Improved Jet Algorithms and the Underlying Event

Matteo Cacciari[1], **Juan Rojo**[1], Gavin Salam[1], Gregory Soyez[2]

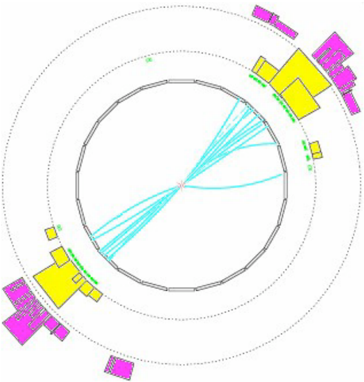
[1]LPTHE, UPMC Paris VI, [2]Brookhaven National Laboratory

QCD/EW KEK-D Workshop, ITH Zurich, 03/06/2008

Outline

JETS AT THE LHC

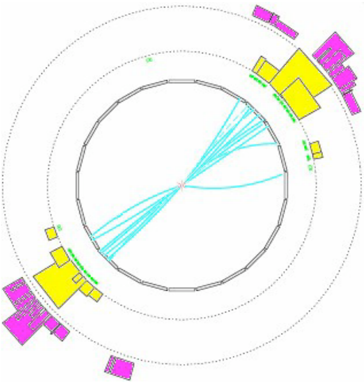
Counting jets



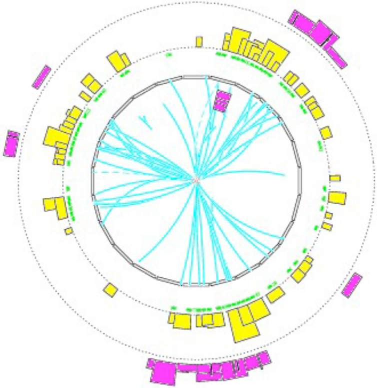
A two jet event!

Three jets? Four jets? ...

Counting jets



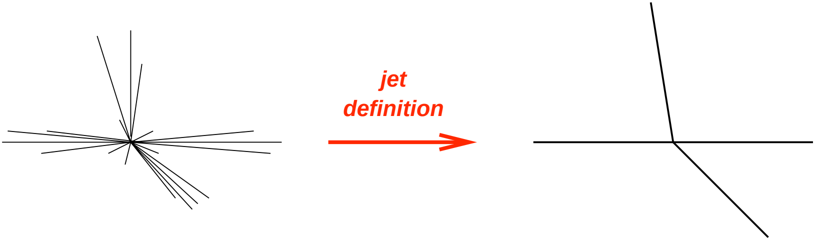
A two jet event!



Three jets? Four jets? ...

Jet algorithms

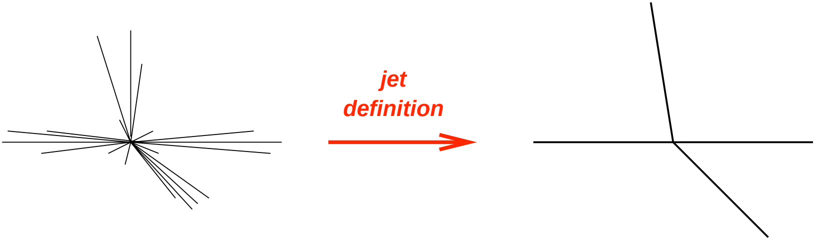
Jet finding not an easy eyeball task ...
A jet algorithm is a non-ambiguous mapping between observed hadrons (or calorimeter towers) and the original hard scattering partons.
It is a systematic procedure that projects away the multi-particle dynamics, so as to leave a simple picture of the event.



Jets are as close as we can get to a physical single quark or gluon.

Jet algorithms

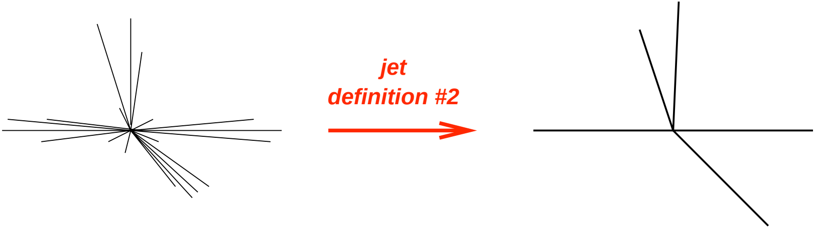
Jet finding not an easy eyeball task ...
A jet algorithm is a non-ambiguous mapping between observed hadrons (or calorimeter towers) and the original hard scattering partons.
It is a systematic procedure that projects away the multi-particle dynamics, so as to leave a simple picture of the event.



Jets are as close as we can get to a physical single quark or gluon.

Jet algorithms

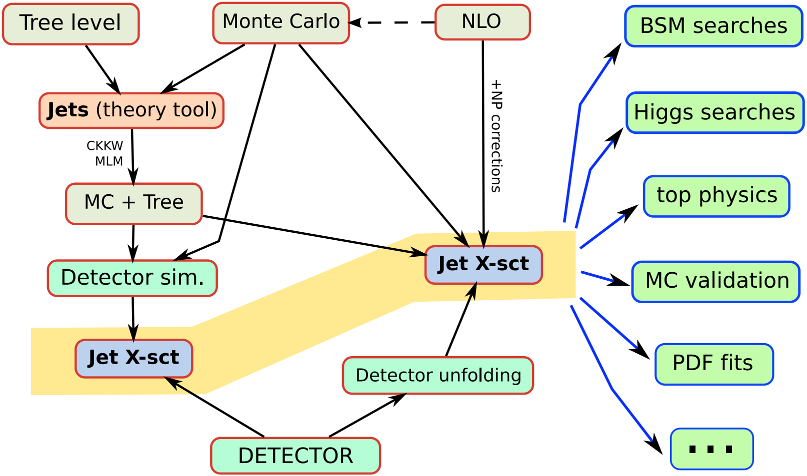
Jet finding not an easy eyeball task ...
A jet algorithm is a non-ambiguous mapping between observed hadrons (or calorimeter towers) and the original hard scattering partons.
It is a systematic procedure that projects away the multi-particle dynamics, so as to leave a simple picture of the event.



Jets are as close as we can get to a physical single quark or gluon.

Analysis flowchart

Jet algorithms present in almost all analysis at the LHC



Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and ep colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,. . .)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,. . .)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and ep colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)⟶ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) ⟶ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

Jet algorithm types

Mainstream jet-algorithms

- Iterative **cone** algorithms (top-down)$\longrightarrow$ (JetClu, ILCA/Midpoint, LHC cones,...)
  Searches for cones centered on regions of energy flow
  Dominant at hadron colliders.
  With split-merge: CDF JetClu, Atlas Cone, PxCone, MidPoint
  Without split-merge: UA1 cone, CellJet, GetJet, CMS cone?
- Sequential **recombination** algorithms (bottom-up) $\longrightarrow$ (k_t, Cambridge/Aachen, Jade)
  Recombine closest pair of particles, next closest, etc.
  Dominant at e^+e^- and *ep* colliders, and successfully used at Tevatron and now included in LHC analysis software.

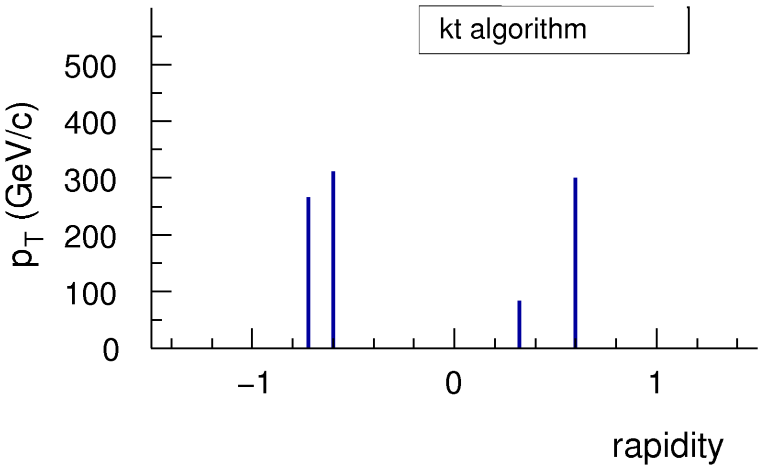
What is **needed** of a jet algorithm

- Must be infrared and collinear (IRC) safe
- Must be identical procedure at parton level, hadron-level and experimental level
  So that theory calculations can be compared to measurements

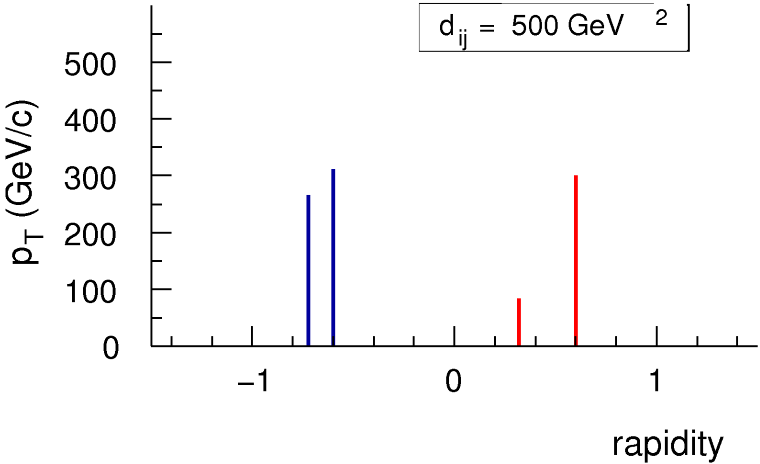
What is **nice** for a jet algorithm

- Shouldn't be too sensitive to hadronization, underlying event and pileup, while being sensitive to perturbative radiation.
- Should be realistically applicable at detector level (i.e. allow fast implementations, to cope with the large multiplicities at hadronic colliders).

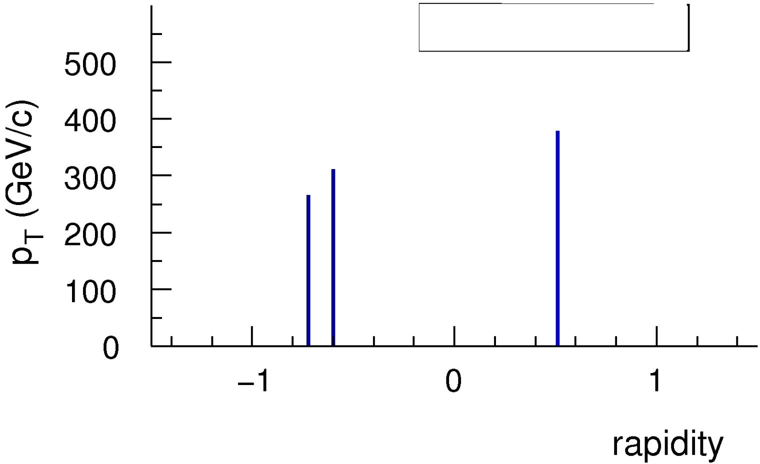
k_t algorithm in action ($R = 1$)



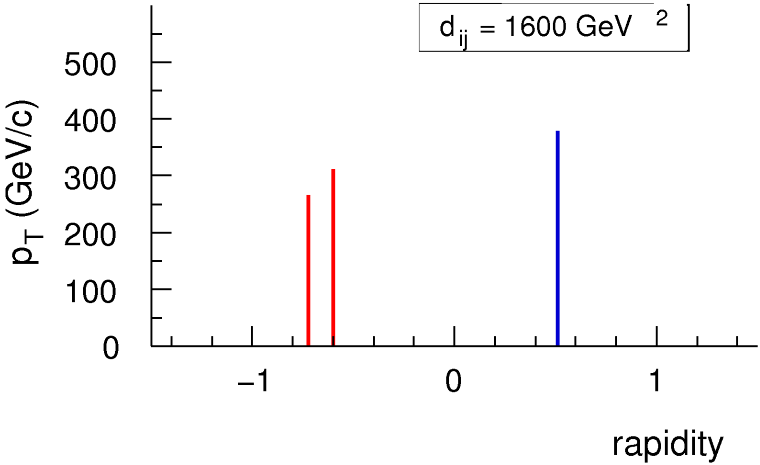
k_t algorithm in action ($R = 1$)



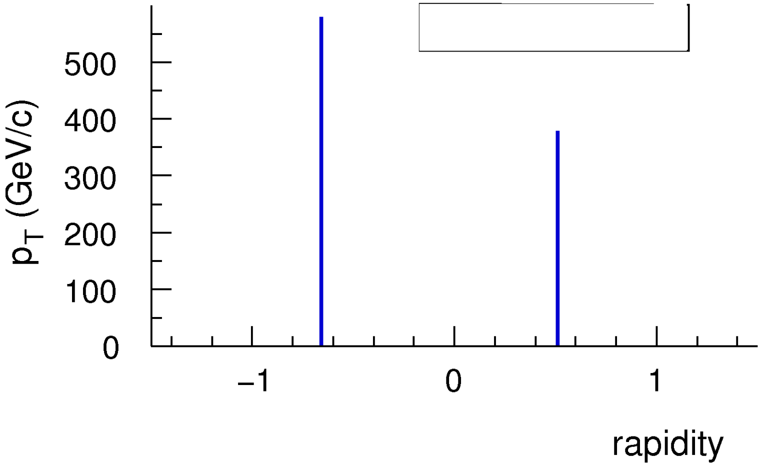
k_t algorithm in action $(R = 1)$



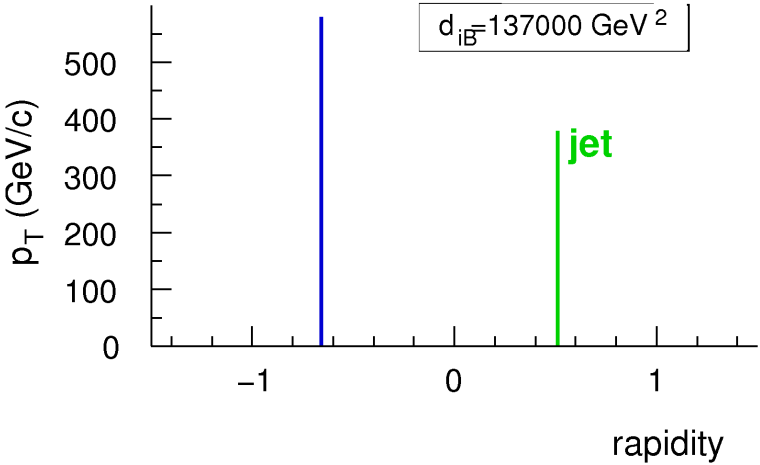
k_t algorithm in action ($R = 1$)



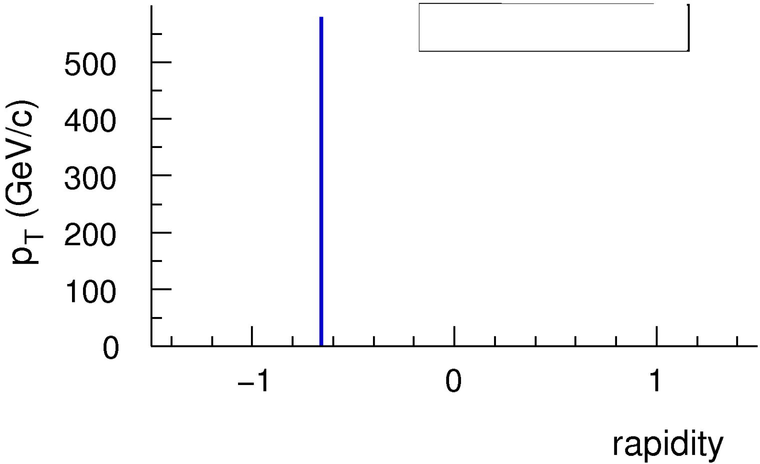
k_t algorithm in action ($R = 1$)



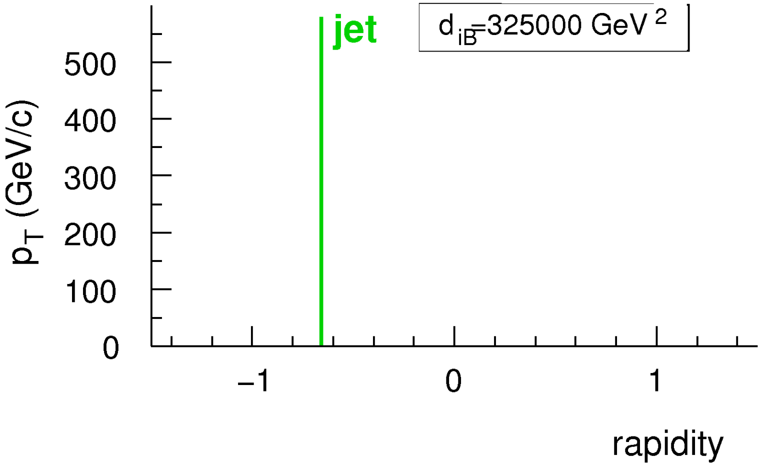
k_t algorithm in action ($R = 1$)



k_t algorithm in action ($R = 1$)



k_t algorithm in action ($R = 1$)



RECENT DEVELOPEMENTS IN JET ALGORITHMS

References:

M. Cacciari and G. Salam, hep-ph/0512210

G. Salam and M. Cacciari, arXiv: arXiv:0704.0292

M. Cacciari, G. Salam and G. Soyez, arXiv:0802.1189

Progress in jet algorithms

The `FastJet` package: a fast implementation of the k_T algorithm (based on computational geometry) with background subtraction methods and external jet finders as plug-ins (Cacciari, Salam and Soyez 05-08).

`http://www.lpthe.jussieu.fr/˜ salam/fastjet/`

New jet algorithms have appeared in the last years:

- `SISCone`: A Seedless and Infrared Safe Cone algorithm (Salam and Soyez 07). IRC safe at all orders of pert. theory. Finds all stable cones using geometrical techniques.
- The anti-k_T algorithm: A generalization of the k_T algorithm with measure:

$$d_{ij} = \min(k_{ti}^{2n}, k_{tj}^{2n})\Delta R_{ij}^2/R^2 \ , n=-1 \qquad , d_{iB} = k_{ti}^2$$

  which acts like a perfect cone algorithm with area $A_{\rm jet} = \pi R^2$ (Cacciari, Salam, Soyez 08). For k_T one has $n=1$ and for Cambridge $n=0$.

Progress in jet algorithms

The FastJet package: a fast implementation of the k_T algorithm (based on computational geometry) with background subtraction methods and external jet finders as plug-ins (Cacciari, Salam and Soyez 05-08).

http://www.lpthe.jussieu.fr/~ salam/fastjet/

New jet algorithms have appeared in the last years:

- SISCone: A Seedless and Infrared Safe Cone algorithm (Salam and Soyez 07). IRC safe at all orders of pert. theory. Finds all stable cones using geometrical techniques.
- The anti-k_T algorithm: A generalization of the k_T algorithm with measure:

$$d_{ij} = \min(k_{ti}^{2n}, k_{tj}^{2n})\Delta R_{ij}^2/R^2 \ , n = -1 \qquad , d_{iB} = k_{ti}^2$$

  which acts like a perfect cone algorithm with area $A_{\rm jet} = \pi R^2$ (Cacciari, Salam, Soyez 08). For k_T one has $n = 1$ and for Cambridge $n = 0$.

Progress in jet algorithms

The `FastJet` package: a fast implementation of the k_T algorithm (based on computational geometry) with background subtraction methods and external jet finders as plug-ins (Cacciari, Salam and Soyez 05-08).

`http://www.lpthe.jussieu.fr/~ salam/fastjet/`

New jet algorithms have appeared in the last years:

- `SISCone`: A Seedless and Infrared Safe Cone algorithm (Salam and Soyez 07). IRC safe at all orders of pert. theory. Finds all stable cones using geometrical techniques.
- The anti-k_T algorithm: A generalization of the k_T algorithm with measure:

$$d_{ij} = \min(k_{ti}^{2n}, k_{tj}^{2n})\Delta R_{ij}^2/R^2 \ , n = -1 \qquad , d_{iB} = k_{ti}^2$$

which acts like a perfect cone algorithm with area $A_{\rm jet} = \pi R^2$ (Cacciari, Salam, Soyez 08). For k_T one has $n = 1$ and for Cambridge $n = 0$.

Progress in jet algorithms

The `FastJet` package: a fast implementation of the k_T algorithm (based on computational geometry) with background subtraction methods and external jet finders as plug-ins (Cacciari, Salam and Soyez 05-08).

`http://www.lpthe.jussieu.fr/~ salam/fastjet/`

New jet algorithms have appeared in the last years:

- `SISCone`: A Seedless and Infrared Safe Cone algorithm (Salam and Soyez 07). IRC safe at all orders of pert. theory. Finds all stable cones using geometrical techniques.
- The anti-k_T algorithm: A generalization of the k_T algorithm with measure:

$$d_{ij} = \min(k_{ti}^{2n}, k_{tj}^{2n}) \Delta R_{ij}^2 / R^2 \ , n = -1 \qquad , d_{iB} = k_{ti}^2$$

which acts like a perfect cone algorithm with area $A_{\rm jet} = \pi R^2$ (Cacciari, Salam, Soyez 08). For k_T one has $n = 1$ and for Cambridge $n = 0$.

Progress in jet algorithms

The FastJet package: a fast implementation of the k_T algorithm (based on computational geometry) with background subtraction methods and external jet finders as plug-ins (Cacciari, Salam and Soyez 05-08).

`http://www.lpthe.jussieu.fr/~ salam/fastjet/`

New jet algorithms have appeared in the last years:

- SISCone: A Seedless and Infrared Safe Cone algorithm (Salam and Soyez 07). IRC safe at all orders of pert. theory. Finds all stable cones using geometrical techniques.
- The anti-k_T algorithm: A generalization of the k_T algorithm with measure:

$$d_{ij} = \min(k_{ti}^{2n}, k_{tj}^{2n})\Delta R_{ij}^2/R^2 \ , n = -1 \qquad , d_{iB} = k_{ti}^2$$

  which acts like a perfect cone algorithm with area $A_{\rm jet} = \pi R^2$ (Cacciari, Salam, Soyez 08). For k_T one has $n = 1$ and for Cambridge $n = 0$.

Infrared safety

Cone algorithms have been known to suffer from Infrared and Collinear unsafety for many years.

For the CDF MidPoint cone algorithm (even worse for the Iterative Cone!):

Observable	1st miss cones at	Last meaningful order
Inclusive jet cross section	NNLO	NLO
$W/Z/H+1$ jet cross section	NNLO	NLO
3 jet cross section	NLO	LO
$W/Z/H+2$ jet cross section	NLO	LO
jet masses in 3 jets, $W/Z/H+2$ jets	LO	none

Table 2: Summary of the order (α_s^4 or $\alpha_s^3\alpha_{EW}$) at which stable cones are missed in various processes with a midpoint algorithm, and the corresponding last order that can be meaningfully calculated. Infrared unsafety first becomes visible one order beyond that at which one misses stable cones.

Theory investment in NLO computations: ~ 50 people $\times$ 10 years $\sim 30-50$ million \$ $\rightarrow$ Lost if IRC unsafe jet algorithms used at the LHC!

Infrared safety

Cone algorithms have been known to suffer from Infrared and Collinear unsafety for many years.
For the CDF MidPoint cone algorithm (even worse for the Iterative Cone!):

Observable	1st miss cones at	Last meaningful order
Inclusive jet cross section	NNLO	NLO
$W/Z/H + 1$ jet cross section	NNLO	NLO
3 jet cross section	NLO	LO
$W/Z/H + 2$ jet cross section	NLO	LO
jet masses in 3 jets, $W/Z/H + 2$ jets	LO	none

Table 2: Summary of the order (α_s^4 or $\alpha_s^3\alpha_{EW}$) at which stable cones are missed in various processes with a midpoint algorithm, and the corresponding last order that can be meaningfully calculated. Infrared unsafety first becomes visible one order beyond that at which one misses stable cones.

Theory investment in NLO computations: ~ 50 people $\times$ 10 years $\sim 30-50$ million \$ $\rightarrow$ Lost if IRC unsafe jet algorithms used at the LHC!

Infrared safety

Cone algorithms have been known to suffer from Infrared and Collinear unsafety for many years.
For the CDF MidPoint cone algorithm (even worse for the Iterative Cone!):

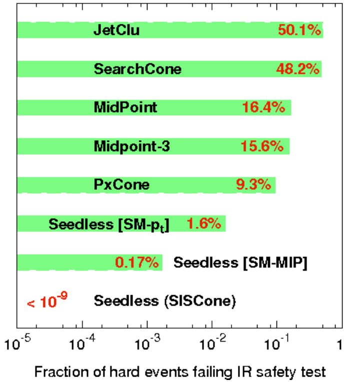
Observable	1st miss cones at	Last meaningful order
Inclusive jet cross section	NNLO	NLO
$W/Z/H + 1$ jet cross section	NNLO	NLO
3 jet cross section	NLO	LO
$W/Z/H + 2$ jet cross section	NLO	LO
jet masses in 3 jets, $W/Z/H + 2$ jets	LO	none

Table 2: Summary of the order (α_s^4 or $\alpha_s^3\alpha_{EW}$) at which stable cones are missed in various processes with a midpoint algorithm, and the corresponding last order that can be meaningfully calculated. Infrared unsafety first becomes visible one order beyond that at which one misses stable cones.

Theory investment in NLO computations: ~ 50 people $\times$ 10 years $\sim 30-50$ million \$ $\rightarrow$ Lost if IRC unsafe jet algorithms used at the LHC!

Infrared safety

IRC unsafety affects a large fraction of events!
For JetClu (similar to Atlas cone), half of events fails IRC safety tests.
Even for the MidPoint cone algorithm, 15% of events fail the test!



CMS recommends the use of SISCone as reference cone algorithm

Progress in jet algorithms - Status 2006

Algorithm	Type	IRC status	Notes
exclusive k_t	$SR_{p=1}$	OK	
inclusive k_t	$SR_{p=1}$	OK	widespread: QCD-th, HERA
Cambridge/Aachen	$SR_{p=0}$	OK	
Run II Seedless cone	SC-SM	OK	slow: $N2^N$!!
CDF JetClu	IC_r-SM	IR_{2+1}	for top physics, searches
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}	$\simeq$ Tev Run II recommend^n
CDF MidPoint searchcone	$IC_{se,mp}$-SM	IR_{2+1}	
D0 Run II cone	IC_{mp}-SM	IR_{3+1}	Tev Run II + cut on cone p_t
ATLAS Cone	IC-SM	IR_{2+1}	
PxCone	IC_{mp}-SD	IR_{3+1}	has cut on cone p_t,
CMS Iterative Cone	IC-PR	$Coll_{3+1}$	
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$	widespread in BSM theory
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$	likewise

SR = seq.rec.; IC = it.cone; FC = fixed cone;
SM = split–merge; SD = split–drop; PR = progressive removal

IR_{n+1}: for n nearby hard partons, 1 soft emitted gluon can change hard jets
$Coll_{n+1}$: for n nearby hard partons, 1 collinear splitting can change hard jets

Progress in jet algorithms - Status 2008

Algorithm	Type	IRC status	Notes
exclusive k_t	$SR_{p=1}$	OK	
inclusive k_t	$SR_{p=1}$	OK	widespread: QCD-th, HERA
Cambridge/Aachen	$SR_{p=0}$	OK	
Run II Seedless cone	SC-SM	OK	slow: $N2^N$!!
CDF JetClu	IC_r-SM	IR_{2+1}	for top physics, searches
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}	$\simeq$ Tev Run II recommend^n
CDF MidPoint searchcone	$IC_{se,mp}$-SM	IR_{2+1}	
D0 Run II cone	IC_{mp}-SM	IR_{3+1}	Tev Run II + cut on cone p_t
ATLAS Cone	IC-SM	IR_{2+1}	
PxCone	IC_{mp}-SD	IR_{3+1}	has cut on cone p_t,
CMS Iterative Cone	IC-PR	$Coll_{3+1}$	
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$	widespread in BSM theory
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$	likewise

SR = seq.rec.; IC = it.cone; FC = fixed cone;
SM = split–merge; SD = split–drop; PR = progressive removal

IR_{n+1}: for n nearby hard partons, 1 soft emitted gluon can change hard jets
$Coll_{n+1}$: for n nearby hard partons, 1 collinear splitting can change hard jets

Progress in jet algorithms - Status 2008

Algorithm	Type	IRC status	Evolution
exclusive k_t	$SR_{p=1}$	OK	$N^3 \to N \ln N$
inclusive k_t	$SR_{p=1}$	OK	$N^3 \to N \ln N$
Cambridge/Aachen	$SR_{p=0}$	OK	$N^3 \to N \ln N$
Run II Seedless cone	SC-SM	OK	$\to$ SISCone
CDF JetClu	IC_r-SM	IR_{2+1}	[$\to$ SISCone]
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}	$\to$ SISCone
CDF MidPoint searchcone	$IC_{se,mp}$-SM	IR_{2+1}	[$\to$ SISCone]
D0 Run II cone	IC_{mp}-SM	IR_{3+1}	$\to$ SISCone [with p_t cut?]
ATLAS Cone	IC-SM	IR_{2+1}	$\to$ SISCone
PxCone	IC_{mp}-SD	IR_{3+1}	[little used]
CMS Iterative Cone	IC-PR	$Coll_{3+1}$	$\to$ anti-k_t
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$	$\to$ anti-k_t
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$	$\to$ anti-k_t

SR = seq.rec.; IC = it.cone; FC = fixed cone;

SM = split–merge; SD = split–drop; PR = progressive removal

Progress in jet algorithms

Disclaimer: topics which cannot be covered include:

- Jet flavour and *b*-jets (Banfi, Salam and Zanderighi, hep-ph/0601139, arXiv:0704.2999 [hep-ph])
- Analytical studies of jet algorithms (Dasgupta, Magnea and Salam, arXiv:0712.3014 [hep-ph])
- Strategies to asses jet algorithms performance, Rabbertz et al., Campanelli et al. in arXiv:0803.0678
- Jet substructure as a new Higgs search channel at the LHC, Butterworth, Davison, Rubin, Salam, arXiv:0802.2470

JET AREAS, THE UNDERLYING EVENT AND ITS MEASURE

References:
M. Cacciari talk at Les Houches 2007
Cacciari and Salam, arXiv:0707.1378
Cacciari, Salam and Soyez, arxiv:0802.1188

Jet areas

Jet areas provide a quantitative definition of the concept of the *size* of a jet (Cacciari, Salam and Soyez, arxiv:0802.1188)

Various definitions (active, passive, Voronoi), coincide for high multiplicity

Active area → add a large number of ghosts (particles with infinitesimal p_T) and perform jet clustering:

$$A(J\,|\,\{g_i\}) = \frac{N_g(J)}{\nu_g}$$

Number of ghosts in jet J

Active area of a single ghosts configuration

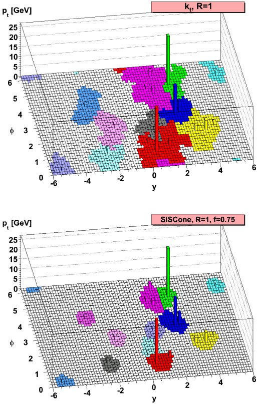
Ghost density

$$A(J) = \lim_{\nu_g \to \infty} \langle A(J\,|\,\{g_i\}) \rangle_g$$

Active area

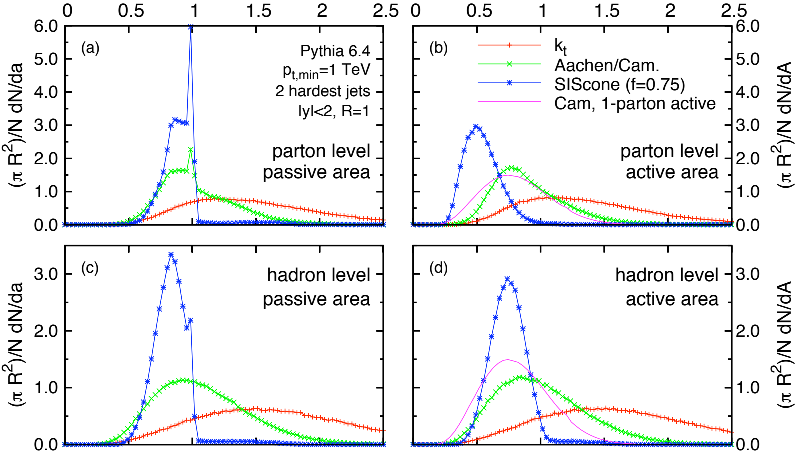
Area computations only meaningful for IRC safe algorithms and practical with fast implementations → Both conditions fulfilled in `FastJet`.

Jet areas



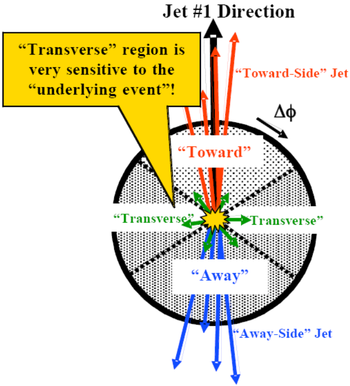
$A_{\rm jet} \neq \pi R^2$ in general (but note anti-k_T)

Jet areas



SISCone smallest active area → Reduced sensitivity to diffuse soft radiation

Standard approach to the UE



Marchesini-Webber idea:
look at transverse region to
measure underlying event

Topological selection

The jets are classified as belonging to the noise on the ground of their **position**

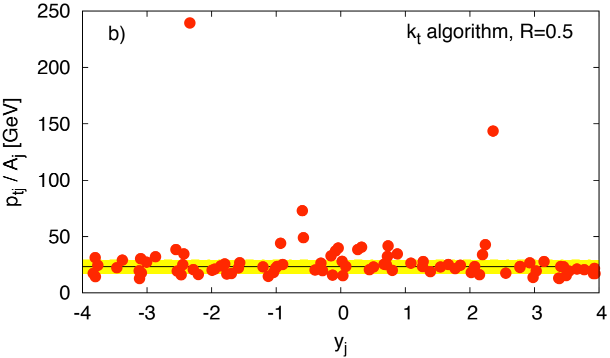
But: difficult to use in busy events, misses point-to-point and event-by event fluctuations

Area-based UE measure and subtraction

We propose a dynamical selection: jets assigned to background due to its characteristics: UE/PU can be measured and subtracted based on jet areas

Basic idea: Jets formed of soft stuff only have a p_T proportional to its area

Take at $p_T \sim 200$ GeV dijet event with 20 min. bias collisions:



Area-based UE measure and subtraction

We propose a dynamical selection: jets assigned to background due to its characteristics: UE/PU can be measured and subtracted based on jet areas

- Determine the noise density per unit area ρ in an event-by-event basis with the k_T−algorithm and $R = 0.5$

$$\rho \equiv \text{median}\left[\left\{\frac{p_{tj}}{A_j}\right\}\right] \quad (1)$$

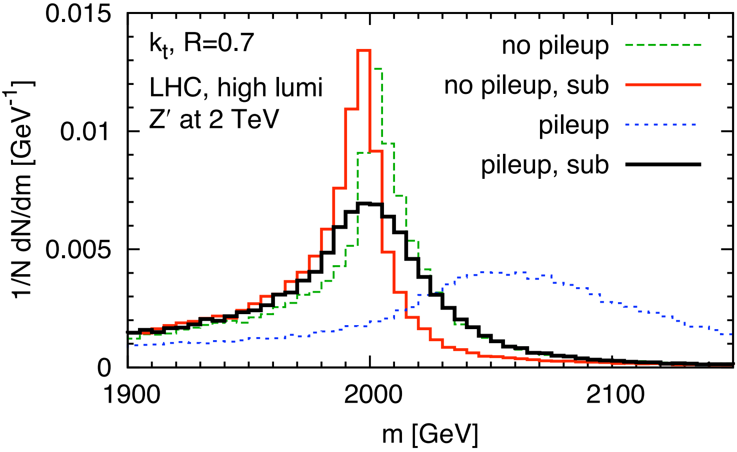
- Subtract the PU contribution to the jet using its area

$$p_{\mu j}^{(\mathrm{sub})} = p_{\mu j} - A_{\mu j}\,\rho \pm \sigma_\rho \sqrt{A_{\mathrm{j}}} \quad (2)$$

Note that no cut in the p_T of input particles required
N.B. : Intrinsic uncertainty in subtraction due to background fluctuations σ_ρ
Other limitations include backreaction (effect of min. bias particles in jet clustering)

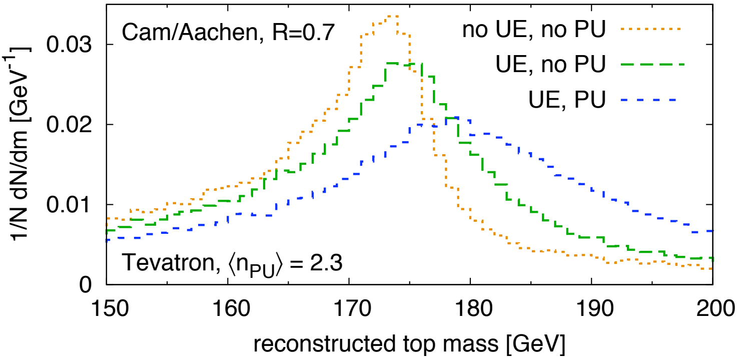
Subtraction in practice

First example: Z' invariant mass distribution



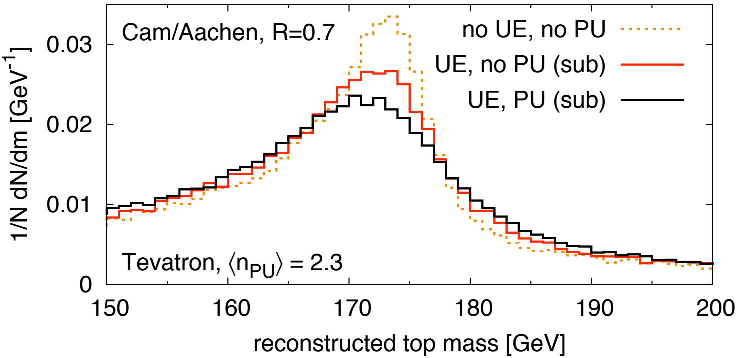
Subtraction in practice

Second example: Z' Z' invariant mass distribution



Subtraction in practice

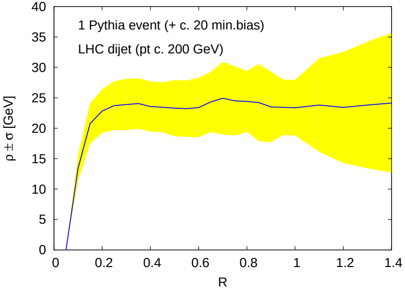
Second example: Z' Z' invariant mass distribution

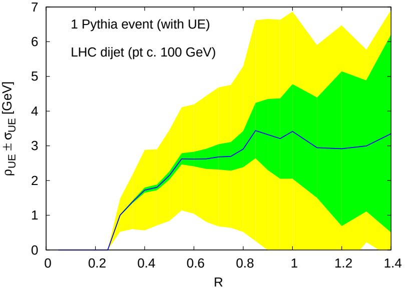


Note that subtraction not only corrects the peak position, it also improves sizably the mass resolution

Measuring PileUp and Underlying event

The area-based subtraction strategy can also be used to measure UE and PU directly from experimental data (or MC models).



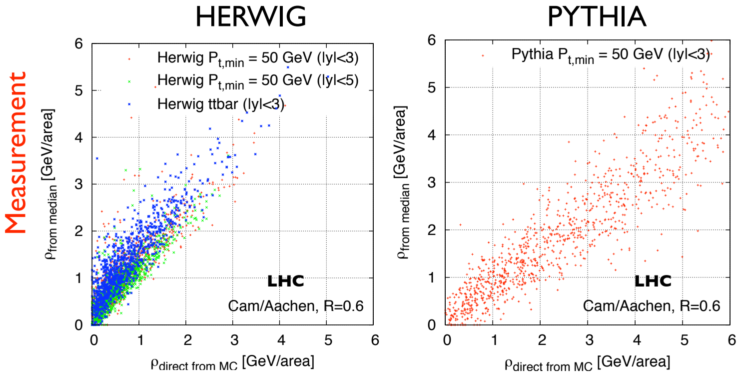


Model independent UE/PU measurement

Note the size of the background fluctuations (basic limit on the accuracy of method)

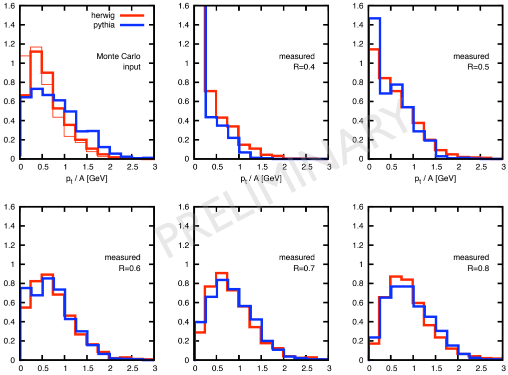
Measuring PileUp and Underlying event

The area-based subtraction strategy can also be used to measure UE and PU directly from experimental data (or MC models).
Compare measured UE with known amount in MC:



Underlying event distributions

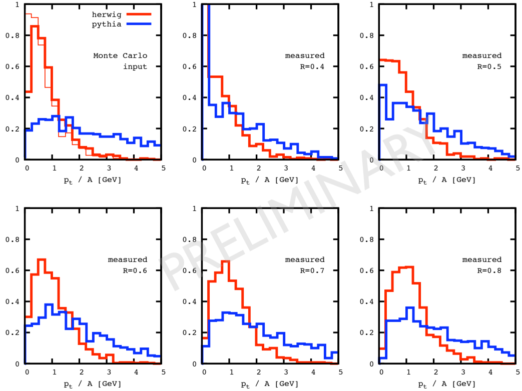
Measure distribution of ρ_{UE} from `Pythia` and `Herwig` at TeV and LHC:



Proposal: Repeat the comparison with modern event generators → Last call for predictions before LHC start-up

Underlying event distributions

Measure distribution of ρ_{UE} from Pythia and Herwig at TeV and LHC:



Proposal: Repeat the comparison with modern event generators → Last call for predictions before LHC start-up

QUANTIFYING THE PERFORMANCE OF JET ALGORITHMS AT THE LHC

References:
M. Cacciari, J. Rojo, G. Soyez and G. Salam, Les Houches 2007, Tools and Jets Summary Report, arXiv:0803.0678

+ Work in progress

The performance of jet algorithms

How to determine which jet algorithm is optimal for a given process?
We need quality measures that:

- Do not depend on ill-defined quantities (like original parton p_T or direction, a parton is an ambiguous concept in pQCD)
- Do not assume any underlying probability distribution of (MC) data, *i.e.*, whether the reconstructed mass distribution is gaussian, asymmetric, ...

Processes studied:

- Fictitious narrow width $Z' \to q\bar{q}$ decays for various $M_{Z'}$ (100 GeV - 4 TeV): Source of mono-energetic quark jets
- Fictitious narrow width $H \to gg$ decays for various M_H (100 GeV - 4 TeV): Source of mono-energetic gluon jets

Quality measures

1. $Q^{w}_{f=z}(R) \rightarrow$ The **width of the smallest (reconstructed) mass window** that contains a fraction $f = z$ of the generated massive objects:

$$f = \left(\frac{\#\ \text{reco. massive objects in window of width}\ w}{\text{Total}\ \#\ \text{generated massive objects}} \right)$$

2. The **max. fraction of evs.** f in window of width $w = x\sqrt{M}$:

$$Q^{f}_{w=x\sqrt{M}}(R) \equiv \left(\frac{\text{Max}\ \#\ \text{reco. mass. obj. in width}\ w = x\sqrt{M}}{\text{Total}\ \#\ \text{generated massive objects}} \right)^{-1}$$

Consistent results obtained with both measures

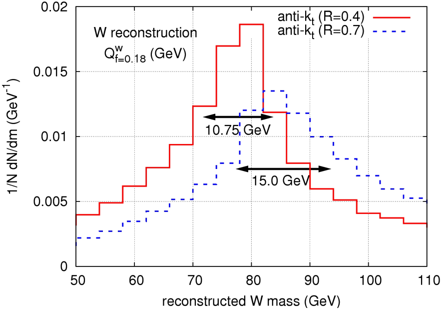
Quality measures

1. $Q^{w}_{f=z}(R) \to$ The **width of the smallest (reconstructed) mass window** that contains a fraction $f = z$ of the generated massive objects:

$$f = \left(\frac{\#\text{ reconstructed massive objects in window of width } w}{\text{Total } \#\text{ generated massive objects}} \right) .$$

2. $Q^{f}_{w=x\sqrt{M}}(R) \to$ The **max. fraction of events** f in window of width $w = x\sqrt{M}$:

$$Q^{f}_{w=x\sqrt{M}}(R) \equiv \left(\frac{\text{Max } \#\text{ reconstructed massive objects in window of width } w = x\sqrt{M}}{\text{Total } \#\text{ generated massive objects}} \right)^{-1} ,$$



Effective luminosity ratio

Mapping variations of quality measures and of effective luminosity ratio $\rho_{\mathcal{L}}$
(Crude) Assumption: flat background, unaffected by jet clustering.
Define effective power to discriminate signal from background:

$$\Sigma^{\mathrm{eff}}\left(\mathrm{JA}, R\right) \equiv \frac{N_{\mathrm{signal}}}{\sqrt{N_{\mathrm{back}}}}$$

$$r_w \equiv \frac{Q^{w}_{f=z}\left(\mathrm{JA}_2, R_2\right)}{Q^{w}_{f=z}\left(\mathrm{JA}_1, R_1\right)} = \frac{N_{\mathrm{back}}\left(\mathrm{JA}_2, R_2\right)}{N_{\mathrm{back}}\left(\mathrm{JA}_1, R_1\right)} \, ,$$

$$\frac{\Sigma^{\mathrm{eff}}\left(\mathrm{JA}_1, R_1\right)}{\Sigma^{\mathrm{eff}}\left(\mathrm{JA}_2, R_2\right)} = \sqrt{r_w} \, , \qquad \rho_{\mathcal{L}} = \frac{1}{r_w} \, .$$

$$r_f \equiv \frac{Q^{f}_{w=x\sqrt{M}}\left(\mathrm{JA}_2, R_2\right)}{Q^{f}_{w=x\sqrt{M}}\left(\mathrm{JA}_1, R_1\right)} = \frac{N_{\mathrm{signal}}\left(\mathrm{JA}_1, R_1\right)}{N_{\mathrm{signal}}\left(\mathrm{JA}_2, R_2\right)} \, ,$$

$$\frac{\Sigma^{\mathrm{eff}}\left(\mathrm{JA}_1, R_1\right)}{\Sigma^{\mathrm{eff}}\left(\mathrm{JA}_2, R_2\right)} = r_f \, , \qquad \rho_{\mathcal{L}} = \frac{1}{r_f^2} \, .$$

Jet algorithms

We study the performance of modern, Infrared and Collinear safe jet algorithms:

1. k_T algorithm
2. Cambridge/Aachen (Cam/Aa) algorithm
3. SISCone algorithm
4. Anti-k_T algorithm
5. Cam/Aa(filt) $\longrightarrow$ novel jet finding strategy based on Cambridge/Aachen with subjet filtering

Subjet filtering strategy

We would like a to have a jet algorithm which keeps most perturbative radiation while *filtering away* most of the UE/PileUp noise →

1. Cluster **all the particles in the event** with a given jet definition (JA_1,R_1).
2. Take **each of the jets** of the event and **cluster its constituents** only with another jet definition (JA_2,R_2) with $R_2 < R_1$. This operation will provide us with a **set of subjets** of the original jet.
3. **Keep the** n_{sj} **subjets of a jet with largest** p_T and **throw way the remaining subjets**.
4. Original jets are replaced merging the selected subjets

This procedures defines the subjet filtering strategy.
Natural choice → JA_1=JA_2 → Cambridge/Aachen algorithm (distance d_{ij} based on angular separation).

In our analysis we use $n_{\mathrm{sj}} = 2$ and $R_2 = R_1/2$.

Subjet filtering strategy

We would like a to have a jet algorithm which keeps most perturbative radiation while *filtering away* most of the UE/PileUp noise $\rightarrow$

1. Cluster **all the particles in the event** with a given jet definition (JA_1,R_1).
2. Take **each of the jets** of the event and **cluster its constituents** only with another jet definition (JA_2,R_2) with $R_2 < R_1$. This operation will provide us with a **set of subjets** of the original jet.
3. **Keep the** n_{sj} **subjets of a jet with largest** p_T and **throw way the remaining subjets**.
4. Original jets are replaced merging the selected subjets

This procedures defines the subjet filtering strategy.
Natural choice $\rightarrow$ JA_1=JA_2 $\rightarrow$ Cambridge/Aachen algorithm (distance d_{ij} based on angular separation).

In our analysis we use $n_{\mathrm{sj}} = 2$ and $R_2 = R_1/2$.

Subjet filtering strategy

We would like a to have a jet algorithm which keeps most perturbative radiation while *filtering away* most of the UE/PileUp noise $\rightarrow$

1. Cluster **all the particles in the event** with a given jet definition (JA_1,R_1).
2. Take **each of the jets** of the event and **cluster its constituents** only with another jet definition (JA_2,R_2) with $R_2 < R_1$. This operation will provide us with a **set of subjets** of the original jet.
3. **Keep the n_{sj} subjets of a jet with largest p_T** and **throw way the remaining subjets**.
4. Original jets are replaced merging the selected subjets

This procedures defines the subjet filtering strategy.
Natural choice $\rightarrow$ JA_1=JA_2 $\rightarrow$ Cambridge/Aachen algorithm (distance d_{ij} based on angular separation).

In our analysis we use $n_{\mathrm{sj}} = 2$ and $R_2 = R_1/2$.

Subjet filtering strategy

We would like a to have a jet algorithm which keeps most perturbative radiation while *filtering away* most of the UE/PileUp noise $\rightarrow$

1. Cluster **all the particles in the event** with a given jet definition (JA_1,R_1).
2. Take **each of the jets** of the event and **cluster its constituents** only with another jet definition (JA_2,R_2) with $R_2 < R_1$. This operation will provide us with a **set of subjets** of the original jet.
3. **Keep the n_{sj} subjets of a jet with largest** p_T and **throw way the remaining subjets**.
4. Original jets are replaced merging the selected subjets

This procedures defines the subjet filtering strategy.
Natural choice $\rightarrow$ JA_1=JA_2 $\rightarrow$ Cambridge/Aachen algorithm (distance d_{ij} based on angular separation).

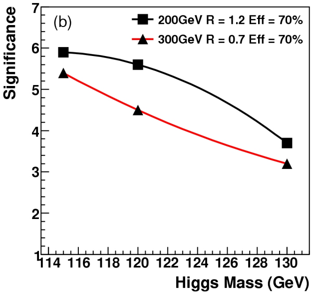
In our analysis we use $n_{\mathrm{sj}} = 2$ and $R_2 = R_1/2$.

Subjet filtering strategy

Does the subjet filtering really improve jet finding?
In J. Butterworth et al., (arXiv:0802.2470 [hep-ph]), it was shown that using a SJA-type algorithm, the Higgs boson search channels $Z/WH(\to b\bar{b})$ at $M_H \sim 120$ GeV transform from background-swamped channels to promising discovery channels.

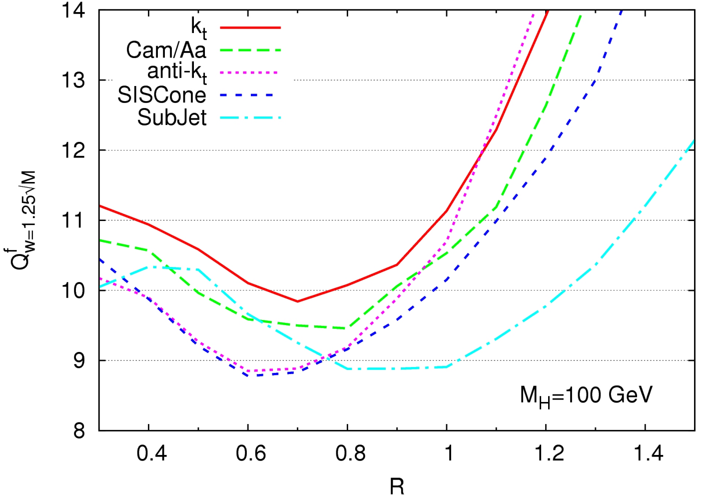
Jet definition	σ_S/fb	σ_B/fb	$S/\sqrt{B\cdot \text{fb}}$
C/A, $R = 1.2$, MD-F	0.57	0.51	0.80
$K_\perp$, $R = 1.0$, y_{cut}	0.19	0.74	0.22
SISCone, $R = 0.8$	0.49	1.33	0.42

TABLE I: Cross section for signal and the Z+jets background in the leptonic Z channel for $200 < p_{TZ}$/GeV < 600 and $110 < m_J$/GeV < 125, with perfect b-tagging; shown for our jet definition, and other standard ones at near optimal R values.

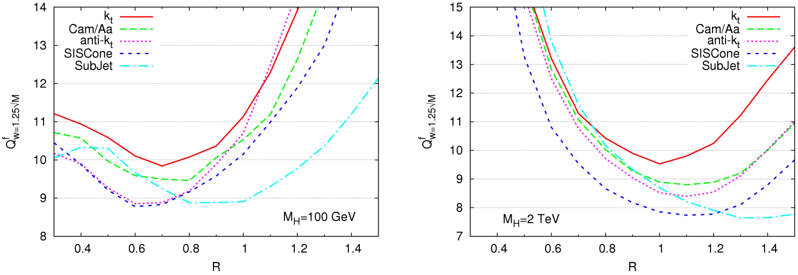


Motivation for study of the subjet filtering in a general context.

The performance of jet algorithms - Narrow $H \to gg$



The performance of jet algorithms - Narrow $H \to gg$



Less favored choices for the $M_H = 2$ TeV case:

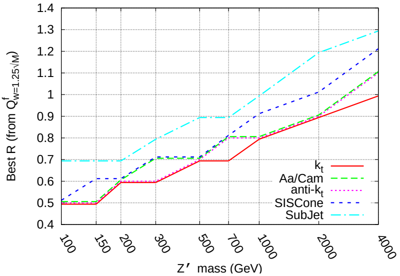
1. Use SISCone, but $R_{\rm best}^{100\ \rm GeV} = 0.6$ instead of $R_{\rm best}^{2\ \rm TeV} = 1.1 \to \rho_{\mathcal{L}} \sim 0.55$
2. Use $R_{\rm best}^{2\ \rm TeV}$, choose not SISCone, SubJet/Filtering but $k_T \to \rho_{\mathcal{L}} \sim 0.6$

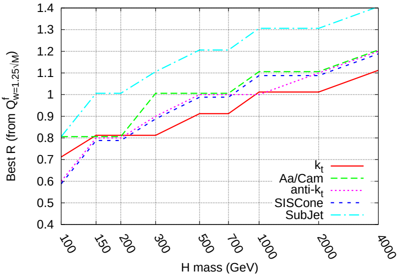
In both cases $\to$ Lose almost half effective discriminating power $\Sigma^{\rm eff}$!

The performance of jet algorithms - $R_{\rm best}$

Gluon jets require larger values of $R_{\rm best}$ than quark jets.
For quark jets with $p_T \geq 250$ GeV $\rightarrow$ $R_{\rm best} \geq 0.7$
For gluon jets with $p_T \geq 250$ GeV $\rightarrow$ $R_{\rm best} \geq 0.9$
Good jet resolution requires large values of R

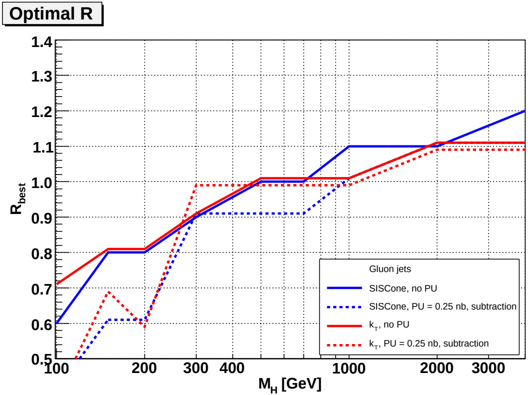
Approximate scaling $R_{\rm best} \sim \ln M$ (pert. radiation)





The PileUp case

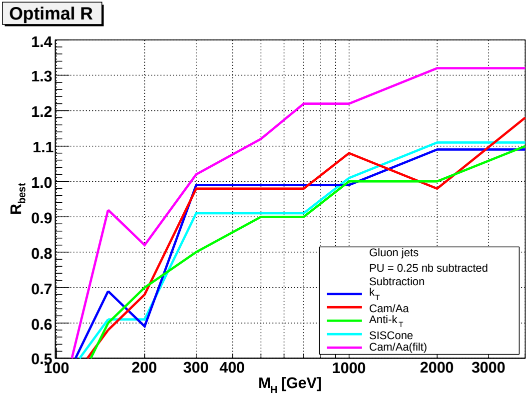
Check robustness of results when PileUp present ($\mathcal{L}_{\rm high} = 0.25 \mathrm{mb}^{-1}$)



After subtraction $R_{\rm best}^{\rm PU} \sim R_{\rm best}^{\rm no\ PU}$ for $p_T^{\rm jet} \geq 150$ GeV

The PileUp case

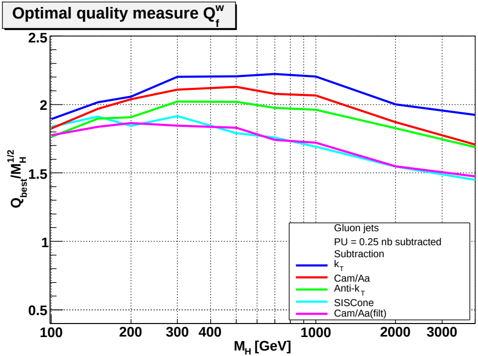
Check robustness of results when PileUp present ($\mathcal{L}_{\rm high} = 0.25 {\rm mb}^{-1}$)



For $p_T^{\rm jet} \geq 250$ GeV jets $R_{\rm best}^{\rm PU} \geq 0.9$ also in the presence of high lumi PU
Cam/Aa(filt) has $R_{{\rm best} \geq 1}$ due to its reduced jet area

The PileUp case

Check robustness of results when PileUp present ($\mathcal{L}_{\rm high} = 0.25 \mathrm{mb}^{-1}$)



- Same hierarchy of jet algorithms that in the no PU case
- SISCone and Cam/Aa(filt) are the best jet algorithms for all values of the jet p_T ($p_T^{\rm jet} \sim M_H/2$)

JET RECONSTRUCTION IN HEAVY ION COLLISIONS

References:

M. Cacciari, G. Salam, arXiv:0707.1378

N. Armesto, M. Cacciari, J. Rojo, G. Salam, C. Salgado, G. Soyez, work in progress

Jets in heavy ion collisions

Jets will be of paramount importance to fully exploit the potential of the HIC program at the LHC

- Jets will be most abundant hard probes in HIC at the LHC
  From CMS HIC TDR (J. Phys. G: Nucl. Part. Phys. 34 2307)

Table 1.1. The expected yield of several hard probes in 10^6 s PbPb and pPb LHC runs

	PbPb $\sqrt{s_{NN}} = 5.5$ TeV $\mathcal{L} = 5 \times 10^{26}$ cm^{-2} s^{-1}		pPb $\sqrt{s_{NN}} = 8.8$ TeV $\mathcal{L} = 1.4 \times 10^{30}$ cm^{-2} s^{-1}	
Process	Yield/10^6 s	Ref.	Yield/10^6 s	Ref.
	$\lvert\eta\rvert \leqslant 2.4$			
jet ($p_T > 50$ GeV/c)	2.2×10^7	[47]	1.5×10^{10}	[48]
jet ($p_T > 250$ GeV/c)	2.2×10^3	[47]	5.2×10^6	[48]
Z^0	3.2×10^5	[49]	6.8×10^6	[48]

Jets in heavy ion collisions

Jets will be of paramount importance to fully exploit the potential of the HIC program at the LHC

- Jets will be most abundant hard probes in HIC at the LHC
- Jets free of inclusive particle measurements biases
- A solid pQCD baseline is required to detect and quantify medium effects
- Need to asses to which extend can reconstructed QCD jets be disentangled from background, and which is the minimum size of medium effects which could then be probed
- All the successful jet technology from *pp* can be transferred to the extremely dense environment of heavy ion collisions

Event characterization

Wide range of predictions for UE in HIC at the LHC

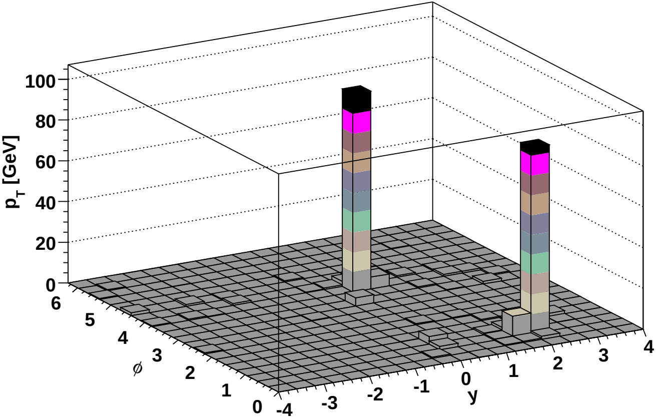
Process	$\langle N_{\rm particles} \rangle$	$\left\langle \frac{dN}{d\eta}\Big\vert_{\eta=0} \right\rangle$	$\left\langle \frac{dN_{\rm ch}}{d\eta}\Big\vert_{\eta=0} \right\rangle$	$\left\langle \rho_{(\eta,\phi)=(0,0)} \right\rangle$	T [s]
$pp \to gg$	160	30	15	0.5 GeV	$2 \cdot 10^{-4}$
$pp \to gg(+\mathrm{PbPb/SH})$	$4.7 \cdot 10^4$	5350	3020	450 GeV	1.2
$pp \to gg(+\mathrm{PbPb/NSH})$	$2.7 \cdot 10^4$	2230	1230	150 GeV	0.2

Note that estimations for background density $\rho \in (100 - 500)$ GeV satisfy tipically $\rho \geq p_T^{\rm jet}$ $\longrightarrow$ New regime for jet finding

Note also clustering times (without cuts in p_T)

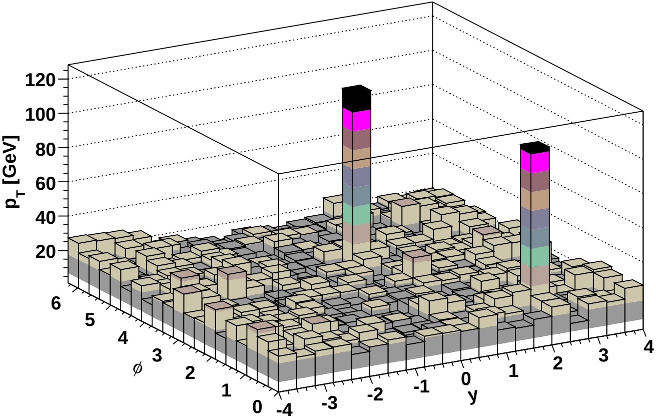
A typical dijet event in HIC at LHC

$pp \to gg$ events with $p_T^{\text{jet}} \sim 100$ GeV and $R = 0.4$ - No PbPb



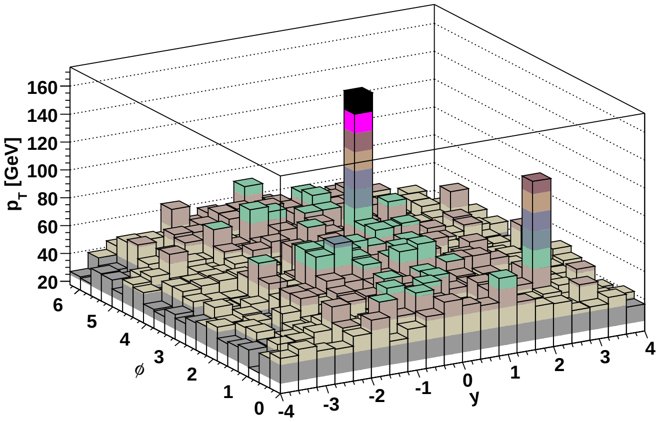
A typical dijet event in HIC at LHC

$pp \longrightarrow gg$ events with $p_T^{\mathrm{jet}} \sim 100$ GeV and $R = 0.4$ - PbPb model NSH



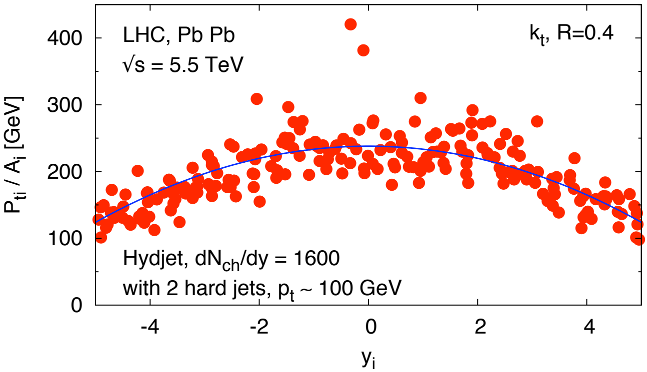
A typical dijet event in HIC at LHC

$pp \longrightarrow gg$ events with $p_T^{\rm jet} \sim 100$ GeV and $R = 0.4$ - PbPb model SH



Background subtraction

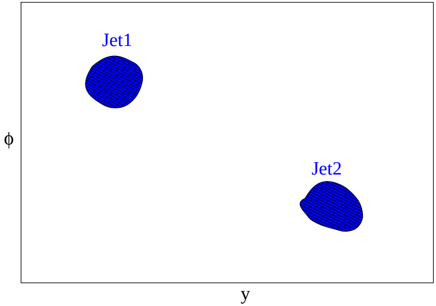
The large UE in HIC can be measured and subtracted with similar techniques as in the pp case
Caveat: background level depends on jet position in (η, ϕ) plane
First option: Extract parabolic dependence $\rho(y) = \rho_0 + \rho_1 y^2$



Background subtraction

The large UE in HIC can be measured and subtracted with similar techniques as in the pp case

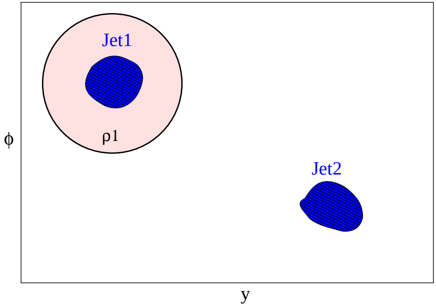
Second option: Circular range of $D = kR$ centered on jet axis (reduce sensitivity to background structure details)



Background subtraction

The large UE in HIC can be measured and subtracted with similar techniques as in the pp case

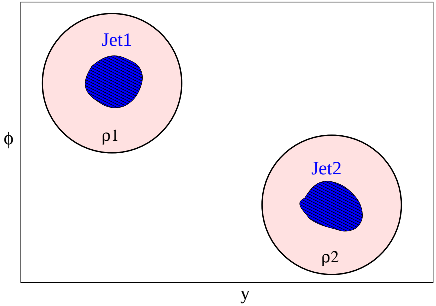
Second option: Circular range of $D = kR$ centered on jet axis (reduce sensitivity to background structure details)



Background subtraction

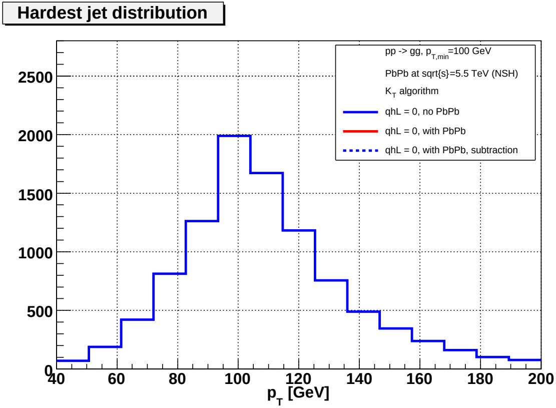
The large UE in HIC can be measured and subtracted with similar techniques as in the pp case

Second option: Circular range of $D = kR$ centered on jet axis (reduce sensitivity to background structure details)



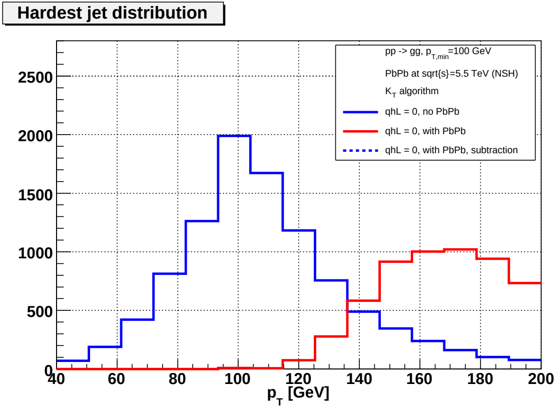
Background subtraction in practice

Example: Generate Pythia events with $p_T^{\rm min} = 100$ GeV
Hardest jet distribution from k_T jet algorithm and $R = 0.4$



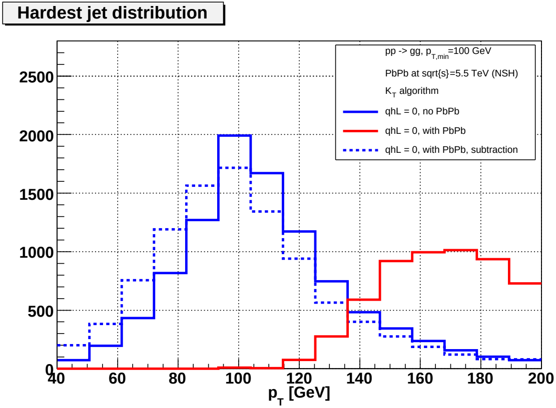
Background subtraction in practice

Example: Generate Pythia events with $p_T^{\min} = 100$ GeV
Hardest jet distribution from k_T jet algorithm and $R = 0.4$



Background subtraction in practice

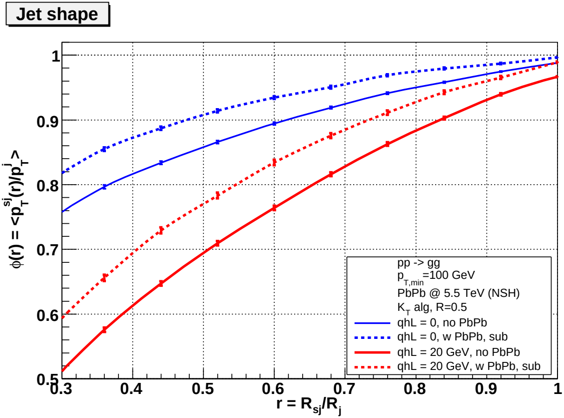
Example: Generate Pythia events with $p_T^{\rm min} = 100$ GeV
Hardest jet distribution from k_T jet algorithm and $R = 0.4$



Jets down to $p_T \sim 50$ GeV can be reconstructed even in worst HIC scenario

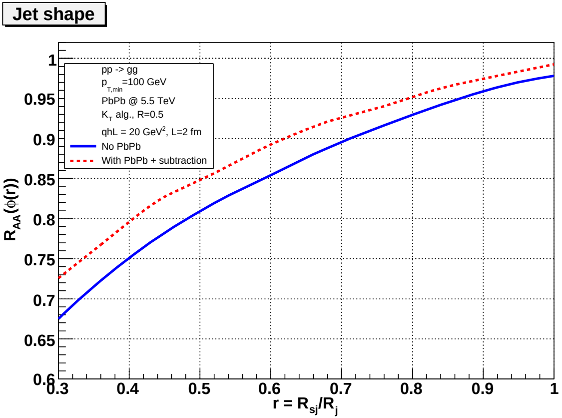
Jet shape

Jet substructure $\phi(r)$ useful discriminator of medium effects
Cluster jet constituents with $R_{\rm sj}$ ($r \equiv R_{\rm sj}/R_{\rm jet} < 1$) and keep hardest subjet with $p_T^{\rm sj}(\leq p_T^{\rm jet})$. $R_{\rm jet} = 0.5$, $0.15 \leq R_{\rm sj} \leq 0.5$



Jet shape

$R_{AA}(\phi(r)) \equiv \phi_{\rm med}(r)/\phi_{\rm vac}(r)$, with $\phi_{\rm vac}(r)$ from pp jets

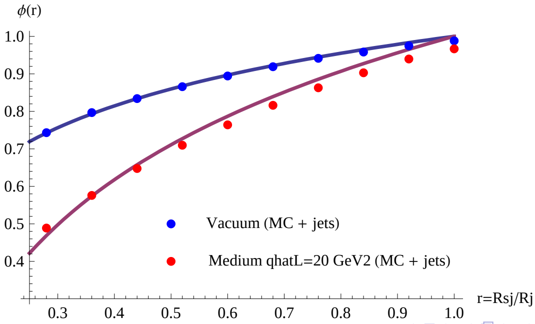


Jet shape - LL pQCD

If medium effects parametrized by $(1+f_{\rm med})$ in the singular part of the splitting functions (Borghini et al. 05) then:

$$\phi_{\rm med}(r, f_{\rm med}) = 1 - \frac{\alpha_s}{\pi} \ln \frac{R_{\rm j}}{R_{\rm sj}} \left[C_A \left(\frac{3 f_{\rm med}}{8} + 2\ln 2 - \frac{43}{96} \right) + \frac{7 N_f T_R}{48} \right]$$

for $\alpha_s = 0.2$ and $f_{\rm med} \sim 3$ $[\phi_{\rm vac}(r) = \phi_{\rm med}(r, f_{\rm med}=0)] \rightarrow$ Agreement with $\phi(r)$ results from MC simulations + subjets ($L = 2$ fm, $\hat{q}L = 20$ GeV^2)



Conclusions and outlook

- Sizable progress in jet algorithms in the recent years
- Area-based method provide an alternative strategy for measurement/subtraction of Underlying Event and Pile-Up
- It is necessary now to quantify the performance of jet algorithms and define optimal choices
- Modern jet finding technology has also important potential applications in heavy ion collisions
- Still room for further improvements, both in theory and in experiment: close collaboration essential to fully exploit the LHC potential!

Thanks for your attention!

Conclusions and outlook

- Sizable progress in jet algorithms in the recent years
- Area-based method provide an alternative strategy for measurement/subtraction of Underlying Event and Pile-Up
- It is necessary now to quantify the performance of jet algorithms and define optimal choices
- Modern jet finding technology has also important potential applications in heavy ion collisions
- Still room for further improvements, both in theory and in experiment: close collaboration essential to fully exploit the LHC potential!

Thanks for your attention!

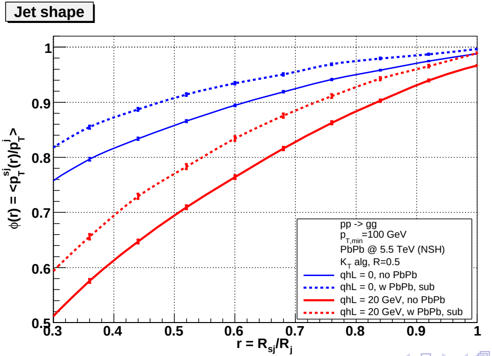
ADDITIONAL MATERIAL

The Anti-k_T algorithm

The Anti-k_T algorithm has a very reduced sensitivity to Back-Reaction (effect of UE particles in jet clustering):

$$\Delta p_T^{\mathrm{BR}}\Big|_{k_T} \sim 5\mathrm{GeV} \ , \qquad \Delta p_T^{\mathrm{BR}}\Big|_{\mathrm{Anti}-k_T} \sim 1\mathrm{GeV}$$

for $p_T^{\mathrm{jet}} \sim 100$ GeV, $R = 0.5$, $\rho \sim 150$ GeV.

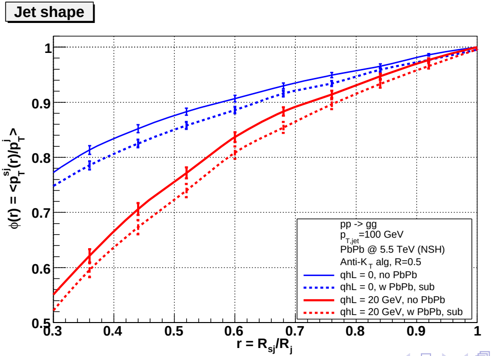


The Anti-k_T algorithm

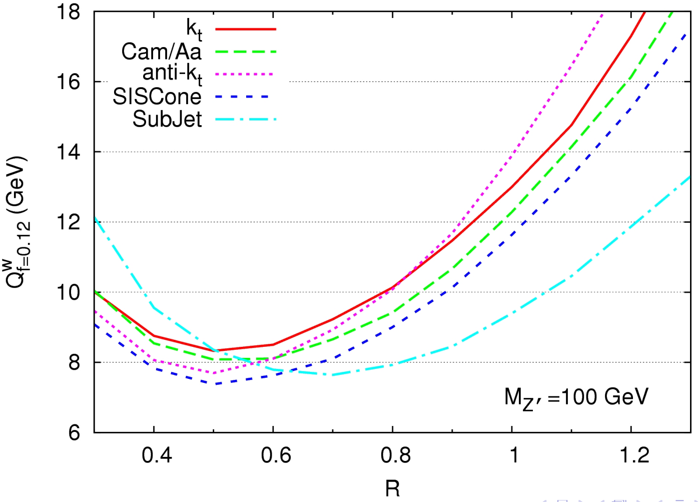
The Anti-k_T algorithm has a very reduced sensitivity to Back-Reaction (effect of UE particles in jet clustering):

$$\Delta p_T^{\mathrm{BR}}\Big|_{k_T} \sim 5\mathrm{GeV} \ , \qquad \Delta p_T^{\mathrm{BR}}\Big|_{\mathrm{Anti}-k_T} \sim 1\mathrm{GeV}$$

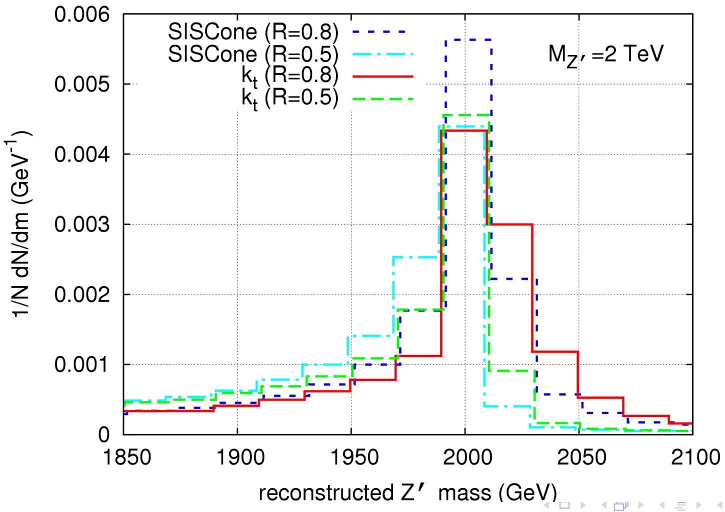
for $p_T^{\mathrm{jet}} \sim 100$ GeV, $R = 0.5$, $\rho \sim 150$ GeV.



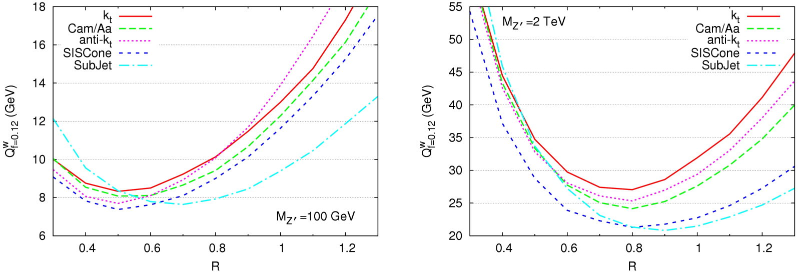
The performance of jet algorithms - Narrow $Z' \rightarrow q\bar{q}$



The performance of jet algorithms - Narrow $Z' \to q\bar{q}$



The performance of jet algorithms - Narrow $Z' \to q\bar{q}$



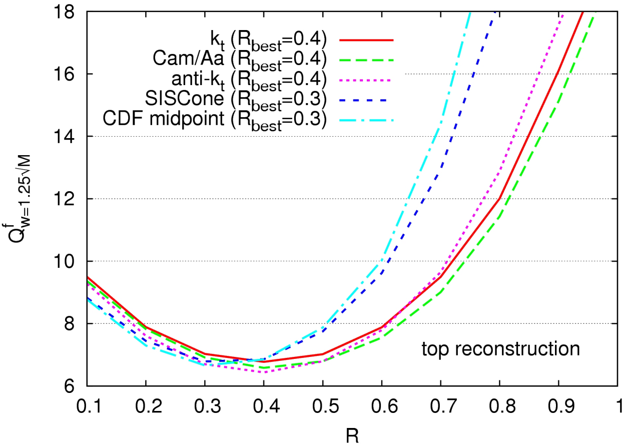
Less favored choices for the $M_{Z'} = 2$ TeV case:

1. Use SISCone, but $R_{\rm best}^{100\ \rm GeV} = 0.5$ instead of $R_{\rm best}^{2\ \rm TeV} = 0.8 \to \rho_{\mathcal{L}} \sim 0.7$
2. Use k_T instead of SISCone or SubJet/Filtering $\to \rho_{\mathcal{L}} \sim 0.8$

In both cases $\to$ Lose $\sim$ 25% of effective discrimination power $\Sigma^{\rm eff}$!

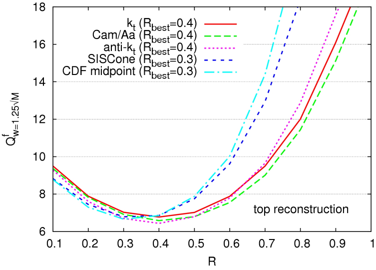
The performance of jet algorithms - Hadronic $t\bar{t}$

Probe jet algorithms performance in a complicated, multi-jet environment.



The performance of jet algorithms - Hadronic $t\bar{t}$

Probe jet algorithms performance in a complicated, multi-jet environment.



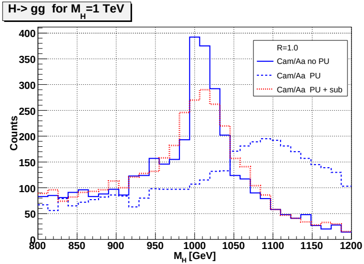
All jet algs. blue similar performance near $R_{\rm opt} \sim 0.3-0.4$
Use SISCone, but with $R=0.7$ instead of $R_{\rm best} \sim 0.4 \rightarrow \rightarrow \rho_{\mathcal{L}} \sim 0.5$
The effective lumi ratio $\rho_{\mathcal{L}}$ worsens very rapidly for large values of $R > R_{\rm best}$.

SJA with Pile-Up

The SJA leads to better signal discrimination and is less sensitive to Pile-Up than Cam/Aa:

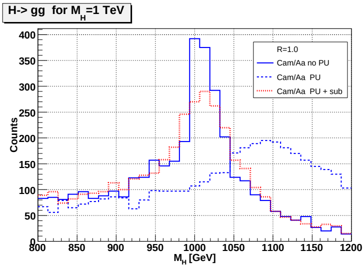
SJA with Pile-Up

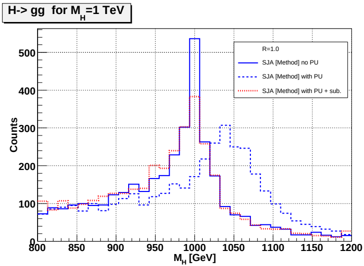
The SJA leads to better signal discrimination and is less sensitive to Pile-Up than Cam/Aa:



SJA with Pile-Up

The SJA leads to better signal discrimination and is less sensitive to Pile-Up than Cam/Aa:





Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta\phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)

3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1
k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta\phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)

3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1

k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta\phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)

3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1

k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta \phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)

3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1
k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta\phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)

3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1

k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta \phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)
3. If any particles are left, repeat from step 1.

One parameter: R (like cone radius), whose natural value is 1

k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

Sequential recombination algorithms

Example: the k_t algorithm:

1. Calculate (or update) distances between all particles i and j, and between i and beam:

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = k_{ti}^2\,, \qquad \Delta R_{ij}^2 = \Delta y_{ij}^2 + \Delta\phi_{ij}^2$$

2. Find smallest of d_{ij} and d_{iB}
   - If d_{ij} is smallest, recombine i and j (add result to particle list, remove i, j)
   - if d_{iB} is smallest call i a jet (remove it from list of particles)
3. If any particles are left, repeat from step 1.

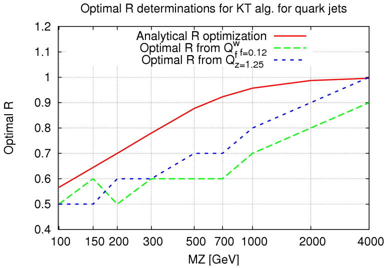
One parameter: R (like cone radius), whose natural value is 1
k_t algorithm attempts approximate inversion of the QCD shower branching process $\longrightarrow$ Theoretical sound basis.

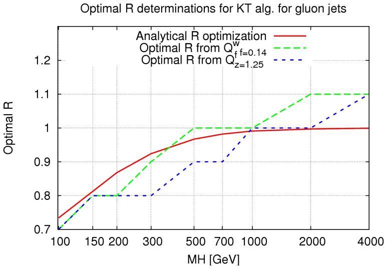
The performance of jet algorithms - $R_{\rm best}$

Compare the MC results with analytical estimations of $R_{\rm best}$ from arXiv:0712.3014

Better for gluons that for quarks

(*N.B.* analytical results in the small-R approximation)

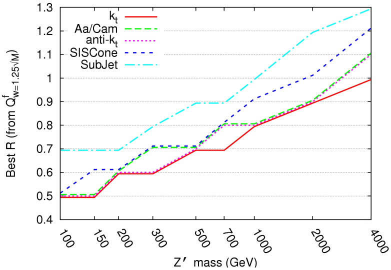




The performance of jet algorithms - $R_{\rm best}$

Compare $R_{\rm best}$ after subtraction of pile-up

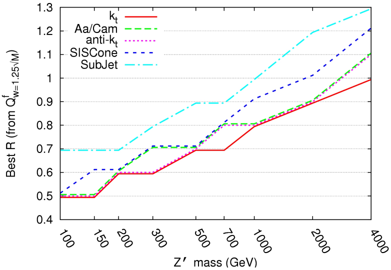
For low lumi (0.05 nb^{-1}) very small effects, at high lumi (0.25 nb^{-1}) one finds moderate effects $R_{\rm best}^{\rm PU} \sim R_{\rm best}^{\rm no\ PU}$.

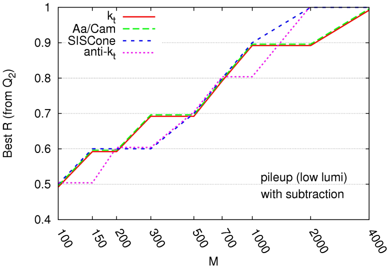


The performance of jet algorithms - R_{best}

Compare R_{best} after subtraction of pile-up

For low lumi (0.05 nb^{-1}) very small effects, at high lumi (0.25 nb^{-1}) one finds moderate effects $R_{\mathrm{best}}^{\mathrm{PU}} \sim R_{\mathrm{best}}^{\mathrm{no\ PU}}$.

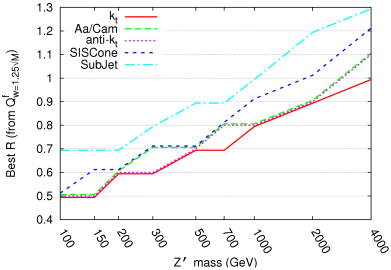


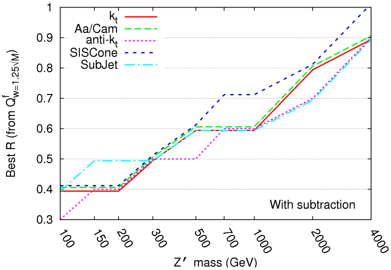


The performance of jet algorithms - $R_{\rm best}$

Compare $R_{\rm best}$ after subtraction of pile-up

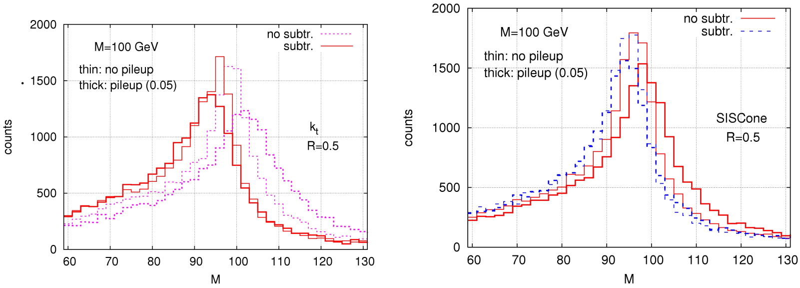
For low lumi (0.05 nb^{-1}) very small effects, at high lumi (0.25 nb^{-1}) one finds moderate effects $R_{\rm best}^{\rm PU} \sim R_{\rm best}^{\rm no\ PU}$.





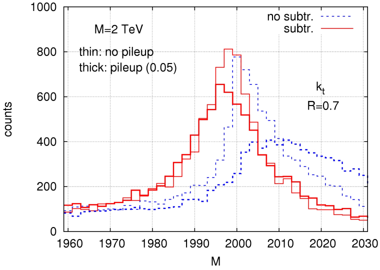
The performance of jet algorithms - Z'

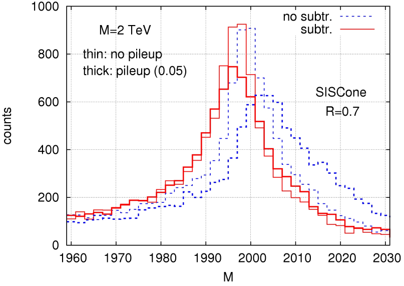
Pile-up (at high LHC lumi) needs to be subtracted for a meaningful comparison (Use `FastJet` area-based tools):



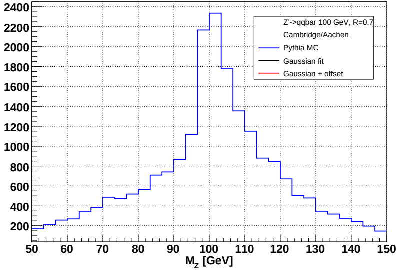
The performance of jet algorithms - Z'

Pile-up (at high LHC lumi) needs to be subtracted for a meaningful comparison (Use `FastJet` area-based tools):

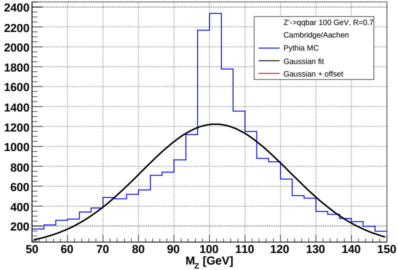




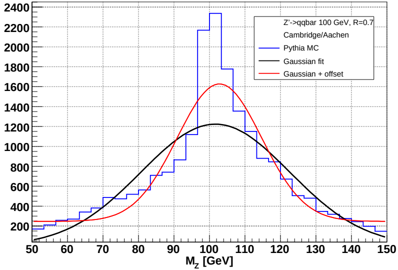
Fitting simple distributions



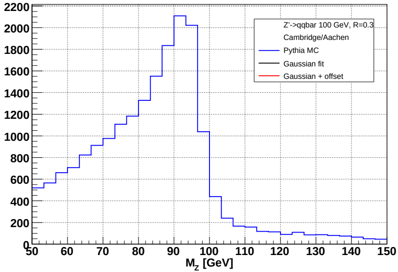
Fitting simple distributions



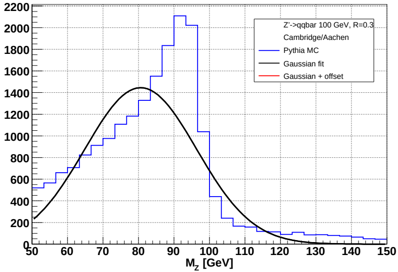
Fitting simple distributions



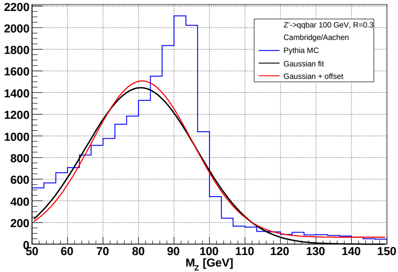
Fitting simple distributions



Fitting simple distributions



Fitting simple distributions



Fitting simple distributions