An automated tool to facilitate consistent test-driven development of trigger selections for LHCb's Run 3

Ross Hunter[1,*], *Olli* Lupton[1,**], *Rosen* Matev[2,***], *Sascha* Stahl[2,****], and *Mika* Vesterinen[1,†]

[1]Department of Physics, University of Warwick, Coventry, United Kingdom

[2]European Organization for Nuclear Research (CERN), Geneva, Switzerland

Abstract. Upon its restart in 2022, the LHCb experiment at the LHC will run at higher instantaneous luminosity and utilize an unprecedented full-software trigger, promising greater physics reach and efficiency. On the flip side, conforming to offline data storage constraints becomes far more challenging. Both of these considerations necessitate a set of highly optimised trigger selections. We therefore present `HltEfficiencyChecker`: an automated extension to the LHCb trigger application, facilitating trigger development before data-taking driven by trigger rates and efficiencies. Since the default in 2022 will be to persist only the event's signal candidate to disk, discarding the rest of the event, we also compute efficiencies where the decision was due to the true MC signal, evaluated by matching it to the trigger candidate hit-by-hit. This matching procedure – which we validate here – demonstrates that the distinction between a "trigger" and a "trigger-on-signal" is crucial in characterising the performance of a trigger selection.

1 Introduction

The upgraded LHCb detector in Run 3 (2022-2025) [1] will take data at five times the previous instantaneous luminosity, yielding on average five to six *pp* interactions per bunch crossing [2]. LHCb will also become the first hadron collider experiment with a full-software trigger. The previous level-0 (L0) hardware-based trigger will be removed, and instead the first level of the software-based High Level Trigger (HLT) will process the full 30 MHz *pp* collision rate. Across two levels, HLT1 and HLT2, the HLT performs a complete reconstruction of each collision (or *event*), and persists only those events we consider to be interesting, thereby reducing the amount of data we store by three orders of magnitude. We denote this reconstruction as the *online reconstruction* here, because it occurs before any data are sent to permanent storage. Filtering is achieved by requiring that each event satisfies at least one

[*]e-mail: ross.hunter@warwick.ac.uk
[**]e-mail: oliver.lupton@cern.ch
[***]e-mail: rosen.matev@cern.ch
[****]e-mail: sascha.stahl@cern.ch
[†]e-mail: mika.vesterinen@warwick.ac.uk

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

of many selection requirements at both HLT1 and HLT2. Each of these selections is known as a trigger *line*, which typically defines how a trigger candidate is reconstructed from tracks up to a software object that represents either a full or partial signal decay, and also a variety of complex thresholds that the candidate must satisfy. Removal of the L0 hardware stage is projected to increase the trigger efficiency on typical hadronic decays of beauty hadrons by a factor of two [1, 3]. With the increase in interactions per event, this equates to a factor of ten increase in the signal yield per unit time with respect to Run 2 (2015-2018). Further gains are also expected in physics reach due to the collaboration's 2020 decision to build the first stage of the trigger, HLT1, solely on GPUs [4, 5].

All signal events retained by the trigger must be stored offline and made accessible for physics analysis. These gains therefore come with costs in the form of disk storage; the output bandwidth that can be written out of HLT2 will be restricted to 10 GB/s [2]. Whilst there has been considerable effort in reducing and compressing the size of the events written out [6, 7], this is not sufficient alone to hit the bandwidth target. We must also ensure that our trigger lines are highly-optimised: striking an appropriate balance between high signal efficiency and offline data storage and processing capacity.

During Run 2, trigger efficiencies were typically evaluated using the data-driven "TISTOS" method [8–10]. In a busy hadron collider environment there is the risk that a positive trigger decision may not be related to the presence of the signal particle. Thus, in the TISTOS approach, the efficiency of the trigger firing on signal was calculated by matching the trigger candidate – the reconstructed physics object made from detector hits representing e.g. the decay of a B meson – to a candidate constructed by a more complex and higher-purity *offline* (meaning that it was ran on the data saved on disk) reconstruction and selection, which was assumed to be the true signal in that event (if present). Therefore, a valid positive trigger decision was one where the trigger built and accepted a candidate that matched the "true" offline candidate. This is known as the *trigger-on-signal*, or `TOS`[1] efficiency, and provides an accurate measurement of the true signal efficiency. However, calculating efficiencies with respect to an offline reconstruction and selection gives scope for complication, inconsistency and bias. An offline reconstruction and selection also necessitates that the full raw event be saved. Disk storage constraints make this impractical for the majority of the physics processes in Run 3. Instead, the majority of trigger lines will persist to the *Turbo stream*, which discards the raw event and saves only the reconstructed signal candidate in the event, in a format ready for physics analysis [2, 6]. This drastically reduces the size of each saved event, but makes an offline reconstruction and selection from the raw event impossible. The only remaining relevant set of trigger-unbiased events we can take to measure trigger efficiencies is simulated signal events.

Nevertheless, the case for a "TOS-like" efficiency is still clear. In Turbo-persisted events, the candidate that the trigger reconstructed and selected is treated as signal, regardless of its true origin. Triggering on background is not a new problem, but information on how likely such a "false positive" will be is critical for a line author. For lines that will persist part/all of the raw event, the only candidates which have a trigger efficiency that can be well understood, in a typical physics analysis, are those that can be matched to the signal.

At LHCb, line development has typically been done by independent line authors with no centralized or automated tools in place. This presents a variety of problems: it is not an optimal usage of person-power, and leads to variations in approach, definitions of efficiencies

[1] Analogously, the numerator of the "trigger-independent-of-signal" or "TIS" efficiency is the number of positive trigger candidates that do not match the offline candidate.

and the degree of testing and validation. Our aim was to build a tool that enables the crucial work of line tuning across the full spectrum of LHCb's trigger lines, and further does so in a well-defined, user-friendly and automated way. Whilst the tool should be flexible for different signals and trigger lines, the definitions of efficiencies used should be transparent and consistent for most use cases. We also aimed to build a lightweight algorithm that could match trigger candidates to the true signal in a simulated event, such that we could evaluate a representative and robust "trigger-on-signal" efficiency. In the following we shall denoted this as the `TOS` efficiency for brevity, despite its differences to the traditional meaning of `TOS` described above.

In the next section we give a brief outline of the implementation of `HltEfficiencyChecker`, and define the relevant quantities of interest in line tuning. We also go into more detail on the `TOS` matching procedure. In Section 3 we perform a validation study of this procedure. This is followed by Section 4, which showcases the tool. Limitations and possibilities for future work are included in Section 5. We give some conclusions on the study in Section 6.

2 Implementation

The `HltEfficiencyChecker` [11] package is built on top of the LHCb trigger application `Moore`, as well as the LHCb `Analysis` project, which houses tools used for offline analysis. The user provides a simple script which configures the trigger, the simulated sample to run over and the results that the user wishes to see. `HltEfficiencyChecker` then runs the trigger on the simulated sample as a subprocess. Offline analysis tools are then called to persist trigger information, as well as the kinematics of the true Monte-Carlo (MC) signal candidate in the event, to the `ROOT` ntuple format. A second subprocess is then called, which is one of a set of analysis scripts which calculate and plot the observables of interest. The configuration can be provided in two ways depending on the experience level of the user with regards to the trigger software.

We define a trigger decision (`DEC`) efficiency as

$$\varepsilon = \frac{N_{\text{Triggered}}}{N_{\text{Pre-selected}}}, \tag{1}$$

where $N_{\text{Pre-selected}}$ is the number of MC signal candidates passing a pre-selection cut, or *denominator*. In the interest of consistency, `HltEfficiencyChecker` has a hard-coded dictionary of denominators indexed by key via the configuration script[2]. The default denominator, chosen for simplicity and to give the most unbiased results, requires that all the final-state children of the signal decay are charged and produced long tracks[3] in the fiducial acceptance of the detector. As is detailed later, we will use a representative B decay as a test case, so in accordance with previous LHCb trigger efficiencies [3], we also require that the parent B meson has a lifetime greater than 0.2 ps. Such a cut is typical, as there is too much background at low lifetimes, making these B candidates very hard to analyse. These requirements are made on the truth-level information of the true MC signal candidate, rather than the candidate built by the trigger out of reconstructed tracks, hereafter the *trigger candidate*. Owing to the excellent reconstruction, we expect the difference between the two to be negligible when reconstructing the true MC signal. The `TOS` efficiency is

[2] A new denominator can of course be hard-coded in, but this is deliberately left inflexible.

[3] A "long" track traverses the LHCb tracking system: it has hits in the VELO, before the magnet, and the Scintillating Fibre Tracker downstream of the magnet [12].

$$\varepsilon = \frac{N_{\text{Triggered \& Matched}}}{N_{\text{Pre-selected}}}\,, \tag{2}$$

where events in the numerator require that at least one trigger candidate (many trigger candidates per event are possible) could be "matched" to the true MC signal candidate in the event. The matching algorithm collects the detector hits of the trigger candidate's tracks and the true signal's tracks, and requires that 70% or more of the trigger candidate hits are present in the true signal for the trigger candidate to be `TOS` with respect to the true MC signal for that trigger line. For simplicity, this algorithm does not account for different signal topologies: regardless of the number of tracks and how they combine, all of the hits are collected into a single container. Trigger lines often trigger on a subset of the decay products of a complex decay. Since this *matching fraction* is a fraction of trigger candidate hits, rather than true signal hits, a trigger candidate reconstructing of a subset of a decay, e.g. the J/ψ in $B_s \to J/\psi\phi$ will be `TOS` with respect to the intermediate particle (J/ψ) and the parent particle (B_s). This is desirable since an analyst may require that the trigger fired on at least one part of the decay, particularly in HLT1, where simpler, looser, inclusive one and two-track selections are made. Keeping with the same example, the J/ψ itself is not the final state that we trigger on, but it decays to two muons, and the J/ψ trigger candidate should be made up of two final-state muon tracks. The same is true for the ϕ, which decays here to two kaons. There is a possibility in our algorithm that a kaon and a muon track are combined to make the trigger candidate of the two-track line. This may or may not be considered as a useful source of efficiency for an offline analysis. Here, we prefer not to consider events triggered on a muon-kaon pair as useful. We can filter them out by additionally defining a B_s `TOS OR` efficiency:

$$\varepsilon = \frac{N_{\text{Triggered \& (Match on } J/\psi \text{ or on } \phi)}}{N_{\text{Pre-selected}}}\,, \tag{3}$$

which is the same as Eq. (2), except that we require that at least one sub-decay in the true signal decay could be properly matched to a trigger candidate. In $B_s \to J/\psi\phi$, this corresponds to a two-track trigger candidate being `TOS` on either the J/ψ or the ϕ in the event, or a one-track candidate being matching to one of the kaons or muons. This `TOS OR` efficiency can be generalised to other multi-body decays through intermediate states.

The final quantity of interest in this study is the rate of a trigger line:

$$\varepsilon = \frac{N_{\text{Triggered}}}{N_{\text{Events}}} \times \text{Input Rate}\,. \tag{4}$$

which is evaluated on minimum bias simulation with no pre-selection. The input rate of events depends on the trigger stage in question, e.g. it is 30 MHz for HLT1 in Run 3.

3 Validation of matching procedure

In order to validate the matching procedure, we define a simple test setup of the high-level trigger. We use a simulated sample of the representative heavy flavour decay $B_s \to J/\psi\phi$ (with $J/\psi \to \mu^+\mu^-$ and $\phi \to K^+K^-$), and run two trigger lines: one searching for a single track and the other for a two-track vertex, similar to that present in LHCb's HLT1. This enables us to test the matching algorithm when matching to different topologies: the one-track line should match to one of the two muons or kaons, whereas the two-track line should

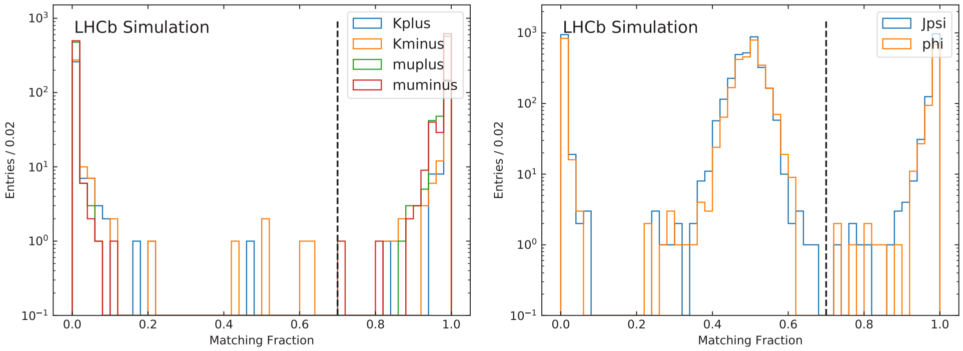


Figure 1: Histograms of the trigger candidate matching fractions to each of the true signal candidates, for the one-track line (left) and the two-track line (right) for triggered $B_s \to J/\psi\phi$ events. The dashed line is at a matching fraction of 0.7.

trigger on the J/ψ and ϕ. The selection they apply is, in both cases, based on a trained multivariate analysis (MVA) classifier to identify the products of a heavy-flavour decay, but the exact specifics of the lines are not important here.

In the matching algorithm, the only free parameter is the minimum matching fraction that we declare the trigger candidate is TOS. We anticipate that the TOS efficiency will vary with minimum matching fraction. The absolute value of the TOS efficiency is however not hugely important, so long as it is not biased too high by false positives (e.g. spurious trigger candidates that overlap with the true signal but do not come from it) or biased too low by throwing away good matches. Historically, the minimum matching fraction used in the TISTOS method was 0.7 (although this varied for hits in different subdetectors). This is also the matching fraction commonly used to match reconstructed tracks to true MC tracks when calculating LHCb's track reconstruction efficiency, where it has been validated to have a high efficiency of matching well-reconstructed tracks [13]. We therefore choose to start with 0.7.

To investigate the suitability of this minimum matching fraction, in Fig. 1 we have plotted a histogram of the matching fractions of the trigger candidates made by the one- and two-track lines in response to $B_s \to J/\psi\phi$. In the case of the one-track line, the distribution of candidates is skewed towards a matching fraction of either 0 or 1, with almost no trigger candidates falling inbetween. The tail of candidates at low matching fractions demonstrates that spurious tracks/tracks from the underlying event can overlap to a small degree with the true signal track. We infer that the tail at high matching fractions is due to extra hits not present in the true signal; for instance due to random hits that happen to align with the track direction and are picked up by the pattern recognition algorithms. We consider such tracks to still be good matches to the true signal track, and therefore conclude that the minimum matching fraction should not be set too close to 1. The lack of overlap between the populations at 0 and 1 suggests there is almost no likelihood that a spurious track can imitate a true signal track well enough to be declared TOS with a matching fraction of 0.7 or more.

In the two-track case, we observe the same populations at 0 and 1, but also a large population peaking around 0.5. This intermediate population suggests that there are candidates formed of one well-matched track and a second track that is picked up from the underlying event or otherwise spurious. The broadness of the peak reflects the tail of overlap present at low and

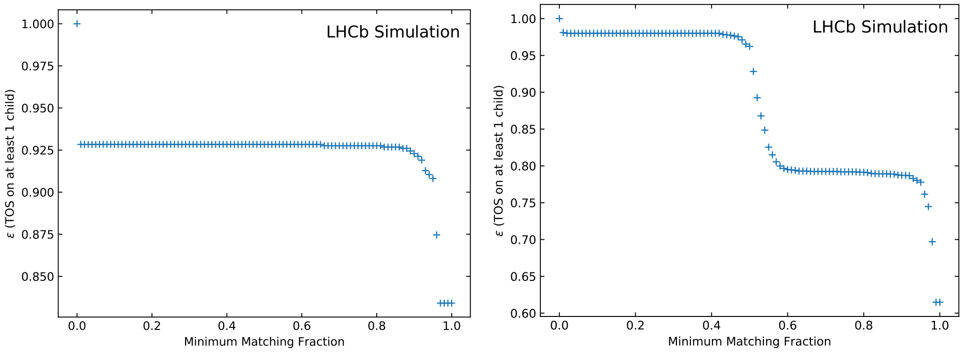


Figure 2: (left) The efficiency that at least one one-track trigger candidate matched one of the kaons or muons in triggered events and (right) the efficiency that at least one two-track trigger candidate matched the J/ψ or ϕ in $B_s \to J/\psi\phi$, as a function of the minimum matching fraction requirement.

high matching fractions in the one-track case, and that the two tracks need not have the same absolute number of hits.

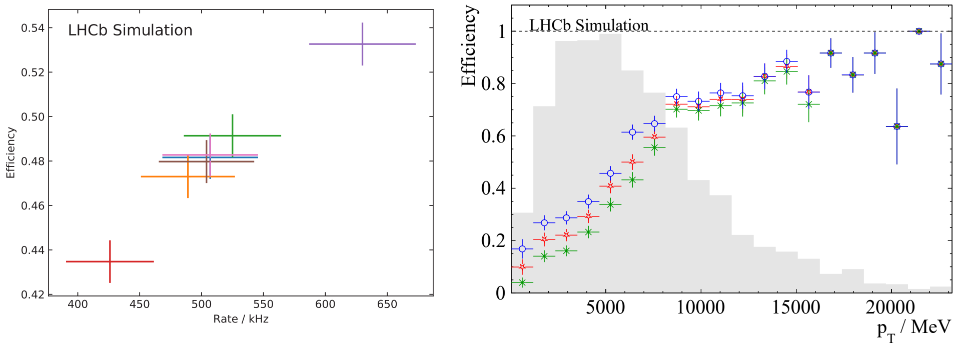
In both the one- and two-track cases, a minimum matching fraction of 0.7 is sufficient to separate the well-matched population from those candidates that we consider to not be adequately matched at lower matching fractions. This good separation suggests that, near 0.7, the `TOS` efficiency will not be a strong function of the matching fraction. This can indeed be seen in Fig. 2. The efficiency is flat as a function of matching fraction in the region near to 0.7. We also see here a large shift in TOS efficiency near 0.5 in the two-track case, indicative of the loss of the candidates where only one of the two tracks was well-matched. We also see a drop as we move towards a matching fraction of 1 which is the result of – what we assert to be – well-matched candidates that should not be thrown away.

From these plots we can conclude that a minimum matching fraction of 0.7 selects the trigger candidates that we believe to be correctly matched with both high efficiency and purity. In the region around 0.7, the `TOS` efficiency is stable, which reassures us that the absolute choice does not have a large impact.

4 Line tuning with HltEfficiencyChecker

In the previous two sections, we have outlined and validated the implementation of `HltEfficiencyChecker` as a tool for line development. The aim was to provide line developers with the key information they need to be able to tune their line. In this section we showcase how the tool does this, by showing some of the plots that it can be configured to make. This is not an exhaustive list, and the plots shown here should not be taken as indicative of the trigger performance that LHCb predicts to achieve in Run 3.

We keep the example of $B_s \to J/\psi\phi$ and the simple one- and two-track lines from the previous section. The LHCb HLT2 trigger application `Moore` allows us to configure multiple copies of a line with slightly differing thresholds, so in Fig. 3a we show the rate on minimum bias and the trigger `DEC` efficiency for six versions of the same line, making slightly different choices on the three MVA parameters.



(a)

(b)

Figure 3: (a) Rate on minimum bias vs. DEC efficiency on $B_s \rightarrow J/\psi\phi$ for the one-track line, plus six toy variations of the same line with different MVA parameters. (b) The DEC efficiencies on $B_s \rightarrow J/\psi\phi$ of the one-track line (blue), plus two toy variations of the line (red, green), plotted as function of B_s transverse-momentum $p_T(B_s)$. The scaled $p_T(B_s)$ distribution is under-laid in grey.

How the signal efficiency varies with the kinematics of the signal may be of interest to the line author as well. In Fig. 3b we show the DEC efficiency of the one-track line, plus two of the slightly-varied lines from Fig. 3a, plotted against the true transverse-momentum of the B_s in $B_s \rightarrow J/\psi\phi$ decays.

The DEC efficiency, TOS efficiency with respect to each child, and the TOS OR efficiency can all be plotted in a variety of combinations against a variety of true kinematic quantities. Since this flexibility is available, the line developer can exploit it to study the difference between the difference types of efficiency, which give them more insight on the performance of their line. For example, in Fig. 4 we have plotted both the DEC and TOS OR efficiencies of the one- and two-track lines in response to $B_s \rightarrow J/\psi\phi$. A non-negligible difference is seen between the two efficiencies in both cases, which is perhaps most clear when comparing the integrated efficiencies listed in the legend.

5 Limitations and future work

In section 3 we restricted the validation of the TOS matching procedure to just a one- and two-track line for the sake of simplicity. However, there will be lines in LHCb's HLT2 that will build more complex three- and four-track combinations. We cannot draw the same conclusions of our matching procedure in these cases. For example, it is easy to see that, with the same procedure, a four-track candidate would be flagged as TOS even if only three out of the four tracks in the trigger candidate were from the true signal, as this would achieve a matching fraction of roughly 0.75. We previously argued that such a case, where one of the tracks is not from the signal, should not be TOS. Further work is therefore needed to generalise the matching algorithm to make it valid for these more complex cases. This could for instance be done by instead requiring that each individual track in the trigger candidate matches to a signal track with greater than 70% of the hits.

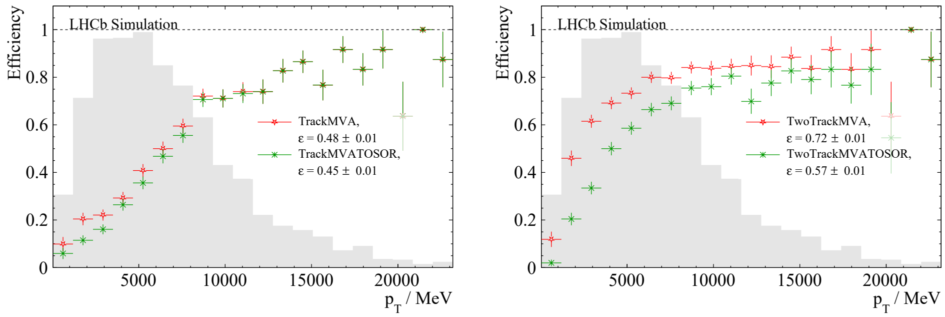


Figure 4: The `DEC` (red) and `TOS OR` (green) efficiencies of the one-track line (left) and two-track line (right) in response to $B_s \to J/\psi\phi$, as function of B_s transverse-momentum $p_T(B_s)$. The scaled $p_T(B_s)$ distribution is overlaid in grey, and the efficiencies averaged over $p_T(B_s)$ are listed in the legend.

As mentioned in section 3, the Run 2 offline implementation of `TOS` matching was slightly more complex, in that it required slightly differing fractions of overlap for hits in the different LHCb subdetectors. Such an approach was not studied here, and could give slightly different efficiencies, although we believe that this will be very much a second order effect.

Finally, this work comes as part of the large effort by the LHCb RTA project to validate and optimise the LHCb HLT. For example, in recent years a nightly CI testing and performance regression suite has been put in place [14]. `HltEfficiencyChecker` could be used in these contexts to perform automated regression tests of the high-level physics performance of the trigger, for example in response to merge requests. This possibility requires significant extra work in plumbing things together, but would be a large step forward in building a more reliable and more performant trigger in time for Run 3.

6 Conclusion

When the LHCb experiment restarts data-taking in 2022, there will be substantial challenges in meeting the requirements placed on the trigger by the available offline data storage. We can only overcome these challenges with a well-tuned set of trigger lines. In this paper we have presented the `HltEfficiencyChecker` package, which extends the LHCb trigger application and enables development driven by rates and efficiencies. This allows tuning to begin before data-taking, and ensures that these key observables are calculated in a consistent manner. Whilst we have not given a tutorial on the tool here, we believe that the tool is also flexible and easy to use. The tool has already been used by several line authors even at this early stage of trigger development, and positive feedback has been received. It was also used in the LHCb HLT1 technology decision study [5].

As well as providing the simple trigger "decision" efficiency to line authors, `HltEfficiencyChecker` also gives the efficiency that the trigger fired on a true signal particle in the event, known as the "trigger-on-signal" or `TOS` efficiency. The matching algorithm to determine this has been validated on a simple test case of one- and two-track lines; we find that it gives results that are stable with respect to the choice of the matching requirement and separates out matches to the true signal with high efficiency and purity, giving a good estimation of the true signal efficiency. This test case does not extend to all the possible trigger lines that will be present in the Run 3 HLT, so further work should be done generalise the matching approach.

Finally, we then showcased how `HltEfficiencyChecker` can be used in line development. In the test line development scenario, we find non-negligible differences between the "decision" efficiency and the `TOS` efficiency. We expect this discrepancy to be common, and it demonstrates that information on both of these quantities is crucial for the user to properly characterise the performance of their line. This tool is now available to the LHCb community, and will form a critical part of ensuring that the LHCb HLT is fit for purpose in time for the start of Run 3.

Acknowledgements

The authors would like to thank the LHCb computing and simulation teams for their support and for producing the simulated LHCb samples used in the paper. We would also like to thank the full RTA team for their valuable inputs to this project.

References

[1] LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report* (2012), `LHCb-TDR-12`

[2] LHCb Collaboration, *Computing Model of the Upgrade LHCb experiment* (2018), `LHCB-TDR-018`

[3] LHCb collaboration, *LHCb Trigger and Online Upgrade Technical Design Report* (2014), `LHCB-TDR-016`

[4] LHCb Collaboration, *LHCb Upgrade GPU High Level Trigger Technical Design Report* (2020), `LHCB-TDR-021`

[5] R. Aaij et al., Submitted to Comput. Softw. Big Sci. (2021)

[6] R. Aaij, S. Amato, L. Anderlini, S. Benson, M. Cattaneo, M. Clemencic, B. Couturier, M. Frank, V. Gligorov, T. Head et al., Comput. Phys. Comm. **208**, 35–42 (2016)

[7] R. Aaij et al., JINST **14**, P04006 (2019)

[8] R. Aaij, et al., JINST **8**, P04022 (2013)

[9] R. Aaij, et al., JINST **14**, P04013 (2019)

[10] S. Tolk, J. Albrecht, F. Dettori, A. Pellegrino (2013), `LHCb-INT-2013-038`

[11] *Hltefficiencychecker [software]* (2021), [Online; accessed 02-06-2021], `https://gitlab.cern.ch/lhcb/MooreAnalysis/-/tree/master/HltEfficiencyChecker`

[12] P. Li, E. Rodrigues, S. Stahl (2020), `LHCb-INT-2020-005`

[13] R. Aaij, et al., Journal of Instrumentation **10**, P02007–P02007 (2015)

[14] R. Currie, C. Fitzpatrick (LHCb Collaboration), EPJ Web Conf. **214**, 05042. 4 p (2019)