Argonne NATIONAL LABORATORY

ANL-HEP-TP-09-110

ATLAS NOTE
ATL-COM-PHYS-2009

High Energy Physics Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Availability of This Report

This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL-HEP-TP-09-110

ATLAS NOTE
ATL-COM-PHYS-2009

by
J. Boomsma and S. Chekanov
High Energy Physics Division, Argonne National Laboratory

November 20, 2009

ATLAS NOTE

ATL-COM-PHYS-2009-xxx

November 20, 2009

General searches for new particles

J. Boomsma and S. Chekanov

HEP division, Argonne National Laboratory, 9700 S. Cass Av., Argonne, IL 60439, USA

Abstract

The program InvMass has been developed to perform a general model-independent search for new particles using the ATLAS detector at the Large Hadron Collider (LHC), a proton-proton collider at CERN. The search is performed by examining statistically significant variations from the Standard Model predictions in exclusive event classes classified according to the number of identified objects. The program, called InvMass, finds all relevant particle groups identified with the ATLAS detector and analyzes their production rates, invariant masses and the total transverse momenta. The generic code of InvMass can easily be adapted for any particle types identified with the ATLAS detector. Several benchmark tests are presented.

1 Introduction

The results of pp collisions at the LHC will allow physicists to search for new particles that can corroborate or disprove current theories such as the Standard Model (SM). The ATLAS detector is a general-purpose detector that can identify many types of known particles which then can be used for searches when the LHC begins its operation.

One promising path to discoveries at the LHC is through general, model-independent searches in which events can be classified in exclusive classes according to the number of identified high-ET objects. Such event classes can be analyzed by reconstructing event rates, invariant masses and other event characteristics. A general search analyzes all the collected data and attempts to find new parent particles over the specified kinematic range of that data. Unlike a model-based approach, such signature-based approach is not sensitive to particular models.

To handle the scale of general searches, a computer program has to be developed. Since it is possible to identify many different particles using the ATLAS detector, and since parent particles can decay into any number of daughter particles, the number of histograms to create and analyze becomes very large. For example, if it is possible to identify 7 types of high-ET objects and a search is made for a parent object which decays into 2, 3, or 4 daughter particles, 322 unique daughter groups must be investigated. If the search includes groups of 5 daughter particles as well, the number increases to 784 groups. To manually code every possible outcome would be difficult and time consuming, but a computer program can determine each possible grouping easily.

The idea for a general search is not new. A general search was preformed using the H1 detector at HERA, an electron-proton collider. The HERA search [1] broke identified particles and jets into unique groups and analyzed the transverse momenta (PT), invariant mass (IM), and event distribution of the groups. The same type of search was performed using the Tevatron collider at Fermi National Laboratory. The searches were based on the Vista program (examines populations and kinematic features of the high-pT data), the Bump Hunter [2] (for resonances in invariant mass combinations) and the Sleuth program developed for studies of excesses at high sum-pT [2, 3]. However, neither search provided conclusive results.

The InvMass discussed in this node is a ROOT/C++ based program designed for a general search using the ATLAS detector. InvMass separates high-ET identified objects found in each event into groups of daughter particles. Using those groups, it fills corresponding histograms with information about the groups of identified particles; their event rate, IM and total transverse momenta. InvMass can also automatically analyze the histograms and output a summary of the peaks that may be found.

2 Program structure

As stated, InvMass is a program that has been written at ANL to execute a general search using high-ET identified objects. It makes use of the ROOT/C++ framework for data analysis, and its architecture is compartmentalized so that the code is easy to understand and edit. However, InvMass is the very last step in the data analysis process. It requires an Athena based program to create the ROOT ntuple files it uses.

At present, InvMass uses the PromptGamma athena program [4] which converts AOD/DPD files into compact ROOT ntuples. The ntuples organize the data by particle type and by individual particles. The data to be stored includes the momentum, energy, and charge of each particle found in the events. Once the ntuples are created by the Athena program, InvMass can be used. InvMass will execute a general search, and by analyzing the ROOT ntuples it will output histograms which will be used to identify parent particles. Once those histograms are created, they can be analyzed further by other programs if desired. A schematic representation of the general-search program is shown in Figure 1.

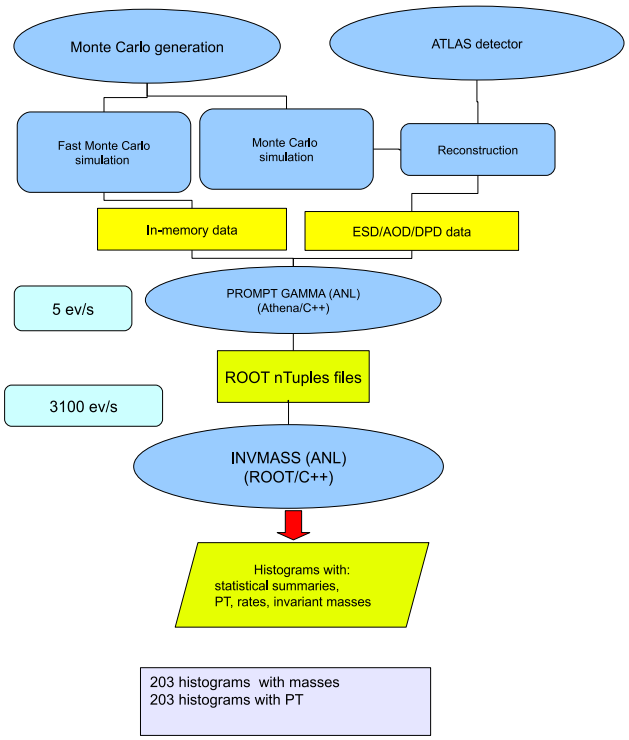

Figure 1: A schematic representation of the InvMass reconstruction. For 6 object types (for example, for jet, γ, μ, missET, e, τ), there are 21 combinations for 2-body decays, 56 combinations for 3 decays, 126 combinations for 4 decays. In total, this gives 406 histograms with for invariant masses and transverse-momentum distributions.

There are four steps in InvMass that are used to execute the search. An overview of these steps will shed light on the overall architecture of InvMass and how it works. Figure 2 may be a helpful visual representation of the description. The first step is the file loop. InvMass is written such that it can read multiple ROOT files each time that it is run. Therefore, no matter how many ROOT ntuples are being analyzed, all of the data is outputted in one location. The file loop is located in the main function of the program and makes use of a class called Class Global to hold the locations of the files to run over. Within the file loop is the next step, the event loop. As previously stated, ROOT ntuples are organized by events. The event loop is nested in the file loop and uses a separate class, Class EventLoop, to access the individual events from the ROOT files. Within the event loop the particles from each event are accessed and stored, organized by the particle's type. Next is the particle-type loop. Nested within the event loop, the particle-type loop is executed by the Class InvMass. Within the event loop Class InvMass is passed TClonesArrays, arrays that are full of objects found in ROOT libraries, for each particle type. The particle-type loop collects each possible unique combination of these particle types (see Table 2). Lastly, there is the particle loop. Nested within the particle-type loop, it is the deepest and final step of InvMass. Also executed by Class InvMass, the particle loop runs through the particle-type groups and

finds each unique group of individual particles to analyze (Table 2). The analysis of the general search is preformed within this final loop, and the IM and total PT of the particle groups is determined. The information is stored in histograms based on the particle-type group the particles are from. For example, the IM of all electron-electron pairs are stored in one histogram, and the total PT of all jet-jet-gamma groups are stored in another, separate histogram.

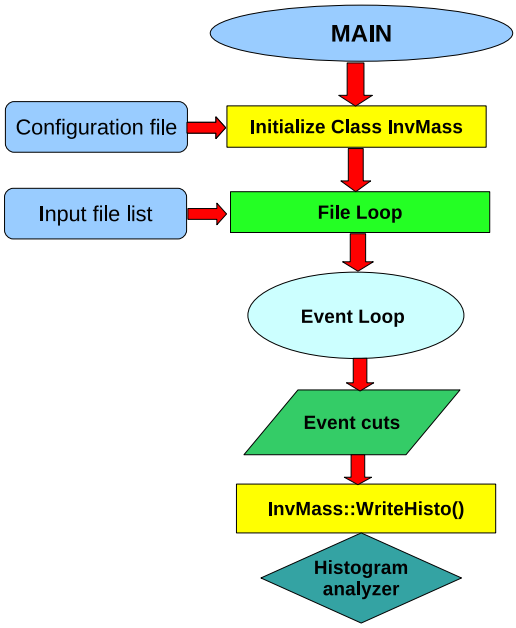

Figure 2: A schematic representation of the InvMass program structure.

3 Features

InvMass has many different features which can be accessed. The general search can be altered to fit different data sets and search ranges, while maintaining its generality. The search can perform basic cuts to maintain its integrity, and the output can be controlled to the specifications desired. All of the features can be changed with little to no alteration of the code, so the program will rarely need to be recompiled. This is possible through the use of a configuration file. Data sets and search ranges can easily be altered. There are a few options that control these tools. InvMass can be altered to work for a specific data set based on the particle types the data contains. It can be modified to select which of those particle types to use in the search. Furthermore, InvMass can be told how many particles to place in the decay groups being searched. To alter InvMass to fit a different data set is done through coding. It is the only change which requires the program to be recompiled. In this case, a different data set is the set of all data files stored in ROOT ntuples which contain data for a certain group of particle types. For example, if one data set contains information of electrons, jets, and photons, a different data set may contain information about electrons, photons, and muons. To alter InvMass to a specific data set requires an alteration of the code in the event loop. Within the event loop a vector is filled with the TClonesArrays of each particle type. But for each different data set, the particle-types change in some way. The vector created and passed by the event loop must be altered to contain every particle type available and only those particle types which are available based on the ROOT ntuples being analyzed. Once this is changed, the program

must be recompiled before it is used for that specific data set. However, in order to modifying InvMass to choose the particle types to use does not require coding. The configuration file for InvMass is used to make this change, so the program does not need to be recompiled. Within the configuration file there is a list of particle names. This list must match exactly to the vector created in the event loop, in the correct order, or else the output data will be misrepresented perhaps by electrons being labeled jets and vice versa.

Once the list is correct and complete, the particle types in the search can be chosen by commenting out those which are not desired. When a particle type is commented out, it is removed from the vector created in the event loop before the particle type loop is performed. Likewise, changing the number of particles to use in the decay groups is set in the configuration file. There is a value, numDecays, which can be set to any number 2 or greater. For example, if numDecays is set to 4 the program will than perform a search for all decay groups that contain 2, 3, or 4 particles. Any value can be chosen, there is no limit the search can be run for decay groups of 2 to 5, 2 to 10, and even 2 to 100 if desired. Since the feature is set through the configuration file, the program does not need to be recompiled. Performing Cuts InvMass also has the capability to perform generic cuts to maintain the integrity of the search. There are three cuts available; a minimum cut on PT for individual particles, a maximum rapidity cut for individual particles, and overlap removal for particles that were counted in two categories. The cuts are performed before the particle type loop is executed for higher efficiency.

All cuts can be set in a configuration file. First there is the PT cut. It is executed on a particle type basis, so each type is given a minimum value. Each individual particle is checked against its type's minimum value. If the particle's PT is too low, the particle is removed from the TClonesArray it was stored in. The rapidity cut is performed in a similar way. There is one major difference though only one cut is performed on all particles, regardless of their type. So, there is only one value to set, the maximum pseudorapidity value. As with the PT cut, all individual particles are checked and if a particle has a pseudorapidity value greater than the value set, it is removed from the TClonesArray that it was stored in. There is a third type of cut, overlap removal. This cut has a boolean option to turn it on and off, as well as specific options for how to perform it.

When data is collected and stored, some particles can be tagged as two different types, for example as both a jet and an electron. Only one of the tags is correct, so the other should be removed in order to maintain data integrity. The problem for InvMass became how to perform a general cut using overlap removal without knowing what the particles being cut were. The program needed to make a decision about which of the tags should be kept. So, a precedence value was added for each particle type. The particle type which has the higher precedence value is the particle type the double counted particle is removed from. However, there must also be some value used to determine if a particle is being double counted. Therefore, the value of dR is used; particles must be greater than a minimum dR value apart or else they are the same particle. If they are the same particle, they are removed according to the precedence value from the associated TClonesArray.

The output can be controlled using the input configuration file. There are three main sections of output that can be modified; the output file name, the histogram bin and range details, and peak searching. All are fully modifiable through the configuration file. The output file name is customizable. It is set as a relative path, so it can be placed anywhere relative to the executable of the program. By default it is set to write to the "out" folder. The file name is set as a stub, without the file extension. The extension is added by InvMass, based on what the output is, either the ROOT file which holds the histograms or the text file which holds the summary of the peaks found by the peak search. The histogram details are also fully customizable. There are two types of histograms created by InvMass; IM histograms and total PT histograms. Both types can be set to different specifications. The user determines the number of bins and the range covered by the histograms. Finally, there is a peak searching tool. There is a boolean option to turn peak searching on and off. The peak search InvMass uses is done by an object of the TSpectrum class

from ROOT. When the search is performed there are many different options to control how it determines which peaks are important enough to mark and which histograms are important enough to search. The ideal values for these search parameters have yet to be found.

4 Tests

4.1 Tests using signal events

Using a several Monte Carlo samples with signal events, the entire analysis chain was tested, starting from AOD events, creation of ROOT ntuples using the PromptGamma program and processing ntuples in parallel using the InvMass program.

For the Monte Carlo simulation, we have mixed several samples with arbitrary weights using PYTHIA. The Monte Carlo sample included $Z \to e^-e^+$, $\mu^-\mu^+$, $\tau^-\tau^+$, $G \to \gamma\gamma$ and $\gamma + jet$ events. The created ntuples include electrons, jets, photons, muons, Z bosons, and missing ET. No particular trigger selection was used. A typical even rate of the entire analysis chain is shown in Fig. 1.

The transverse momentum cut was set to 20 GeV for all particles, the rapidity cut was set to 2.4, and the dR value for overlap removal was set to 0.3. All particles types in the data sets were used, and decay groups of up to 4 particles were analyzed. The output histograms included invariant-mass distributions, transverse momenta of groups of particles and their rates. In case if a group contains a missing ET, the transverse invariant mass was calculated.

Due to a large number of generated histograms, we show only a few invariant-mass histograms for ”representative” channels with the expected peaks (histograms for event rates and transverse momenta are not shown). Figures 3 show several such channels with the peaks for the graviton $G \to \gamma\gamma$, Z-boson and peaks due to reflection from the $G \to \gamma\gamma$ decay (when γ was misreconstructed as a jet). The peak for γ-jet channel indicates that the overlap removal discussed before is not optimal and has to be further tuned.

4.2 Tests using SM background samples

The program was tested by simulating the standard model QCD background processes. For this, ATLfast program was used. It created in-memory store-Gate objects (see Fig. 1) for each event class and then saved to ntuples using the PrompGamma program. The whole analysis was performed using a Tier3 computer setup as described in [5].

As before, the event sample was generated using the PYTHIA model with the JF17 settings, after increasing the minimum transverse momentum cut to 130 GeV (CKIN(3) parameter). 100M events have been generated, which corresponds to 0.5 fb^{-1}. No trigger selection was applied. The events have been accepted for the $jet + X$ event topologies ($X = \mu, e, \tau, missET$) or $X + X$ event topologies ($X = \mu, e, \tau, missET$). Events with dijets were not considered due to reduce event rate for such tests. Each object should have the minimum transverse momenta 150 GeV. Events have been analyzed up to 4-body decays, and event rates, invariant masses and transverse-momenta of group of particles have been reconstructed.

Figures 4-6 show the most representative distributions for invariant masses for several event classes. The minimum invariant mass was set to 80 GeV and the rapidity range was restricted to 2.4.

5 Conclusion

Exotic physics will be a large part of the searches being performed using the LHC data. Where a range is predicted for finding a certain particle, that range can be analyzed through the general searches. For unforeseen physics, the broad view of the general search may unearth possibilities in ranges unlikely to be searched otherwise. A general search will be a first report of interesting areas, it will help to explore existing theories and perhaps it will even reveal new searches that should be performed.

The InvMass program is designed for general, model-independent searches for new physics over a wide kinematic range and arbitrary depth in grouping of identified objects. The program is easily adaptable to work for any collection of particles that the ATLAS detector can record. Some work must be done to fully understand the overlap removal algorithm in presence of many different types of objects and the peak search options.

A general search does have its limitations. The broad range does not allow for a very precise search with detailed background removal. Nevertheless, it can be used as a first report and a starting point for more detailed searches, and can provide a broad view of decay patterns for further studies. The program is available at [6].

6 Acknowledgments

I thank R. Blair, J. Proudfoot and R. Yoshida for the discussion of the results of this note. The work of J.B. was sponsored by the DOE Student Undergraduate Internship (SULI), ANL.

References

[1] H1 Collaboration, F. D. Aaron *et al.*, Phys. Lett. **B674**, 257 (2009), 0901.0507.

[2] CDF Collaboration, T. Aaltonen *et al.*, Phys. Rev. D (2009), 0809.3781.

[3] G. Choudalakis, Ph.D thesis, MIT, arXiv:0805.3954, 2008.

[4] PromptGamma package, https://twiki.cern.ch/twiki/bin/view/AtlasProtected/PromptGamma.

[5] S. Chekanov, S.and Yoshida, A PC farm for ATLASi Tier3 analysis ATL-COM-GEN-2009-016, 2009.

[6] InvMass package, http://atlaswww.hep.anl.gov/asc/WebSVN/listing.php?repname=InvMass.

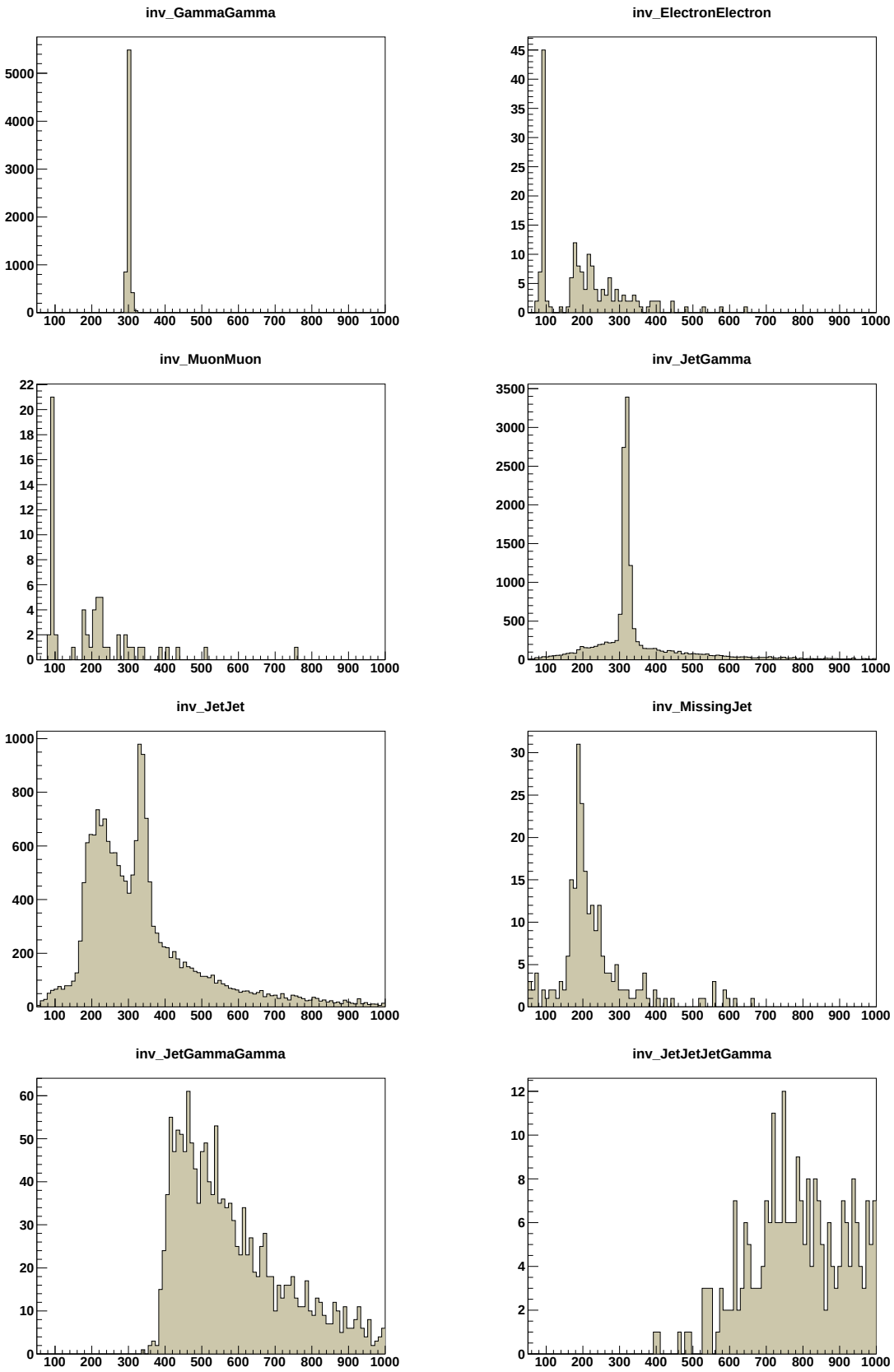

Figure 3: Invariant masses for several representative channels using the full Monte Carlo simulation and several signal channels.

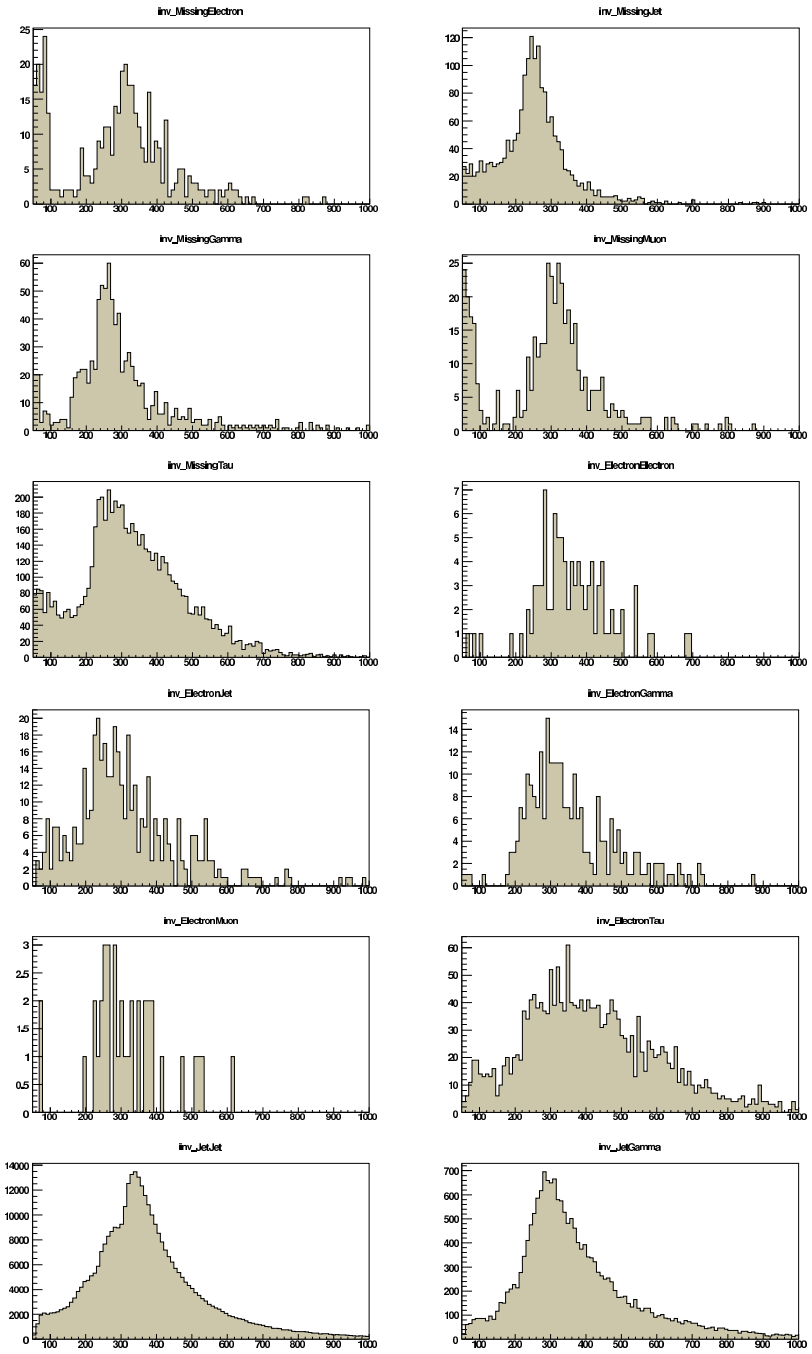

Figure 4: Invariant masses for several representative channels for fully inclusive pp collisions using a fast Monte Carlo simulation based on the PYTHIA model.

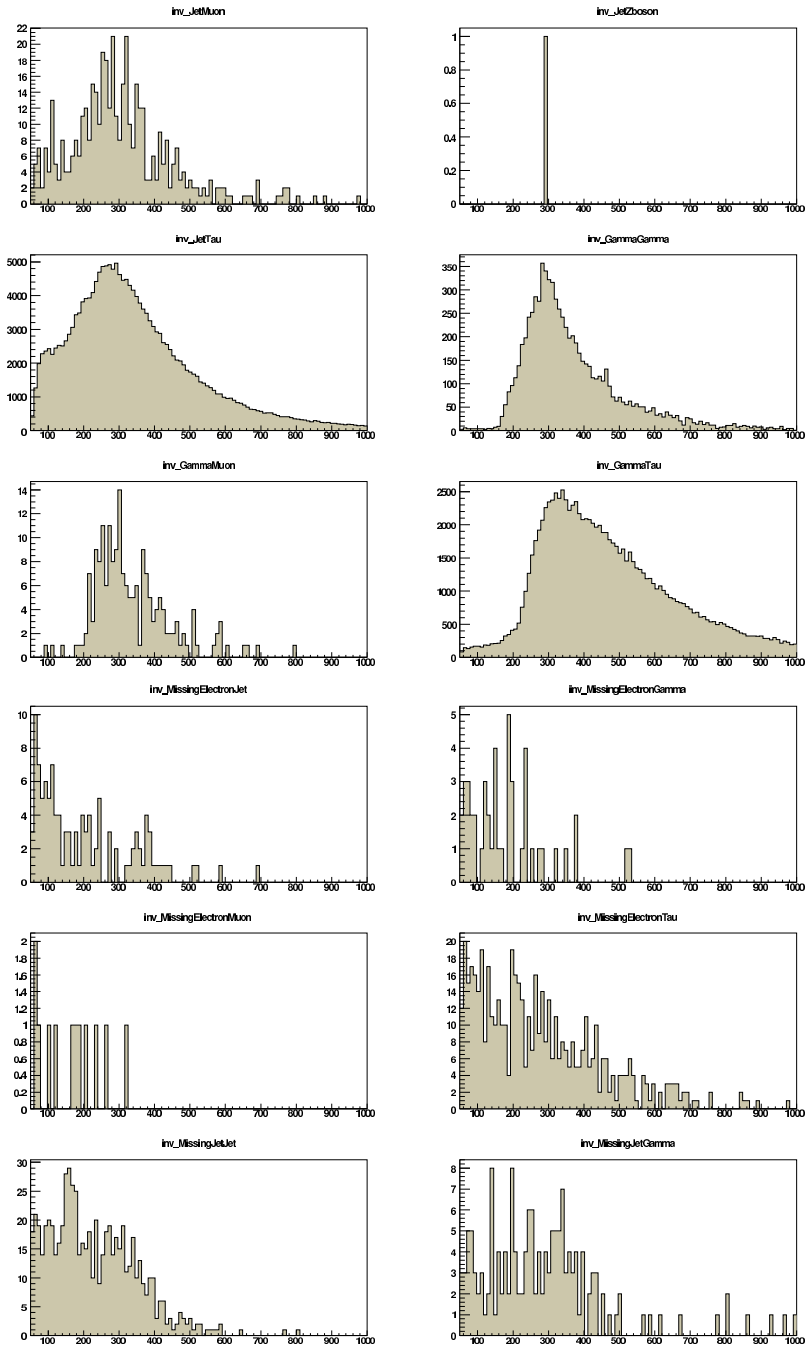

Figure 5: Invariant masses for several representative channels for fully inclusive pp collisions using a fast Monte Carlo simulation based on the PYTHIA model.

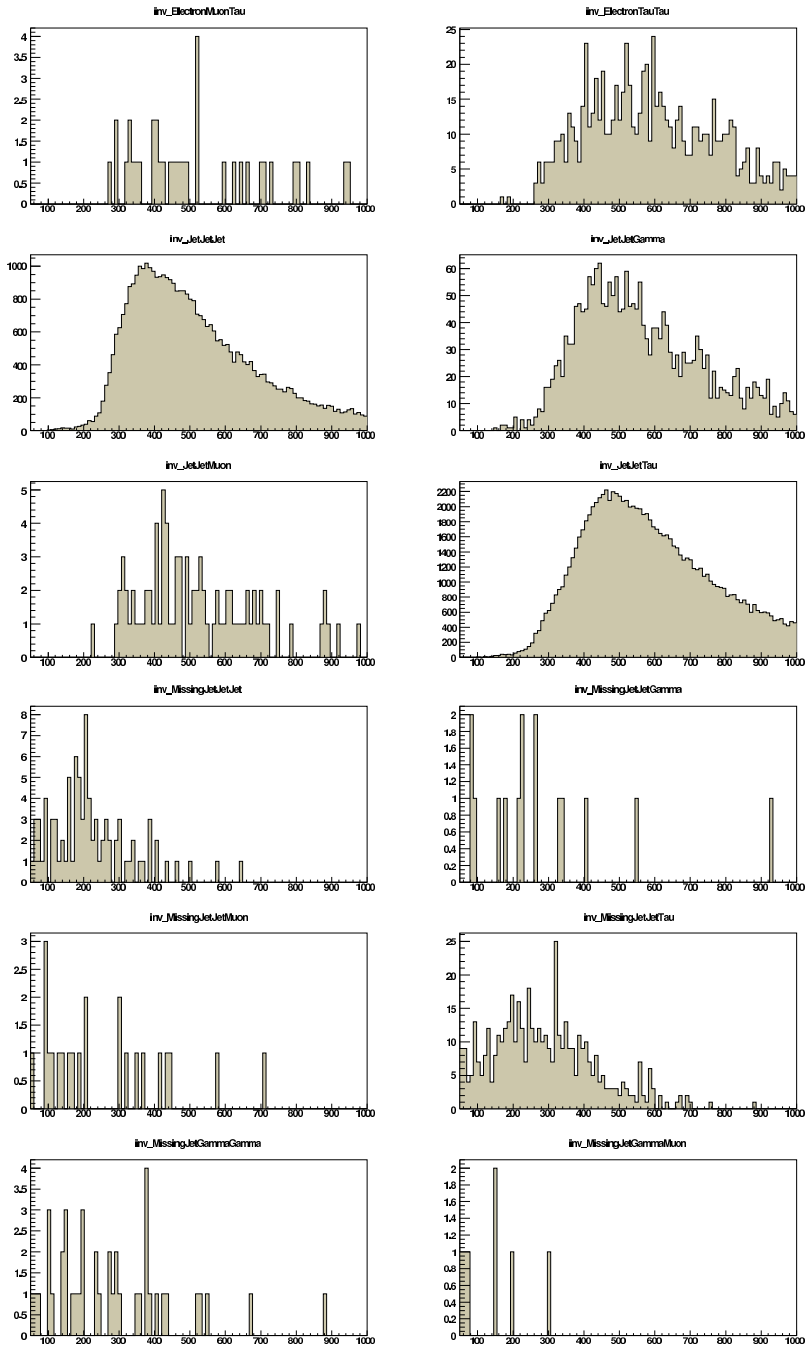

Figure 6: Invariant masses for several representative channels for fully inclusive pp collisions using a fast Monte Carlo simulation based on the PYTHIA model.

High Energy Physics Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 360
Argonne, IL 60439-4814

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC