

THE CMS SIMULATION VALIDATION SUITE

S. Abdullin, H. Cheung, D. Elvira*, J. Yarba, FNAL, Batavia, IL 60510, USA
P. Arce, CIEMAT, Madrid, Spain
S. Banerjee, Tata Institute of Fundamental Research, Mumbai, India
X. Ding, Nanjing University, China
A. Karchilava, M. Strang, SUNY Buffalo, Buffalo, NY, USA
X. Huang, Universidad de Pto. Rico, Mayaguez, Puerto Rico
L. Shabalina, UIC, Chicago, IL, USA

Abstract

We describe the characteristics and functionalities of the CMS Simulation Validation Suite (SVS), developed under the OSCAR framework. The purpose of this software system is to validate each new version of the CMS simulation software, comparing current values of simulation quantities with those of reference files. A natural extension is to incorporate generator, digis, and reconstructed objects to an extended Physics Software Validation Suite (PSVS). The PSVS is currently under development in the CMSSW framework, based on the OSCAR SVS.

INTRODUCTION

The purpose of the Simulation Validations Suite (SVS) is to validate each new version of the CMS simulation software, comparing current values of simulation quantities with those of reference files. Validation variables are either hits or hit derived quantities, as well as geometry, magnetic field, or Geant4 [1] objects. The SVS described in this note was developed and tested under the OSCAR simulation framework. An expanded Physics Software Validation Suite (PSVS) will also include objects at the generation, digitization, mixing, and reconstruction levels. The PSVS is currently under development in the CMSSW framework.

GENERAL DESCRIPTION

The validation process is divided into three stages to make the process clear, and the software easy to maintain:

- Individual detector system, geometry, or field packages do “on the fly” analysis on a set of validation samples, producing ROOT [2] output files. The initial packages include: `SimG4TrackerValidation`, `SimG4EcalValidation`, `SimG4HcalValidation`, `SimG4MuonValidation`, `SimG4FieldValidation`, and `SimG4GeomValidation`. Each package consists of one or more tests associated with sub-detectors within a detector system, barrel/endcap and pre-shower in the case of the Ecal. Different tests have been defined and configured in some sub-detectors to

use either low-level or high-level simulation information in the validation. The same test may be run many times under different conditions, or samples. ROOT trees with pre-processed information are written into ROOT browsable files. This information typically includes bare hit quantities or complex ones derived from hits or Geant4 objects.

- Macros are run on the ROOT trees and process information into validation objects, such as numbers in ASCII files and/or histograms.
- OVAL [3] is the integration tool, used for launching the Suite from the `SimG4Validation` directory. Histogram comparison tools, such as the Statistical Tool Kit [4] or ROOT are used to compare the “current” histograms values with those stored in reference files. The differences are stored in ASCII files.
- OVAL is used to find differences between the ASCII files with current difference information and reference ASCII files.

VALIDATION PROCEDURE

The CMS production group would run the integrated Suite before each release. A certification procedure will be developed to analyze the output OVAL files and take corrective measures if necessary.

SUB-DETECTOR PACKAGE DESCRIPTION

Tracker

Description The `SimG4TrackerValidation` package is intended to test the simulation of the tracking system comprised of the Silicon Strip Tracker and the Pixel Detector System. The former consists of the Tracker Inner Barrel (TIB), the Tracker Outer Barrel (TOB), the Tracker Inner Disks (TID), and the Tracker End Cap (TEC), while the latter includes the Pixel Barrel and the Pixel End Cap. (The Pixel End Cap is also referred to as the Forward Pixel Detector.) The tests are performed for each subsystem separately due to their different geometries.

* daniel@fnal.gov

The `SimG4TrackerValidation` package accesses collections of hits belonging to different subsystems through the `TrackerHitsObject`. It produces a ROOT tree with the simulated hit information.

Simple analysis software is implemented in ROOT macro scripts to produce histograms associated with the final quantities to validate. In the `SimG4TrackerValidation` package, samples may be generated using a single particle gun, or read from an HBOOK file in HepEVT format using Ntuple reader software. The two available tracker validation options are controlled from the `OvalFile` configuration file.

ROOT Tree Content One assistant class, `SimHitTrackerTree` provides the interface to ROOT. The information stored in the ROOT tree is divided into two branches:

1. Global Information

- Event Number
- Run Number
- Total Number of Incident Particles
- Incident Particle type
- Incident Particle Energy
- Incident Particle Momentum
- Incident Particle η
- Incident Particle ϕ
- Number of vertices
- Vertex coordinates
- Total Number of Hits

2. Hit information

- Subsystem type
- ID of the detector unit (in Geant internal numbering)
- Hit position (entry and exit point coordinates in the local system, local position and direction)
- Deposited Energy of Hit
- ID of the track that produced a hit
- Process and particle momentum and type
- Time-of-flight

Validation Quantities The comparison tests are performed on the following quantities, constructed from the ROOT tree leaves:

- Sample: Single muons, electrons or pions with $p_T = 15$ GeV, 1500 events in 12 bins of η in the range $-3 < \eta < 3$
- Quantities
 - Energy deposition
 - Distrib. of track entry and exit points in x, y, z
 - Distrib. of local x, y track entry point coordinates
 - Number of hits in each subsystem

Test variables are plotted separately for each subsystem. Examples of energy loss for muons traversing different tracker sub-systems are presented in Figs.1.

Run Configurations A single package performs the validation analysis for all the tracker sub-systems. The validation sample covers six η bins in the $|\eta| < 3$ range. The `test` subdirectory under `SimG4TrackerValidation` contains the `OvalFile` and the C-shell scripts to execute the package and tests. The scripts are controlled from the `SimG4Validation` area via the OVAL tool.

Validation Tests OVAL is also used to detect differences between current and reference values stored in two separate histogram files. Statistical tools available in ROOT - `UnBinned KSTest` and `Binned Chi2Test` - are used to compare the sample and the reference histograms.

Electromagnetic Calorimeter

Description The `SimG4EcalValidation` package is associated with the Electromagnetic Calorimeter (ECAL), both the barrel and the end-caps, as well as with the Pre-Shower sub-detector. The two tests implemented in `SimG4EcalValidation` are `Ecalorimeter` and `Pre-Shower`. Both tests perform a preliminary on-the-fly analysis of basic simulation quantities, hits and `G4Step` objects, and store the results in a ROOT tree contained in a ROOT file. ROOT macro scripts contain the analysis to produce histograms associated with the final quantities to validate. In the `SimG4EcalValidation` package, samples may be generated using the single particle gun, or read from an HBOOK file in HepEVT format using an Ntuple reader. The two available options are controlled from a C-Shell script file.

ROOT Tree Content One assistant class, `SimHitEcalTree` provides the interface to ROOT. The information stored in the ROOT tree is divided into three branches:

1. Global Information

- Event Number
- Run Number
- Total Number of Incident Particles
- Particle Type of Every Incident Particle
- Energy of Every Incident Particle
- Momentum of Every Incident Particle
- Vertex coordinates
- Total Number of Hits in Barrel Calorimeter
- Total Number of Hits in EndCap Calorimeter
- Total Energy deposited in Barrel Calorimeter
- Total Energy deposited in EndCap Calorimeter
- Energy deposited in 1x1 crystal cluster
- Energy deposited in 2x2 crystal cluster
- Energy deposited in 3x3 crystal cluster
- Energy deposited in 4x4 crystal cluster
- Energy deposited in 5x5 crystal cluster

Energy deposited by EM Particles
Energy deposited by Hadrons Particles

2. Hit information

Hit position: etaHit, phiHit
Hit energy: energyEMHit, energyHadrHit
Global time

3. Step Information

Energy in every X0 (radiation length): EX0[25].

Validation Quantities The comparison tests are performed on the following quantities, constructed from the ROOT tree leaves:

- Lower-Level Validation for ECalorimeter
 - Sample: 30 GeV photons(2000 Events)
 - Quantities
 - Plot the occupancy (eta versus phi)
 - Plot 1-D histogram of E1, E4, E9, E16, E25
 - Plot 1-D histogram of Ratio: E1/E4, E4/E9, E9/E16, E16/E25, E1/E25, E9/E25
 - Plot the percentage of energy deposited in Barrel and End-Cap
 - Plot the longitudinal shower development, fixed Energy
- Higher-Level Validation for ECalorimeter
 - Sample: 10 GeV, 20 GeV, 30 GeV, 40 GeV, and 50 GeV photons (2000 events)
 - Quantities
 - Plot the E25 resolution versus the energy of incident particle
 - Plot the longitudinal development of Shower, energy scan
- Lower-Level Validation for Pre-Shower
 - Sample: 30 GeV photons (2000 events)
 - Quantities
 - Plot the occupancy (eta versus phi)
 - Plot 1-D histogram of E1, E4, E9, E16, E25
 - Plot 1-D histogram of Ratio: E1/E4, E4/E9, E9/E16, E16/E25, E1/E25, E9/E25
 - Plot the percentage of energy deposited in Barrel, End-Cap, and Pre-Shower
 - Plot the longitudinal shower development, fixed energy.
- Higher-Level Validation for Pre-Shower
 - Sample: 10 GeV, 20 GeV, 30 GeV, 40 GeV, and 50 GeV photons (2000 events)

– Quantities

Plot the E25 resolution versus the energy of the incident particle
Plot the longitudinal development of the shower for different energies
Plot the coefficient of Lead absorption versus the energy of incident particle

Run Configurations The validation is performed currently at two levels for each of the ECAL sub-systems: barrel/end-caps, and pre-shower detectors. This two levels are the Low-Level and the High-Level modes. These four possible tasks are executed from a set of C-shell scripts that, in turn, are configured and controlled via the OVAL tool. The `OvalFile` and the two C-shell scripts, `EcalTest.csh` and `PreShowerTest.csh`, are located in the `test` subdirectory under `SimG4EcalValidation`. The scripts contain parameters such as the number of jobs to be run and the location of input data, as well as commands to rename the root files, and compare between current and reference histograms.

Validation Tests In addition to the role of controlling the tests, OVAL is also used to detect differences between current and reference values stored in a set of ASCII files. The Statistics Tool Kit is used to check the consistency between the contents of current and reference histograms. The class `SmartTestHisto` is introduced to perform this task, by reading the data stored in the reference ROOT file and the current ROOT file. If the test requires binned data, histograms are created from the ROOT tree information, otherwise, unbinned data are stored in vectors. In both cases, the Statistics Tool Kit classes are used to perform the comparison and output the results to the OVAL log files. As currently implemented, the `SmartTestHisto` class is used as an alternate option to the ROOT macro analysis. In the example provided under the `SimG4EcalValidation/test` directory, `Compare.cc`, two methods, `compareWithChi2Test()` and `compare2DWithChi2Test()`, are used to compare the current histograms with the reference histograms. Currently, all the lower-level quantities (as given in 3.2.3) are tested against reference samples using the `Chi2Test` method.

Finally, all the results are saved to two postscript files. Examples of these are shown in Fig.2. The 1-D histograms from the current and the reference samples are drawn in the same plot to aid in evaluating the differences between them, while the 2-D histograms, such as the longitudinal profile and occupancy, are drawn in two plots separately. Blue is used for the reference data and red for the current data. They were generated with the same OSCAR version but different initial seeds. PV stands for the probability value calculated with the `Chi2Test` method.

Hadronic Calorimeter

Description The `SimG4HcalValidation` package collects different type of information about HCAL simulated hits, using both `EndOfEvent` and `G4Step` Geant4 observers. It also performs some NxN cells cluster analysis and even cluster finding with a cone algorithm. The collected information is stored in a ROOT tree. A csh script `job_ntuple.csh` executes the package, plots the histograms, and compares some of them with those in the reference file by means of a ROOT macro. The input data sample contains 1000 single pions with $p=30$ GeV shot at the center of the $(\eta, \varphi) = (4, 4)$ tower.

ROOT Tree Content The ROOT tree is divided into several branches. There is a switch which enables the information to be stored at three levels of complexity. These three levels are denoted as L1, L2, L3. The most essential primary information is collected in the block of `CaloSimHit` information (L1):

nHits - number of hits
layerHits - number of HCAL layer to which the hit belongs
etaHits - hit η
phiHits - hit φ
eHits - hit energy
tHits - hit time
idHits - hit ID number
jitterHits - hit jitter with respect to the expected time of arrival.

The L2 branch includes a block of information on an $R=0.5$ cone or 7×7 tower clusters, transverse NxN cluster profiles, energy deposited in the HCAL scintillator layers and depths, and energy in the HF fibers. The L3 branch contains jet information, reconstructed at the hit level.

Validation Quantities The validation is performed for a sub-set of the more than 70 histograms previously described. These are compared with those in a reference file using the χ^2 test available in ROOT. The validation histograms include the energy deposition in the first 6 scintillator layers closest to the ECAL, the time distribution of all hits, and the energy-weighted time distribution of hits in a 7×7 ECAL+HCAL tower matrix defined around the pion entry point. Also included in the comparison are the distributions of the number of hits in the ECAL and HCAL. All the remaining histograms are not validated but contain useful information which might be required to investigate discrepancies unveiled by the tests. Figure 3 illustrates on one of the comparison tests currently implemented: the energy deposited in the HCAL scintillator just behind the ECAL.

Run Configurations In the test area, the default geometry configuration, `OSCARconfiguration.xml`, includes only ECAL and HCAL in the simulation. In the cluster

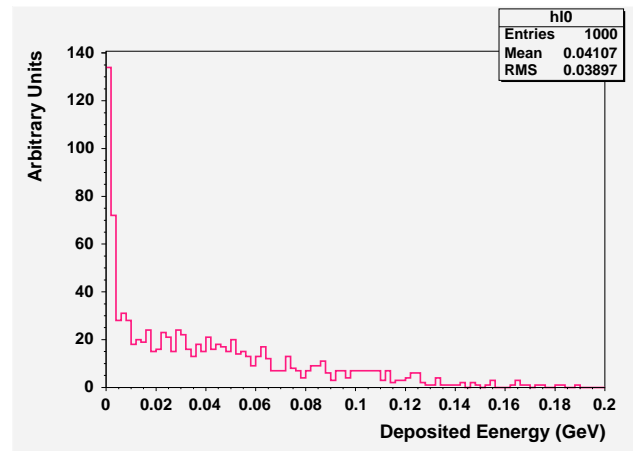


Figure 3: Energy deposition in layer 0 of the HB by a pion shower.

analysis there is a choice to take into account both ECAL and HCAL energy, or only the HCAL one (default). The ECAL energy is always taken as it is, while the HCAL energy for cluster analysis can be taken as either the original calorimeter hit energy (default), or the energy corrected by the same sampling factors as in the digitization code. The package operator defines the cone size of the jet finder, as well as a set of cuts on particle energy and direction.

Validation Tests The `0valFile` in the `SimG4HcalValidation/test` directory controls the execution of the validation package. By default, the validation level is set to `SimG4HcalValidation:InfoLevel=2` in the `job_ntuple.csh` script.

Geometry

Description The `SimG4GeomValidation` package writes out the number of volumes and materials for comparing them with a reference version using the OVAL tool. It also tracks particles with neither physics nor magnetic field and compares the total number of radiation lengths traversed until the end of the detector event-by-event. If desired, it also constructs a few histograms and a ROOT tree with material budget information (number of radiation lengths), and a text file with accumulated material budget step-by-step.

Validation quantities There are two different kinds of variables printed for testing with the OVAL tool:

- The geometry summary, that is the total number of different materials in the simulation, and the total number of logical volumes, physical volumes and touchables (copies of each volume).
- The number of radiation lengths after traversing all the detector. The events are read from an ntuple that contains one thousand muons of energy 10 TeV with random η in the range $-5.0 < \eta < 5.0$ and random

ϕ , and starting at a displaced vertex ($x = 8.3, y = 1.5, z = 6.3$) mm, to avoid as much as possible tracking along volume boundaries. In fact these muons behave as *geantinos* (non-interacting fictitious particles), because there is no magnetic field, and all the physics processes except transportation are switched off.

Material budget information As well as running the test, the operator may choose to produce output files with either histograms, a ROOT tree, or a text file for help in understanding the differences with respect to the reference version.

This histogram file would include:

1D profile histogram of material budget vs η
1D histo η
1D profile histogram of material budget vs ϕ
1D histo ϕ
2D profile histogram of material budget vs η and ϕ
1D histo η vs ϕ .

The ROOT tree, for each track includes:

MB: Total material budget (sum of step length divided by radiation length for all steps)
ETA: η direction at start of track
PHI: ϕ direction at start of track.

For each step within the track (if this option is selected):

DELTAMB: accumulated material budget
X: X position
Y: Y position
Z: Z position
VOLUID: volume ID (constructed in alphabetical order)
MATEID: material ID (constructed in alphabetical order).

In the text file, there is a first line for each track:

Track number
Eta direction at start of track
Phi direction at start of track.

There is also step-by-step information about the volume traversed and the accumulated material budget:

Accumulated track length (mm)
Volume name
Volume copy number
Accumulated material budget (Number of radiation lengths)
Material radiation length (rad. length / mm),

And, at the end of the track:

Accumulated material budget.

Run configuration The execution of the test is controlled via OVAL and is configured entirely within the `OvalFile` located in the `test` subdirectory under `SimG4GeometryValidation`. It compares the current values with those in the reference file, `testGeometry.ref`. Several parameters control the different output choices in the form of histograms, a ROOT tree or a text file.

Field

Description The `SimG4FieldValidation` package checks the tracking in the CMS magnetic field. The test compares the deviation at the end of the track in position and direction.

Validation quantities The validation sample consists of single muon events in four groups of 100 events with different energies: 1, 10, 100, 1000 GeV. The muons behave as charged *geantinos*, given that the physics processes are switched off.

The deviation of these tracks after traversing all CMS is compared with a reference version. This deviation is calculated in two directions: one perpendicular to the initial track direction and to z, ϕ direction, and the other one perpendicular to this first one and the initial track direction. Also the change in momentum projected in the above mentioned two directions is checked, as well as the change in kinetic energy.

Run configuration The execution of the test is controlled via OVAL and is configured entirely within the `OvalFile` in the `test` subdirectory under `SimG4FieldValidation`. Current values of the quantities in `test` are compared with the ones in the reference file `testFieldProp.ref`.

Muon System

Description The `SimG4MuonValidation` test checks the muon physics in Geant4, by using 100 GeV muons. It writes out ROOT histograms of the relevant quantities and compares these histograms with those from a reference file using the OVAL tool.

Validation quantities The test compares all the quantities that are relevant to high energy muon physics in Geant4. The validation sample consists of 1000 muon events with $p=100$ GeV/c. The secondary particles are killed before being tracked.

For each muon the package fills the following histograms:

Energy lost
Energy deposited
Deviation in position
Deviation in angle (deg)

Number of tracking steps.

For each type of muon process, i.e. ionization, bremsstrahlung, e^+e^- production, muon nuclear interaction, decay and capture at rest, it makes histograms of the quantities:

Total energy taken by secondaries
Energy of secondary particles
Secondary track angle with respect to muon track.

Run configuration The execution of the test is controlled via OVAL and is configured entirely within the `OvalFile` in the `test` subdirectory under `SimG4FieldValidation`. Current values of the quantities in `test` are compared with the ones in the reference file `testFieldProp.ref`.

Validation tests The histograms are compared with the reference ones at the end of job using the Statistics Toolkit. This tool performs a χ^2 comparison and prints the probability to obtain that χ^2 value or higher. This probability value is compared with the value of a reference file, `testMuonPhysics.ref` using the OVAL tool.

The plot in Fig. 4 is an example of the comparison plots used in the validation process, between the current and the reference samples.

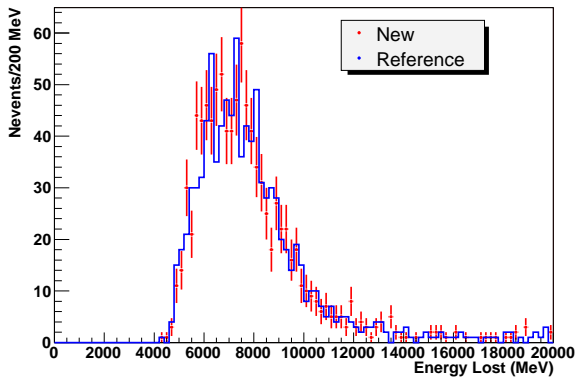


Figure 4: Energy lost for 100 GeV muons traversing all the CMS detector.

Global Validation

Description The `SimG4ValidationGlobal` package is not part of the SVS which runs under the OSCAR framework. It is under development within the CMSSW framework, and is intended to test the simulation of the entire detector (all sub-detectors in parallel) for expected types of physics events with an active magnetic field. It is designed to allow reproduction of the plots from the ORCA package `ExSimHitsStatistics` within the new framework.

The package makes use of the `GlobalValidation` class which acts as an event producer within the framework adding the relevant simulated information from each sub-detector to the `PGlobalSimHit` object. This object is then exported as a branch to the POOL output file, `simevent.root`.

ROOT macro scripts will then be used to access the root file and create the relevant histograms. Validation comparisons with respect to reference plots will then be performed in a manner similar to the existing validation packages.

ROOT Tree Content In addition to the standard CMSSW output, the `PGlobalSimHit` object is stored in a unique branch of the `simevent.root` output file. This object contains:

1. Monte Carlo

- Number of vertices from Geant4
- Number of tracks from Geant4
- Number of raw generated particles
- Position (x,y,z) of each Geant4 vertex
- p_T of each Geant4 track
- Energy of each Geant4 track

2. Electromagnetic Calorimeter

- Number of Ecal hits
- Energy of each Ecal hit
- Time of flight for each Ecal hit
- Global ϕ of each Ecal hit
- Global η of each Ecal hit
- Number of pre-shower Hits
- Energy of each pre-shower hit
- Time of flight for each pre-shower hit
- Global ϕ of each pre-shower hit
- Global η of each pre-shower hit

3. Hadronic Calorimeter

- Number of hits
- Energy of each hit
- Time of flight for each hit
- Global ϕ of each hit
- Global η of each hit

4. Tracker

- Number of Pixel hits
- Global ϕ of each Pixel hit
- Global η of each Pixel hit
- Time of flight of forward Pixel hits
- Time of flight of barrel Pixel hits
- Global R of barrel Pixel hits
- Global Z of forward Pixel hits
- Number of Silicon hits
- Global ϕ of each Silicon hit
- Global η of each Silicon hit

Time of flight of forward Silicon hits
Time of flight of barrel Silicon hits
Global R of barrel Silicon hits
Global Z of forward Silicon hits

5. Muon

Number of hits
Global ϕ of hits
Global η of hits
Time of flight for DT hits
Global R of DT hits
Time of flight for CSC hits
Global Z of CSC hits
Time of flight for RPC barrel hits
Global R of RPC barrel hits
Time of flight for RPC forward hits
Global Z of RPC barrel hits

Validation Quantities Validation studies will be performed on a subset of these variables as best determined by future studies.

Run configuration The creation of the initial root file is handled as a typical CMSSW job, using the `runP.cfg` configuration file to set the initial data source and other running conditions for the simulation. The file used for this package can be found in the `SimG4Validation/Global/test` subdirectory of the CMSSW package. The code is run using the `cmsRun -p runP.cfg` command.

ROOT macros will be written which create the full suite of plots as found in the old `ExSimHitsStatistics` package as well as the plots of interest for the current validation.

In the future, the validation will be performed using the OVAL tool as with the existing simulation packages.

Validation tests The validation is performed using the statistics tools available in ROOT including the Kolmogorov-Smirnov test and the χ^2 test.

INTEGRATION AND OPTIMIZATION

The tool of choice for the integration of the SVS is OVAL, which is a testing tool created to help developers to detect unexpected changes in the behavior of their software. Detailed information on the OVAL project may be found in Ref. [3].

OVAL includes the following features:

- Executes shell commands and scripts.
- Compiles and links validation programs.
- Runs the test programs and scripts within a specified Unix environment.

- Creates on the fly the needed configuration parameter sets or uses pre-defined external configuration files.
- Executes recursively a user-specified set of targets, including a hierarchy of subdirectories.
- Allows modular, configurable sub-tests within one subdirectory.

The SVS is executed from the `/SimG4Validation` base directory.

Comparison Test Optimization Studies

Conclusions based on preliminary studies [5] show that the KS test using binned data can be just as powerful in a discrimination test as using unbinned data, but the p-value must be set lower when using binned data. The second preliminary conclusion is that the KS test using binned data is preferred over a χ^2 test as it is a more power discrimination tool.

EXTERNAL TOOLS

OVAL [3] is the main tool used in SVS to control its execution and perform comparisons. This tool enables to compare values in an ASCII file with those in a reference file. The user defines a set of tolerances for the quantities of interest, and OVAL returns a “DIFF” message in an output file when the difference is outside the tolerance range. The evaluation of the usage of OVAL in the SVS project is included in Section .

Tools other than OVAL are being tested for histogram comparison. The two obvious options are the Statistical Tool Kit [4], and the tests available within ROOT [2].

The Statistical Tool Kit is a general purpose tool for statistical analysis, which follows the tool kit philosophy, and is designed to be flexible and extensible. It is component-based, allowing re-use and integration in different frameworks, and has no dependence on any specific analysis tool. The user layer bridges the core statistical component and the user’s analysis, which may be based on different frameworks. The user layer shields the user from the underlying algorithms and design, and only deal with the user’s analysis objects and the choice of comparison algorithm. The kit currently supports a very complete statistics software suite for comparison of two binned and unbinned distributions, including Anderson-Darling, Chi-squared, Fisz-Cramer-von Mises, Tiku tests (binned), Creamer-von Mises, Goodman, Kolmogorov-Smirnov, Kuiper, and Tiku tests (unbinned). The Statistical Tool Kit is currently being used by the Geant4 physics validation group within the LCG Simulation Validation project. The current version of SVS uses the Statistical Tool Kit in the Muon and Ecal packages.

ROOT is an analysis framework widely used in running experiments. ROOT is not a software project that specializes in statistical tools, but the Chi-square and Kolmogorov-Smirnov tests are made available through methods of its histogram class.

While the Statistical Tool Kit is a project specializing on statistical tools and provides a large variety of goodness-of-fit tests, ROOT satisfies the current needs of the SVS, and allows one to perform the tests within the same macro file used to analyze the ROOT trees produced by the validation packages. The most complete solution in the long term could be to use the Statistical Tool Kit as an external library loaded into ROOT. This option would allow the Simulation Validation Group, and CMS in general, to benefit from the large variety of tests in the tool kit, while keeping the analysis within the ROOT framework.

EXTENSIONS

The Suite described in this note validates simulation quantities from Geant4 or hit information. A natural extension is to incorporate generator, digis, and reconstructed objects to an extended Physics Software Validation Suite (PSVS). The PSVS is currently under development in CMS.

ACKNOWLEDGEMENTS

We wish to thank members of the CMS Software and PRS groups for their input in defining the validation quantities. We are also grateful to David Chamont, as well as to Maria Grazia Pia, and Andreas Pfeiffer for valuable discussions on the capabilities of OVAL and the Statistical Tool Kit.

REFERENCES

- [1] NIM A 506 (2003) 250-303
Geant4 Developments and Applications, J. Allison et al.,
IEEE Transactions in Nuclear Science 53, February, 2006 (in
press). <http://geant4.web.cern.ch/geant4/>.
- [2] <http://root.cern.ch/>.
- [3] <http://oval.in2p3.fr/>.
- [4] <http://www.ge.infn.it/statisticaltoolkit/>
- [5] Harry W. K. Cheung and Xiaoping Ding, CMS Note in preparation.

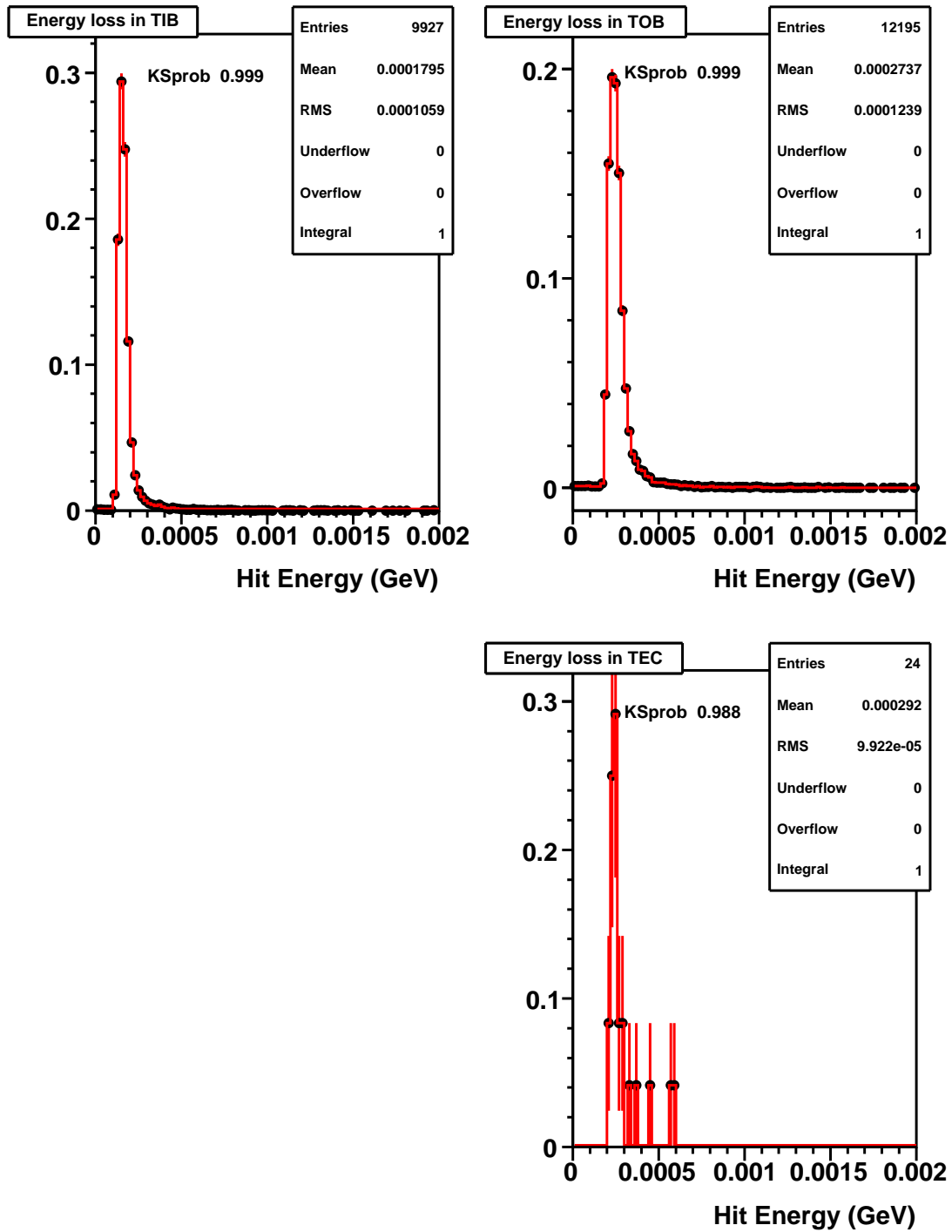


Figure 1: Energy loss for muons traversing different sub-systems of the silicon tracker detector. The Kolmogorov-Smirnov tests are performed on identical samples, current (solid line) and reference (symbols).

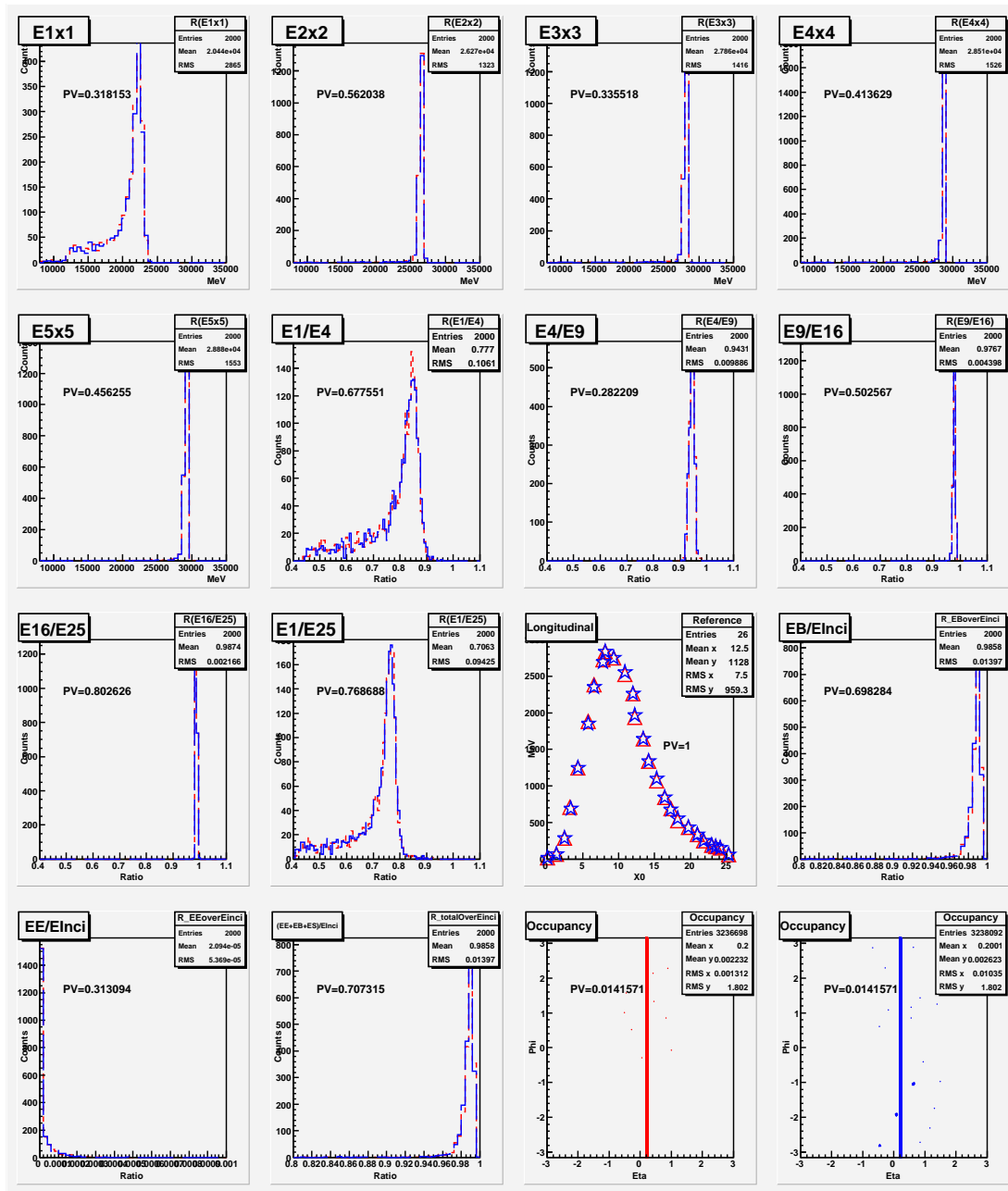


Figure 2: Comparison between current (red) and reference (blue) validation histograms for photons traversing the ECAL barrel detector.