Using EvtGen for Inclusive Decays in ATLAS

Juerg Beringer
Lawrence Berkeley National Laboratory

- Introduction
- Example: b tagging performance on MC events
- Issues when using EvtGen together with other generators
- Conclusions

Introduction

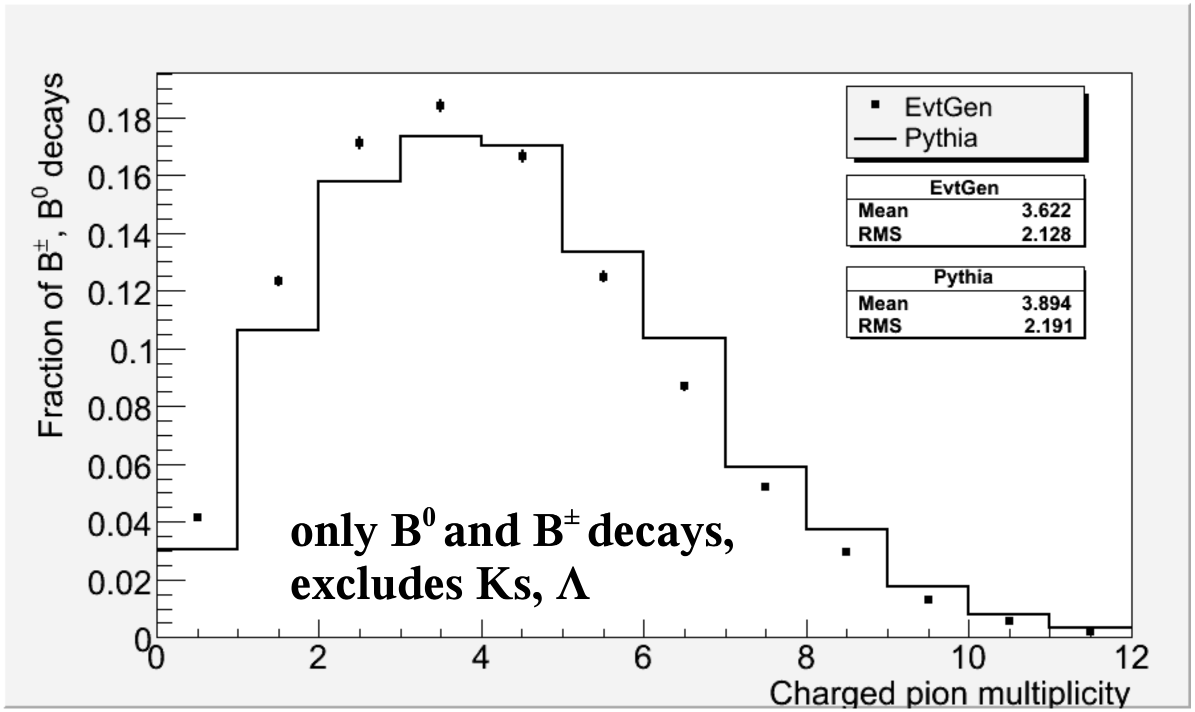
- **“Inclusive Decays” - use EvtGen to simulate “all” decays**
  - Instead of default decay simulation provided by Pythia, Herwig, ...
  - NB: long-lived strange particles (Ks etc) still passed to Geant
- **Does EvtGen's more detailed simulation of B and charm decays matter for high-p_t physics?**
- **Use EvtGen only for certain exclusive decay channels?**
  - Same decay simulated differently, depending on whether particle is prompt or a daughter in a particular exclusive decay (e.g. prompt D decayed by Pythia, D from B decayed by EvtGen)
  - Systematics?
- **Disclaimer**
  - Based on work done in CDF (2005-2006) and ATLAS (2006-2007)
  - May not be up-to-date with most recent developments regarding EvtGen

Example: ATLAS b Tagging Performance (I)

- **Does b tagging performance measured in a MC ttbar sample change if particles are decayed by EvtGen instead of Pythia?**
  - Pythia 6.403
  - ATLAS EvtGen version w/extended decay table for inclusive decays

- **Generate generic ttbar sample, simulating decays either with**
  - standard Pythia
  - EvtGen

- **Results: e.g. different particle multiplicities**
  - E.g. charged pion multiplicity (cf ARGUS: 3.58 ± 0.07)

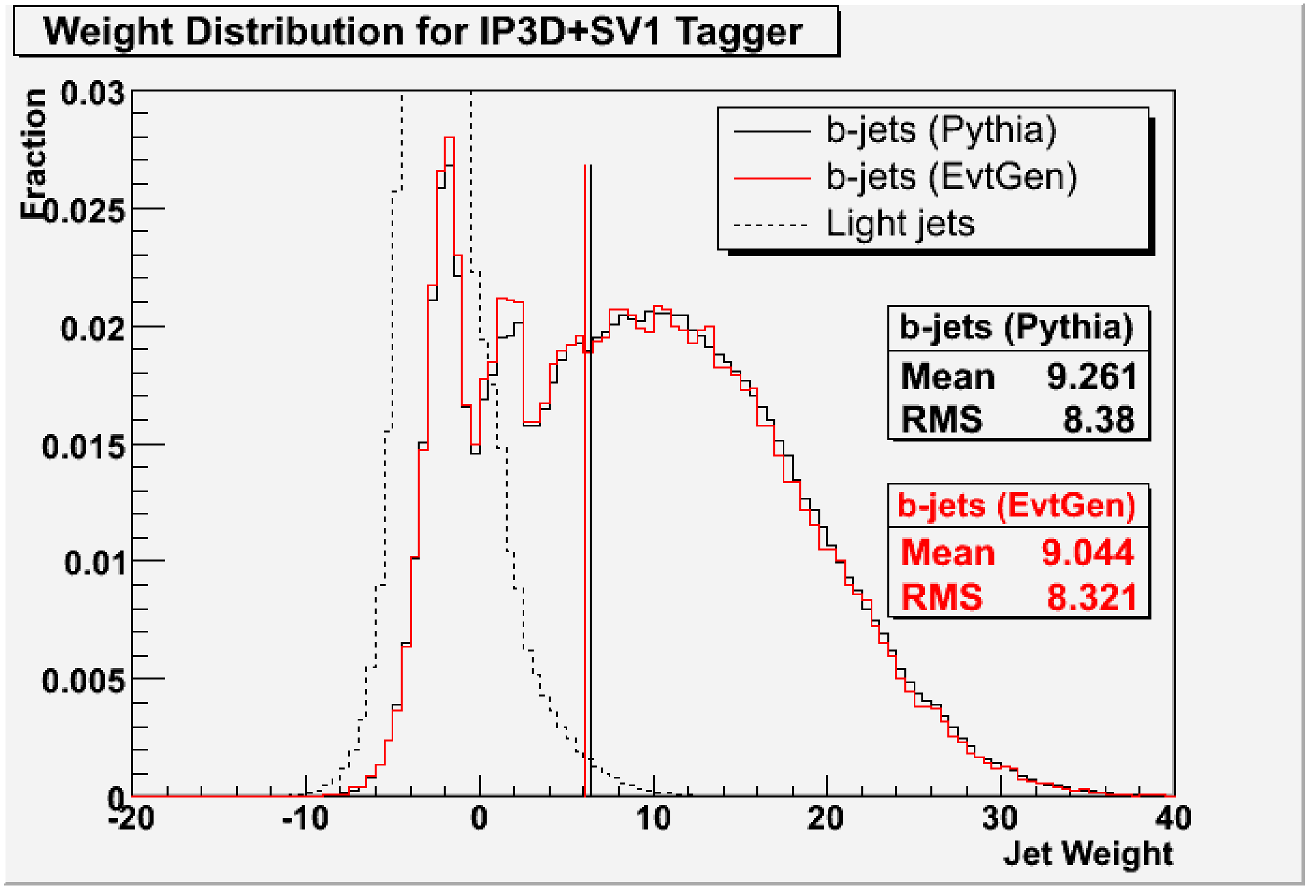


only B^0 and $B^\pm$ decays, excludes Ks, Λ

Example: ATLAS b Tagging Performance (II)

- **Run a standard ATLAS b tagging algorithm (based on signed impact parameter of tracks and secondary vertex reco)**
  - Determines a weight that users can cut on

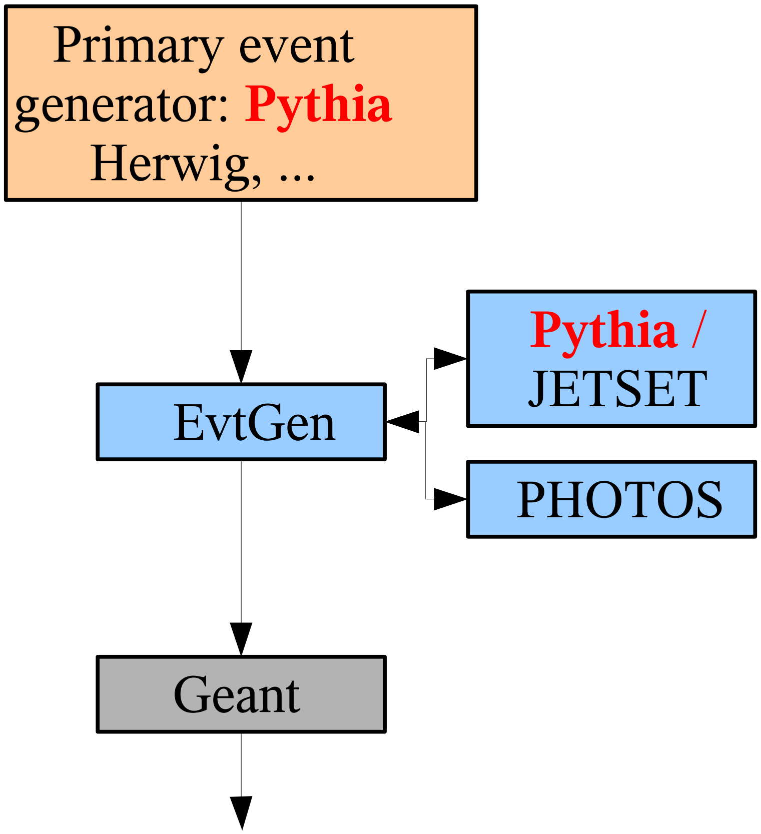
- **Results**
  - Small shift of weight distrib. for b-jets
  - b tagging eff. decreases by about 1%
  - Light-jet rejection at fixed b eff. decreases by ~ 5% to 15% (depending on tagger, b eff.)



EvtGen Interface for Inclusive Decays

- **Receives events from primary generator via HepMC data structure**
  - Primary generator may or may not have simulated decays
  - Tested with Pythia and Herwig (any generator should work)
- **Interface can be configured to**
  - Decay only un-decayed particles for which EvtGen has defined decay modes, or remove existing decays and re-decay the corresponding particles
    - HepMC data structure traversed recursively to determine what to decay
  - Flexible specification of what to decay using black- and whitelists
- **Decay and particle definition tables based on latest versions from BaBar and CDF as of summer 2005**
  - Includes Bs and b baryon decays
  - Known exclusive branching fractions complemented by generic decays to obtain 100% (no rescaling needed)

Issue: Rewriting of Fortran Common Blocks

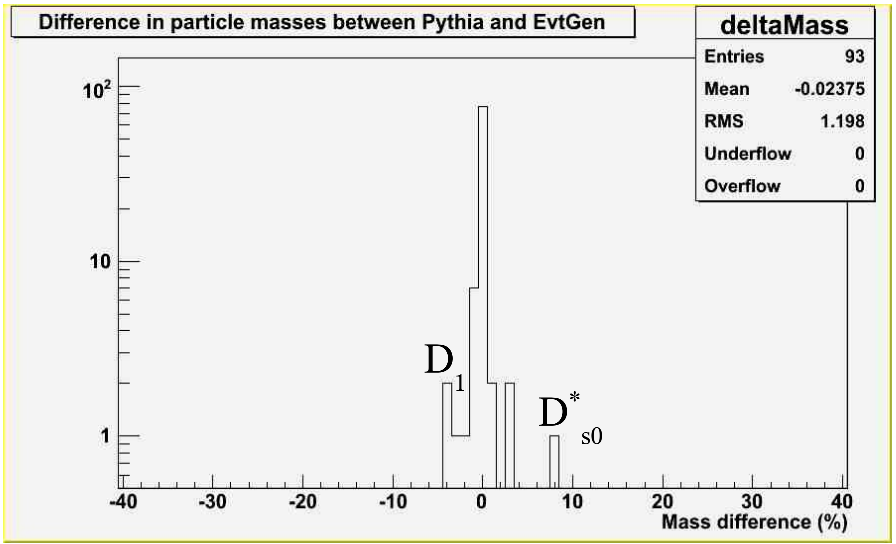


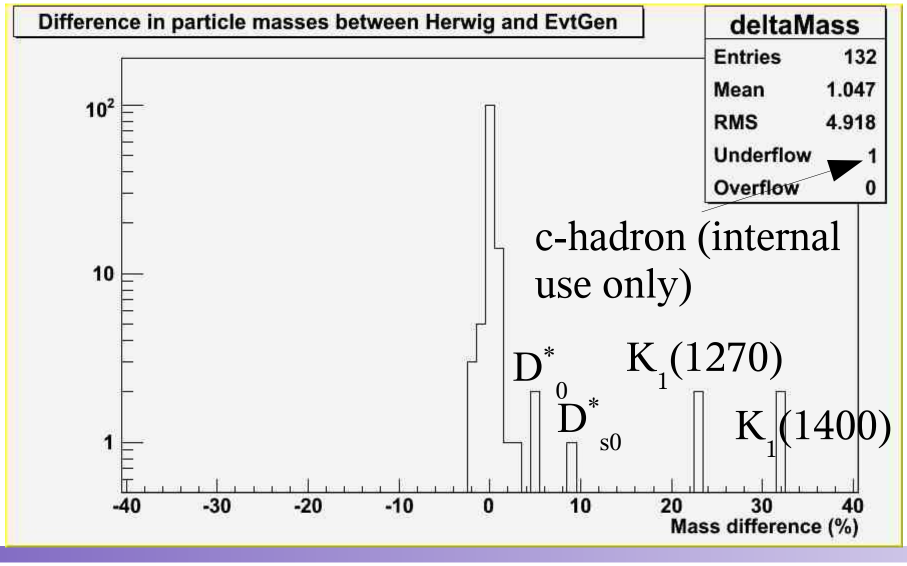
- **EvtGen calls back Pythia/JetSet to simulate generic decays**
  - “Unknown part” of BF
  - Needs specific Pythia configuration
- **Problem arises if primary generator is also Pythia**
  - By default, EvtGen rewrites Pythia configuration at time of first generic decay
- **Solutions (both implemented in ATLAS)**
  - Save/restore Pythia common blocks
  - Bundle specific version of JETSET (renamed common blocks) with EvtGen

Issue: Inconsistent Particle Definitions

- **Particle masses evolve, some PDG codes have changed**
  - Updates of particle tables used by different generators and by EvtGen have not been synchronized
  - When interfacing EvtGen to another generator, must synchronize or translate PDG codes and resolve mass differences

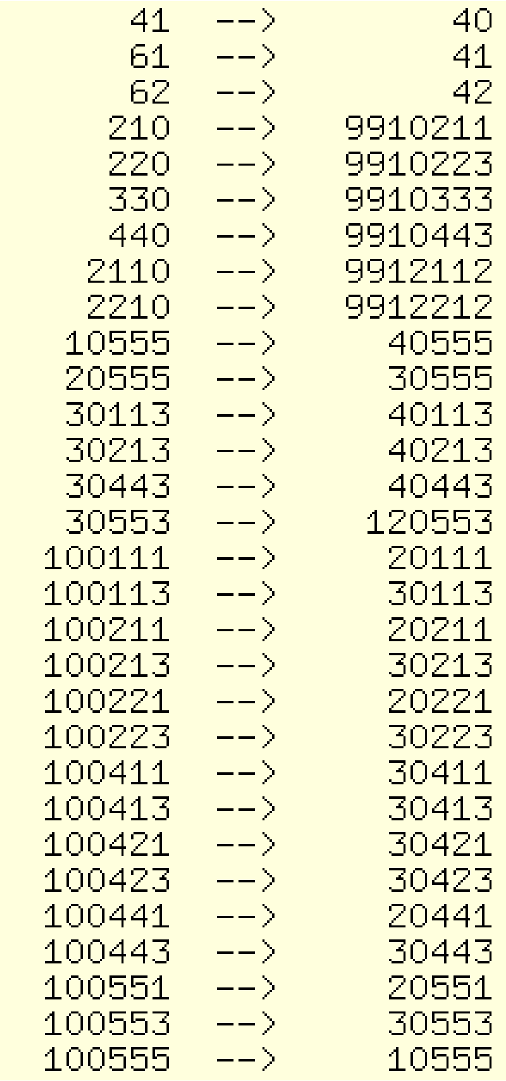
- **Solution**
  - Could all generators/EvtGen use a common particle table/service?
  - May already exist in LHCb?



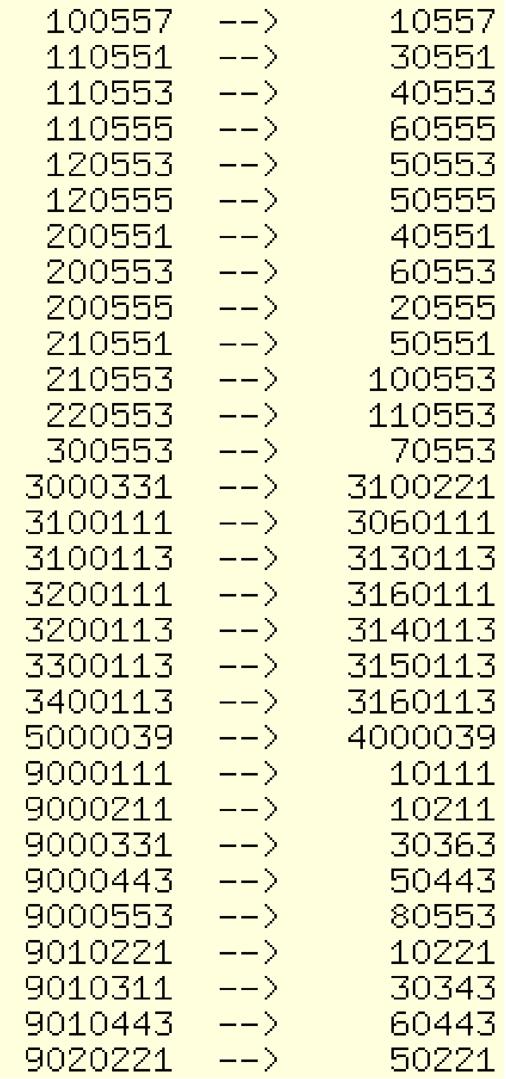


Example: Pythia -> EvtGen Particle Code Translation

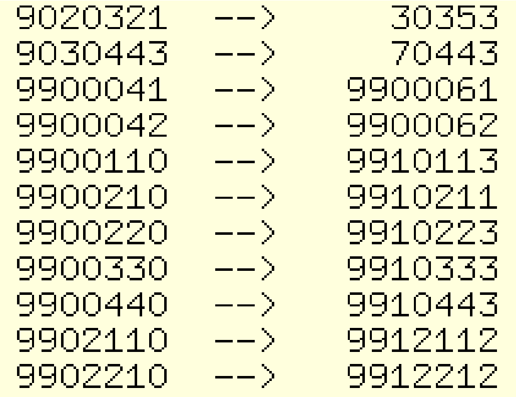
- **From HepPDT PDG code translation routines:**



```
     41   -->          40
     61   -->          41
     62   -->          42
    210   -->     9910211
    220   -->     9910223
    330   -->     9910333
    440   -->     9910443
   2110   -->     9912112
   2210   -->     9912212
  10555   -->       40555
  20555   -->       30555
  30113   -->       40113
  30213   -->       40213
  30443   -->       40443
  30553   -->      120553
 100111   -->       20111
 100113   -->       30113
 100211   -->       20211
 100213   -->       30213
 100221   -->       20221
 100223   -->       30223
 100411   -->       30411
 100413   -->       30413
 100421   -->       30421
 100423   -->       30423
 100441   -->       20441
 100443   -->       30443
 100551   -->       20551
 100553   -->       30553
 100555   -->       10555
```



```
 100557   -->       10557
 110551   -->       30551
 110553   -->       40553
 110555   -->       60555
 120553   -->       50553
 120555   -->       50555
 200551   -->       40551
 200553   -->       60553
 200555   -->       20555
 210551   -->       50551
 210553   -->      100553
 220553   -->      110553
 300553   -->       70553
3000331   -->     3100221
3100111   -->     3060111
3100113   -->     3130113
3200111   -->     3160111
3200113   -->     3140113
3300113   -->     3150113
3400113   -->     3160113
5000039   -->     4000039
9000111   -->       10111
9000211   -->       10211
9000331   -->       30363
9000443   -->       50443
9000553   -->       80553
9010221   -->       10221
9010311   -->       30343
9010443   -->       60443
9020221   -->       50221
```



```
9020321   -->       30353
9030443   -->       70443
9900041   -->     9900061
9900042   -->     9900062
9900110   -->     9910113
9900210   -->     9910211
9900220   -->     9910223
9900330   -->     9910333
9900440   -->     9910443
9902110   -->     9912112
9902210   -->     9912212
```

Miscellaneous

- **Tau decays**
  - Comparison of tau decays in TAUOLA vs EvtGen
  - Interfacing TAUOLA to EvtGen and using it for all tau decays?
- **Potential pitfalls**
  - Double-counting of processes (e.g. photon emission, B mixing)
  - How to avoid user configuration errors?

Conclusions

- **For ATLAS MC b tagging performance, effect of using EvtGen (instead of Pythia) to simulate all particle decays is small**
  - Primarily a result of different charged particle multiplicities, thus in this case would not need full EvtGen (just tweak decay table)

- **Common issues when interfacing EvtGen to other generators**
  - Use of different particle tables by EvtGen and generators leads to
    - Inconsistent use of PDG codes
    - Inconsistent particle masses

    => Could all generators/EvtGen use a common particle table/service?
  - Possible interference of Pythia settings between EvtGen and parameters desired by user when Pythia is used as primary generator in the same job

    => Bundle dedicated JETSET version with EvtGen, or include save/restore code for Pythia parameters/common blocks