Constituent Subtraction Updates

Peter Berta[1], Lucia Masetti[1], David Miller[2], Martin Spousta[3]

[1] JGU Mainz
[2] University of Chicago
[3] Charles University in Prague

July 17, 2018

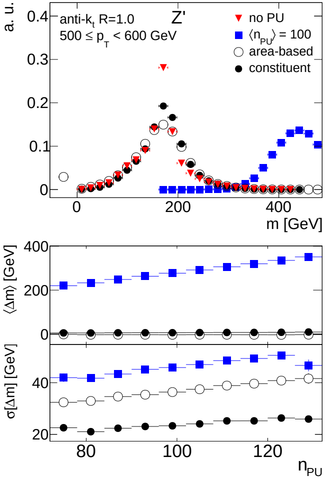
Outline

1. Reminder of the Constituent Subtraction
2. Usage of Constituent Subtraction in experiments
3. New developments

Reminder of the Constituent Subtraction (CS)

Constituent Subtraction (CS)

- arXiv:1403.3108
- Pileup subtraction method at the level of inputs
- Generalization of the Area Subtraction (arXiv:0802.1188)
  - exploits the background p_{T} density (ρ)
- Two possibilities of usage:
  - **jet-by-jet** - first jet clustering, then correction of individual jets
  - **whole event** - first correction of the whole event, then jet clustering
    - jet clustering is less biased
- Package inside fastjet-contrib



Better performance wrt Area Subtraction

Background density ρ

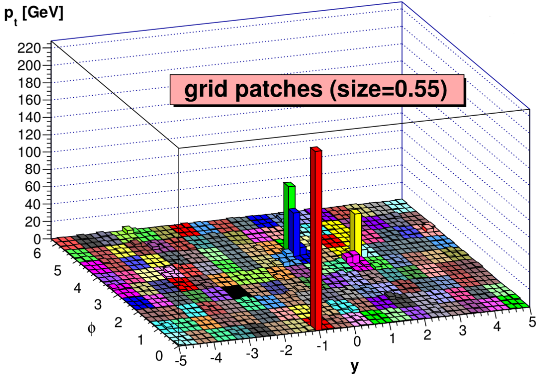
- ρ - amount of p_{T} from pileup particles per unit area in the rapidity - azimuth $(y-\phi)$ space
- many possibilities how to estimate ρ. One of them:

1. event divided into rectangular patches in the $(y-\phi)$ space; p_{T} of each patch:

$$p_{\mathrm{Tpatch}} = \sum_{i \in \mathrm{patch}} p_{\mathrm{T}i}$$

2. the estimated pileup p_{T} density:

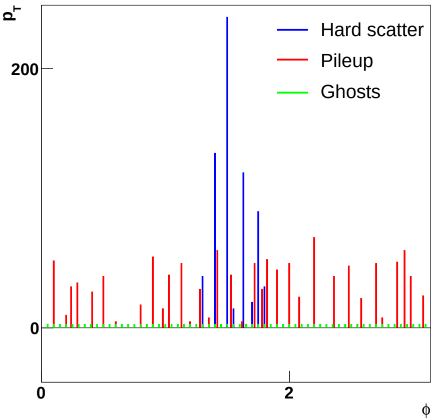
$$\rho = \mathrm{median}_{\mathrm{patches}} \left\{ \frac{p_{\mathrm{Tpatch}}}{A_{\mathrm{patch}}} \right\}$$



Slides from G. Soyez (BOOST2012 conference)

CS jet-by-jet

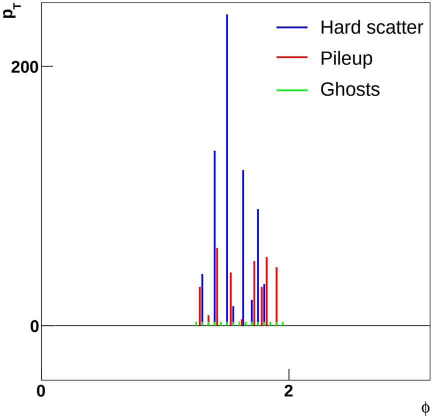
1. **Adding ghosts to the whole event**



Whole event

CS jet-by-jet

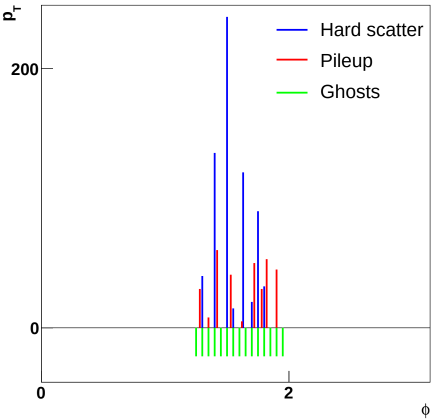
1. Adding ghosts to the whole event
2. **Jet clustering**



Leading jet before correction

CS jet-by-jet

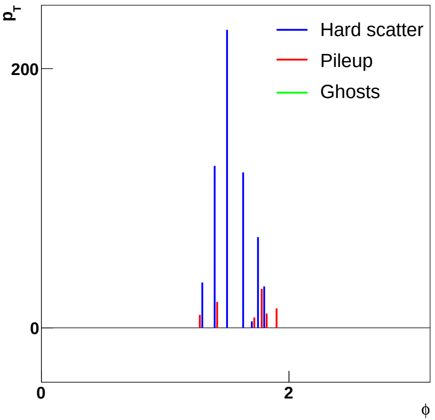
1. Adding ghosts to the whole event
2. Jet clustering
3. **Setting ghosts p_{T} to negative value corresponding to ρ**



Leading jet before correction

CS jet-by-jet

1. Adding ghosts to the whole event
2. Jet clustering
3. Setting ghosts p_{T} to negative value corresponding to ρ
4. **Matching of ghosts to particles**



Leading jet after correction

CS jet-by-jet

1. Adding ghosts to the whole event
2. Jet clustering
3. Setting ghosts p_{T} to negative value corresponding to ρ
4. **Matching of ghosts to particles**

- Evaluate distances between each particle-ghost pair.
  - Distance between particle i and ghost k:

$$\Delta R_{i,k} = p_{\mathrm{T}i}^{\alpha} \cdot \sqrt{(y_i - y_k^g)^2 + (\phi_i - \phi_k^g)^2}$$

- Combine each ghost-particle pair starting from lowest $\Delta R_{i,k}$:

If $p_{\mathrm{T}i} \geq p_{\mathrm{T}k}^g$: $\quad p_{\mathrm{T}i} \to p_{\mathrm{T}i} - p_{\mathrm{T}k}^g$, $\quad p_{\mathrm{T}k}^g \to 0$

otherwise: $\quad p_{\mathrm{T}i} \to 0$, $\quad p_{\mathrm{T}k}^g \to p_{\mathrm{T}k}^g - p_{\mathrm{T}i}$

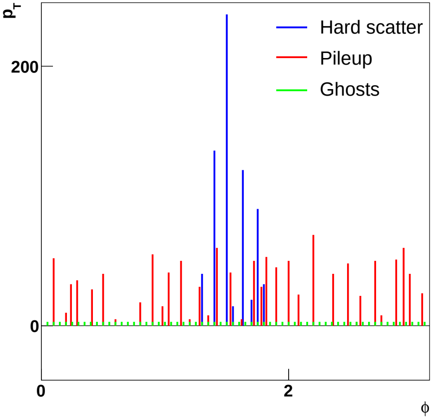
- Procedure stops for $\Delta R_{i,k} > \Delta R^{\max}$

CS jet-by-jet

1. Adding ghosts to the whole event
2. Jet clustering
3. Setting ghosts p_{T} to negative value corresponding to ρ
4. **Matching of ghosts to particles**
   - Free parameters: α and $\Delta R^{\max}$
     - small effect for jet-by-jet CS

CS whole event

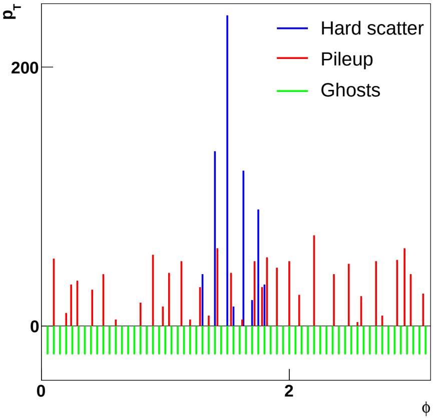
1. **Adding ghosts to the whole event**



Whole event

CS whole event

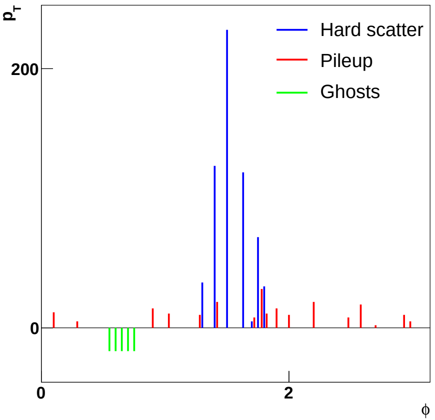
1. Adding ghosts to the whole event
2. **Setting ghosts p_{T} to negative value corresponding to ρ** (no jet clustering)



Whole event before correction

CS whole event

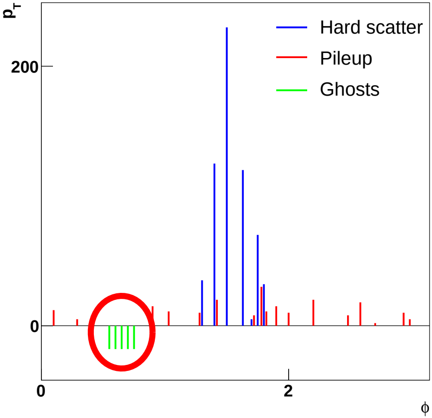
1. Adding ghosts to the whole event
2. Setting ghosts p_{T} to negative value corresponding to ρ
3. **Matching of ghosts to particles**
    - same algorithm as for jet-by-jet correction



Whole event after correction

CS whole event

1. Adding ghosts to the whole event
2. Setting ghosts p_{T} to negative value corresponding to ρ
3. Matching of ghosts to particles
   - same algorithm as for jet-by-jet correction
   - **with finite $\Delta R^{\max}$, some ghosts can be unmatched**
     - discussed later



Whole event after correction

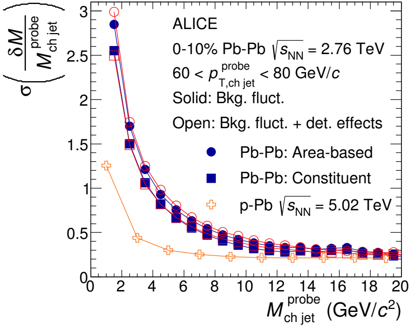
Usage of CS in experiments

Usage of CS in experiments

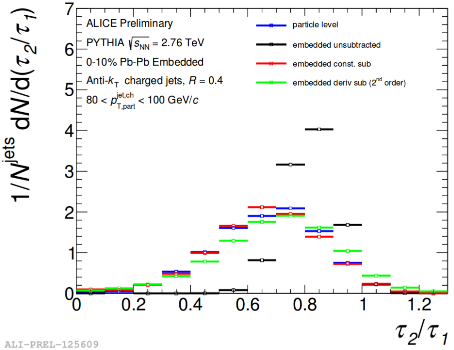
- Several performance studies with CS:
  - ALICE (heavy-ions)
  - ATLAS (proton-proton)
  - CMS (heavy-ions)
  - CMS (proton-proton)
- **Physics results** with CS:
  - ALICE (heavy-ions)
  - CMS (heavy-ions)

CS performance in ALICE (heavy-ions)

- Compared with Area Subtraction in arXiv:1702.00804
- Compared with Shape-expansion method in arXiv:1705.03383
- Jet-by-jet CS
- Observed better performance with CS



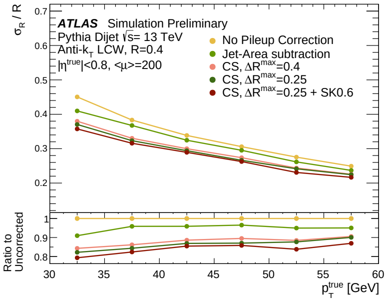
mass resolution
arXiv:1702.00804



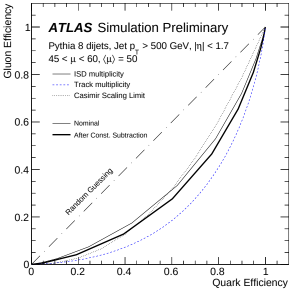
τ_{21} distribution
arXiv:1705.03383

CS performance in ATLAS (proton-proton)

- Low p_{T} performance in ATLAS-CONF-2017-065
- Large-R jets performance in ATL-PHYS-PUB-2017-020
- Jet constituent multiplicity ATL-PHYS-PUB-2018-011
- Improved performance compared to the Area Subtraction
- Combination with SoftKiller (SK) (arXiv:1407.0408) leads to the best performance



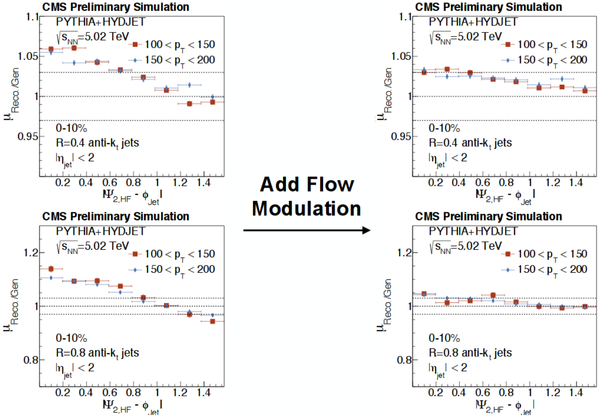
ATLAS-CONF-2017-065



ATL-PHYS-PUB-2018-011

CS performance in CMS (heavy-ions)

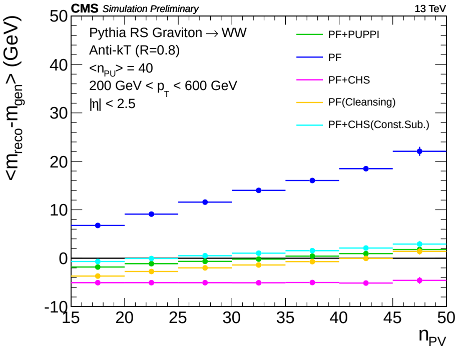
- CMS-DP-2018-024
- Jet-by-jet CS
- Inherent methods for estimation of $y - \phi$ dependence of ρ:
  - y dependence - Iterative Pedestal method, EPJC 50, (2007) 117–123
  - ϕ dependence - Flow Modulation, CMS-HIN-16-019



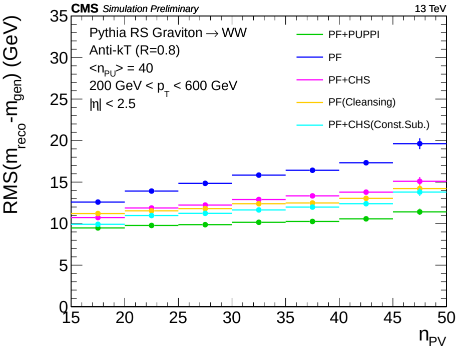
JES closure with CS

CS performance in CMS (proton-proton)

- CMS-PAS-JME-14-001
- Jet-by-jet CS
- The used CS improves jet mass, but PUPPI performs better



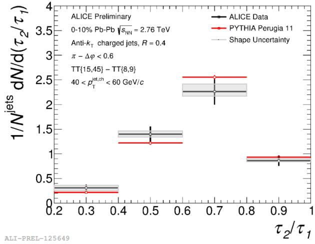
mass response



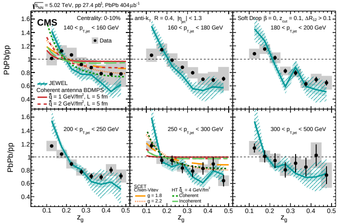
mass resolution

Usage of CS for physics results

- ALICE publications:
  - Role of Coherence Effects on Jet Quenching in Pb-Pb, arXiv:1705.03383
  - Exploring jet substructure with jet shapes in ALICE, arXiv:1704.05230
  - First measurement of jet mass in Pb-Pb and p-Pb collisions, arXiv:1702.00804
  - Jet shapes in pp and Pb–Pb collisions at ALICE, arXiv:1512.07882
- CMS publications:
  - Splitting function in pp and PbPb collisions, arXiv:1708.09429
  - Measurement of the groomed jet mass in PbPb and pp collisions, arXiv:1805.05145



τ_{21}, arXiv:1705.03383



Splitting functions, arXiv:1708.09429

New developments

New developments

- ρ rescaling
- Studies on whole event CS
- Iterative CS

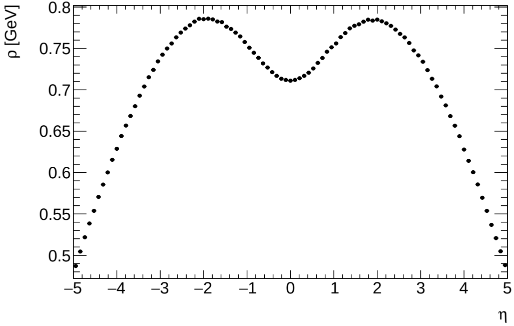
New developments - setup for performance studies

- Pythia8 simulation of signal and pileup events
- Number of pileup events, N_{PU}, is uniformly distributed in range [0,120] (LHC Run 3)
- Particles grouped into massless towers of size 0.1×0.1 in $y - \phi$ space
- CS correction of whole event up to $|\eta| < 5$
- Using CS parameter $\alpha = 0$
- Using ρ rescaling (rapidity dependence)
- Figures of merit:
  - $\text{Bias} = \frac{\langle x - x^{\text{true}} \rangle}{\langle x^{\text{true}} \rangle}$ **- the closer to zero, the better**
  - $\text{Resolution} = \frac{\text{RMS}(x - x^{\text{true}})}{\langle x^{\text{true}} \rangle}$ **- the smaller, the better**

New developments
ρ rescaling

ρ rescaling

- The estimated ρ is by default constant in $y - \phi$ space
- But ρ can depend on y and ϕ:

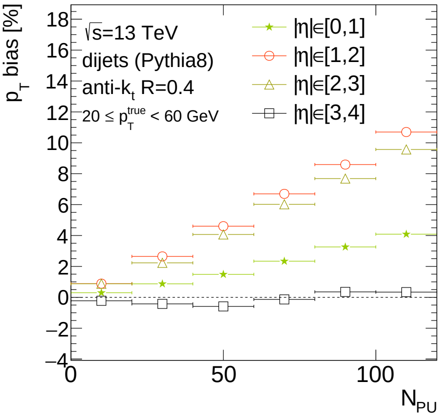


Rapidity dependence from Pythia (massless inputs)

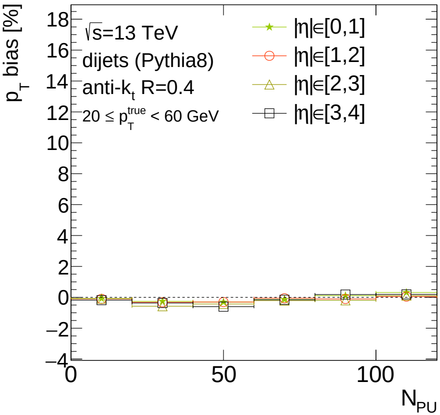
ρ rescaling

- Important for CS:
  - more precise estimation of ρ
  - the ghosts p_{T} are scaled according to the $y - \phi$ dependence
- Done by fastjet's background estimation classes
- Limited number of rescaling classes:
  - rapidity parametrized as polynom
- New rescaling classes within CS:
  - rapidity in 1D histogram
  - rapidity in 1D histogram and azimuth parametrized with elliptic flow parameters,...
- For usage, see example:
  https://fastjet.hepforge.org/trac/browser/contrib/contribs/ConstituentSubtractor/tags/1.3.1/example_background_rescaling.cc

ρ rescaling - demonstration



no rescaling (constant ρ)

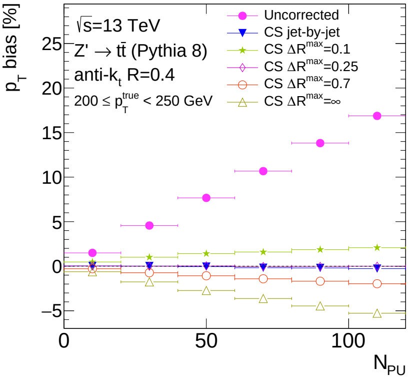


with rescaling

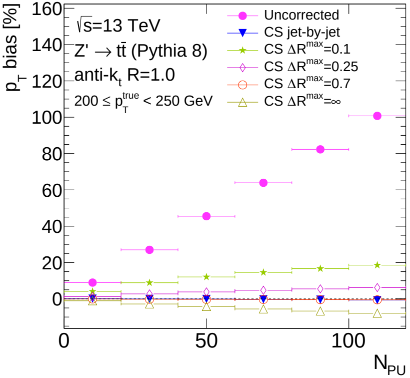
- ρ rescaling using rapidity in 1D histogram (CS with $\Delta R^{\max} = 0.25$)
- Dependence on jets η is removed

New developments
Studies on whole event CS

Whole event CS - p_{T} bias



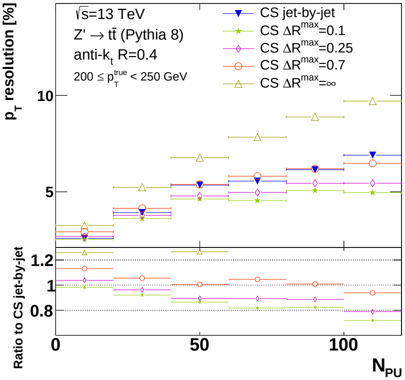
anti-k_t $R = 0.4$



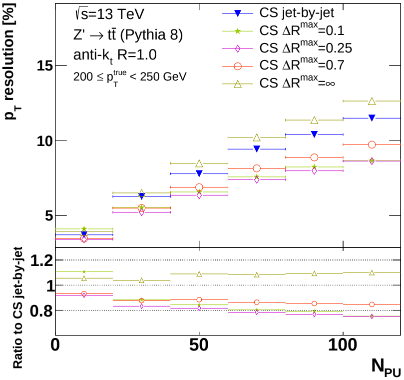
anti-k_t $R = 1.0$

- Too large values of $\Delta R^{\max}$ lead to overcorrection of hard-scatter jets
  - The hard-scatter jets act as magnets for ghosts

Whole event CS - p_T resolution



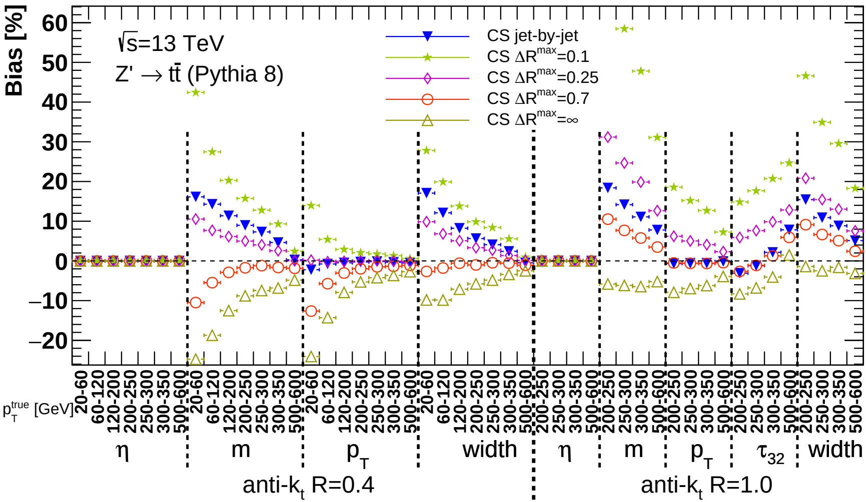
anti-k_t $R = 0.4$



anti-k_t $R = 1.0$

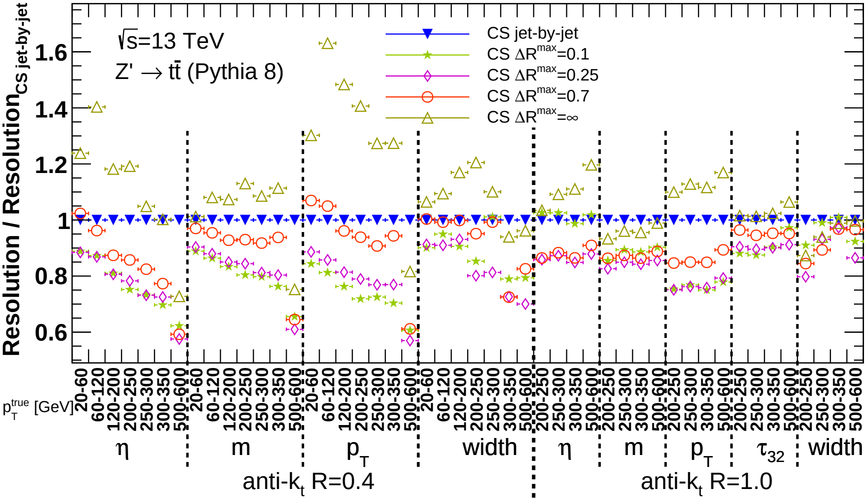
- Best resolution for low $\Delta R^{\max}$ (e.g. 0.25) - but bias is larger for very low values

Whole event CS - bias summary, $N_{\mathrm{PU}} \in [100, 120]$



- To get minimal bias, $\Delta R^{\max} \approx 0.25$ is preferred for small-R jets, while $\Delta R^{\max} \approx 0.7$ is preferred for large-R jets

Whole event CS - resolution summary, $N_{\mathrm{PU}} \in [100, 120]$



- $\Delta R^{\max} \approx 0.25$ provides the best resolution in most cases

Whole event CS - summary

- Small $\Delta R^{\max}$ yields to much better resolution than jet-by-jet CS
  - But it may cause a larger bias
- It is hard to find $\Delta R^{\max}$ parameter which is optimal for each jet definition, jet observable, p_{T} bin.

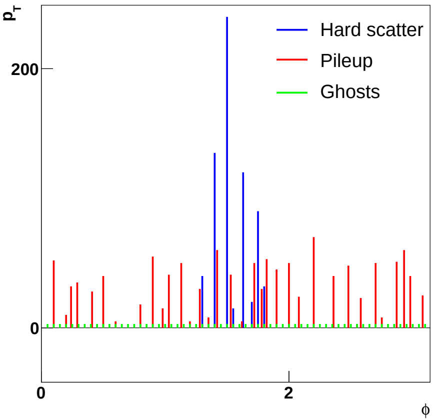
- Improvement to the above problem: **Iterative CS**

New developments
Iterative CS

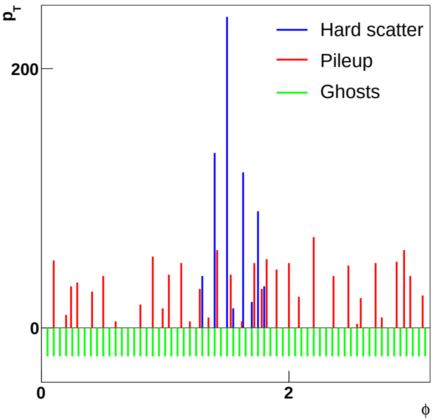
Iterative CS (ICS)

- Application of the whole event CS several times
- After each CS application, the remaining unsubtracted p_{T} is redistributed
- For each iteration, different CS parameters can be used
- Available in fastjet-contrib since ConstituentSubtractor v1.3.0
- For usage, see example:
  https://fastjet.hepforge.org/trac/browser/contrib/contribs/ConstituentSubtractor/tags/1.3.1/example_whole_event_iterative.cc

1. **Adding ghosts to the whole event**

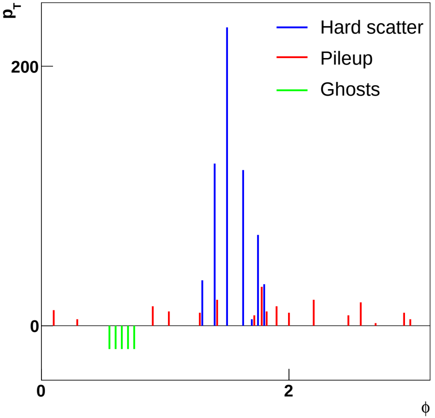


1. Adding ghosts to the whole event
2. **Setting ghosts p_{T} to negative value corresponding to ρ**



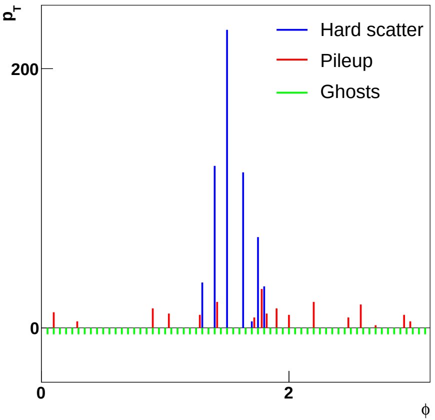
Iterative CS - example with two iterations

1. Adding ghosts to the whole event
2. Setting ghosts p_{T} to negative value corresponding to ρ
3. **1. iteration: matching of ghosts to particles**



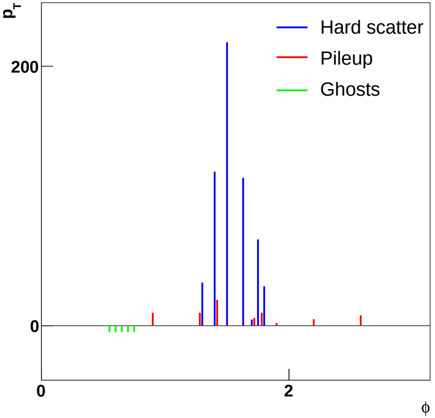
Iterative CS - example with two iterations

1. Adding ghosts to the whole event
2. Setting ghosts p_{T} to negative value corresponding to ρ
3. 1. iteration: matching of ghosts to particles
4. **Redistribution of remaining p_{T}**
    1. evaluate the scalar sum of p_{T} of remaining ghosts
    2. uniformly redistribute among ghosts the p_{T} sum from previous step

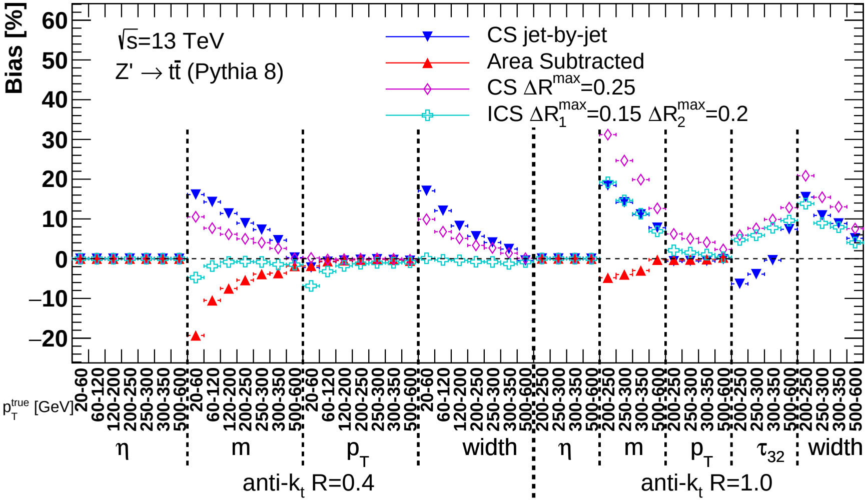


Iterative CS - example with two iterations

1. Adding ghosts to the whole event
2. Setting ghosts p_T to negative value corresponding to ρ
3. 1. iteration: matching of ghosts to particles
4. Redistribution of remaining p_T
    1. evaluate the scalar sum of p_T of remaining ghosts
    2. uniformly redistribute among ghosts the p_T sum from previous step
5. **2. iteration: matching of ghosts to particles**

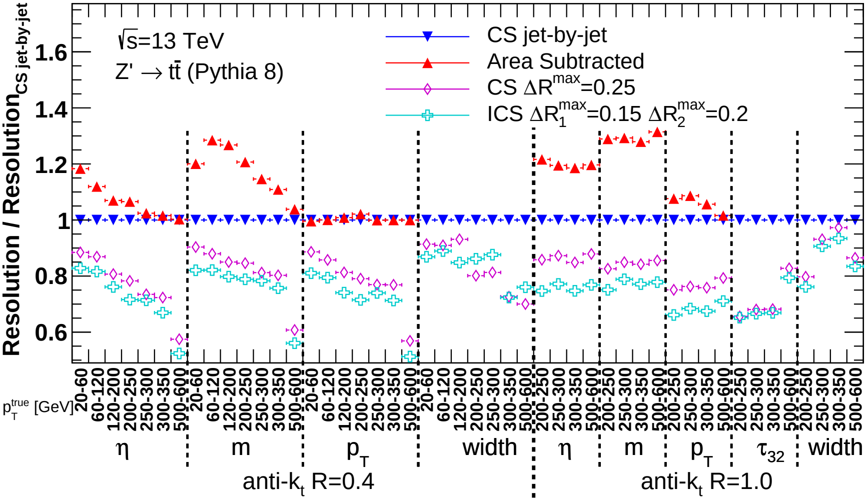


Iterative CS - bias, $N_{\mathrm{PU}} \in [100, 120]$



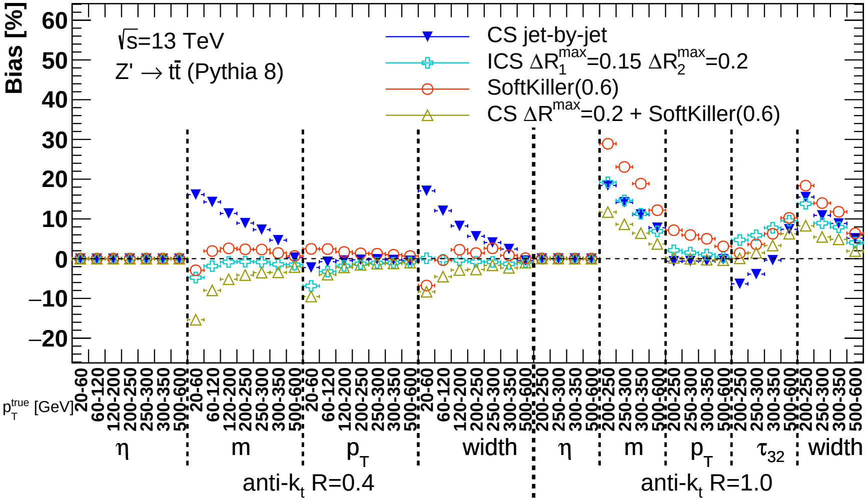
- Bias reduced in most cases with ICS
  - can be further reduced with calibration

Iterative CS - resolution, $N_{\mathrm{PU}} \in [100, 120]$



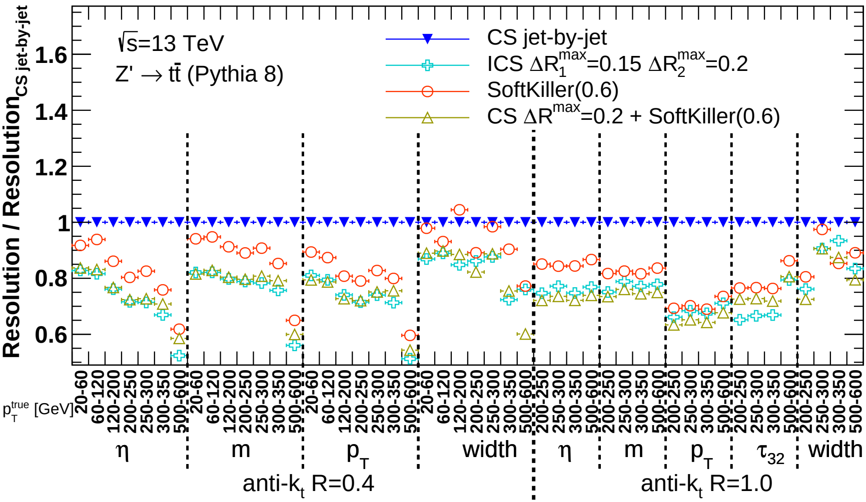
- Resolution much better with ICS

Comparison with SK - bias, $N_{\mathrm{PU}} \in [100, 120]$



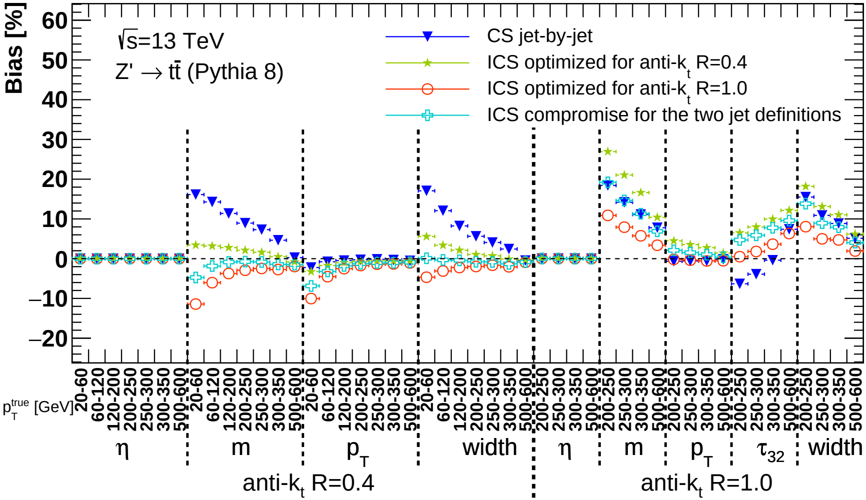
- Performance of SK and CS+SK also depends largely on their parameters
  - The shown CS+SK has larger bias for small-R, but lower bias for large-R jets

Comparison with SK - resolution, $N_{\mathrm{PU}} \in [100, 120]$



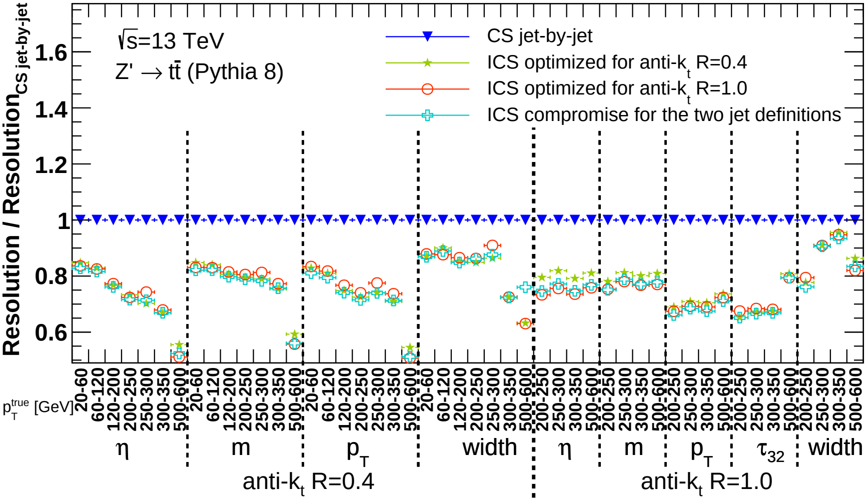
- Resolution similar for CS+SK and ICS
  - better than simple SK

Iterative CS optimization - bias, $N_{\mathrm{PU}} \in [100, 120]$



- Bias can be reduced when optimizing for each jet definition separately

Iterative CS optimization - resolution, $N_{\mathrm{PU}} \in [100, 120]$



- Similar resolution for the three tested ICS methods

Iterative CS - summary

- Improved performance compared to simply doing whole event CS
- Bias can be further reduced when optimizing for each jet definition separately or by calibration
- Investigated two iterations so far
  - three iterations or combination with SK can lead to further improvement
- The performance with real detector can be different
- Each experiment should find its own optimal parameters

Summary

- Jet-by-jet CS used among the heavy-ion community for physics results
- Recommending to use ρ rescaling always.
  - Examples: https://fastjet.hepforge.org/trac/browser/contrib/contribs/ConstituentSubtractor/tags/1.3.1/example_background_rescaling.cc
- Iterative CS brings large improvement in resolution keeping the bias well controlled
  - Please, try it. Example: https://fastjet.hepforge.org/trac/browser/contrib/contribs/ConstituentSubtractor/tags/1.3.1/example_whole_event_iterative.cc
- Let us know if you would like to have new features in the fastjet-contrib implementation of Constituent Subtraction

BACKUP

The correction procedure - jet-by-jet

- for each event
  1. estimate the pileup p_{T} density, ρ, in the event,
  2. add ghosts (infinitesimally small p_{T}^{g}) among particles in the event and apply jet clustering algorithm to all particles and ghosts $\Rightarrow$ the jets are composited from particles and ghosts,
- for each jet in the event
  3. set for each ghost $p_{\mathrm{T}}^{g} = \rho A_{g}$
  4. evaluate distance $\Delta R_{i,k}$ between particle i and ghost k for each possible particle-ghost pair and sort them in ascending order:

$$\Delta R_{i,k} = p_{\mathrm{T}i}^{\alpha} \cdot \sqrt{(y_i - y_k^g)^2 + (\phi_i - \phi_k^g)^2}. \quad (1)$$

     α - free parameter
  5. iteratively change transverse momenta by applying the following procedure for each ghost-particle pair until no more pairs remain or $\Delta R_{i,k} > \Delta R^{\max}$:

$$\text{If } p_{\mathrm{T}i} \geq p_{\mathrm{T}k}^{g}: \quad p_{\mathrm{T}i} \rightarrow p_{\mathrm{T}i} - p_{\mathrm{T}k}^{g}, \; p_{\mathrm{T}k}^{g} \rightarrow 0; \qquad \text{otherwise:} \quad p_{\mathrm{T}i} \rightarrow 0, \; p_{\mathrm{T}k}^{g} \rightarrow p_{\mathrm{T}k}^{g} - p_{\mathrm{T}i}. \quad (2)$$

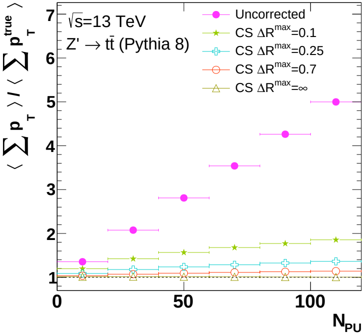
  6. after the iterative process, discard all particles with zero transverse momentum.

CS parameters

- $\Delta R^{\max}$- controls how distant particle-ghost pairs can be combined
  - low effect for jet-by-jet CS
  - huge effect for whole event CS
- α - with $\alpha > 0$, the low p_{T} particles are favored in the subtraction procedure
  - very low effect for jet-by-jet CS
  - it can lead to slightly better performance for whole event CS
- ghost area - specifies the density of ghosts
  - the smaller, the better
  - but the correction time gets larger with smaller ghost area

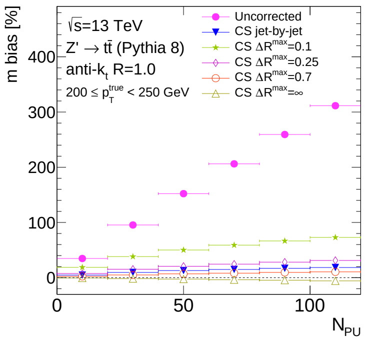
Whole event CS - $\Delta R^{\max}$ parameter

- Evaluated the average scalar sum of p_{T} from all particles within the event

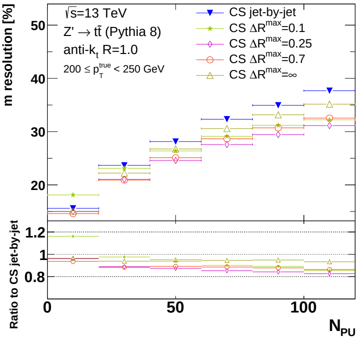


- $\Delta R^{\max} = \infty$ removes p_{T} corresponding to p_{T} from pileup $\Rightarrow$ the background estimation works well
- Lower $\Delta R^{\max}$ keeps certain fraction of pileup in events, e.g. $\Delta R^{\max} = 0.25$ keeps $\sim 12\%$ of pileup p_{T} within event on average

Whole event CS - $\Delta R^{\max}$ parameter



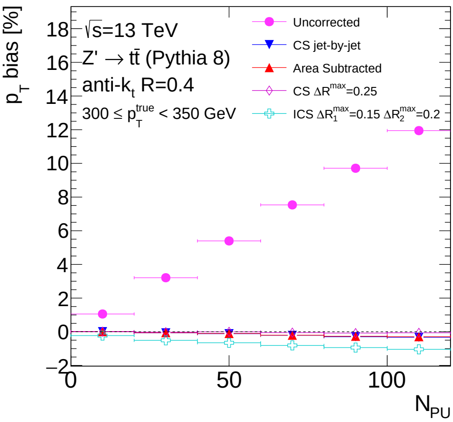
jet mass



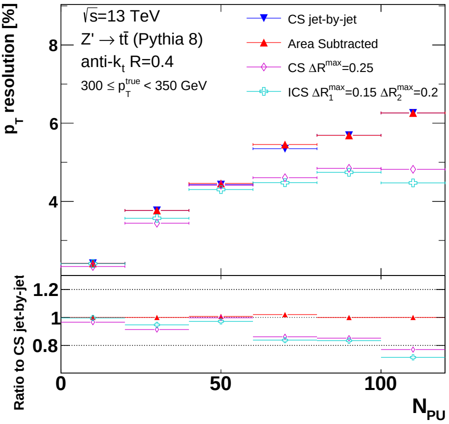
jet mass

- Also the optimal $\Delta R^{\max}$ parameter varies for jet observables, p_{T} ranges, metrics, ...
- In our studies, we found that $\Delta R^{\max} \in [0.2, 0.5]$ always lead to better performance wrt jet-by-jet correction
  - Experiments should find their own optimal value of $\Delta R^{\max}$

Iterative CS, jet pt

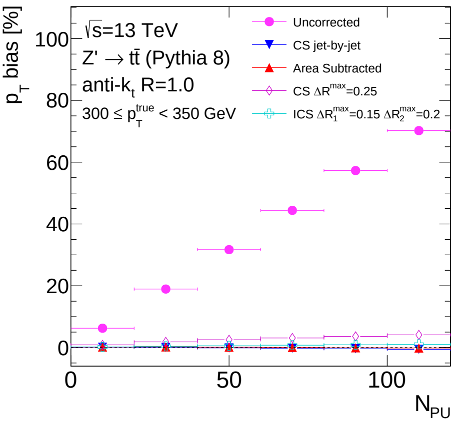


bias

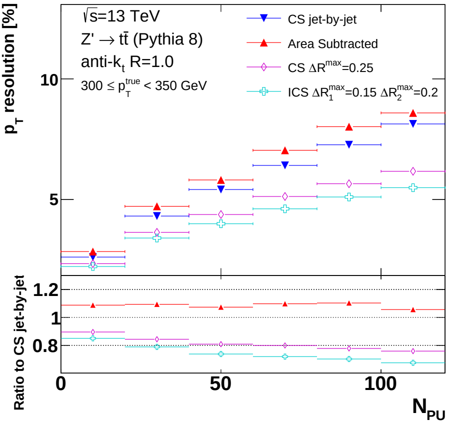


resolution

Iterative CS, jet pt

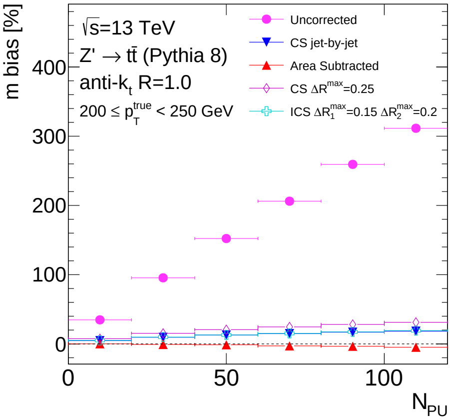


bias

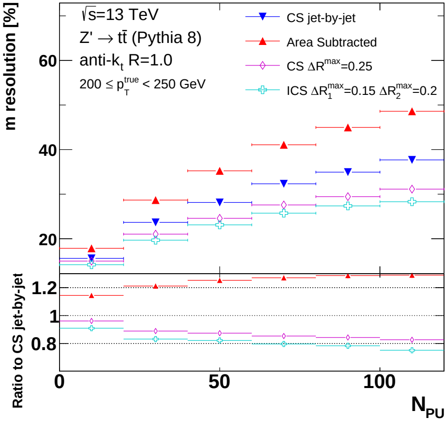


resolution

Iterative CS, jet mass



bias



resolution