2018

A Study of Physics Beyond the Standard Model

Robert Orlando

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS & SCIENCES

A STUDY OF PHYSICS BEYOND THE STANDARD MODEL

By

ROBERT ORLANDO

A Thesis submitted to the
Department of Physics in partial fulfillment of the
requirements for graduation with
Honors in the Major

Degree Awarded:
Summer, 2018

The members of the Defense Committee approve the thesis of Robert Orlando defended on March 30th, 2018.

Dr. Harrison Prosper
Thesis Director

Dr. Paolo Aluffi
Outside Committee Member

Dr. Susan Blessing
Committee Member

A Study of Physics Beyond the Standard Model

Robert Orlando
Department of Physics, Florida State University, Tallahassee

Abstract
This paper describes a study of an experimentally viable decay mode of the heavy neutral Higgs boson (H) that could be created in proton proton collisions at a center of mass of 13 TeV. The heavy neutral Higgs boson is one of several hypothesized particles of the phenomenological Minimal Supersymmetric Standard Model (pMSSM). We study the decay of H into a $\tau^+\tau^-$ pair. Events were generated using the `PYTHIA8` program and analyzed at the pre-detector level. We find that H is created with low transverse momentum and the resulting τ jets are back-to-back. The H has a broad width and a cross section that is about one thousand times smaller than that for $pp \to Z \to \tau^+\tau^-$.

1 Introduction

This paper describes the results of a study of the process

$$pp \to H \to \tau^+\tau^-, \tag{1}$$

where H is the heavy neutral Higgs boson of the Minimal Supersymmetric Standard Model (MSSM). The events are generated with the `PYTHIA8` program, which creates events in `HepMC` format.[4] The goal of the study was to develop an understanding of such processes and to develop a strategy to search for them at the Large Hadron Collider.

First we motivate the need for a theory like the MSSM given current experimental observations. We argue that this theory is essentially untestable with the current resources of High Energy Physics research collaborations. Then, we consider a sub-model of the MSSM called the phenomenological MSSM (pMSSM), which permits an indirect, but practical, approach to testing the MSSM and explain why the pMSSM is the most reasonable and testable theory among the various sub-models of the MSSM. We follow with a discussion of how we determined that the above decay mode was the most prominent for the heavy neutral Higgs boson. Then, we briefly describe the procedure of this project, difficulties encountered, and solutions to those difficulties. This is followed by a discussion of the findings. Finally, we explore how machine learning techniques can be used to discriminate between heavy Higgs boson events and Standard Model (SM) events.

1.1 Motivation

The Standard Model is a highly successful theory that describes the interaction and properties of elementary particles. This theory gives a precise and well-tested description of the strong, weak and electromagnetic forces. The theory predicted the gluon, which was discovered in 1979, the W and Z bosons discovered in 1983, the top quark discovered in 1995, the tau neutrino discovered in 2000, and (most recently) the Higgs boson discovered in 2012[1].

Despite the successes of the SM there are many questions that remain unanswered and experimental observations that suggest the SM is an incomplete theory. The most compelling of these is the fact that no particle in the SM can serve as a candidate for dark matter. It is known that if we account for the gravity produced by every visible gravitational source within a galaxy, the galaxy would rip itself apart due to its own rotation. In other words, the gravity of the visible matter within galaxies is not enough for them to exist as stable structures, which is an obvious contradiction. The preferred solution to this problem is to presume the existence of dark matter, which does not interact with light or ordinary matter, or at least interacts very weakly. Recent observations suggest that dark matter accounts for 26% of the energy of the universe as compared to the 5% contribution from ordinary matter[2]. The fact that the SM does not account for one fourth of the energy content of the universe is a discrepancy that cannot be overlooked. (In fact, the situation is worse than this because the SM does not explain the remaining component either, the dark energy.)

The SM cannot account for the existence of gravity.[2] The other three forces of the universe (weak, strong, electromagnetic) have force-carrying particles that mediate the force (e.g. the photon is the force-carrying particle of the electromagnetic force); however, physicists have no experimental verification of the graviton. This leaves the most familiar force of nature without an explanation via elementary particles. The SM cannot be a complete theory when we reach the Planck scale since it must account for quantum gravity effects which it does not.[2] Additionally, the SM cannot explain why the Higgs boson mass is so low. Calculating the correction to the mass would yield a value much higher than the measured mass of 125 GeV. The discrepancy between the measured mass and Planck mass is referred to as the hierarchy problem. One way to resolve this problem is to introduce more particles into the SM.[2]

The SM does not account for the baryon asymmetry. It is an obvious fact that the universe is made of matter. One may think it childish to ask: why is the universe made of matter and not something else? This question is much deeper than it seems on the surface. It is theorized today that there was an equal amount of matter and antimatter created in the Big Bang[2]. So why is the universe made of matter and not antimatter, or equal amounts of both? We currently do not know the answer to this question. If there was as much matter as there was antimatter would not everything annihilate and result in a vast sea of radiation? Obviously not, since we are here. We believe there are three necessary conditions for this baryon asymmetry: (i) charge conjugation and parity violation, (ii) absence of thermal equilibrium, (iii) at least one baryon number violating process (baryon numbers will be described more in the next paragraph). (ii) and (iii) do not exist in the SM and even if they did there is not enough parity violation to account for the observed baryon asymmetry.

An issue of a more theoretical nature with the SM is there is no compelling reason for the conservation of baryon and lepton numbers[2]. In the SM, baryons, mesons, and anti-baryons are assigned a baryon number of 1, 0, and -1 respectively. Leptons and their neutrinos have a lepton number of 1 while anti-leptons and their associated anti-neutrinos have a lepton number of -1. Every other particle in the SM has a lepton number of 0. In every SM process baryon and lepton numbers are conserved. These numbers could be analogous to the principal, orbital angular momentum, and magnetic quantum numbers. Back in the days before the quantum theories of Schrödinger and Heisenberg, a detailed understanding of these quantum numbers and how they are related to the dynamics of the atom was unknown. The physicists of that period merely recognized these numbers follow certain rules. They then discovered that these numbers actually indicated the geometry of the atomic orbitals. Having a property as fundamental as conservation indicates there is probably a deeper explanation of baryon and lepton numbers not available in the SM as was the case with the aforementioned quantum numbers.

In addition to the lepton and baryon numbers there exist an electron number, muon number, and tau number. Again we find these numbers are conserved in particle processes without any explanation as to why. Can these numbers provide a theoretical explanation as to what makes an electron an electron or a muon a muon? We once again find ourselves in similar shoes as the physicists before the theories of Bohr, Heisenberg, and Schrödinger; we currently have no explanation of these quantum numbers of the SM. My previous sentiment holds, conservation has proven too fundamental in the understanding of physical phenomena to simply be a coincidence; there has to be a deeper explanation of these quantum numbers that reveals something about the nature of electrons, muons, and taus.

Another theoretical question the SM does not explain is why the electron, muon, and neutrino have the masses they do. We currently look at the masses of the electron, muon, and neutrino as experimental facts of life; they are quantities we measure. There is currently no equation whose solutions result in the mass of any of these three objects. Having experimental observations and mathematical explanations of phenomena is not only in the true nature of physics, it is required for us to reasonably claim that we understand how the universe came to be and how the universe behaves.

These shortcomings of the SM demonstrate that our modern view of the universe is incomplete and the SM needs to be extended. There have been multiple attempts to rectify some or all of these issues. One class of attempts, the subject of an enormous effort over the past approximately three decades, is supersymmetric theories. We will explore these in the next section.

1.2 Possible Solutions

The modern view of SM is that it is the low energy limit of a more complete and accurate theory.[3] A candidate for this more complete theory is the Minimal Supersymmetric Standard Model (MSSM). The MSSM can be visualized as a massive collection of points. Each point has a list of parameters whose values vary from point to point. Each point corresponds, in effect, to a universe of particles with different attributes that are completely determined from the parameters of each point. This list of parameters includes the masses of the supersymmetric counterparts of the particles in the SM. The task of testing the MSSM is like an Easter egg hunt where we do not even know if there is an egg to be found. We wish to find the point in the MSSM whose parameters correctly predict experimental results and therefore describe our reality. If we found such a point, it would imply that there exists a whole host of undiscovered supersymmetric particles. The theory is named quite poorly as it is by no means as minimal as we would like it to be; each point contains a parameter list of 119 entries. There is simply not enough funding, manpower, or time available at CERN (or anywhere else for that matter) to take on the task of testing a 119 parameter theory.

To shorten the list of parameters, high energy theorists imposed a number of strict assumptions about the physics at energy scales of 10^{15} GeV to arrive at several sub-models of the MSSM, including the constrained Minimal Supersymmetric Standard Model (cMSSM). The list of parameters for each point in the cMSSM is four entries long. This would seem an attractive theory because of the short list of parameters; however, the theory is becoming less and less viable because of a growing number of experimental results from the Large Hadron Collider (LHC). It seems to be an incorrect theory because no experiment has been able to find any evidence supporting its predictions.

Some argue that this spells the demise of the MSSM and perhaps the idea of supersymmetry. However, the near demise of the cMSSM may simply mean that the assumptions at an energy scale that not even our most energetic accelerators can achieve are just untenable. If we were to relax these assumptions and make only those that are experimentally viable without changing the phenomenology of the MSSM, we arrive at the phenomenological Minimal Supersymmetric Standard Model (pMSSM), the topic of this paper. This theory was first proposed in 2009, but was largely ignored in favor of the cMSSM since the latter was perceived as much simpler to test. However, today, since the cMSSM is close to being ruled out, physicists are becoming aware of the potential of the pMSSM; it is becoming more mainstream.

1.3 The pMSSM

The list of parameters in this theory is 19 entries long; longer than the cMSSM but much shorter than the 119 entry long parameter list. Testing the validity of the pMSSM is quite attractive to a high energy experimentalist because the theory makes unambiguous claims (e.g. there exists a supersymmetric and heavy counterpart to the SM Higgs boson) that can be tested at the LHC given the anticipated data to be obtained over the next quarter century.

In 2011, the CMS collaboration sampled approximately 7,000 parameter points of the pMSSM and compared the predictions from each of these points with data. Half of these parameter points have been rejected as having nothing to do with reality after work by CMS. We refer to these points as excluded pMSSM points. Appropriately, the other half of points that have not been ruled out are referred to as non-excluded. Half of these non-excluded points have cross sections above 10 femto-barns (1 barn = 10^{-24} cm^2) yet the scientists at CMS were unable to make any definitive conclusions regarding these pMSSM points. The reason is because all the currently excluded pMSSM points predicted a relatively high missing transverse momentum. Missing transverse momentum (equivalently energy) is the negative of the vectorial sum of the transverse momenta of all observed particles in a collision. The initial momentum in the transverse plane is zero and because of momentum conservation must be zero after a collision. Large missing transverse momentum is a signature of weakly interacting particles that escape direct detection. This led scientists to believe that the non-excluded points should also have a high missing transverse momentum which is not the case. In fact the missing transverse momentum of a large number of non-excluded points was lower than the threshold imposed by physicists at the LHC. This warns us that supersymmetry may be hidden in plain sight but given previous methods we have not been able to detect it.

The 2012 discovery of the SM Higgs boson (h) was detected by analyzing the process

$$pp \to h, \tag{2}$$

followed by the decay of the Higgs boson into two photons and four leptons. Since CMS has experience and was successful in detecting the Higgs boson studying this process, it is reasonable to apply similar methods to

$$pp \to H, \tag{3}$$

to detect a heavier variant of the same particle. Of course, the heavy neutral Higgs boson differs in important ways from the SM Higgs boson. For example, the SM Higgs boson has a sharp resonance less than a few GeV wide. We will see this is not the case of the heavy Higgs boson which will make detection much more difficult. The assertion that a heavy neutral Higgs boson exists is characteristic of many theories of new physics which makes this study all the more pertinent to further our understanding of physics.

It should be noted that the cMSSM is a subset of the pMSSM. It seems that the cMSSM does not reflect reality, but this does not rule out the pMSSM. However, if it was reversed and we could say with certainty that the pMSSM is not correct, then we would be able to say with certainty that the cMSSM is not correct as well. Indeed, all of the sub-models of the MSSM that have been studied and tested over last quarter-century would be ruled out.

2 Project Description

In preparation for this honors project, I worked to determine which decay mode of the heavy neutral Higgs boson within the pMSSM was most prevalent and experimentally viable. This was done by generating events for each non-excluded pMSSM parameter point using `PYTHIA8` and calculating the cross-section multiplied by the branching ratio (σBR) for each decay process. (The pMSSM file including all SM decays is provided in appendix A.) Of all decay processes of the heavy Higgs boson the decay

$$H \to b\bar{b}, \tag{4}$$

had the highest value of σBR consistently for each pMSSM point. Unfortunately, this process is very difficult to detect at the LHC because it competes with other processes resulting in a $b\bar{b}$ pair that have much higher cross-sections. We therefore chose the decay whose σBR was second-highest for each pMSSM point which was

$$H \to \tau^+\tau^-, \tag{5}$$

For a given pMSSM parameter point the cross section of process (4) and of process (5) was the same. However, the branching ratio (the probability a particle will decay a certain way) of process (4) was in most cases above 75% and below 95% while for process (5) it was in most cases only above 10% and below 20%. There is one decay mode that should be mentioned and that is the process

$$H \to t\bar{t}. \tag{6}$$

This decay mode was ignored for the purposes of this project because for each pMSSM point its cross section was at least one order of magnitude less than that of process (4) and process (5) and most of the time had a branching ratio less than 1%. All of these facts combined led us to do an in depth study of process (5).

2.1 Figures

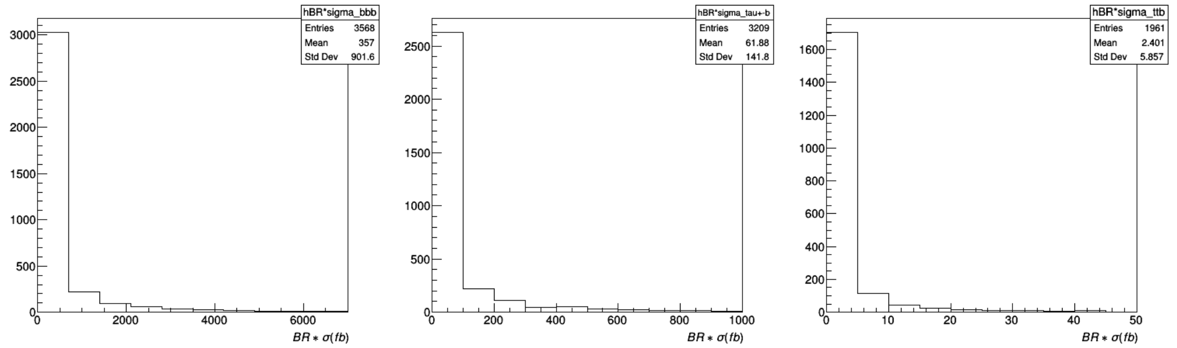

Fig. 1: Above are the histograms for σBR for process (4) (left), process (5) (center), and process (6) (right). We can see that this multiplicative factor is large for process (4) but too small for process (6). However, the cross-section of process (4) has to compete with other processes with much larger cross sections making it harder to detect.

As stated above, the goal of this project was to study the decay of the heavy Higgs boson, specifically the decay (5). We wanted to understand the kinematic characteristics of this decay that could possibly assist in the detection of heavy Higgs boson events at the LHC. First we will describe the procedure of this work, summarize our findings, and highlight difficulties as well as describe solutions implemented.

2.2 Procedure

The range of the pMSSM Higgs boson mass over the pMSSM parameter points has a lower threshold of approximately 340 GeV and an upper threshold of approximately 2800 GeV. Choosing a parameter point near the upper limit of this threshold would be misguided; the cross-sections for H of this mass is low compared to that of a Higgs boson of intermediate mass. Therefore, we chose to study a pMSSM point that yielded a heavy Higgs boson of 1700 GeV, which is reasonable given our current experimental capabilities and properties of the decay.

We then generated 50,000 events for this pMSSM point using PYTHIA8 using the PYTHIA8 card file shown below

```
! ----------------------------------------------------------------------
! File: pMSSM_4_699476.txt
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter or digit are comments.
! Names are case-insensitive  -  but spellings-sensitive!
! The changes here are illustrative, not always physics-motivated.
! ----------------------------------------------------------------------
```

```
! 1) Settings that will be used in a main program.
Main:numberOfEvents  =  50000        ! number of events to generate/read
Main:timesAllowErrors = 500        ! abort run after this many flawed events
! ----------------------------------------------------------------------
! 2) Settings related to output in init(), next() and stat().
Init:showChangedSettings = on      ! list changed settings
Init:showAllSettings = off         ! list all settings
Init:showChangedParticleData = on  ! list changed particle data
Init:showAllParticleData = off     ! list all particle data
Next:numberCount = 1000            ! print message every n events
Next:numberShowLHA = 1             ! print LHA information n times
Next:numberShowInfo = 1            ! print event information n times
Next:numberShowProcess = 10         ! print process record n times
Next:numberShowEvent = 1           ! print event record n times
Stat:showPartonLevel = on          ! additional statistics on MPI
! ----------------------------------------------------------------------
! 3) Beam parameter settings. Values below agree with default ones.
Beams:idA = 2212                   ! first beam, p = 2212, pbar = -2212
Beams:idB = 2212                   ! second beam, p = 2212, pbar = -2212
Beams:eCM = 13000.                 ! CM energy of collision
! ----------------------------------------------------------------------
! 4a) Process

HiggsBSM:allH2        = on         ! H0 (H2) Heavy scalar Higgs
Higgs:useBSM                  = on        ! Use BSM parameters rather than those of SM

SLHA:file                    = pMSSM_4_699476.slha
SLHA:keepSM             = on
SLHA:minMassSM          =  200.0

SLHA:allowUserOverride = true

35:mMin = 600.

35:onMode = off
35:onIfMatch = 15 -15
```

Blocks (1) to (3) indicate how we want PYTHIA8 to print pertinent information to the screen and are somewhat self explanatory given the heavy commenting throughout. Block (4) is the more fundamental and cryptic of the card file. Every line starting with a "!" is commented out and is ignored. The first two lines indicate that we wish to generate events with the heavy scalar Higgs boson. Next we indicate which pMSSM point we are analyzing and that we wish to keep SM decay modes above a certain mass threshold (200 GeV). We then impose a cut of 600 GeV dependent on the width of the mass peak indicated in the pMSSM file and tell PYTHIA8 we only wish to keep events that decay into a $\tau^+\tau^-$ pair (which have particle identification numbers of 15 and -15 respectively). The width of the Higgs boson mass peak was about 160 GeV which indicates that this object decays very rapidly. The package

https://github.com/hbprosper/pythia

was used to run the PYTHIA8 generator, both to calculate cross sections and to write events in HepMC format. The HepMC file created by PYTHIA8 was converted to a ROOT file via a python script to make it possible to analyze the events in a simple manner. The histograms generated were of the transverse momentum of the heavy Higgs boson, transverse momentum of the two most energetic τ jets, the angle between the τ jets, and the mass distribution of the heavy Higgs boson. In a real analysis we would in fact measure the charged particles that result from the decay of the τ.

2.3 Encountered Difficulties

In the documentation of PYTHIA8 it is not stated how, for supersymmetric events, one can force the program to pick only events with a certain decay process. As a result, we took many detours via python scripts to try to filter out events of decay (5) when it turns out that to do this one essentially only needs to enter a few lines of code into the card file provided to PYTHIA8. To learn of this neat trick we had to contact one of the developers of the PYTHIA8 software.

A second difficulty is that for each pMSSM point there is a steep increase in the event rate as we tend to a mass of 0 for the heavy Higgs boson. It is not clear in the documentation that this steep increase can corrupt the mass calculation in PYTHIA8. The solution to this issue is to apply a cut on the heavy Higgs boson mass dependent on the width of the peak about the resonant mass of the heavy Higgs boson. Since we made a cut on the mass we also need to impose a cut on the transverse momentum of the τ jets so we do not bias our results. A derivation of how the cut was determined is included below. We first note that the mass of both τ particles is very small compared to the mass of H making them negligible. Also, it is known that both jets are back-to-back and the magnitude of their momentum is the same.

$$H = \tau_1 + \tau_2, \tag{7}$$

where τ_1 and τ_2 are 4-vectors of the τ particles. Squaring the above will yield

$$m_H^2 = 2\tau_1\tau_2 > m_0^2, \tag{8}$$

where m_0 is the imposed mass cut on the heavy Higgs boson.

$$\tau_1\tau_2 = E_1E_2 - \vec{p}_{T1} \cdot \vec{p}_{T2} - p_{T_{z1}}p_{T_{z2}}, \tag{9}$$

where $p_{T_{z_i}}$ is the magnitude of the momentum of the τ's in the z direction and $\vec{p_{Ti}}$ is the transverse momentum vector of the τ's. Keep in mind that the momenta of the τ particles are back-to-back. We know from Einstein that the energies of the τ particles (in units of c = 1) are given by

$$E_i = \sqrt{m^2 + p_{Ti}^2 + p_{zi}^2} \approx p_{Ti} \tag{10}$$

setting p_{zi} to its minimum value of zero and neglecting the mass of the taus. This then implies that

$$\tau_1\tau_2 = 2p_1^2 \tag{11}$$

$$p_{T_1} > \frac{m_0}{2} \tag{12}$$

As a general rule, we chose a mass cut around four times the peak width. The peak width for this pMSSM point was 160 GeV making the mass cut 600 GeV. Therefore, we chose a cut of approximately 500 GeV for the transverse momentum of the τ jets.

A difficulty that does not directly influence this project is the actual width of the mass peak. For the SM Higgs boson the width of the resonance is less than a few GeV as compared to the 160 GeV width of a 1700 GeV heavy Higgs boson. This will make detection of the heavy Higgs boson much more difficult because the peak is not as sharp.

2.4 Findings

We find that the heavy Higgs boson, a high mass object, is essentially at rest when it is produced. This can be seen in the histogram (Figure (2)) of the transverse momentum of the Higgs boson. Then if we apply our understanding of momentum conservation we can reason that the τ jets are 180 degrees to each-other and should have similar transverse momentum distributions. We find this to be true if we histogram these values (Figure (3)) using the TLorentzVector class of ROOT.

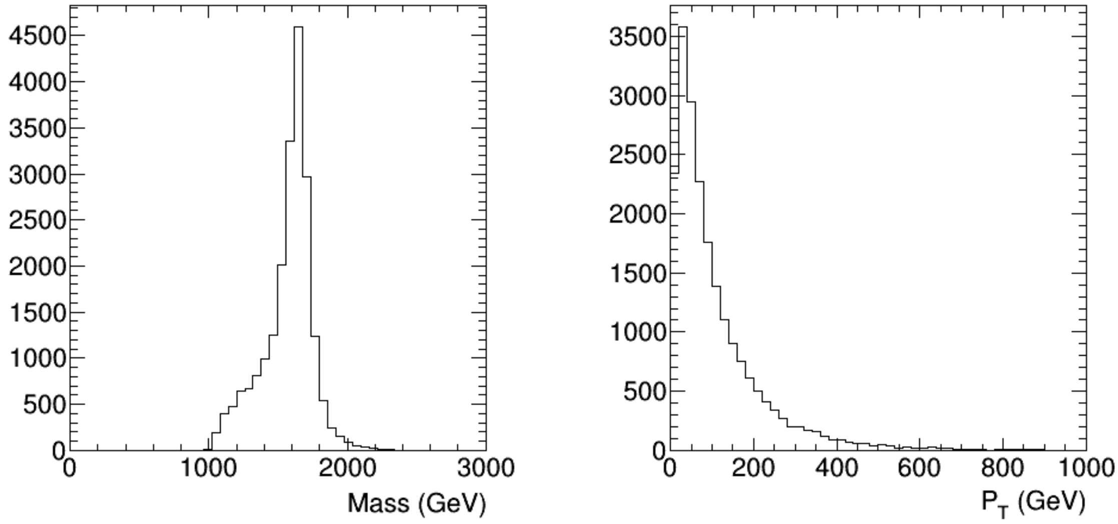

Fig. 2: (left) Mass distribution of the heavy Higgs boson for the provided pMSSM point with a lower threshold of 600 GeV. We can see there is an unmistakable peak at 1700 GeV which is the mass indicated in the pMSSM file. (right) A histogram of the transverse momentum of the generated heavy Higgs boson. An overwhelming majority of the generated pMSSM Higgs bosons are essentially at rest.

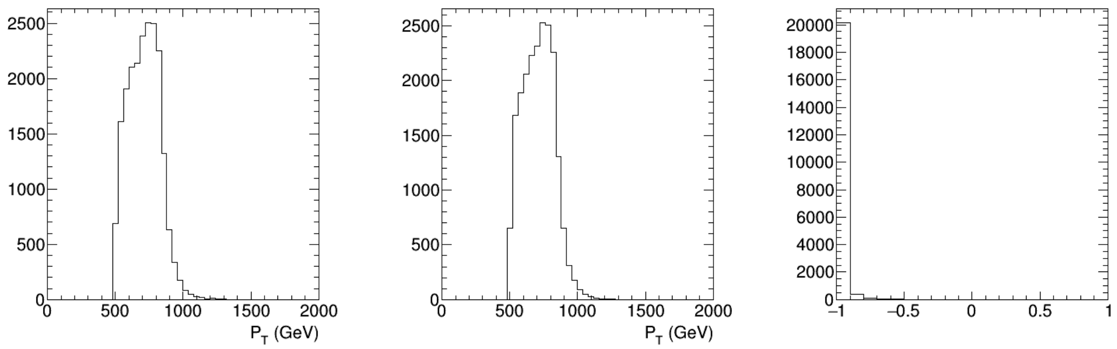

Fig. 3: (left and center) Transverse momentum distribution of the tau jets. We can see that the distributions are extremely similar since the heavy Higgs boson is at rest. (right) Cosine of the angle between the tau jets. Almost all tau jets are back-to-back.

2.4.1 Figures

3 Machine Learning Techniques

We plan to use machine learning in the next stage of this study to discriminate H from non-H events. Machine learning is widely used to create highly effective functions for classifying objects into groups. Since we will have millions of events to classify in a real analysis, fast classification is an essential property of our discrimination method. We plan to implement machine learning to automate the classification process. An analogy to how machine learning will be implemented in this project is given by facial recognition technology. We can ask how we can program a phone to distinguish between Sarah's face and Lauren's face. This is an exceedingly difficult question to answer with a hand-written computer program. Instead, one can ask, why not take many different pictures of Lauren's face and likewise for

Sarah, feed in the “raw-est" data of those pictures, the pixels, and then have the phone read the pixels? The phone would then “decide" for itself what separates Lauren’s face from Sarah’s face based on those pixels. The same approach is taken in this project, but our goal is to separate H events from non-H events based on the 4-vectors and particle types of each reaction. It should be noted that the machine will not be 100% efficient, but our goal is to achieve the best discrimination that is possible. We will use the machine learning package called `scikit-learn`, which is extremely user friendly. In this package, there are multiple machine learning methods, each of which has parameters than can be chosen to guide the training.

4 Summary

Testing the MSSM has been a priority research topic for more than two decades. But so far no evidence in favor of it has been found. However, physicists cannot yet rule out the MSSM because of the difficulty of testing a theory with a 119-dimensional parameter space. One very compelling subset of the MSSM, the pMSSM, is attractive experimentally because it makes unambiguous claims and attractive theoretically because it preserves the phenomenology of the MSSM. We have learned key characteristics of the most prominent and experimentally viable decay mode of the heavy Higgs boson. Namely that the tau jets are back-to-back and have extremely similar energy signatures. After applying machine learning techniques to theτ 4-vectors we will have developed a tool that is useful and efficient in testing the pMSSM via process (5).

Acknowledgments

I wish to thank the committee of Dr. Aluffi, Dr. Blessing, and Dr. Prosper for their guidance and advice on this project. I also wish to thank Dr. Mrenna for being so communicative on the "in’s and out’s" of `PYTHIA8`.

Dr. Aluffi, I greatly appreciate your willingness to be on the committee despite your travels. Your wonderful communication has made working with you a pleasure. Your inclusion has, in no small part, made this thesis a reality.

Dr. Blessing, throughout my undergraduate years you have been a wonderful source of insight and no-nonsense advice in academic and personal matters. It has proven extremely formative in my transition from a teenager into a young adult. Your advice has consistently led me in the right direction.

Dr. Prosper, I remember walking into your office as a lowly, starry-eyed freshman asking you for research experience. Despite my lack of experience you mentored me and showed me unparalleled patience throughout my new experience. All that I know about High Energy Physics is because of you. It has been an absolute privilege to work with you these four years.

References

[1] CERN. $https://home.cern/about/physics/standard\text{-}model$ - The Standard Model

[2] N. Strobbe, “The Razor Boost analysis, Another step in the hunt for new physics at CMS”

[3] A. Romanino. The Standard Model of Particle Physics.

[4] `HepMC` $http://lcgapp.cern.ch/project/simu/HepMC/$

5 Appendix

5.1 Appendix A - pMSSM File

```
#
#                              =====================
#                              | THE SUSYHIT OUTPUT |
#                              =====================
#
#
#              -----------------------------------------------------
#              |     This is the output of the SUSY-HIT package     |
#              |  created by A.Djouadi, M.Muehlleitner and M.Spira. |
#              |  In case of problems with SUSY-HIT email to        |
#              |           margarete.muehlleitner@cern.ch           |
#              |           michael.spira@psi.ch                     |
#              |           abdelhak.djouadi@cern.ch                 |
#              -----------------------------------------------------
#
#              -----------------------------------------------------
#              |  SUSY Les Houches Accord - MSSM Spectrum + Decays  |
#              |              based on the decay programs           |
#              |                                                    |
#              |                     SDECAY 1.3b                    |
#              |                                                    |
#              |  Authors: M.Muhlleitner, A.Djouadi and Y.Mambrini  |
#              |  Ref.:    Comput.Phys.Commun.168(2005)46           |
#              |           [hep-ph/0311167]                         |
#              |                                                    |
#              |                     HDECAY 3.4                     |
#              |                                                    |
#              |  By: A.Djouadi,J.Kalinowski,M.Muhlleitner,M.Spira  |
#              |  Ref.:    Comput.Phys.Commun.108(1998)56           |
#              |           [hep-ph/9704448]                         |
#              |                                                    |
#              |                                                    |
#              |  If not stated otherwise all DRbar couplings and   |
#              |  soft SUSY breaking masses are given at the scale  |
#              |  Q=  0.19122178E+04                                |
#              |                                                    |
#              -----------------------------------------------------
#
#
BLOCK DCINFO  # Decay Program information
     1   SDECAY/HDECAY # decay calculator
     2   1.3b  /3.4    # version number
#
BLOCK SPINFO  # Spectrum calculator information
     1    SOFTSUSY    # spectrum calculator
     2    3.3.1       # version number
#
BLOCK MODSEL  # Model selection
     1     0   #  nonUniversal
#
BLOCK SMINPUTS  # Standard Model inputs
         1     1.27908953E+02   # alpha_em^-1(M_Z)^MSbar
         2     1.16637000E-05   # G_F [GeV^-2]
         3     1.18540000E-01   # alpha_S(M_Z)^MSbar
```

```
         4     9.11876000E+01   # M_Z pole mass
         5     4.66000000E+00   # mb(mb)^MSbar
         6     1.76987000E+02   # mt pole mass
         7     1.77700000E+00   # mtau pole mass
#
BLOCK MINPAR  # Input parameters - minimal models
         3     5.36400000E+01   # tanb
#
BLOCK EXTPAR  # Input parameters - non-minimal models
         1    -2.79720000E+02   # M_1(MX)
         2     1.37052000E+03   # M_2(MX)
         3     2.44299000E+03   # M_3(MX)
        11    -2.27185000E+03   # At(MX)
        12     2.25917000E+03   # Ab(MX)
        13    -4.38541000E+03   # Atau(MX)
        23    -2.61950000E+03   # mu(MX)
        26     1.66325000E+03   # mA(pole)
        31     2.11105000E+03   # meL(MX)
        32     2.11105000E+03   # mmuL(MX)
        33     2.08798000E+03   # mtauL(MX)
        34     2.89587000E+03   # meR(MX)
        35     2.89587000E+03   # mmuR(MX)
        36     1.37121000E+03   # mtauR(MX)
        41     2.07150000E+03   # mqL1(MX)
        42     2.07150000E+03   # mqL2(MX)
        43     2.64606000E+03   # mqL3(MX)
        44     8.28680000E+02   # muR(MX)
        45     8.28680000E+02   # mcR(MX)
        46     1.37831000E+03   # mtR(MX)
        47     1.21292000E+03   # mdR(MX)
        48     1.21292000E+03   # msR(MX)
        49     2.51871000E+03   # mbR(MX)
#
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
        24     8.03932692E+01   # W+
        25     1.22014911E+02   # h
        35     1.66308917E+03   # H
        36     1.66329053E+03   # A
        37     1.66489494E+03   # H+
         5     5.34455584E+00   # b-quark pole mass calculated from mb(mb)_Msbar
   1000001     2.15128841E+03   # ~d_L
   2000001     1.29658823E+03   # ~d_R
   1000002     2.14995927E+03   # ~u_L
   2000002     9.21772559E+02   # ~u_R
   1000003     2.15128841E+03   # ~s_L
   2000003     1.29658823E+03   # ~s_R
   1000004     2.14995927E+03   # ~c_L
   2000004     9.21772559E+02   # ~c_R
   1000005     2.50817285E+03   # ~b_1
   2000005     2.73392691E+03   # ~b_2
   1000006     1.47283218E+03   # ~t_1
   2000006     2.69130439E+03   # ~t_2
   1000011     2.11936693E+03   # ~e_L
   2000011     2.89887794E+03   # ~e_R
   1000012     2.11758362E+03   # ~nu_eL
```

```
    1000013     2.11936693E+03   # ~mu_L
    2000013     2.89887794E+03   # ~mu_R
    1000014     2.11758362E+03   # ~nu_muL
    1000015     1.36688608E+03   # ~tau_1
    2000015     2.09421747E+03   # ~tau_2
    1000016     2.08741215E+03   # ~nu_tauL
    1000021     2.48835496E+03   # ~g
    1000022    -2.77273505E+02   # ~chi_10
    1000023     1.40756636E+03   # ~chi_20
    1000025    -2.59501462E+03   # ~chi_30
    1000035     2.59650584E+03   # ~chi_40
    1000024     1.40777746E+03   # ~chi_1+
    1000037     2.59750940E+03   # ~chi_2+
#
BLOCK NMIX  # Neutralino Mixing Matrix
  1  1     9.99843745E-01   # N_11
  1  2     9.86001678E-05   # N_12
  1  3    -1.75532385E-02   # N_13
  1  4     2.08784087E-03   # N_14
  2  1     5.79174395E-04   # N_21
  2  2     9.98920621E-01   # N_22
  2  3     4.11607778E-02   # N_23
  2  4     2.15185515E-02   # N_24
  3  1     1.38864108E-02   # N_31
  3  2    -1.38982121E-02   # N_32
  3  3     7.06789959E-01   # N_33
  3  4    -7.07150593E-01   # N_34
  4  1     1.09232383E-02   # N_41
  4  2    -4.43218130E-02   # N_42
  4  3     7.06006818E-01   # N_43
  4  4     7.06732363E-01   # N_44
#
BLOCK UMIX  # Chargino Mixing Matrix U
  1  1     9.98309328E-01   # U_11
  1  2     5.81247405E-02   # U_12
  2  1    -5.81247405E-02   # U_21
  2  2     9.98309328E-01   # U_22
#
BLOCK VMIX  # Chargino Mixing Matrix V
  1  1     9.99536459E-01   # V_11
  1  2    -3.04444771E-02   # V_12
  2  1    -3.04444771E-02   # V_21
  2  2    -9.99536459E-01   # V_22
#
BLOCK STOPMIX  # Stop Mixing Matrix
  1  1     6.39633389E-02   # cos(theta_t)
  1  2     9.97952249E-01   # sin(theta_t)
  2  1    -9.97952249E-01   # -sin(theta_t)
  2  2     6.39633389E-02   # cos(theta_t)
#
BLOCK SBOTMIX  # Sbottom Mixing Matrix
  1  1    -4.81137440E-01   # cos(theta_b)
  1  2     8.76645176E-01   # sin(theta_b)
  2  1    -8.76645176E-01   # -sin(theta_b)
  2  2    -4.81137440E-01   # cos(theta_b)
#
```

```
BLOCK STAUMIX  # Stau Mixing Matrix
  1  1    -9.03612153E-02   # cos(theta_tau)
  1  2     9.95909057E-01   # sin(theta_tau)
  2  1    -9.95909057E-01   # -sin(theta_tau)
  2  2    -9.03612153E-02   # cos(theta_tau)
#
BLOCK ALPHA  # Higgs mixing
          -1.84619735E-02   # Mixing angle in the neutral Higgs boson sector
#
BLOCK HMIX Q=  1.91221776E+03  # DRbar Higgs Parameters
         1    -2.61949999E+03   # mu(Q)MSSM
         2     5.54568052E+01   # tan
         3     2.43198302E+02   # higgs
         4     7.36203676E+06   # mA^2(Q)MSSM
#
BLOCK GAUGE Q=  1.91221776E+03  # The gauge couplings
     1     3.63510557E-01   # gprime(Q) DRbar
     2     6.36150202E-01   # g(Q) DRbar
     3     1.02688771E+00   # g3(Q) DRbar
#
BLOCK AU Q=  1.91221776E+03  # The trilinear couplings
  1  1    -2.27185000E+03   # A_u(Q) DRbar
  2  2    -2.27185000E+03   # A_c(Q) DRbar
  3  3    -2.27185000E+03   # A_t(Q) DRbar
#
BLOCK AD Q=  1.91221776E+03  # The trilinear couplings
  1  1     2.25917000E+03   # A_d(Q) DRbar
  2  2     2.25917000E+03   # A_s(Q) DRbar
  3  3     2.25917000E+03   # A_b(Q) DRbar
#
BLOCK AE Q=  1.91221776E+03  # The trilinear couplings
  1  1    -4.38541000E+03   # A_e(Q) DRbar
  2  2    -4.38541000E+03   # A_mu(Q) DRbar
  3  3    -4.38541000E+03   # A_tau(Q) DRbar
#
BLOCK Yu Q=  1.91221776E+03  # The Yukawa couplings
  1  1     0.00000000E+00   # y_u(Q) DRbar
  2  2     0.00000000E+00   # y_c(Q) DRbar
  3  3     8.43295535E-01   # y_t(Q) DRbar
#
BLOCK Yd Q=  1.91221776E+03  # The Yukawa couplings
  1  1     0.00000000E+00   # y_d(Q) DRbar
  2  2     0.00000000E+00   # y_s(Q) DRbar
  3  3     1.11838507E+00   # y_b(Q) DRbar
#
BLOCK Ye Q=  1.91221776E+03  # The Yukawa couplings
  1  1     0.00000000E+00   # y_e(Q) DRbar
  2  2     0.00000000E+00   # y_mu(Q) DRbar
  3  3     5.18542586E-01   # y_tau(Q) DRbar
#
BLOCK MSOFT Q=  1.91221776E+03  # The soft SUSY breaking masses at the scale Q
         1    -2.79720000E+02   # M_1(Q)
         2     1.37052000E+03   # M_2(Q)
         3     2.44299000E+03   # M_3(Q)
        21    -3.85194345E+06   # mH1^2(Q)
        22    -6.92209647E+06   # mH2^2(Q)
```

```
        31     2.11105000E+03   # meL(Q)
        32     2.11105000E+03   # mmuL(Q)
        33     2.08798000E+03   # mtauL(Q)
        34     2.89587000E+03   # meR(Q)
        35     2.89587000E+03   # mmuR(Q)
        36     1.37121000E+03   # mtauR(Q)
        41     2.07150000E+03   # mqL1(Q)
        42     2.07150000E+03   # mqL2(Q)
        43     2.64605999E+03   # mqL3(Q)
        44     8.28679989E+02   # muR(Q)
        45     8.28679989E+02   # mcR(Q)
        46     1.37830998E+03   # mtR(Q)
        47     1.21291999E+03   # mdR(Q)
        48     1.21291999E+03   # msR(Q)
        49     2.51870999E+03   # mbR(Q)
#
#
#
#                             =================
#                             |The decay table|
#                             =================
#
# - The QCD corrections to the decays gluino -> squark  + quark
#                                     squark -> gaugino + quark_prime
#                                     squark -> squark_prime + Higgs
#                                     squark -> gluino  + quark
#   are included.
#
# - The multi-body decays for the inos, stops and sbottoms are included.
#
# - The loop induced decays for the gluino, neutralinos and stops
#   are included.
#
# - The SUSY decays of the top quark are included.
#
#
#         PDG            Width
DECAY         6     1.53636039E+00   # top decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2           5        24   # BR(t ->  b    W+)
#
#         PDG            Width
DECAY   1000021     1.58854193E+02   # gluino decays
#          BR         NDA      ID1       ID2
     9.92548671E-03    2     1000001        -1   # BR(~g -> ~d_L  db)
     9.92548671E-03    2    -1000001         1   # BR(~g -> ~d_L* d )
     8.25757538E-02    2     2000001        -1   # BR(~g -> ~d_R  db)
     8.25757538E-02    2    -2000001         1   # BR(~g -> ~d_R* d )
     9.99818762E-03    2     1000002        -2   # BR(~g -> ~u_L  ub)
     9.99818762E-03    2    -1000002         2   # BR(~g -> ~u_L* u )
     1.15824125E-01    2     2000002        -2   # BR(~g -> ~u_R  ub)
     1.15824125E-01    2    -2000002         2   # BR(~g -> ~u_R* u )
     9.92548671E-03    2     1000003        -3   # BR(~g -> ~s_L  sb)
     9.92548671E-03    2    -1000003         3   # BR(~g -> ~s_L* s )
     8.25757538E-02    2     2000003        -3   # BR(~g -> ~s_R  sb)
     8.25757538E-02    2    -2000003         3   # BR(~g -> ~s_R* s )
```

```
     9.99818762E-03    2     1000004        -4   # BR(~g -> ~c_L  cb)
     9.99818762E-03    2    -1000004         4   # BR(~g -> ~c_L* c )
     1.15824125E-01    2     2000004        -4   # BR(~g -> ~c_R  cb)
     1.15824125E-01    2    -2000004         4   # BR(~g -> ~c_R* c )
     6.33528943E-02    2     1000006        -6   # BR(~g -> ~t_1  tb)
     6.33528943E-02    2    -1000006         6   # BR(~g -> ~t_1* t )
#
#         PDG            Width
DECAY   1000006     3.09601982E+00   # stop1 decays
#          BR         NDA      ID1       ID2
     9.99652297E-01    2     1000022         6   # BR(~t_1 -> ~chi_10 t )
     3.47702640E-04    2     1000024         5   # BR(~t_1 -> ~chi_1+ b )
#
#         PDG            Width
DECAY   2000006     2.83122459E+01   # stop2 decays
#          BR         NDA      ID1       ID2
     1.39258894E-02    2     1000022         6   # BR(~t_2 -> ~chi_10 t )
     1.87904046E-01    2     1000023         6   # BR(~t_2 -> ~chi_20 t )
     0.00000000E+00    2     1000025         6   # BR(~t_2 -> ~chi_30 t )
     3.87398835E-01    2     1000024         5   # BR(~t_2 -> ~chi_1+ b )
     1.14424626E-02    2     1000037         5   # BR(~t_2 -> ~chi_2+ b )
     8.47060594E-02    2     1000021         6   # BR(~t_2 -> ~g      t )
     3.45215108E-01    2     1000006        25   # BR(~t_2 -> ~t_1    h )
     1.40328967E-01    2     1000006        23   # BR(~t_2 -> ~t_1    Z )
    -1.70921367E-01    2     1000005        24   # BR(~t_2 -> ~b_1    W+)
#
#         PDG            Width
DECAY   1000005     7.01565115E+00   # sbottom1 decays
#          BR         NDA      ID1       ID2
     1.91595201E-01    2     1000022         5   # BR(~b_1 -> ~chi_10 b )
     2.11646492E-01    2     1000023         5   # BR(~b_1 -> ~chi_20 b )
     4.09575709E-01    2    -1000024         6   # BR(~b_1 -> ~chi_1- t )
     5.75454554E-03    2     1000021         5   # BR(~b_1 -> ~g      b )
     1.81428052E-01    2     1000006       -24   # BR(~b_1 -> ~t_1    W-)
#
#         PDG            Width
DECAY   2000005     4.06849092E+01   # sbottom2 decays
#          BR         NDA      ID1       ID2
     1.05232740E-02    2     1000022         5   # BR(~b_2 -> ~chi_10 b )
     9.74651457E-02    2     1000023         5   # BR(~b_2 -> ~chi_20 b )
     8.69118266E-03    2     1000025         5   # BR(~b_2 -> ~chi_30 b )
     8.14068968E-03    2     1000035         5   # BR(~b_2 -> ~chi_40 b )
     1.88950082E-01    2    -1000024         6   # BR(~b_2 -> ~chi_1- t )
     1.51987834E-01    2     1000021         5   # BR(~b_2 -> ~g      b )
     1.76390973E-02    2     1000005        25   # BR(~b_2 -> ~b_1    h )
     2.16365306E-01    2     1000005        23   # BR(~b_2 -> ~b_1    Z )
     3.00237389E-01    2     1000006       -24   # BR(~b_2 -> ~t_1    W-)
#
#         PDG            Width
DECAY   1000002     9.36307131E+00   # sup_L decays
#          BR         NDA      ID1       ID2
     3.19769419E-02    2     1000022         2   # BR(~u_L -> ~chi_10 u)
     3.22546171E-01    2     1000023         2   # BR(~u_L -> ~chi_20 u)
     6.45476887E-01    2     1000024         1   # BR(~u_L -> ~chi_1+ d)
#
#         PDG            Width
```

```
DECAY   2000002     1.86946578E+00   # sup_R decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2     1000022         2   # BR(~u_R -> ~chi_10 u)
#
#         PDG            Width
DECAY   1000001     9.36716074E+00   # sdown_L decays
#          BR         NDA      ID1       ID2
     3.19159967E-02    2     1000022         1   # BR(~d_L -> ~chi_10 d)
     3.22999869E-01    2     1000023         1   # BR(~d_L -> ~chi_20 d)
     6.45084135E-01    2    -1000024         2   # BR(~d_L -> ~chi_1- u)
#
#         PDG            Width
DECAY   2000001     7.05967507E-01   # sdown_R decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2     1000022         1   # BR(~d_R -> ~chi_10 d)
#
#         PDG            Width
DECAY   1000004     9.36307131E+00   # scharm_L decays
#          BR         NDA      ID1       ID2
     3.19769419E-02    2     1000022         4   # BR(~c_L -> ~chi_10 c)
     3.22546171E-01    2     1000023         4   # BR(~c_L -> ~chi_20 c)
     6.45476887E-01    2     1000024         3   # BR(~c_L -> ~chi_1+ s)
#
#         PDG            Width
DECAY   2000004     1.86946578E+00   # scharm_R decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2     1000022         4   # BR(~c_R -> ~chi_10 c)
#
#         PDG            Width
DECAY   1000003     9.36716074E+00   # sstrange_L decays
#          BR         NDA      ID1       ID2
     3.19159967E-02    2     1000022         3   # BR(~s_L -> ~chi_10 s)
     3.22999869E-01    2     1000023         3   # BR(~s_L -> ~chi_20 s)
     6.45084135E-01    2    -1000024         4   # BR(~s_L -> ~chi_1- c)
#
#         PDG            Width
DECAY   2000003     7.05967507E-01   # sstrange_R decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2     1000022         3   # BR(~s_R -> ~chi_10 s)
#
#         PDG            Width
DECAY   1000011     1.06620455E+01   # selectron_L decays
#          BR         NDA      ID1       ID2
     2.52416905E-01    2     1000022        11   # BR(~e_L -> ~chi_10 e-)
     2.49586638E-01    2     1000023        11   # BR(~e_L -> ~chi_20 e-)
     4.97996456E-01    2    -1000024        12   # BR(~e_L -> ~chi_1- nu_e)
#
#         PDG            Width
DECAY   2000011     1.49592881E+01   # selectron_R decays
#          BR         NDA      ID1       ID2
     9.99987294E-01    2     1000022        11   # BR(~e_R -> ~chi_10 e-)
     1.99611997E-07    2     1000023        11   # BR(~e_R -> ~chi_20 e-)
     7.75336772E-06    2     1000025        11   # BR(~e_R -> ~chi_30 e-)
     4.75308726E-06    2     1000035        11   # BR(~e_R -> ~chi_40 e-)
#
#         PDG            Width
```

```
DECAY   1000013     1.06620455E+01   # smuon_L decays
#          BR         NDA      ID1       ID2
     2.52416905E-01    2     1000022        13   # BR(~mu_L -> ~chi_10 mu-)
     2.49586638E-01    2     1000023        13   # BR(~mu_L -> ~chi_20 mu-)
     4.97996456E-01    2    -1000024        14   # BR(~mu_L -> ~chi_1- nu_mu)
#
#         PDG            Width
DECAY   2000013     1.49592881E+01   # smuon_R decays
#          BR         NDA      ID1       ID2
     9.99987294E-01    2     1000022        13   # BR(~mu_R -> ~chi_10 mu-)
     1.99611997E-07    2     1000023        13   # BR(~mu_R -> ~chi_20 mu-)
     7.75336772E-06    2     1000025        13   # BR(~mu_R -> ~chi_30 mu-)
     4.75308726E-06    2     1000035        13   # BR(~mu_R -> ~chi_40 mu-)
#
#         PDG            Width
DECAY   1000015     6.57764270E+00   # stau_1 decays
#          BR         NDA      ID1       ID2
     1.00000000E+00    2     1000022        15   # BR(~tau_1 -> ~chi_10  tau-)
#
#         PDG            Width
DECAY   2000015     1.90740165E+01   # stau_2 decays
#          BR         NDA      ID1       ID2
     1.41993510E-01    2     1000022        15   # BR(~tau_2 -> ~chi_10  tau-)
     1.30759963E-01    2     1000023        15   # BR(~tau_2 -> ~chi_20  tau-)
     2.60255265E-01    2    -1000024        16   # BR(~tau_2 -> ~chi_1-  nu_tau)
     2.35999521E-01    2     1000015        25   # BR(~tau_2 -> ~tau_1   h)
     2.30991741E-01    2     1000015        23   # BR(~tau_2 -> ~tau_1   Z)
#
#         PDG            Width
DECAY   1000012     1.06393901E+01   # snu_eL decays
#          BR         NDA      ID1       ID2
     2.52552322E-01    2     1000022        12   # BR(~nu_eL -> ~chi_10 nu_e)
     2.48913354E-01    2     1000023        12   # BR(~nu_eL -> ~chi_20 nu_e)
     4.98534324E-01    2     1000024        11   # BR(~nu_eL -> ~chi_1+ e-)
#
#         PDG            Width
DECAY   1000014     1.06393901E+01   # snu_muL decays
#          BR         NDA      ID1       ID2
     2.52552322E-01    2     1000022        14   # BR(~nu_muL -> ~chi_10 nu_mu)
     2.48913354E-01    2     1000023        14   # BR(~nu_muL -> ~chi_20 nu_mu)
     4.98534324E-01    2     1000024        13   # BR(~nu_muL -> ~chi_1+ mu-)
#
#         PDG            Width
DECAY   1000016     1.85934369E+01   # snu_tauL decays
#          BR         NDA      ID1       ID2
     1.42309791E-01    2     1000022        16   # BR(~nu_tauL -> ~chi_10 nu_tau)
     1.34004436E-01    2     1000023        16   # BR(~nu_tauL -> ~chi_20 nu_tau)
     2.69038309E-01    2     1000024        15   # BR(~nu_tauL -> ~chi_1+ tau-)
     4.54647465E-01    2    -1000015       -24   # BR(~nu_tauL -> ~tau_1+ W-)
#
#         PDG            Width
DECAY   1000024     9.80225591E-04   # chargino1+ decays
#          BR         NDA      ID1       ID2
     3.57691546E-01    2    -1000015        16   # BR(~chi_1+ -> ~tau_1+  nu_tau)
     6.42308454E-01    2     1000022        24   # BR(~chi_1+ -> ~chi_10  W+)
#
```

```
#         PDG            Width
DECAY   1000037     4.93281880E+01   # chargino2+ decays
#          BR         NDA      ID1       ID2
     6.85723152E-05    2     1000002        -1   # BR(~chi_2+ -> ~u_L     db)
     2.48760538E-04    2    -1000001         2   # BR(~chi_2+ -> ~d_L*    u )
     6.85723152E-05    2     1000004        -3   # BR(~chi_2+ -> ~c_L     sb)
     2.48760538E-04    2    -1000003         4   # BR(~chi_2+ -> ~s_L*    c )
     6.03105301E-01    2     1000006        -5   # BR(~chi_2+ -> ~t_1     bb)
     2.21003254E-05    2     1000012       -11   # BR(~chi_2+ -> ~nu_eL   e+  )
     2.21003254E-05    2     1000014       -13   # BR(~chi_2+ -> ~nu_muL  mu+ )
     1.76390030E-02    2     1000016       -15   # BR(~chi_2+ -> ~nu_tau1 tau+)
     8.00200129E-05    2    -1000011        12   # BR(~chi_2+ -> ~e_L+    nu_e)
     8.00200129E-05    2    -1000013        14   # BR(~chi_2+ -> ~mu_L+   nu_mu)
     7.18503312E-02    2    -1000015        16   # BR(~chi_2+ -> ~tau_1+  nu_tau)
     4.48403907E-04    2    -2000015        16   # BR(~chi_2+ -> ~tau_2+  nu_tau)
     8.56986885E-02    2     1000024        23   # BR(~chi_2+ -> ~chi_1+  Z )
     3.33167217E-02    2     1000022        24   # BR(~chi_2+ -> ~chi_10  W+)
     8.32106241E-02    2     1000023        24   # BR(~chi_2+ -> ~chi_20  W+)
     9.24717757E-02    2     1000024        25   # BR(~chi_2+ -> ~chi_1+  h )
     1.14202445E-02    2     1000022        37   # BR(~chi_2+ -> ~chi_10  H+)
#
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
#
#         PDG            Width
DECAY   1000023     9.68828356E-04   # neutralino2 decays
#          BR         NDA      ID1       ID2
     4.80788881E-01    2     1000022        23   # BR(~chi_20 -> ~chi_10   Z )
     1.58546267E-01    2     1000022        25   # BR(~chi_20 -> ~chi_10   h )
     6.64376373E-04    2     2000002        -2   # BR(~chi_20 -> ~u_R      ub)
     6.64376373E-04    2    -2000002         2   # BR(~chi_20 -> ~u_R*     u )
     1.94523794E-05    2     2000001        -1   # BR(~chi_20 -> ~d_R      db)
     1.94523794E-05    2    -2000001         1   # BR(~chi_20 -> ~d_R*     d )
     6.64376373E-04    2     2000004        -4   # BR(~chi_20 -> ~c_R      cb)
     6.64376373E-04    2    -2000004         4   # BR(~chi_20 -> ~c_R*     c )
     1.94523794E-05    2     2000003        -3   # BR(~chi_20 -> ~s_R      sb)
     1.94523794E-05    2    -2000003         3   # BR(~chi_20 -> ~s_R*     s )
     1.79081565E-01    2     1000015       -15   # BR(~chi_20 -> ~tau_1-   tau+)
     1.79081565E-01    2    -1000015        15   # BR(~chi_20 -> ~tau_1+   tau-)
#
#         PDG            Width
DECAY   1000025     4.86446281E+01   # neutralino3 decays
#          BR         NDA      ID1       ID2
     1.33244596E-02    2     1000022        23   # BR(~chi_30 -> ~chi_10   Z )
     7.89786098E-02    2     1000023        23   # BR(~chi_30 -> ~chi_20   Z )
     8.43310183E-02    2     1000024       -24   # BR(~chi_30 -> ~chi_1+   W-)
     8.43310183E-02    2    -1000024        24   # BR(~chi_30 -> ~chi_1-   W+)
     2.19069500E-02    2     1000022        25   # BR(~chi_30 -> ~chi_10   h )
     7.69168156E-03    2     1000022        35   # BR(~chi_30 -> ~chi_10   H )
     3.93061373E-03    2     1000022        36   # BR(~chi_30 -> ~chi_10   A )
     8.08133864E-03    2     1000023        25   # BR(~chi_30 -> ~chi_20   h )
     5.41001456E-06    2     1000002        -2   # BR(~chi_30 -> ~u_L      ub)
     5.41001456E-06    2    -1000002         2   # BR(~chi_30 -> ~u_L*     u )
     2.65638567E-05    2     2000002        -2   # BR(~chi_30 -> ~u_R      ub)
     2.65638567E-05    2    -2000002         2   # BR(~chi_30 -> ~u_R*     u )
     1.16412962E-05    2     1000001        -1   # BR(~chi_30 -> ~d_L      db)
```

```
     1.16412962E-05    2    -1000001         1    # BR(~chi_30 -> ~d_L*     d )
     5.12757102E-06    2     2000001        -1    # BR(~chi_30 -> ~d_R      db)
     5.12757102E-06    2    -2000001         1    # BR(~chi_30 -> ~d_R*     d )
     5.41001456E-06    2     1000004        -4    # BR(~chi_30 -> ~c_L      cb)
     5.41001456E-06    2    -1000004         4    # BR(~chi_30 -> ~c_L*     c )
     2.65638567E-05    2     2000004        -4    # BR(~chi_30 -> ~c_R      cb)
     2.65638567E-05    2    -2000004         4    # BR(~chi_30 -> ~c_R*     c )
     1.16412962E-05    2     1000003        -3    # BR(~chi_30 -> ~s_L      sb)
     1.16412962E-05    2    -1000003         3    # BR(~chi_30 -> ~s_L*     s )
     5.12757102E-06    2     2000003        -3    # BR(~chi_30 -> ~s_R      sb)
     5.12757102E-06    2    -2000003         3    # BR(~chi_30 -> ~s_R*     s )
     2.96761481E-01    2     1000006        -6    # BR(~chi_30 -> ~t_1      tb)
     2.96761481E-01    2    -1000006         6    # BR(~chi_30 -> ~t_1*     t )
     6.07724177E-03    2     1000005        -5    # BR(~chi_30 -> ~b_1      bb)
     6.07724177E-03    2    -1000005         5    # BR(~chi_30 -> ~b_1*     b )
     4.23364823E-07    2     1000011       -11    # BR(~chi_30 -> ~e_L-     e+)
     4.23364823E-07    2    -1000011        11    # BR(~chi_30 -> ~e_L+     e-)
     4.23364823E-07    2     1000013       -13    # BR(~chi_30 -> ~mu_L-    mu+)
     4.23364823E-07    2    -1000013        13    # BR(~chi_30 -> ~mu_L+    mu-)
     3.70571072E-02    2     1000015       -15    # BR(~chi_30 -> ~tau_1-   tau+)
     3.70571072E-02    2    -1000015        15    # BR(~chi_30 -> ~tau_1+   tau-)
     8.70369064E-03    2     2000015       -15    # BR(~chi_30 -> ~tau_2-   tau+)
     8.70369064E-03    2    -2000015        15    # BR(~chi_30 -> ~tau_2+   tau-)
     5.71363936E-06    2     1000012       -12    # BR(~chi_30 -> ~nu_eL    nu_eb)
     5.71363936E-06    2    -1000012        12    # BR(~chi_30 -> ~nu_eL*   nu_e )
     5.71363936E-06    2     1000014       -14    # BR(~chi_30 -> ~nu_muL   nu_mub)
     5.71363936E-06    2    -1000014        14    # BR(~chi_30 -> ~nu_muL*  nu_mu )
     6.37617464E-06    2     1000016       -16    # BR(~chi_30 -> ~nu_tau1  nu_taub)
     6.37617464E-06    2    -1000016        16    # BR(~chi_30 -> ~nu_tau1* nu_tau )
#
#         PDG            Width
DECAY   1000035     5.02010166E+01   # neutralino4 decays
#          BR         NDA      ID1       ID2
     2.07891141E-02    2     1000022        23    # BR(~chi_40 -> ~chi_10   Z )
     7.62462866E-03    2     1000023        23    # BR(~chi_40 -> ~chi_20   Z )
     8.19791784E-02    2     1000024       -24    # BR(~chi_40 -> ~chi_1+   W-)
     8.19791784E-02    2    -1000024        24    # BR(~chi_40 -> ~chi_1-   W+)
     1.28317338E-02    2     1000022        25    # BR(~chi_40 -> ~chi_10   h )
     3.80748096E-03    2     1000022        35    # BR(~chi_40 -> ~chi_10   H )
     7.43628595E-03    2     1000022        36    # BR(~chi_40 -> ~chi_10   A )
     8.30459283E-02    2     1000023        25    # BR(~chi_40 -> ~chi_20   h )
     7.42022225E-05    2     1000002        -2    # BR(~chi_40 -> ~u_L      ub)
     7.42022225E-05    2    -1000002         2    # BR(~chi_40 -> ~u_L*     u )
     1.59392048E-05    2     2000002        -2    # BR(~chi_40 -> ~u_R      ub)
     1.59392048E-05    2    -2000002         2    # BR(~chi_40 -> ~u_R*     u )
     8.91553384E-05    2     1000001        -1    # BR(~chi_40 -> ~d_L      db)
     8.91553384E-05    2    -1000001         1    # BR(~chi_40 -> ~d_L*     d )
     3.07790320E-06    2     2000001        -1    # BR(~chi_40 -> ~d_R      db)
     3.07790320E-06    2    -2000001         1    # BR(~chi_40 -> ~d_R*     d )
     7.42022225E-05    2     1000004        -4    # BR(~chi_40 -> ~c_L      cb)
     7.42022225E-05    2    -1000004         4    # BR(~chi_40 -> ~c_L*     c )
     1.59392048E-05    2     2000004        -4    # BR(~chi_40 -> ~c_R      cb)
     1.59392048E-05    2    -2000004         4    # BR(~chi_40 -> ~c_R*     c )
     8.91553384E-05    2     1000003        -3    # BR(~chi_40 -> ~s_L      sb)
     8.91553384E-05    2    -1000003         3    # BR(~chi_40 -> ~s_L*     s )
     3.07790320E-06    2     2000003        -3    # BR(~chi_40 -> ~s_R      sb)
```

```
     3.07790320E-06    2    -2000003         3   # BR(~chi_40 -> ~s_R*     s )
     3.00151859E-01    2     1000006        -6   # BR(~chi_40 -> ~t_1      tb)
     3.00151859E-01    2    -1000006         6   # BR(~chi_40 -> ~t_1*     t )
     5.48118281E-03    2     1000005        -5   # BR(~chi_40 -> ~b_1      bb)
     5.48118281E-03    2    -1000005         5   # BR(~chi_40 -> ~b_1*     b )
     1.68157959E-05    2     1000011       -11   # BR(~chi_40 -> ~e_L-     e+)
     1.68157959E-05    2    -1000011        11   # BR(~chi_40 -> ~e_L+     e-)
     1.68157959E-05    2     1000013       -13   # BR(~chi_40 -> ~mu_L-    mu+)
     1.68157959E-05    2    -1000013        13   # BR(~chi_40 -> ~mu_L+    mu-)
     3.56274471E-02    2     1000015       -15   # BR(~chi_40 -> ~tau_1-   tau+)
     3.56274471E-02    2    -1000015        15   # BR(~chi_40 -> ~tau_1+   tau-)
     8.54299495E-03    2     2000015       -15   # BR(~chi_40 -> ~tau_2-   tau+)
     8.54299495E-03    2    -2000015        15   # BR(~chi_40 -> ~tau_2+   tau-)
     2.98476823E-05    2     1000012       -12   # BR(~chi_40 -> ~nu_eL    nu_eb)
     2.98476823E-05    2    -1000012        12   # BR(~chi_40 -> ~nu_eL*   nu_e )
     2.98476823E-05    2     1000014       -14   # BR(~chi_40 -> ~nu_muL   nu_mub)
     2.98476823E-05    2    -1000014        14   # BR(~chi_40 -> ~nu_muL*  nu_mu )
     3.32965313E-05    2     1000016       -16   # BR(~chi_40 -> ~nu_tau1  nu_taub)
     3.32965313E-05    2    -1000016        16   # BR(~chi_40 -> ~nu_tau1* nu_tau )
#
#         PDG            Width
DECAY        25     3.46773784E-03   # h decays
#          BR         NDA      ID1       ID2
     6.28813991E-01    2           5        -5   # BR(h -> b       bb     )
     7.63585799E-02    2         -15        15   # BR(h -> tau+    tau-    )
     2.70297439E-04    2         -13        13   # BR(h -> mu+     mu-     )
     5.24573932E-04    2           3        -3   # BR(h -> s       sb      )
     2.18458526E-02    2           4        -4   # BR(h -> c       cb      )
     7.44711826E-02    2          21        21   # BR(h -> g       g       )
     2.46302483E-03    2          22        22   # BR(h -> gam     gam     )
     1.36571319E-03    2          22        23   # BR(h -> Z       gam     )
     1.73340543E-01    2          24       -24   # BR(h -> W+      W-      )
     2.05462412E-02    2          23        23   # BR(h -> Z       Z       )
#
#         PDG            Width
DECAY        35     1.60555574E+02   # H decays
#          BR         NDA      ID1       ID2
     9.33265962E-01    2           5        -5   # BR(H -> b       bb      )
     6.60201269E-02    2         -15        15   # BR(H -> tau+    tau-    )
     2.33406224E-04    2         -13        13   # BR(H -> mu+     mu-     )
     2.81432854E-04    2           3        -3   # BR(H -> s       sb      )
     1.39912134E-09    2           4        -4   # BR(H -> c       cb      )
     1.57366362E-04    2           6        -6   # BR(H -> t       tb      )
     2.46002601E-05    2          21        21   # BR(H -> g       g       )
     1.48902368E-09    2          22        22   # BR(H -> gam     gam     )
     1.29605758E-08    2          23        22   # BR(H -> Z       gam     )
     1.72914254E-06    2          24       -24   # BR(H -> W+      W-      )
     8.61104792E-07    2          23        23   # BR(H -> Z       Z       )
     8.03255121E-06    2          25        25   # BR(H -> h       h       )
     6.46657640E-06    2     1000022   1000022   # BR(H -> ~chi_10 ~chi_10)
#
#         PDG            Width
DECAY        36     1.60579488E+02   # A decays
#          BR         NDA      ID1       ID2
     9.33268295E-01    2           5        -5   # BR(A -> b       bb      )
     6.60196295E-02    2         -15        15   # BR(A -> tau+    tau-    )
```

```
     2.33403403E-04    2          -13        13   # BR(A -> mu+     mu-    )
     2.81430740E-04    2            3        -3   # BR(A -> s       sb     )
     1.34683068E-09    2            4        -4   # BR(A -> c       cb     )
     1.55009031E-04    2            6        -6   # BR(A -> t       tb     )
     3.31552474E-05    2           21        21   # BR(A -> g       g      )
     8.25194910E-09    2           22        22   # BR(A -> gam     gam    )
     1.87006980E-08    2           23        22   # BR(A -> Z       gam    )
     1.71000177E-06    2           23        25   # BR(A -> Z       h      )
     7.33896664E-06    2      1000022   1000022   # BR(A -> ~chi_10 ~chi_10)
#
#         PDG            Width
DECAY        37     1.97135865E+02   # H+ decays
#          BR         NDA      ID1       ID2
     1.52038588E-03    2            4        -5   # BR(H+ -> c       bb     )
     5.38289898E-02    2          -15        16   # BR(H+ -> tau+    nu_tau )
     1.90305057E-04    2          -13        14   # BR(H+ -> mu+     nu_mu  )
     9.73043231E-06    2            2        -5   # BR(H+ -> u       bb     )
     1.09559517E-05    2            2        -3   # BR(H+ -> u       sb     )
     2.25338242E-04    2            4        -3   # BR(H+ -> c       sb     )
     9.44212895E-01    2            6        -5   # BR(H+ -> t       bb     )
     1.39986354E-06    2           24        25   # BR(H+ -> W+      h      )
     7.08967058E-14    2           24        36   # BR(H+ -> W+      A      )
```

5.2 Appendix B - hepmc2root.py Conversion File

```
#!/usr/bin/env python
# ------------------------------------------------------------------------
# File: hepmc2root.py
# Description: write events in HepMC2 format to a flat ROOT ntuple using
#              variable length arrays
#
#
#      status = +- (10 * i + j)
#      + : still remaining particles
#      - : decayed/branched/fragmented/... and not remaining
#      i = 1 - 9 : stage of event generation inside PYTHIA
#      i = 10 -19 : reserved for future expansion
#      i >= 20 : free for add-on programs
#      j = 1 - 9 : further specification
#
# In detail, the list of used or foreseen status codes is:
#
#      11 - 19 : beam particles
#          11 : the event as a whole
#          12 : incoming beam
#          13 : incoming beam-inside-beam (e.g. gamma inside e)
#          14 : outgoing elastically scattered
#          15 : outgoing diffractively scattered
#      21 - 29 : particles of the hardest subprocess
#          21 : incoming
#          22 : intermediate (intended to have preserved mass)
#          23 : outgoing
#          24 : outgoing, nonperturbatively kicked out in diffraction
#
# Created: fall   2017 Harrison B. Prosper
# Updated: 04-Dec-2017 HBP add creation vertex (x,y,z) of particles.
# ------------------------------------------------------------------------
import os, sys, ROOT
from string import split, strip, atoi, atof, upper
from math import sqrt
from time import ctime
from pnames import particleName
# ------------------------------------------------------------------------
def nameonly(s):
    import posixpath
    return posixpath.splitext(posixpath.split(s)[1])[0]

TREENAME= "Events"
MAXPART = 5000
debug = 0

class hepmcstream:

    def __init__(self, filename, outfilename=None, treename=TREENAME, complevel=2):

        # check that file exists

        if not os.path.exists(filename):
            sys.exit("** hepmcstream: can't open file %s" % filename)
        self.inp = open(filename)
```

```
inp = self.inp

# get version number of HepMC

self.header = [] # cache HepMC header
version = None
for line in inp:
    self.header.append(line)
    version = strip(line)
    if version == '': continue
    token = split(version)
    if token[0] == 'HepMC::Version':
        version = token[1]
    break
else:
    sys.exit("** hepmcstream: format problem in file %s" % filename)

print "HepMC version: %s" % version

# skip start of listing

for line in inp:
    self.header.append(line)
    break

# open output root file

if outfilename == None:
    outfilename = '%s.root' % nameonly(filename)

self.file = ROOT.TFile(outfilename, "recreate")
self.tree = ROOT.TTree(treename, 'created: %s HepMC %s' % (ctime(), version))

# define event struct

self.struct = '''struct Bag {
int    Event_number;
int    Event_numberMP;
double Event_scale;
double Event_alphaQCD;
double Event_alphaQED;
int    Event_barcodeSPV;
int    Event_numberV;
int    Event_barcodeBP1;
int    Event_barcodeBP2;
int    Event_numberP;

double Xsection_value;
double Xsection_error;

int    PDF_parton1;
int    PDF_parton2;
double PDF_x1;
double PDF_x2;
double PDF_Q2;
double PDF_x1f;
```

```
        double PDF_x2f;
        int    PDF_id1;
        int    PDF_id2;

        double Particle_x[%(size)d];
        double Particle_y[%(size)d];
        double Particle_z[%(size)d];
        double Particle_ctau[%(size)d];

        double Particle_barcode[%(size)d];
        int    Particle_pid[%(size)d];
        double Particle_px[%(size)d];
        double Particle_py[%(size)d];
        double Particle_pz[%(size)d];
        double Particle_energy[%(size)d];
        double Particle_mass[%(size)d];
        int    Particle_status[%(size)d];
        int    Particle_d1[%(size)d];
        int    Particle_d2[%(size)d];
};''' % {'size': MAXPART}

        # indices to vertices

        self.pvertex = [0]*MAXPART

        # create struct

        ROOT.gROOT.ProcessLine(self.struct)
        from ROOT import Bag
        self.bag = Bag()

        # create branches

        self.branch = []
        recs = split(self.struct, '\n')[1:-1]
        for rec in recs:
            t = split(rec)
            if len(t) == 0: continue

            fmt, name = t
            T = upper(fmt[0])
            name = name[:-1] # skip ";"
            # check for variable length array
            if name[-1] == ']':
                field = split(name, '[')[0]
                fmt   = '%s[Event_numberP]/%s' % (field, T)
            else:
                field = name
                fmt   = '%s/%s' % (field, T)
            self.branch.append(self.tree.Branch(field,
                                                ROOT.AddressOf(self.bag, field),
                                                fmt))
        # list branches

        for ii, b in enumerate(self.branch):
            bname = b.GetName()
```

```
            leaves= b.GetListOfLeaves()
            if leaves == None:
                sys.exit("** hepmcstream: no list of leaves found for branch %s" % bname)
            leaf = leaves[0]
            if leaf == None:
                sys.exit("** hepmcstream: no leaf found for branch %s" % bname)
            leafname = leaf.GetName()
            leaftype = leaf.GetTypeName()
            print "%4d\t%-20s\t%s" % (ii+1, bname, leaftype)

    def __del__(self):
        self.tree.Write("", ROOT.TObject.kOverwrite)

    def __str__(self, index):
        bag = self.bag
        d   = " <%4d, %4d>" % (bag.Particle_d1[index], bag.Particle_d2[index])
        px  = bag.Particle_px[index]
        py  = bag.Particle_py[index]
        pt  = sqrt(px**2+py**2)
        rec = '%-14s %7d %4d %3d %7.1f (%7.1f, %7.1f, %7.1f, %7.1f)%s' \
          % (particleName(bag.Particle_pid[index]),
             bag.Particle_pid[index],
             bag.Particle_barcode[index],
             bag.Particle_status[index],
             pt,
             bag.Particle_energy[index],
             bag.Particle_px[index],
             bag.Particle_py[index],
             bag.Particle_pz[index],
             d)
        return rec

    def __call__(self):
        inp = self.inp
        bag = self.bag

        self.event = [] # cache HepMC event in original format

        # find start of event

        token = None
        for line in inp:
            self.event.append(line)
            token = split(line)
            key   = token[0]
            if key != 'E': continue
            if debug > 0:
                print 'BEGIN event'
            break
        else:
            return False

        if token == None:
            sys.exit("** hepmcstream: can't find start of event")

        bag.Event_number     = atoi(token[1])
```

```
bag.Event_numberMP   = atoi(token[2])  # number of multi-particle interactions
bag.Event_scale      = atof(token[3])
bag.Event_alphaQCD   = atof(token[4])
bag.Event_alphaQED   = atof(token[5])
bag.Event_processID  = atoi(token[6])
bag.Event_barcodeSPV = atoi(token[7])
bag.Event_numberV    = atoi(token[8])  # number of vertices in event
bag.Event_barcodeBP1 = atoi(token[9])  # barcode beam particle 1
bag.Event_barcodeBP2 = atoi(token[10]) # barcode beam particle 2
bag.Event_numberP    = 0               # number of particles

if debug > 0:
    print "\tbarcode 1: %d" % self.barcode1
    print "\tbarcode 2: %d" % self.barcode2

self.vertex = {}

for line in inp:
    self.event.append(line)
    token = split(line)
    key = token[0]

    if key == 'C':
        # CROSS SECTION
        bag.Xsection_value = atof(token[1])
        bag.Xsection_error = atof(token[2])
        if debug > 0:
            print "\tcross section: %10.3e +\- %10.3e pb" % \
              (bag.Xsection_value, bag.Xsection_error)

    elif key == 'F':
        # PDF INFO
        bag.PDF_parton1  = atoi(token[1])
        bag.PDF_parton2  = atoi(token[2])
        bag.PDF_x1       = atof(token[3])
        bag.PDF_x2       = atof(token[4])
        bag.PDF_Q2       = atof(token[5])
        bag.PDF_x1f      = atof(token[6])
        bag.PDF_x2f      = atof(token[7])
        bag.PDF_id1      = atoi(token[8])
        bag.PDF_id2      = atoi(token[9])

        if debug > 0:
            print '\tfound PDF info'

    elif key == 'V':
        # VERTEX
        vbarcode = atoi(token[1])
        self.vertex[vbarcode] = [-1, -1]
        x    = atof(token[3])
        y    = atof(token[4])
        z    = atof(token[5])
        ctau = atof(token[6])
        nout = atoi(token[8])
        if debug > 0:
            if debug > 1:
```

```
                print "\t%s" % token
            print '\tvertex(barcode): %10d' % vbarcode
            print '\tvertex(count):   %10d' % nout

        # particles pertaining to this vertex follow immediately
        # after the vertex
        for ii in xrange(nout):
            for line in inp:
                self.event.append(line)
                token  = split(line)
                if debug > 1:
                    print "\t%s" % token
                key    = token[0]
                if key != 'P':
                    sys.exit("** hepmcstream: faulty event record\n" + line)

                if bag.Event_numberP < MAXPART:
                    index = bag.Event_numberP
                    bag.Event_numberP += 1

                    bag.Particle_x[index]       = x
                    bag.Particle_y[index]       = y
                    bag.Particle_z[index]       = z
                    bag.Particle_ctau[index]    = ctau

                    bag.Particle_barcode[index] = atoi(token[1])
                    bag.Particle_pid[index]     = atoi(token[2])
                    bag.Particle_px[index]      = atof(token[3])
                    bag.Particle_py[index]      = atof(token[4])
                    bag.Particle_pz[index]      = atof(token[5])
                    bag.Particle_energy[index]  = atof(token[6])
                    bag.Particle_mass[index]    = atof(token[7])
                    bag.Particle_status[index]  = atoi(token[8])
                    self.pvertex[index]         = atoi(token[11])

                    if ii == 0:
                        self.vertex[vbarcode][0] = index
                    else:
                        self.vertex[vbarcode][1] = index

                break
            else:
                return False

    if len(self.vertex) >= bag.Event_numberV:
        for index in xrange(bag.Event_numberP):
            code = self.pvertex[index]
            if self.vertex.has_key(code):
                d = self.vertex[code]
                bag.Particle_d1[index] = d[0]
                bag.Particle_d2[index] = d[1]
            else:
                bag.Particle_d1[index] = -1
                bag.Particle_d2[index] = -1

        # fill ntuple
```

```
                self.file.cd()
                self.tree.Fill()

                return True
        else:
            return False

    def printTable(self):
        for ii in xrange(self.bag.Event_numberP):
            print "%4d\t%s" % (ii, self.__str__(ii))
# ---------------------------------------------------------------------------
def main():
    argv = sys.argv[1:]
    argc = len(argv)
    if argc < 1:
        sys.exit('''
    Usage:
        ./hepmc2root.py <HepMC-file> [output root file = <name>.root]
        ''')

    filename = argv[0]
    if argc > 1:
        outfilename = argv[1]
    else:
        outfilename = '%s.root' % nameonly(filename)

    stream = hepmcstream(filename, outfilename)

    ii = 0
    while stream():
        if ii % 100 == 0:
            print ii
        ii += 1
# ---------------------------------------------------------------------------
try:
    main()
except KeyboardInterrupt:
    print '\nciao!'
```

5.3 Appendix C - analyzer.py Histogram File

```
#!/usr/bin/env python
# ---------------------------------------------------------------------------
#  File:        analyzer.py
#  Description: Analyzer for simple ntuples, such as those created by
#               TheNtupleMaker
#  Created:     Mon Dec  4 00:14:59 2017 by mkanalyzer.py
#  Author:      Shakespeare's ghost
# ---------------------------------------------------------------------------
import os, sys, re
from tnm import *
from ROOT import *
from math import cos
# ---------------------------------------------------------------------------
# -- Constants, procedures and functions
# ---------------------------------------------------------------------------


# ---------------------------------------------------------------------------
def get_Higgs_location(event):

    HIGGS = 35
    TAU = 15

    for row in xrange(event.Event_numberP):
        PID = event.Particle_pid[row]
        if PID != HIGGS: continue
        #print event.Particle_mass[row]
        ii = event.Particle_d1[row]
        did = event.Particle_pid[ii]
        if abs(did) != TAU: continue
        return row

    return -1

def create_4vector(event, row):

    px = event.Particle_px[row]
    py = event.Particle_py[row]
    pz = event.Particle_pz[row]
    energy = event.Particle_energy[row]
    return TLorentzVector(px, py, pz, energy)

def main():

    cl = commandLine()

    # Get names of ntuple files to be processed
    filenames = fileNames(cl.filelist)

    # Create tree reader
    stream = itreestream(filenames, "Events")
    if not stream.good():
        error("can't read input files")

    # Create a buffer to receive events from the stream
```

```
ev = eventBuffer(stream)

nevents = ev.size()
print "number of events:", nevents

# Create file to store histograms
of = outputFile(cl.outputfilename)

# ---------------------------------------------------------------------------
# Define histograms
# ---------------------------------------------------------------------------
setStyle()

hHiggsPt = mkhist1('higgsPt', 'P_{T} (GeV)', '', 50, 0, 1000)
hTau1Pt = mkhist1('Tau1Pt', 'P_{T} (GeV)', '', 50, 0, 2000)
hTau2Pt = mkhist1('Tau2Pt', 'P_{T} (GeV)', '', 50, 0, 2000)
hdphi = mkhist1('delataPhi', '', '', 20, -1, 1)
hHiggsmass = mkhist1('HiggsMass', 'Mass (GeV)', '', 50, 0, 3000)

# ---------------------------------------------------------------------------
# Loop over events
# ---------------------------------------------------------------------------

for entry in xrange(nevents):
    ev.read(entry)

    # Uncomment the following line if you wish to copy variables into
    # structs. See the header eventBuffer.h to find out what structs
    # are available. Alternatively, you can call individual fill
    # functions, such as ev.fillJets().
    #ev.fillObjects();

    # analysis

    index = get_Higgs_location(ev)
    if index < -1:
        sys.exit('Could not find a SUSY Higgs.')

    higgs = create_4vector(ev, index)
    d1 = ev.Particle_d1[index]
    d2 = ev.Particle_d2[index]
    Tau1 = create_4vector(ev, d1)
    Tau2 = create_4vector(ev, d2)

    if not (Tau1.Pt() >= 500.0 and Tau2.Pt() >= 500): continue
    pt = higgs.Pt()
    hHiggsPt.Fill(pt)
    hHiggsmass.Fill(higgs.M())

    hTau1Pt.Fill(Tau1.Pt())
    hTau2Pt.Fill(Tau2.Pt())
    hdphi.Fill(cos(Tau1.DeltaPhi(Tau2)))


ev.close()
of.close()
```

```
# ----------------------------------------------------------------------------
try:
   main()
except KeyboardInterrupt:
   print "bye!"
```