LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-738295

Manual for the Jet Event and Background Simulation Library

M. Heinz, R. Soltz, A. Angerami

September 12, 2017

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed 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 Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Manual for the Jet Event and Background Simulation Library (JEBSimLib)

LLNL-TR-xxxxxx [1]

Matthias Heinz, Ron Soltz, and Aaron Angerami

Abstract

Jets are the collimated streams of particles resulting from hard scattering in the initial state of high-energy collisions. In heavy-ion collisions, jets interact with the quark-gluon plasma (QGP) before freezeout, providing a probe into the internal structure and properties of the QGP. In order to study jets, background must be subtracted from the measured event, potentially introducing a bias. We aim to understand and quantify this subtraction bias. PYTHIA, a library to simulate pure jet events, is used to simulate a model for a signature with one pure jet (a photon) and one quenched jet, where all quenched particle momenta are reduced by a user-defined constant fraction. Background for the event is simulated using multiplicity values generated by the TRENTO initial state model of heavy-ion collisions fed into a thermal model consisting of a 3-dimensional Boltzmann distribution for particle types and momenta. Data from the simulated events is used to train a statistical model, which computes a posterior distribution of the quench factor for a data set. The model was tested first on pure jet events and then on full events including the background. This model will allow for a quantitative determination of biases induced by various methods of background subtraction.

1 Motivation

2 Jet Event

2.1 Model

The jets are produced using PYTHIA8 [5], a standard tool for Monte Carlo generation of hard scattering particle interactions based on the Lund string fragmentation model [7]. JEBSimLib simply provides a wrapper around a PYTHIA instance with reasonable defaults. In addition to the pure jet model provided by PYTHIA, a model for the interaction of the jet with the quark-gluon plasma (QGP) medium is required to create the quenching behavior we want to detect later on. The model used applies a linear quenching factor to all particles generated by the chosen quenched jet, where a jet is defined as the particles resulting from the fragmentation and decays of an outgoing particle of the hard process. If a photon is produced by the hard process, the other jet will be quenched. If not, one of the two jets will be selected at random to be quenched.

2.2 Implementation

The PYTHIA wrapper is called `EventGenerator`. Creating the object initializes the PYTHIA settings, allowing the user to then quickly generate many similar events without redoing the setup. The final container for the data will be the HepMC format [9], which can be written to a file and read back in later for use elsewhere. The `generate` method has two overloads: the first generates the event, applies quenching, converts the data to HepMC, and writes this to a file, returning the filename as a string; the second accepts a pointer to an empty HepMC event object and loads the data into this object instead

[1] This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

of writing the data to a file. The first method allows users to generate data to later modify and handle as needed, while the second allows users that need to immediately process the data to skip writing and reading the data. After initialization, the only aspect of the `EventGenerator` state that can be changed is the quench factor through the `set_quench` method.

2.3 Possible Future Work

At some point, it may be valuable to work with a more sophisticated model for the jet's interaction with the QGP medium. Changing this should not be difficult, but it may demand changes to the initialization and internal structure of the `EventGenerator` object along with the implementation of the model. Another change that may be desired is to have all non-photon jets interact with the QGP medium, which would be trivial to change.

3 Background (Underlying Event)

3.1 Model

The model for the underlying event works in three steps. First, the entropy generated by a run of the TRENTO initial state model for heavy-ion collisions [2]. This is multiplied by two factors to get the event multiplicity (after rounding): a scale-factor determined by the ion type and the collision energy; and the ratio of total particles to charged particles at freezeout. Given the multiplicity, we can sample the particle types from a thermal distribution [1]. The probabilities are given by:

$$P_i(T, \vec{\mu}) = \frac{n_i(T, \vec{\mu})}{\sum_i n_i(T, \vec{\mu})} \tag{1}$$

T is the temperature at freezeout, and $\vec{\mu}$ is the chemical potential at freezeout. For our purposes, we assume that freeze-out occurs under conditions of zero chemical potential ($\vec{\mu} = 0$). As a result:

$$n_i(T, 0) = \frac{T g_i}{2\pi^2} \sum_{k=1}^{\infty} \frac{(\pm 1)^{k+1}}{k} m_i^2 K_2(\frac{k m_i}{T}) \tag{2}$$

where g_i is the spin-isospin degeneracy, m_i is the mass of the particle, the $\pm$ becomes $(+)$ for fermions and $(-)$ for bosons, and K_2 is the modified Bessel function of the second kind. After sampling from this distribution, the particle momenta are sampled from a 3-D Boltzmann distribution. This sampling approach will produce a background that is on average uniform in ϕ and η, but has certain desirable features like hotspots from resonance decays.

3.2 Implementation

To simplify the background generation, we provide a `BackgroundGenerator` class. The initialization sets up the comamnd for the TRENTO shell process and initializes a PYTHIA object to handle the decays of unstable particles. It also uses the temperature to generate a distribution of particle types from which to sample particles. To construct this distribution, the program loads in a list of allowed particles under a mass of 2.5 GeV. The particle list is taken from the Particle Data Group listing [8], restricted by the subset that have decay processes defined in PYTHIA. The particles with masses under 2.5 GeV that are not properly handled and thus excluded are on the heavy side and as a result would contribute little to the final distribution of particle types. Given the particles to include in the distribution, densities and probabilties are calculated, allowing the construction of a cumulative mass function from which particles can be sampled. The infinite sum is truncated when additional terms contribute proportionally negligible amounts to the sum ($epsilon = 10^{-8}$). An initial estimate of the ratio of total particles to charged particles is computed from this distribution, but can be improved upon by running a Monte-Carlo of 10000 sampled particles, decaying them, and observing the final ratio.

Again, `generate` supports two overloads, one to save the data to a HepMC file and return the filename, and one to just save the data in a HepMC event object provided by the user. TRENTO is run as a shell process, so each run of `generate` must open a shell process, pipe the output to a string, and parse the entropy from that string. This seems to be one of the main performance sinks for the background generation. TRENTO entropy values are converted to charged particle multiplicities using a scale-factor determined from fits to experimental data for charged particle multiplicites. These are then converted to total multiplicites using the ratio of total particle to charged particles. The ratio of total particles to charged particles is provided by the CMF object, which is the object from which particles are sampled. Given the entropy and these factors, the multiplicity n is calculated. n particles are then sampled from the CMF, are given momenta as sampled from the 3-D Boltzmann distribution, and are placed into a blank PYTHIA event. PYTHIA is then run to handle the decays of unstable particles and the final event record is converted into the HepMC format for the user.

3.3 Possible Future Work

The current background implementation has no flow. However, TRENTO does provide eccentricities in its output to make implementing flow in the background possible by rescaling momenta relative to the event plane. Current scale-factors for the multiplicity are only provided for Au-Au collisions at RHIC energies. This will need to be extended to LHC Pb-Pb collisions. Also, they are only given for specific values and not as a function of energy. This could be changed given a model that matches current data well and seems to be well-behaved enough to allow for interpolation.

4 Utilities

There are various utilities included in `Utilities.h` that help with the transition from JEBSimLib to FastJet [3] and help with tracking certain important information.

4.1 Jet Preparation

Two methods are provided to convert a HepMC event record into a PseudoJet vector for FastJet. The first, `prep_single_event`, simply takes the undecayed particles in the HepMC event record and adds them as PseudoJets to a PseudoJet vector. The second method, `merge_and_prep`, does the same thing but generates a single vector for two events (intended for an event and a background).

4.2 Locating Hard Photons

For some explorations, it may be beneficial to restrict PYTHIA to generating events with a hard photon. For these cases, there are methods to tag photons, `tag_max_pt_photon`, and later locate the photon in the clustered jets, `locate_tagged_photon` and `contains_tagged_photon`. For situations where the jet finder does not keep clustering information or tagging information is overwritten, the user can use `get_max_pt_photon` to get a PseudoJet object that is a copy of the hard photon. This can then be used with the FastJet internal `delta_R` and `delta_phi_to` methods to identify which jet contains the clustered hard photon.

4.3 Possible Future Work

Utilities will be expanded as needed to include any new methods used frequently accross multiple scripts.

References

[1] P. Braun-Munzinger, K. Redlich and J. Stachel, In *Hwa, R.C. (ed.) et al.: Quark gluon plasma* 491-599 doi:10.1142/9789812795533_0008 [nucl-th/0304013].

[2] J. S. Moreland, J. E. Bernhard and S. A. Bass, Phys. Rev. C **92**, no. 1, 011901 (2015) doi:10.1103/PhysRevC.92.011901 [arXiv:1412.4708 [nucl-th]].

[3] M. Cacciari, G. P. Salam and G. Soyez, Eur. Phys. J. C **72** (2012) 1896 [arXiv:1111.6097 [hep-ph]].

[4] M. Cacciari and G. P. Salam, Phys. Lett. B **641** (2006) 57 [hep-ph/0512210].

[5] T. Sjstrand *et al.*, Comput. Phys. Commun. **191**, 159 (2015) doi:10.1016/j.cpc.2015.01.024 [arXiv:1410.3012 [hep-ph]].

[6] T. Sjostrand, S. Mrenna and P. Z. Skands, JHEP **0605**, 026 (2006) doi:10.1088/1126-6708/2006/05/026 [hep-ph/0603175].

[7] B. Andersson, S. Mohanty and F. Soderberg, hep-ph/0212122.

[8] C. Patrignani *et al.* [Particle Data Group], Chin. Phys. C **40**, no. 10, 100001 (2016). doi:10.1088/1674-1137/40/10/100001

[9] M. Dobbs and J. B. Hansen, Comput. Phys. Commun. **134**, 41 (2001). doi:10.1016/S0010-4655(00)00189-2