Status of EvtGen

Thomas Latham
(on behalf of the Warwick EvtGen team)
11^{th} April 2018

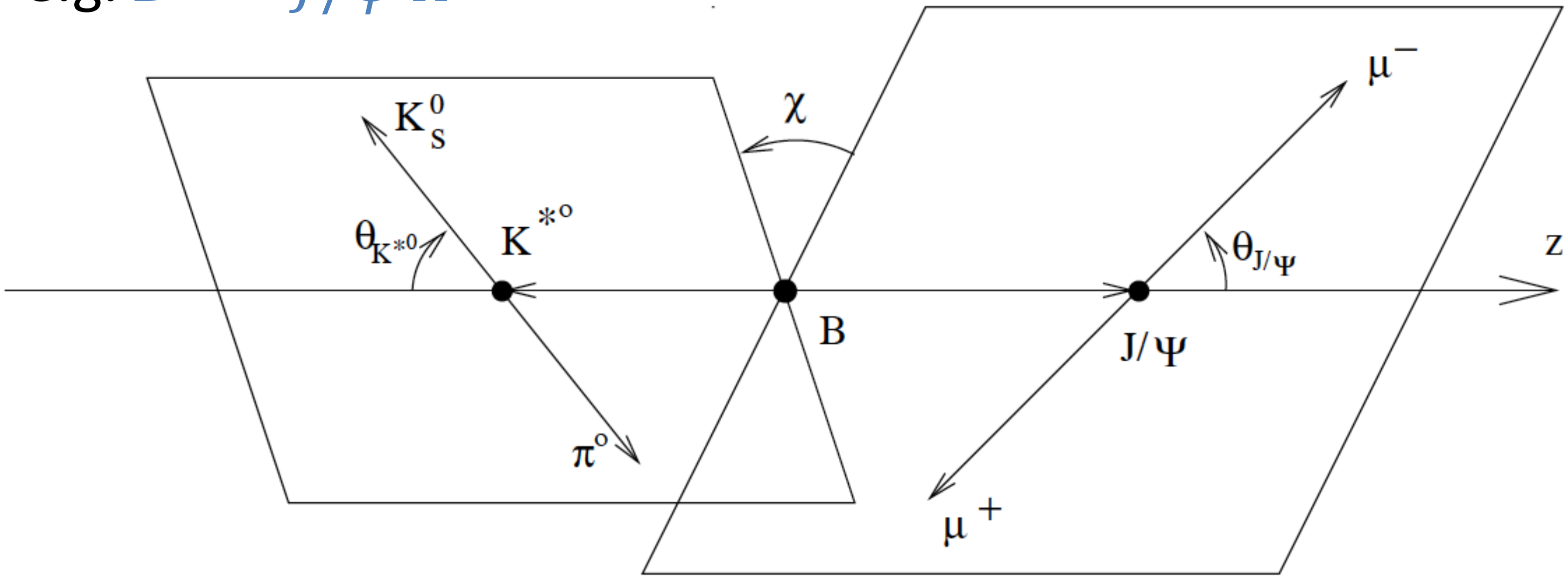
http://evtgen.hepforge.org

What is EvtGen?

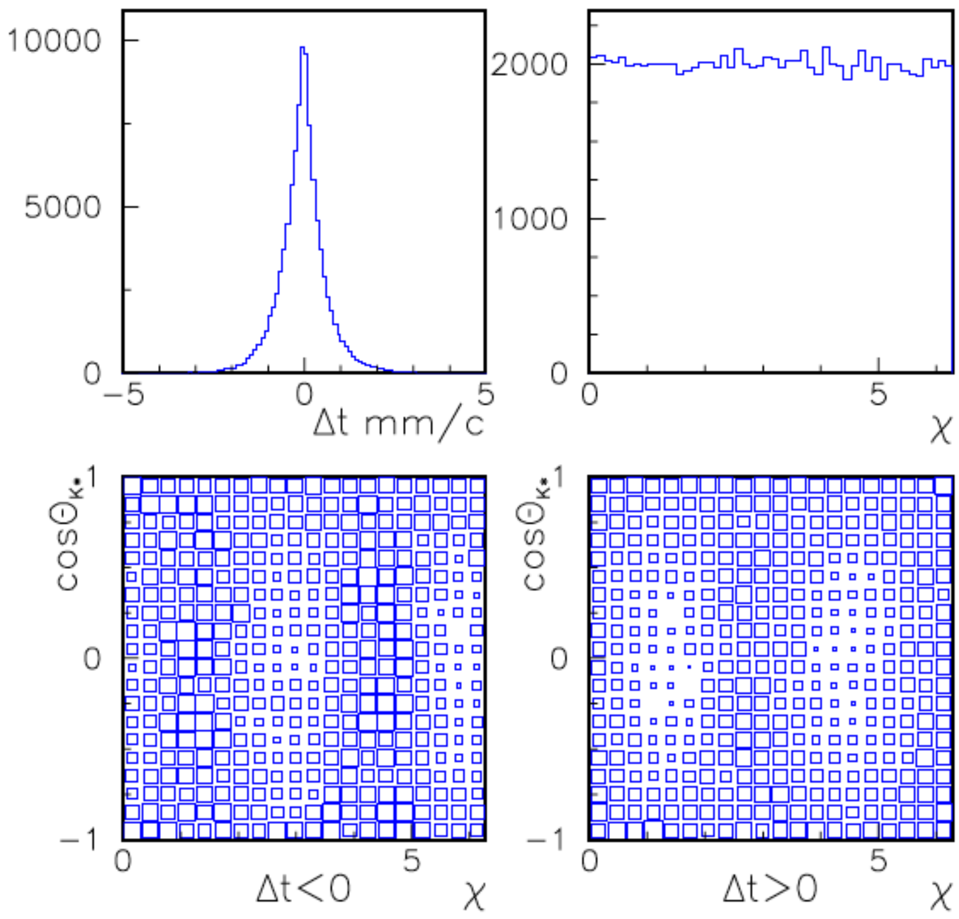
- EvtGen is a **generator of particle decays**
- Primarily for weak decays of **heavy flavour particles**, e.g. b-hadrons, c-hadrons, τ
- Implements the detailed decay dynamics, based on theoretical models
- Originally developed for BaBar and CLEO by Anders Ryd and David Lange
- Used by numerous experiments including ATLAS, Belle II, CMS, LHCb…

Motivation

- Originally designed to handle complicated decay chains
- Can involve correlations between different observables
- In particular, CP violating decays have complicated decay-time distributions, which can be correlated with the angular structure of the decay
- e.g. $B^0 \rightarrow J/\psi\, K^{*0}$



Motivation



10000
5000
0
−5
0
5
Δt mm/c
2000
1000
0
0
5
χ
cosΘ_K*
1
0
−1
0
5
Δt<0
χ
cosΘ_K*
1
0
−1
0
5
Δt>0
χ

Decay chains

- Heavy flavour decays often involve many sequential decays, e.g.
  $B^0 \to D^{*-}\tau^+\nu_\tau$; $D^{*-} \to \overline{D}^0\pi^-$; $\tau^+ \to \pi^+\bar{\nu}_\tau$; $\overline{D}^0 \to K_S^0\pi^+\pi^-$
- To provide a reusable generic tool, need to simulate correctly the full decay tree but implement only the nodes of the tree
- Achieved by using decay amplitudes instead of probabilities
- The modular design and C++ implementation of the package also help to achieve this

EvtGen decay algorithm

1. Input: parent particle ID and 4-momentum
2. Completely determine the decay tree ← Input from decay file
3. Determine properties of all particles in decay tree ← Input from database
4. Accept/reject to determine kinematics according to dynamics model

Decay amplitudes

- EvtGen uses decay amplitudes to model sequences of decays

$$A = \sum_{\lambda_{D^*}\lambda_\tau} A^{B\to D^*\tau\nu}_{\lambda_{D^*}\lambda_\tau} \times A^{D^*\to D\pi}_{\lambda_{D^*}} \times A^{\tau\to\pi\nu}_{\lambda_\tau}$$

- First generate kinematics of *B* decay according to phase space and perform accept/reject based on

$$P_B = \sum_{\lambda_{D^*}\lambda_\tau} |A^{B\to D^*\tau\nu}_{\lambda_{D^*}\lambda_\tau}|^2$$

Decay amplitudes

- After decaying the B, need to propagate the spin state information to the subsequent decays
  - Calculate the spin-density matrix for the D^*

$$\rho^{D^*}_{\lambda_{D^*}\lambda'_{D^*}} = \sum_{\lambda_\tau} A^{B\to D^*\tau\nu}_{\lambda_{D^*}\lambda_\tau}[A^{B\to D^*\tau\nu}_{\lambda'_{D^*}\lambda_\tau}]^*$$

- From this can then generate the decay of the D^*, etc.

$$P_{D^*} = \frac{1}{\mathrm{Tr}\,\rho^{D^*}} \sum_{\lambda_{D^*}\lambda'_{D^*}} \rho^{D^*}_{\lambda_{D^*}\lambda'_{D^*}} A^{D^*\to D\pi}_{\lambda_{D^*}}[A^{D^*\to D\pi}_{\lambda'_{D^*}}]^*$$

Decay amplitudes

- The spin-density matrix for the τ can be determined using the information from the D^* decay

$$\tilde{\rho}^{D^*}_{\lambda_{D^*}\lambda'_{D^*}} = A^{D^*\to D\pi}_{\lambda_{D^*}}[A^{D^*\to D\pi}_{\lambda'_{D^*}}]^*$$

$$\rho^{\tau}_{\lambda_\tau\lambda'_\tau} = \sum_{\lambda_{D^*}\lambda'_{D^*}} \tilde{\rho}^{D^*}_{\lambda_{D^*}\lambda'_{D^*}} A^{B\to D^*\tau\nu}_{\lambda_{D^*}\lambda_\tau}[A^{B\to D^*\tau\nu}_{\lambda'_{D^*}\lambda'_\tau}]^*$$

Decay amplitudes

- Essentially, each node in the tree is implemented as a particular “decay model”
- The EvtGen framework then handles the bookkeeping needed to correctly generate the full tree (using the spin-density matrices)
- Each part of the decay chain can then be generated separately – making the whole process much more efficient
- The decay models only need to provide the decay amplitudes – the framework handles the rest

Decay models

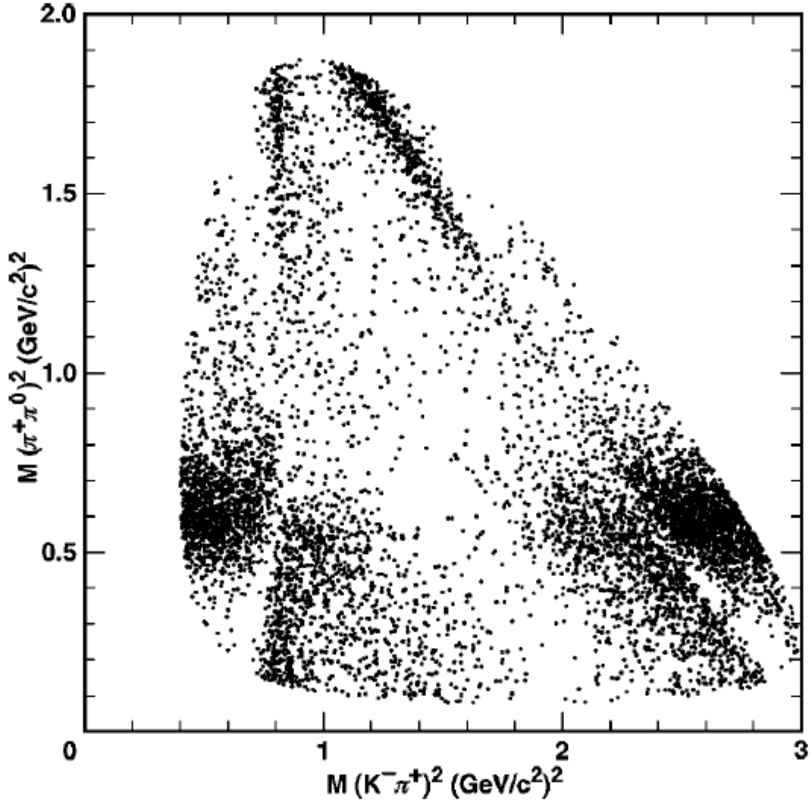
- Many decay models available:
  - General purpose models that decay according to specified helicity or partial wave amplitudes
  - Semi-leptonic form-factor models
  - Dalitz plot decays (some specific models and a generic Dalitz model)
  - Specific models for rare electroweak penguin / radiative decays, e.g. $B \to K^{(*)}\mu\mu$ and $b \to s\gamma$
  - Many models have versions that include CP-violation

- Interface with external packages for additional features:
  - HepMC : For writing events in HepMC format (mandatory)
  - Photos^{++}: FSR γ (optional)
  - Pythia8: Generic decays that have no specific EvtGen model (optional)
  - Tauola^{++}: tau decays (optional)

Example of Dalitz plot

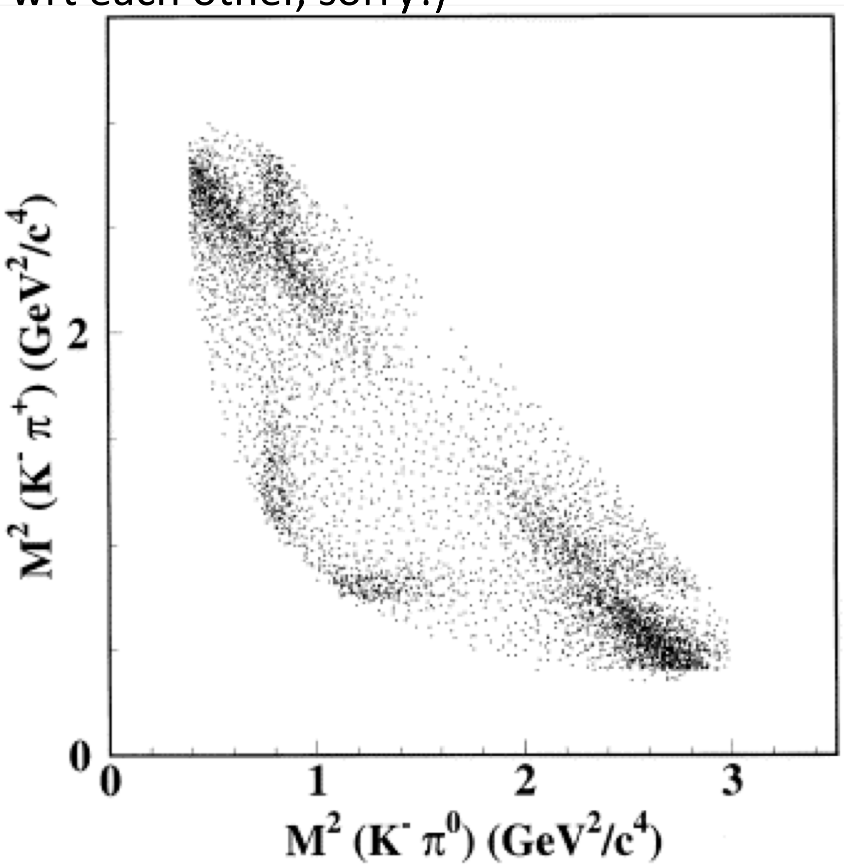
$$D^0 \to K^- \pi^+ \pi^0$$

(NB the plots are rotated wrt each other, sorry!)



CLEO data

[Phys. Rev. **D63**, 092001 (2001)]



EvtGen simulation

[Nucl. Instrum. Meth. A **462**, 152 (2001)]

Decay files

- The generic decay file provided with EvtGen contains extensive list of particle decays
- Updated from PDG at intervals
- Each decay is specified with a branching fraction, list of daughter particles and the decay model
- Provided as both text and XML files

```
<decay name=“anti-B0”>
 <channel br=“0.00044”
          daughters=“J/psi K_S0”
          model=“SVS”/>
 <channel br=“0.00044”
          daughters=“J/psi K_L0”
          model=“SVS”/>
</decay>
```

Decay models

- Documentation exists for the various decay models and how their parameters should be specified in the decays files

  https://evtgen.hepforge.org/doc/models.html
- Some directly on the webpage (e.g. below) or in the original user guide

EvtSSDCP

This model simulates the decay of a B meson to a scalar and one other particle of arbitrary (integer) spin. It expects either 8, 12 or 14 model parameter arguments. An example of using this model is $B \rightarrow J/\psi K_S$:

```
Decay B0
1.000 J/psi K0S    SSD_CP dm dgog |qop| arg(qop)
                   |Af| arg(Af) |Abarf| arg(Abarf)
                   |Afbar| arg(Afbar) |Abarfbar| arg(Abarfbar)
                   |z| arg(z);
Enddecay
```

where dm is the mass difference of the two mass eigenstates (in units of hbar/s), dgog is $2y = 2(\Gamma_H - \Gamma_L)/(\Gamma_H + \Gamma_L)$. The value qop is q/p where $|B_{L,H}\rangle = p|B^0\rangle \pm q|\text{anti-}B^0\rangle$. The values Af and Abarf are the amplitudes for the decay of a B^0 and a anti-B^0, respectively, to the final state f. The set of amplitudes Afbar and Abarfbar corresponds to the decay to the CP conjugate final state. These amplitudes are optional and are by default $A_{fbar} = Abar^*_f$ and $Abar_{fbar} = A^*_f$, consistent with CPT for a common final state of the B^0 and anti-B^0. However, in modes such as $B \rightarrow D^* \pi$ it is useful to be able to specify these amplitudes separately.

The example below shows the decays $B \rightarrow J/\psi K_S$ and $B \rightarrow J/\psi K_L$:

```
Define dm 0.472e12
Define minusTwoBeta -0.85

Decay B0
  0.5000 J/psi K0S   SSD_CP dm 0.0 1.0 minusTwoBeta 1.0 0.0 -1.0 0.0;
  0.5000 J/psi K0L   SSD_CP dm 0.0 1.0 minusTwoBeta 1.0 0.0  1.0 0.0;
Enddecay
```

Note that the sign of the amplitude for the anti-B^0 decay have the oposite sign for the K_S since this final state is odd under parity.

EvtGen development

- Warwick team responsible for package development since 2010 following merge of various experiments' forks by Anders Ryd in 2008

- Development team are LHCb collaboration members
  - LHCb currently uses its own version of EvtGen, which is manually "synchronised" with master repository
  - There are some (minor) technical differences, but they use the same physics models
  - Continually incorporating bug fixes/new physics models:

    LHCb $\leftrightarrow$ master

EvtGen development

- EvtGen master repository, web pages, and mailing lists recently migrated to HepForge
- Web-pages with documentation, bug tracker, etc.: http://evtgen.hepforge.org

- Also migrated the version control system from svn to git
- Currently, only Warwick EvtGen developer team have write access to repository
- Guest read access at http://evtgen.hepforge.org/git/evtgen.git
- HepForge will soon transition to a new web-based platform (similar to github/gitlab), which will make it much easier to add new users and configure permissions, perform code reviews, etc.

EvtGen development

- New tag of package every ~6 months, includes:
  - Bug fixes, model improvements, performance improvements
  - Contributions from users, e.g. new models
  - Changes in external packages (e.g. Photos^{++})
- Current tagged version is 1.7.0 (released 13th December 2017)

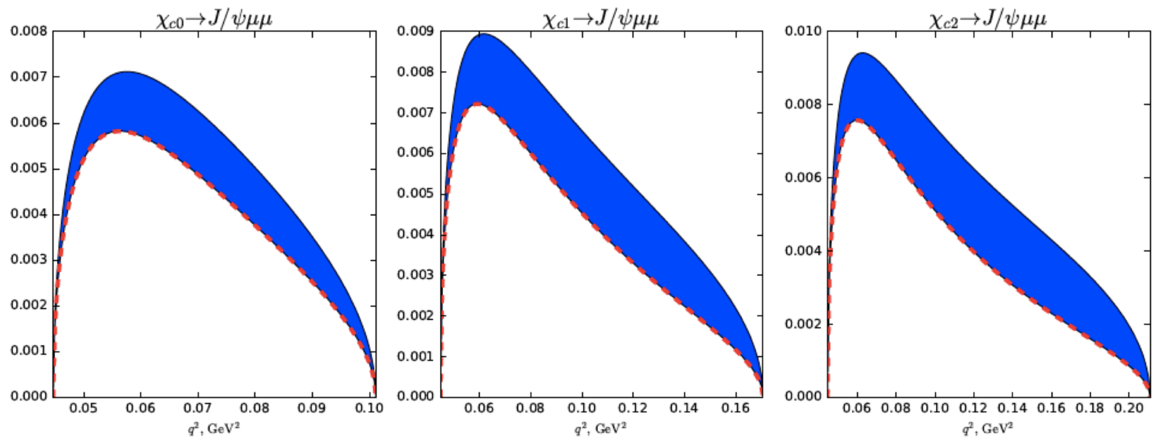
- New versions announced via mailing list:

  evtgen-announce@projects.hepforge.org
  - Should contain MC contacts from all relevant experiments
- Ask us questions using email:

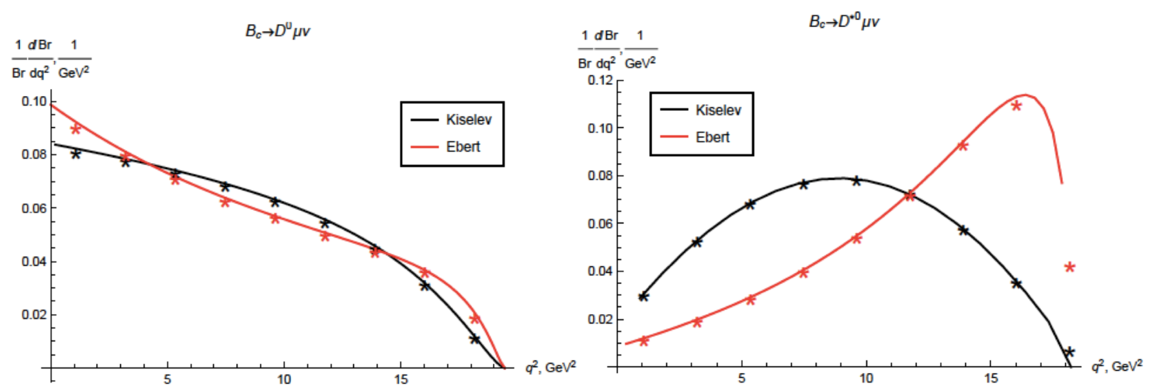
  evtgen@projects.hepforge.org

Recent developments

- Modified EvtSVP, EvtVVP and EvtTVP models to handle both radiative and two-lepton decays



- Updated EvtBcXMuNu models (X = Scalar, Vector, Tensor) to generate $B_c^+ \to D^{(*)0}\mu^+\nu_\mu$ decays

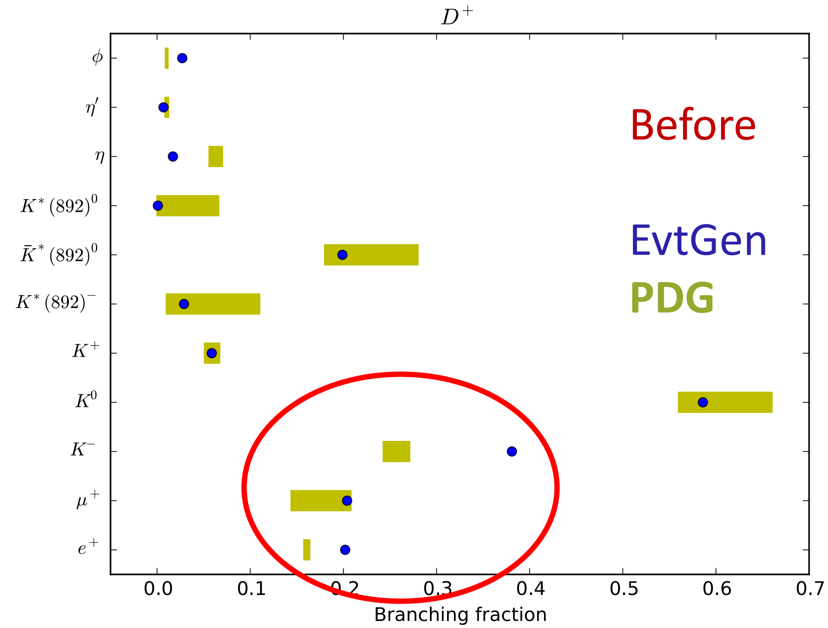


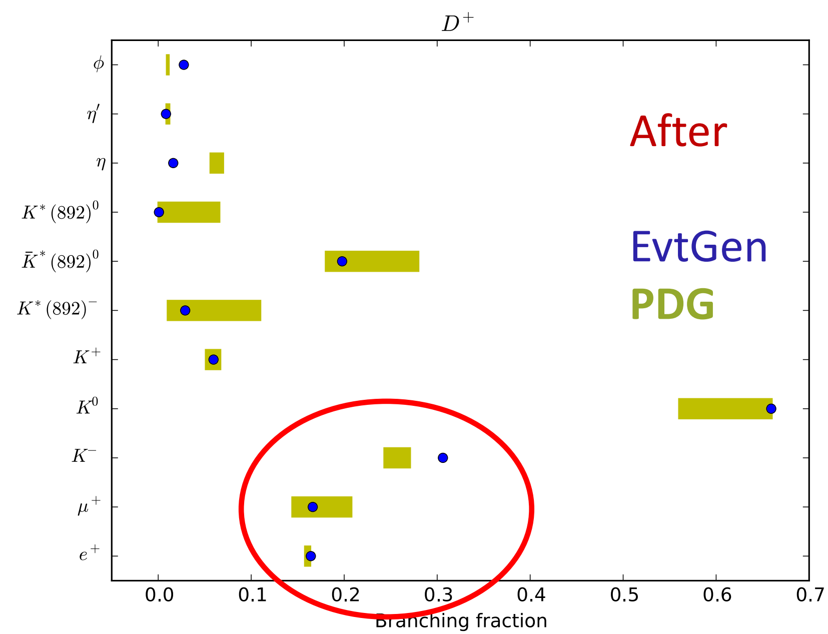
Recent developments

- Modified EvtFlatQ2 model to work for all $B \to Xll$ decay modes
- Bug fixes for:
  - DDalitz modes (daughter ordering)
  - Corrected classification of $B_s^0 \to \bar{K}^{*0}\mu^+\mu^-$ decay as $b \to d$ transition
  - Missing registration of VTOSLL model
- Converted three-body Fortran decay models to C++ – no more Fortran code in package
- Updates to use latest versions of Pythia, Photos and Tauola

Tuning of generic decay table

- Tuning of branching fractions in generic decay table performed in 2015/16
- Attempt to improve agreement with PDG for inclusive BFs
- Focus was mainly on semi-leptonic decays but BFs of some hadronic decays of charm hadrons also modified
- Non-trivial manual effort required – many tricky details
- Changes included in release 1.6.0





Future plans

- Addition of further models for baryon decays
- Further updates/tuning of generic decay information
- Improving mechanisms for keeping consistent particle properties used by EvtGen/Pythia/etc.

- Improve support for multi-threaded applications
- General code modernisation
  - Greater ease of future maintenance
  - Likely some performance increase (to be determined)
- Improved testing and validation suite with continuous integration

- Significant improvements to the documentation are also planned:
  - Additional online documentation (Doxygen)
  - Updated user manual
  - Journal publication

Conclusion

- EvtGen designed to provide detailed models for simulating decays of heavy flavoured particles
- Used by numerous experiments with a range of physics goals
- Recently migrated to HepForge site for hosting of git repository, webpages, mailing lists, etc.
  - Forthcoming new development platform should make more straightforward interaction between development team and community, e.g. bug reports, contribution of new models

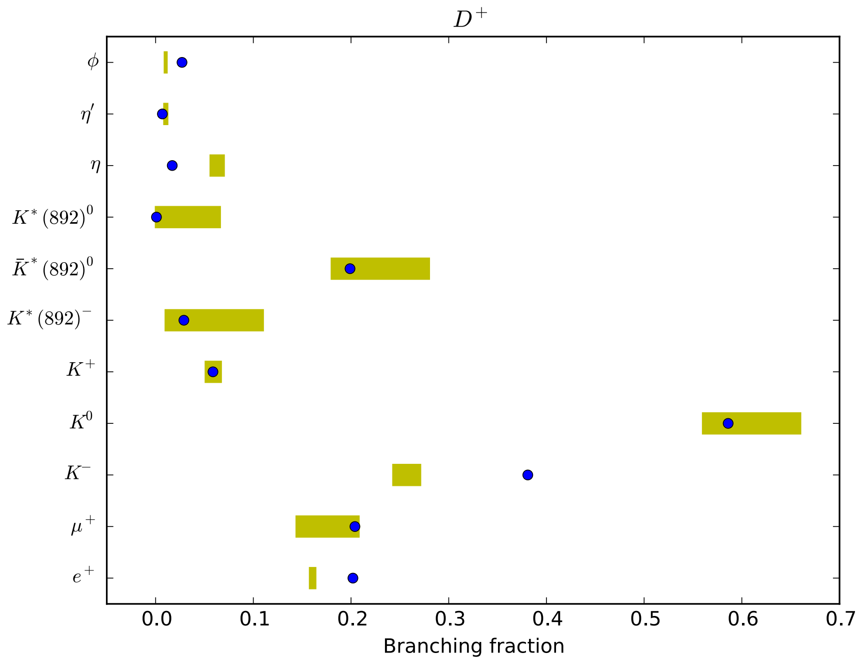
- Significant effort made ~2 years ago to tune inclusive decays in generic decay table - is there a pressing need to repeat this exercise?
- Planning gradual revision of all classes in package for modernisation, improving support for multithreading, documentation
- We welcome input from the community about prioritising these various efforts

http://evtgen.hepforge.org/

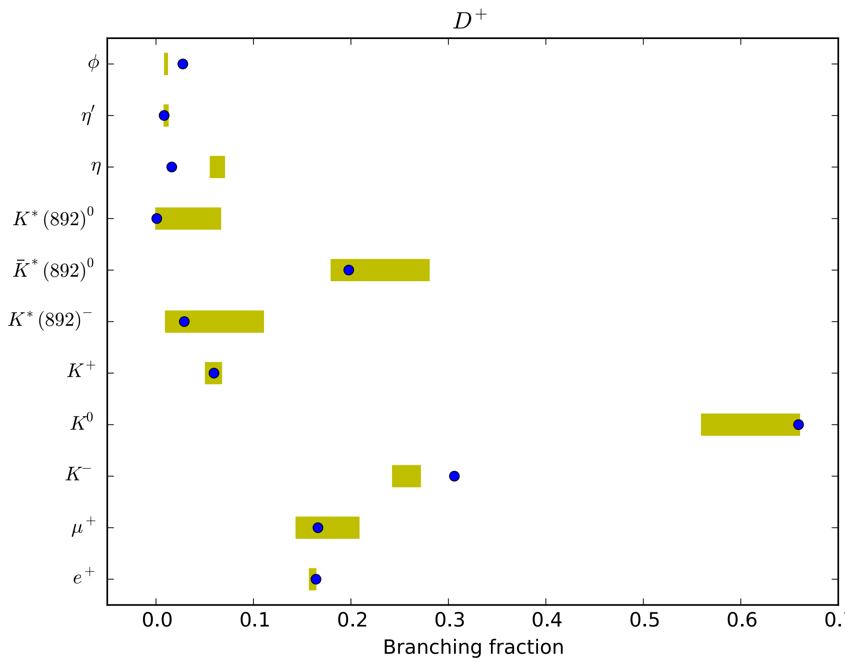
Backup

BF tuning: D^+

Before

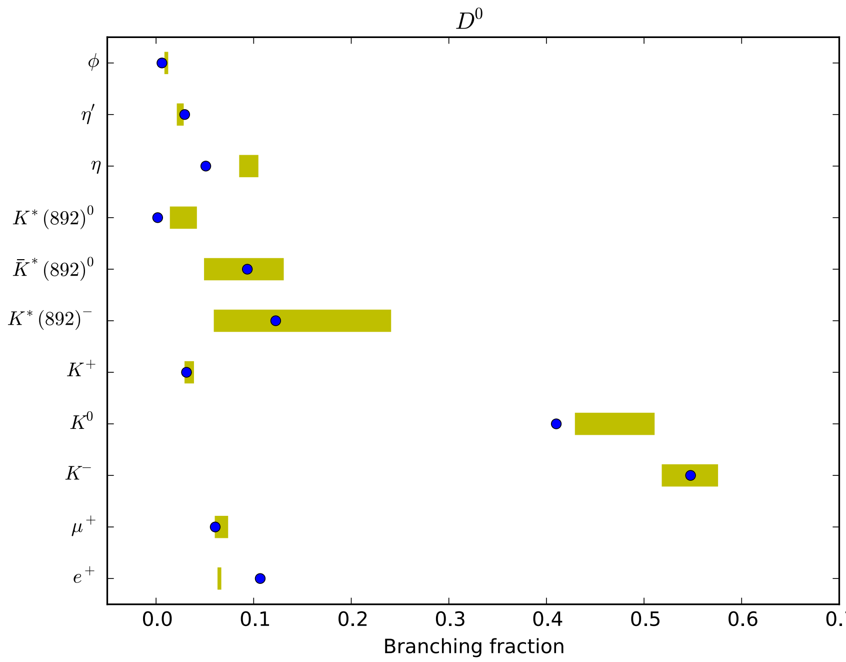


After

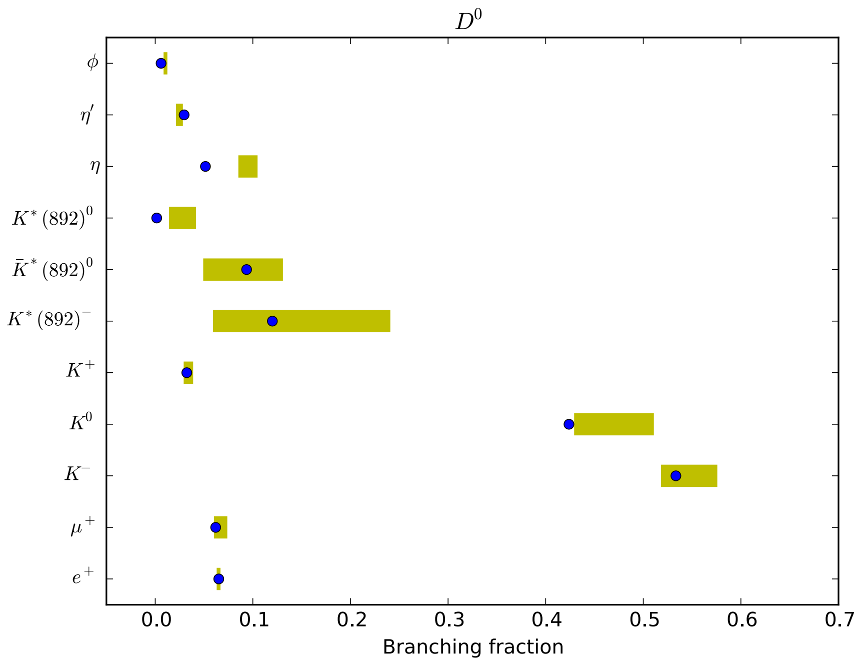


BF tuning: D^0

Before

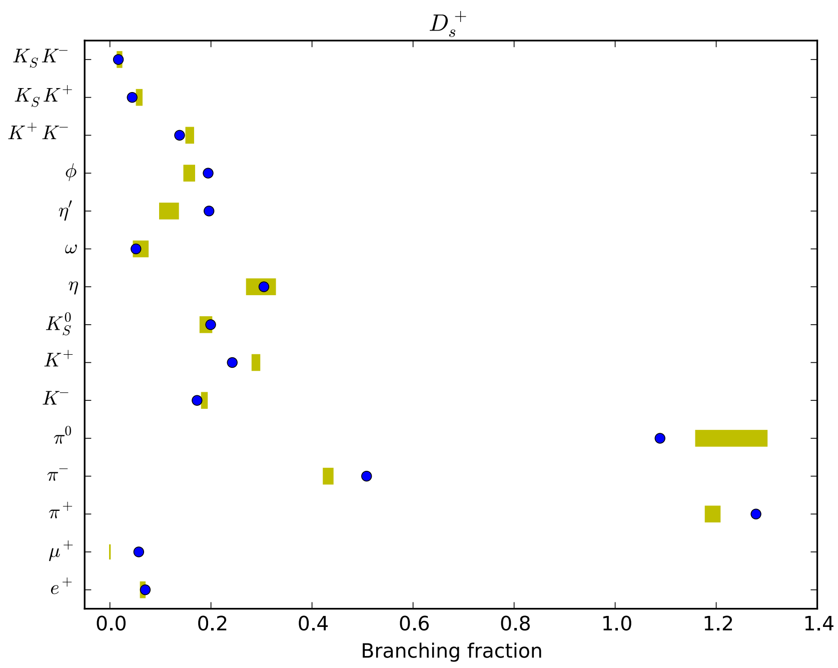


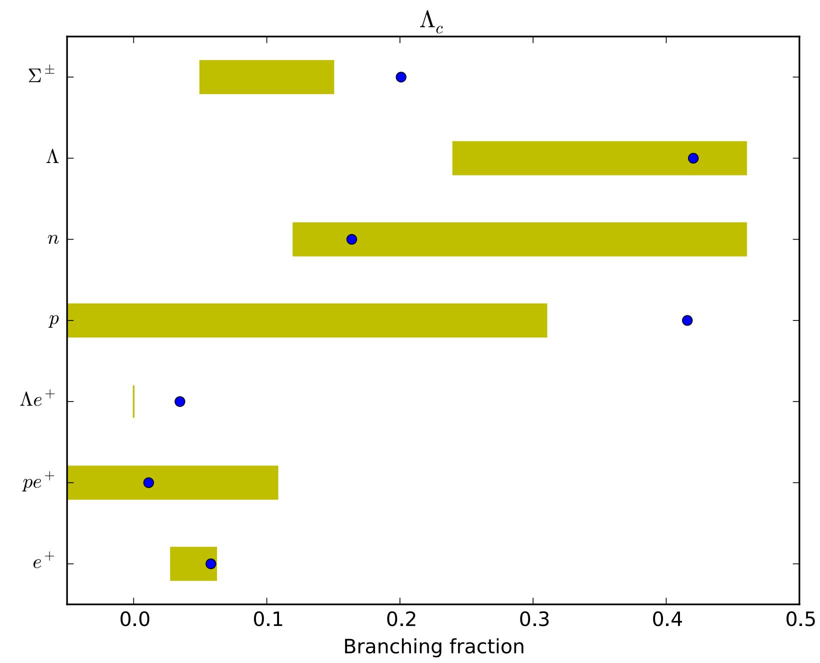
After



BF tuning: D_s^+ and Λ_c^+

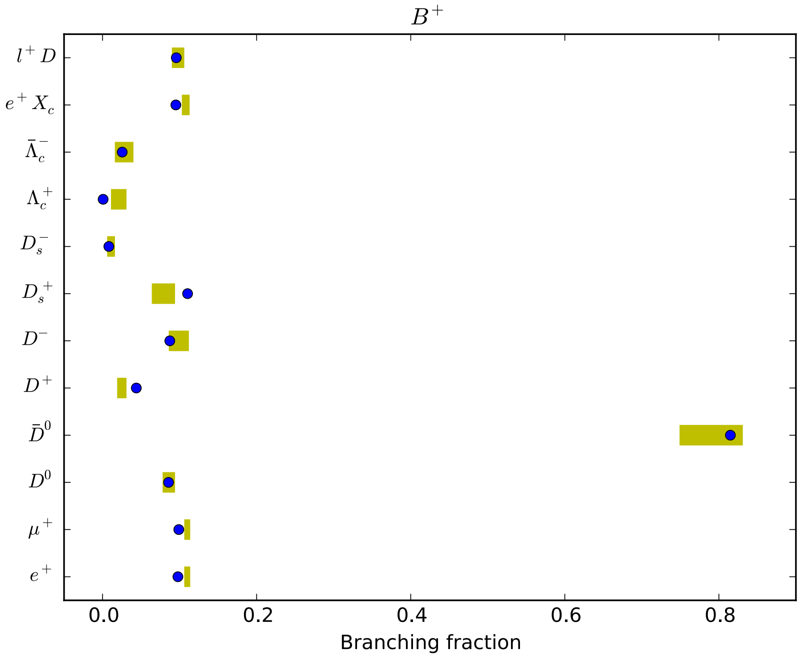
D_s^+ No changes made **Λ_c^+**



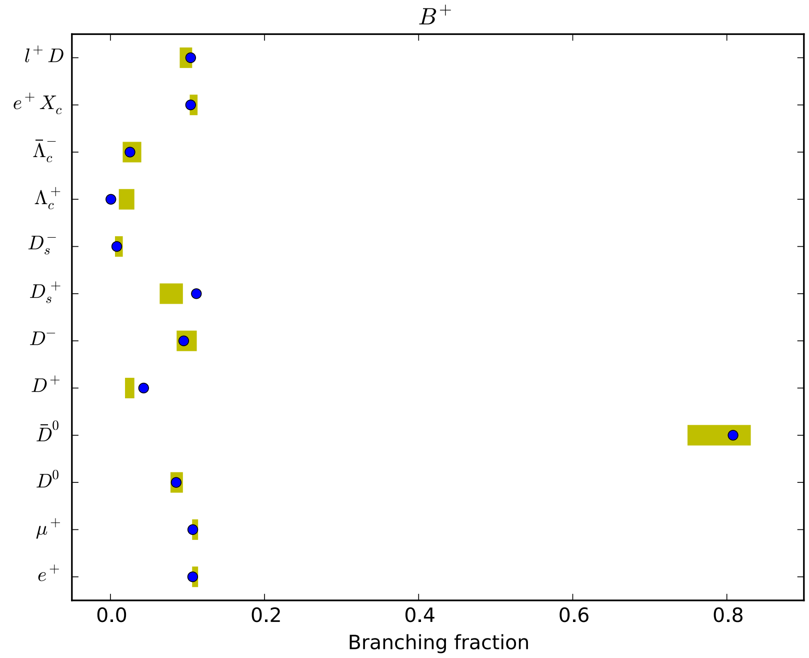


BF tuning: B^+

Before

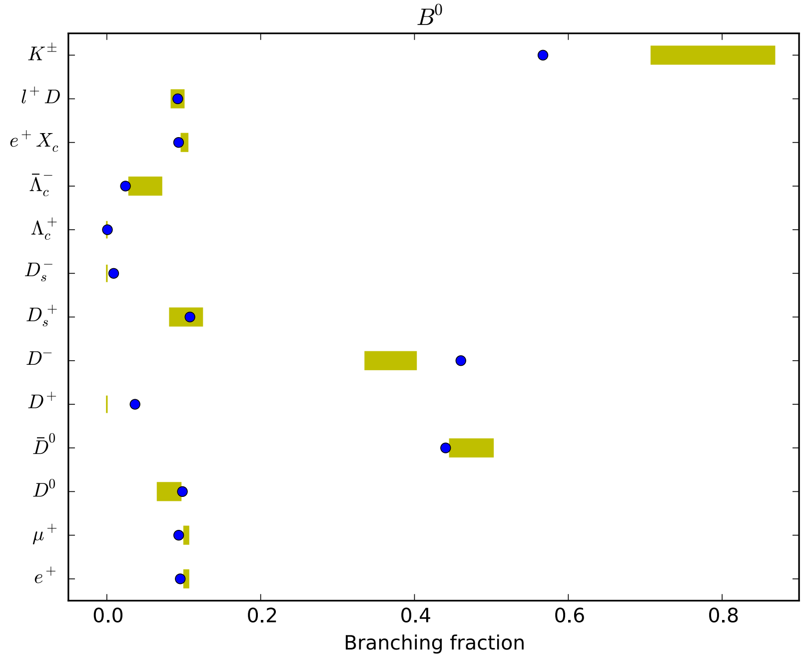


After

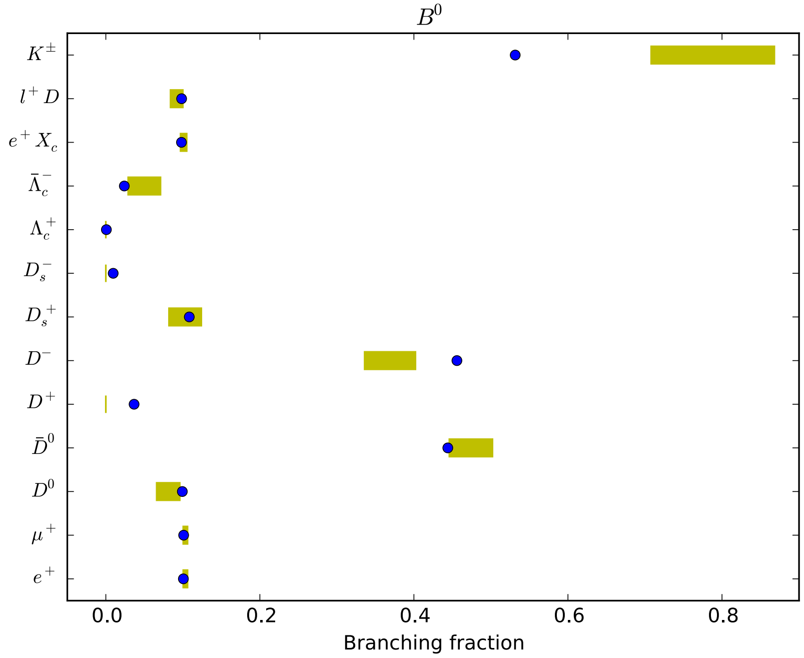


BF tuning: B^0

Before



After



BF tuning: B_s^0 and Λ_b^0

B_s^0 No changes made **Λ_b^0**