Targeted Monte Carlo: Filtered Simulations at STAR

Michael Betancourt[2], Victor Perevoztchikov[1], Jérôme Lauret[1], Jason Webb[1], Qinghua Xu[3], Jan Balewski[2]

[1] *Brookhaven National Laboratory, Upton, NY 11973, USA*
[2] *Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[3] *Shandong University, Jinan, Shandong 250100, China*

Abstract

We discuss the general utility of filtered simulations at STAR and the filtering framework currently implemented within the STAR software library. Particular attention is paid to the current state of *pp* simulations, possible biases in the existing methods, and the potential for filtering as a general solution.

1 Introduction

Simulation is a critical element of precision physics measurements. By studying the evolution of realistic physics events through full detector reconstruction, detector effects can be characterized and entire physics analyses studied for potential biases and other flaws.

In practice, these realistic physics events are Monte Carlo samples generated from programs such as Pythia [1]. Drawn from differential cross sections, which from a probabilistic perspective are really just probability distributions, these samples,

$$\mathbf{x}_i \sim p\left(\mathbf{x}\right) \propto \frac{d^n\sigma\left(\mathbf{x}_i\right)}{dy_1 \ldots dy_n},$$

are used to approximate various expectations

$$\bar{f} = \int d\mathbf{x}\, p\left(\mathbf{x}\right) f\left(\mathbf{x}\right) \approx \sum_i f\left(\mathbf{x}_i\right)$$

such as histogram populations.

The detector reconstruction, commonly performed by Geant, is also inherently a Monte Carlo approach where interactions between high energy particles and detector material are sampled from previously measured interaction cross sections.

Readers unfamiliar with Monte Carlo techniques are highly encouraged to first consult the excellent introductions in MacKay [2] and Bishop [3].

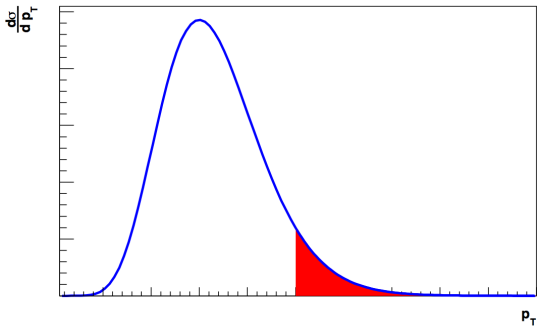


Figure 1: Often Monte Carlo generators can efficiently sample only from a bulk distribution (blue) when the desired physics lies within a much smaller region, here in the tail (red).

2 The Proverbial Needle in the Proverbial Haystack

When practicalities limit the phase space of a given physics process, however, efficient generation of the desired Monte Carlo samples may not be possible. Consider, for example, a Monte Carlo generator that can draw samples from only the full phase space of particular physics process where as the desired physics is restricted to a small corner of phase space (Figure 1).

Most of the computational resources in a naive simulation would be spent generating, and then reconstructing, samples from the bulk of the phase space. Very few samples relevant to the physics of interest would be produced.

By examining each sample before the computationally expensive detector reconstruction and rejecting those not satisfying the desired criteria, however, the generation of desired samples can be made significantly more efficient. This additional rejection sampling criteria, or filter, offers the potential for dramatically improved simulations.

2.1 Needles Defined By Nature

Often event generators do not offer sufficient customization of the final state. While Pythia allows the selection of the hard interaction and certain restrictions to the partonic final state, for example, the customization is not inclusive. In particular, the detailed structure of hadronic showers is a stochastic process whose phase space cannot be limited without violating the delicate correlations between the individual particles.

The introduction of a filter immediately after the event generation, however, allows for specific showers to be isolated and continued through reconstruction. Samples rich in high z pions or rare hadrons become a practical reality.

2.2 Needles Defined By Measurements

Event phase space can be restricted even further by restricting properties of the reconstructed final state, in particular the actual interactions of the final state with the surrounding detectors. Data written to tape is fundamentally limited by triggering conditions and other detector inefficiencies, and any physics simulation will contain extraneous events that would never have made it to disk.

Introducing a filter with a reconstruction criteria allows these extraneous events to be trimmed away and more resources spent on generating events relevant to the data. As the reconstruction becomes more and more inefficient, this becomes a more and more powerful technique.

Filtering by observed detector response is not just a means of improving simulation statistics. A close examination of the current simulation methodology at STAR reveals a dangerous vulnerability to sampling biases, and filtering is exactly the tool necessary for a general solution.

2.2.1 Partitioning Phase Space

High energy physics cross sections feature dramatic variations in scale, for example falling many orders of magnitude within the transverse momenta accessible at STAR. This immense range makes comprehensive sampling inherently difficult: the majority of computational resources are spent drawing samples at low p_T while the higher p_T tails, and the desired physics, are sparsely sampled.

In order to sample the tails more efficiently, the event phase space can be partitioned and each partition sampled separately. Such a partitioning in PYTHIA is furnished with the `CKIN` variables which restrict the kinematic phase space of the final state in the $2 \to 2$ hard interaction. STAR, for example, utilizes `CKIN(3)` and `CKIN(4)`, which provide lower and upper bounds on the transverse momentum of the final state partons and allow for comprehensive sampling of higher p_T physics.

Samples in the individual partitions must be weighted to ensure that each contributes appropriately to the Monte Carlo expectations. For the PYTHIA scheme above, the weights ensure that the integrated luminosities of each partition are equal. Consequently, the Monte Carlo expectations become

$$\bar{f} = \sum_j w_j \bar{f}_j,$$

where $\bar{f}_j$ is the Monte Carlo expectation from the jth partition,

$$\bar{f}_j = \sum_{i_j} f\left(\mathbf{x}_{i_j}\right),$$

and the weights are given by

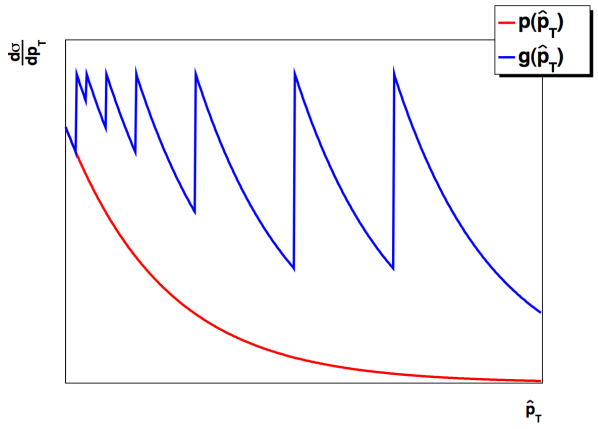


Figure 2: Partitioning phase space by partonic transverse momentum defines an auxiliary distribution $g\left(\hat{p}_T\right)$ for importance sampling. Each partition is thoroughly sampled and then weighted to conform to the true distribution $p\left(\hat{p}_T\right)$.

$$w_j = \frac{\int \mathrm{d}t\, \mathcal{L}_{\text{nominal}}}{\int \mathrm{d}t\, \mathcal{L}_j}$$
$$w_j = \frac{\int \mathrm{d}t\, \mathcal{L}_{\text{nominal}}}{N_j/\sigma_j}.$$

Formally, this procedure falls into a class of Monte Carlo techniques known as importance sampling [2, 3]. Instead of sampling directly from the desired distribution $p\left(\mathbf{x}\right)$, samples are drawn from an auxiliary distribution $g\left(\mathbf{x}\right)$ (Fig 2) and the Monte Carlo estimates are given by

$$\bar{f} = \sum_i \frac{p\left(\mathbf{x}_i\right)}{g\left(\mathbf{x}_i\right)} f\left(\mathbf{x}_i\right).$$

Importance sampling proves particularly useful when sampling from the auxiliary distribution $g\left(\mathbf{x}\right)$ is much easier than sampling from $p\left(\mathbf{x}\right)$ itself.

2.2.2 The Problem With Importance Sampling

Importance sampling, however, is not without its weaknesses. In regions where the auxiliary distribution does not cover the true distribution,

$$g\left(\mathbf{x}_i\right) < p\left(\mathbf{x}_i\right)$$

the local contributions to the Monte Carlo estimates are dominated by a few highly weighted events. Because the effective sample size is small, the resulting expectations are prone to bias and the large weights only amplify the error. More technical discussions are given in MacKay [2] and Bishop [3].

In the simple picture considered above (Fig 2) the weights appear to all be bounded by unity with the importance sampling estimates sound, but a closer inspection of the `CKIN` variables reveals a less appealing truth. The problem is that, while measurements are defined in the collider frame, the `CKIN` variables are not.

When PYTHIA samples the final state of the $2 \to 2$ hard interaction, it does so from distributions defined in the frame where the incident partons lie along the z axis with equal momenta. The incident partons drawn independently from each proton, however, will likely have different Bjorken x and hence unbalanced momenta. Moreover, initial state radiation introduces nontrivial transverse momentum into the sytem. In general, then, the initial partons will have to be boosted and rotated before PYTHIA can properly sample a final state.

The phase space restrictions defined by the `CKIN` variables are imposed also in this transformed, hard interaction frame. Consequently, the transverse momenta defining the partitions is not the collider p_T but rather the hard interaction $\hat{p}_T$.

Transforming back to the collider frame, the hard boundaries between the partitions in $\hat{p}_T$ become soft boundaries in p_T (Fig 3a). These tails stretch across p_T such that any given p_T bin contains contributions not only from the relative $\hat{p}_T$ bin but also from neighboring bins. Because of the steeply falling cross sections, the contributions of tails from low $\hat{p}_T$ samples are particularly significant (Fig 3b).

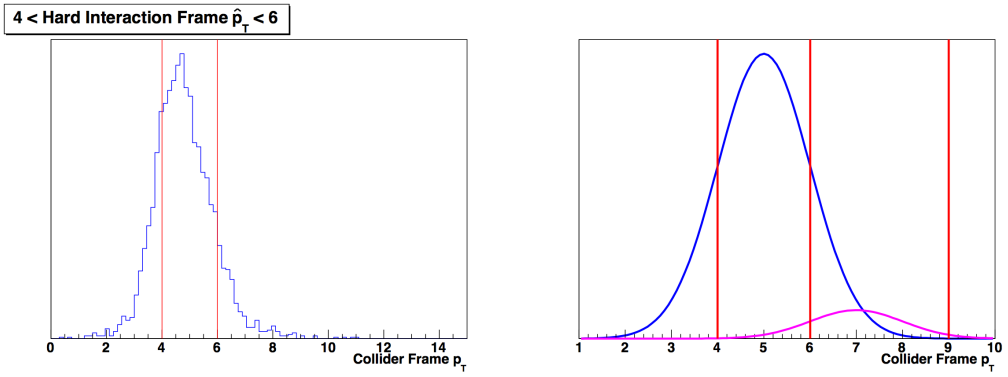
Note that additional effects, such as detector and reconstruction resolution, also induce tails in the p_T distributions. In practice, however, these are subdominant contributions.

Unfortunately, these low $\hat{p}_T$ samples have been inadequately sampled, if sampled at all, in STAR simulations. Because of the dramatic increase in cross section, generating sufficient samples and passing each through the entire simulation chain is computationally infeasible. Any allocation of finite resources results only in sparse, highly weighted, samples and the resultant Monte Carlo estimates enter into the regime where importance sampling breaks down.

A more efficient approach would avoid unnecessary reconstruction of events not ultimately contributing to the physics, instead focusing on generating copious PYTHIA events and reconstructing only those in the high p_T tails (Fig 4). In other words, one needs a filter.

3 Filtering

In order to avoid unnecessary reconstruction, a decision must be made to accept or reject each event based on a desired signal in the final reconstructed event. Where would such a



(a) (b)

Figure 3: (a) Tails induced from the boost and rotation between the hard scattering frame, where the `CKIN` cuts (red) are applied, and the collider frame. (b) Cartoon demonstrating the importance of tails when sampling from steeply falling cross sections. Significant contributions to the $6 < p_T < 9$ reconstructed bin (magenta) come not only from the bulk of the $6 < \hat{p}_T < 9$ distribution but also the tail of the $4 < \hat{p}_T < 6$ distribution (blue).

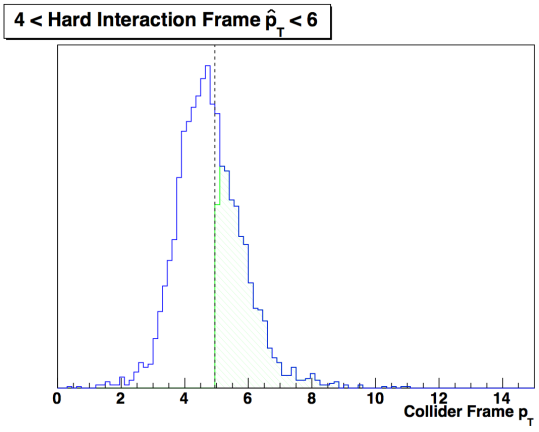


Figure 4: Sufficient sampling of a low $\hat{p}_T$ sample (blue) relies on rejecting the bulk which falls below the reconstruction threshold (here the dashed black line) in order to focus resources on the tail falling above the threshold (green).

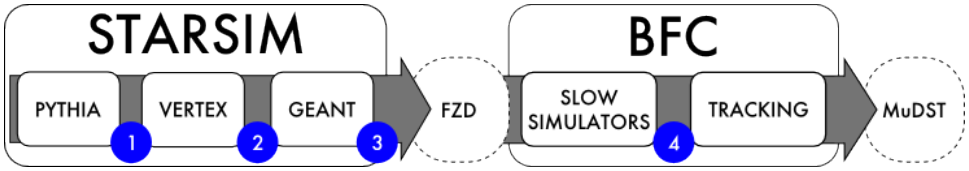


Figure 5: The positions within the STAR simulation framework where a filtering decision could be made. 1, 2, and 3 are implemented in the `StMCFilter` while 4 is implemented in the `StFilterMaker`.

decision be made?

The STAR simulation chain begins with STARSIM, where PYTHIA events are generated and tracked through GEANT one by one before being saved. The output of STARSIM is then input to the Big Full Chain, or BFC, where the GEANT hits are organized into detector energy depositions and event reconstruction such as TPC tracking is performed.

Interrupting STARSIM and rejecting events immediately after PYTHIA would avoid both computationally expensive GEANT tracking and event reconstruction, conserving the most resources. The PYTHIA event, however, is not equivalent to the final reconstructed event: many particle decays are suppressed, for example, and particles have not yet showered in the calorimeters. This discrepancy between the reconstructed event and the PYTHIA event fundamentally limits the power of filtering at this point in the simulation chain.

Not until GEANT tracking has been performed is the bulk of the reconstructed event available as input to a filtering decision. Although the cost of tracking would be inescapable, filtering here would still attenuate unnecessary event reconstruction in the BFC as well as reducing the ultimate storage footprint.

A framework incorporating both approaches, a filter after PYTHIA and another after GEANT, would maximize the possible rejection and optimize the sampling of the tails. The STAR filtering framework takes this inclusive approach (Fig 5), providing a flexible platform for comprehensive filtering needs with two software components.

`StMCFilter` This C++ class interfaces directly with STARSIM, allowing for the rejection of events before they are saved to disk. Methods allow for events to be rejected immediately after PYTHIA, immediately after the PYTHIA events are postioned into the detector with the generation of an event vertex, and immediately after GEANT. The ability to access the vertex is particularly useful, for example, when limiting the kinematics of the final state in the collider frame.

`StFilterMaker` This StMaker derived class is called from within the BFC, allowing events to be rejected before expensive TPC tracking begins. The information available is the same as the post-GEANT decision of the `StMCFilter`, but the placement in the BFC allows the `StFilterMaker` to take advantage of the detector slow simulators to organize the profusion of GEANT hits.

3.1 The Fine Print

The ability to arbitrarily filter events provides the opportunity to not only avoid the statistical pitfalls of importance sampling, but also produce otherwise superior simulation samples from fewer computational resources and less storage. Proper use of this extensive filtering framework, however, must also consider new, often subtle, issues.

3.1.1 Bias

Given the definition of a reconstruction signal, a filter can be designed to reject events not ultimately satisfying the reconstruction criteria. At the same time, however, the filter should not reject events that would fulfill the criteria. Enough of these false biases can dominate the importance sampling uncertainties and spoil the validity of the sample.

What bias is considered acceptable? Consider the weights of a given sample,

$$
\begin{aligned}
w_j &= \frac{\int dt\, \mathcal{L}_{\text{nom}}}{\int dt\, \mathcal{L}_j} \\
w_j &= \frac{\int dt\, \mathcal{L}_{\text{nom}}}{N_j/\sigma_j} \\
w_j &= \left(\int dt\, \mathcal{L}_{\text{nom}} \right) \frac{\sigma_j}{N_j}.
\end{aligned}
$$

The uncertainty in the weights is

$$
\begin{aligned}
\delta w_i &\approx \left(\int dt\, \mathcal{L}_{\text{nom}} \right) \delta \frac{\sigma_j}{N_j} \\
\delta w_i &\approx \left(\int dt\, \mathcal{L}_{\text{nom}} \right) \sqrt{\left(\frac{1}{N_j} \delta \sigma_j \right)^2 + \left(\frac{\sigma_j}{N_j^2} \delta N_j \right)^2} \\
\delta w_i &\approx \left(\int dt\, \mathcal{L}_{\text{nom}} \right) \frac{\sigma_j}{N_j} \sqrt{\left(\frac{\delta \sigma_j}{\sigma_j} \right)^2 + \left(\frac{\delta N_j}{N_j} \right)^2} \\
\delta w_i &\approx \frac{\int dt\, \mathcal{L}_{\text{nom}}}{\int dt\, \mathcal{L}_i} \sqrt{\left(\frac{\delta \sigma_j}{\sigma_j} \right)^2 + \left(\frac{\delta N_j}{N_j} \right)^2} \\
\delta w_i &\approx w_i \sqrt{\left(\frac{\delta \sigma_j}{\sigma_j} \right)^2 + \left(\frac{\delta N_j}{N_j} \right)^2}
\end{aligned}
$$

or

$$\frac{\delta w_i}{w_i} \approx \sqrt{\left(\frac{\delta \sigma_j}{\sigma_j}\right)^2 + \left(\frac{\delta N_j}{N_j}\right)^2}.$$

The cross section σ_j is itself a Monte Carlo estimate from PYTHIA, with a counting uncertainty

$$\frac{\delta \sigma_j}{\sigma_j} \approx \frac{1}{\sqrt{N_{gen}}},$$

where N_{gen} is the number of PYTHIA events generated in the calculation of the cross section [1]. Practical considerations[1] limit N_{gen} to $O\left(10^6\right)$, giving the cross section uncertainty

$$\frac{\delta \sigma_j}{\sigma_j} \approx 10^{-3}$$

Any false rejections contribute directly to δN_j, and if they are to be considered negligible they must satisfy

$$\frac{\delta N}{N} \ll \frac{\delta \sigma}{\sigma}$$

$$\frac{\delta N}{N} \ll 10^{-3}$$

$$\delta N \ll 10^{-3} N$$

Vetting the bias of a filter is then accomplished by generating thousands of events and ensuring that significantly less than 10^{-3}, 10^{-4} is a good target, are falsely rejected.

Note that any validation holds only for the exact settings used in the test. If settings relevant to detector reconstruction, such as gains, are changed, as is done in systematic studies, then the filtered sample is not guaranteed to remain valid as the changes can decrease the underlying filter acceptance and increased false rejections along with it. Only when bias tests are done with settings that minimize the filter acceptance will tuning be acceptable for all relevant settings.

3.1.2 Truth vs. Reconstructed Truth

Care must be taken when implementing filtering simulations utilizing any criteria based on underlying truth. Because the same requirement cannot be made of the raw data, these simulations cannot be compared to the data until significant analysis has been performed. Even then, limitations in the analysis may prevent the comparison from being exact.

[1] Generating 10^6 PYTHIA events takes a few hours. Improving the cross section uncertainty by another order of magnitude would require 10^8 events and over two weeks.

3.1.3 Signal vs. Reconstructed Signal

When filtering by detector response, the acceptance criteria are defined entirely in terms of a reconstructed signal and, consequently, physics events that will not reconstruct are rejected the same as background. These rejected events, however, are critical for accurate calculation of reconstruction efficiencies. If the filtered sample will be used for such calculations then the true physics, in other words the PYTHIA event record, must be saved for any rejected events (given the sparsity of the PYTHIA record, the requisite disk space is negligible).

Design of a filter must carefully consider the ultimate physics, ensuring that all physics events will either pass the filter (particularly amenable to post-PYTHIA filtering) or have their event record saved (more appropriate for post-GEANT filters, most easily implemented in a `StFilterMaker`).

3.1.4 Timing

The cost of conventional simulation is dominated by GEANT tracking and the BFC – the expense of generating of a PYTHIA is negligible in comparison. Knowing the time necessary to run a single event through the simulation chain will then entirely determine the structure of a practical simulation request. When filtering is introduced, however, the PYTHIA can become significant.

Well designed filters acting immediately after PYTHIA can rejects hundreds of thousands of PYTHIA events before an acceptable event is passed to the rest of the simulation chain. The cost of generating a single effective PYTHIA event dramatically increases, the resources needed for the rest of the chain become negligible, and the entire structure of the request must be reconsidered. Powerful post-GEANT filters, where the cost of tracking each rejected event quickly integrates, are even worse.

When filtering is involved, naive simulation requests can lead to very poor allocation of resources. When the filter acceptance rate is low, generation of a single event can easily exceed tens of minutes and a conventional job requiring hundreds of events would run for days. Consequently, careful timing studies must be done for any filtered request to ensure feasibility.

3.1.5 Storing Log Files

Lastly, the log files from each simulation jobs must be considered as critical as the simulation itself. Only here can the luminosity of each sample, and hence the necessary weighting, be calculated.

3.2 So You Want To Make A Filter...

In order to illustrate the finer details important when implementing a filter, consider the following example. The goal will be to implement a simple filter for a prompt photon simulations.

Step 1 - Define Reconstruction Signal

Expecting prompt photons to leave spatially isolated calorimeter depositions, the reconstruction signal will be defined as any event with a 3×3 tower cluster in the BEMC with total $E_T > 6$GeV. In addition, the event must satisfy the l2BemcGamma trigger emulator.

Step 2 - Design Filters

Filtering will consist of two decisions, one after a vertex has been generated (immediately before GEANT) and the second in the BFC. The first mocks up the tower cluster by summing true particle energy in an η-ϕ cone just large enough to contain the 9 towers. Events passing the filter require a cone with true $E_T > E_1$. Provided that $E_1 < 6\,\text{GeV}$, which is already necessary to avoid bias, all true prompt photon events will immediately pass this filter.

The latter performs a proper 3×3 cluster on the GEANT energy depositions and requires at least one cluster with $E_T > E_2$. When generating true prompt photon events, the filter also saves the PYTHIA event record of each incident event.

Step 3 - Tune Filters

Both decisions are tuned by setting E_1 and E_2 as close to possible to $6\,\text{GeV}$ without inducing significant bias (Table 3.2).

When emulating a trigger decision, ADCs must be mocked up by converting from the deposited energy (corrected for sampling fraction) with the gains and, because of nontrivial uncertainty in the determination of the these gains, the actual values will be varied for final systematic studies. In the BEMC convention a small gain yields a larger ADC, a higher probability of passing the trigger, and larger filter acceptance. Consequently, tunes are done using the smallest gain employed in the final systematic study.

Timing Studies

Given the final tune, additional tests are run to determine the resources necessary for each $\hat{p}_T$ bin (Table 3.2). These studies finally enable the design of a feasible simulation request.

4 Conclusion

The STAR filtering framework, `StMCFilter` and `StFilterMaker`, enables the filtering of events at each each step of the simulation chain based on arbitrary criteria. When carefully

applied, this framework allows for the efficient generation of large simulation samples safe from the statistical bias inherent in conventional sampling.

5 Filtering History

From its inception, STARSIM has featured a single particle filter allowing for simple filtering requirements. The first extension, implemented as modifications of the PYTHIA event generation code itself, was introduced by Qinghua Xu and enabled more complex filtering decisions, although early uses were limited to filtering events by their underlying truth. Jan Balewski first studied the potential gains from this framework.

Inspired by artifacts in existing simulations, Michael Betancourt discovered the subtleties of the `CKIN` cuts, formalized their use from an importance sampling perspective, and introduced filtering criteria based on reconstructed characteristics in order to avoid the statistical biases. Realizing the need for further flexibility, he also designed the BFC filtering framework.

Victor Perevoztchikov generalized the original PYTHIA filter with the `StMCFilter`, creating a more systematic and flexible framework applicable to any event generator implemented in STARSIM.

The STAR software team, particularly Jérôme Lauret and Jason Webb, offered invaluable support throughout the evolution of the filtering framework.

References

[1] Torbjörn Sjörstrand, Stephen Mrenna, and Peter Skands *Pythia 6.4 Physics and Manual*, `http://home.thep.lu.se/~torbjorn/Pythia.html`, March, 2006

[2] MacKay, D. J. C. (2003) *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, New York

[3] Bishop, C.M. (2007) *Pattern Classification and Machine Learning.* Springer, New York

Table 1: Hypothetical filter tuning results – the numbers are for illustrative purposes only and should not be used to judge real filter implementations. The tabular presentation is particularly useful when proposing that the filter be used in an official simulation request.

Tuning

CKIN(3) (GeV)	CKIN(4) (GeV)	StMCFilter Acceptance	StMCFilter Bias	StFilterMaker Acceptance	StFilterMaker Bias	Total Acceptance
2	3.5	0.0010 %	3/50000	13.4%	1/50000	0.00013 %
3.5	4.5	0.016%	1/50000	16.5 %	0/50000	0.0026%
...	...	...	...	...	...	...
25	40	78.3%	0/50000	57.21%	0/50000	44.8%
40	∞	81.7%	0/50000	85.21%	0/50000	69.7%

Timing

CKIN(3) (GeV)	CKIN(4) (GeV)	Time/Event (s)	Time/Event (hours)	Size/Event (GB)
2	3.5	446	0.12	0.001
3.5	4.5	289	0.08	0.001
...	...	...	...	...
25	40	144	0.04	0.001
40	∞	58	0.02	0.001

Final Request

CKIN(3) (GeV)	CKIN(4) (GeV)	Cross Section (pb)	Raw Events For 1 pb^{-1}	Filtered Events	Number of Files	Events/File	Time/File (hours)	Size/File (GB)
2	3.5	8,992,000,000	8992000000	11860	360	33	4	0.033
3.5	4.5	614,100,000	614,100,000	15917	318	50	4	0.050
...	...	...	...	...	...	...	...	...
25	40	10,610	10,610	4754	48	100	4	0.100
40	∞	126.80	127	89	1	200	4	0.200