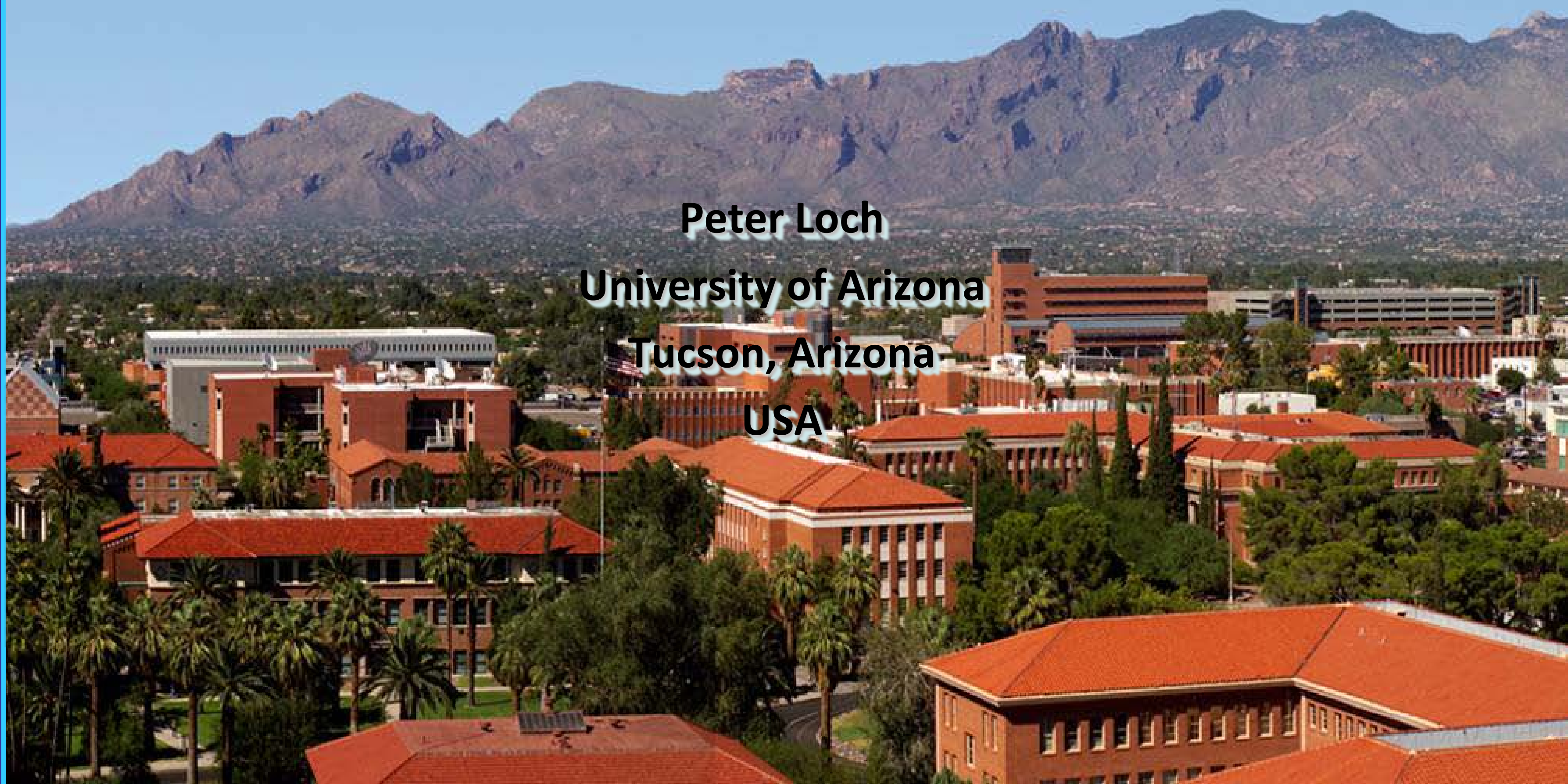
THE UNIVERSITY OF ARIZONA.

Introduction to Hadronic Final State Reconstruction in Collider Experiments (Part VI)



Peter Loch
University of Arizona
Tucson, Arizona
USA

THE UNIVERSITY OF ARIZONA.

Theoretical Requirements For Jet Finders

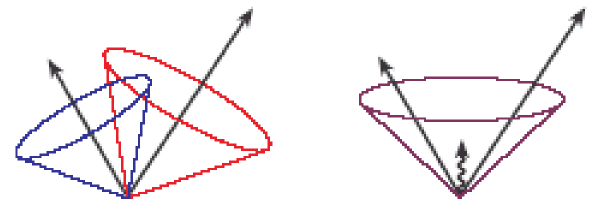
P. Loch
U of Arizona

Very important at LHC

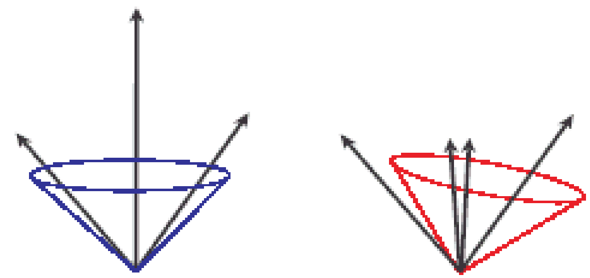
- Often LO (or even NLO) not sufficient to understand final states
  - Potentially significant K-factors can only be applied to jet driven spectra if jet finding follows theoretical rules
    - E.g., jet cross-section shapes
- Need to be able to compare experiments and theory
  - Comparison at the level of distributions
    - ATLAS and CMS will unfold experimental effects and limitations independently – different detector systems

Theoretical guidelines

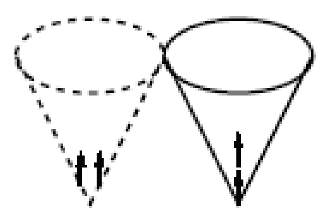
- Infrared safety
  - Adding or removing soft particles should not change the result of jet clustering
- Collinear safety
  - Splitting of large pT particle into two collinear particles should not affect the jet finding



infrared sensitivity
(soft gluon radiation merges jets)



collinear sensitivity (1)
(sensitive to E_t ordering of seeds)



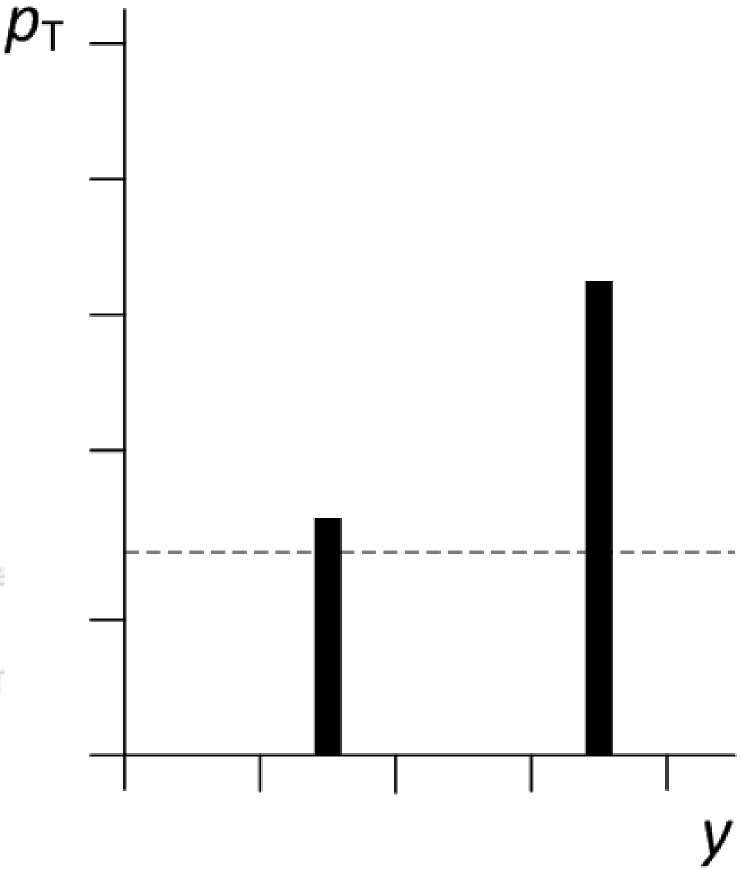
collinear sensitivity (2)
(signal split into two towers below threshold)

ATLAS

Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets



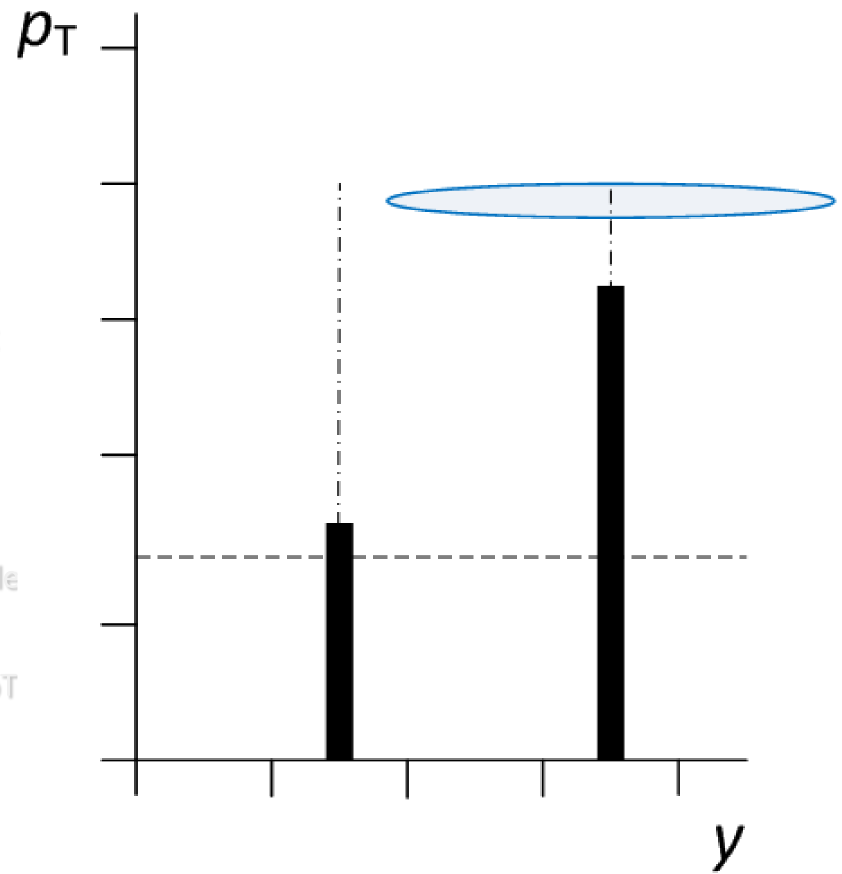
Iterative Seeded Fixed Cone

P. Loch
U of Arizona

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$



THE UNIVERSITY OF ARIZONA.

Iterative Seeded Fixed Cone

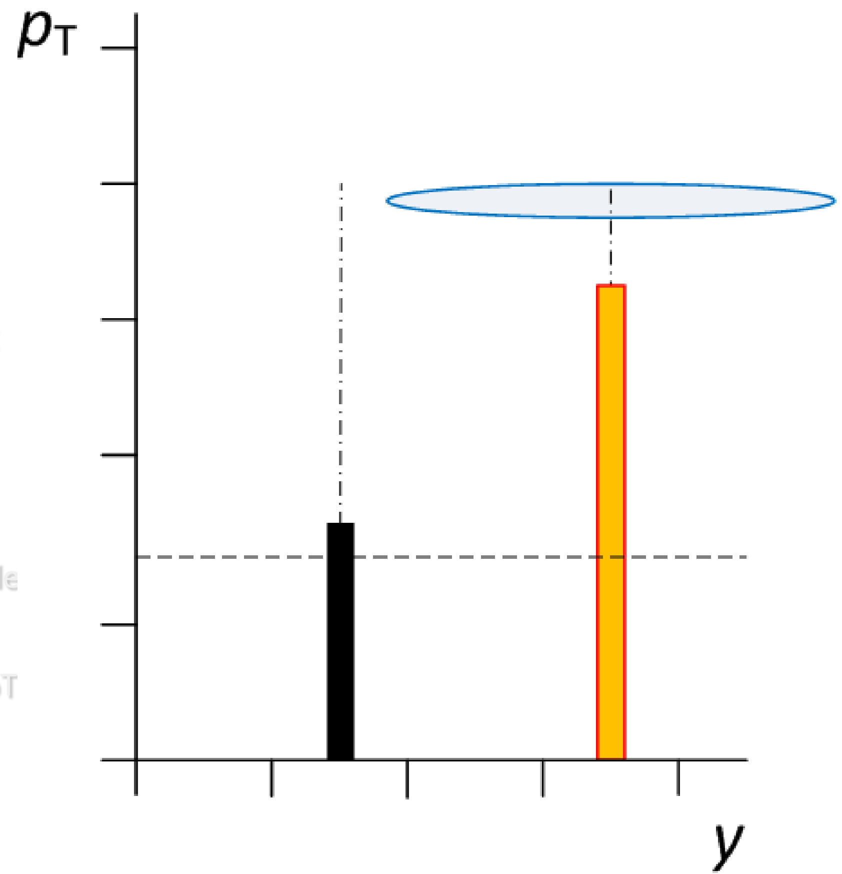
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(first protojet)



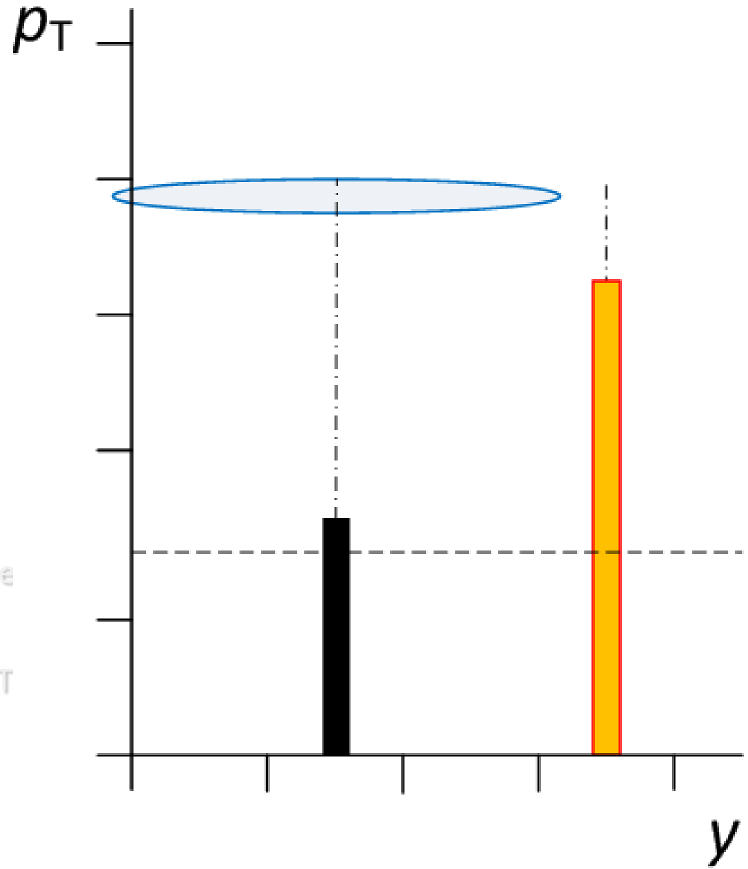
ATLAS

Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$



THE UNIVERSITY OF ARIZONA.

Iterative Seeded Fixed Cone

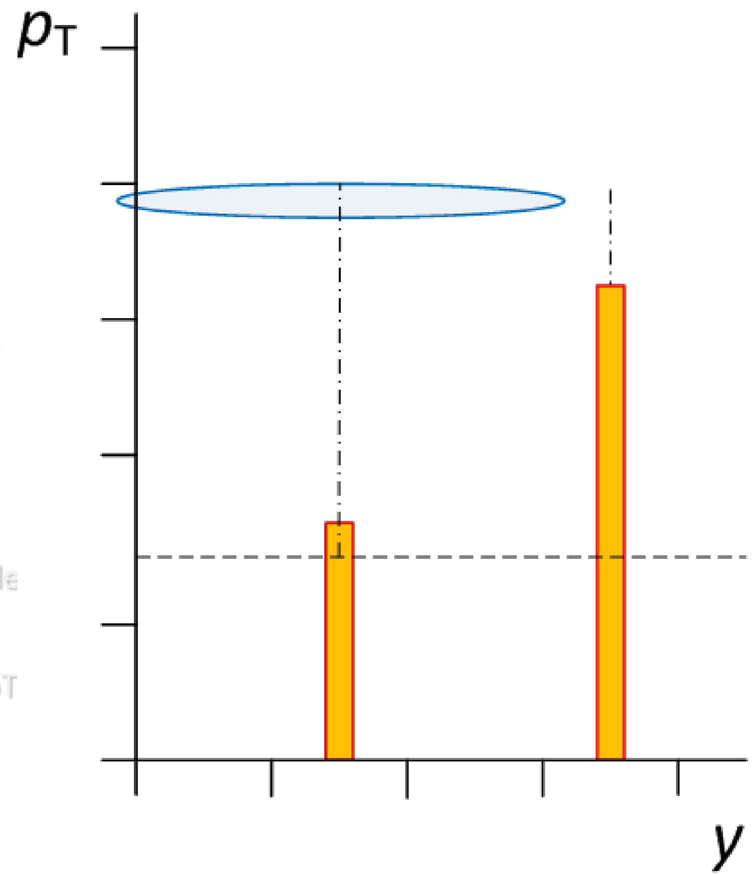
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(second protojet)



THE UNIVERSITY OF ARIZONA.

Iterative Seeded Fixed Cone

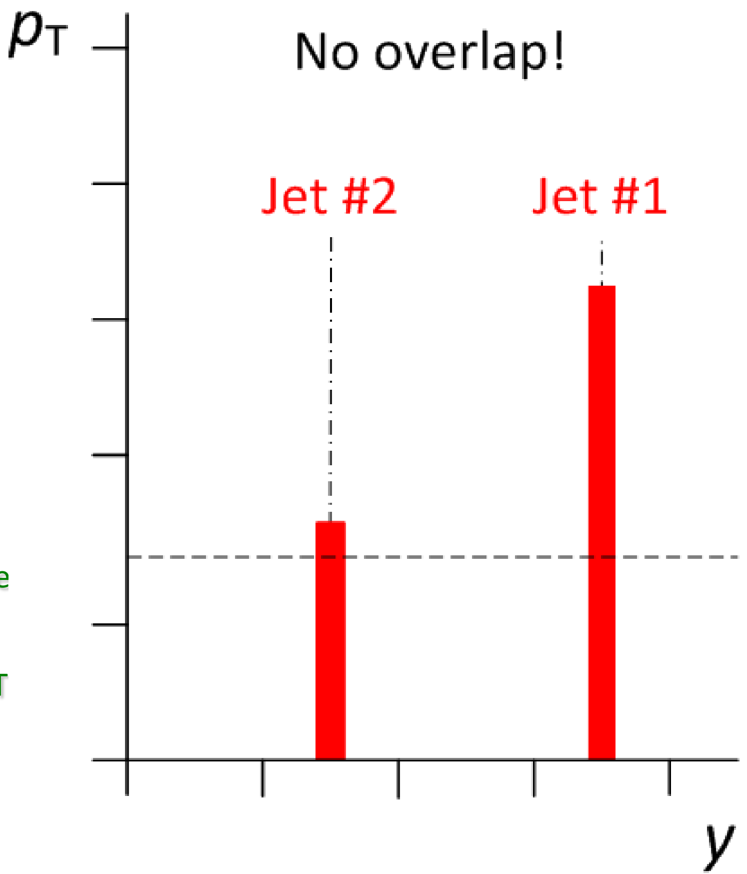
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (**merge**)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (**split**)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(two jets)



THE UNIVERSITY OF ARIZONA

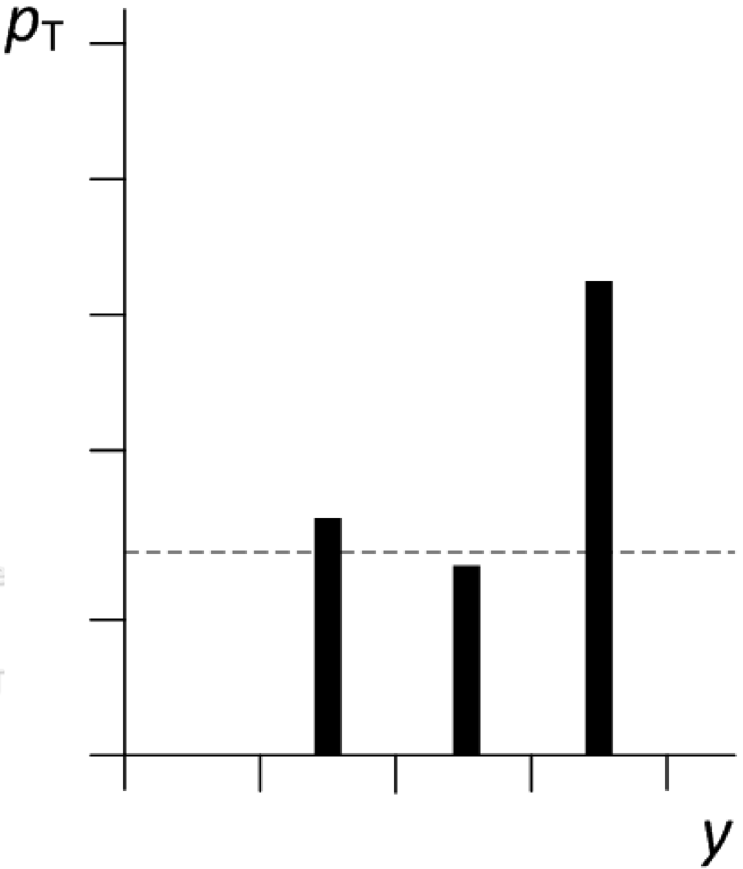
Iterative Seeded Fixed Cone

P. Loch
U of Arizona

Use following jet finder rules:

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets



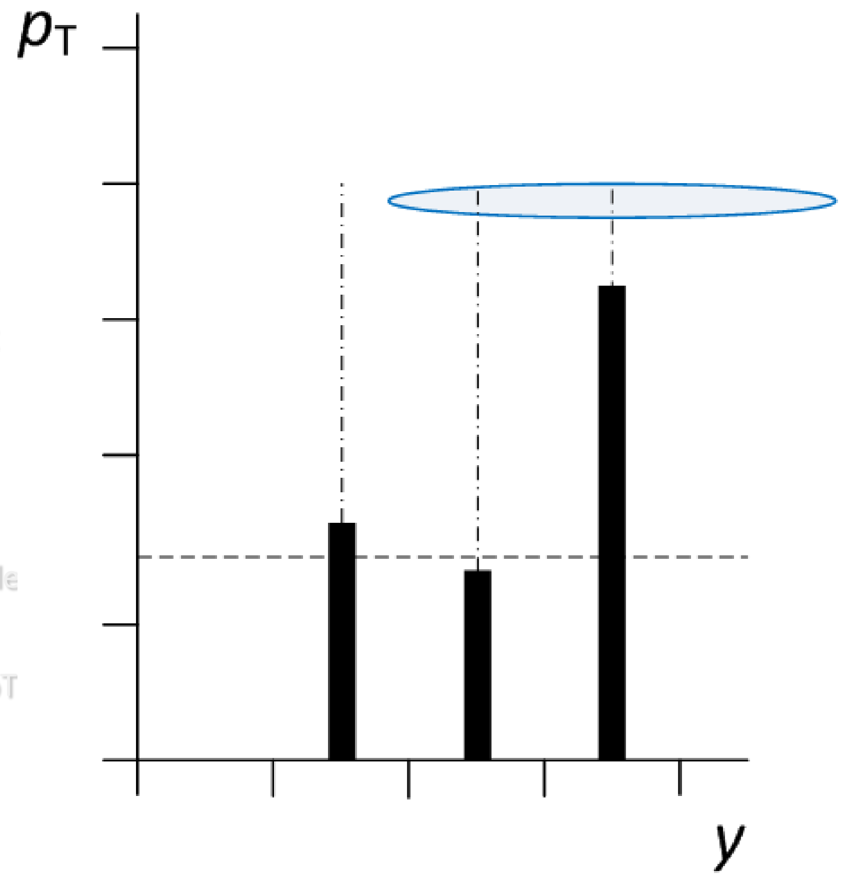
P. Loch
U of Arizona
March 09, 2010

Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\mathrm{cone}}$$

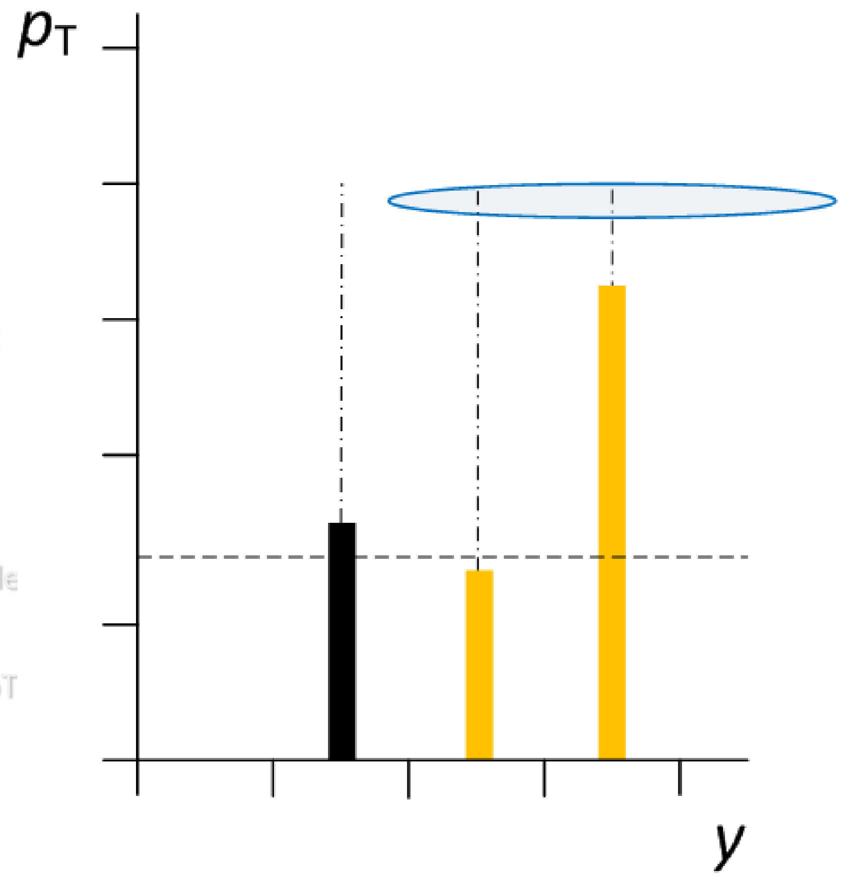


Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$



Iterative Seeded Fixed Cone

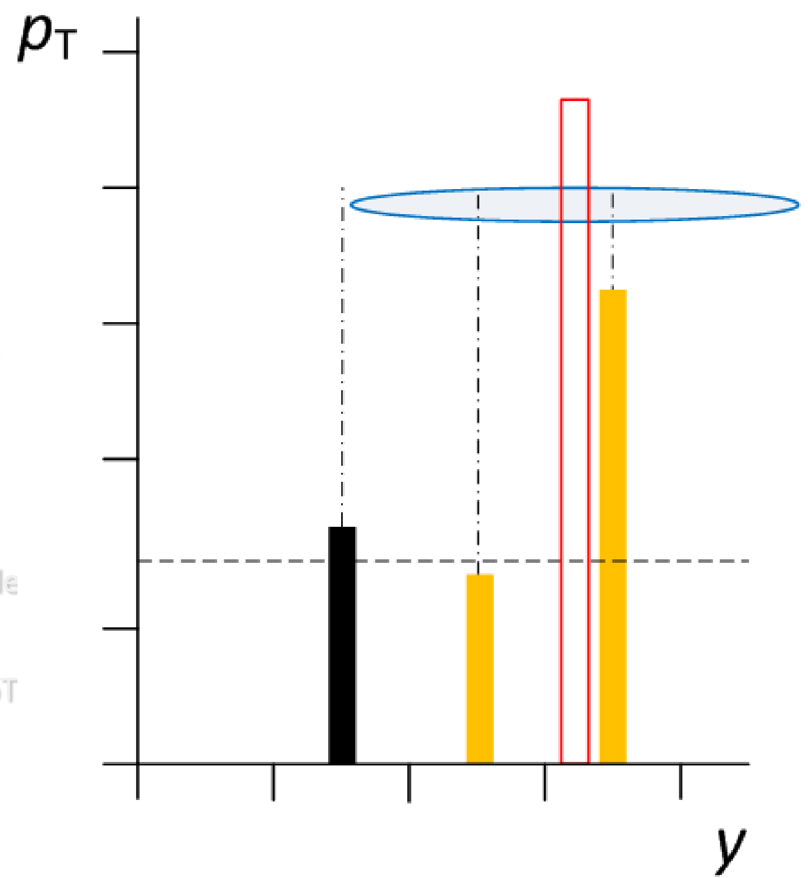
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(first protojet)



THE UNIVERSITY OF ARIZONA

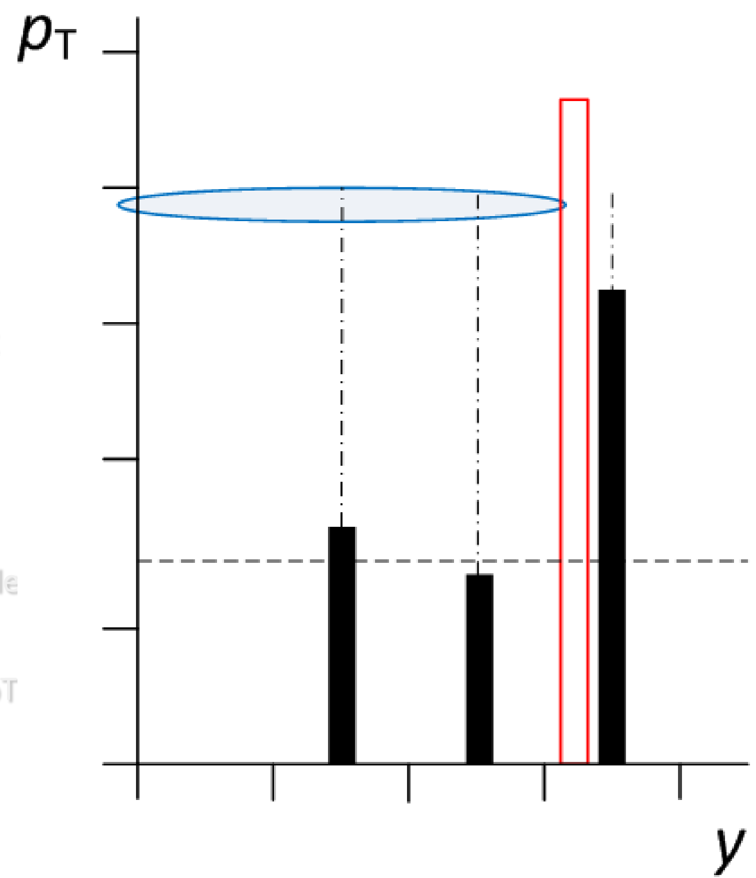
Iterative Seeded Fixed Cone

P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$



THE UNIVERSITY OF ARIZONA.

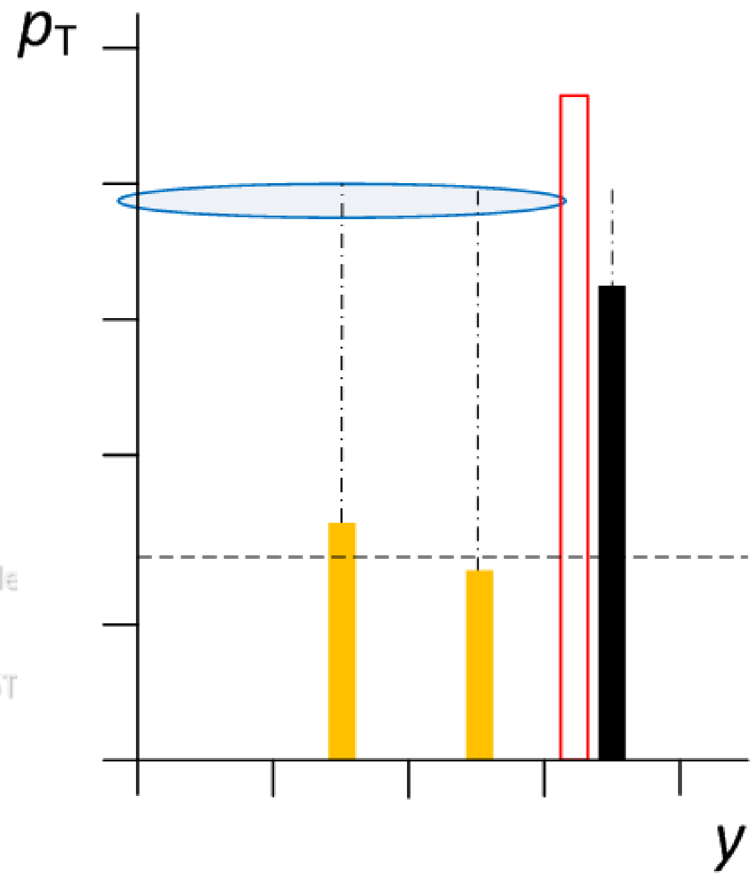
Iterative Seeded Fixed Cone

P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

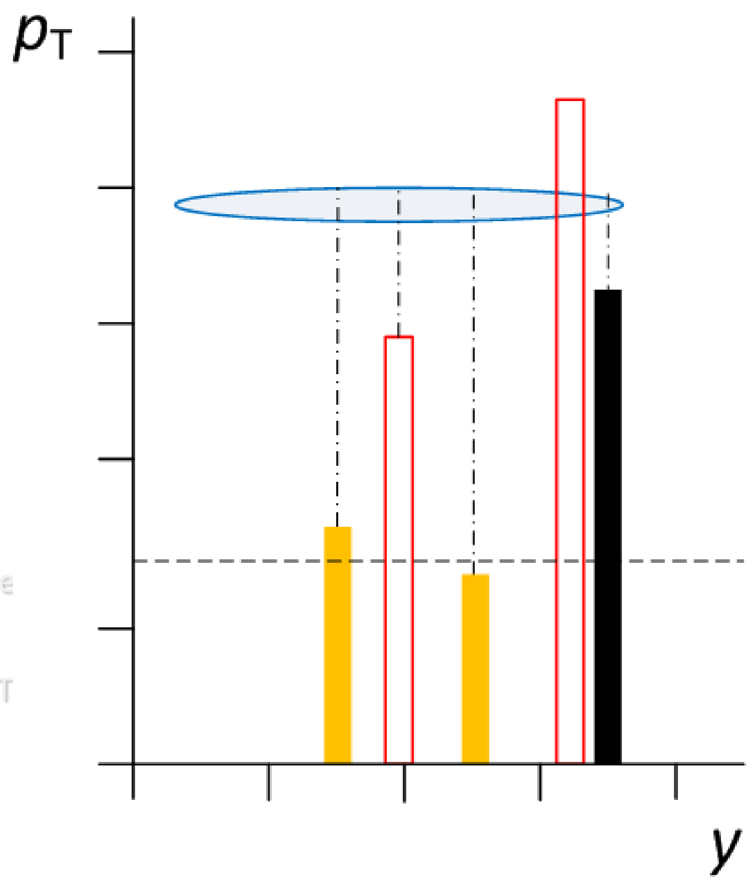


Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

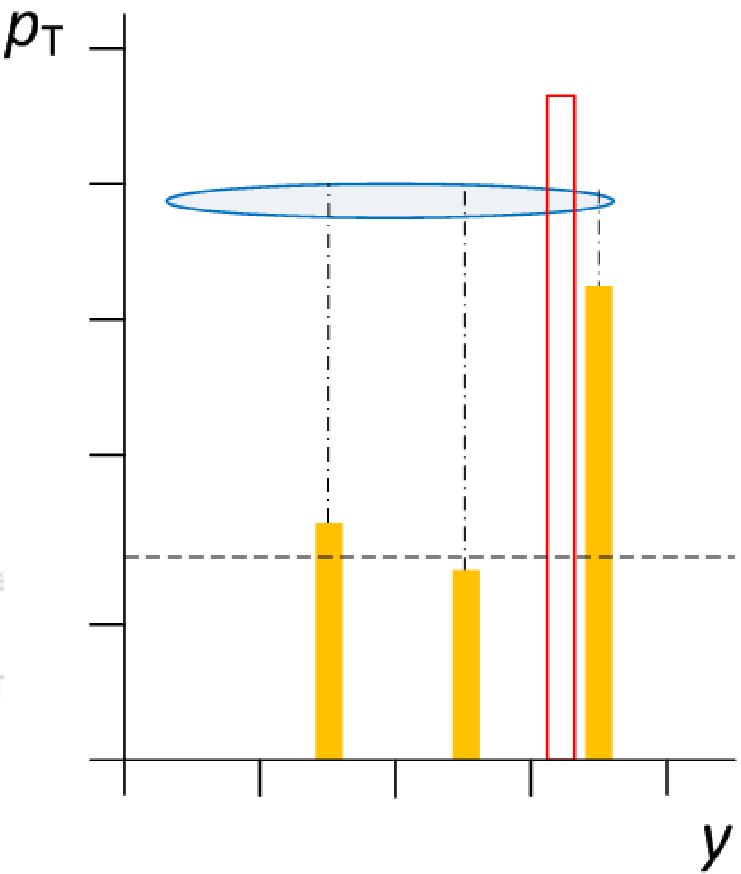


Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$



THE UNIVERSITY OF ARIZONA.

Iterative Seeded Fixed Cone

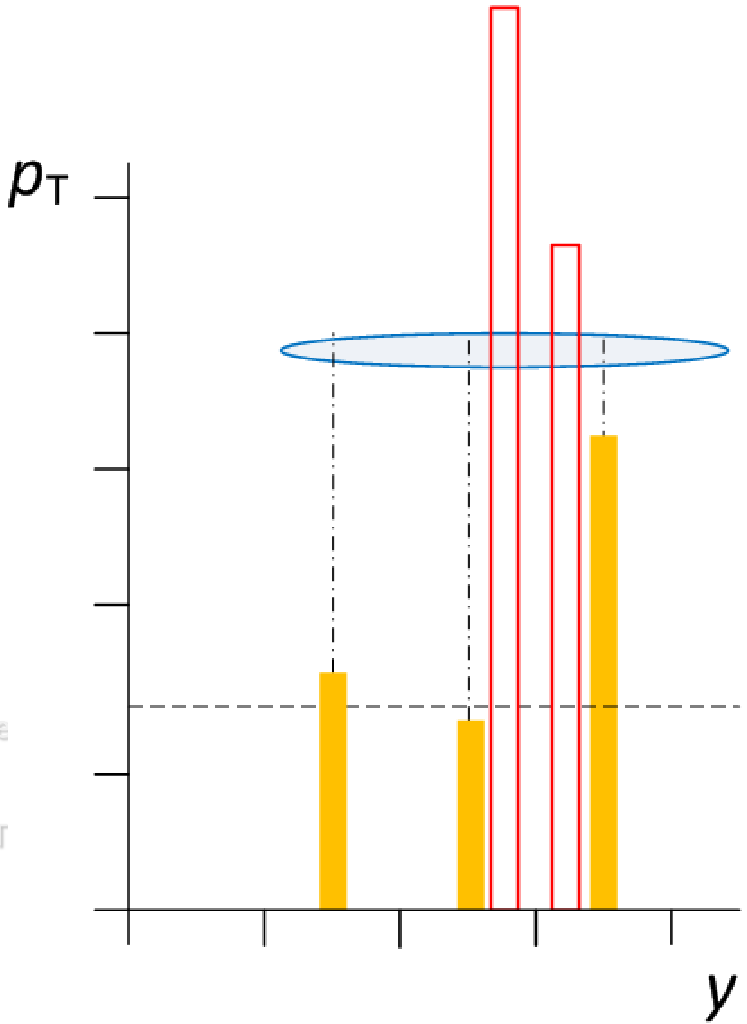
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (merge)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (split)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(second protojet)



ATLAS

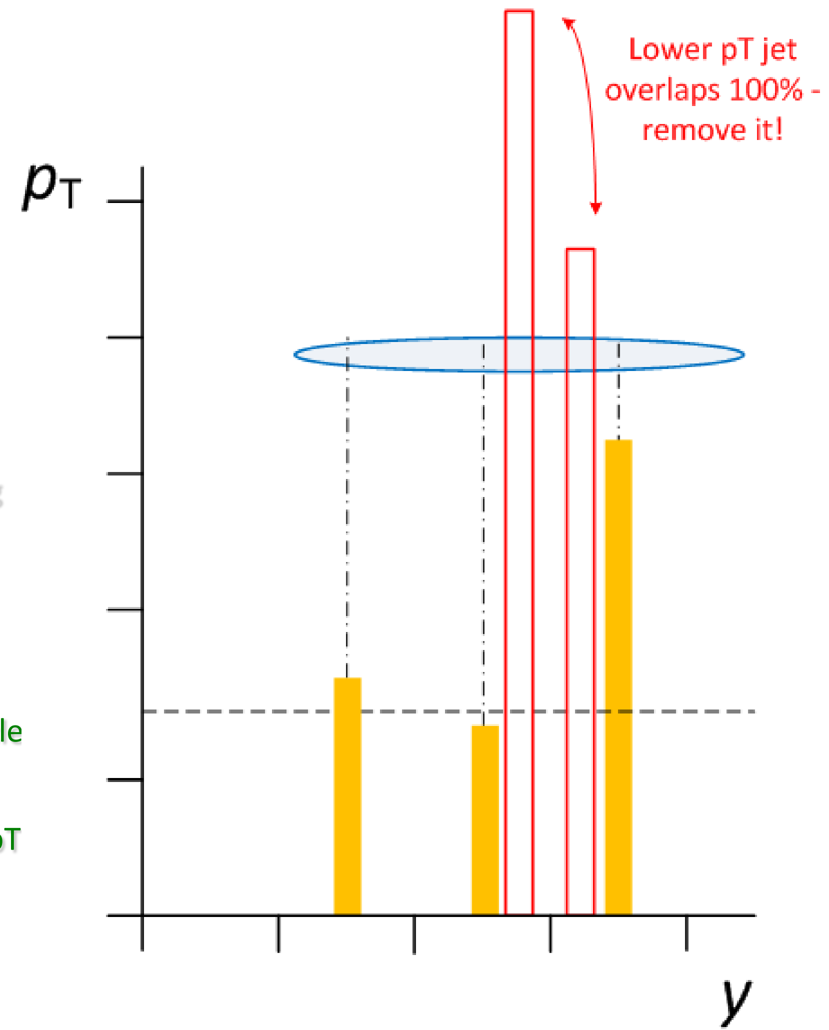
Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (**merge**)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (**split**)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(second protojet)



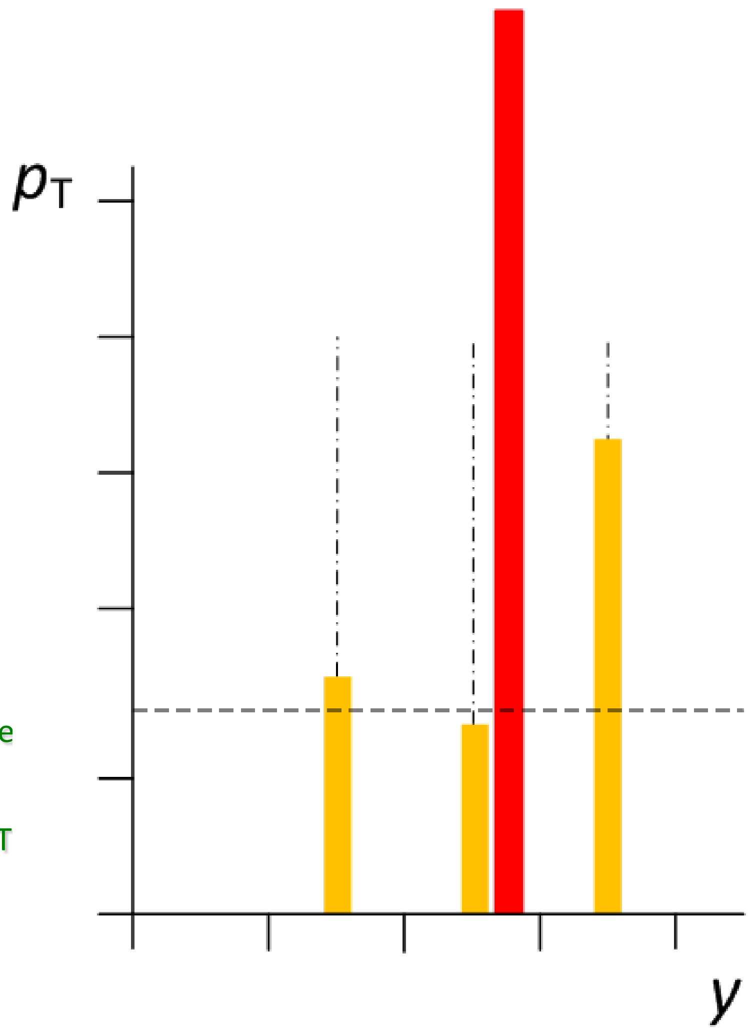
Iterative Seeded Fixed Cone

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum re-summation
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (**merge**)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (**split**)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(one jet)



THE UNIVERSITY OF ARIZONA.

Iterative Seeded Fixed Cone

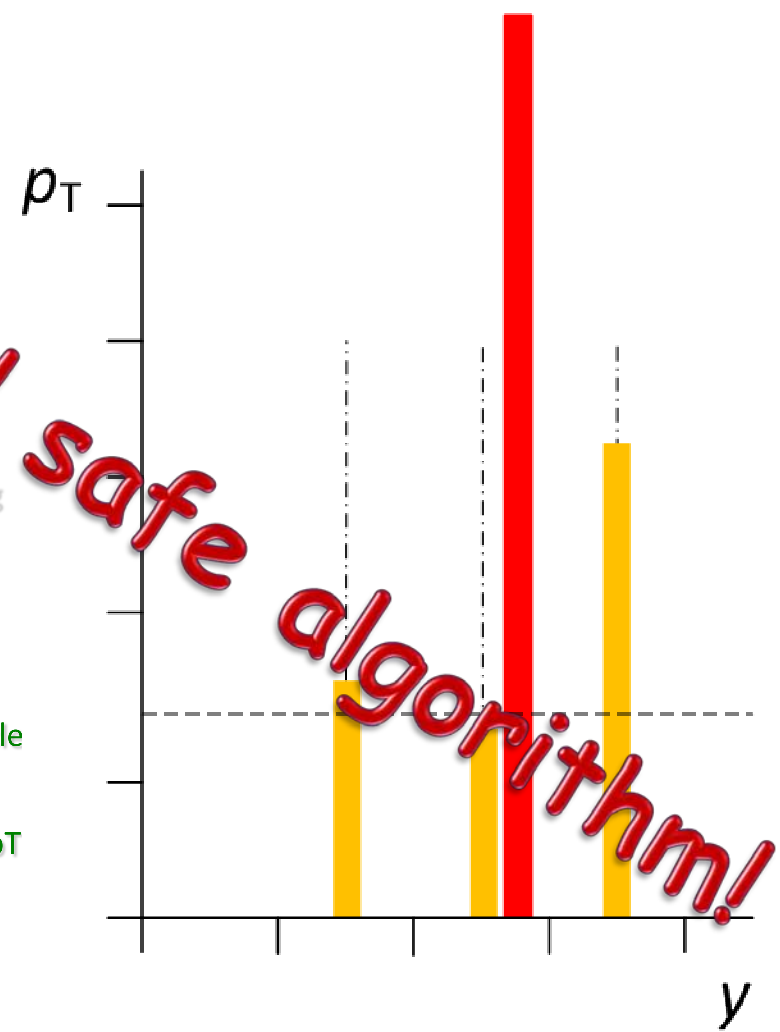
P. Loch
U of Arizona
March 09, 2010

Use following jet finder rules:

- Find particle with largest pT above a seed threshold
  - Create an ordered list of particles descending in pT and pick first particle
- Draw a cone of fixed size around this particle
  - Resolution parameter of algorithm
- Collect all other particles in cone and re-calculate cone directions from those
  - Use four-momentum recombination
- Collect particles in new cone of same size and find new direction as above
  - Repeat until direction does not change → cone becomes stable
- Take next particle from list if above pT seed threshold
- Repeat procedure and find next proto-jet
  - Note that this is done with all particles, including the ones found in previous cones
- Continue until no more proto-jets above threshold can be constructed
  - The same particle can be used by 2 or more jets
- Check for overlap between proto-jets
  - Add lower pT jet to higher pT jet if sum of particle pT in overlap is above a certain fraction of the lower pT jet (**merge**)
  - Else remove overlapping particles from higher pT jet and add to lower pT jet (**split**)
- All surviving proto-jets are the final jets

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\varphi^2} < R_{\text{cone}}$$

(one jet)

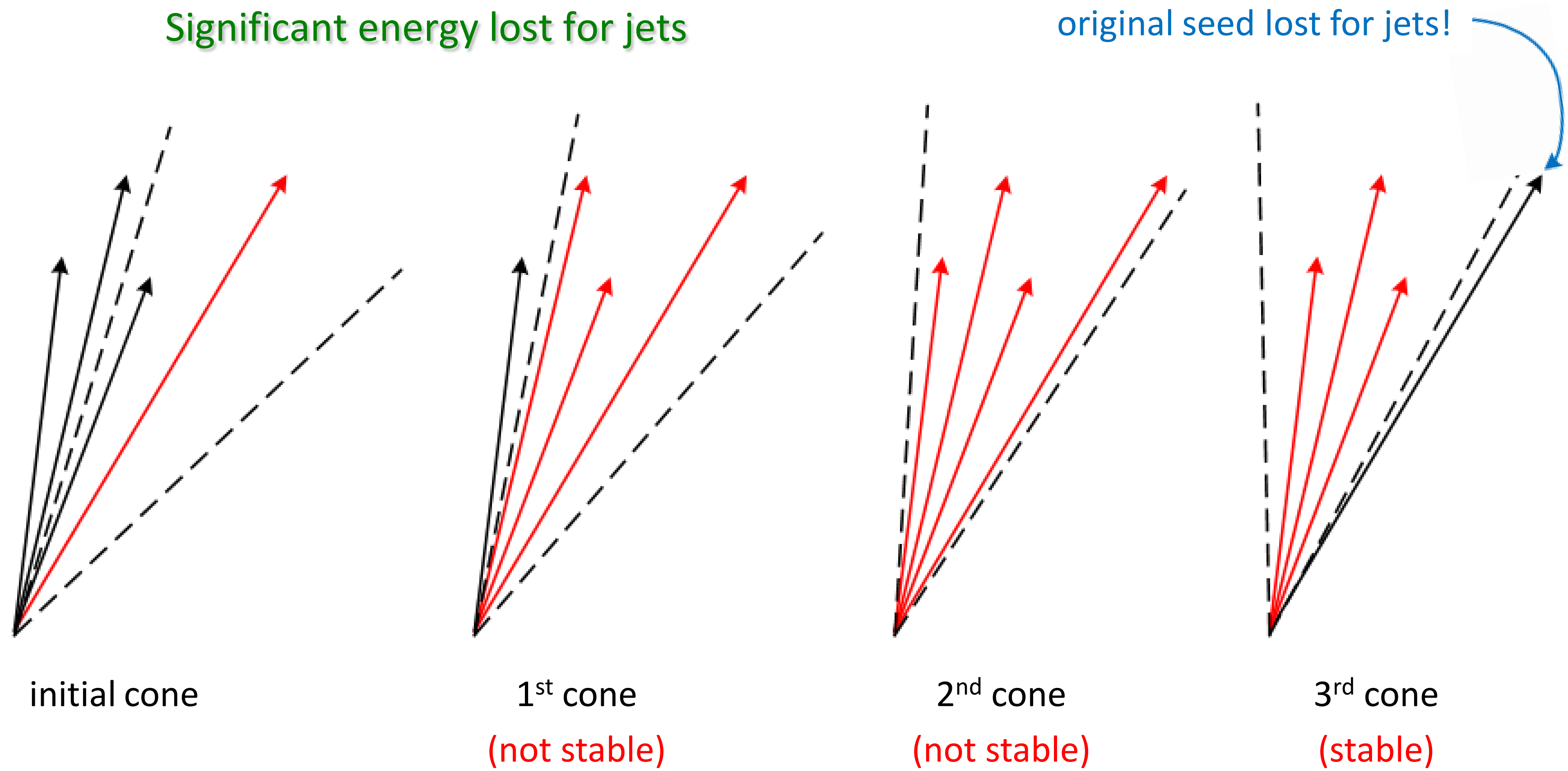


not an infrared safe algorithm!

Iterative Seeded Fixed Cone

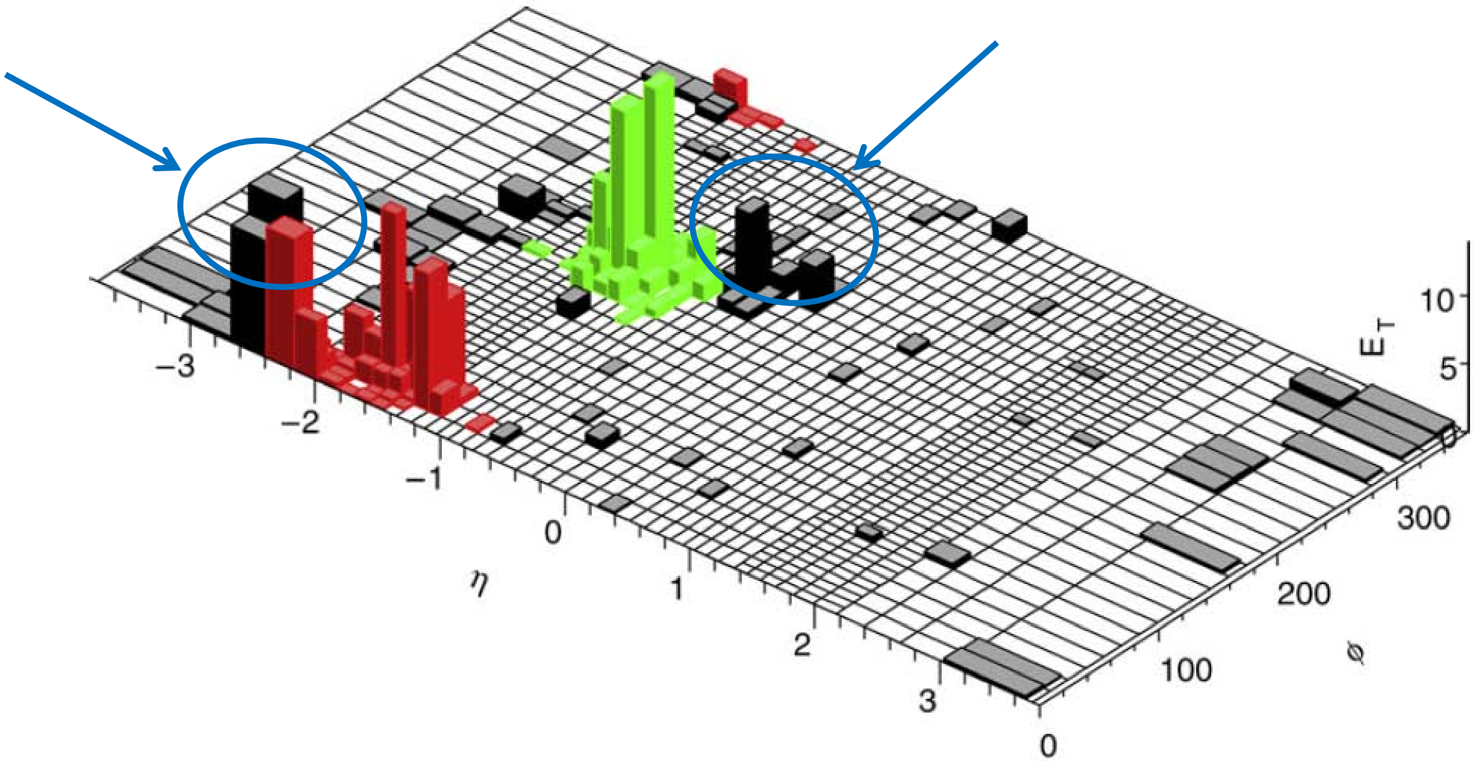
Other problems with iterative cone finders:

- "Dark" tower problem
  - Original seed moves out of cone
  - Significant energy lost for jets



Other problems with iterative cone finders:

- "Dark" tower problem
  - Original seed moves out of cone
  - Significant energy lost for jets



Iterative Seeded Fixed Cone

Advantages

- Simple geometry based algorithm
  - Easy to implement
- Fast algorithm
  - Ideal for online application in experiment

Disadvantages

- Not infrared safe
  - Can partially be recovered by splitting & merging
    - Introduces split/merge pT fraction *f* (typically 0.50 - 0.75)
  - Kills "trace" of pertubative infinities in experiment
    - Hard to confirm higher order calculations in "real life" without infinities!
- Not collinear safe
  - Used pT seeds (thresholds)
- Jets not cone shaped
  - Splitting and merging potentially makes jets bigger than original cone size and changes jet boundaries

Recursive Recombination

Motivated by gluon splitting function

- QCD branching happens all the time
  - Attempt to undo parton fragmentation
- Pair with strongest divergence likely belongs together
  - kT/Durham, first used in e^+e^-
  - Catani, Dokshitzer, Olsson, Turnock & Webber 1991

$$\left[dk_j\right]\left|M^2_{g\to g_i g_j}(k_j)\right| \simeq \frac{2\alpha_s C_A}{\pi}\frac{dE_j}{\min(E_i,E_j)}\frac{d\Theta_{ij}}{\Theta_{ij}}$$

$$(E_j \ll E_i, \Theta_{ij} \ll 1)$$

Distance between all particles *i* and *j*

$$y_{ij} = \frac{2\min(E_i^2,E_j^2)(1-\cos\Theta_{ij})}{Q^2}$$

$y_{ij} < y_{cut} \rightarrow$ combine *i* and *j*, else stop

Longitudinal invariant version for hadron colliders

- Transverse momentum instead of energy
  - Catani, Dokshitzer, Seymour & Webber 1993
  - S.D. Ellis & D. Soper 1993
- Valid at all orders!

Drop normalization to Q^2 (not fixed in pp)

$$y_{ij} \rightarrow d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2,\ d_{i,j} = p_{\mathrm{T}i,j}^2$$

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\varphi_i - \varphi_j)^2$$

$d_{ij} < d_{cut} \rightarrow$ combine *i* and *j*, else stop

(exclusive kT)

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2 / R^2$$

Recursive Recombination

P. Loch
U of Arizona
March 09, 2010

Motivated by gluon splitting function

- QCD branching happens all the time
  - Attempt to undo parton fragmentation
- Pair with strongest divergence likely belongs together
  - kT/Durham, first used in e^+e^-
  - Catani, Dokshitzer, Olsson, Turnock & Webber 1991

$$\left[dk_j\right]\left|M^2_{g\to g_i g_j}(k_j)\right| \simeq \frac{2\alpha_s C_A}{\pi}\frac{dE_j}{\min(E_i,E_j)}\frac{d\Theta_{ij}}{\Theta_{ij}}$$

$$(E_j \ll E_i, \Theta_{ij} \ll 1)$$

Distance between all particles *i* and *j*

$$y_{ij} = \frac{2\min(E_i^2,E_j^2)(1-\cos\Theta_{ij})}{Q^2}$$

$y_{ij} < y_{cut} \rightarrow$ combine *i* and *j*, else stop

Longitudinal invariant version for hadron colliders

- Transverse momentum instead of energy
  - Catani, Dokshitzer, Seymour & Webber 1993
  - S.D. Ellis & D. Soper 1993
- Valid at all orders!

Drop normalization to Q^2 (not fixed in pp)

$$y_{ij} \rightarrow d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2,\ d_{i,j} = p_{T,i,j}^2$$

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\varphi_i - \varphi_j)^2$$

$d_{ij} < d_{cut} \rightarrow$ combine *i* and *j*, else stop

(exclusive kT)

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2 / R^2$$

Recursive Recombination

Motivated by gluon splitting function

- QCD branching happens all the time
  - Attempt to undo parton fragmentation
- Pair with strongest divergence likely belongs together
  - kT/Durham, first used in e^+e^-
  - Catani, Dokshitzer, Olsson, Turnock & Webber 1991

Longitudinal invariant version for hadron colliders

- Transverse momentum instead of energy
  - Catani, Dokshitzer, Seymour & Webber 1993
  - S.D. Ellis & D. Soper 1993
- Valid at all orders!

$$\left[dk_j\right]\left|M^2_{g\rightarrow g_i g_j}(k_j)\right| \simeq \frac{2\alpha_s C_A}{\pi}\frac{dE_j}{\min(E_i,E_j)}\frac{d\Theta_{ij}}{\Theta_{ij}}$$

$$(E_j \ll E_i, \Theta_{ij} \ll 1)$$

Distance between all particles *i* and *j*

$$y_{ij} = \frac{2\min(E_i^2,E_j^2)(1-\cos\Theta_{ij})}{Q^2}$$

$y_{ij} < y_{cut} \rightarrow$ combine *i* and *j*, else stop

Drop normalization to Q^2 (not fixed in pp)

$$y_{ij} \rightarrow d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2,\ d_{i,j} = p_{\mathrm{T}i,j}^2$$

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\varphi_i - \varphi_j)^2$$

$d_{ij} < d_{cut} \rightarrow$ combine *i* and *j*, else stop

(exclusive kT)

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i,d_j)\Delta R_{ij}^2 / R^2$$

Recursive Recombination

Motivated by gluon splitting function

- QCD branching happens all the time
  - Attempt to undo parton fragmentation
- Pair with strongest divergence likely belongs together
  - kT/Durham, first used in e^+e^-
  - Catani, Dokshitzer, Olsson, Turnock & Webber 1991

Longitudinal invariant version for hadron colliders

- Transverse momentum instead of energy
  - Catani, Dokshitzer, Seymour & Webber 1993
  - S.D. Ellis & D. Soper 1993
- Valid at all orders!

$$\left[dk_j\right]\left|M^2_{g\to g_i g_j}(k_j)\right| \simeq \frac{2\alpha_s C_A}{\pi}\frac{dE_j}{\min(E_i,E_j)}\frac{d\Theta_{ij}}{\Theta_{ij}}$$

$$(E_j \ll E_i, \Theta_{ij} \ll 1)$$

Distance between all particles *i* and *j*

$$y_{ij} = \frac{2\min(E_i^2,E_j^2)(1-\cos\Theta_{ij})}{Q^2}$$

$y_{ij} < y_{\mathrm{cut}} \rightarrow$ combine *i* and *j*, else stop

Drop normalization to Q^2 (not fixed in pp)

$$y_{ij} \rightarrow d_{ij} = \min(d_i,d_j)\Delta R^2_{ij},\ d_{i,j} = p^2_{\mathrm{T}i,j}$$

$$\Delta R^2_{ij} = (y_i - y_j)^2 + (\varphi_i - \varphi_j)^2$$

$d_{ij} < d_{\mathrm{cut}} \rightarrow$ combine *i* and *j*, else stop

(exclusive kT)

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i,d_j)\Delta R^2_{ij}/R^2$$

kT –like Algorithms

P. Loch
U of Arizona
March 09, 2010

Classic procedure

- Calculate all distances d_{ji} for list of particles
  - Uses distance parameter
- Calculate d_i for all particles
  - Uses pT
- If minimum of both lists is a d_{ij}, combine *i* and *j* and add to list
  - Remove *i* and *j*, of course
- If minimum is a d_i, call *i* a jet and remove from list
- Recalculate all distances and continue all particles are removed or called a jet

Features

- Clustering sequence is ordered in kT
  - Follows jet structure

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i, d_j)\Delta R_{ij}^2 / R^2$$

$$d_i = p_{\mathrm{T}i}^2$$

Alternatives

- Cambridge/Aachen clustering
  - Uses angular distances only
  - Clustering sequence follows jet structure
- Anti-kT clustering
  - No particular ordering, sequence not meaningful

kT –like Algorithms

Classic procedure

- Calculate all distances d_{ji} for list of particles
  - Uses distance parameter
- Calculate d_i for all particles
  - Uses pT
- If minimum of both lists is a d_{ij}, combine i and j and add to list
  - Remove i and j, of course
- If minimum is a d_i, call i a jet and remove from list
- Recalculate all distances and continue all particles are removed or called a jet

Features

- Clustering sequence is ordered in kT
  - Follows jet structure

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i, d_j)\Delta R_{ij}^2 / R^2$$

$$d_i = p_{Ti}^{2n}$$

Cambridge/Aachen $(n = 0)$

cluster smallest d_{ij} first until d_{ij}>1

Alternatives

- Cambridge/Aachen clustering
  - Uses angular distances only
  - Clustering sequence follows jet structure
- Anti-kT clustering
  - No particular ordering, sequence not meaningful

kT –like Algorithms

Classic procedure

- Calculate all distances d_{ji} for list of particles
  - Uses distance parameter
- Calculate d_i for all particles
  - Uses pT
- If minimum of both lists is a d_{ij}, combine *i* and *j* and add to list
  - Remove *i* and *j*, of course
- If minimum is a d_i, call *i* a jet and remove from list
- Recalculate all distances and continue all particles are removed or called a jet

Features

- Clustering sequence is ordered in kT
  - Follows jet structure

Inclusive longitudinal invariant clustering

$$d_{ij} = \min(d_i, d_j) \Delta R_{ij}^2 / R^2$$

$$d_i = p_{\mathrm{T}i}^{2n}$$

Cambridge/Aachen $(n = 0)$

cluster smallest d_{ij} first until $d_{ij} > 1$

Anti-kT $(n = -1)$

follow classic algorithm

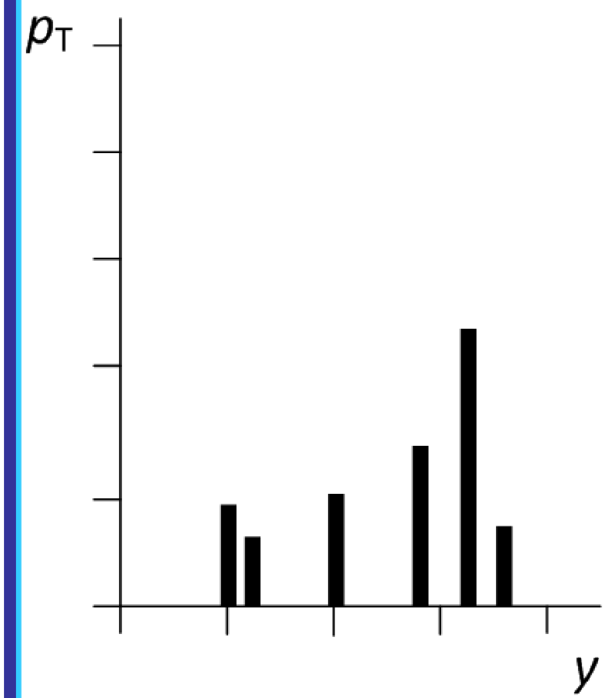
Alternatives

- Cambridge/Aachen clustering
  - Uses angular distances only
  - Clustering sequence follows jet structure
- Anti-kT clustering
  - No particular ordering, sequence not meaningful

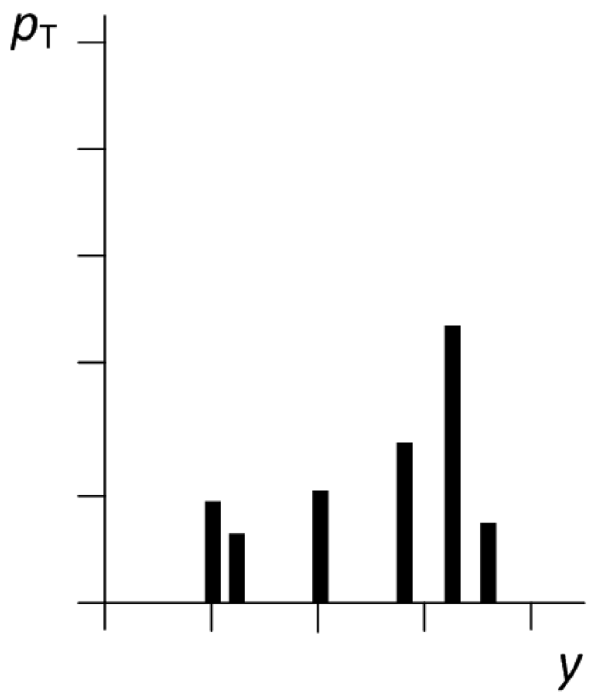
kT Examples

P. Loch
U of Arizona
March 09, 2010

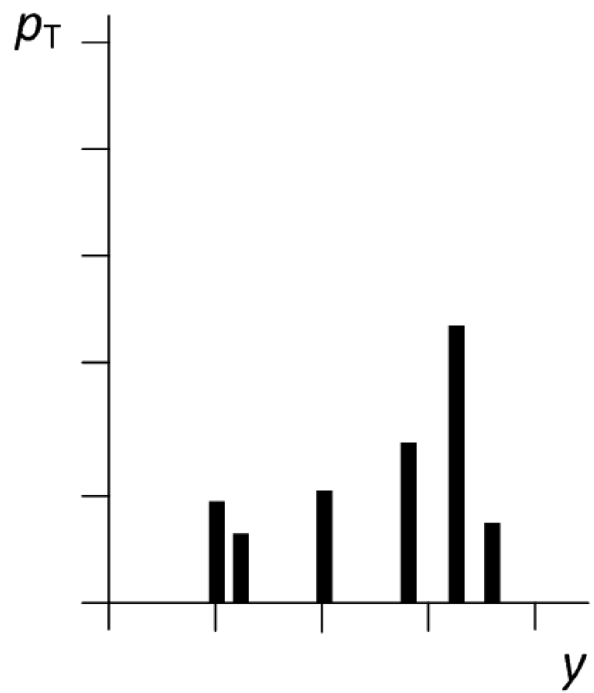
kT, *n*=1



Anti-kT, *n*=-1



Cambridge/Aachen, *n*=0

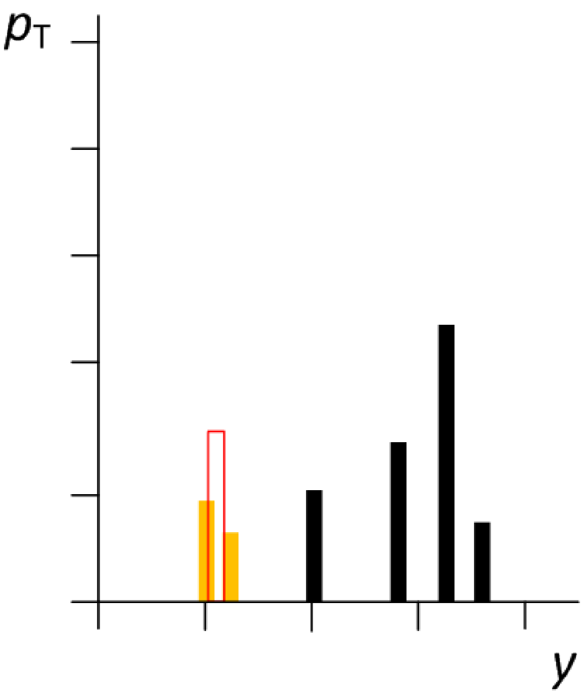


THE UNIVERSITY OF ARIZONA.

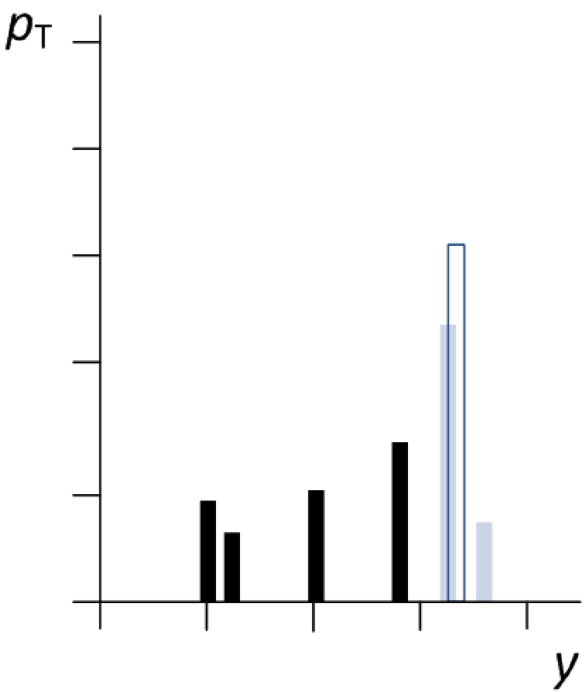
kT Examples

P. Loch
U of Arizona
March 09, 2010

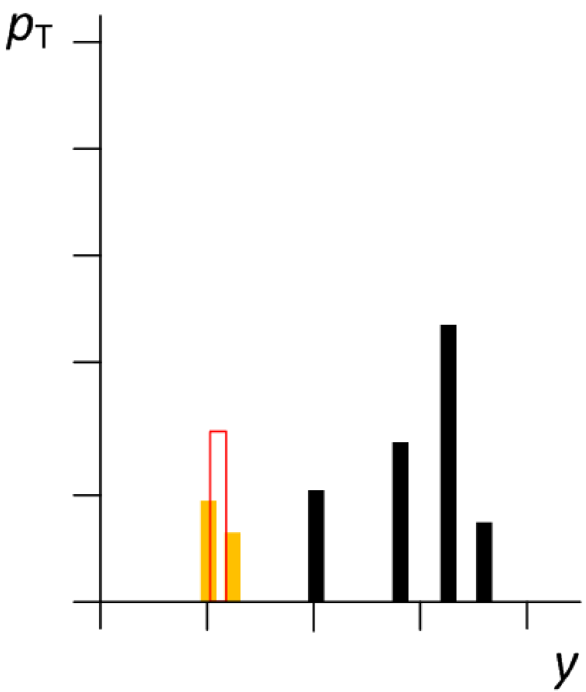
kT, *n*=1



Anti-kT, *n*=-1



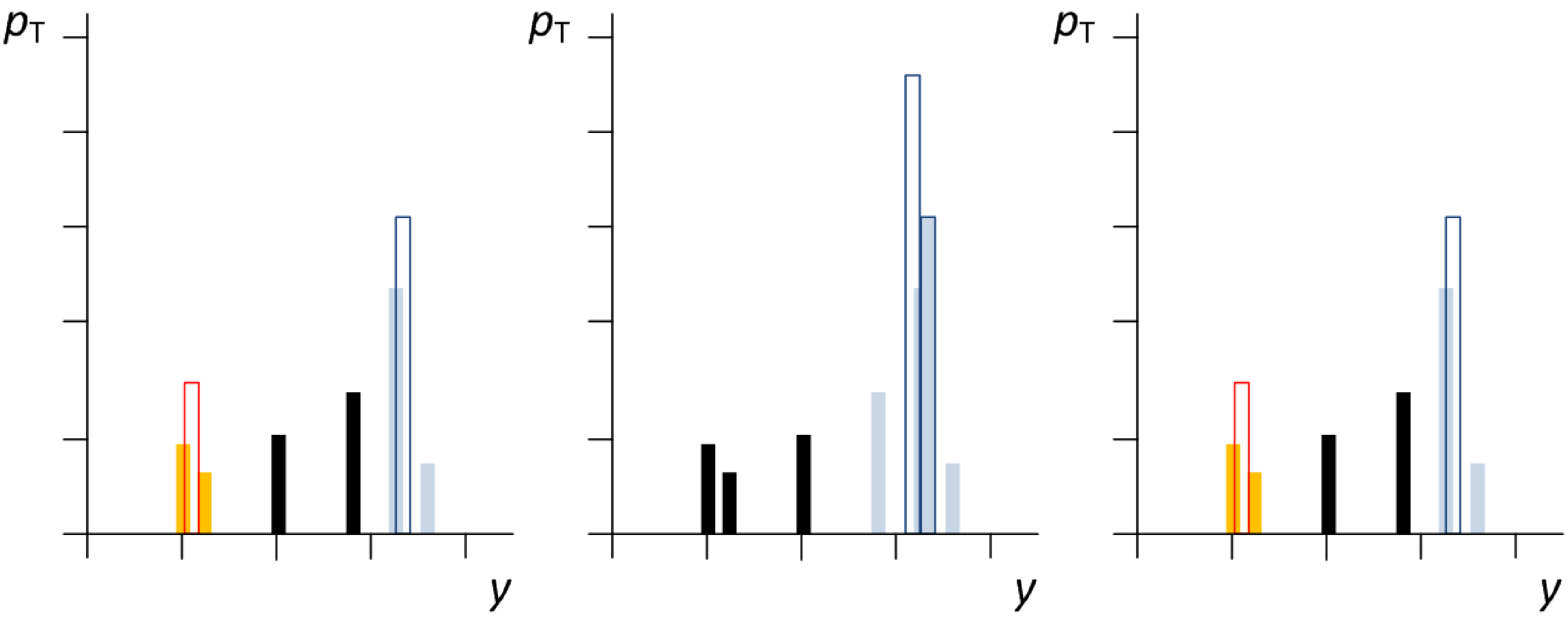
Cambridge/Aachen, *n*=0



kT Examples

P. Loch
U of Arizona
March 09, 2010

kT, n=1 | Anti-kT, n=-1 | Cambridge/Aachen, n=0

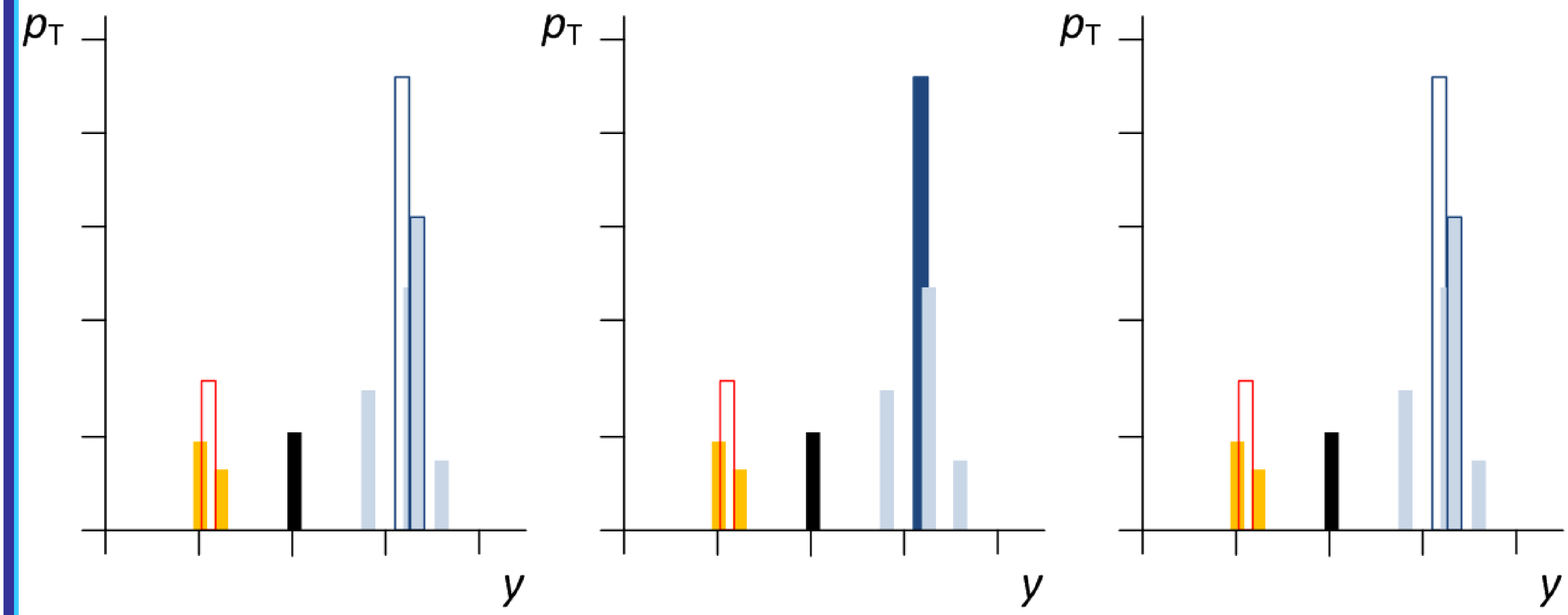


THE UNIVERSITY OF ARIZONA.

kT Examples

P. Loch
U of Arizona
March 09, 2010

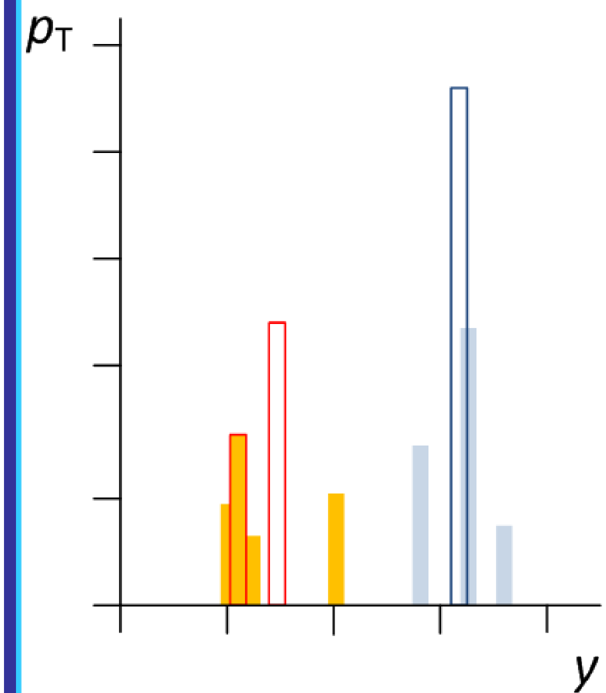
kT, n=1 — Anti-kT, n=-1 — Cambridge/Aachen, n=0



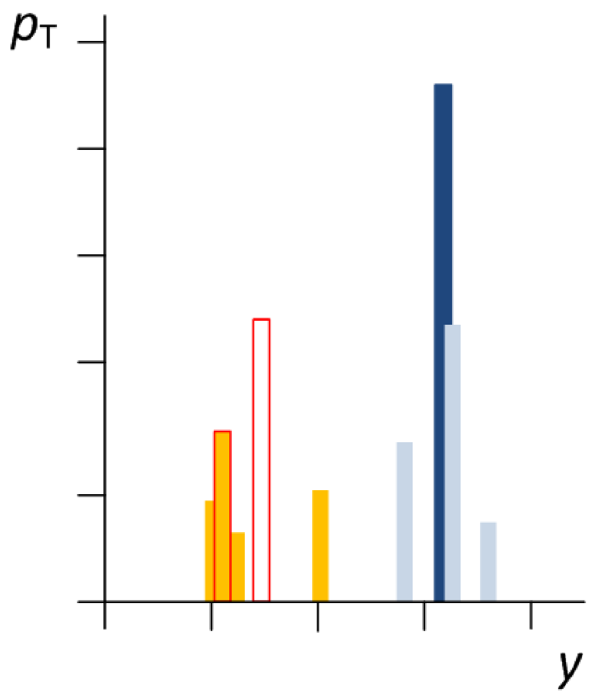
kT Examples

P. Loch
U of Arizona
March 09, 2010

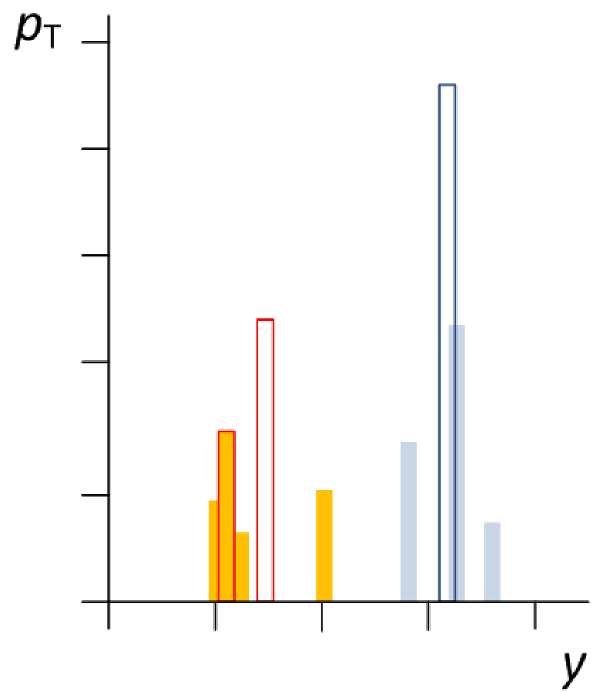
kT, n=1



Anti-kT, n=-1



Cambridge/Aachen, n=0

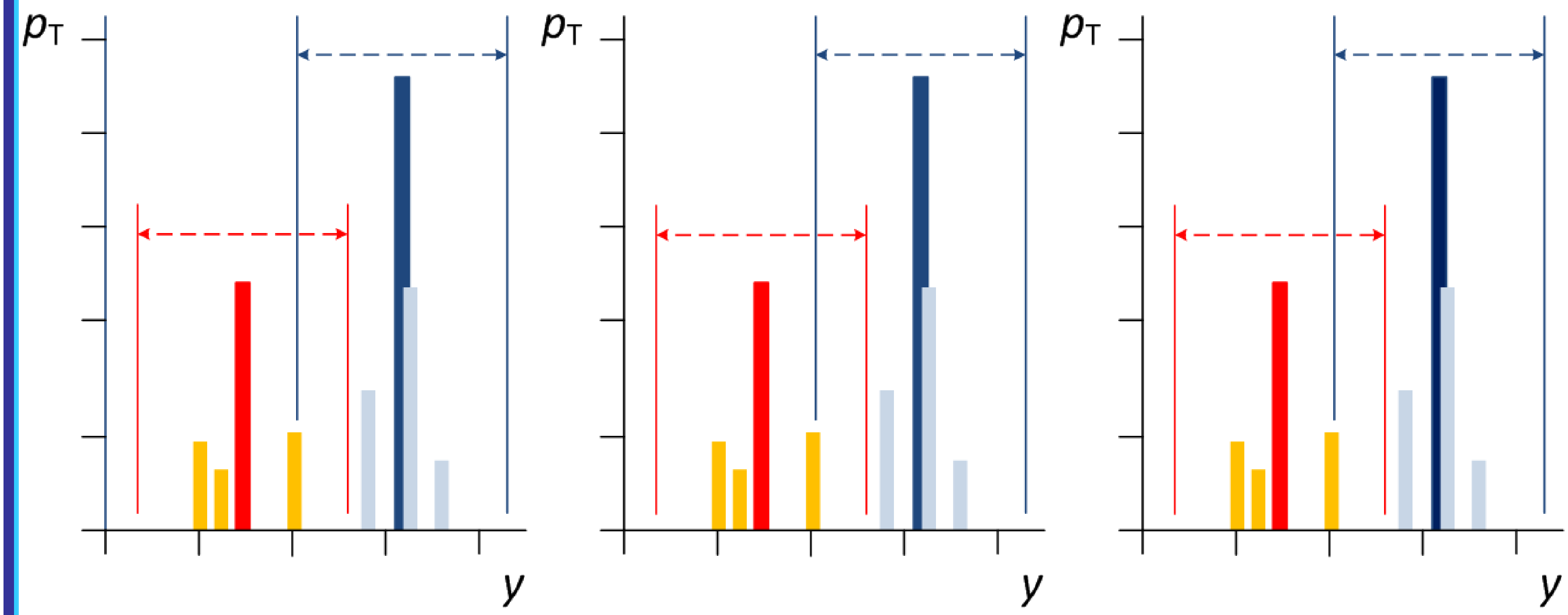


kT Examples

kT, *n*=1

Anti-kT, *n*=-1

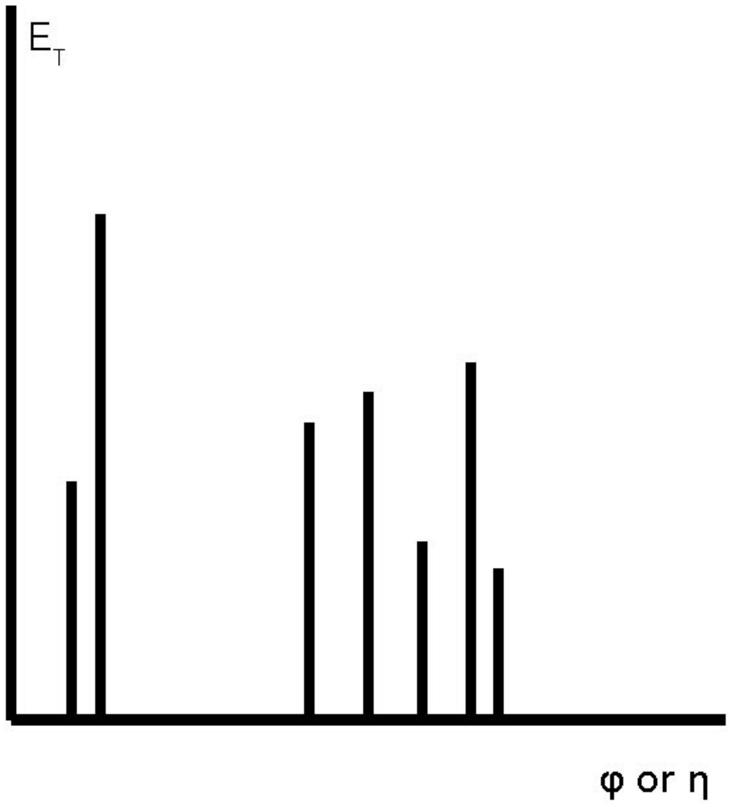
Cambridge/Aachen, *n*=0



Recursive Recombination Algorithms

Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets



Clustering Algorithms:

- Define a distance d_{ij} between two objects i, j:

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

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

and a distance d_{iB} between one object i and the beam direction B:

$$d_{iB} = k_{ti}^2$$

- Find the smallest of d_{ij}, d_{iB} .

  If d_{ij} recombine i, j ;

  If d_{iB} , i is a jet.

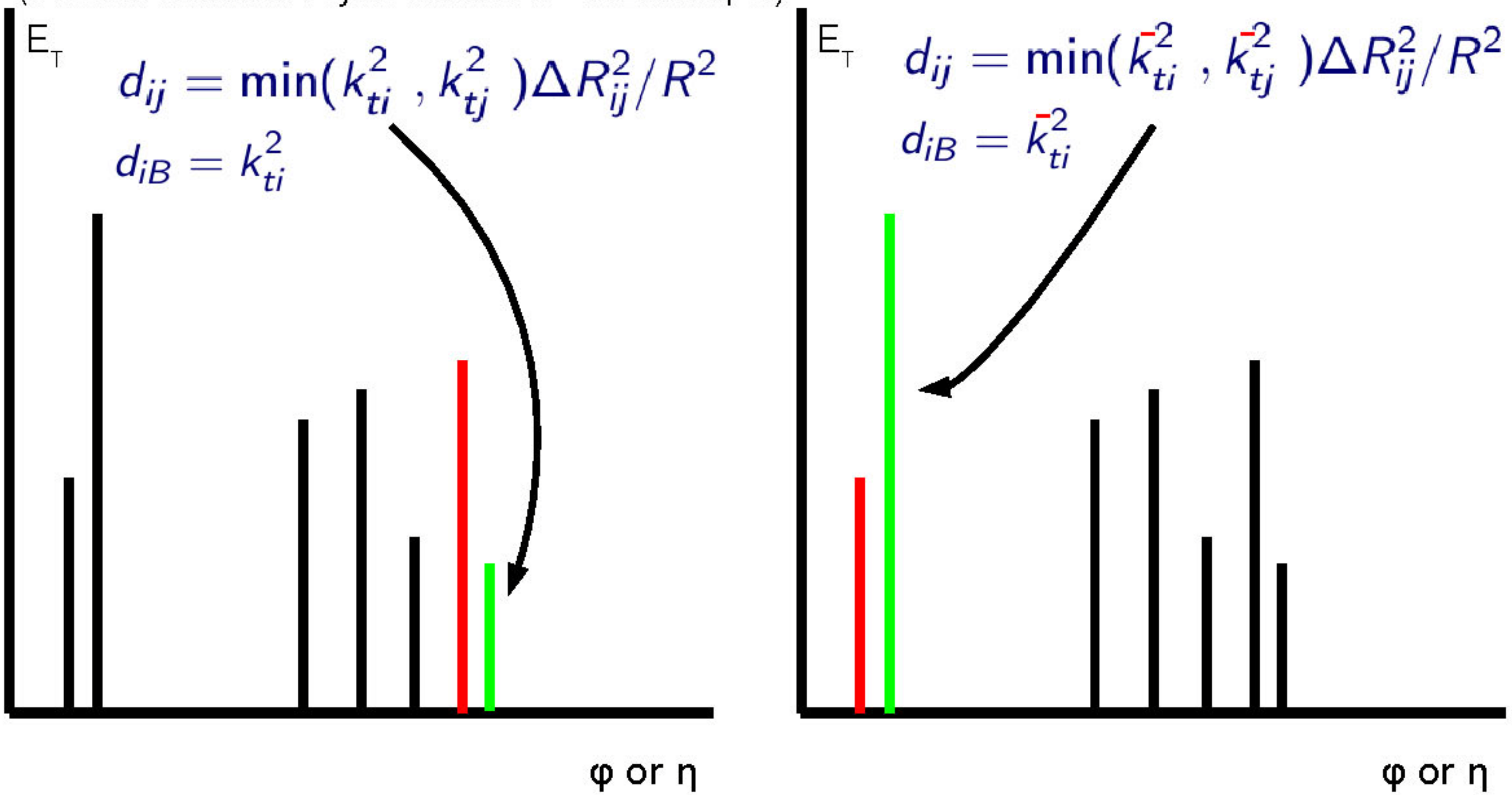
Recursive Recombination Algorithms

Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets

Kt (Catani/Dokshitzer/Seymour/Webber - S.Ellis/Soper)

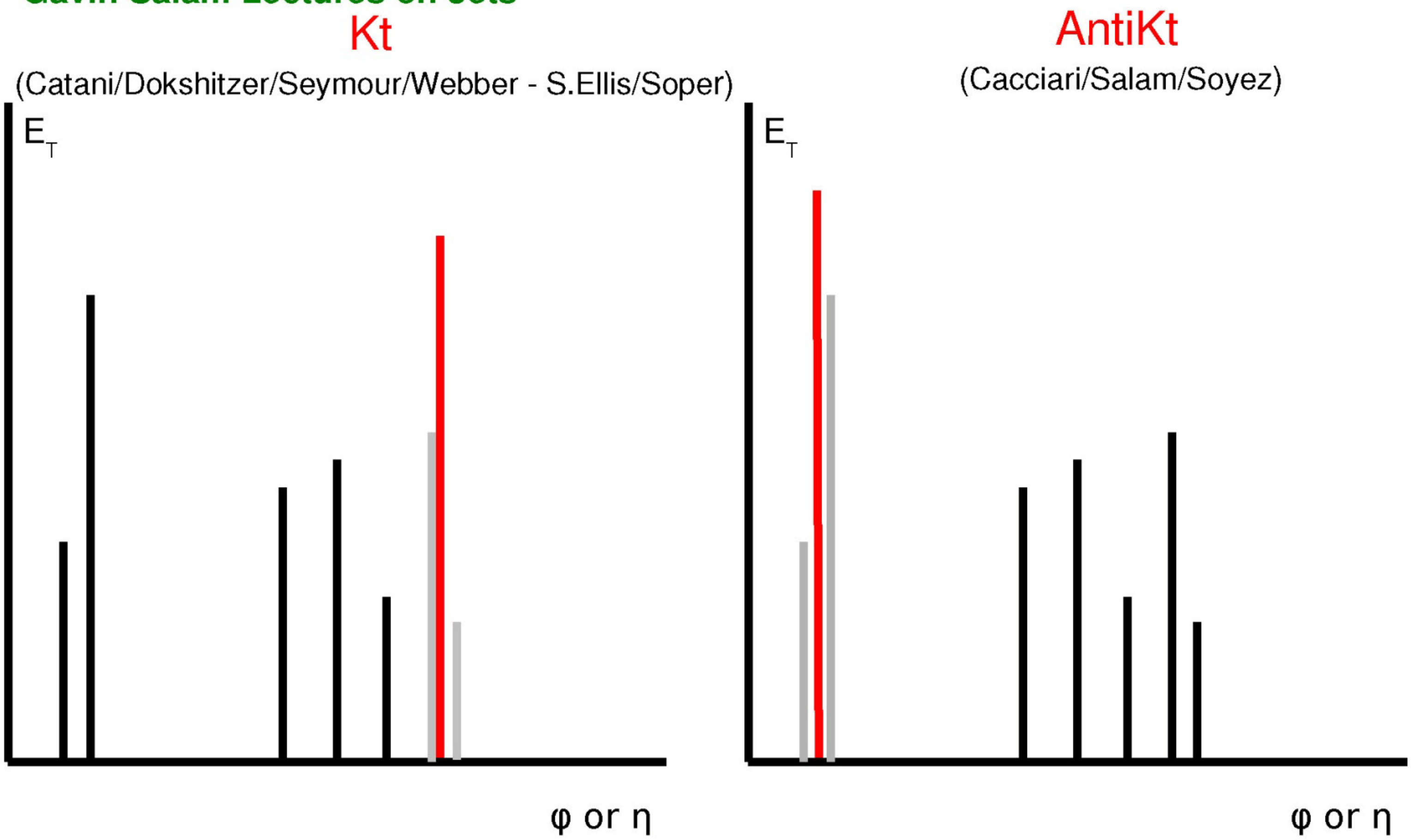
AntiKt (Cacciari/Salam/Sovez)



Recursive Recombination Algortihms

Clustering Algorithms

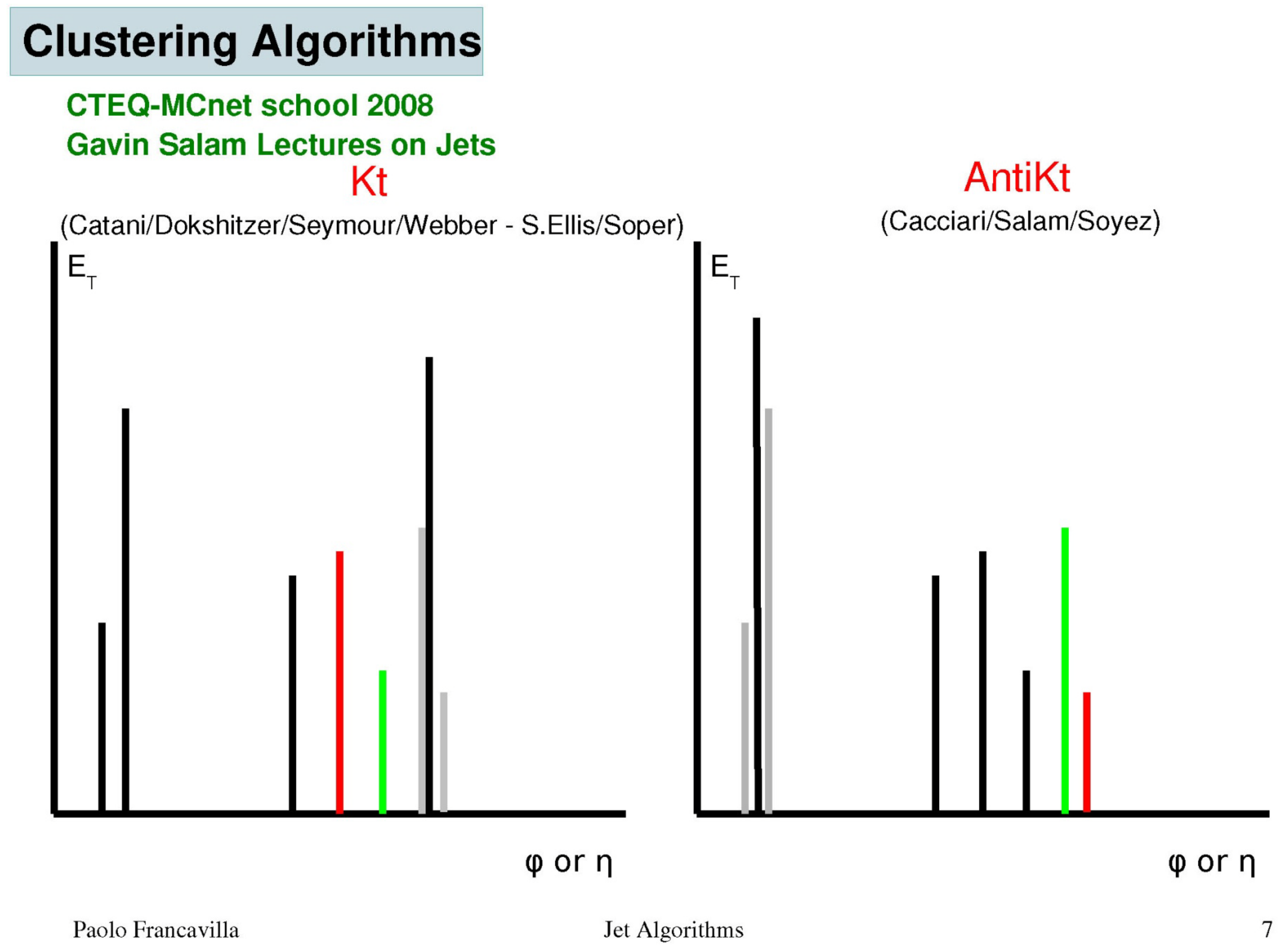
CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets



THE UNIVERSITY OF ARIZONA

Recursive Recombination Algortihms

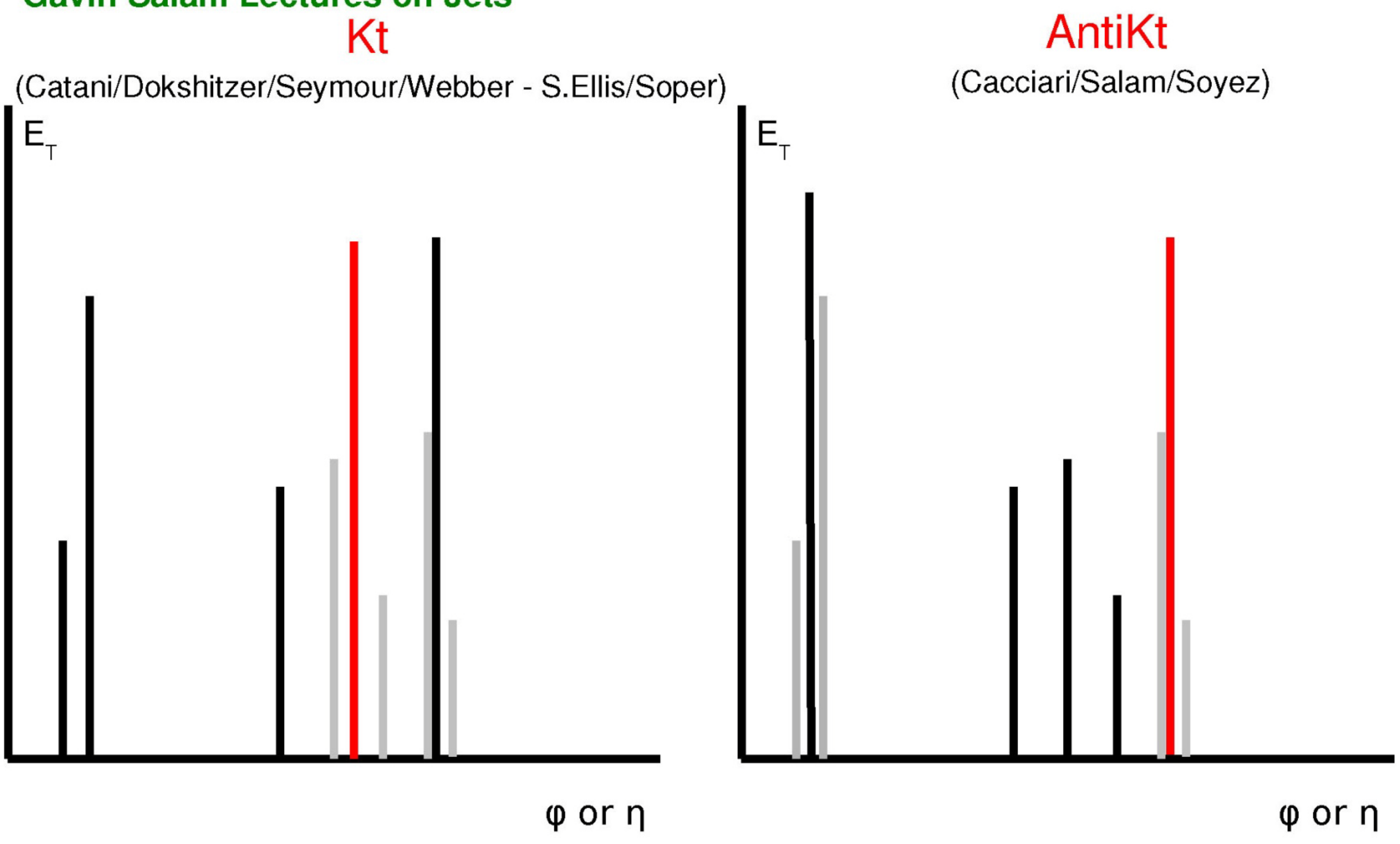
P. Loch
U of Arizona
March 09, 2010



Recursive Recombination Algorithms

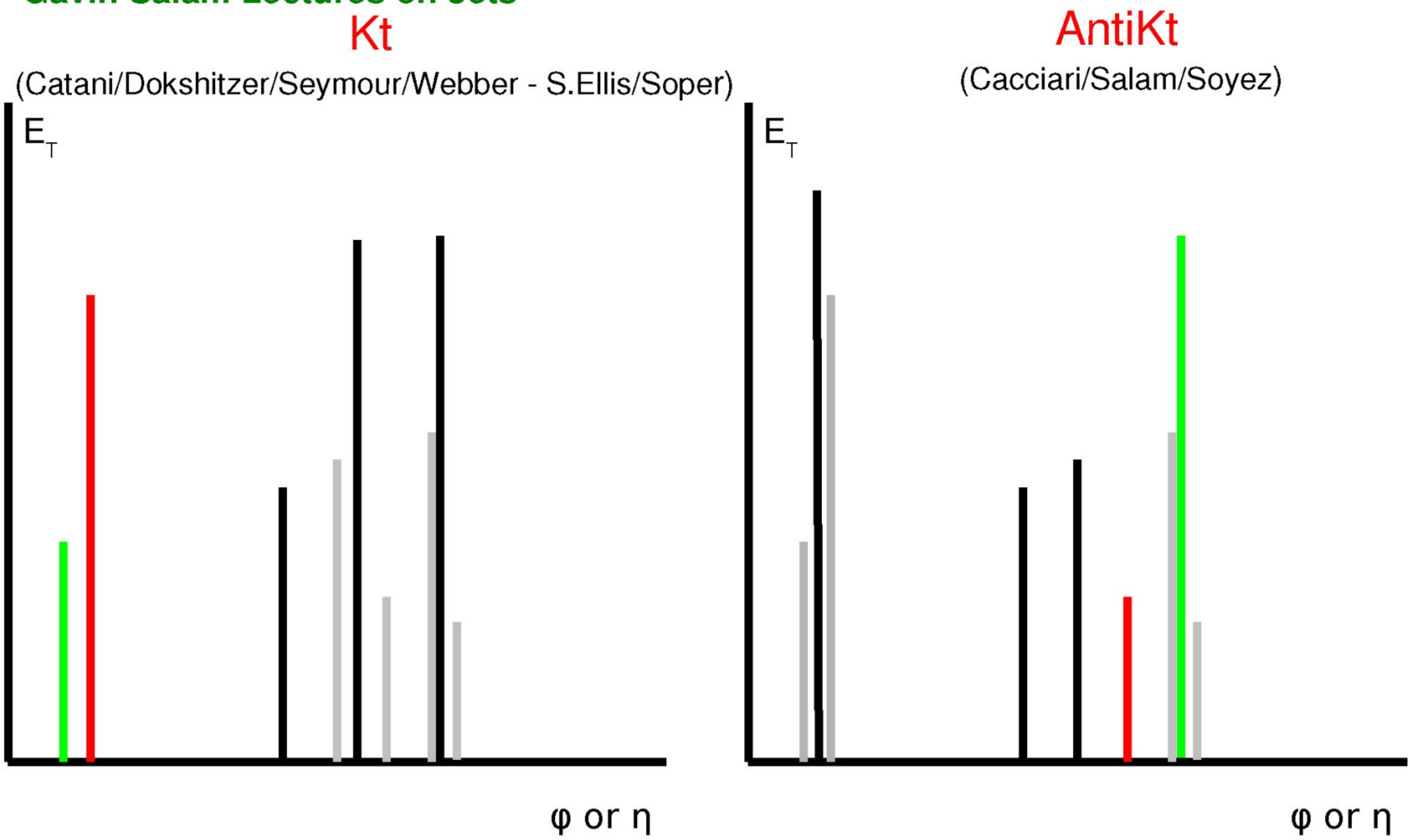
Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets



Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets

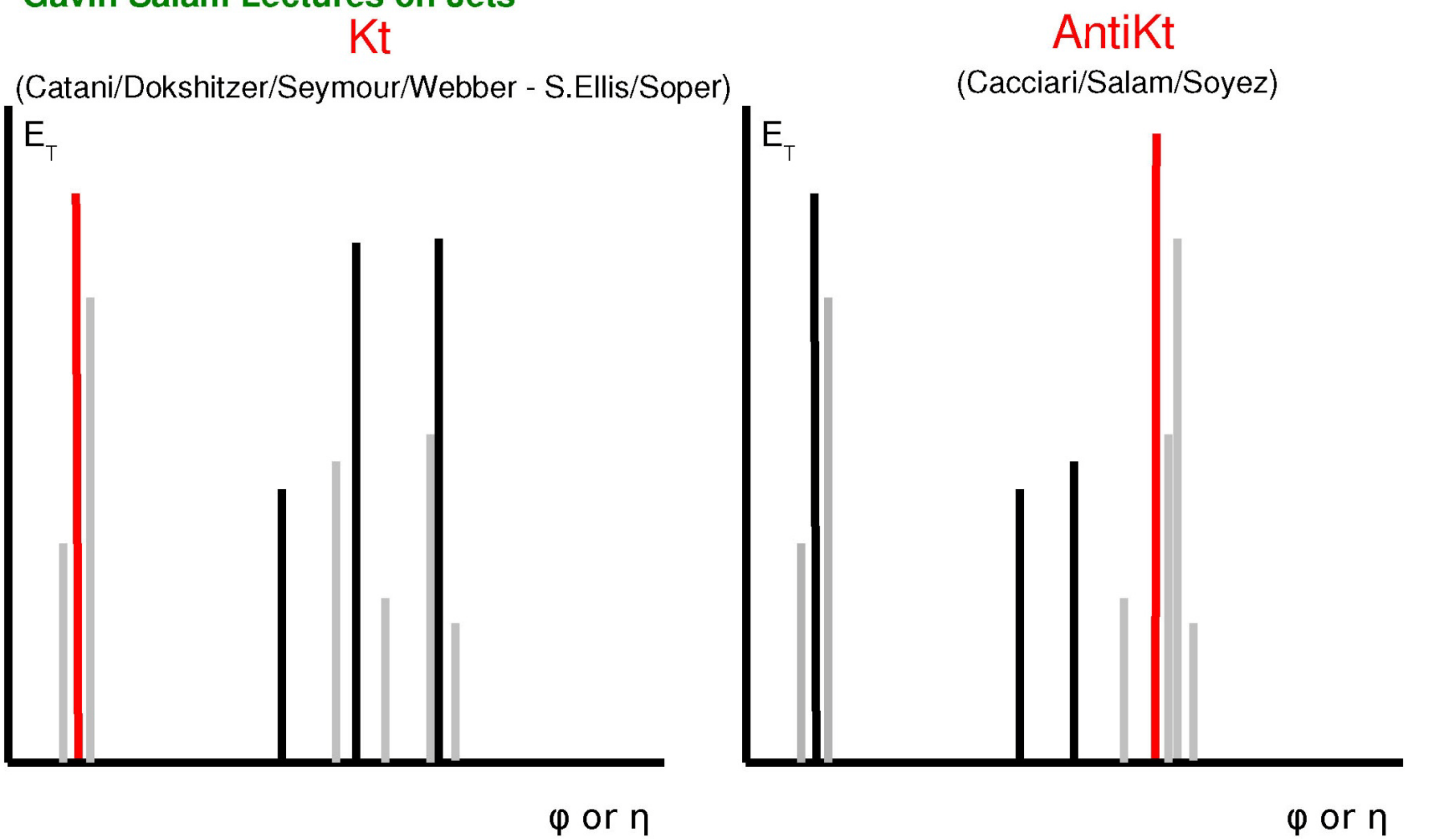


THE UNIVERSITY OF ARIZONA.

Recursive Recombination Algorithms

Clustering Algorithms

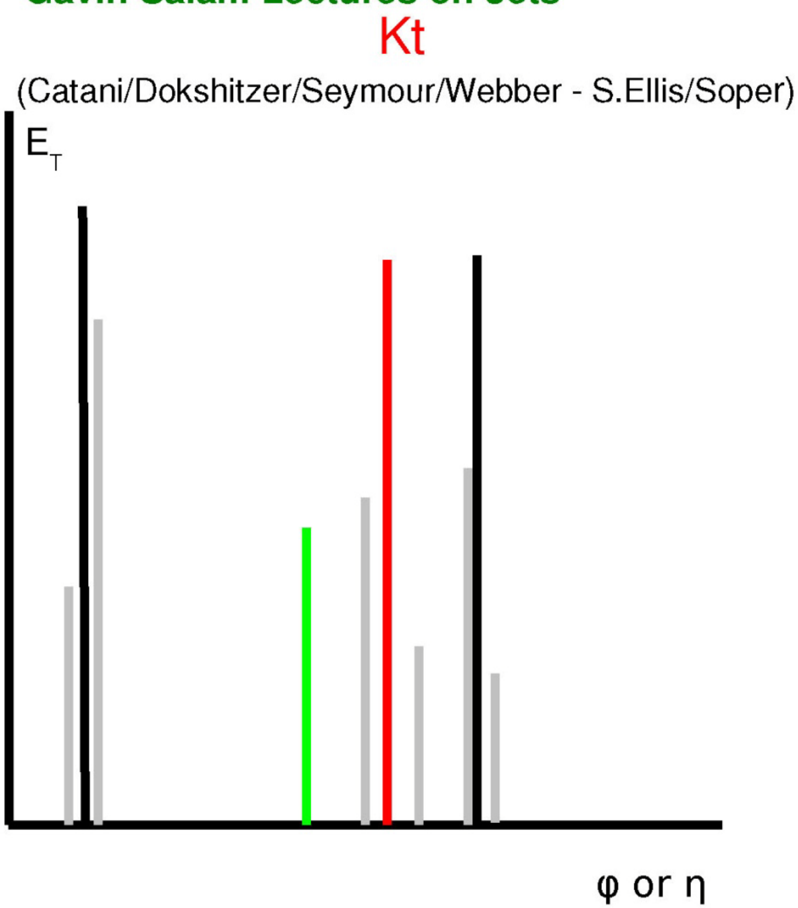
CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets

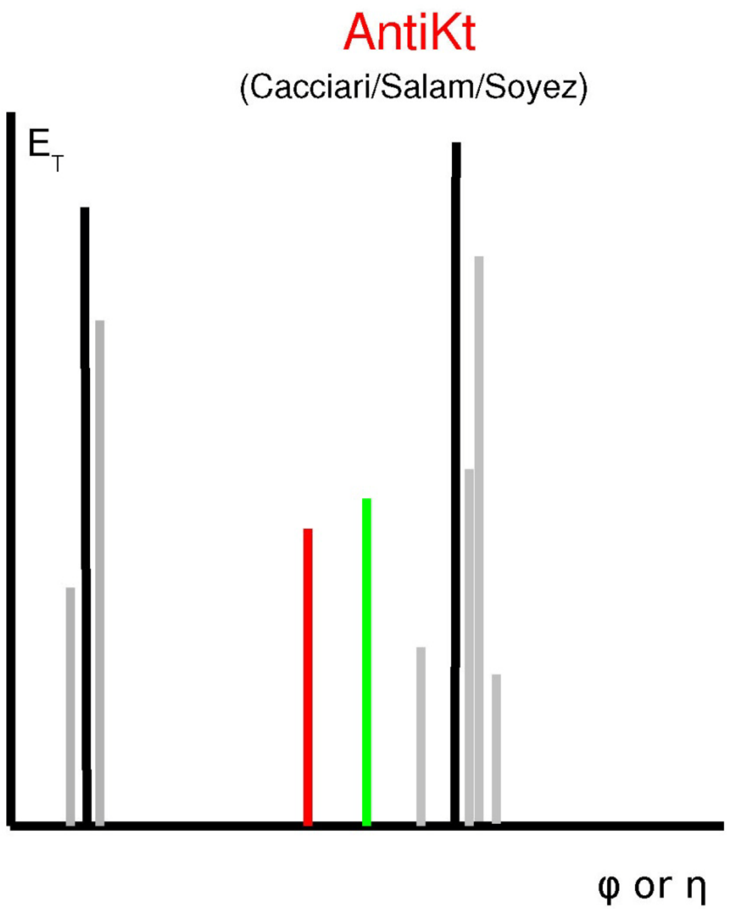


Recursive Recombination Algorithms

Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets

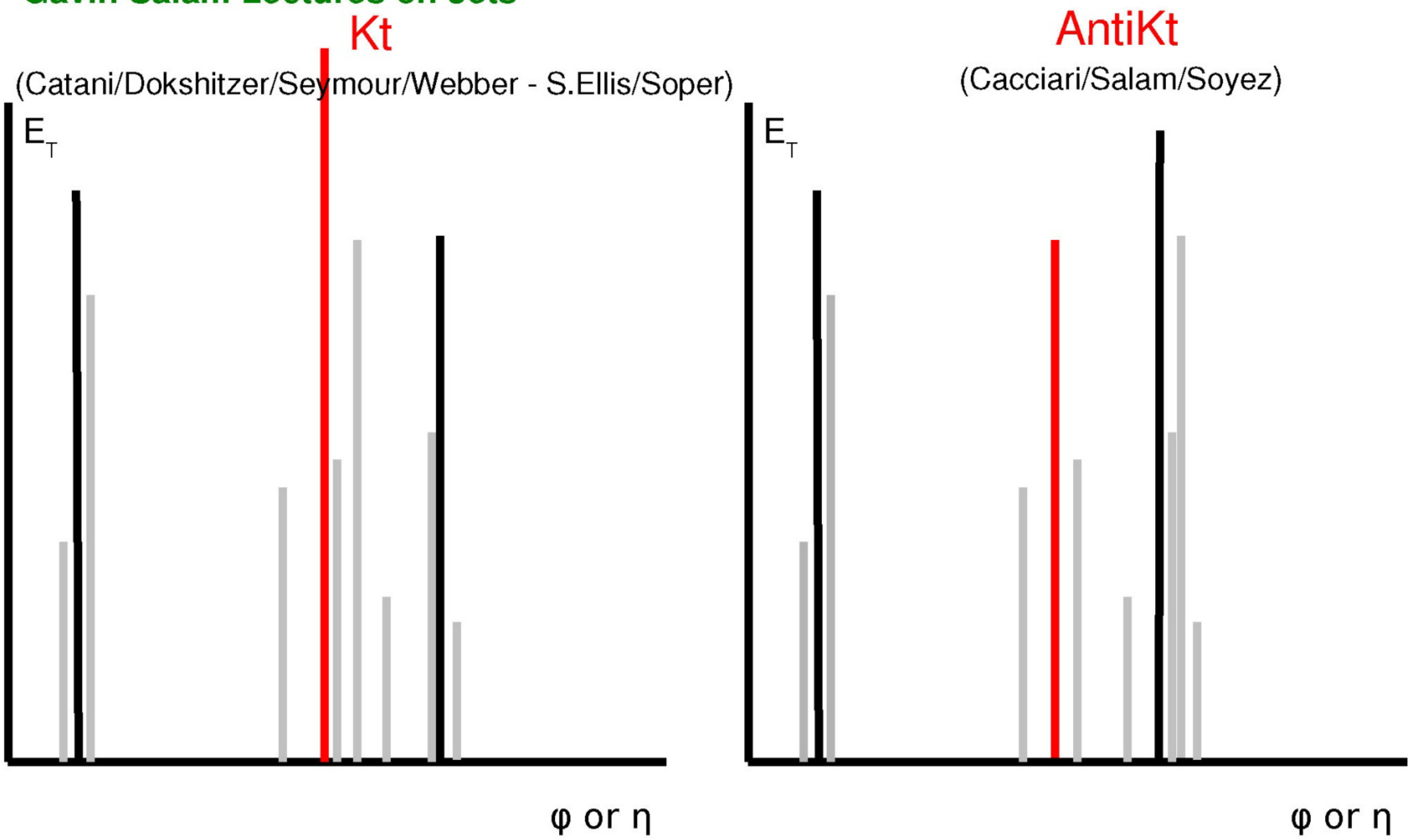




Recursive Recombination Algorithms

Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets



Recursive Recombination Algorithms

Clustering Algorithms

CTEQ-MCnet school 2008
Gavin Salam Lectures on Jets