WeightContainer: Towards a Unified Weights Structure

Leif Gellersen

Lund University

leif.gellersen@thep.lu.se

Pythia meeting, $\mathfrak{Y}$[Lund]
April 21st, 2020

Introduction: Weights in Pythia8

- Pythia 8 baseline: unweighted events, each event represents equal fraction of total cross section
- Some cases: event comes with weight(s), need to be applied when filling histograms
- Examples:
  - Les Houches events can come with weights (+-1, relative or even in pb, variations)
  - Multi-jet merging (CKKWL, UMEPS, NLO)
  - User hooks weights, for biased phase space sampling or enhanced emissions
  - Automated variation of shower parameters
  - Heavy ion collision: impact parameter sampling

Accessing Weights

Info::weight() Nominal weight, includes Les Houches weight, heavy ion weight, biased selection weight, and by default CKKWL/UMEPS merging weights.

Info::mergingWeight() LO merging weight (1. if included in Info::weight())

Info::mergingWeightNLO() NLO merging weight, not included in Info::weight()

UserHooks::biasedSelectionWeight() Weight to compensate modified phase space sampling

UserHooks::getEnhancedEventWeight() Enhanced emissions weight

Info::nVariationGroups(); Info::getGroupName(int iG); Info::getGroupWeight(int iG) Groups of parameter variations in shower

Info::nWeights(); Info::weightLabel(int i); Info::weight(int i) Individual parameter variations in shower

and more...

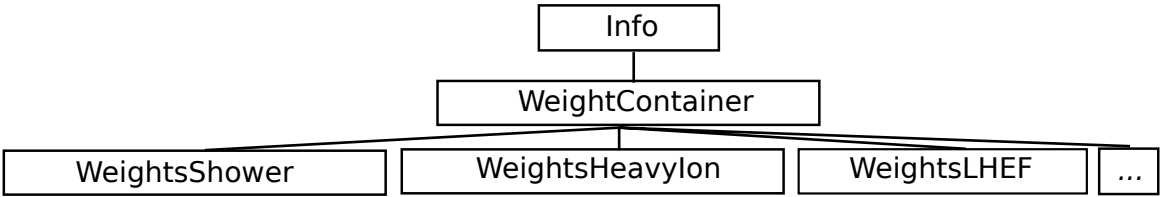
Challenges

- More weights: LHEF variation weights, merging variation weights
- Many different data structures
- Many different setters & getters for specific weights
- Combination and manipulation prone to errors

Proposed by Stefan: Use WeightContainer class to collect and store all weight information

Goal: provide user with one vector of weights and weightnames

New Data Structures



- One weight class per purpose: all derive from WeightsBase class
  - Contains vector of weight names and weight values
  - Provides getters and setters
  - Method *collectWeightNames()* and *collectWeightValues()* called from container to collect weights. For now, just returns values, but can be used to combine variation groups
- WeightContainer contains all individual weight classes. Can combine weights in meaningful ways before providing them to user. For now: just collects.

New User Interface

WeightContainer (and Info) provide weights to user:

int numberOfWeights() Total number of weights provided to user
double weightValueByIndex(int key) Individual weight value
string weightNameByIndex(int key) Individual weight name
vector⟨double⟩ weightValueVector() Vector of weight values
vector⟨string⟩ weightNameVector() Vector of weight names

These vectors are used for weight output to HepMC files

8.302 Status

- WeightContainer has been implemented
- Old data structures remain in place for now
- Weight classes so far: WeightsShower, WeightsHeavyIon, WeightsLHEF
- New structure available, and used for HepMC output, see main44.cc and main45.cc

Next Steps

Created branch *pythia83lg-multiweights-mergingvariations* to continue work

- Added weightsMerging
- Moved shower weight structures from Info to weightsShower (getters still in place)
- . . . now implementing merging variations in 8.3. . .
- . . . then combine LHEF variations, shower variations and merging variations in WeightContainer

To be discussed:

- Which weights and combinations should be given to user?
- Name conventions (e.g. user-defined names for weight groups, LHEF conventions MUR2 for combined LHEF-shower-merging weights?)
- Which weights to provide with AUX_ prefix, so available if needed, but ignored by Rivet?
- For scalar weights (HI, biased selection, enhanced emissions): return individually, or combine to nominal weight, or both?