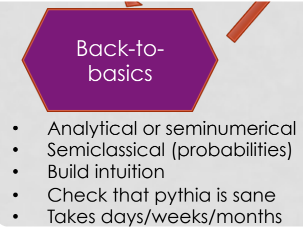
Jet Reclustering

CERN Pileup Workshop

Ben Nachman, Pascal Nef, Ariel Schwartzman, **Maximilian Swiatlowski**

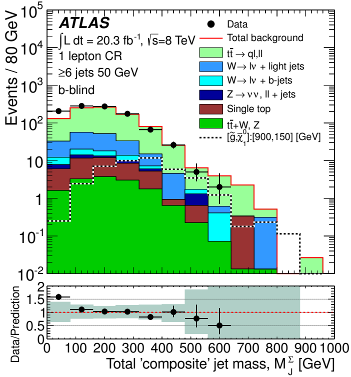
17 May, 2014

- Theorists's theme for BOOST2013 and beyond: "Back to Basics"
- Experimentalists shouldn't be left out!
- One basic question of jet performance: what algorithm and R should we use?
  - → Answer should be decided by the *physics*
  - But in practice this can be limited: calibrations, uncertainties, resolution hold us back
  - How do we make this modular and configurable for analyzers?



One Solution: Jet Reclustering

- ATLAS (and now CMS!) have one existing, exquisitely understood jet collection: $r = 0.4$
- Can use small-r jets as input to large-R algorithm
  - Already used by ATLAS in several analyses, mostly for technical reasons (lack of information in datasets)
- Immediate benefit: substructure becomes **accessible** and **flexible**
  - In this talk we perform a systematic study and optimization of this approach to gain a deeper understanding
  - What do we lose (or gain?) when using small jets as inputs?



arXiv:1308.1841

Jet Reclustering Details

1. Start with small-r jets
   - These are calibrated and have small uncertainties
   - **NB**: These have a minimum p_T (what the experiments can calibrate)
   - Can optionally apply a track-based cut to remove pile-up
2. Run a large-R ($r \ll R$) jet algorithm with small-r jets as input (as in ATLAS multijet)
3. New idea: use **reclustered trimming** (RT) to improve mass resolution
   - Remove 'subjets' with $p_T^{sub}/p_T^{total} < f_{cut}$

→ In general: need small-r, small algorithm, large-R, large algorithm
   - Small algorithms would be selected by performance group: large algorithm by the analysis

Jet Reclustering Details

1. Start with small-r jets
   - These are calibrated and have small uncertainties
   - **NB**: These have a minimum p_T (what the experiments can calibrate)
   - Can optionally apply a track-based cut to remove pile-up
2. Run a large-R ($r \ll R$) jet algorithm with small-r jets as input (as in ATLAS multijet)
3. New idea: use **reclustered trimming** (RT) to improve mass resolution
   - Remove 'subjets' with $p_T^{sub}/p_T^{total} < f_{cut}$

→ In general: need small-r, small algorithm, large-R, large algorithm

- Small algorithms would be selected by performance group: large algorithm by the analysis

Jet Reclustering Details

1. Start with small-r jets
   - These are calibrated and have small uncertainties
   - **NB**: These have a minimum p_T (what the experiments can calibrate)
   - Can optionally apply a track-based cut to remove pile-up
2. Run a large-R ($r \ll R$) jet algorithm with small-r jets as input (as in ATLAS multijet)
3. New idea: use **reclustered trimming** (RT) to improve mass resolution
   - Remove 'subjets' with $p_T^{sub}/p_T^{total} < f_{cut}$

→ In general: need small-r, small algorithm, large-R, large algorithm
   - Small algorithms would be selected by performance group: large algorithm by the analysis

Jet Reclustering Details

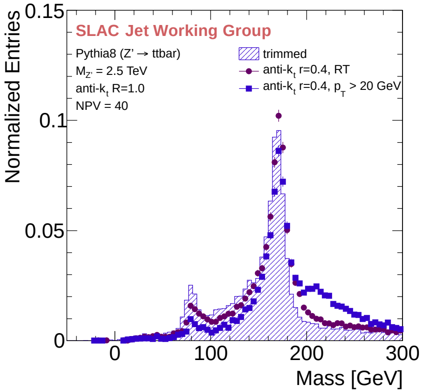
1. Start with small-r jets
   - These are calibrated and have small uncertainties
   - **NB**: These have a minimum p_T (what the experiments can calibrate)
   - Can optionally apply a track-based cut to remove pile-up
2. Run a large-R ($r \ll R$) jet algorithm with small-r jets as input (as in ATLAS multijet)
3. New idea: use **reclustered trimming** (RT) to improve mass resolution
   - Remove 'subjets' with $p_T^{sub}/p_T^{total} < f_{cut}$

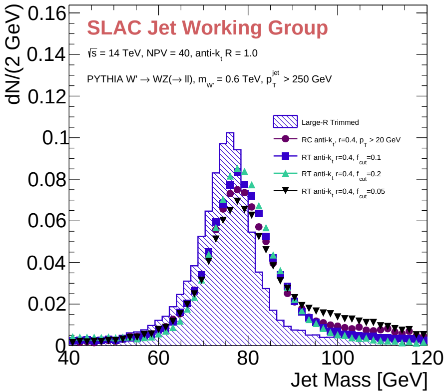
$\rightarrow$ In general: need small-r, small algorithm, large-R, large algorithm

- Small algorithms would be selected by performance group: large algorithm by the analysis

Fixed Cut vs Trimming

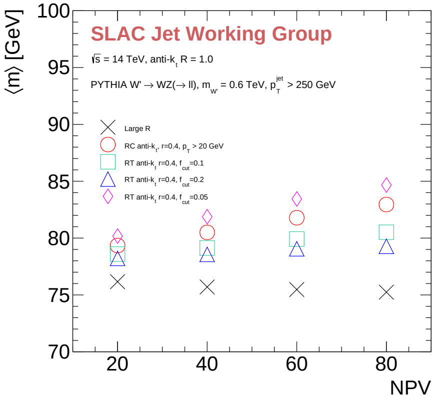
SLAC

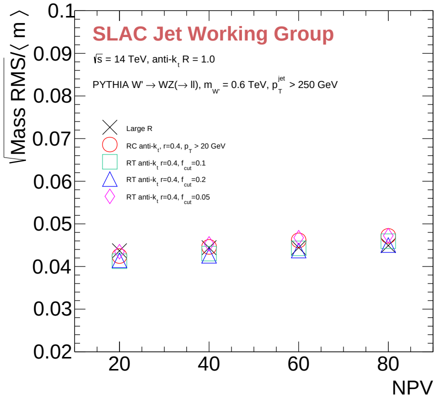




- Reclustered jets, with fixed p_T cut or $f_{cut} = 10\%$ comparable to trimmed jets – depends on jet p_T!

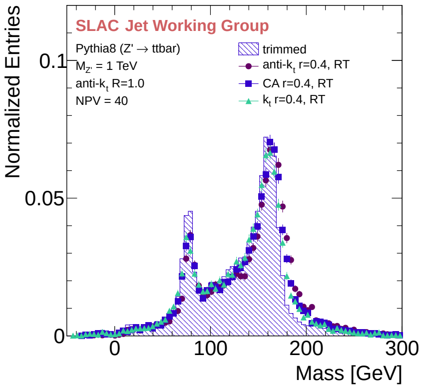
Fixed Cut vs Trimming: Bias and Dispersion

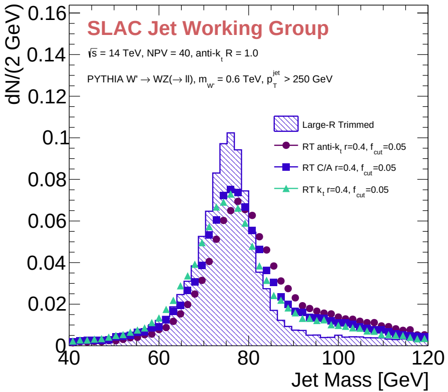




- Reclustered Trimming provides low bias, no change in dispersion for large range of NPV

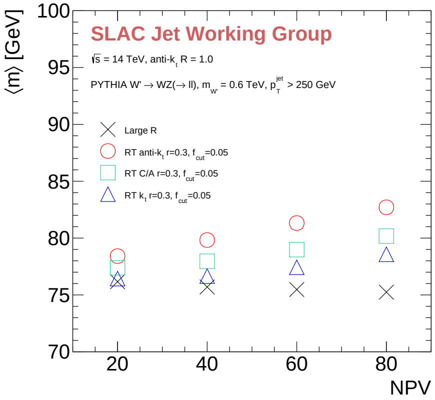
Comparing Jet Algorithm

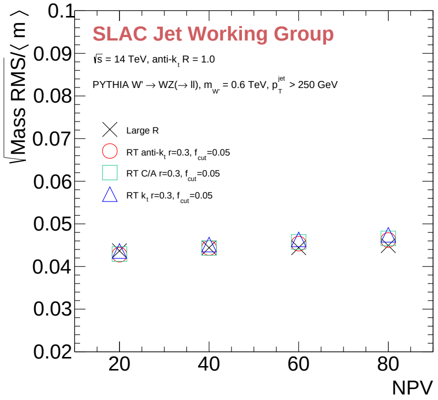




- No large sensitivity to input jet algorithm: anti-k_T performs well

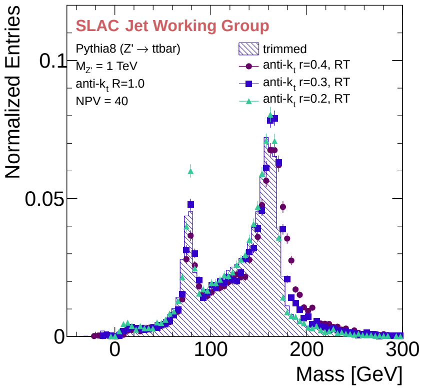
Jet Algorithms: Bias and Dispersion

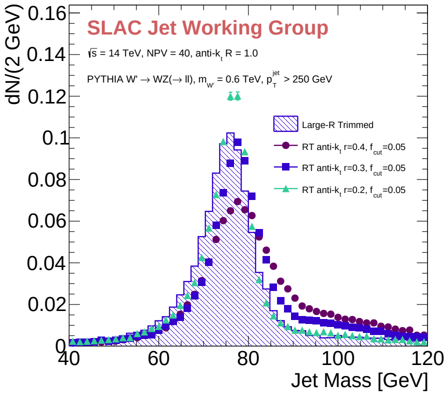




- anti-k_T shows the most bias, but still very low, and no change in dispersion

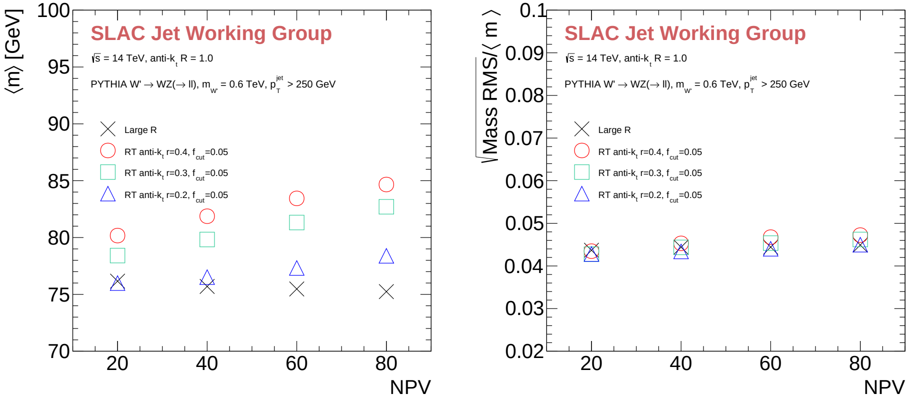
Comparing Jet Radius





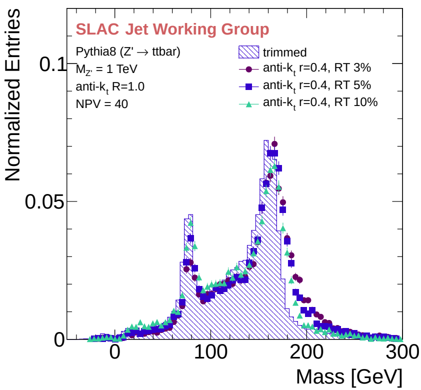
- Also no large dependence on input jet r: smaller is preferred, especially in W (lower p_T)

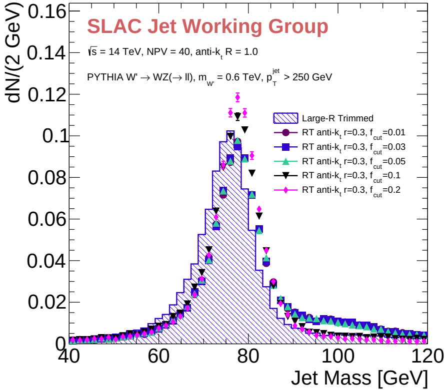
Jet Radius: Bias and Dispersion



- Smaller *r* means tighter cut, lower bias– dispersion remains constant

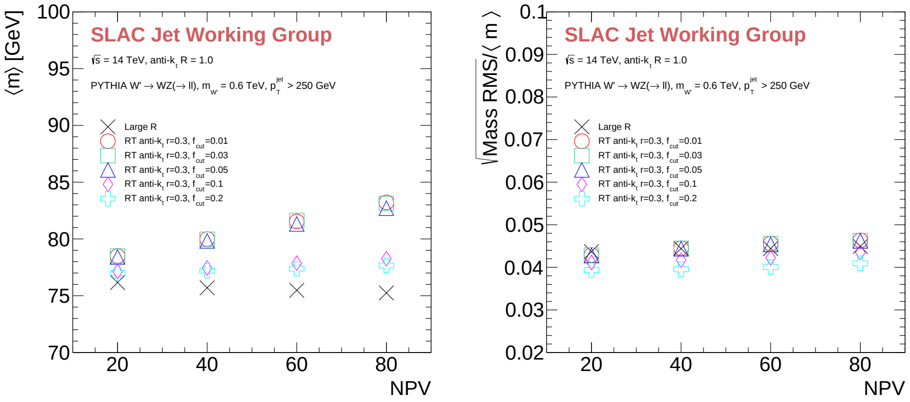
Comparing f_{cut}





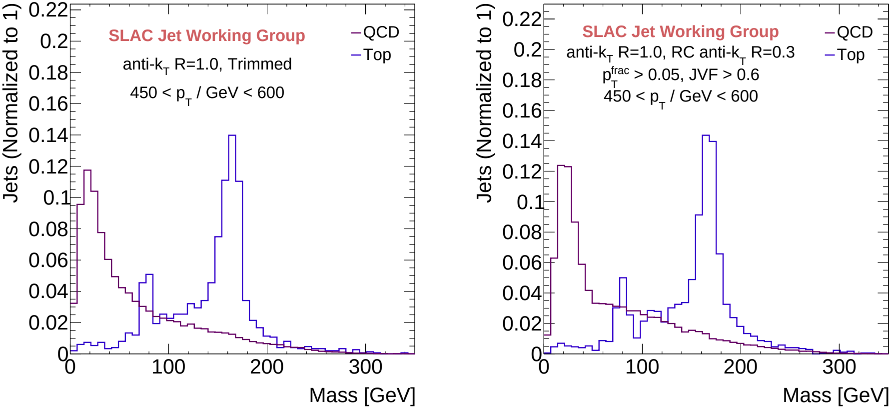
- Tighter f_{cut} increases sharpness of peak– and can easily be tuned by each analysis to suit their needs

f_{cut}: Bias and Dispersion



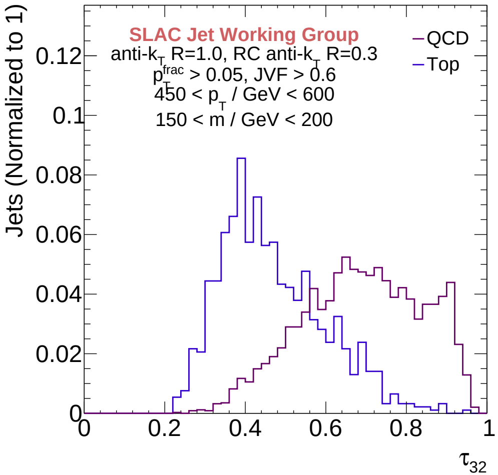
- Harder cut can remove bias and lower dispersion as well
- Not a very hard cut: $O(20-40)$ GeV

Comparing to QCD



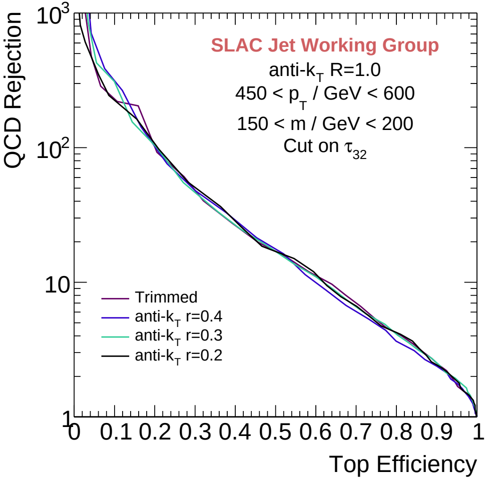
- Doesn't work just on signal jets– also works on QCD just like trimming

N-Subjettiness



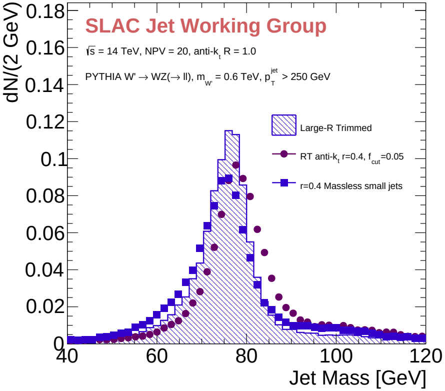
- Jet shapes are of some concern, but strong rejection of QCD remains
- NB: need additional pileup correction for shapes

N-Subjettiness Performance



- Efficiency/rejection is identical to performance on trimmed jets

A Small Caveat: Input Masses



- Need **mass calibration** of small jets to have best performance– significant portion of mass comes from inputs

Conclusions

- Jet Reclustering is a (very) simple algorithm for creating large-R jets
- Allows experimentalists to focus calibrations on one jet algorithm
  - Uncertainties and resolutions propogate from the low level to the high level jets
  - Pileup corrections applied to this collection stabilize the large-R jets
- With the addition of trimming, performance can be **essentially identical** to normal trimmed jets
  - Needs to be experimentally validated! Which low level algorithm to use needs to be decided.
- We can be more flexible when designing analyses– tuning algorithms, using multiple interpretations, and more
  - The basic dream of jets is to always use the right tool for the right job: Reclustered Trimming can help

Thank you for your attention!

Appendix

Comparing corrJVF

- Does using a corrJVF cut matter?
- Turns out not so much– the f_{cut} removes pileup entirely by itself
    - Similar result to corrJVF based trimming from Pascal
- Provides direct way to implement track-based pileup corrections– will be more important in the future