Tuning of Monte Carlo Event Generators

Leif Gellersen

PhD Student at Lund University

leif.gellersen@thep.lu.se

October 17th, 2018

Introduction

Theory

QFT: Lagrangian formulation of physics

- Standard Model: $\mathcal{L}_{\mathrm{SM}}$
- Beyond the Standard Model: $\mathcal{L}_{\mathrm{BSM}}$

Experiments

Collider experiments with complex detectors

- LHC with ATLAS, CMS, ...
- Reconstruction of individual events

Simulation

Linking theory & experiment

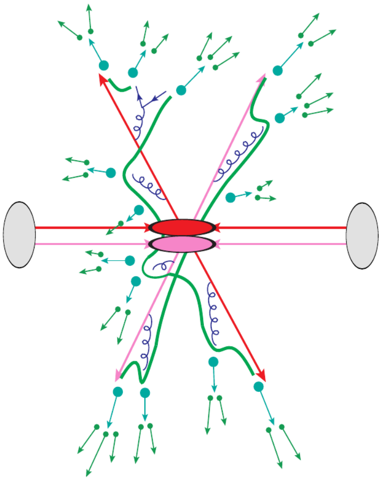
- MC generators: Stochastic simulation of events
- Allow to compare theory and experiment

MC Event Generators: An Overview

Predict fully exclusive final state

- Perturbative methods well known
  - Matrix elements (LO/NLO)
  - Parton shower in initial and final state
- Non-perturbative models
  - Multiple interactions
  - Hadronization
  - Hadron decays

Models include many parameters

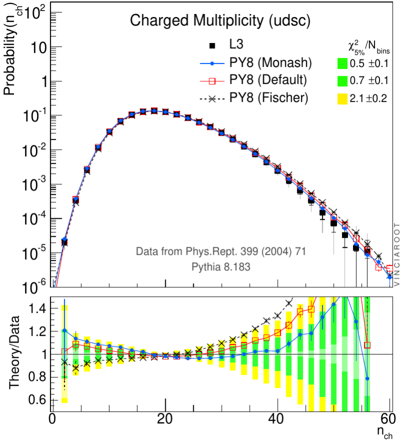


Borrowed from S. Prestel

Tuning

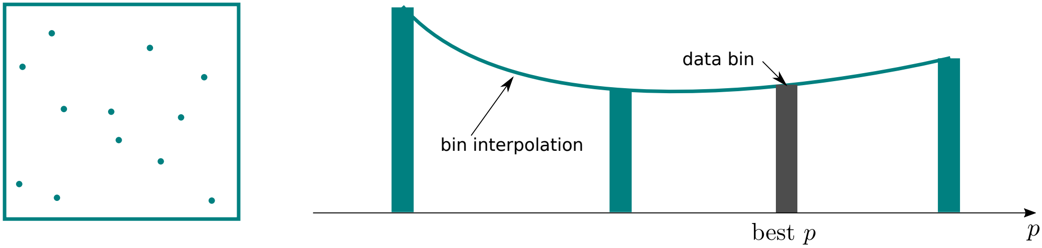
- Optimize parameters based on well-measured data
- Factorize as much as possible (assuming universality)

FSR	e^+e^- data: LEP event shapes
Hadronization	Many parameters, model dependent. Use LEP identified particle spectra
ISR and UE	Use hadron collider data



arXiv:1404.5630, P. Skands et al., 2014

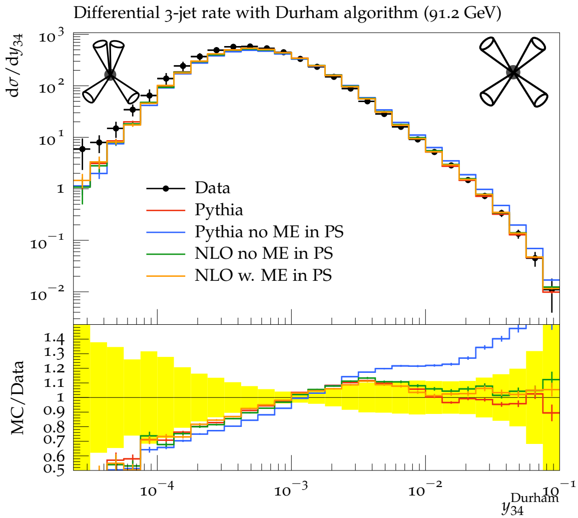
How to Tune



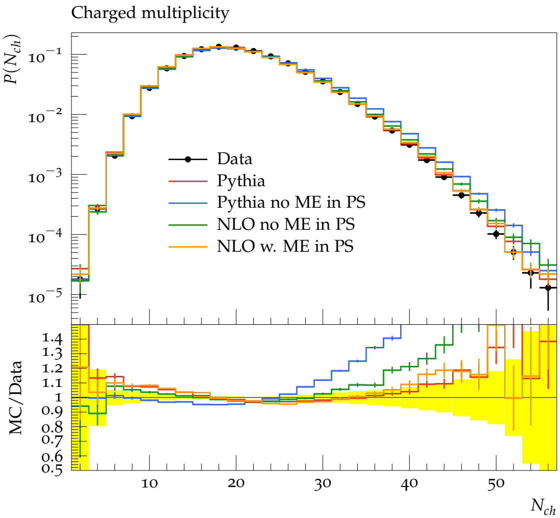
- Generate MC pseudodata f_b, compare to measured data bin $\mathcal{R}_b$ with uncertainty Δ_b
- Minimize $\chi^2(\vec{p}) = \sum_b w_b \frac{(f^{(b)}(\vec{p}) - \mathcal{R}_b)^2}{\Delta_b^2}$
- Iterative MC event generation slow $\rightarrow$ Use bin-wise parametrization of MC generator response
- Professor: Python package for MC tuning, highly automated, includes validation tools
  arXiv:0907.2973, A. Buckley et al., 2009

Pythia Matching & Merging Tune work with Stefan Prestel & Malin Sjödahl

- Use of NLO pQCD calculations more and more common
  NLO Matching Use NLO matrix elements, higher fixed order accuracy
  Multi-jet Merging Use higher multiplicity matrix elements to improve event shapes
- Problem: Default tune based on LO with ME corrections for first emission
- Not too bad for some matched & merged calculations
- More precise perturbative calculation $\rightarrow$ less freedom to tune models
- Retuning might allow for some improvements, more universal tune



Data from Jade & Opal Collaborations, 2000
arXiv:hep-ex/0001055



Data from L3 Collaboration, 2004
arXiv:hep-ex/0406049

AutoTunes work with Johannes Bellm

Problem

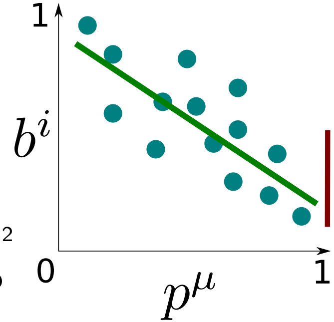
- Polynomial interpolation only possible for $\lesssim 10$ parameters
- Interpolation only good if ranges small enough
- χ^2 depends on weights $\rightarrow$ physicist dependent

Goal

- Framework to reduce human interaction & make tune reproducible
- Tune many parameters at once: automatically divide into sub-tunes
- Set weights for observables automatically
- Allow for iterations with revised parameter ranges

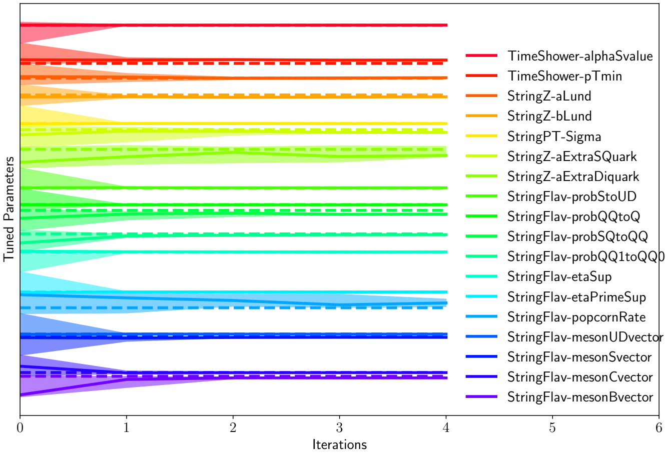
Autotunes: The Idea

- Normalize each bin b_i and each parameter p^μ to $[0, 1]$
- Find slopes S_i^μ
- $\vec{S}_i$ vector in parameter space
- $\vec{S}_i$ points along parameters of high influence on bin
- Normalize: $N_i^\mu = \frac{S_i^\mu}{\sum_i S_i^\mu}$
- Find $\vec{J} = (1, 0, 0, 1, 0, \ldots, 1)$ that maximizes $w = \sum_i (\vec{N}_i \cdot \vec{J})^2$
  $\rightarrow$ "Most correlated" subset of parameters: tune in one step
- Use weights $w_i = \vec{N}_i \cdot \vec{J}$, emphasizes relevant data bins



Iterative Pythia Tuning against Pythia Pseudodata

Try to reproduce - - - values, ≈ 6000 DOF & 18 parameters



Summary

- Tuning essential to convert measured physics to good event generator predictions
- Matching & merging requires tune validation and retuning of soft physics models
- Working on AutoTunes: Framework for more automated tuning with many parameters