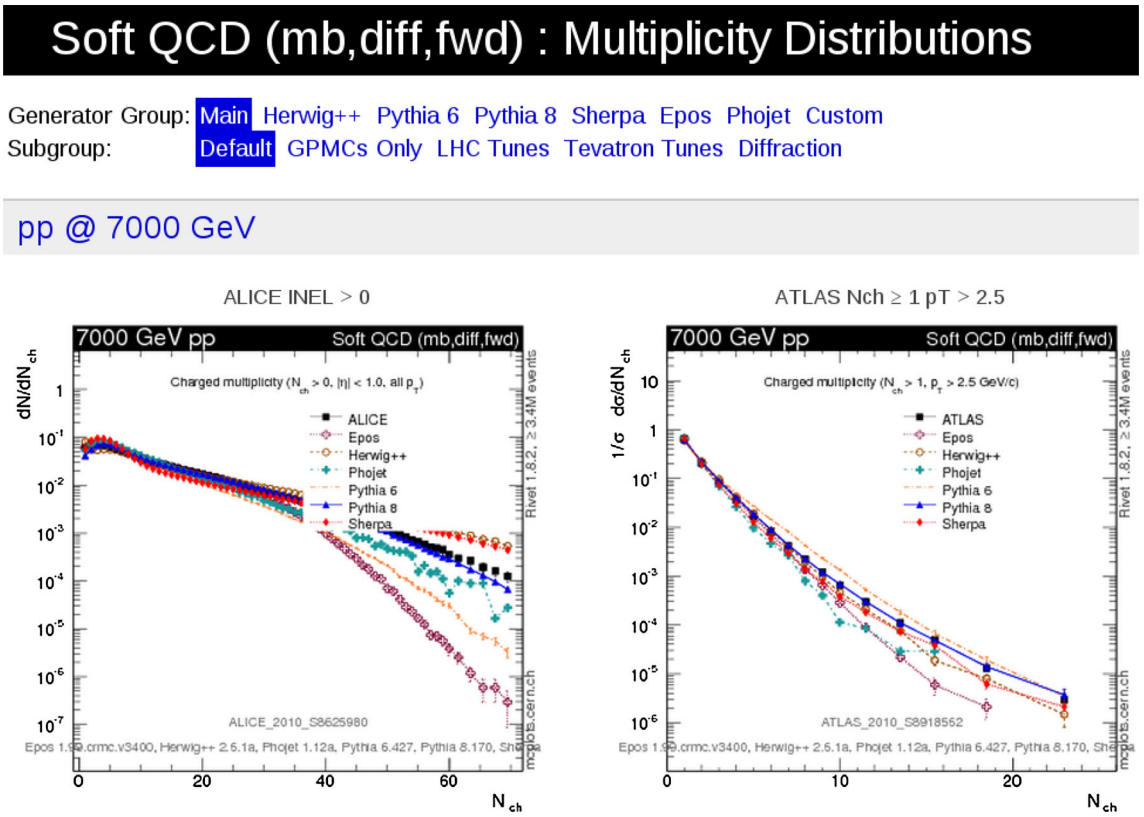
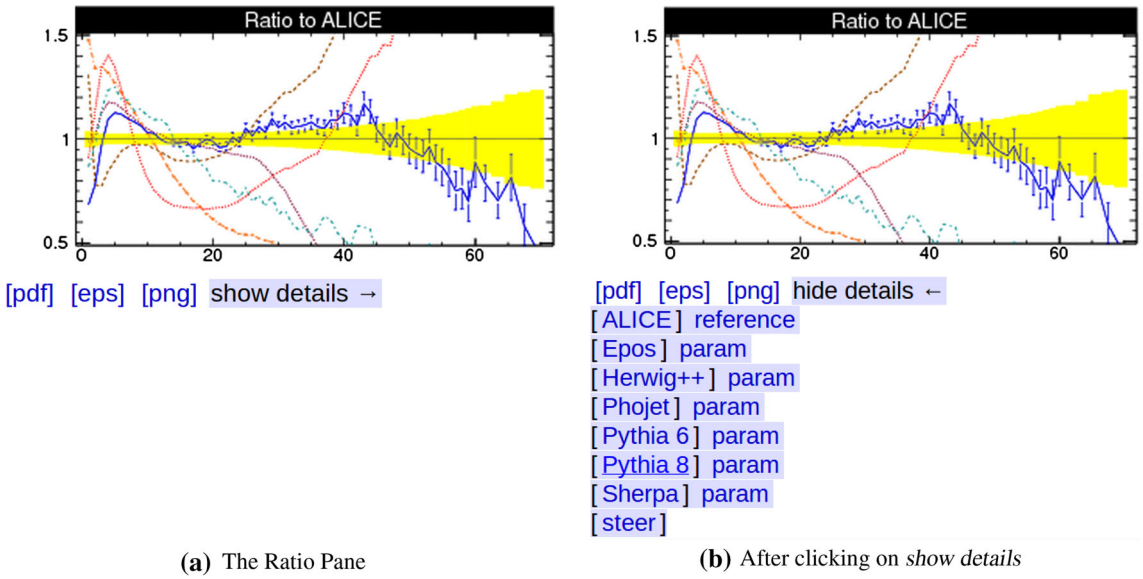


Fig. 4 The plot page. The generator and tune group selections are at the *top*, followed by the available plots for the chosen observable, ordered by CM energy and subordered alphabetically

At the top of the plot page, Fig. 4, you can select which generators and tune combinations you want to see on the page. By default, you are shown the results obtained with default settings of the available generators, but the links for each generator give you access to see results for different tune and model variations. Use the *Custom* link to specify your own set of generators and tunes. The available plots for the chosen settings are shown starting with the highest CM energies at the top of the page, and, for each CM energy, cascading from left to right alphabetically.

For many observables, measurements have been made using a variety of different cuts and triggers. These are indicated both above the plots and on the plots themselves, so as to minimize the potential for misinterpretation. In the example of charged-particle multiplicity distributions shown in Fig. 4, the two first plots that appear are thus ALICE (left), for their INEL > 0 cuts [41], and ATLAS (right), using their $N_{\mathrm{ch}} \geq 1$ and $p_T > 2.5$ GeV cuts [42]. We explain how to find the correct references and run cards for each plot and generator below. Note that, for the Monte Carlo runs, the number of events in the smallest sample is shown along the right-hand edge of each plot. I.e., if two generators were used, and the statistics were N_1 and N_2 events, respectively, the value printed is $\min(N_1, N_2)$.

Underneath each plot is shown a ratio pane, showing the same results normalized to the data (or to the first MC curve if there are no data points on the plot). This is illustrated in Fig. 5.



(a) The Ratio Pane **(b)** After clicking on *show details*

Fig. 5 The ratio pane, before (**a**) and after (**b**) clicking on the *show details* link. The example corresponds to the left-hand plot shown in Fig. 4

The vertical range of the ratio plot is fixed to [0.5, 1.5], thus larger deviations than this will exceed the boundaries of the ratio plot. The central shaded (yellow) band corresponds to the experimental uncertainty, at the 1σ level, as reported by RIVET. For MC-to-MC comparisons, it represents the statistical uncertainty of the reference MC.

Immediately below the ratio pane are links to download the plot (main plot + ratio pane) in higher resolution and/or in vector graphics formats. Currently, *pdf*, *eps*, and *png* formats are available. For publications or presentations, we strongly recommend using these high-resolution versions rather than the web-optimized ones displayed on the page, to avoid undesirable pixelation effects.

Additional information about the plot, such as data tables, references, and run cards, can be accessed by clicking on the *show details* dropdown menu, illustrated in Fig. 3b. In this example, clicking on *[ALICE]* gives you a text file containing a table of the experimental data points, along with additional information (from RIVET) about the plot. Clicking on *reference* sends you to the inSPIRE page for the experimental paper in which the measurement was presented. Clicking on a generator name will give you a text file containing a table of the results obtained with that generator, together with additional technical information from MCPLOTS (including an additional table which is used by MCPLOTS to combine the results of several different runs). You can use or ignore the additional information in these files as you wish. Clicking on *param* gives you the exact generator run card that was used to make the plot, so that you can see precisely how the results were generated. These cards can also be useful as examples to start generating your own standalone results, or to check that you can reproduce ours. Finally the *steer* link contains the steering card used by the ROOT-based tool that makes the actual plots you see. Though this tool is not yet publicly accessible on the site, just contact us if you would like a copy, e.g. to use it to make your own standalone plots outside of MCPLOTS.

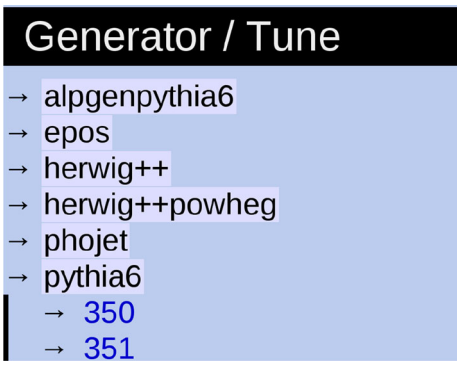


Fig. 6 Generator validation view: excerpt of the generator/tune selection menu

2.4 The generator validation view

Clicking on *Generator Validation* (see the main MCPLOTS menu in Fig. 1) opens the validation view. (You can click on *Front Page* to get back to the default view at any time.) In this view, instead of the *Analysis Filter* you will see a list of event generators beneath the main MCPLOTS menu. You can click on each generator to see a list of the available tunes and model variations for that generator. This is illustrated in Fig. 6.

As an example, clicking on *pythia8* → *default* will show the validation view for default settings of the PYTHIA 8 gen-

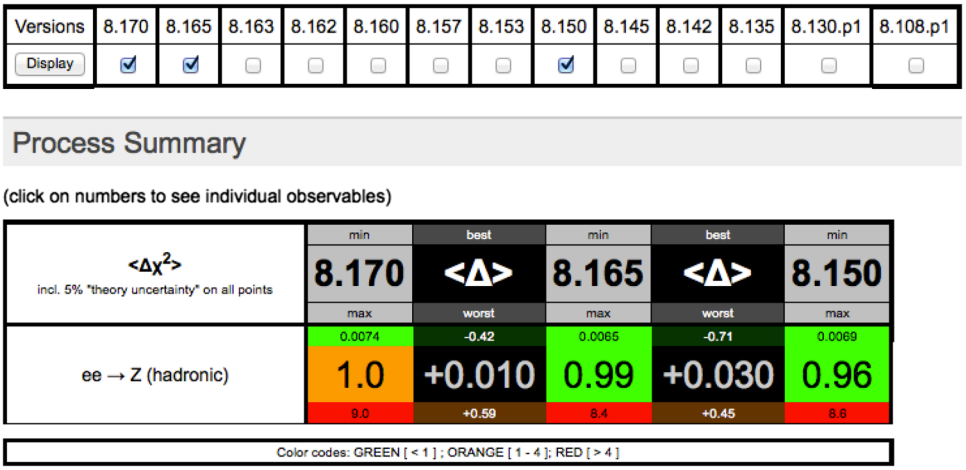


Fig. 7 Generator validation: example showing (an excerpt of) the validation view for default settings of the PYTHIA 8 generator

erator, in the right-hand side of the page, illustrated in Fig. 7. In this view, no plots are shown immediately. Instead you are presented with a table of $\langle\chi^2\rangle$ values, averaged over all measurements within each process category. Note that we use a slightly modified definition of $\langle\chi^2\rangle$,

$$\langle\chi^2\rangle_{\text{MCPLOTS}} = \frac{1}{N_{\text{bins}}}\sum_{i=1}^{N_{\text{bins}}}\frac{(\text{MC}_i - \text{Data}_i)^2}{\sigma^2_{\text{Data},i} + \sigma^2_{\text{MC},i} + (\epsilon_{\text{MC}}\text{MC}_i)^2}, \tag{1}$$

where $\sigma_{\text{Data},i}$ is the uncertainty on the experimental measurement (combined statistical and systematic) of bin number i, and $\sigma_{\text{MC},i}$ is the (purely statistical) MC uncertainty in the same bin. The additional relative uncertainty, ϵ_{MC}, associated with the MC prediction, is commented on below. From the MC and data histograms alone, it is difficult to determine unambiguously whether the number of degrees of freedom for a given distribution is N_{bins} or $(N_{\text{bins}} - 1)$, hence we currently use $1/N_{\text{bins}}$ as the normalization factor for all $\langle\chi^2\rangle$ calculations.

At the top of the page, you can select which versions of the generator you want to include in the table. Click the *Display* button to refresh the table after making modifications.

Below the line labelled *Process Summary*, we show the main table of $\langle\chi^2\rangle$ values for the versions you have selected, as well as the relative changes between successive versions, thus allowing you to look for any significant changes that may have resulted from improvements in the modelling/tuning (reflected by decreasing χ^2 values) or mistunings/bugs (reflected by increasing χ^2 values). The largest and smallest individual $\langle\chi^2\rangle$ values (and changes) in the relevant data set are also shown, in smaller fonts, above and below the average values. To aid the eye, values smaller than 1 are shaded green (corresponding to less than 1σ average deviation from the data), values between 1 and 4 are shaded orange (corresponding to less than 2σ deviation), and values greater than 4 are shaded red, following the spirit of the Les Houches "tune killing" exercise reported on in [10]. In the example shown in Fig. 7, the changes are less than a few per cent, indicative of no significant overall change.

Bear in mind that the statistical precision of the MC samples plays a role, hence small fluctuations in these numbers are to be expected, depending on the available numbers of generated events for each version. A future revision that could be considered would be to reinterpret the statistical MC uncertainties in terms of uncertainties on the calculated $\langle\chi^2\rangle$ values themselves. This would allow a better distinction between a truly good description (low $\langle\chi^2\rangle$ with small uncertainty) and artificially low $\langle\chi^2\rangle$ values caused by low MC statistics (low $\langle\chi^2\rangle$ with large uncertainty). Such improvements would certainly be mandatory before making any rigorous conclusions using this tool, as well as for objective interpretations of direct comparisons between different generators. For the time being, the tool is not intended to provide such quantitative discriminations, but merely to aid authors in making a qualitative assessment of where their codes and tunes work well, and where there are problems.

Note: to make these numbers more physically meaningful, the generator predictions are assigned a flat $\epsilon_{\text{MC}} = 5\,\%$ "theory uncertainty" in addition to the purely statistical MC uncertainty, see equation (1), as a baseline sanity limit for the achievable theoretical accuracy with present-day models. A few clear cases of GIGO[3] are excluded from the χ^2 calculation, but some problematic cases remain. Thus, e.g., if a calculation returns a too small cross section for a dimensionful quantity, the corresponding χ^2 value will be large, even though the shape of the distribution may be well described. It could be argued how this should be treated, how much uncertainty should be allowed for each observable, how to compare consistently across models/tunes with different numbers of generated events, whether it is reasonable to include observ-

[3] Garbage In, Garbage Out.

Fig. 8 Generator validation: example showing (an excerpt of) the detailed observable-by-observable validation view for default settings of the PYTHIA 8 generator, comparing two different versions

pp/ppbar → Underlying Event

Observable	Cut	Energy	χ^2 (8.165)	Δ	χ^2 (8.108.p1)
<pT> vs Nch (AWAY)	ATLAS pT > 0.5	7000	0.042	-0.23	0.27
	ATLAS pT > 0.5	900	0.13	-0.050	0.18
<pT> vs Nch (TRNS)	ATLAS pT > 0.5	7000	0.082	-0.29	0.37
	ATLAS pT > 0.5	900	0.35	-0.24	0.59

ables that a given model is not supposed to describe, etc. These are questions that we do not believe can be meaningfully (or reliably) addressed by a fully automated site containing tens of thousands of model/observable combinations. In the end, the interpretation of the information we display is up to you, the user. That is also why, at least for the time being, we do not display any direct comparisons between different MC generators.

To see the values for all of the individual distributions that enter a given process type, click on any of the $\langle\chi^2\rangle$ values in the table. Or, to see a comparison between two successive versions, click on any of the $\Delta\langle\chi^2\rangle$ values. The result of doing the latter is illustrated in Fig. 8. You can now scroll down the list of observables to get a more detailed view of which observables actually changed, and how. Also in this view, you can click on the numbers in the table.

Doing so takes you to the final level of validation detail, which is a plot showing the two generator versions compared on the given observable. In the example of Fig. 8, clicking on the value -0.29 in the third row will produce the plot shown in Fig. 9. As usual, you can use the links below the plot to download it in various formats or obtain all the numerical information that was used to make it.

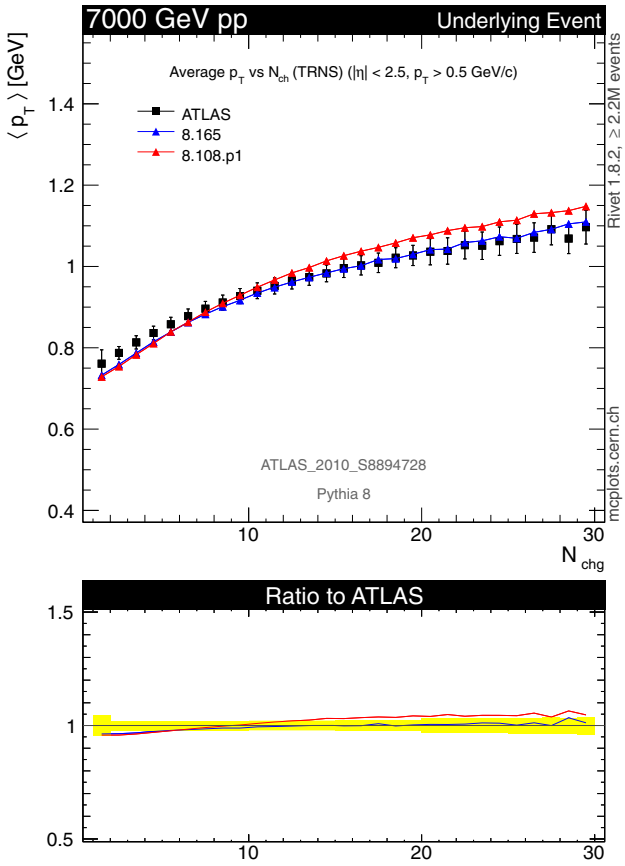


Fig. 9 Generator validation: example showing the plot produced by clicking on the number -0.29 in the $\langle\chi^2\rangle$ table shown in Fig. 8 (downloaded in PDF format). Data from [43]

3 Architecture and implementation

3.1 Histogram and data format

On MCPLOTS, a common numerical file format is used for both experimental data and MC generator results. The format is based on RIVET's "flat format" for histograms[4] with modifications to store additional information which is necessary for histograms merging. The format itself is a plain text ASCII file which has the advantage of being both human and machine readable and writable. Each file is divided into sections with the format

```
# BEGIN type
param1=value1
param2=value2
bin1 contents
bin2 contents
...
# END type
```

The following types of sections have been defined:

- PLOT—description of plot parameters, for example axis labels or scale type (linear/logarithmic).
- HISTOGRAM—plot data (bin contents), used by the plotting tool to produce the plots.

[4] See Section 4.1 of the RIVET manual [14] for details. MCPLOTS uses RIVET's `aida2flat` script to convert the RIVET-output to the flat format, and thus currently relies on RIVET's initial, AIDA-based, histogramming. A future update is planned to upgrade the site to use RIVET 2.0, which employs a new histogram format called YODA.

- HISTOSTATS, PROFILESTATS—additional histogram (or profile) statistical data, used to merge histograms from different subruns.
- METADATA—description of MC generator parameters, for example generator name and version, simulated process and cross section, etc.

Section contents can be of two kinds: parameter value definitions or bin value definitions. Parameter value definitions have the same format for all types of sections. Bin value definitions contain several values on each line, e.g. describing the bin position, contents, and uncertainties, with the following specific formats for each section type:

```
HISTOGRAM:
 xLo xMid xHi   yValue error- error+

HISTOSTATS:
 xLo xMid xHi   entries sumW sumW2 sumXW sumX2W

PROFILESTATS:
 xLo xMid xHi   entries sumW sumW2 sumXW sumX2W sumYW sumY2W sumY2W2
```

where:

- xLo, xHi—bin edges low and high positions.
- xMid—bin centre position.
- yValue—bin value.
- error−, error+ —bin negative and positive errors.
- entries—number of bin fills.
- sum*—sums of various quantities accumulated during bin fills: weight, weight*weight, X*weight, X*X*weight, Y*weight, Y*Y*weight, Y*Y*weight*weight.

As mentioned above, the sum-type quantities provide the additional statistical information needed to combine histograms from different runs.

3.2 Directory structure

MCPLOTS is structured as a directory tree, with separate subdirectories for different purposes (WWW content, MC production scripts, documentation, …). In the following, we assume that this directory structure has been installed in a global home directory, which we call

```
$HOME     = /home/mcplots     # MCPLOTS home directory
```

The location `/home/mcplots` corresponds to the current implementation on the **mcplots.cern.ch** server. There are five important subdirectories in the home directory:

```
$DOC      = $HOME/doc         # Documentation and Help
$PLOTTER  = $HOME/plotter     # Source code for ROOT-based plotting tool
$POOL     = $HOME/pool        # Output from BOINC cluster
$RELEASE  = $HOME/release     # Collection of generated (MC) data
$SCRIPTS  = $HOME/scripts     # Production and update scripts
$WWW      = $HOME/www         # Front of house: WWW pages and content
```

The `$DOC` directory contains documentation and help content that extends and updates this write-up and should be consulted by future developers. The `$PLOTTER` directory contains the C++ source code and Makefile for the small plotting utility used to generate the plots on the mcplots web pages. It only depends on ROOT and takes histogram/data files in the format used on the mcplots site as input. It can be copied and/or modified for personal use, if desired, as e.g., in [44]. Any changes to the code located in the `$PLOTTER` directory itself, on the MCPLOTS server, will be reflected in the plots appearing on the site, after the plotter has been recompiled and the browser cache is refreshed.

The `$POOL` directory contains the MC data sets generated by the Test4Theory BOINC cluster.

The `$RELEASE` directory contains MC data sets generated manually and combinations of data sets (so-called releases).

For example typically the data visible on the public site **mcplots.cern.ch** is a combination of BOINC data sets (which have large statistics) and a number of manually generated data sets applied on top, for distributions that cannot be run on the BOINC infrastructure, or to add new versions of generators which were not yet available at the time of generation of the BOINC data set.

The `$SCRIPTS` directory contains various scripts used to organize and run generator jobs, and to update the contents of the `$WWW` directory which contains all the HTML and PHP source code, together with style and configuration files, for the **mcplots.cern.ch** front end of the site.

This directory structure is clearly visible in the MCPLOTS SVN repository, which is located at https://svn.cern.ch/reps/mcplots/trunk/, and which can be accessed through a web browser at https://svnweb.cern.ch/trac/mcplots/browser/trunk/. Neither the `$RELEASE` nor the `$POOL` directory are part of the SVN repository, so as to keep the repository minimal.

4 Implementing a new analysis

In this section, we describe how to implement additional analyses in the MCPLOTS generation machinery so that the results will be displayed on the **mcplots.cern.ch** web page.

The implementation of new analyses relies on four main steps, the first of which only concerns comparisons to experimental data (it is not needed for pure MC comparisons). Furthermore, both the first and second steps are becoming standard practice for modern experimental analyses, minimizing the remaining burden. The steps are described in Sects. 4.1–4.4.

4.1 HEPDATA

First, the relevant set of experimentally measured data points must be provided in the Durham HEPDATA repository, which both RIVET and MCPLOTS rely on, see [13] for instructions on this step. Important aspects to consider is whether the data points and error bars are all physically meaningful (e.g., no negative values for a positive-definite quantity) and whether a detector simulation is required to compare MC generators to them or not.

If a detector simulation is needed ("detector-level" data), it may not be possible to complete the following steps, unless some form of parametrized response- or smearing-function is available, that can bring MC output into a form that can be compared directly to the measured data. For this reason, data corrected for detector effects within a phase-space region corresponding roughly to the sensitive acceptance of the apparatus ("particle-level" with a specific set of cuts defining the acceptance) is normally preferred.[5]

For data corrected to the particle level, the precise definition of the particle level (including definitions of stable-particle lifetimes, phase-space cuts/thresholds, and any other corrections applied to the data) must be carefully noted, so that exactly the same definitions can be applied to the MC output in the following step.

4.2 RIVET

A RIVET analysis must be provided, that codifies the observable definition on MC generator output. As already mentioned, this is becoming standard practice for an increasing number of SM analyses at the LHC, a trend we strongly encourage and hope will continue in the future. RIVET analyses already available to MCPLOTS can e.g. be found in the `analyses.html` documentation file of the RIVET installation used by MCPLOTS, located in the RIVET installation directory. For instructions on implementing a new RIVET analyses, see [14], which also includes ready-made templates and examples that form convenient starting points.

For comparisons to experimental data, see the comments in Sect. 4.1 above on ensuring an apples-to-apples comparison between data and Monte Carlo output.

4.3 Event generation

Given a RIVET analysis, inclusion into the production machinery and display of MCPLOTS can be achieved by editing a small number of steering files. We illustrate the procedure with the concrete example of an OPAL analysis of charged-hadron momentum spectra in hadronic Z decays [45], for which a RIVET analysis indeed already exists, called `OPAL_1998_S3780481`.

First, input cards for the MC generator runs must be provided, which specify the hard process and any additional settings pertaining to the desired runs. (Note that random number seeds are handled automatically by the MCPLOTS machinery and should not be modified by the user.) One such card must be provided for each MC generator for which one wishes results to be displayed on the site. Several cards are already available in the following directory,

```
CARDS = $SCRIPTS/mcprod/configuration
```

where the location of the `$SCRIPTS` directory was defined in Sect. 3.2. For the example of the OPAL analysis, we wish to

[5] Corrections to full phase space (4π coverage, without any tracking or calorimeter thresholds) are only useful to the extent the actual measured acceptance region is reasonably close to full phase space in the first place, and should be avoided otherwise, to avoid possible inflation of errors by introducing model-dependent extrapolations.

run a standard set of hadronic Z decays, and hence we may reuse the existing cards:

```
$CARDS/herwig++-zhad.params
$CARDS/pythia6-zhad.params
$CARDS/pythia8-zhad.params
$CARDS/sherpa-zhad.params
$CARDS/vincia-zhad.params
```

The generator and process names refer to the internal names used for each generator and process on the MCPLOTS site. (The site is constructed such that any new process names automatically appear on the web menus, and each can be given a separate HTML and LaTeX name, as described in Sect. 4.4.) In this example, hadronic Z decays are labelled `zhad`. For processes for which the `$CARDS` directory does not already contain a useful set of card files, new ones must be defined, by referring to the documentation of examples of each generator code separately. This is the only generator-dependent part of the MCPLOTS machinery.

Having decided which cards to use for the hard process(es), information on which observables are available in the RIVET analysis corresponding to our OPAL measurement are contained in the file

```
$RIVET/share/Rivet/OPAL_1998_S3780481.plot
```

Decide which individual distributions of that analysis you want to add to MCPLOTS, and what observables and cuts they correspond to. In our case, we shall want to add the distributions `d01-x01-y01`, `d02-x01-y01`, `d05-x01-y01` and `d06-x01-y01`, which represent the x distributions (with x defined as momentum divided by half the centre-of-mass energy) in light-flavour events (`d01`), the x distributions in c-tagged events (`d02`), and the $\ln(x)$ distributions in light-flavour (`d05`) and c-tagged (`d06`) events, respectively.

The new analysis is added to the MCPLOTS production machinery by adding one line to the file

```
$CARDS/rivet-histograms.map
```

for each new observable. In the case of the OPAL analysis, we would add the following lines

```
#== MC Initialization and Cuts in GeV ===  ========= Rivet ========  ====== mcplots =======
# beam process  Ecm  pTMin,Max,mMin,Max    RivetAnalysis_Hist          Obs              Cut

  ee    zhad    91.2        -        OPAL_1998_S3780481_d01-x01-y01    x     opal-1998-uds
  ee    zhad    91.2        -        OPAL_1998_S3780481_d02-x01-y01    x     opal-1998-c
  ee    zhad    91.2        -        OPAL_1998_S3780481_d05-x01-y01    xln   opal-1998-uds
  ee    zhad    91.2        -        OPAL_1998_S3780481_d06-x01-y01    xln   opal-1998-c
```

If several different analyses include the same observable (or approximately the same, e.g., with different cuts), we recommend to assign them the same consistent name in the `Obs` column (such as `x` and `lnx` above). This will cause the corresponding plots to be displayed on one and the same page on the MCPLOTS web pages, rather than on separate ones, with `Cut` giving a further labelling of the individual distributions on each page.

As an option to optimize the MC production, `rivet-histograms.map` allows the specification of a set of phase-space cuts within which to restrict the hard-process kinematics. These optional settings can be provided by changing the optional `pTMin`, …, `mMax` columns above (in our example, such cuts are not desired, which is indicated by the `-` symbol). Note that such cuts must be applied with extreme care, since they refer to the hard partonic subprocess, not the final physical final state, and hence there is always the risk that bremsstrahlung or other corrections (e.g., underlying event) can cause events to migrate across cut thresholds. In the end, only the speed with which the results are obtained should depend on these cuts, not the final physical distributions themselves (any such dependence is a sign that looser cuts are required). We would like to point out that a separate generator run is required for each set of MC cuts. Hence, it is useful to choose as small a set of different cuts as possible. The following is an excerpt from `rivet-histograms.map` which concerns a CDF analysis of differential jet shapes [46], in which two different generator-level $\hat{p}_\perp$ cuts are invoked to ensure adequate population of a much larger number of jet $p_\perp$ bins (here ranging from 37 to 112 GeV):

```
#=== MC Initialization and Cuts in GeV ==    ======= Rivet ======  === mcplots ===
# beam process    Ecm  pTMin,Max,mMin,Max      RivetAnalysis_Hist     Obs      Cut
 ppbar    jets   1960        17      CDF_2005_S6217184_d01-x01-y01 js_diff cdf3-037
 ppbar    jets   1960        17      CDF_2005_S6217184_d01-x01-y02 js_diff cdf3-045
 ppbar    jets   1960        37      CDF_2005_S6217184_d02-x01-y02 js_diff cdf3-073
 ppbar    jets   1960        37      CDF_2005_S6217184_d02-x01-y03 js_diff cdf3-084
```

After changing `rivet-histograms.map`, it is necessary to run the available MC generators in order to produce the new histograms, and then update the MCPLOTS database in order to display the results. We will describe how to include new generator tunes, run the generators and update the database in Sect. 5. For now, we will assume that the results

of MC runs are already available, and continue by discussing how to translate the language of `rivet-histograms.map` into the labels that are displayed on the **mcplots.cern.ch** pages.

4.4 Displaying the results on MCPLOTS

The next step after implementing a new analysis is to define a correspondence between the internal names (as declared in the `rivet-histograms.map` file) and the names that will be displayed on the web page and on plots.Correspondence definitions are collected in the configuration file

```
$WWW/mcplots.conf
```

```
ctm           =            ! Transverse Minor              ! Central Transverse Minor
gapfr-vs-dy-fb = Jets + Veto ! Gap fraction vs &Delta;y (FB) ! Gap fraction vs #Deltay (FB)
```

All internal process-, observable- and cut names defined by adding new lines to the histogram map should be defined in the configuration file. Process names are translated by adding a line

```
process_name  =  !  HTML name ! plot label in
                 LaTeX format
```

to the list of processes at the beginning of `mcplots.conf`. Please note that the sequence of process name definitions in this file also determines the order of processes in the web page menu. In Fig. 10, the "Jets" label appears before the "Total σ" label because the relevant definitions are in consecutive lines in `mcplots.conf`. The labels associated with the "jets" process (encased by red boxes in Fig. 10) have been generated by including the line

```
jets  =  !  Jets  !  Jet production
```

The internal name (*jets*) is left of the equality sign, the HTML name (*Jets*, i.e the text in solid red boxes) is defined in the centre, while the plot label (*Jet production*, i.e. the text in dashed red boxes) stands to the right. MCPLOTS allows for both an HTML name and a plot label so that (a) the web page is adequately labelled, and (b) the information on the plot is sufficient to distinguish it from all other figures of the reproduced article, even if the plot is stored separately. Since the plotting tool uses LaTeX, the plot label should be specified in LaTeX format, except that the hash symbol should be usedinstead of the back-slash symbol. Note that the exclamation marks are mandatory, as they are also used to separate different types of observable names.

When defining observable names, it is possible to group a set of observables into a submenu by specifying an optional submenu name after the equality sign, but before the first exclamation mark. This means that the translation of internal observable names proceeds by adding lines like

```
observable_name = (HTML submenu) ! HTML name
                  ! plot label in LaTeX format
```

to `mcplots.conf`. It is not necessary to follow a predefined ordering when adding observable name correspondences. Figure 10 illustrates how parts of the declaration lines are related to the web page layout. The HTML submenu name (e.g. *Jets + Veto*, i.e. the text in the solid black box) will appear as a link on a grey field. The HTML name of the displayed plot (*Transverse Minor*, i.e. the text in solid green boxes) enters both in the menu and on top of the page, while in the LaTeX observable name (*Central Transverse Minor*, i.e. the text in the dashed green box) is printed directly onto the plot.

Finally, cut names have to be translated as well. This is achieved by expanding `mcplots.conf` with cut declaration lines in the format

```
cut_name     = ! HTML name
               ! plot label in LaTeX format
```

The features resulting from adding the line

```
cms1-pt090  = ! CMS  90 < pT < 125
              ! 90 < p_{#perp} < 125
```

are shown, encased in magenta boxes, in Fig. 10. Again, the plot label ($90 < p_\perp < 125$, i.e. the text in the dashed magenta box) is included in the plot—in parentheses, and after the observable label.

After translating the internal process, observable and cut names, the display on MCPLOTS is fixed. For the sample OPAL analysis discussed in Sect. 4.3, the current layout would be obtained by adding the lines

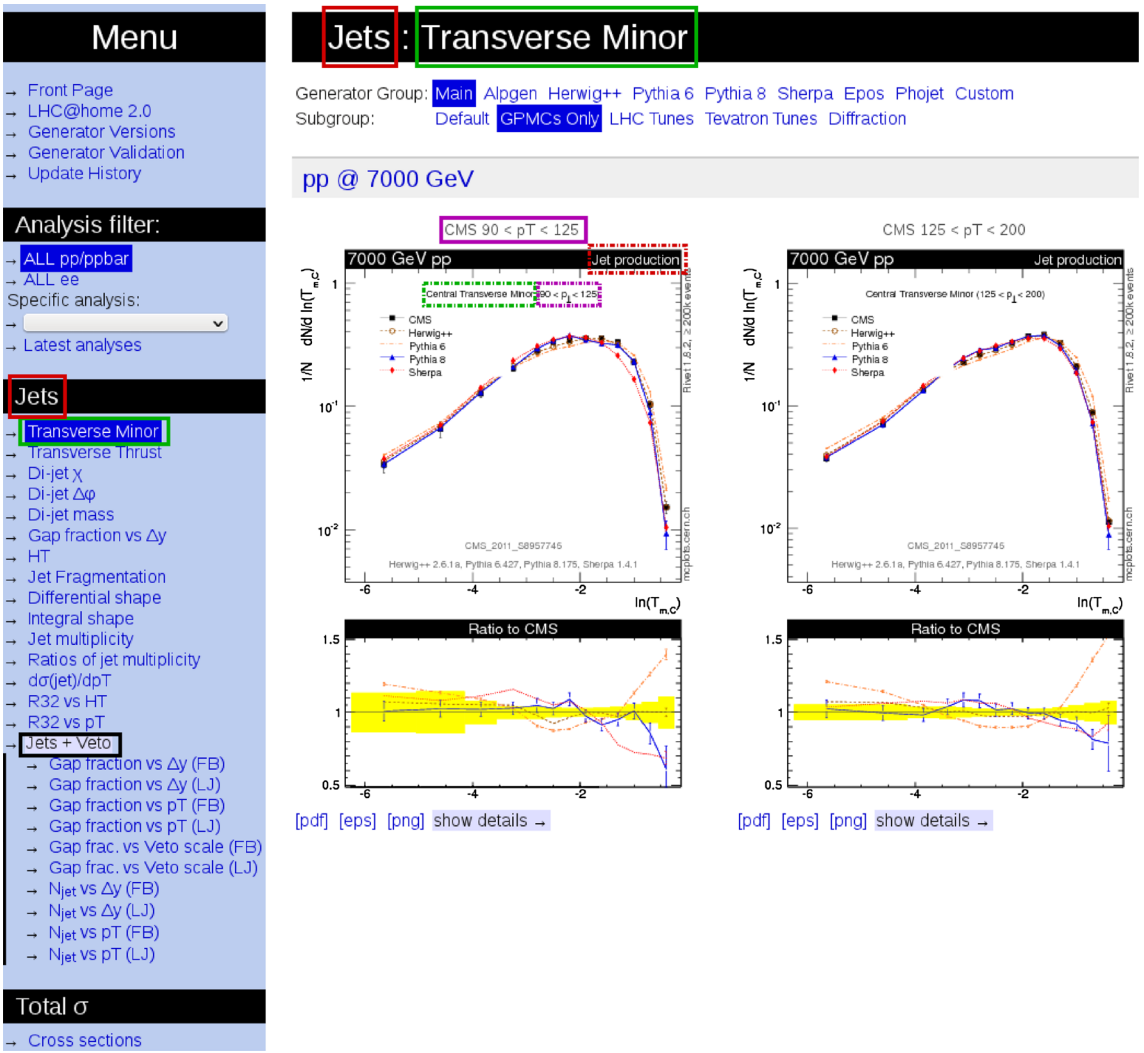


Fig. 10 Snapshot of an MCPLOTS page, to serve as an illustration of web page labels. Most label items are produced by processing the file `$WWW/mcplots.conf`. Section 4.4 discusses changes of `$WWW/mcplots.conf`, using the text encased by coloured boxes as examples

```
# Process names
  zhad              = ! Z(hadronic)              ! Z(Hadronic)
# Observable names
  x                 = ! Scaled momentum          ! Scaled momentum
  xln               = ! Log of scaled momentum   ! Log of scaled momentum
# Cut names
  opal-1998-uds     = ! OPAL u,d,s events        !  OPAL u,d,s events
  opal-1998-c       = ! OPAL c events            !  OPAL c events
```

to the configuration file `$WWW/mcplots.conf`. These steps include a new analysis into the MCPLOTS display framework, so that new plots will be visible after the next update of the database. Before discussing how to update the MCPLOTS framework to produce MC runs for the new analysis and display the corresponding plots (Sect. 5), we will discuss some advanced display possibilities that are steered by `mcplots.conf`.

4.4.1 Tune groups

MCPLOTS further allows to manipulate which of the available generator runs should be displayed together in one plot. This is possible by defining *tune groups* in `mcplots.conf`. Tune groups apply globally to all processes and observables. All available groups will be displayed as "Generator groups" or "Subgroups" between the black observable label bar and

the grey collider information bar. The definition of a tune group requires three steps. To begin with, a correspondence between the internal generator name[6] and the public name has to be defined by including a line in the format

```
generator_name   = ! generator name !
```

in the same section of `mcplots.conf` as other generator names. This should be followed by the definition of the tune name through a line like

```
generator_name.tune_name  = ! tune name !
```

in the tune name section of the configuration file. Furthermore, a line style for this particular tune has to be defined in the format

```
generator_name.tune_name
 = ColR ColG ColB lineStyle lineWidth
      \ markerStyle markerSize
```

where we have used the "\" character to imply line continuation and the available options for the style settings are documented in the ROOT web documentation.[7] Once multiple generator tunes have been named, tune subgroups can be defined. For this, add lines in the format

```
generator_group_name.subgroup_name
  = tune_1, tune_2, tune_3
```

at the tune group section of `mcplots.conf`. As an example, let us look at the "HERWIG++ vs. SHERPA" tune group, which is part of the "HERWIG++" generator group menu. The necessary definitions to construct this tune group are

```
# Tune names
  herwig++.default   = ! Herwig++  !
  sherpa.default     = ! Sherpa !
# Generator names
  herwig++           = ! Herwig++  !
  sherpa             = ! Sherpa !
# Tune line styles
  herwig++.default   = 0.6  0.3  0.0   2 1.5   24 1.25
  sherpa.default     = 1.0  0.0  0.0   3 1.5   33 1.4
# Tune group
  Herwig++.Herwig++ vs Sherpa  = herwig++.default, sherpa.default
```

This concludes the discussion of manipulations of the configuration file `mcplots.conf`.

5 Updating the MCPLOTS site

The previous section described how to add new processes and new analyses to the MCPLOTS framework, and how to modify their organization and labelling on the web site. After a new analysis has been implemented, or when updating the site with new tunes, generators, or RIVET versions, the next step is to update the database with new MC generator runs.

In this section, we will briefly discuss how to update existing MC generators by including new generator versions and tunes. (How to implement a completely new event generator is described separately, in Sect. 6). This is followed by a description of how to manually produce MC results, and how to update the database of the development page **mcplots-dev.cern.ch** (which is publicly visible but updates can only be done by MCPLOTS authors), which serves as a pre-release testing server for MCPLOTS. We finish by explaining how to make a public MCPLOTS release, transferring the contents of the development page to the public one (again an operation restricted to MCPLOTS authors). Aside from MCPLOTS authors, these instructions may be useful in the context of standalone (private) clone(s) of the MCPLOTS structure, created e.g. via the public SVN repository, cf. Sect. 3.2.

5.1 Updating existing generators

MCPLOTS takes generator codes from the GENSER repository, which can be found in

```
/afs/cern.ch/sw/lcg/external/MCGenerators*
```

Only generator versions in this repository can be added to the MCPLOTS event generation machinery. To introduce a new generator version on MCPLOTS, changes of the file

```
$SCRIPTS/mcprod/runAll.sh
```

are required.[8] Specifically, it is necessary to update the `list_runs()` function. To include, for example, version 2.7.0 of HERWIG++, with a tune called "default", the line

[6] How to include add new generators and tunes to the event generation machinery will be explained in Sect. 5.

[7] See http://root.cern.ch, specifically the pages on http://root.cern.ch/root/html/TAttLine.htmllineStyle and http://root.cern.ch/root/html/TAttMarker.htmlmarkerStyle. A nice helping tool to create your own colour schemes is http://colorschemedesigner.com.

[8] For generator chains like ALPGEN +HERWIG++, changing the file `$SCRIPTS/mcprod/runRivet.sh` might also be necessary to add accepted chains of versions. This will be explained below.

```
echo ''$mode $conf - herwig++ 2.7.0 default
  $nevt $seed''
```

has to be added. Please note that the `list_runs()` function of `runAll.sh` groups the runs of generators into blocks (e.g. all HERWIG++ runs follow after the comment `# Herwig++`). This order should be maintained. The necessary changes of `list_runs()` are slightly different depending on if we want to include a completely new generator version or simply a new tune for an existing generator version. The above line is appropriate for the former. For the latter, let us imagine we want to add HERWIG++ v. 2.7.0, with two tunes called "default" and "myTune". Then, we should add the string

```
mul ''$mode $conf - herwig++ 2.7.0 @
  $nevt $seed'' ''default myTune''
```

to the HERWIG++ block of `runAll.sh`.

After this, it is necessary to include the novel generator versions and tunes in the file

```
$CARDS/generator_name-tunes.map
```

where "generator_name" is the name of the MC generator to be updated. For our second example, the addition of the lines

```
2.7.0  default
2.7.0  myTune
```

to `$CARDS/herwig++-tunes.map` is necessary. Depending on how tunes are implemented in the MC generator, it might also be necessary to include a new file with the tune parameters into the `$CARDS` directory. Currently, this is the case for HERWIG++ tunes and non-supported tune variations in PYTHIA 8. Say the "myTune" tune of our example would only differ from default HERWIG++ by having unit probability for colour reconnections. The corresponding HERWIG++ input setting

```
set /Herwig/Hadronization/ColourReconnector:
 ReconnectionProbability 1.00
```

should then be stored in a file called `herwig++-myTune.tune` in the `$CARDS` directory. After following the above steps, we have included a new MC generator version (and/or new tunes) into the MCPLOTS event generation machinery.

See Sect. 4.4.1 for how to include new tunes in new or existing "tune groups" on the site, including how to assign tune-specific default marker symbols, line styles, etc.

5.2 Producing MC results

Once new processes, analyses, versions or tunes have been added, we need to produce results for these new settings. This can be done through volunteer computing or manually. We will here, since manual small scale production can be very useful for debugging purposes, describe how to produce MC results manually. The top-level script for producing and analysing of MC generator results is

```
$SCRIPTS/mcprod/runAll.sh [mode] [nevt]
 {filter} {queue}
```

where the two first arguments are mandatory and the two latter ones are optional, with the following meanings:

- `mode`: Parameter governing the distribution of MC generator runs. If `mode` is set to *list*, the script simple returns a list of all possible generator runs. *dryrun* will set up the event generation machinery, but not execute the generation. *local* will queue all desired runs on the current desktop, while *lxbatch* distributes jobs on the lxplus cluster at CERN.
- `nevt`: The number of MC events per run
- `filter`: Optional filter of the MC runs. For instance, if `filter` = *herwig++*, `runAll.sh` will only produce HERWIG++ results.
- `queue`: Queue to be used on the lxplus cluster.

The `runAll.sh` script executes the event generation and analysis steering script

```
$SCRIPTS/mcprod/runRivet.sh
```

for all the desired input settings. It is in principle also possible to run this script separately.

5.3 Updating the MCPLOTS database

To display the results of new MC runs, it is necessary to update the database with the new histograms. MCPLOTS has both a development area (with plots shown on **mcplots-dev.cern.ch**) and an official release area (with plots shown on **mcplots.cern.ch**).

The development pages are updated as follows. Let us assume that we have produced new EPOS results,[9] and these new results are stored in `~/myResults`. Then, log into `mcplots-dev` by

```
$ ssh myUserName@lxplus.cern.ch
$ ssh mcplots-dev
```

Then copy the new information and to the directory `$HOME/release` on this machine. It is encouraged to include the mcplots revision number (e.g. 2000) into the directory name. For example

```
$ mkdir $HOME/release/2000.epos
$ cd $HOME/release/2000.epos
$ cp ~/myResults/info.txt .
$ find ~/myResults/*.tgz | xargs -t -L 1 tar zxf
```

[9] For example by running `$SCRIPTS/mcprod/runAll.sh lxbatch 100k "epos"`.

Now, the new results are stored in the directory `$HOME/release/2000.epos/dat`. To display these (and only these) new histograms on **mcplots-dev.cern.ch**, re-point and update the database by

```
$ cd $WWW
$ ln -sf $HOME/release/2000.epos/dat
$ $SCRIPTS/updatedb.sh
```

These actions update the database, and the new plots will be visible on the **mcplots-dev.cern.ch** pages. The last step is to update the HTML file `$WWW/news.html` including any relevant new information similarly to what has been done for previous releases. Updating the development web pages is a fairly common task while working on mcplots.

Updating the official web pages (**mcplots.cern.ch**) should of course only be done with great caution. For completeness, we here briefly describe the procedure. The contents of a specific revision of the development web page can be transferred to the official site by executing a simple script. For this, log into `mcplots-dev.cern.ch` and execute

```
$ $SCRIPTS/updateServer.sh -r revision
-d/path/to/dir/dat
```

where `revision` is the SVN revision number of the mcplots code that you want sate the server to, and `/path/to/dir/dat` is the full path to the plot data that you would like to display on the web page. Either of the arguments can be omitted, and at least one argument is necessary. For backward compatibility, we advise to display plot data that has been stored on mcplots-dev according to the suggestions of the development web page update. To give some examples, the code of the public web page can be updated to revision number 2000 by

```
$ $SCRIPTS/updateServer.sh -r 2000
```

Further changing the web page to only show results of the above EPOS runs means running

```
$ $SCRIPTS/updateServer.sh -r 2000 -d
 $HOME/release/2000.epos/dat
```

After these steps, the official release of **mcplots.cern.ch** is publicly available. Since official releases are usually scheduled on a monthly (or longer) time scale, we advocate caution when running the update script. Always thoroughly check the site and functionality after performing an update, and roll back to a previous version if any problems are encountered. There is nothing more damaging to a public web service than faulty operation.

6 Implementing a new event generator

In this section, we provide guidelines for adding a new generator to the MCPLOTS framework. An up-to-date set of instructions is maintained in the MCPLOTS documentation files.[10] The guidelines are accompanied by concrete examples and comments, drawn from the experience obtained by the implementation of the ALPGEN generator in the framework.

When adding a new generator, one should note that the MCPLOTS framework relies on the libraries with the code needed for the event generation being available in the format and with configurations as used by the GENSER project. The generator libraries are accessed from the CERN-based AFS location accordingly:[11]

```
/afs/cern.ch/sw/lcg/external/MCGenerators*
```

or from the CVMFS replica which is used by CernVM virtual machines running in the BOINC cluster:

```
/cvmfs/sft.cern.ch/lcg/external/MCGenerators*
```

Runs of MCPLOTS scripts with a generator not supported by GENSER is currently not implemented. For generator not supported by GENSER user can only run the MCPLOTS scripts by checking them out, making private modifications and private generation runs.

We explain the updates needed for implementation of the new generator supported by GENSER with the help of the figure Fig. 11. The figure shows the schematic representation of the path from the Event Generation to the Physical Distributions produced within the MCPLOTS framework. The sequential steps are labelled 1–4.

6.1 Steps 1–3: from the event generation to RIVET histograms

Before running the event generation in step 1 of Fig. 11, the generator configuration scripts and any other necessary files should be set up.

The top-level wrapper script `runAll.sh` (discussed in Sect. 5.1) contains the commands for multiple runs of steps 1–3 of the MCPLOTS framework. Generator-specific code and calls to generator-specific files for steps 1–2 are implemented in the script:

```
$SCRIPTS/mcprod/rungen.sh .
```

The third step, executing RIVET, is not generator specific and is handled by the script

```
$SCRIPTS/mcprod/runRivet.sh .
```

The `runRivet.sh` relies on `rungen.sh` and enables performing steps 1–3 in one go.

[10] http://svnweb.cern.ch/world/wsvn/mcplots/trunk/doc/readme.txt, section on "Adding a new generator".

[11] Contact `genser-dev@cern.ch` for new generator support.

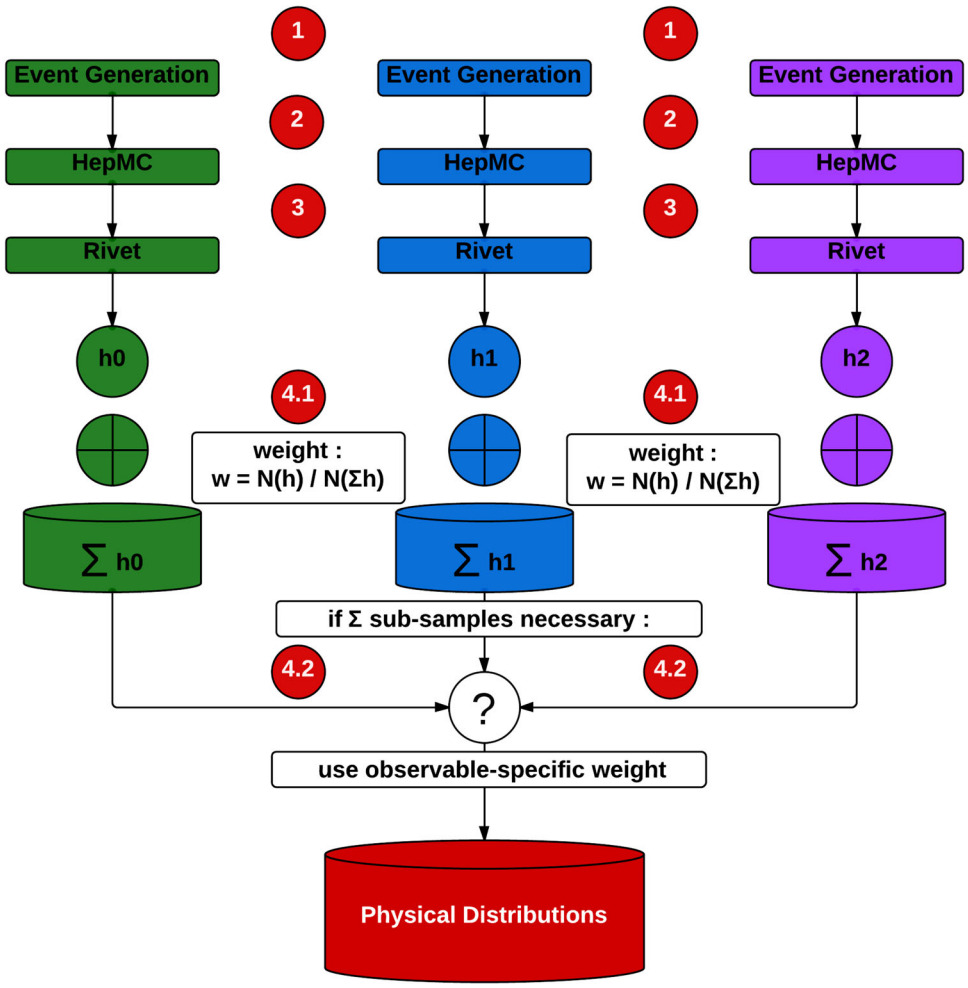


Fig. 11 A schematic representation of the path from the Event Generation to the Physical Distributions produced within the MCPLOTS framework. The sequential steps are labelled 1–4. Their technical implementation in the MCPLOTS framework is described in Sect. 6

The script `rungen.sh` handles calls of the generator-specific files and provides parsing of the generator-specific parameters passed to the upstream `runAll.sh` and `runRivet.sh`.

As discussed in Sect. 4.3, the physics process-configuration files for each event generator should reside in the directory

```
$SCRIPTS/mcprod/configuration.
```

(See also the discussion of "tune groups" in Sect. 4.4.1.) The location of the `$SCRIPTS` directory was defined in Sect. 3.2. Additional generator-specific files may be needed. Examples are linking against the generator libraries of the GENSER project, parsing the generator configuration scripts, or providing generator-level event filters, used by some of the analyses in order to enable efficient filling of the histograms. (Some concrete examples are given below.) Such generator-specific files should be put into a dedicated directory

```
$SCRIPTS/mcprod/(generator).
```

Using ALPGEN as an example, the ALPGEN-specific files reside in the directory

```
$SCRIPTS/mcprod/alpgen.
```

The process-specific files are passed in text format using the standard MCPLOTS naming convention ("`generator-process.params`"), for example:

```
alpgenherwigjimmy-winclusive.params
alpgenpythia6-jets.params.
```

The parameters in these `.params` files are grouped into sections responsible for for various steps of event generation, using the standard ALPGEN formats:

- parameters in sections `[genwgt]` and `[genuwgt]` are used for the generation of the weighted and unweighted events accordingly
- the parameters used for MLM matching (see [47]) are passed in section `[addPS]`
- any shower-specific parameters (e.g., tune selection) to be passed to the parton-shower generator are contained in `[steerPS]`.
- optionally, additional sections can be added as well.

Examples of parameter files for W + jets are available on the MCPLOTS site.

The code responsible for parameter parsing resides in the directory:

```
$SCRIPTS/mcprod/alpgen/utils_alp
```

which also contains a dictionary of unweighting and MLM matching efficiencies that enables the generation of a number of final unweighted ALPGEN + parton-shower events as specified by the user.

The driver scripts for weighted and unweighted event generation reside in the directory:

```
$SCRIPTS/mcprod/alpgen/mcrun_alp
```

This directory also contains generator-level event filters, used by some of the analyses in order to enable efficient filling of the histograms.

The files in the two directories just mentioned, `utils_alp/` and `mcrun_alp/`, are used by the `rungen.sh` script. The script supports optional additional arguments beyond the standard `$mode $conf` ones (described in Sects. 5.1 and 5.2), so that extra generator-specific parameters can be included. In the case of ALPGEN, two extra parameters are currently given: the parton multiplicity of the matrix element and inclusive/exclusive matching of the given run. (Tune-specific ALPGEN parameters are passed at the time of the tune setup, see above. The event-generation scripts support the usage of parameters not directly related to the tune.)

The step 2 in Fig. 11 corresponds to ensuring that the output of the event generation is provided in the standard event record format: HEPMC. For many of the FORTRAN generators, the conversion to the HEPMC event record, as well as utilities for parameter settings and the event generation executable, are provided by A Generator Interface Library, AGILE [48]. The C++ generators are generally able to handle these tasks directly. The ALPGEN event record can be written in HEPMC format either by invoking AGILE using the unweighted events as inputs or by using dedicated interface code following the ALPGEN-internal parton shower interface code examples and adopting the standard HEPEVT to HEPMC conversion utilities. The implementation of the FORTRAN generators in the AGILE framework in order to obtain the events in the HEPMC format is not a prerequisite for running in the MCPLOTS framework. It should however be noted that using AGILE simplifies running of RIVET in step 3 in Fig. 11. It is therefore preferred over dedicated interfaces.

AGILE allows, by setting an input flag, to pass parton level events in LHEF format [29] to the supported General Purpose Event Generators. We thus anticipate that, when showering events in LHEF format with FORTRAN event generators such as PYTHIA and HERWIG, no dedicated code will be necessary for the steps 2 and 3 in figure Fig. 11.

6.2 Steps 4.1, 4.2: from single run histograms to physical distributions

After the analysis run in step 3 in Fig. 11, the MCPLOTS framework merges the single run histograms with the rest of the compatible analysis runs preceding it (steps 4.1 and 4.2). In some cases the single run histograms already correspond to Physical Distributions and the merging is done in order to improve the statistical precision. In other cases several single run histograms need to be combined in order to obtain the Physical Distributions. For the histogram merging it is assumed that the histograms scale linearly with the number of events.

The merging code is located in:

```
$SCRIPTS/mcprod/merge.
```

Once a new source histogram (h0, h1 or h2) in Fig. 11 is obtained, it should be merged to a common destination with the prior compatible histograms in the database. The run of the merging script:

```
$SCRIPTS/mcprod/merge/merge.sh [source]
 [destination]
```

handles the merging of the ASCII histogram inputs as well as the book-keeping of the event multiplicities and the sample cross-section. The merging code runs in two steps as follows:

- first histograms that populate exactly the same regions of phase-space are merged in order to increase the statistical precision of the results. This is denoted as step 4.1 in Fig. 11.
- In the second pass of the code, the use-cases where a number of distinct generation runs are best suited to populate the phase-space needed for the final histograms are handled. This is denoted as step 4.2 in Fig. 11. This step is omitted in cases where a single generation run populates the whole phase-space.

The source histograms are stored in a directory and name structure, from which the analysis and generator setup can be inferred. The exact paths are specified in the `runRivet.sh` script and contain the entries of the `rivet-histograms.map` described in Sect. 4.3 as well as the generator, version, tune and any generator-specific parameters passed to the MCPLOTS scripts. The source file path and name thus provide the analysis and generator information to the merging script. This information is used in the script to decide which of the steps to perform and other merging details (detailed below).

An example use-case that requires only one pass of the merging code is, the standard one of combining multiple runs with different random-number seeds but otherwise identical settings.

An example use-case of the run that requires the second pass of the merging code is filling of the histograms for the production of W/Z bosons in association with jets at the LHC, in the context of multi-leg matrix-element matched samples. The physics distributions of interest have been measured up to large high-p_T jet multiplicities by the LHC experiments [49–56]. The multi-leg generators such as ALPGEN [20], HELAC [57], MADGRAPH [58,59], SHERPA [26], and WHIZARD [60] are well suited for the physics case. For ALPGEN and MADGRAPH, the efficient population of phase-space needed for prediction of high extra jet multiplicities is frequently obtained by producing separate runs with fixed number of extra partons from the matrix element (`Np`) that are passed through the parton shower and hadronization. In this way, the high `Np` sub-samples which are more probable to populate the high extra jet bins in the physics events can be produced with larger integrated luminosity than the low `Np` respectively. A concrete example is the measurement of W + jets by the ATLAS collaborating [49], where the cross-section decreases by an order of magnitude per extra jet while the differential measurement is available for up to ≥ 4 extra jets. Another example requiring the second pass of the merging code is also a combine the physics distributions from generator runs that populate different regions of phase space using generator-level cuts or filters. This is a possible scenario for any generator. The merging code and the histogram structure described in Sect. 3.1 is suited to address such use-cases.

In Fig. 11 the event generation branches in which the histograms h0, h1, h2(…) are produced correspond to the sub-samples populating independent phase-space. An example are sub-samples with 0, 1, 2(…) extra partons in the case of ALPGEN or MADGRAPH in which case the exclusive matching criterion must be used for all but the highest multiplicity sub-sample which is matched inclusively. Multiple sub-samples could also be produced with different phase-space cuts to enhance the number of events in the target corners of the phase-space, such as cuts on the transverse momentum of the hard process partons, such that the high-p_T events can be produced with higher integrated luminosity than the low-p_T events. The merging code contains a procedure to detect and correctly sum the resulting histograms as illustrated in Fig. 11.

The correct merging procedure of the histogram data depends on the normalization already assumed in the RIVET routines. In practice all the current use-cases could be addressed by adding two merging modes `ALPGEN_XSECT` and `ALPGEN_FIXED` to the merging machinery. The `ALPGEN_XSECT` mode is used for histograms from RIVET runs, when no normalization is performed by RIVET. The `ALPGEN_FIXED` mode is used for histograms that are normalized according to the cross-section by RIVET. The new histograms from the run with `Np` partons is merged with the existing histograms for the `Np` sub-sample. The weights for merging of all the `Np` samples needed to form the generator prediction for the observable are than evaluated such that all the `Np` sub-samples are normalized to the same integrated luminosity in `ALPGEN_XSECT`, or with unit weights for `ALPGEN_FIXED` mode correspondingly. The choice of the correct merging procedure is implemented in the merging script `merge.sh` and relies on the histogram naming conventions. In particular the entries of `rivet-histograms.map`, that are constituents of the histogram path, can serve to assign the correct merging mode according to the analysis and generator details. For example the choice of the merging procedure for currently implemented ALPGEN runs uses the `Obs` and `process` fields of `rivet-histograms.map` (Sect. 4.3).

The histogram METADATA (e.g. cross-section and the number of events) described in Sect. 3.1 is consistently updated with the merged values. In addition the sub-sample specific fields are added in the format:

```
samples_X=X_h0:X_h1:X_h2.
```

Here `X` denotes a quantity, e.g. cross-section. The `X_h` denote the value of the quantity for the individual sub-sample. Hence merging the `Np0, Np1, Np2` sub-samples in `ALPGEN_FIXED` mode would result in the following METADATA for `X=crosssection` (in [pb]):

```
crosssection=25554.1
samples_crosssection=20831.403148:
 4285.7315005:436.96532523
```

It should be noted that the merging in the MCPLOTS needs to proceed on-the-fly, since new samples are produced continuously. This makes the merging more challenging than in standard case, where the final statistics and cross-sections of all the `Np` samples is known at the time of analysis and plot production. Thus, in the standard case, the `Np` samples can be normalized to the same integrated luminosity prior to the plot production, while the MCPLOTS machinery needs to deal with merging of the already produced plots.

In case the need arises, further merging modes could be added to the merging structure. For this it is however crucial, that the information needed for consistent merging is available after the RIVET run and correctly transferred to the merging routine. An example piece of such information is the cross-section in the MLM-matching applications, where the final cross-section is only known after the parton shower. Hence, in case the cross-section is not correctly transferred from the matrix-element to the parton-shower generator, the automatic extraction of the information needed for the consistent merging would fail.

7 LHC@home 2.0 and the Test4Theory project

LHC@HOME started as an outreach project for CERNs 50th Anniversary in 2004. The project calculated the stability of proton orbits in the LHC ring, using a software called SIXTRACK [61], distributed to volunteers via the Berkeley Open Infrastructure for Network Computing (BOINC) [37], a popular middleware for volunteer computing. When MCPLOTS was initially conceived, it was clear that significant sustained computing resources would be needed, and it was natural to consider if a setup similar to that of SIXTRACK could be created to meet those needs. The three main reasons were: 1) the computing resources envisaged for MCPLOTS would have been comparable to (or larger than) the total amount of computing power then available to the CERN theory group; 2) the developers of LHC@HOME were keen to explore possibilities to expand the volunteer computing framework towards HEP physics simulations; 3) we saw event simulations as providing a natural way to involve the public in doing LHC science, without the complications that would have accompanied analysis of real data.

A major challenge for the SIXTRACK project had been the heterogenous nature of the resources provided by volunteers. In particular, it is mandatory to support Windows platforms, which the HEP scientific-computing community is significantly less familiar with than variants of Linux, UNIX, or Mac OSX. In the context of event-generator simulations, we viewed the time-consuming task of porting and maintaining our code over such a large range of platforms as a showstopper. An elegant solution to this problem, which factorizes the IT issues almost completely from the scientific software development, is virtualization. The development at CERN of a Scientific-Linux based Virtual-Machine architecture (CERNVM) along with a generic and scalable infrastructure for integrating it into cloud-computing environments (COPILOT) were the two main innovations that allowed us to start the TEST4THEORY project, which now provides the computing backbone for MCPLOTS. This represented the first virtualization-based volunteer computing project in the world, and, with the new additions, the name was updated to LHC@HOME 2.0.

CERNVM itself can be installed using any of a number of different so-called virtualization hypervisors (virtual-machine host software), most of which have been designed to add very little overhead to the virtualized simulation. One that is open-source and available on all platforms is VIRTUALBOX, which thus is a prerequisite for running simulations for TEST4THEORY.

The file system of CERNVM is designed so that only files that are actually accessed are downloaded to the local disk; a huge virtual library can therefore in principle be made available, without causing a large local footprint or long download times.

At the time of writing, the LHC@HOME 2.0 project still uses BOINC as the main middleware for distributing computing jobs to volunteers. What is different with respect to the older SIXTRACK project is that the BOINC tasks in TEST4THEORY are merely wrappers for a VM that is started up on the remote computer. Each such VM task has a default lifetime of 24 h, since volunteers only gain "BOINC credits" each time a BOINC task completes. Inside the wrapper, the VM itself constantly receives jobs, processes them, and sends output back to MCPLOTS. This communication is handled by COPILOT.

Alpha testing of the system internally at CERN began in October 2010. This first test phase was quite technical, concentrating on the requirements on the virtual-machine architecture, on the stability and steady supply of jobs, and on development of the simulation packages themselves. During 2011, a small number of external volunteers were gradually connected as well, many of whom participated actively in testing and debugging this first edition of the system. By the end of the alpha test stage, in July 2011, the system operated smoothly and continuously with about 100 machines connected from around the globe.

During beta testing, in the latter half of 2011, the main line of attack was the scalability of the system. In a first beta trial in August 2011, the system was opened up to the broader public in combination with a press release from CERN. This resulted in such a massive amount of new subscriptions that the system then in place could not handle it, resulting in crashes. A procedure was then introduced by which volunteers wishing to participate could sign up for participation codes which were issued incrementally. This gradually brought the number of connected participants up to around 2,500, while we were able to monitor and improve the scalability significantly. During its second phase, the participation-code restriction was removed, and the number of successfully connected hosts increased to about 6,500 machines by the end of the beta trial (with a noticeable spike around the CERN Press Release on Dec 13, 2011 concerning the possible hints of a Higgs boson; another significant spike was seen on July 4th 2012, when the discovery was confirmed).

The system is now fully operational, with several thousand volunteers donating CPU cycles on a daily basis, and individual volunteers already having generated billions of events each, cf. the *Test4Theory Leaderboard* reproduced in Fig. 12. At the time of writing, in June 2013, the number of volunteers who have completed at least one work package is 8,000, with a combined total of roughly 13,500 machines. The number of instantaneously connected machines fluctuates between 600–800.

These contributions have made a crucial difference in being able to generate the massive amounts of statistics required for the innumerable combinations of generators, tunes, and experimental analyses provided on MCPLOTS.

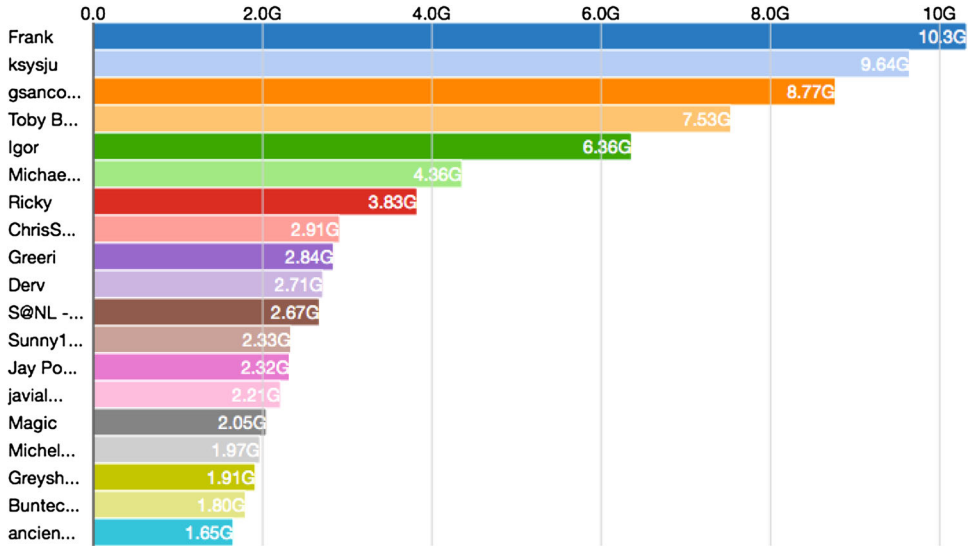


Fig. 12 TEST4THEORY Leaderboard (top contributors in number of events generated), as of May 2013. The up-to-date online version can be consulted at [40].

8 Conclusions

In this paper, we have provided an elementary user guide for the MCPLOTS site, and summarized its technical implementation. We intend the site to be broadly useful to the particle physics community, as a resource for MC validation and tuning, and as an explicit browsable repository of RIVET analyses. It also provides a possibility for the public to engage in the scientific process, via the LHC@HOME 2.0 project TEST4THEORY.

In the near future, we plan to extend the site by adding more possibilities for users to create their own plots and control how they look. We will also aim to provide event-generator authors with a pre-release validation service, in which a would-be new version of a generator or tune parameter card can be uploaded securely and a private version of the site made available on which the corresponding results can be compared with the standard set of generators and tunes.

Plans for future developments of TEST4THEORY include minimizing the size of the downloadable CERNVM image for TEST4THEORY (μCERNVM), a new delivery method for TEST4THEORY that would allow one to download, install, and run TEST4THEORY directly from a web browser (TEST4THEORY*Direct*), and the development of a new interactive citizen-science application based on the TEST4THEORY framework.

Note to users of MCPLOTS: we are of course very grateful if a reference to this work is included when using content from MCPLOTS, but even more importantly, we ask our users to please acknowledge the original sources for the data, analyses, physics models, and computer codes that are displayed on MCPLOTS. We have tried to make this as easy as possible, by including links to the original experimental papers together with each plot. Other references that may be appropriate, depending on the context, include HEPDATA [13], RIVET [14], MC generator manuals [20–26], and relevant physics and/or tuning papers.

As an example of good practice, and to ensure maximal clarity and reproducibility, all plots on MCPLOTS include explicit generator names and version numbers for all curves appearing on the plot, together with the relevant experimental reference. When combining an ME generator X with a (different) shower generator Y, both names and version numbers are shown explicitly. This is not only to give proper credit to the authors, but since the physics interpretation of the calculation depends on how it was performed.

Acknowledgments We are indebted to many members of the Monte Carlo community for their feedback and help with validations of the generators and analyses included on MCPLOTS. Many short-term students, visitors, and fellows have contributed with RIVET analyses and other content to the site, in particular D. Konstantinov (the MCPLOTS plotting tool), A. Pytel (server infrastructure and the MC validation view), S. Sen (forward LHC analyses in RIVET and on MCPLOTS, and implementation of the EPOS and PHOJET generators), and J. Winter (the $t\bar{t}$ MC analyses). We thank J. Katzy, for carefully reading this manuscript and giving feedback on the MCPLOTS site, M. Mangano, for his strong support of this project from its inception to completion and for help with the ALPGEN implementation, and T. Pierog, for help with the EPOS implementation. We express our profound admiration of the teams responsible for the development of RIVET, CERNVM, and LHC@HOME 2.0, without which the TEST4THEORY project would have been impossible. Thanks also to the many volunteers who have provided computing resources, especially to those who have taken active part in the alpha and beta testing phases of TEST4THEORY. We also thank the CERN PH-SFT group, in particular the Generator Services (GENSER) team, and the CERN IT Division. This work was supported in part by the LHC Physics Center at CERN (LPCC), by the Marie Curie research training network "MCnet" (contract number MRTN-CT-2006-035606), and by the National Science Foundation under Grant No. NSF PHY11-25915.

References

1. A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Höche et al., MCnet Collaboration, General-purpose event generators for LHC physics. Phys. Rep. **504**, 145–233 (2011). [arXiv:1101.2599]
2. P. Skands, Introduction to QCD, 1207.2389. Lectures given at TASI 2012
3. M.H. Seymour, M. Marx, Monte Carlo event generators, 1304.6677. Lectures given at 69th Scottish UniversitiesGÇÖ Summer School in Physics
4. A. Buckley, H. Höth, H. Lacker, H. Schulz, J.E. von Seggern, Systematic event generator tuning for the LHC. Eur. Phys. J. C65, 331–357 (2010). [arXiv:0907.2973][hep-h].
5. P.Z. Skands, Tuning Monte Carlo generators: the Perugia tunes. Phys. Rev. **D82**, 074018 (2010). [arXiv:hep-th/1005345]
6. R. Corke, T. Sjöstrand, Interleaved parton showers and tuning prospects. JHEP **1103**, 032 (2011). [arXiv:1011.1759]
7. S. Gieseke, C.A. Rohr, A. Siodmok, Tuning of the multiple parton interaction model in Herwig++ using early LHC data. In: Proceedings of MPI@LHC 2011, DESY-PROC-2012-03
8. H. Schulz, P. Skands, Energy scaling of minimum-bias tunes. Eur. Phys. J. **C71**, 1644 (2011). [arXiv:1103.3649]
9. ATLAS Collaboration, Summary of ATLAS Pythia 8 tunes, ATL-PHYS-PUB-2012-003, ATL-COM-PHYS-2012-738
10. J. Alcaraz Maestre et al., The SM and NLO multileg and SM MC working groups: summary report [arXiv:1203.6803]
11. G.P. Salam, Elements of QCD for hadron colliders, 1011.5131. Lectures given at ESHEP 2009
12. J. Butterworth, S. Butterworth, JetWeb: a WWW interface and database for Monte Carlo tuning and validation. Comput. Phys. Commun. **153**, 164–178 (2003). [arXiv:hep-ph/0210404]
13. A. Buckley, M. Whalley, HepData reloaded: reinventing the HEP data archive [1006.0517]. http://durpdg.dur.ac.uk/
14. A. Buckley, J. Butterworth, L. Lönnblad, D. Grellscheid, H. Höth et al., Rivet user manual. Comput. Phys. Commun. **184**, 2803–2819 (2013). [arXiv:1003.0694]
15. M. Dobbs, J.B. Hansen, The HepMC C++ Monte Carlo event record for High Energy Physics. Comput. Phys. Commun. **134**, 41–46 (2001)
16. M. Cacciari, G.P. Salam, Dispelling the N^3 myth for the k_t jet-finder. Phys. Lett. **B641**, 57–61 (2006) [arXiv:hep-ph/0512210]. http://fastjet.fr
17. M. Cacciari, G.P. Salam, G. Soyez, FastJet user manual. Eur. Phys. J. **C72**, 1896 (2012). [arXiv:1111.6097]
18. M. Whalley, D. Bourilkov, R. Group, The Les Houches accord PDFs (LHAPDF) and LHAGLUE [arXiv:hep-ph/0508110]
19. D. Bourilkov, R.C. Group, M.R. Whalley, LHAPDF: PDF use from the Tevatron to the LHC [arXiv:hep-ph/0605240]
20. M.L. Mangano, M. Moretti, F. Piccinini, R. Pittau, A.D. Polosa, ALPGEN, a generator for hard multiparton processes in hadronic collisions. JHEP **0307**, 001 (2003). [arXiv:hep-ph/0206293]
21. K. Werner, I. Karpenko, T. Pierog, M. Bleicher, K. Mikhailov, Event-by-event simulation of the three-dimensional hydrodynamic evolution from flux tube initial conditions in ultrarelativistic heavy ion collisions. Phys. Rev. **C82**, 044904 (2010). [arXiv:1004.0805]
22. M. Bähr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton et al., Herwig++ physics and manual. Eur. Phys. J. **C58**, 639–707 (2008). [arXiv:0803.0883]
23. F.W. Bopp, R. Engel, J. Ranft, Rapidity gaps and the PHOJET Monte Carlo [arXiv:hep-ph/9803437]
24. T. Sjöstrand, S. Mrenna, P.Z. Skands, PYTHIA 6.4 physics and manual. JHEP **0605**, 026 (2006). [arXiv:hep-ph/0603175]
25. T. Sjöstrand, S. Mrenna, P. Skands, A brief introduction to PYTHIA 8.1. Comput. Phys. Commun. **178**, 852–867 (2008). [arXiv:0710.3820]
26. T. Gleisberg, S. Höche, F. Krauss, M. Schonherr, S. Schumann et al., Event generation with SHERPA 1.1. JHEP **0902**, 007 (2009). [arXiv:0811.4622]
27. W. Giele, D. Kosower, P. Skands, Higher-order corrections to time-like jets. Phys. Rev. **D84**, 054003 (2011). [arXiv:1102.2126]
28. E. Boos, M. Dobbs, W. Giele, I. Hinchliffe, J. Huston, et al., Generic user process interface for event generators [arXiv:hep-ph/0109068]
29. J. Alwall, A. Ballestrero, P. Bartalini, S. Belov, E. Boos et al., A standard format for Les Houches event files. Comput. Phys. Commun. **176**, 300–304 (2007). [arXiv:hep-ph/0609017]
30. LHC@home 2.0. http://lhcathome2.cern.ch/
31. B. Segal, P. Buncic, D.G. Quintas, C. Aguado Sanchez, J. Blomer, et al., LHC cloud computing with CernVM, PoS ACAT2010, 004 (2010)
32. C. Aguado Sanchez, J. Blomer, P. Buncic, G. Chen, J. Ellis et al., Volunteer clouds and citizen cyberscience for LHC physics. J. Phys. Conf. Ser. **331**, 062022 (2011)
33. D. Lombraña Gonzalez, F. Grey, J. Blomer, P. Buncic, A. Harutyunyan, et al., Virtual machines and volunteer computing: experience from LHC@Home: Test4Theory project, PoS ISGC2012, 036 (2012)
34. P. Buncic, C. Aguado Sanchez, J. Blomer, L. Franco, A. Harutyunian, et al., CernVM: a virtual software appliance for LHC applications. J. Phys. Conf. Ser. **219**, 042003 (2010). http://cernvm.cern.ch
35. A. Harutyunyan, J. Blomer, P. Buncic, I. Charalampidis, F. Grey et al., CernVM Co-Pilot: an extensible framework for building scalable computing infrastructures on the cloud. J. Phys. Conf. Ser. **396**, 032054 (2012)
36. A. Harutyunyan, C. Aguado Sanchez, J. Blomer, P. Buncic, CernVM Co-Pilot: a framework for orchestrating virtual machines running applications of LHC experiments on the cloud. J. Phys. Conf. Ser. **331**, 062013 (2011)
37. The BOINC Project. http://boinc.berkeley.edu
38. N. Hoimyr, J. Blomer, P. Buncic, M. Giovannozzi, A. Gonzalez et al., BOINC service for volunteer cloud computing. J. Phys. Conf. Ser. **396**, 032057 (2012)
39. The Generator Services Project. http://sftweb.cern.ch/generators/
40. The Test4Theory Project. http://lhcathome2.cern.ch/about-test4theory
41. ALICE Collaboration, K. Aamodt et al., Charged-particle multiplicity measurement in proton-proton collisions at $\sqrt{s} = 7$ TeV with ALICE at LHC. Eur. Phys. J. **C68**, 345–354 (2010) [arXiv:1004.3514]
42. ATLAS Collaboration, G. Aad et al., Charged-particle multiplicities in pp interactions measured with the ATLAS detector at the LHC. New J. Phys. **13**, 053033 (2011) [arXiv:1012.5104]
43. ATLAS Collaboration, G. Aad et al., Measurement of underlying event characteristics using charged particles in pp collisions at $\sqrt{s} = 900GeV$ and 7 TeV with the ATLAS detector. Phys. Rev. **D83**, 112001 (2011) [arXiv:1012.0791]
44. B. Cooper, J. Katzy, M. Mangano, A. Messina, L. Mijovic et al., Importance of a consistent choice of alpha(s) in the matching of AlpGen and Pythia. Eur. Phys. J. **C72**, 2078 (2012). [arXiv:1109.5295]
45. OPAL Collaboration, K. Ackerstaff et al., Measurements of flavor dependent fragmentation functions in Z0 to q anti-q events. Eur. Phys. J. **C7**, 369–381 (1999) [arXiv:hep-ex/9807004]

46. CDF Collaboration, D. Acosta et al., Study of jet shapes in inclusive jet production in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV. Phys. Rev. **D71**, 112002 (2005) [arXiv:hep-ex/0505013]
47. M.L. Mangano, M. Moretti, F. Piccinini, M. Treccani, Matching matrix elements and shower evolution for top-quark production in hadronic collisions. JHEP **0701**, 013 (2007). [arXiv:hep-ph/0611129]
48. The AGILe Project. http://agile.hepforge.org
49. ATLAS Collaboration, G. Aad et al., Measurement of the production cross section for W^- bosons in association with jets in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector. Phys. Lett. **B698**, 325–345 (2011) [arXiv:1012.5382]
50. CMS Collaboration, S. Chatrchyan et al., Study of the dijet mass spectrum in $pp \to W+$ jets events at $\sqrt{s} = 7$ TeV. Phys. Rev. Lett. **109**, 251801 (2012) [arXiv:1208.3477]
51. C.M.S. Collaboration, S. Chatrchyan et al., Event shapes and azimuthal correlations in Z + jets events in pp collisions at $\sqrt{s} = 7$ TeV. Phys. Lett. **B722**, 238–261 (2013). [arXiv:1301.1646]
52. ATLAS Collaboration, G. Aad et al., Measurement of kT splitting scales in $W^- \to \ell\nu$ events at $\sqrt{(}s) = 7$ TeV with the ATLAS detector. Eur. Phys. J. **C73**, 2432 (2013) [arXiv:1302.1415]
53. ATLAS Collaboration, G. Aad et al., Measurement of the cross-section for W boson production in association with b-jets in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector. JHEP **1306**, 084 (2013) [arXiv:1302.2929]
54. CMS Collaboration, S. Chatrchyan et al., Studies of jet mass in dijet and W/Z + jet events. JHEP **1305**, 090 (2013) [arXiv:1303.4811]
55. ATLAS Collaboration, G. Aad et al., Measurement of the production cross section of jets in association with a Z boson in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector. JHEP **1307**, 032 (2013) [arXiv:1304.7098]
56. CMS Collaboration, S. Chatrchyan et al., CMS Collaboration Measurement of the hadronic activity in events with a Z and two jets and extraction of the cross section for the electroweak production of a Z with two jets in pp collisions at $\sqrt{s} = 7$ TeV. JHEP **1310**, 101 (2013) [arXiv:1305.7389]
57. A. Cafarella, C.G. Papadopoulos, M. Worek, Helac-Phegas: a generator for all parton level processes. Comput. Phys. Commun. **180**, 1941–1955 (2009). [arXiv:0710.2427]
58. J. Alwall, P. Demin, S. de Visscher, R. Frederix, M. Herquet et al., MadGraph/MadEvent v4: the new web generation. JHEP **0709**, 028 (2007). [arXiv:0706.2334]
59. J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer, MadGraph 5: going beyond. JHEP **1106**, 128 (2011). [arXiv:1106.0522]
60. W. Kilian, T. Ohl, J. Reuter, WHIZARD: simulating multi-particle processes at LHC and ILC. Eur. Phys. J. **C71**, 1742 (2011). [arXiv:0708.4233]
61. G. Robert-Demolaize, R. Assmann, S. Redaelli, F. Schmidt, A new version of SixTrack with collimation and aperture interface. Conf. Proc. **C0505161**, 4084 (2005)