Integration of decay weights into the EvtGen framework

Anna Casali
a.casali.1@research.gla.ac.uk

Monte Carlo support tools workshop
IPPP Durham, 17.03.2026

Decay reweighting

- No built-in way to evaluate **impact of decay BR uncertainties** in current EvtGen framework

- Custom decay tables often created by users (e.g. from different PDG versions)

  ↳ multiple samples required to study impact of these variations (limited resources)

EvtGen: generator that simulates decays of HF particles (primarily *B* and *D* mesons)

Solution: introduce decay weights to evaluate impact of decay table variations

↓

apply weights post-simulation to check variations impact on:

- physics observables
- efficiencies
- bkg rates
- ...

Strategy

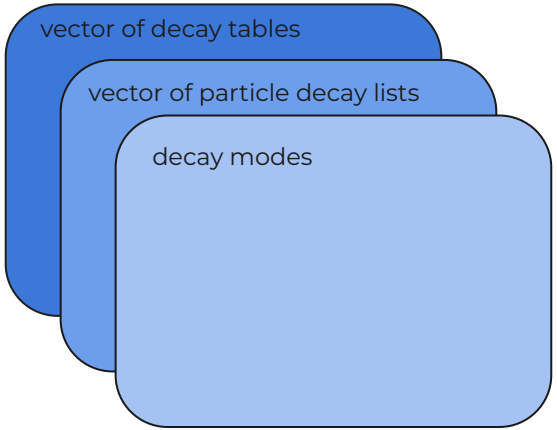
- **Setup** framework: simplified decay table versions with single-step chain (then expanded to full decay chains)
- Introduce **container** for decay weights
- Update EvtGen framework to handle **alternative decay tables**
- Implement **calculation of weights** based on BR variations in different decay tables (generalised to handle **full decay chains**)
- **Validation**: with custom tables and real tables from ATLAS
- Decay weights made accessible through **Rivet**
- Enable **selective reweighting** based on physics

Structure

default + alternatives

decay table object: contains multiple decay tables which contain multiple decay lists which store multiple decay modes

BR, daughters, model



weights implementation

where	**computation**
where EvtGen decides on the decay channel	$w_i = \frac{\mathrm{BR}_i^{\mathrm{alt}}}{\mathrm{BR}_i^{\mathrm{def}}}$

normalised BR from alternative decay table

normalised BR from default table

- randomly pick a decay mode from default table (based on default BR)
- for that mode: look up BR at same index in each alternative table
- compute weights for each alternative table
- store weights in particle object (to be used later for **reweighting events**)

$$w = \prod_{i \in \text{decay chain}} \frac{\mathrm{BR}_i^{\mathrm{alt}}}{\mathrm{BR}_i^{\mathrm{def}}}$$

Assumptions made so far:

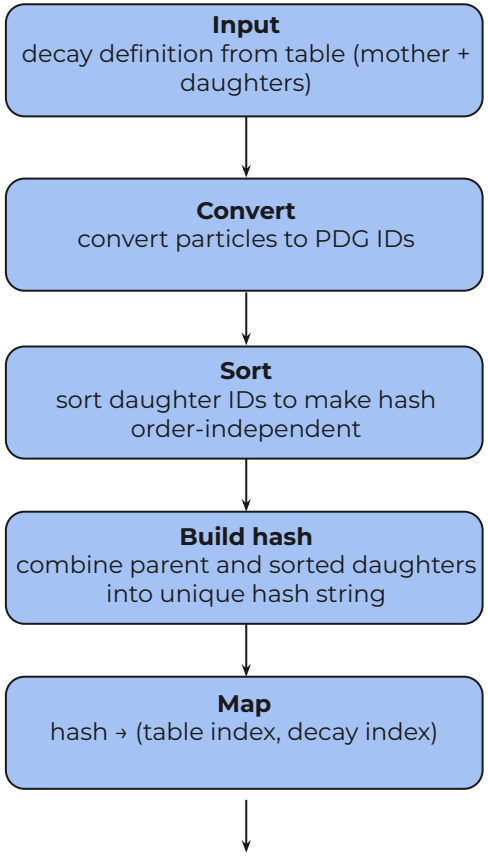
- all decay tables have the same number of modes per parent
- all decay modes in the same order in every table
- daughter particles all in the same order

complications for large tables

Hash mapping system

Method to identify and map decay modes to their associated BRs across different decay tables

- new class (EvtDecayHash)
- creates a unique identifier (hash) for each decay mode
- designed to be independent of the order of daughters



Example:

B0 → D- D+ B0 → D+ D-

both produce the same hash:

511: -411, 411

Hash mapping system

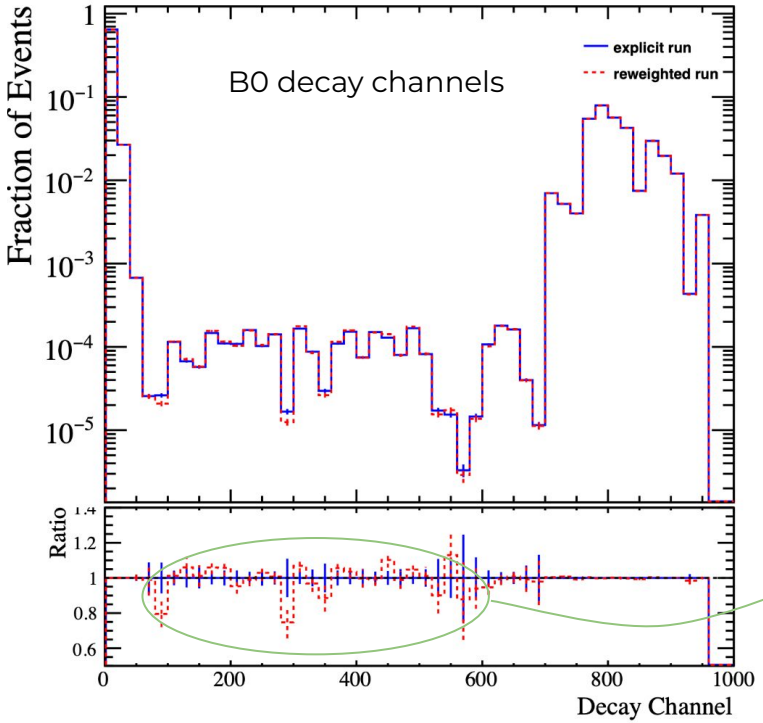
Problem: quark-level decays modelled by PYTHIA

→ same decay string, but different topology depending on the model

In these cases the **order** of the $q\bar{q}$ string (and the Pythia **model** used) matters

Solution: differentiate between PYTHIA (or other external tools, e.g. TAUOLA) and EvtGen native decays in the hash creation

Validation: EvtGen decay tables



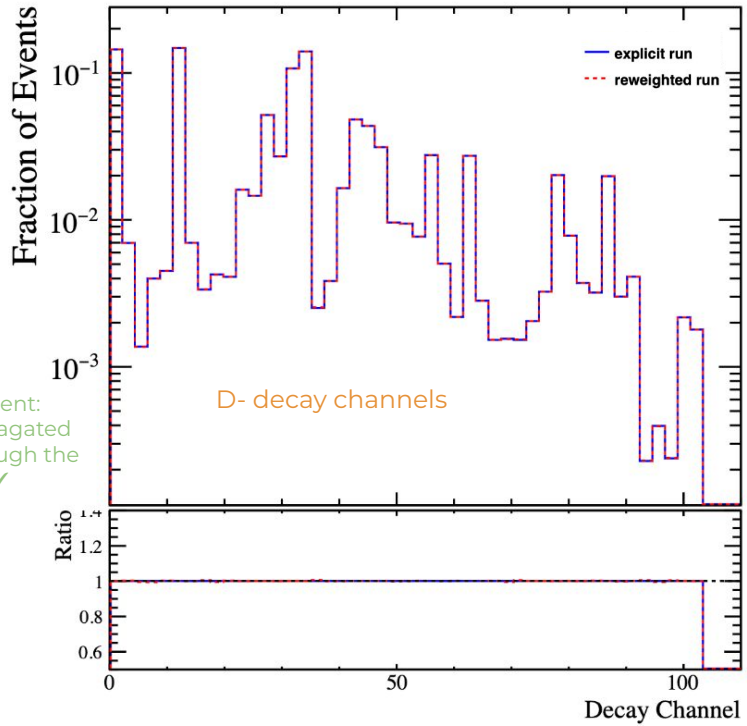
blue: events generated using the alternative decay table as default
red: events reweighted post-simulation

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events
- all decays from central decay table
- some BRs (and order) **modified** in alternative table

Reweighting and mapping work as expected ✓

agreement, except for deviations at low stats

Validation: EvtGen decay tables



good agreement: weights propagated properly through the decay chain ✓

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events
- all decays from central decay table
- some BRs (and order) **modified** in alternative table

Upsilon(4S) granddaughters

deeper into the decay chain { **B0 → D- pi+ pi0**; anti-B0 → D+ pi- pi0

blue: events generated using the alternative decay table as default
red: events reweighted post-simulation

Selective reweighting

Selective reweighting enabled ✓

- add **switches** to enable/disable reweighting based on physics (e.g. HF, strangeness, baryons/mesons, …)
- let users control which reweighting matters for the specific study
- being selective reduces random-walk weight accumulation, improving statistical stability

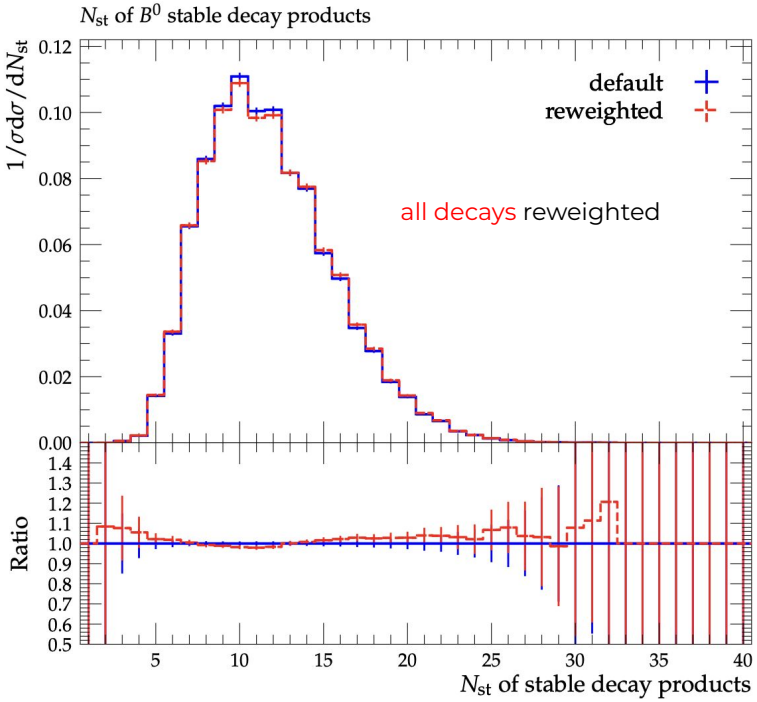
Implementation

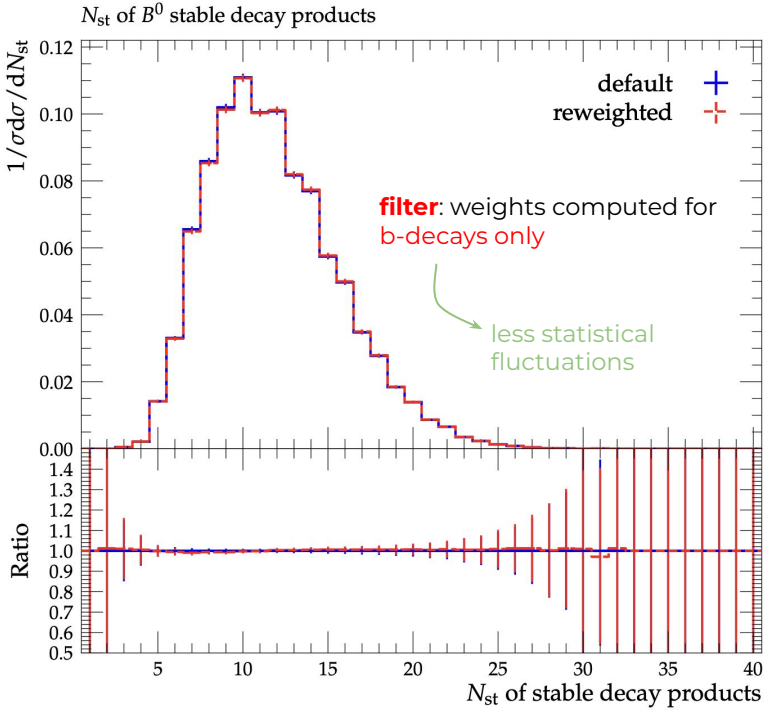
- Where: argument of the function that returns the total event weight (full decay chain)
- User-provided filter to select which decays are multiplicatively included

if no filter function is provided, weights are by default computed for all decays

Selective reweighting

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events
- all decays from central decay table
- some BRs (and order) **modified** in alternative table

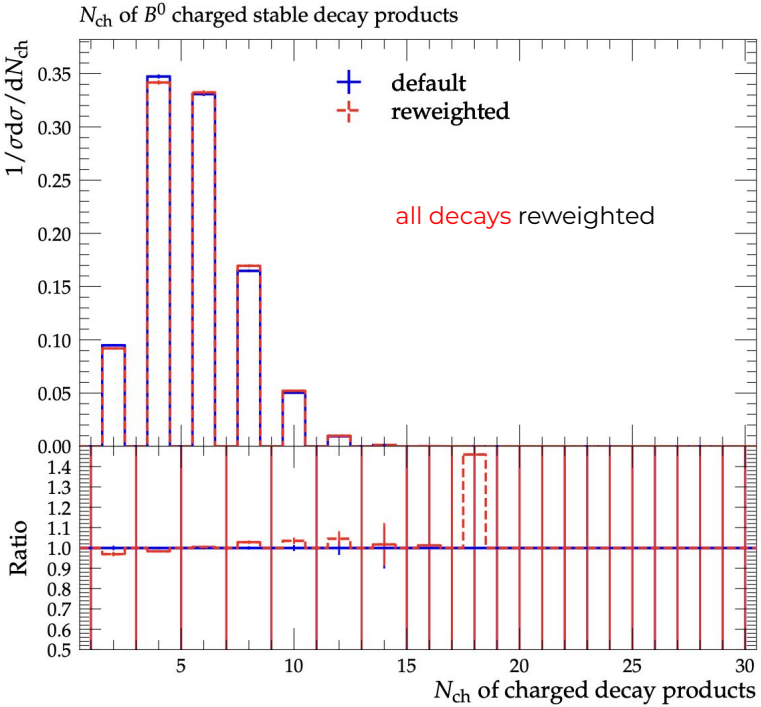


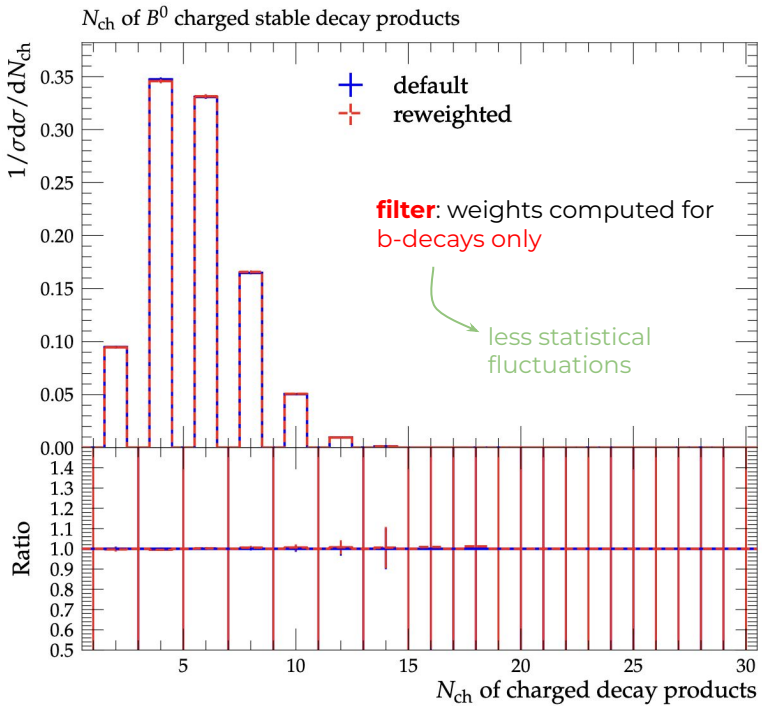


New: EvtGen weights written as standard HepMC3 named event weights, allowing Rivet to propagate them as variations into the YODA histograms

Selective reweighting

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events
- all decays from central decay table
- some BRs (and order) **modified** in alternative table





New: EvtGen weights written as standard HepMC3 named event weights, allowing Rivet to propagate them as variations into the YODA histograms

Validation: real tables

tables from **ATLAS athena**: https://gitlab.cern.ch/atlas/athena/-/tree/main?ref_type=heads

Issues: tables that come from experiments cannot be used to test the framework directly, as they come with inconsistencies

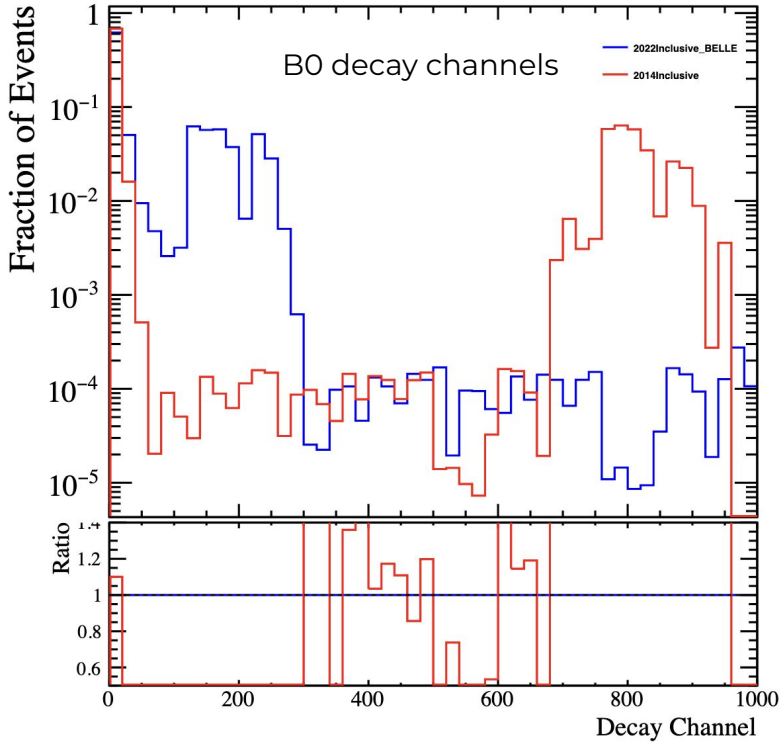
→ e.g. different pythia conventions, different lists of decays, different models / model parameters etc

→ Tables need to be made **compatible** for use

Example: real tables

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events

tables from **ATLAS athena**:
https://gitlab.cern.ch/atlas/athena/-/tree/main?ref_type=heads



Example:

blue: 2022Inclusive_BELLE.dec
red: 2014Inclusive.dec

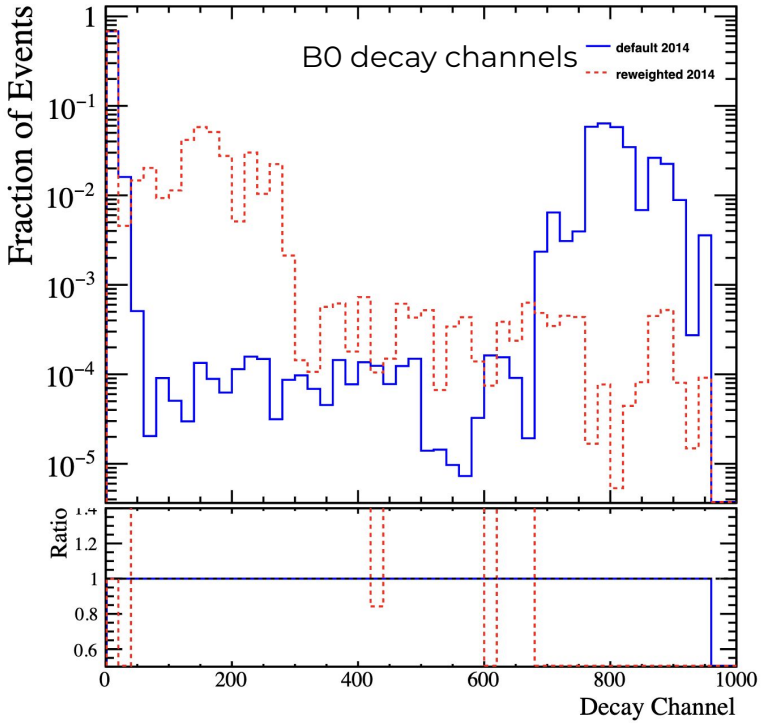
- the two tables differ not only in BRs, but in some decay models (kinematics) too
- different lists of decays and particle definitions in the two tables

→ cannot match exactly the two tables

Example: real tables

- **Upsilon(4S) → B0 anti-B0**
- model: VSS
- no FSR
- 10M events

tables from **ATLAS athena**: https://gitlab.cern.ch/atlas/athena/-/tree/main?ref_type=heads



blue: 2014Inclusive.dec as default
red: 2014Inclusive.dec reweighted from 2022Inclusive_BELLE.dec

Even if the actual decay modes are reweighted correctly, it cannot be visible from this plot

each bin contains several decay modes, and there are decays that are in one table but not in the other one

Validation: real tables

First attempt: to check that the explicit and reweighting runs are fully matching → Fill histograms only with the decays that are present in both tables

problem: total normalisation at generation stage would be different between the two tables

Solution: script to **prepare the tables** with the correct aliases, same pythia conventions, same number of decays (input: tables from experiments, output: modified tables ready for use)

→ give output tables as input to EvtGen, which then re-normalises the total BRs to 1

this will work even for different decay models, because the choice of the decay ID is based on the BR

Note: the modifications left are not going into the EvtGen source code, which is now complete and will be part of Release 3 (scheduled for next month)

Summary & next steps

- **Integration of decay weights** into EvtGen: completed
  - framework handles **alternative decay tables**
  - weights calculated based on **BRs** *
  - **selective reweighting** based on physics enabled
  - weights made accessible through **Rivet**

*note: reweighting based on kinematics not possible because current structure of EvtGen does not allow to have a second model associated with a decay (not until next modernisation campaign)

- EvtGen first **merge request**: under revision, positive feedback → add selective reweighting feature
  Link: https://gitlab.cern.ch/evtgen/evtgen/-/merge_requests/47

- In progress: script to prepare tables to give as input to EvtGen