An Overview of Jet-Finding Algorithms

By: Vetri Velan

March 30, 2016

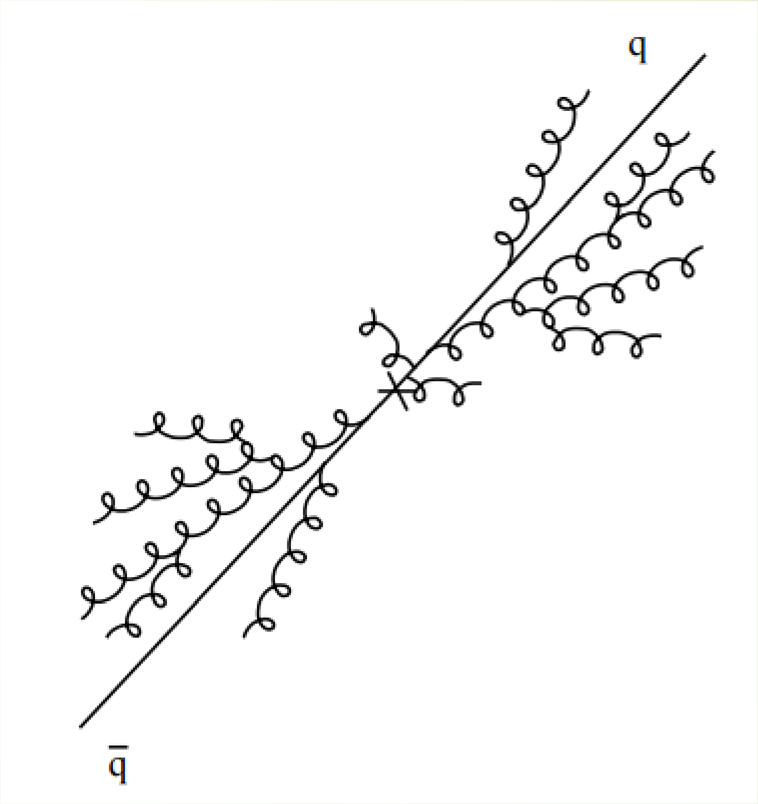
Outline

- What are jets?
- Requirements for jet-finding algorithms
- Cone algorithms
- Sequential Clustering Algorithms
- Jet Substructure

Disclaimer: Most of this material taken from lectures by Gavin Salam, including many slides (full list of references and hyperlinks at the end of the slides)

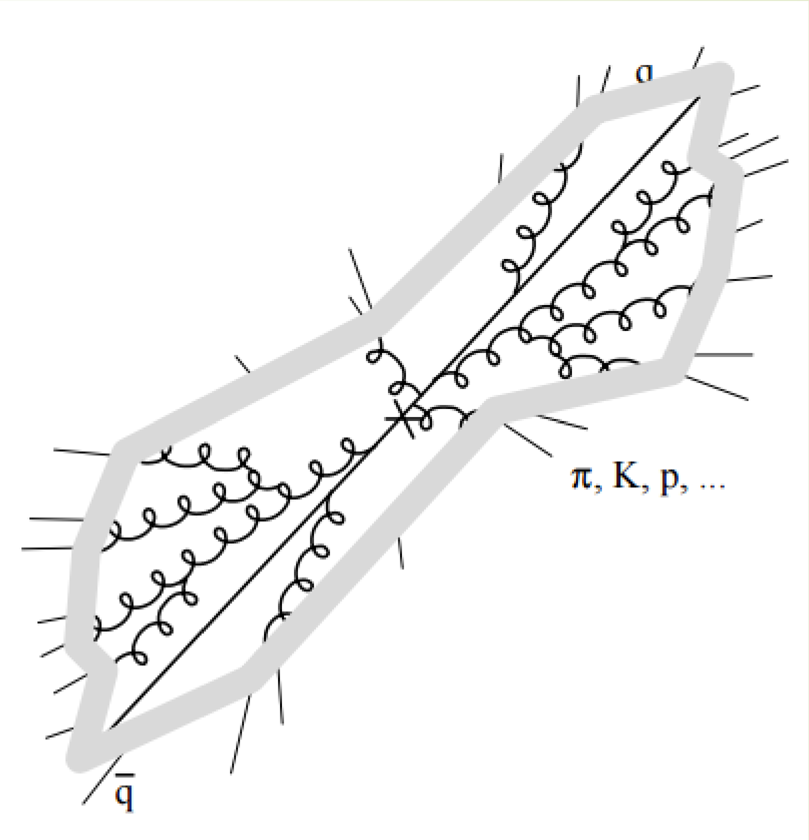
What are jets?

- Simple answer: streams of collimated hadronic particles created from quarks or gluons
- Example: $q\bar{q}$ pair created at primary vertex, then these can emit gluons, which emit gluons, etc…



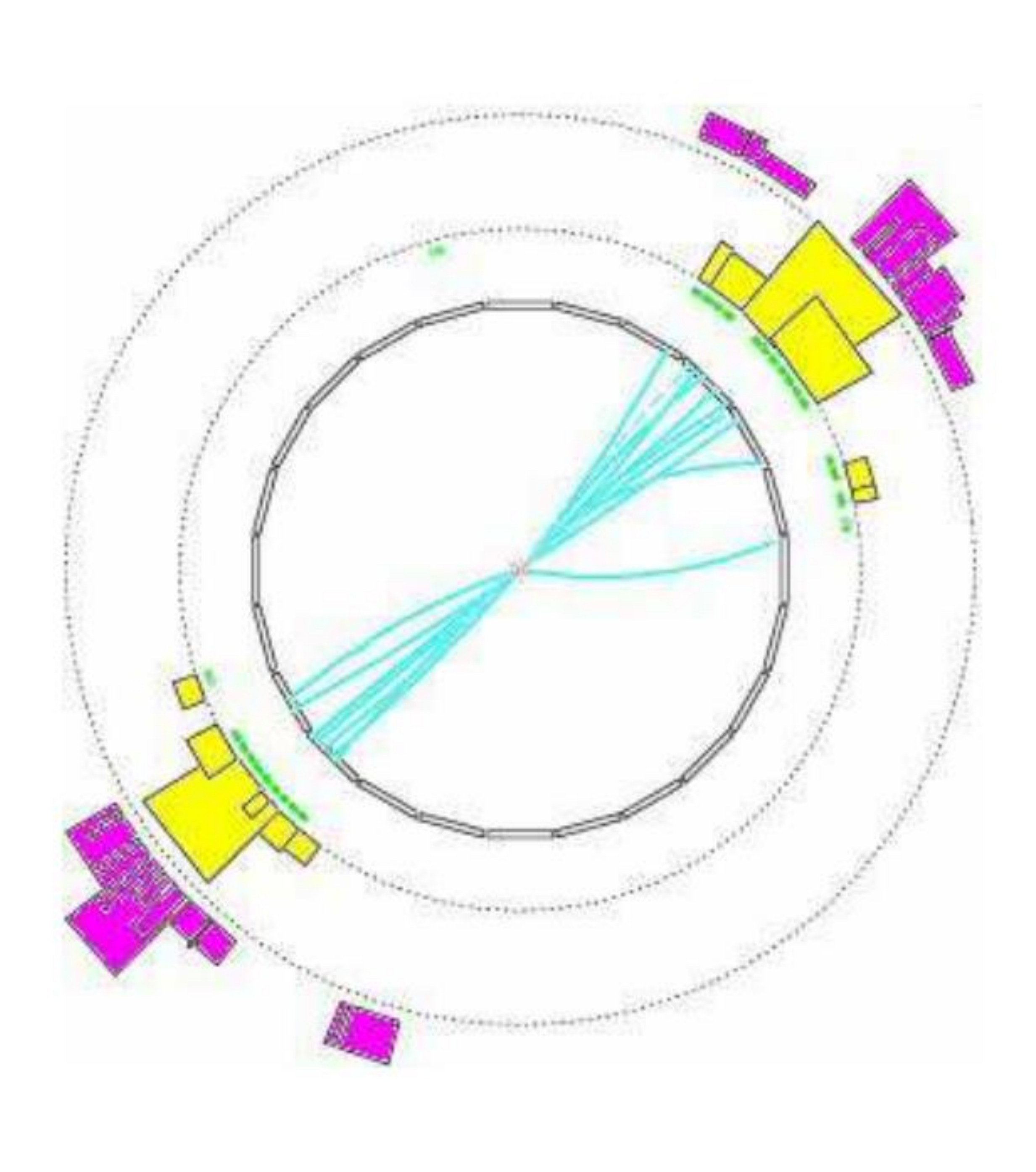
What are jets?

- Simple answer: streams of collimated hadronic particles created from quarks or gluons
- Example: $q\bar{q}$ pair created at primary vertex, then these can emit gluons, which emit gluons, etc…
- These hadronize and form mesons and baryons



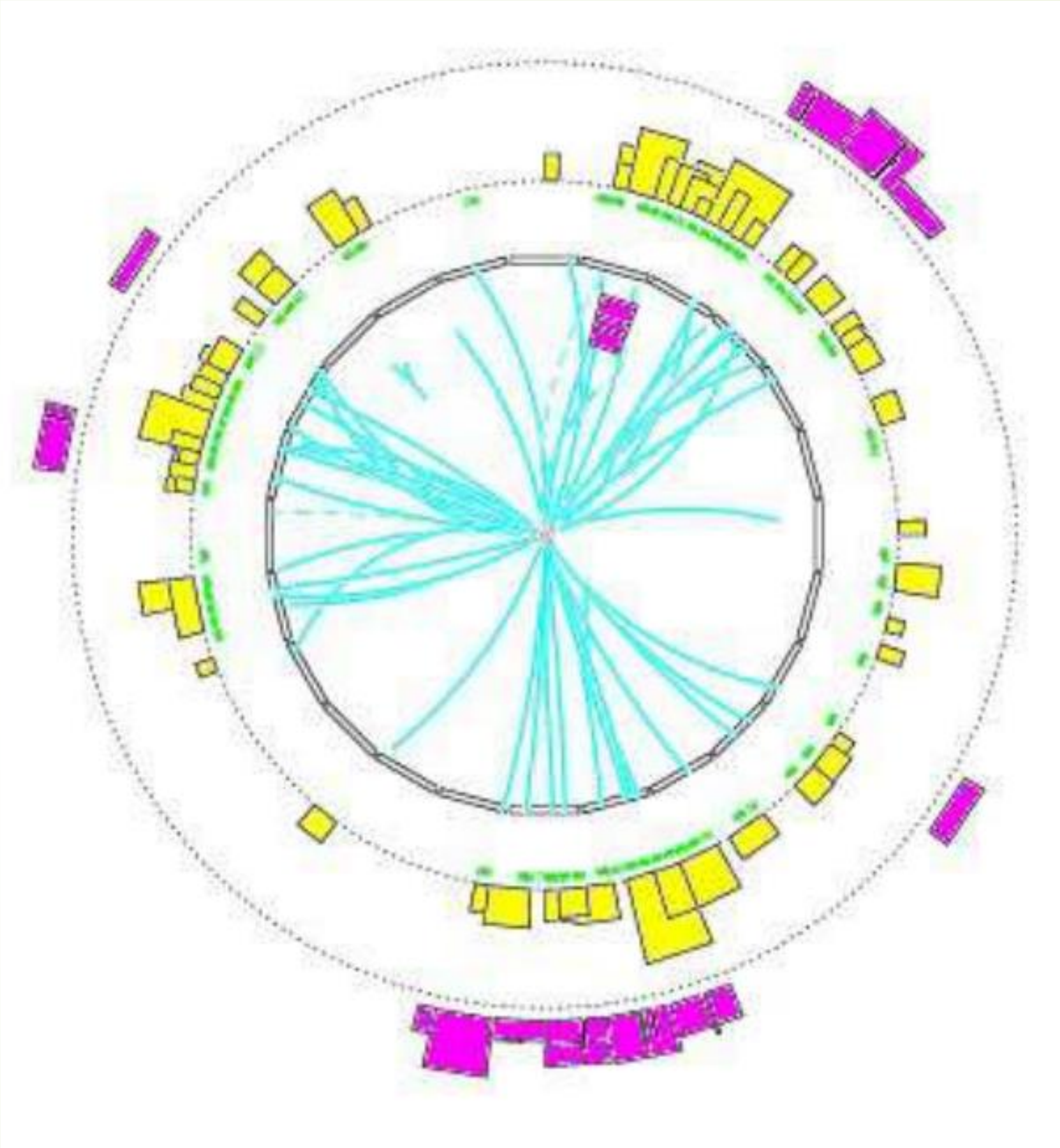
What are jets?

- Example at right: 2 clear jets that come back-to-back (as they should)



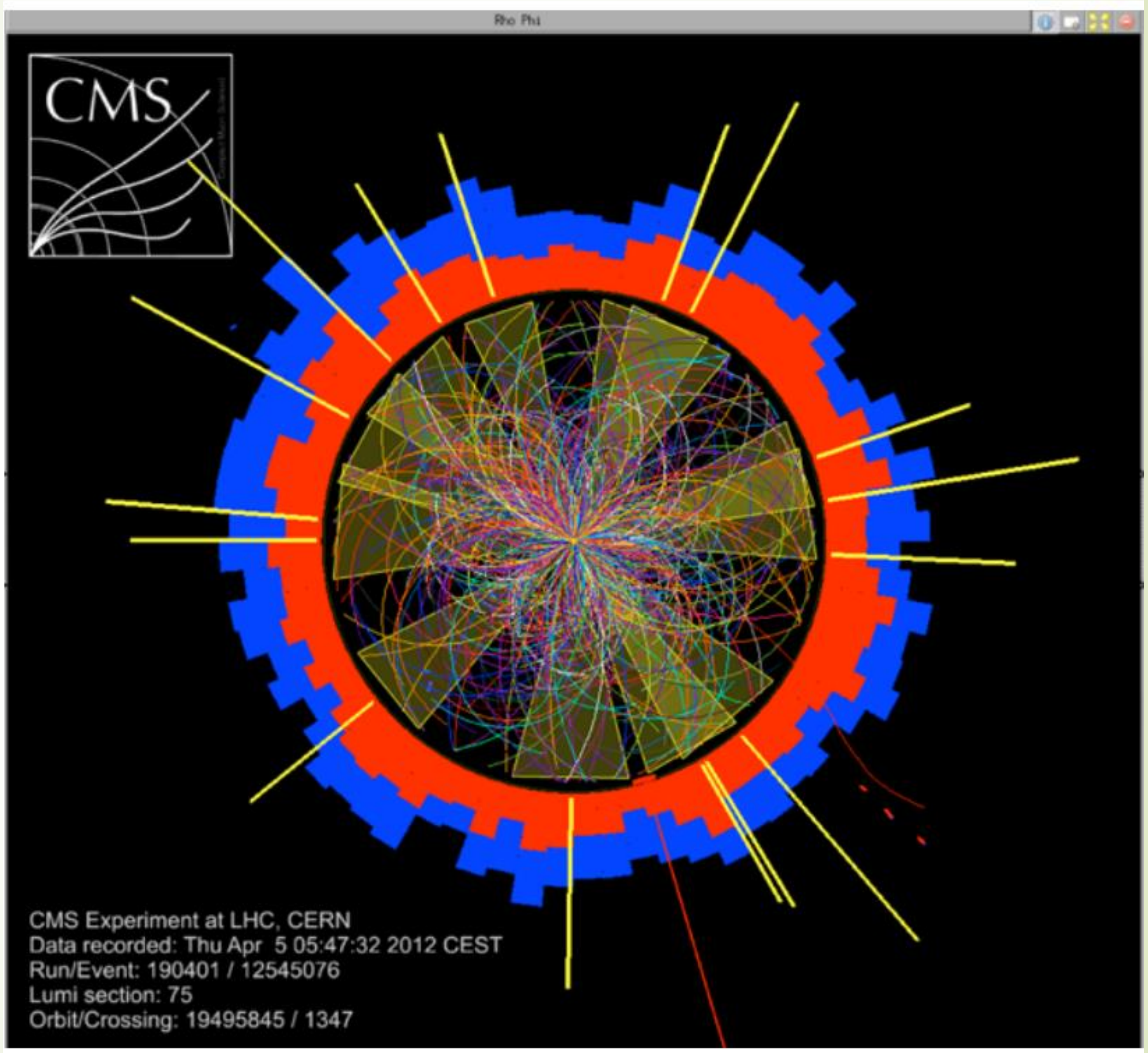
What are jets?

- But this is not so easy
- How many jets are in the figure at right?



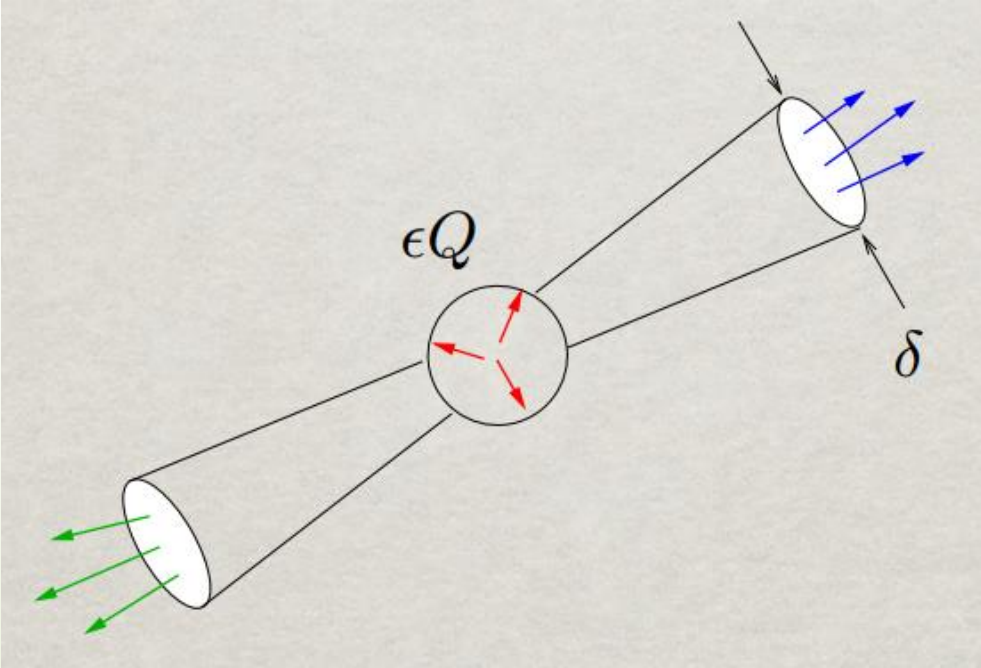
What are jets?

- But this is not so easy
- How many jets are in the figure at right?
- Can get much, much worse…



What are jets?

- To do any type of useful analysis at the LHC (or Tevatron, etc.), we need a better definition of a jet
- One of the first definitions: Sterman and Weinberg, PRL 39, 1436 (1977)
  - Cones of opening angle δ containing all but a fraction $\epsilon \ll 1$ of the total collision energy



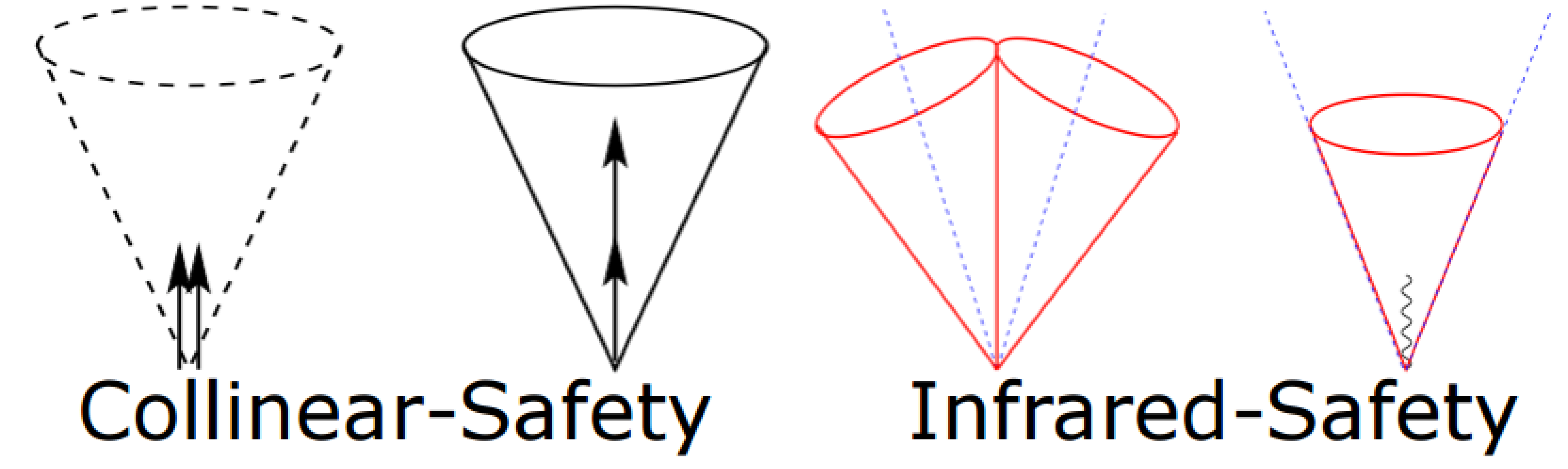
What are jets?

- "Snowmass Accord", J. E. Huth et al., "Toward a standardization of jet definitions", FNAL-C-90-249-E
  1. Simple to implement in an experimental analysis
  2. Simple to implement in theoretical calculation
  3. Defined at any order of perturbation theory
  4. Yields finite cross sections at any order in perturbation theory
  5. Yields a cross section that is relatively insensitive to hadronization

- Michael Tannenbaum: "…read more like legal contracts between experimentalists and theorists than like scientific papers" (PoS, Aug 2007, arXiv:0707.1706)

What are jets?

- For our purposes today, we deal with two primary requirements:
  - Infrared safety: Definition of a jet should be insensitive to “soft” (low-energy) gluons emitted by a quark
  - Collinear safety: Definition of a jet should be insensitive to the emission of collinear gluons
  - Often referred to as just “infrared-safety” or “IRC-safety”



What are jets?

- Why IR safety?
- Transition rate (matrix element squared) for gluon emission is proportional to:

$$dS = \frac{2\alpha_s C_F}{\pi} \frac{dE}{E} \frac{d\theta}{\sin\theta} \frac{d\phi}{2\pi}$$

- Singularities at E = 0 (soft gluon) and $\theta = 0, \pi$ (collinear emission)
- These infinities are removed via cancellations in virtual corrections
- IR-safe jet definitions will allow these cancellations to occur → we can calculate jet cross sections using perturbative QCD

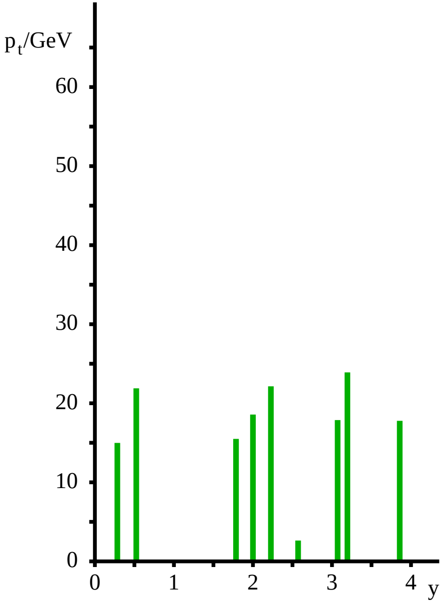
Jet-Finding Algorithms

- Purpose: Take data, figure out how many jets, where the jets are, and characterize them (primarily E, p)
- In light of what we've discussed, "jet-finding algorithms" are perhaps better called "jet-defining algorithms"
- Two primary classes of algorithms:
  - Cone algorithms
  - Sequential clustering algorithms

Cone Algorithms

- General idea behind a cone algorithm is to sort the data into "cones" of hadronic particles, and call these jets
- Iterative Cone, Progressive Removal (IC-PR):
  - Take most energetic particle as "seed" for axis of the cone
  - Draw cone around the seed, with some specified radius and angular width
  - Sum the momenta of the particles inside the cone; use the axis of the resultant vector as the axis of a new cone
  - Iterate until the cone is stable, within some precision
  - When stable, call this cone a "jet" and remove from the event; iterate until all particles are split into jets
- Example of IC-PR on next several slides (taken from G. Salam)

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

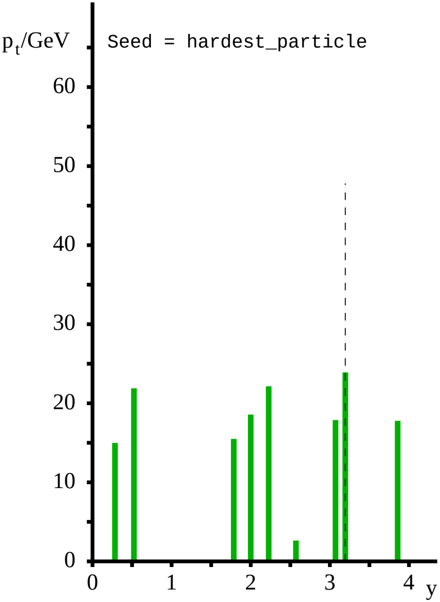
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

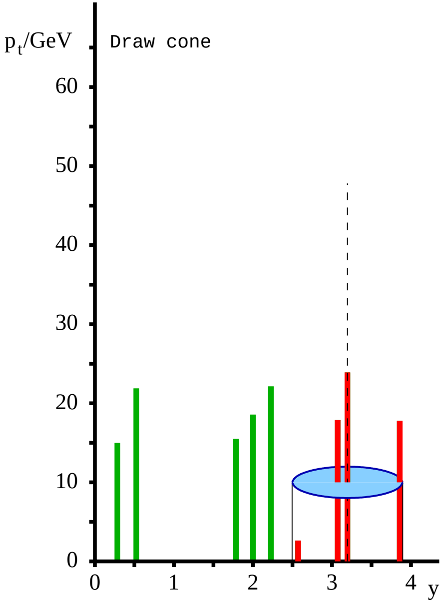
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

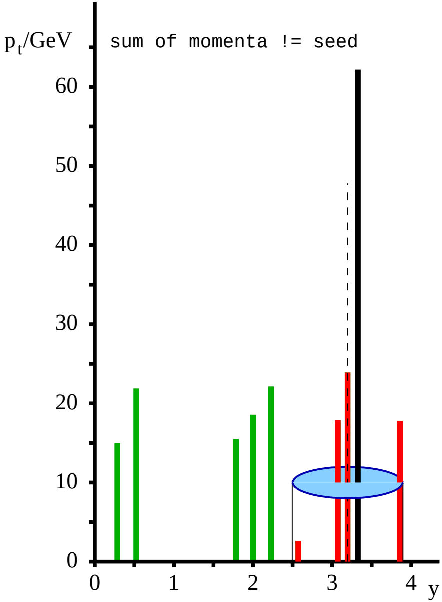
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

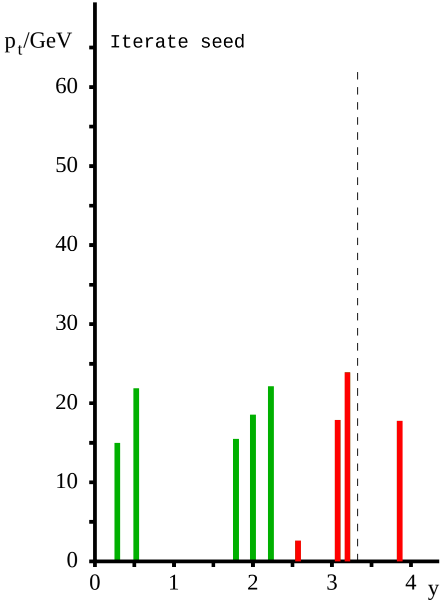
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

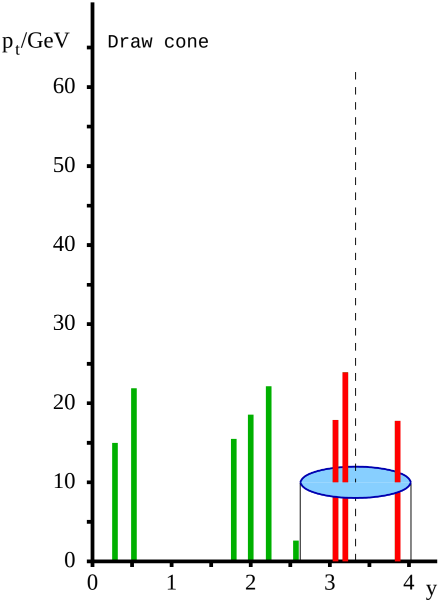
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

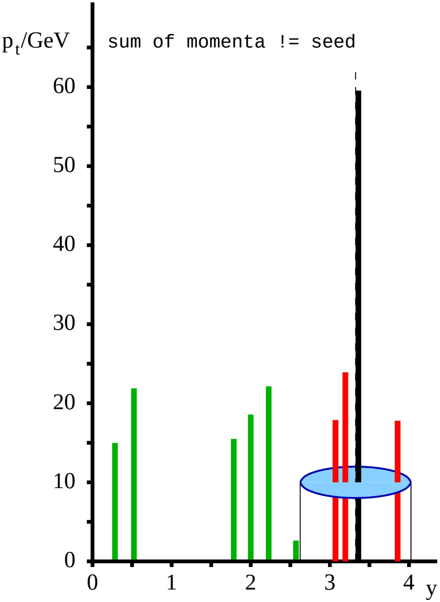
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



<u>One of the simpler cones</u>

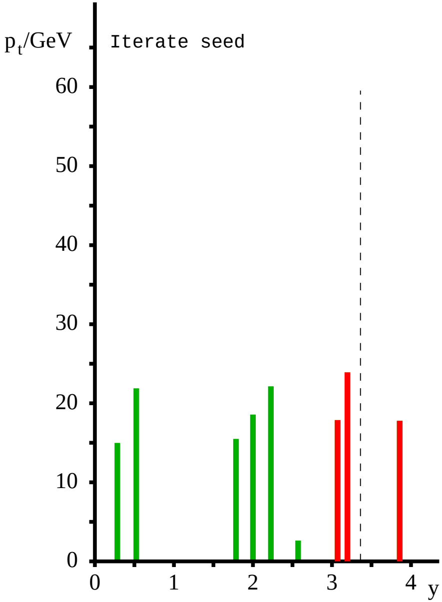
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

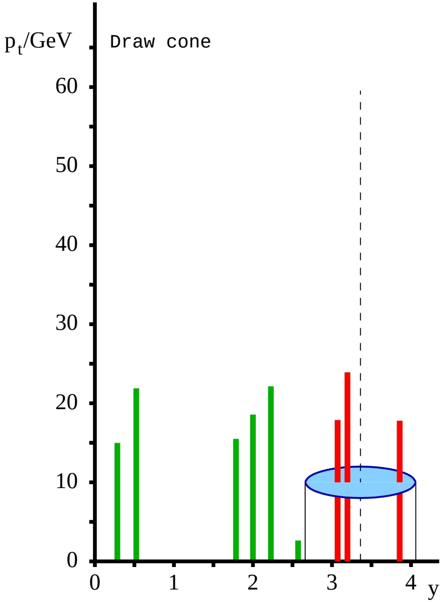
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

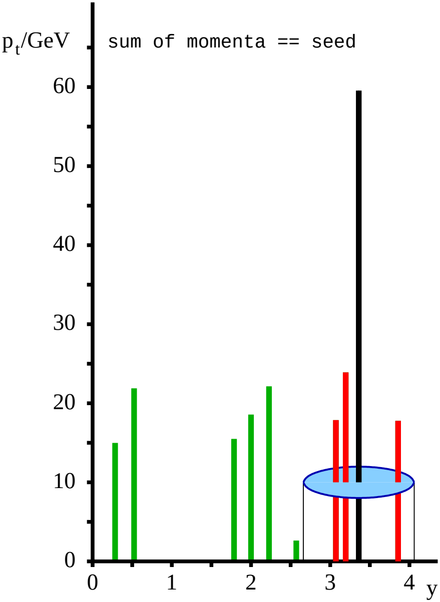
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

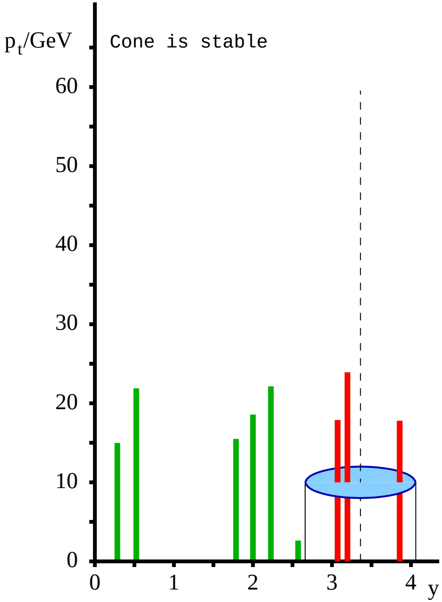
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe … more right away…

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

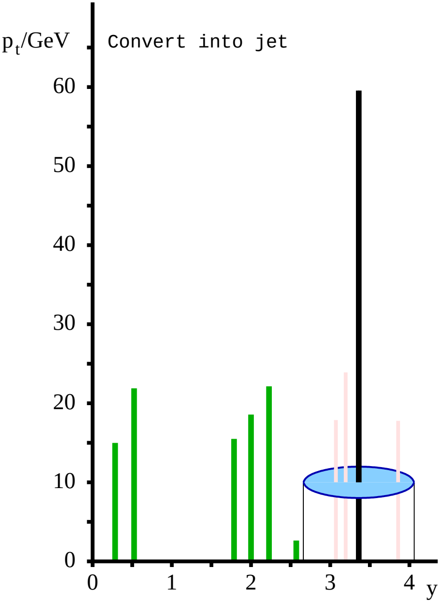
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)

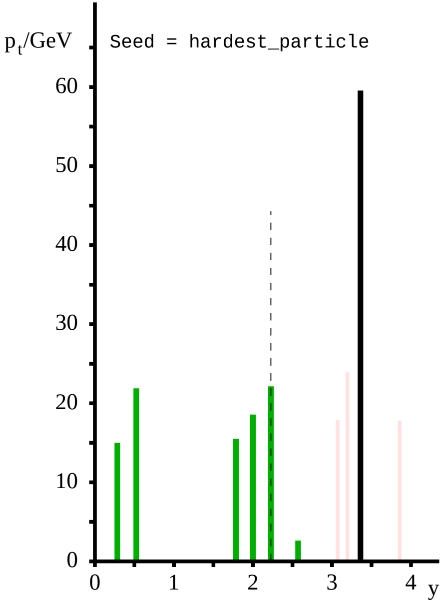


One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

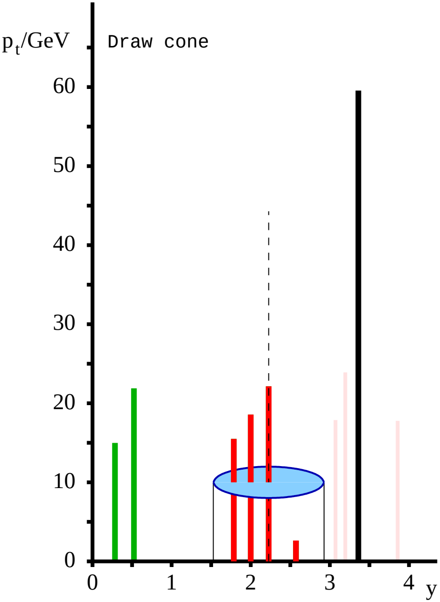
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

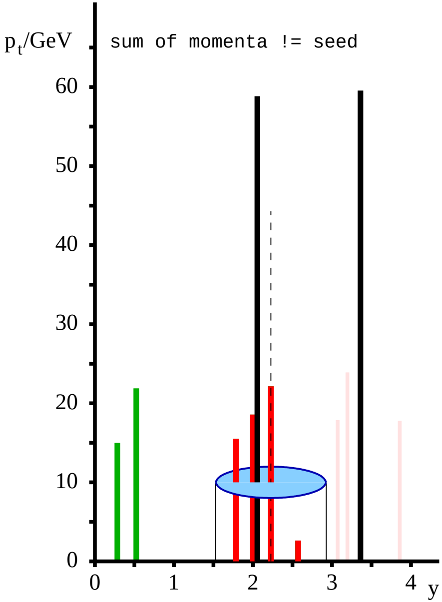
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

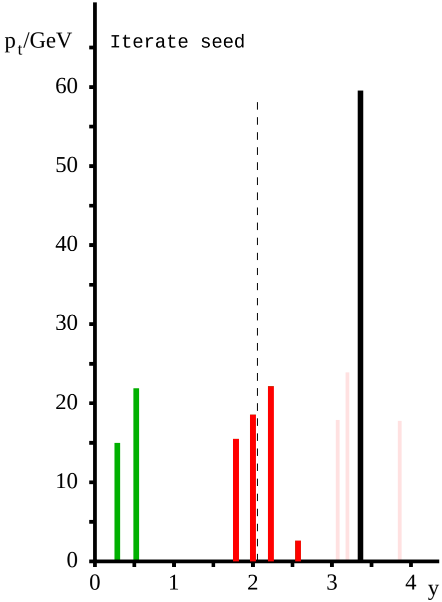
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



<u>One of the simpler cones</u>

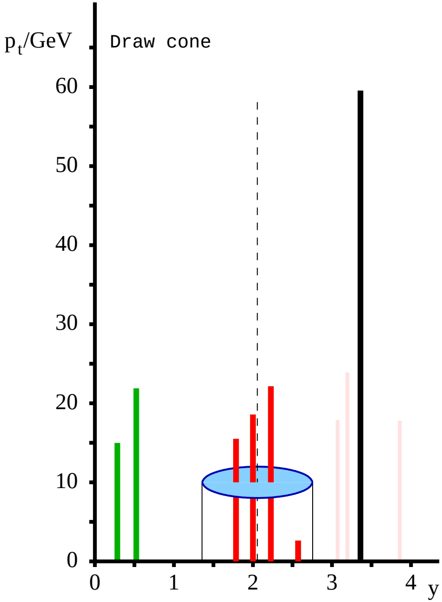
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

<u>Notes</u>

- “Hardest particle” is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

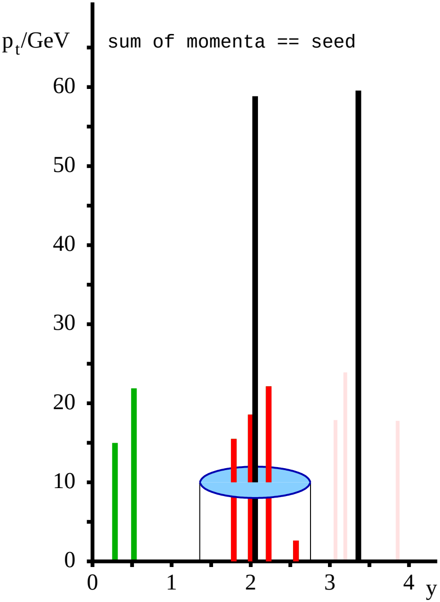
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe … more right away…

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

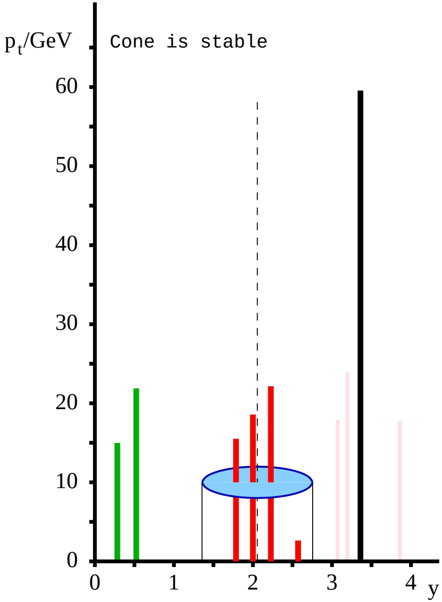
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Notes

- “Hardest particle” is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



<u>One of the simpler cones</u>

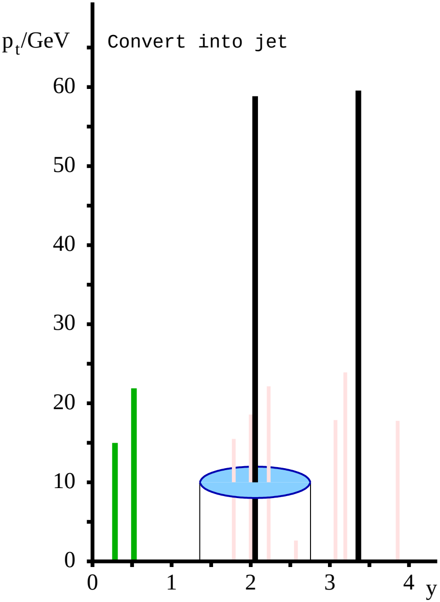
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

<u>Notes</u>

- “Hardest particle” is collinear unsafe … more right away…

Iterative Cone, Prog Removal (IC-PR)

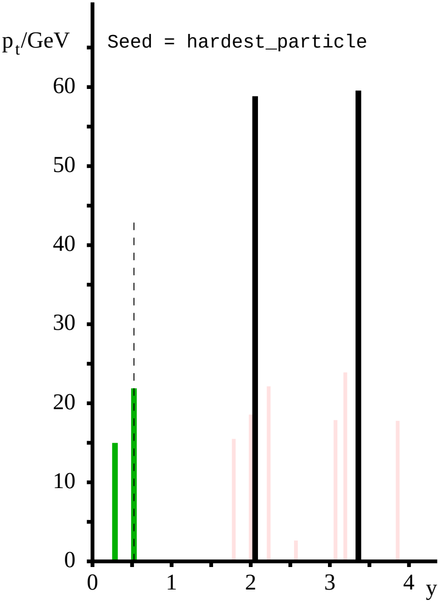


One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Iterative Cone, Prog Removal (IC-PR)

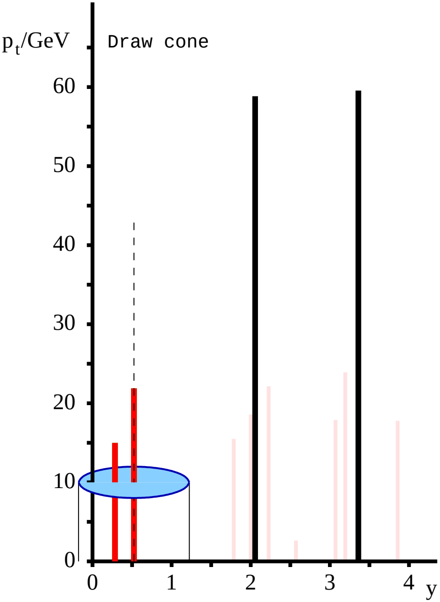


One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Iterative Cone, Prog Removal (IC-PR)

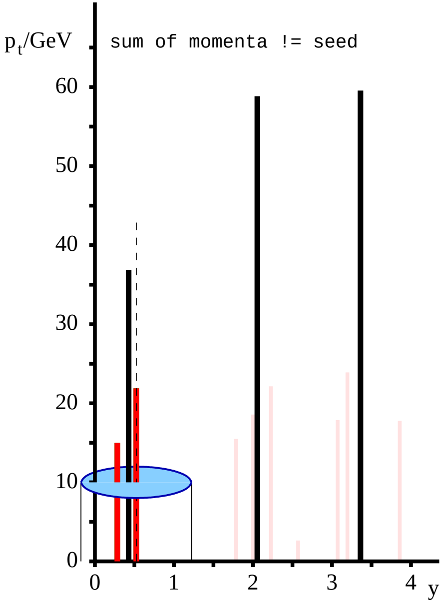


One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

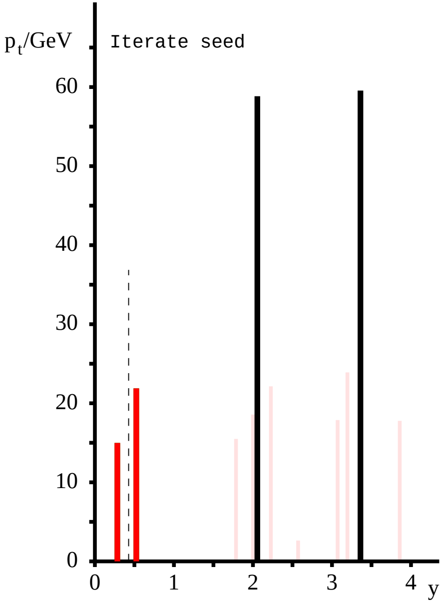
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe ... more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

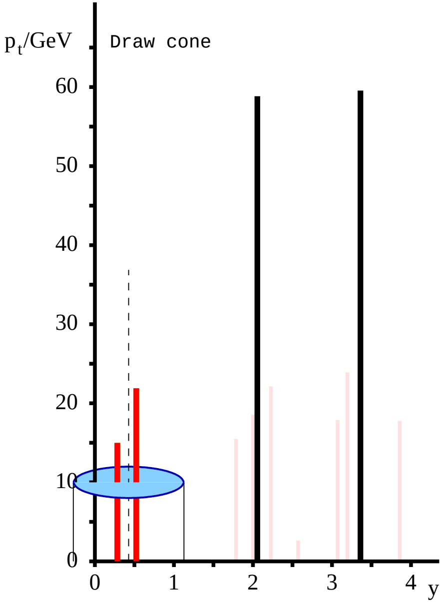
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



<u>One of the simpler cones</u>

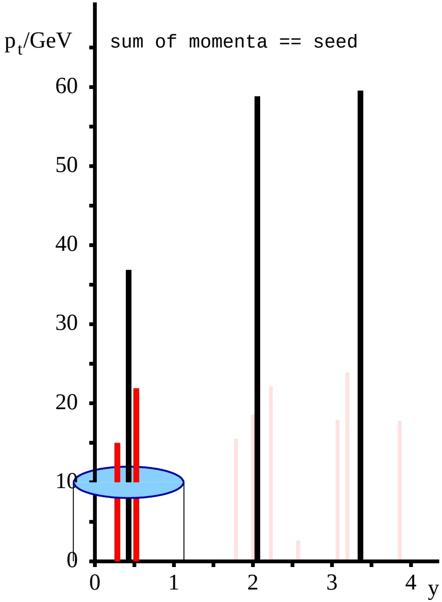
e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a “jet” and remove from event

<u>Notes</u>

- “Hardest particle” is collinear unsafe more right away...

Iterative Cone, Prog Removal (IC-PR)



One of the simpler cones

e.g. CMS iterative cone

- Take hardest particle as seed for cone axis
- Draw cone around seed
- Sum the momenta use as new seed direction, iterate until stable
- Convert contents into a "jet" and remove from event

Notes

- "Hardest particle" is collinear unsafe

more right away...

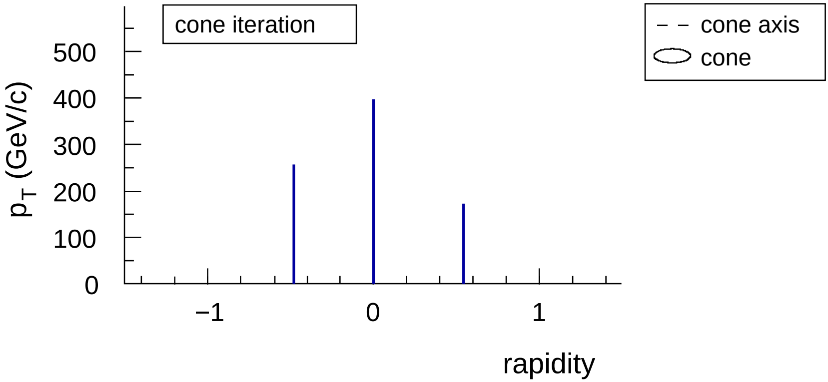
Cone Algorithms

- What's wrong with IC-PR? Any guesses?

Cone Algorithms

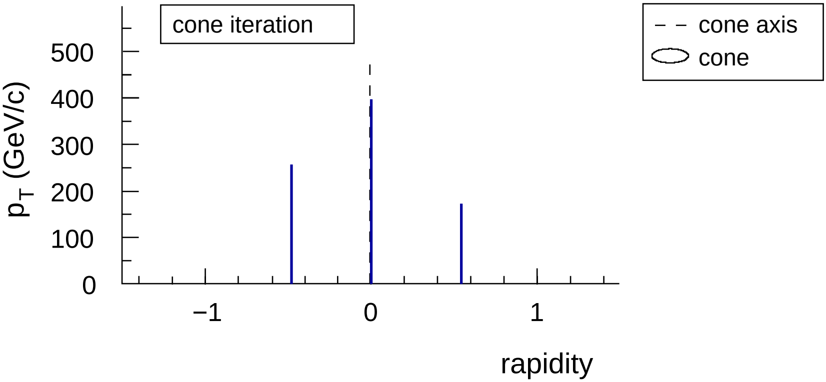
- What's wrong with IC-PR? Any guesses?
  - Not IR Safe!
  - Not collinear safe, specifically, because the initial seed particle is not collinear-safe
  - Let's see an example:

ICPR iteration issue



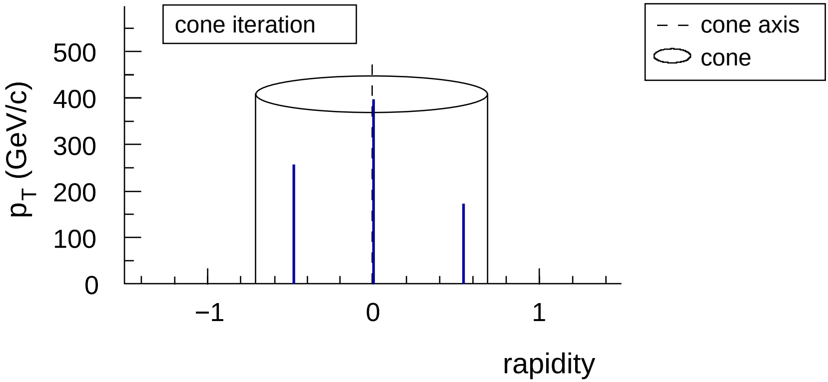
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



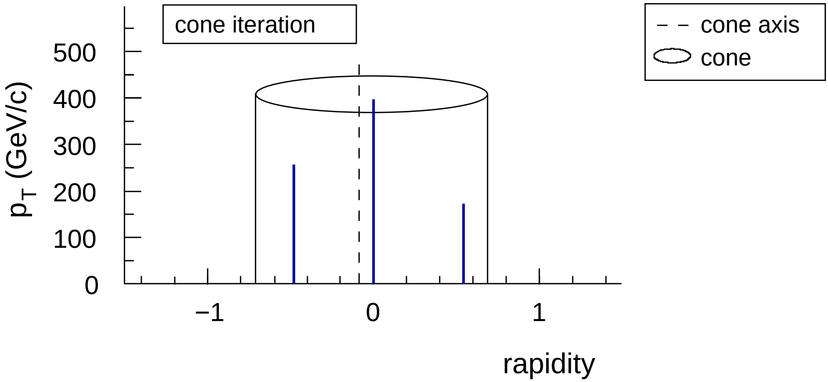
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



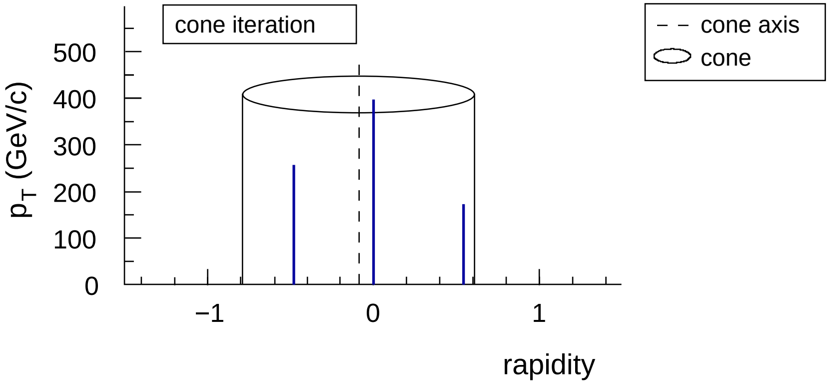
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



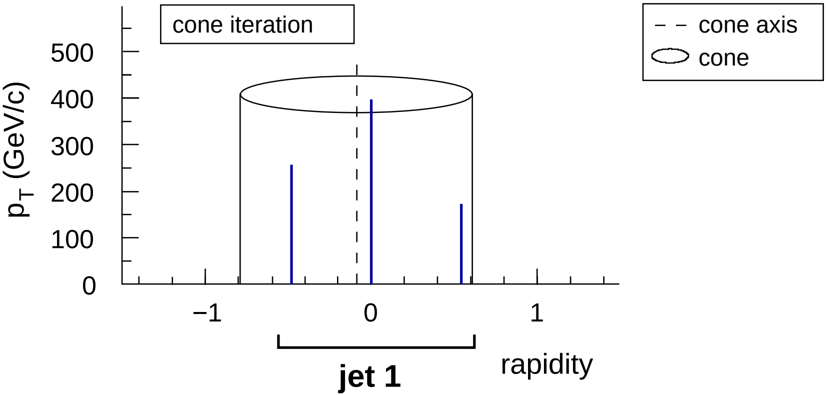
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



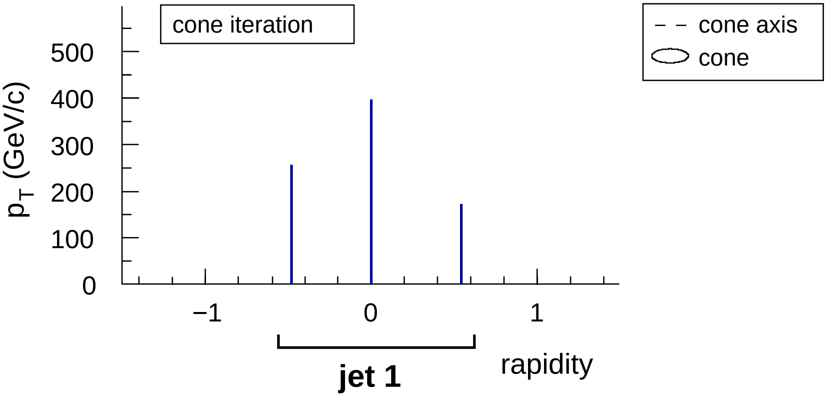
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



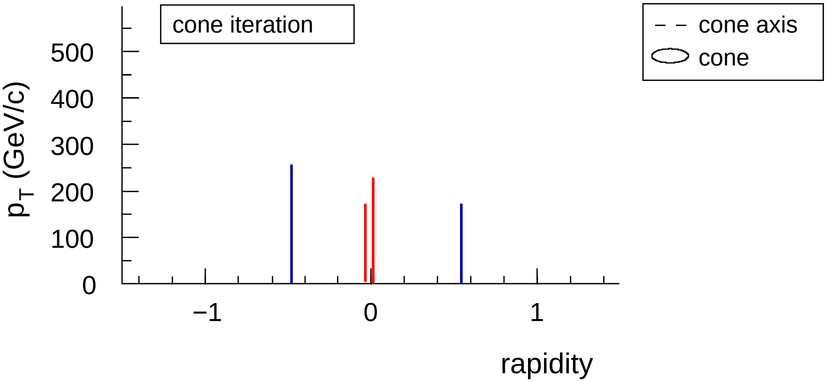
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



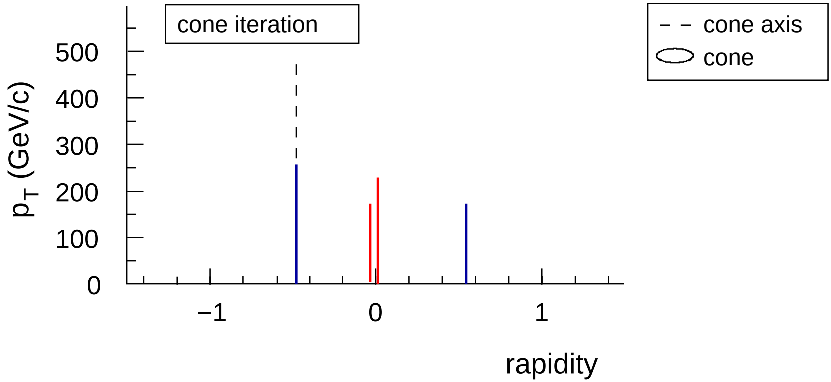
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



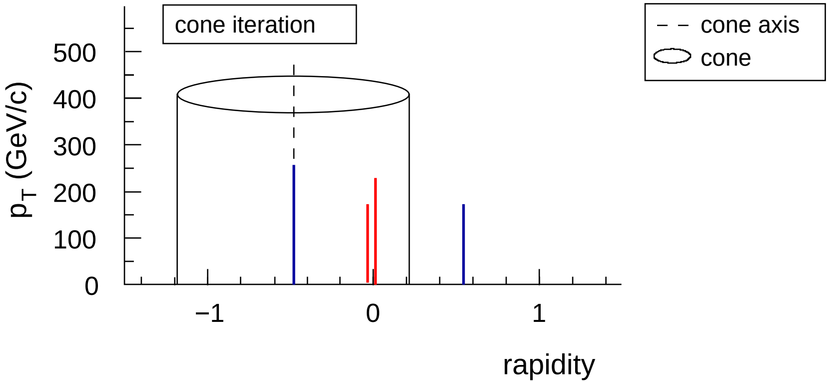
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



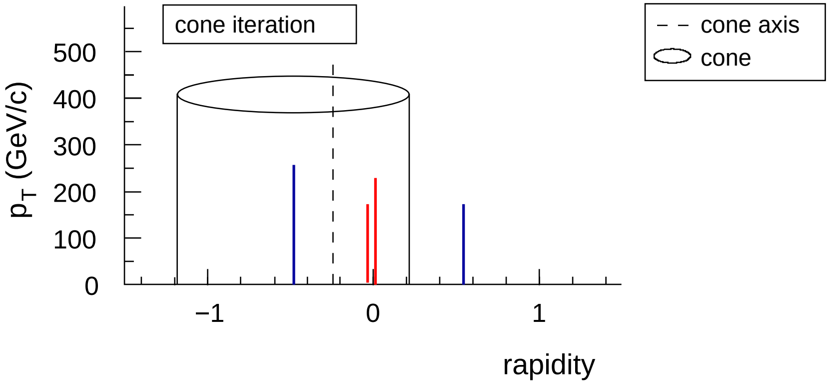
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



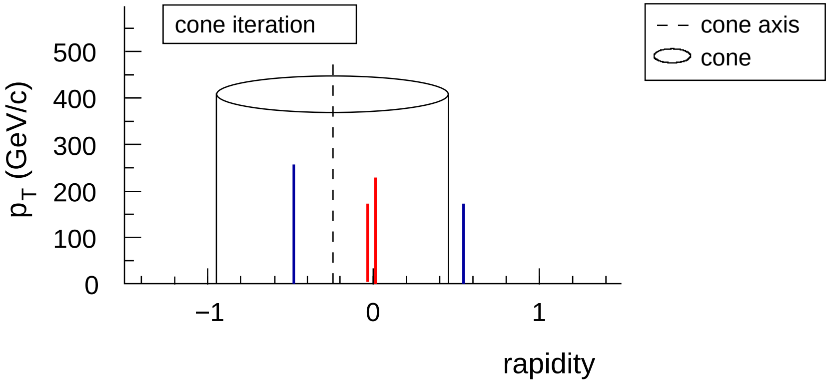
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



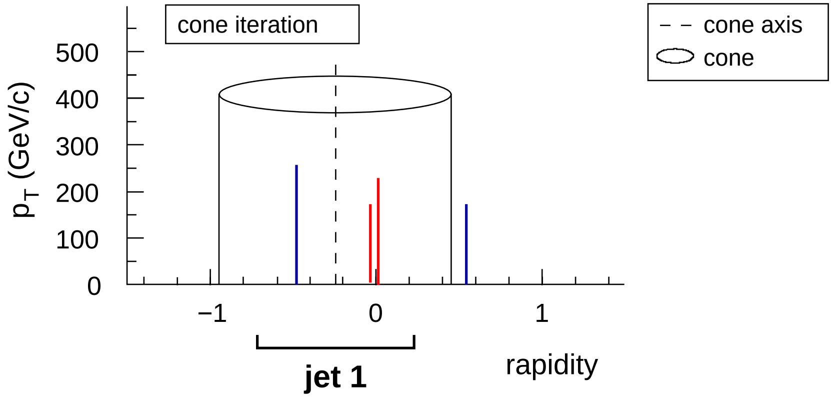
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



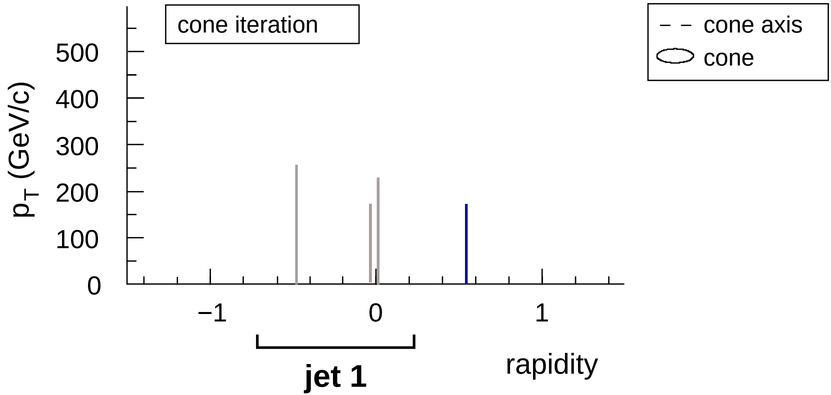
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



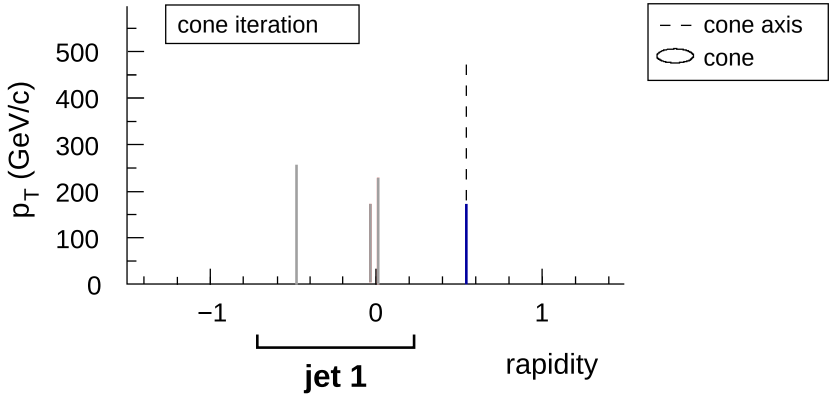
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



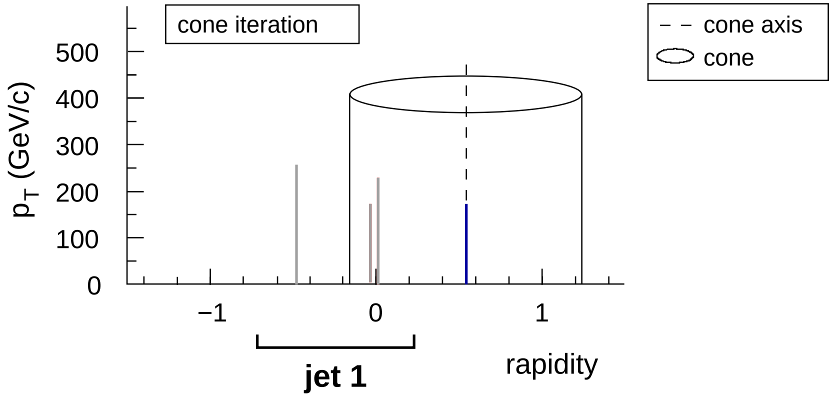
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



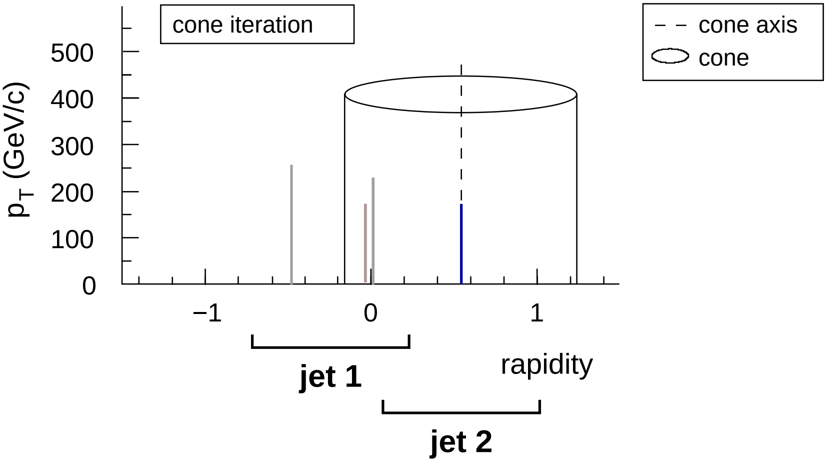
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



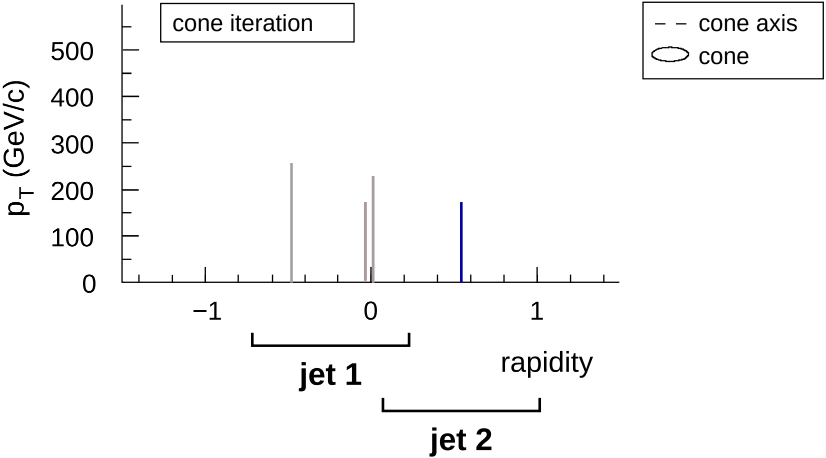
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



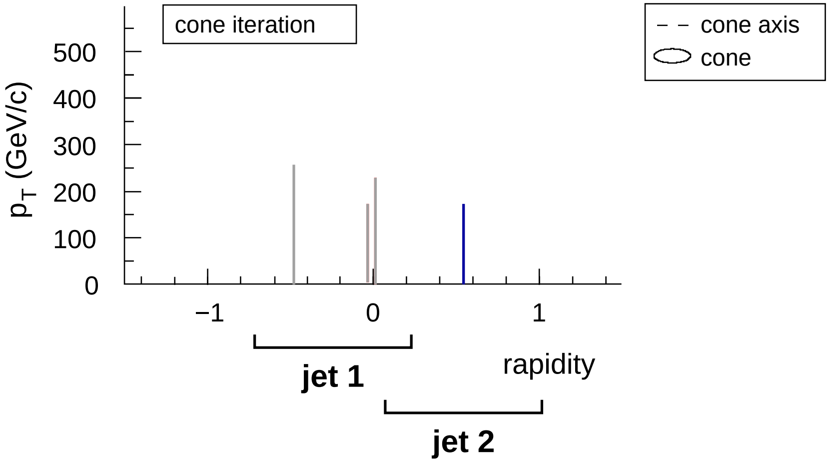
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

ICPR iteration issue



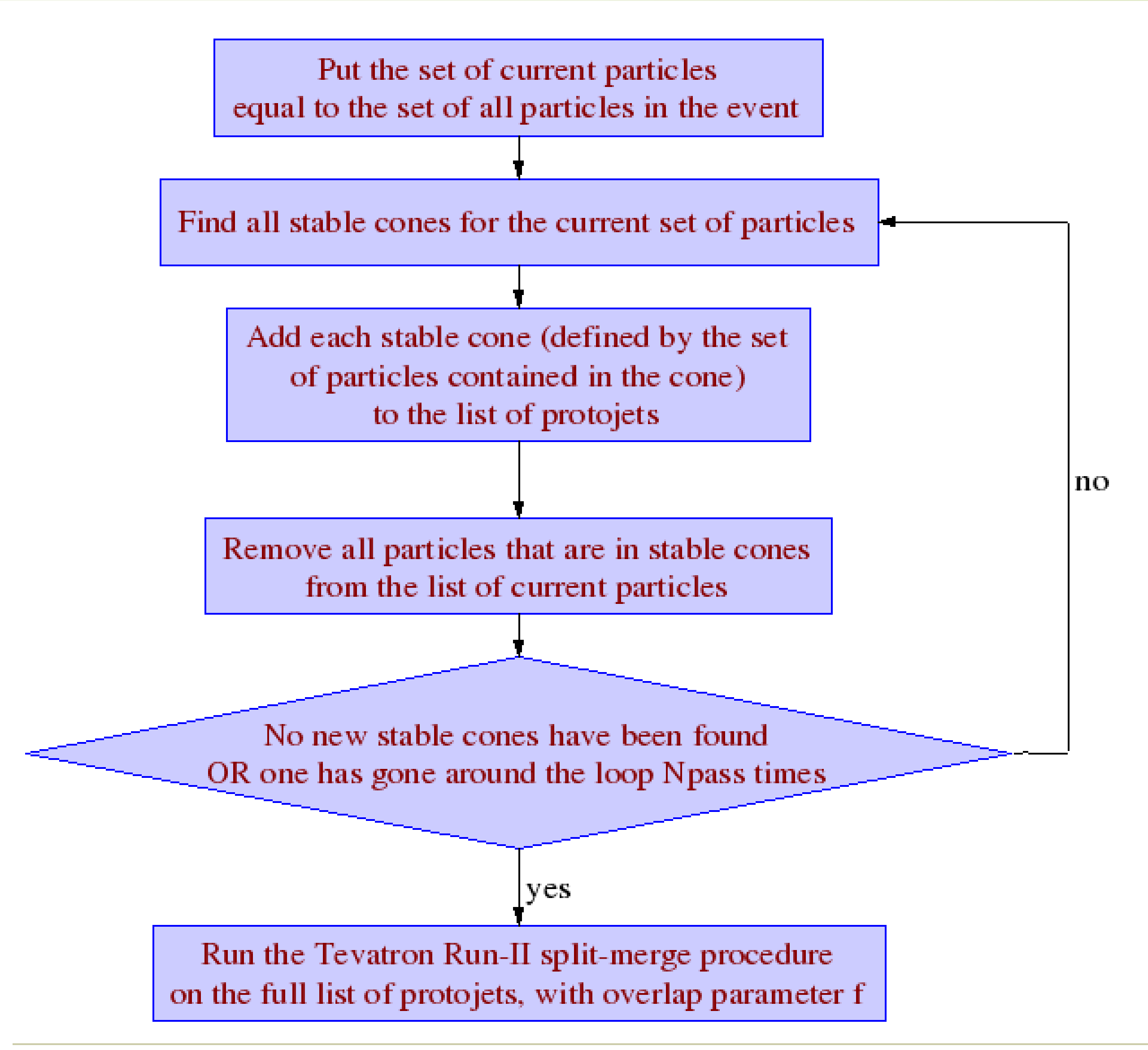
Collinear splitting can modify the hard jets: ICPR algorithms are collinear unsafe $\Longrightarrow$ perturbative calculations give ∞

Cone Algorithms

- Can we modify the ICPR algorithm to be IR safe?
- As far as I can tell, the answer was no…until 2007
- Seedless Infrared-Safe Cone (SISCone) algorithm developed by G. Salam and G. Soyez
  - A cone algorithm that finds **all** stable cones without using an IR-unsafe seed; instead it considers all subsets of particles and checks if each corresponds to a stable cone
  - Doing this by brute force would be $O(N2^N)$, but SISCone method is $O(N^2 \ln N)$

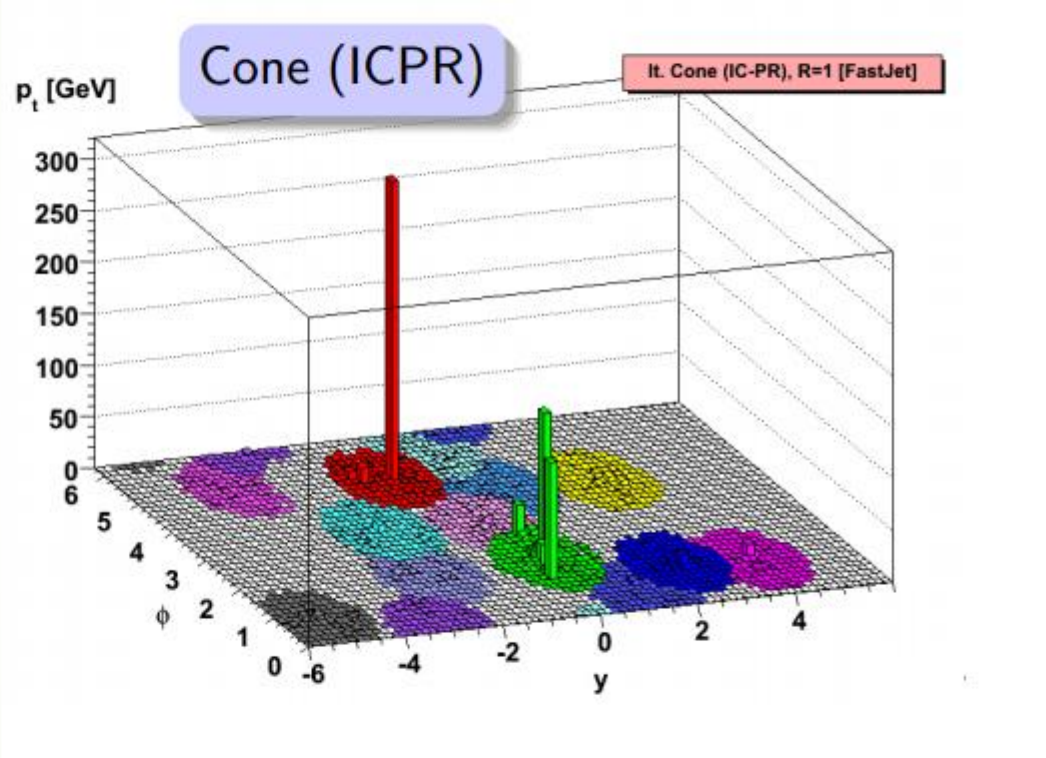
SISCone

- Rough idea of the general SISCone algorithm
- For more specifics, see Salam Soyez JHEP 05 (2007) 086
- https://siscone.hepforge.org/algorithm.html



Cone Algorithms

- What other issues are there with cone algorithms?
  - IR-unsafe → Fixed
  - Long computation time → Fixed
  - Some cone algorithms give jets in y-ϕ plane that are not circular
  - Why do jets need to be circular? Useful for acceptance corrections, modeling backgrounds



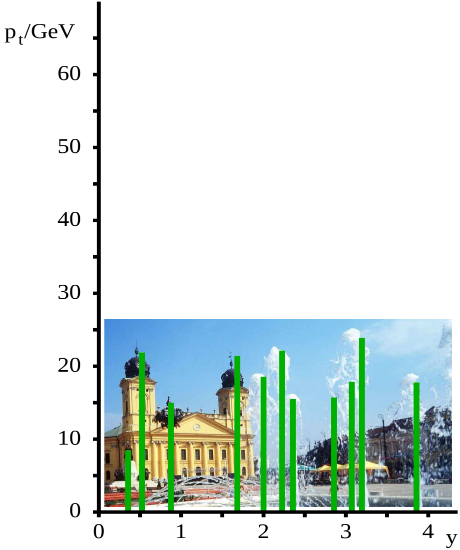
Sequential Clustering Algorithms

- Sequential clustering algorithms (also called sequential recombination algorithms) work by comparing pairs of particles and combining them recursively, according to given rules

Sequential Clustering Algorithms

- Inclusive k_T algorithm
  - Developed by S.D. Ellis and Soper, 1993
  - Choose some "cutoff" angular radius R
  - Go through each pair of events
  - Define $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$
  - Define $d_{ij} = \min(p_{ti}^2, p_{tj}^2)\ \frac{\Delta R_{ij}^2}{R^2}$
  - Define $d_{iB} = p_{ti}^2$
  - Compare d_{ij}, d_{iB}
    - If $d_{ij} < d_{iB}$, combine the particles
    - If $d_{ij} > d_{iB}$, call particle i a jet and remove from particle list
    - Repeat until no particles left
- Example:

Sequential recombination



$\mathbf{k_t}$ alg.: Find smallest of

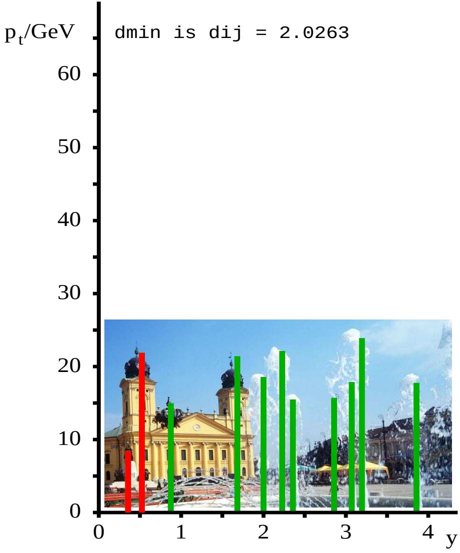
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



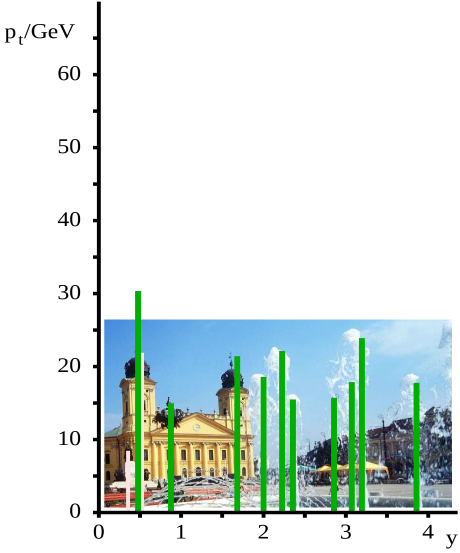
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



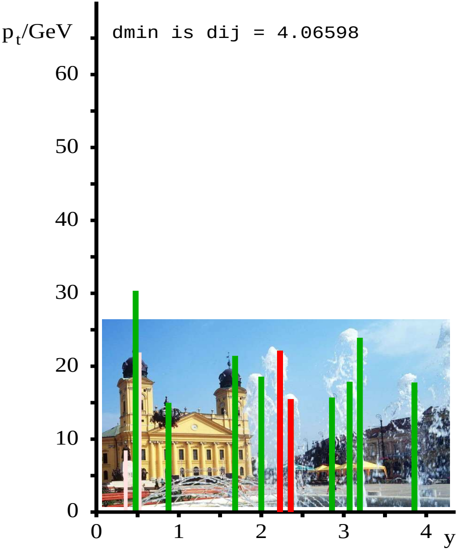
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



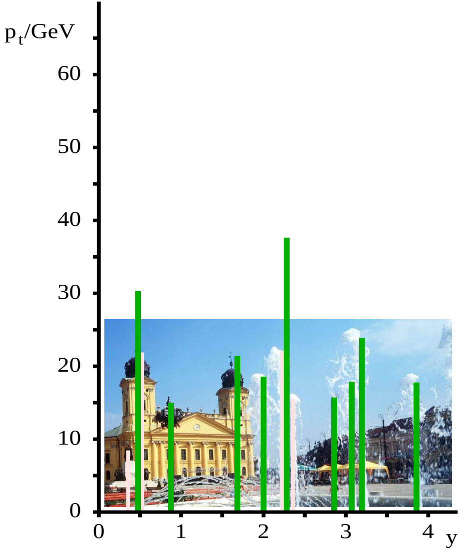
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

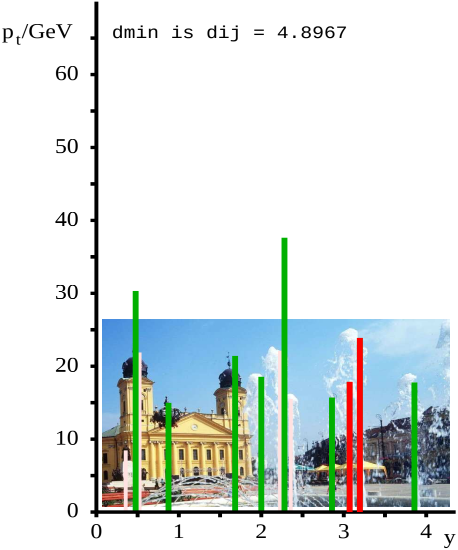
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

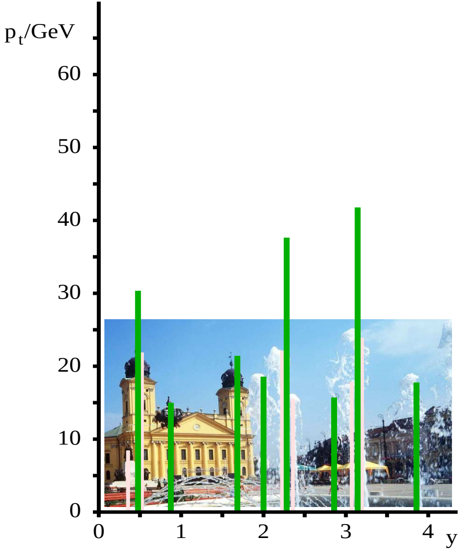
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



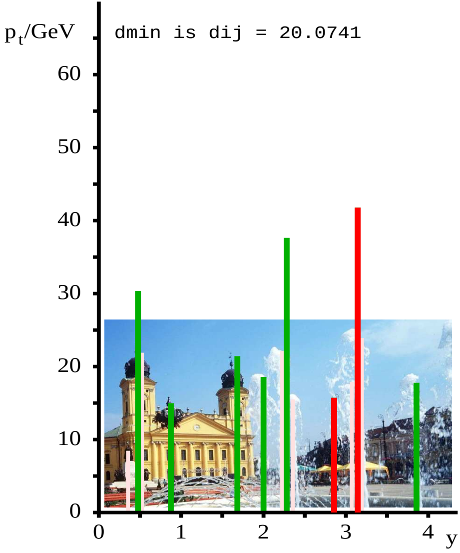
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

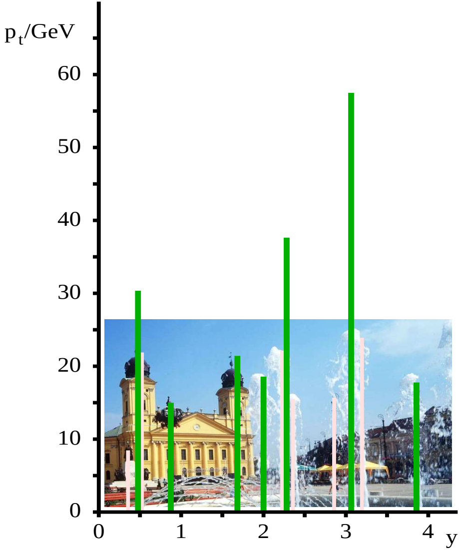
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



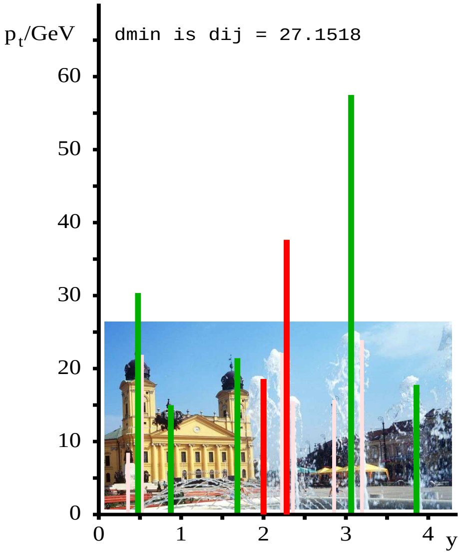
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



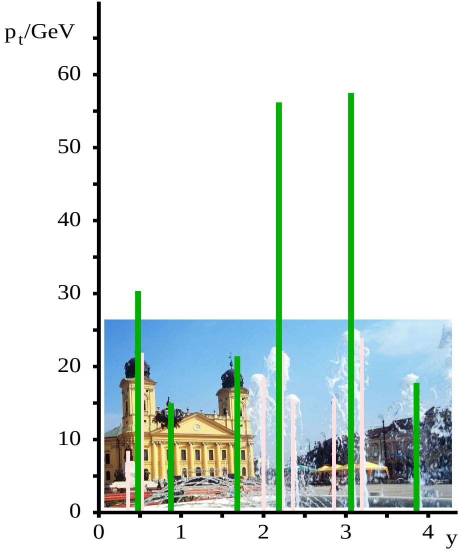
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

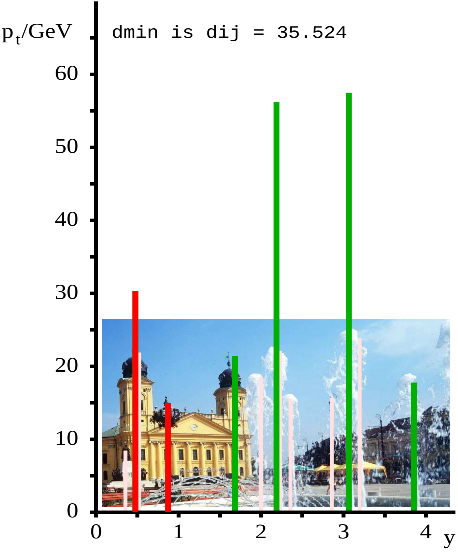
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

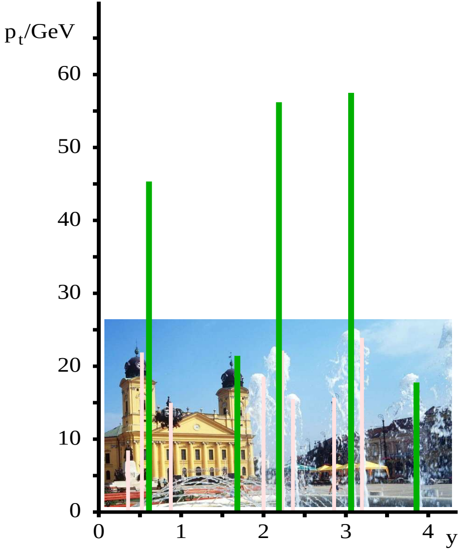
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



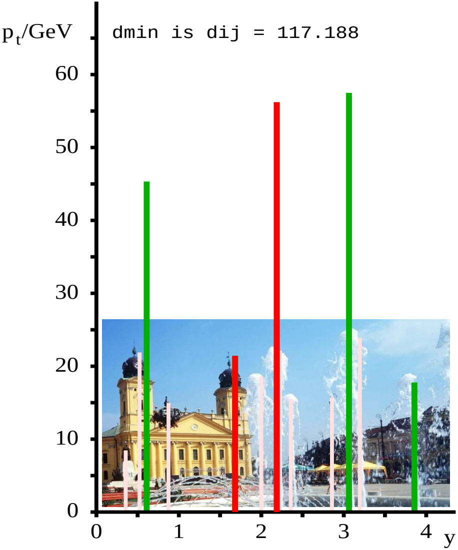
$\mathbf{k_t}$ alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



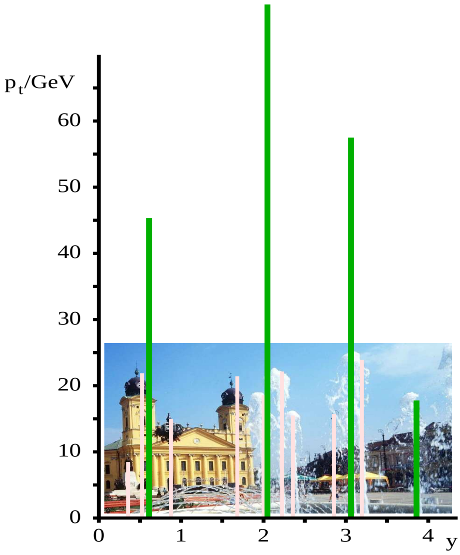
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

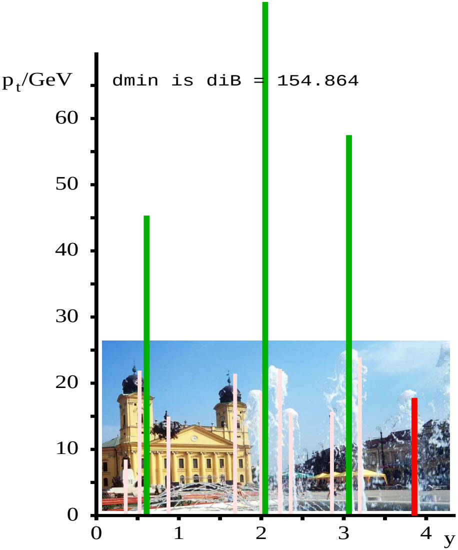
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



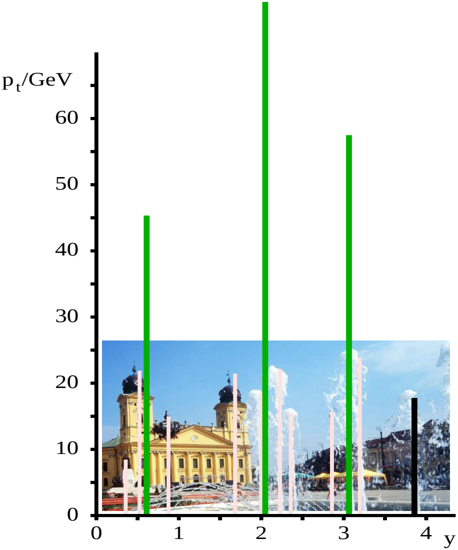
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



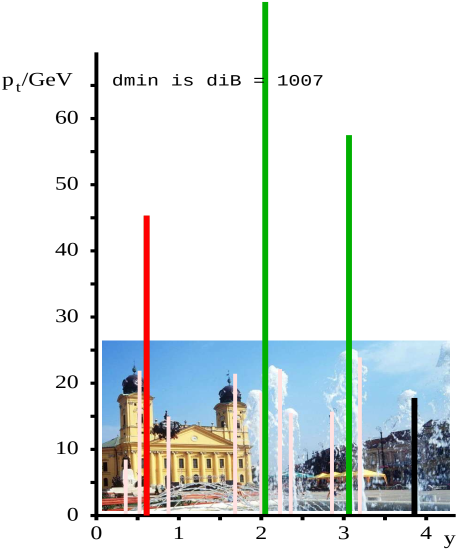
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



$\mathbf{k_t}$ alg.: Find smallest of

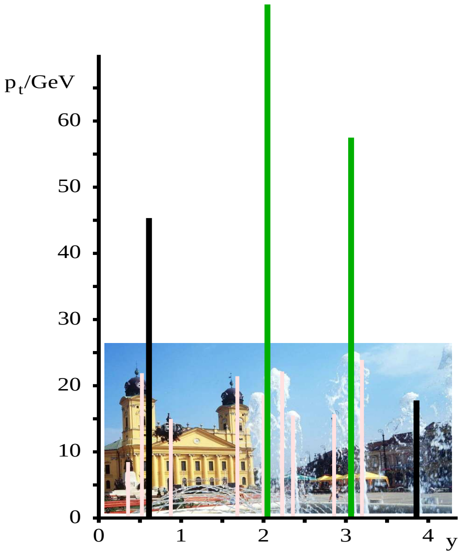
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



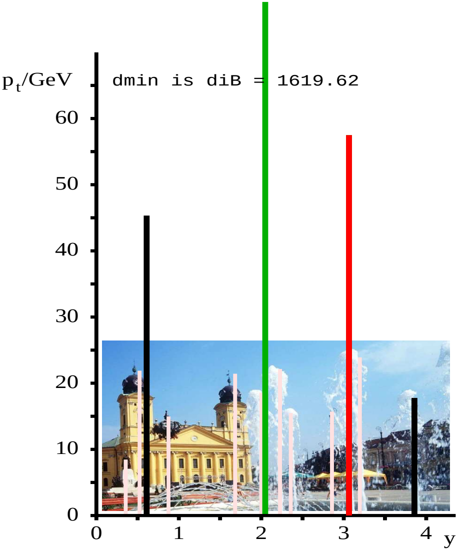
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



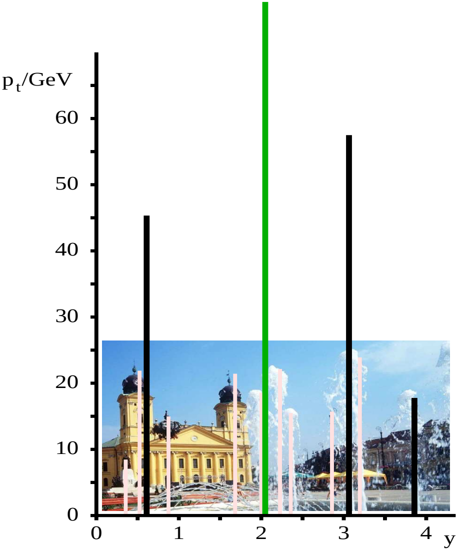
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



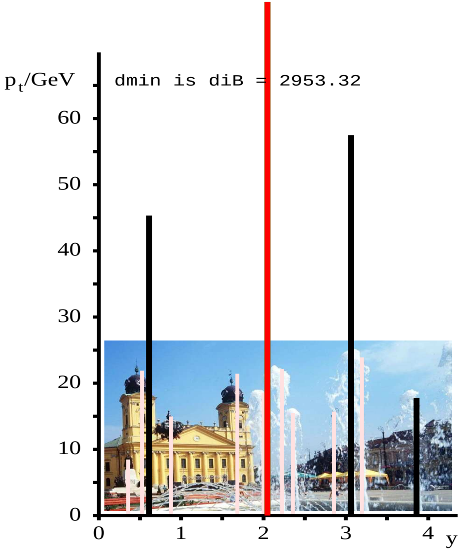
k$_t$ alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



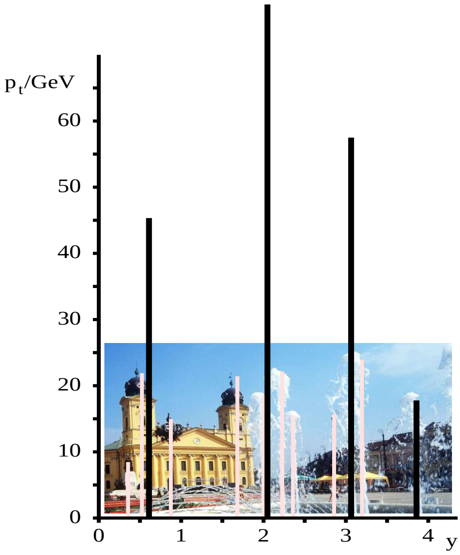
k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination



k_t alg.: Find smallest of

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

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential Clustering Algorithms

- Comments on Inclusive k_T algorithm
  - Implicitly inverting branching process by pairing up particles with the strongest divergence between them
  - Jets all separated by at least R on y, ϕ cylinder
  - Number of jets is NOT IR-safe (there could be soft jets near the beam), but the number of jets above the p_T cut is IR safe
  - Used to be slow $O(N^3)$, but use of computational geometry methods has reduced time to $O(N \ln N)$
  - Jet boundaries are generally not circular; they have irregular shapes

Sequential Clustering Algorithms

- Modifications to Inclusive k_T algorithm
- Cambridge/Aachen method
  - Again, define $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$
  - Recombine pair of objects closest in ΔR_{ij}, repeat until all $\Delta R_{ij} > R$, whatever remains are jets
  - Privileges the collinear divergence at the expense of the soft divergence
  - Non-circular jet boundaries

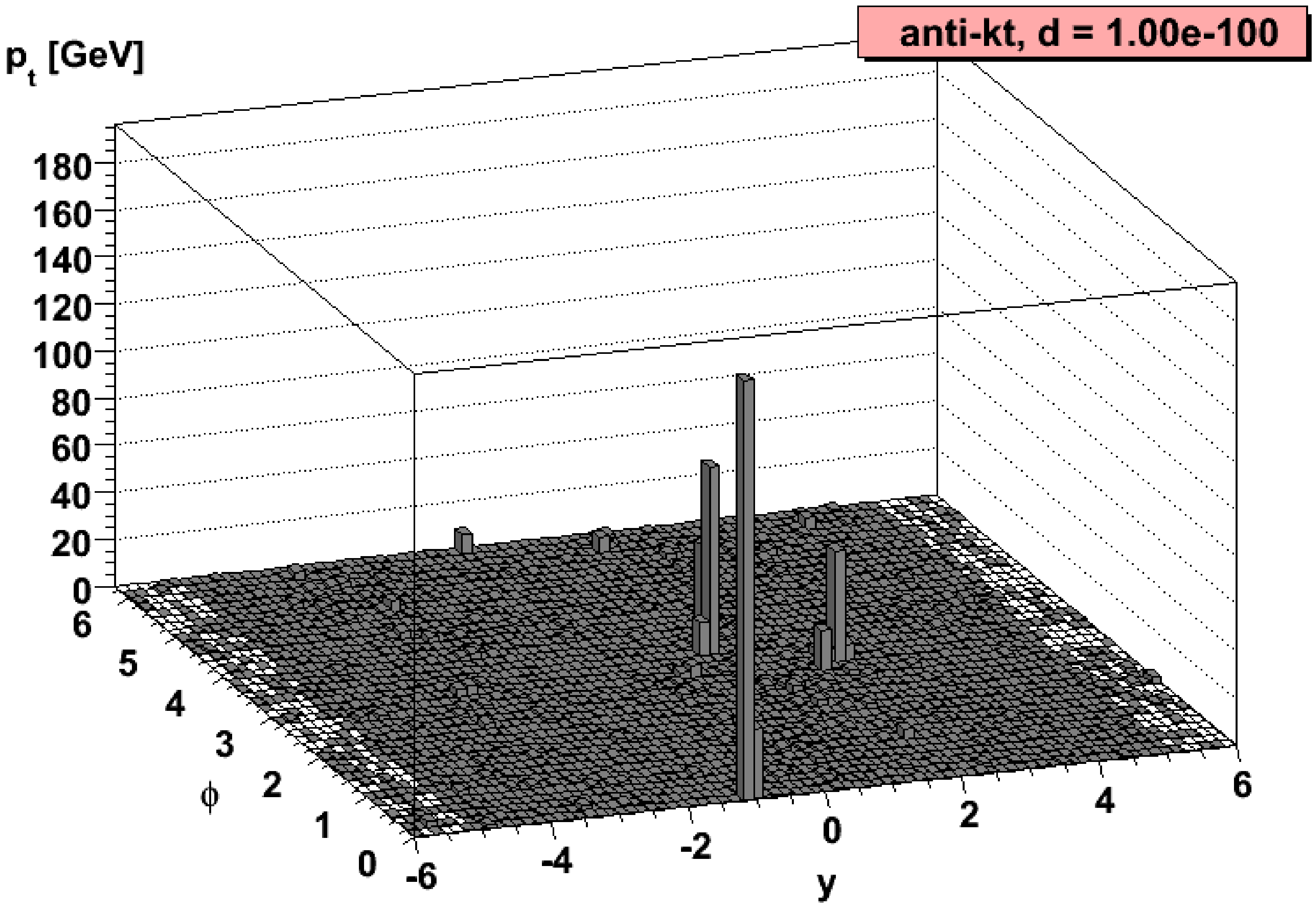
Sequential Clustering Algorithms

- Modifications to Inclusive k_T algorithm
- Anti-k_T method
  - Again, define $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$
  - Define $d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}$
  - Define $d_{iB} = \frac{1}{p_{ti}^2}$
  - Compare d_{ij}, d_{iB}
    - If $d_{ij} < d_{iB}$, combine the particles
    - If $d_{ij} > d_{iB}$, call particle i a jet and remove from particle list
    - Repeat until no particles left
  - Again privileges collinear divergences over soft divergences
  - Clusterings are centered on hard (high-energy) particles
  - Jets are cone-like, and have circular boundaries!

Anti-k$_t$ in action

Clustering grows around hard cores

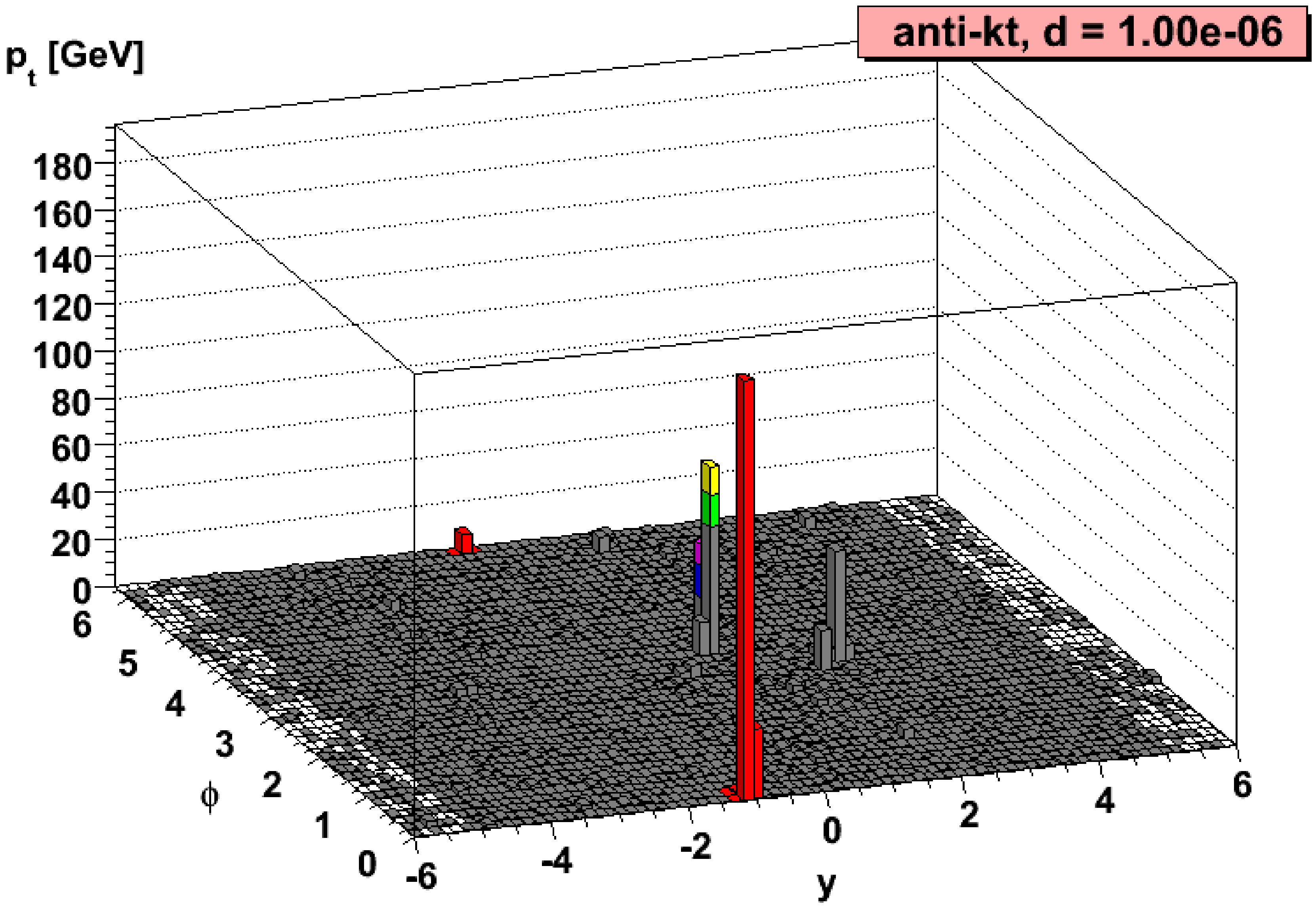
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

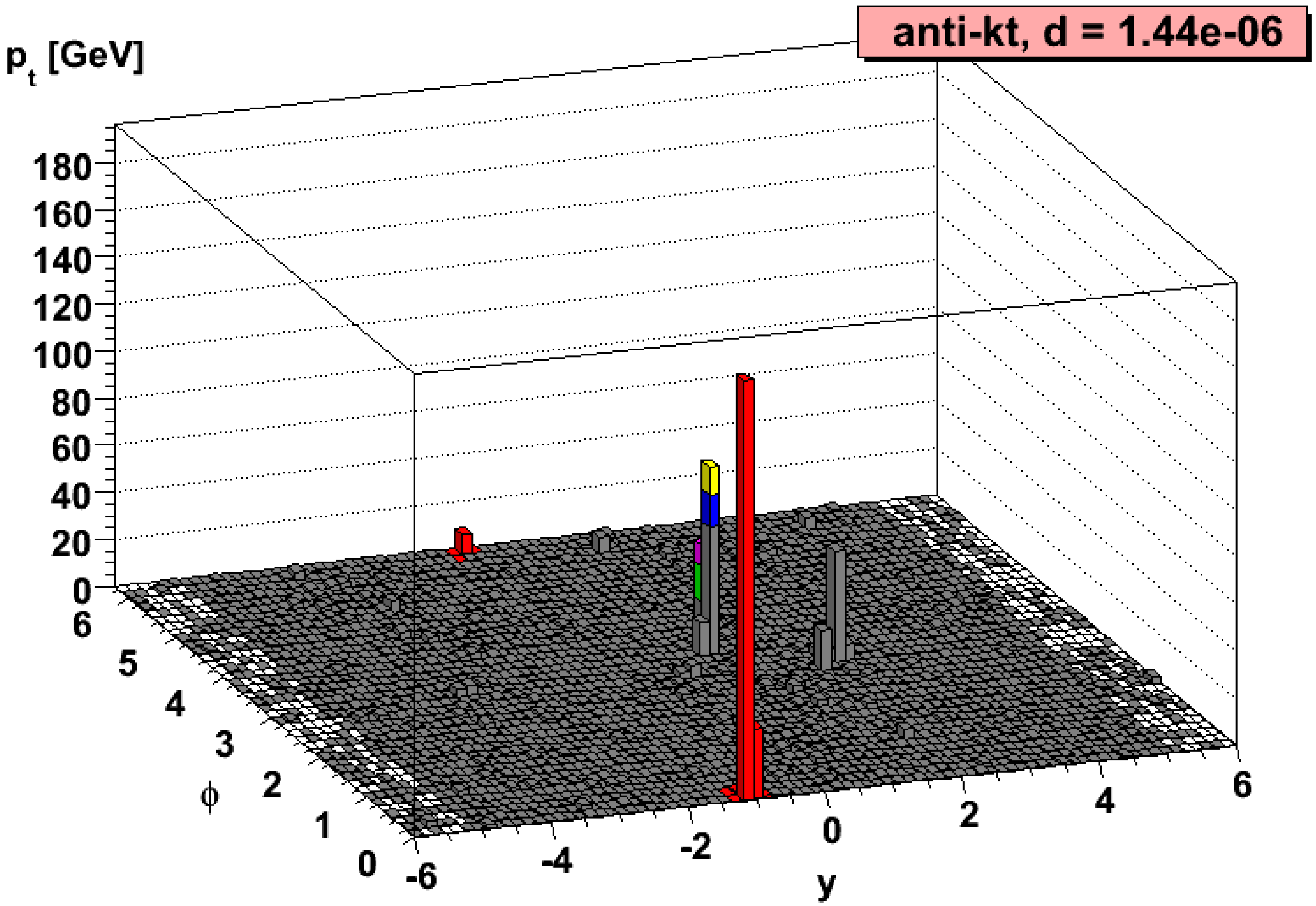
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k$_t$ in action

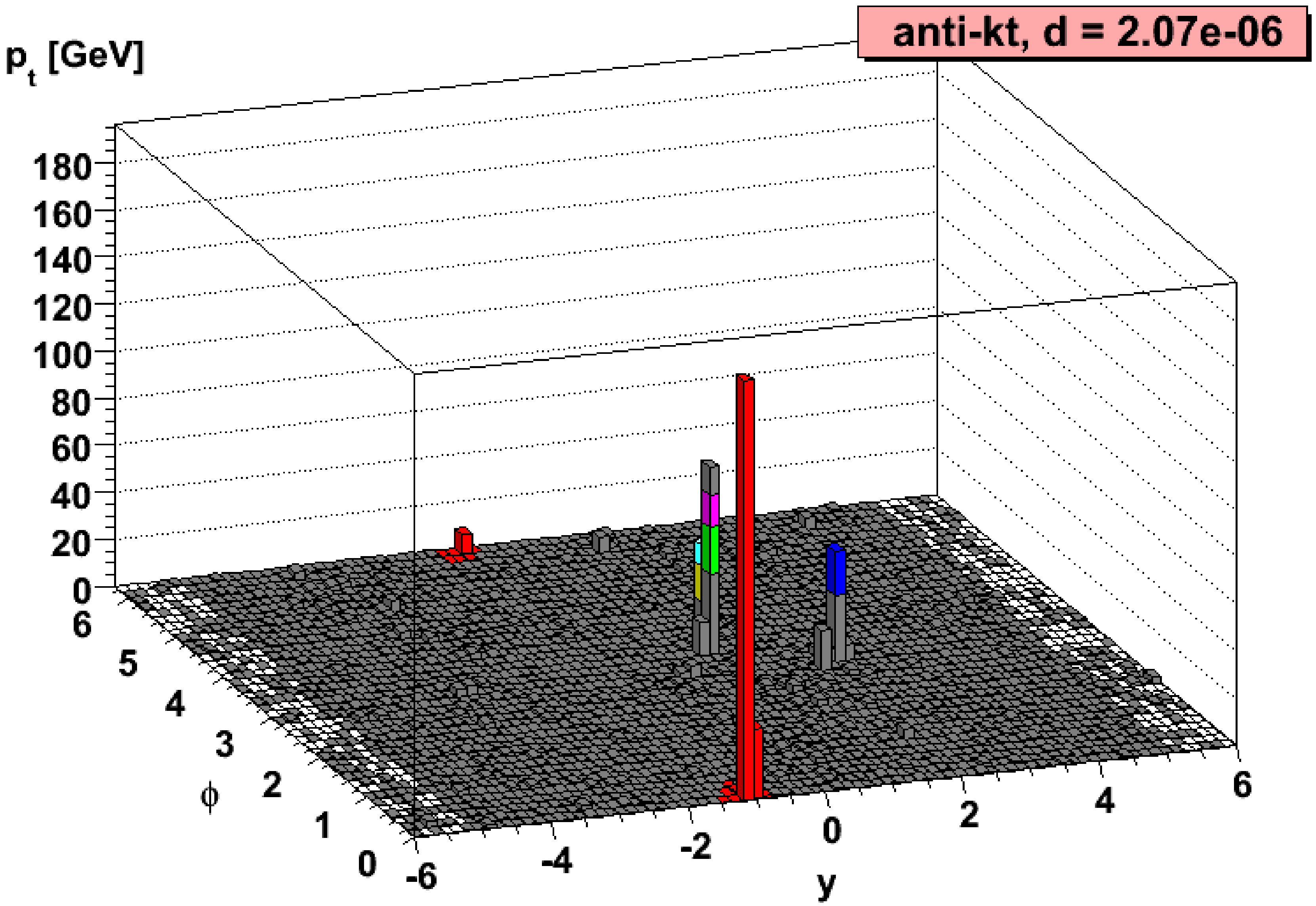
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



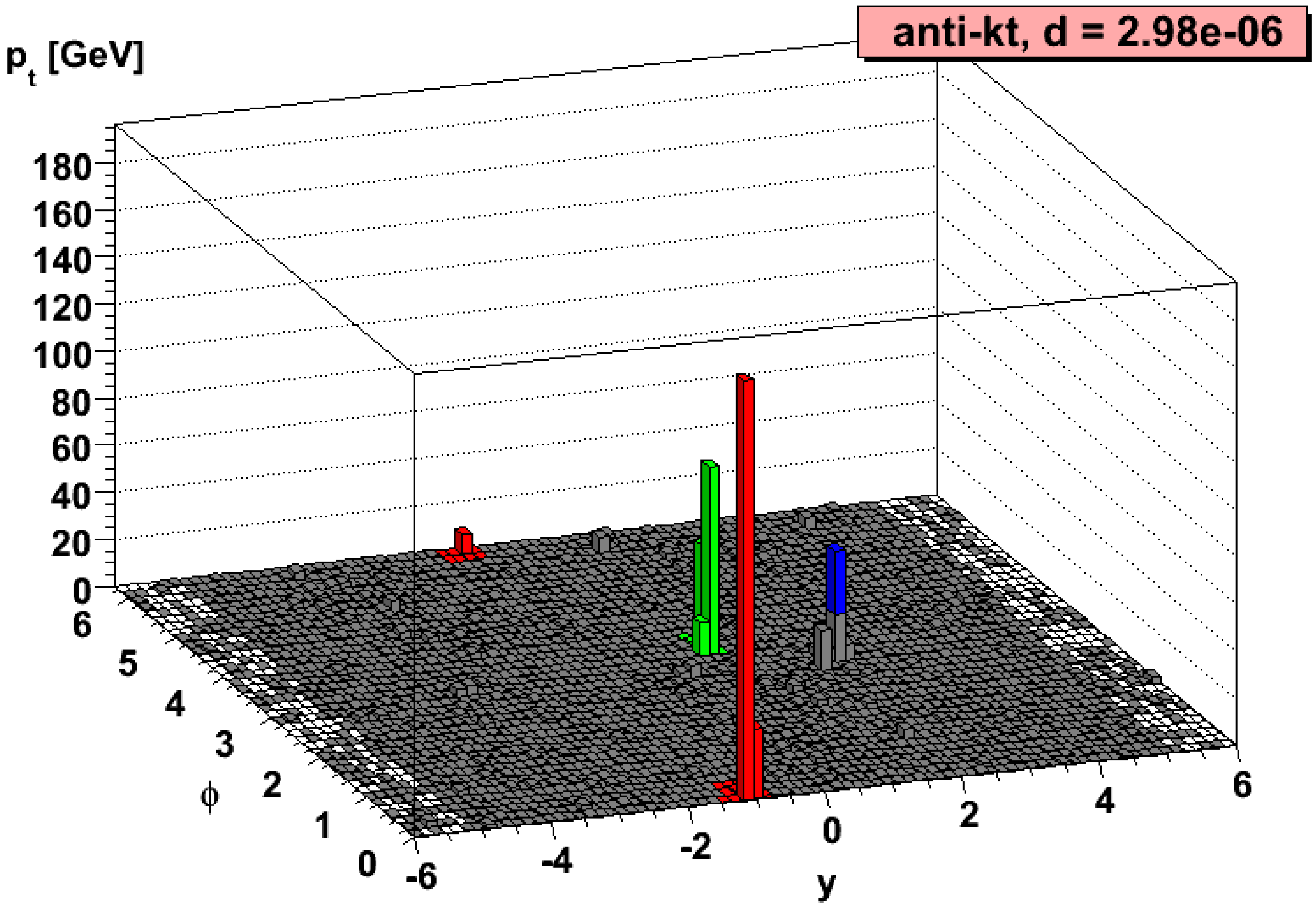
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

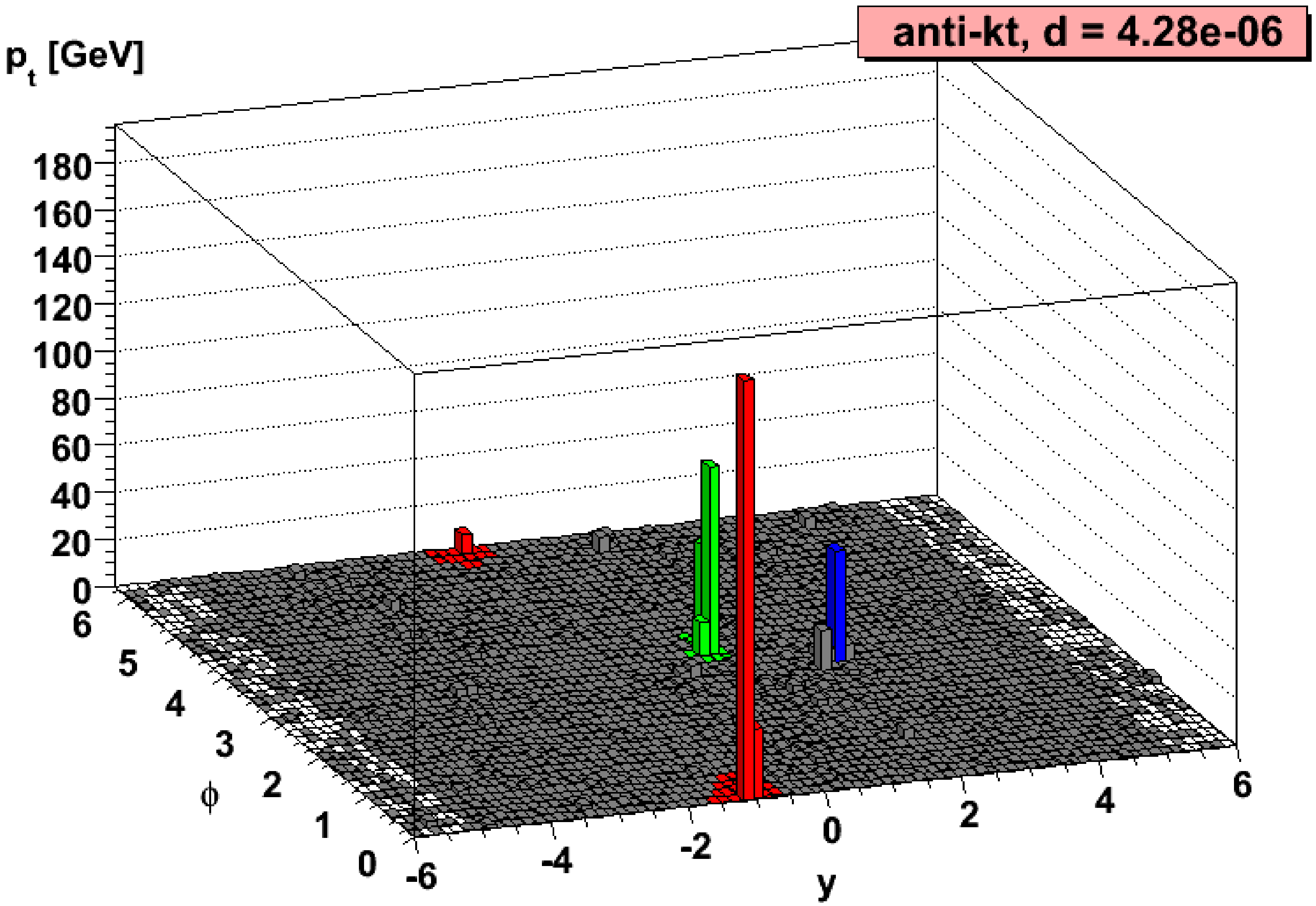
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

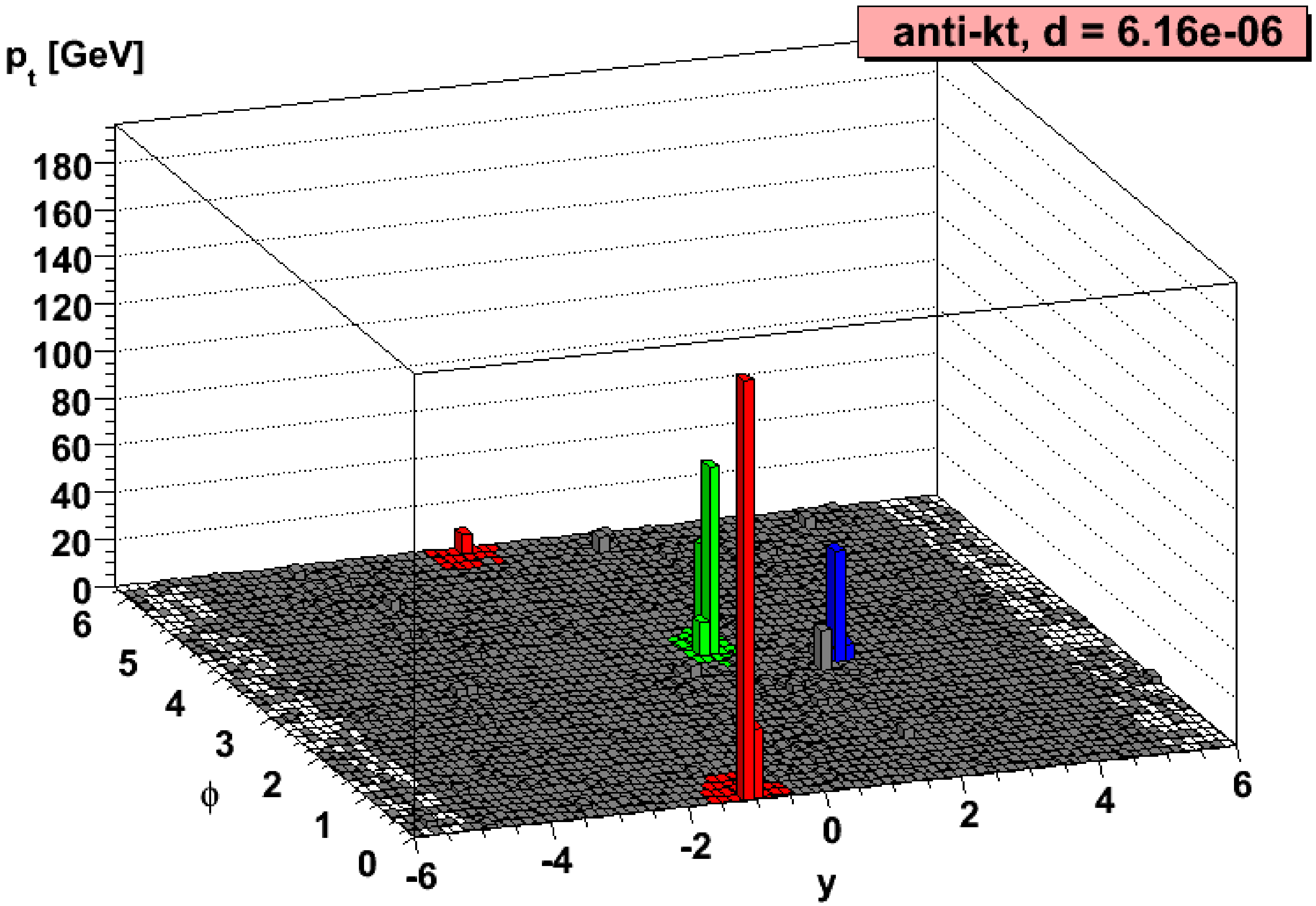
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

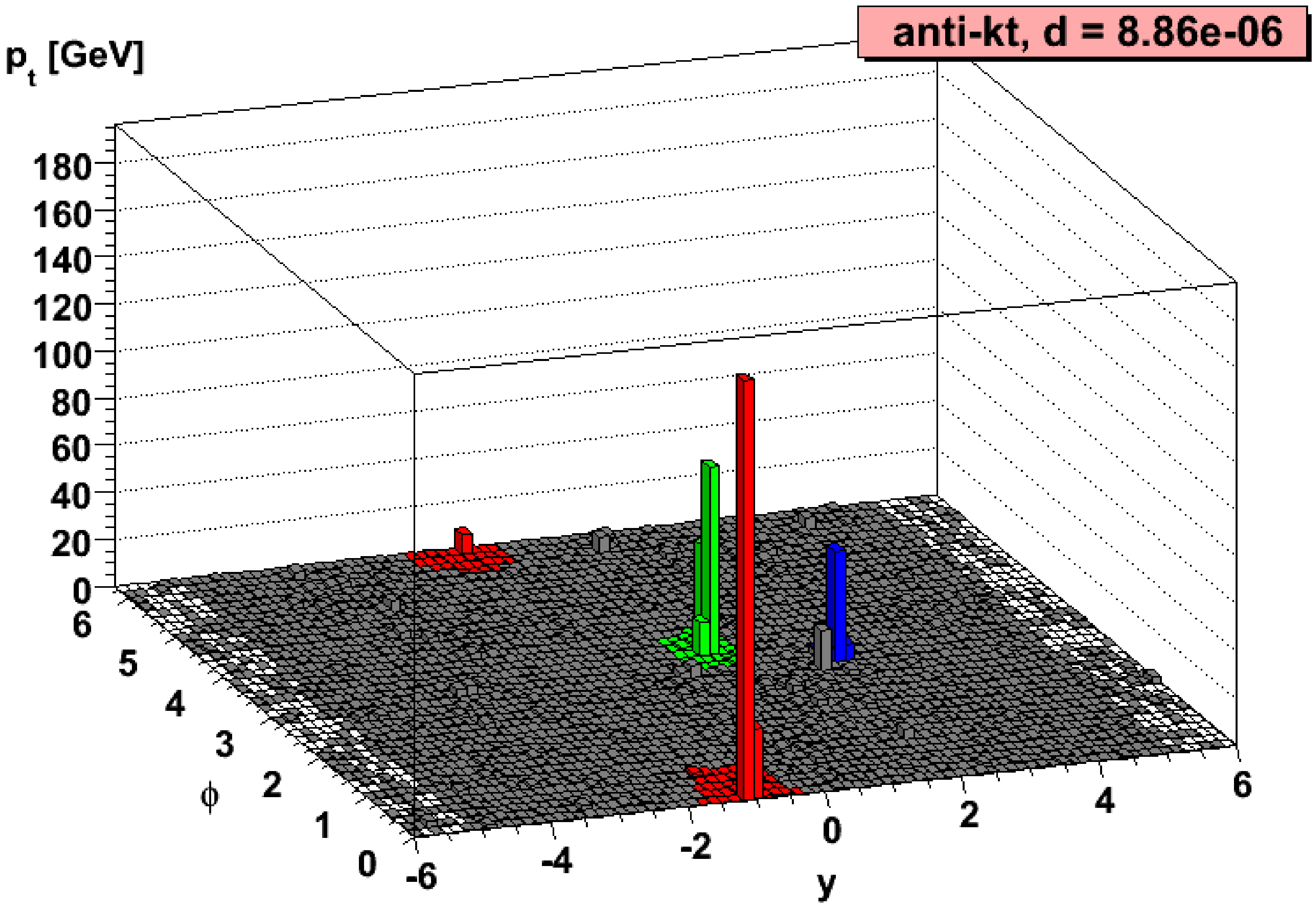
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

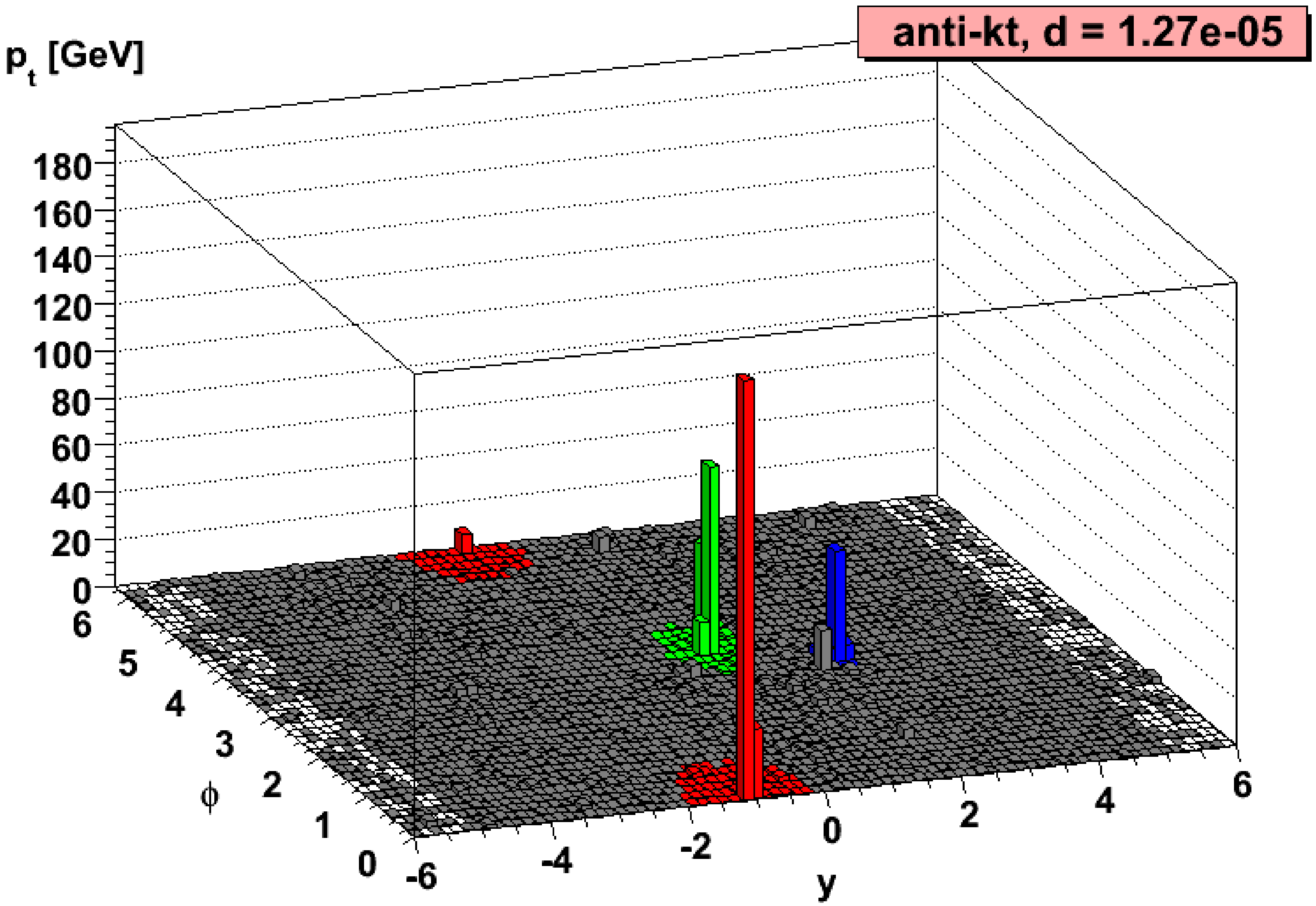
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

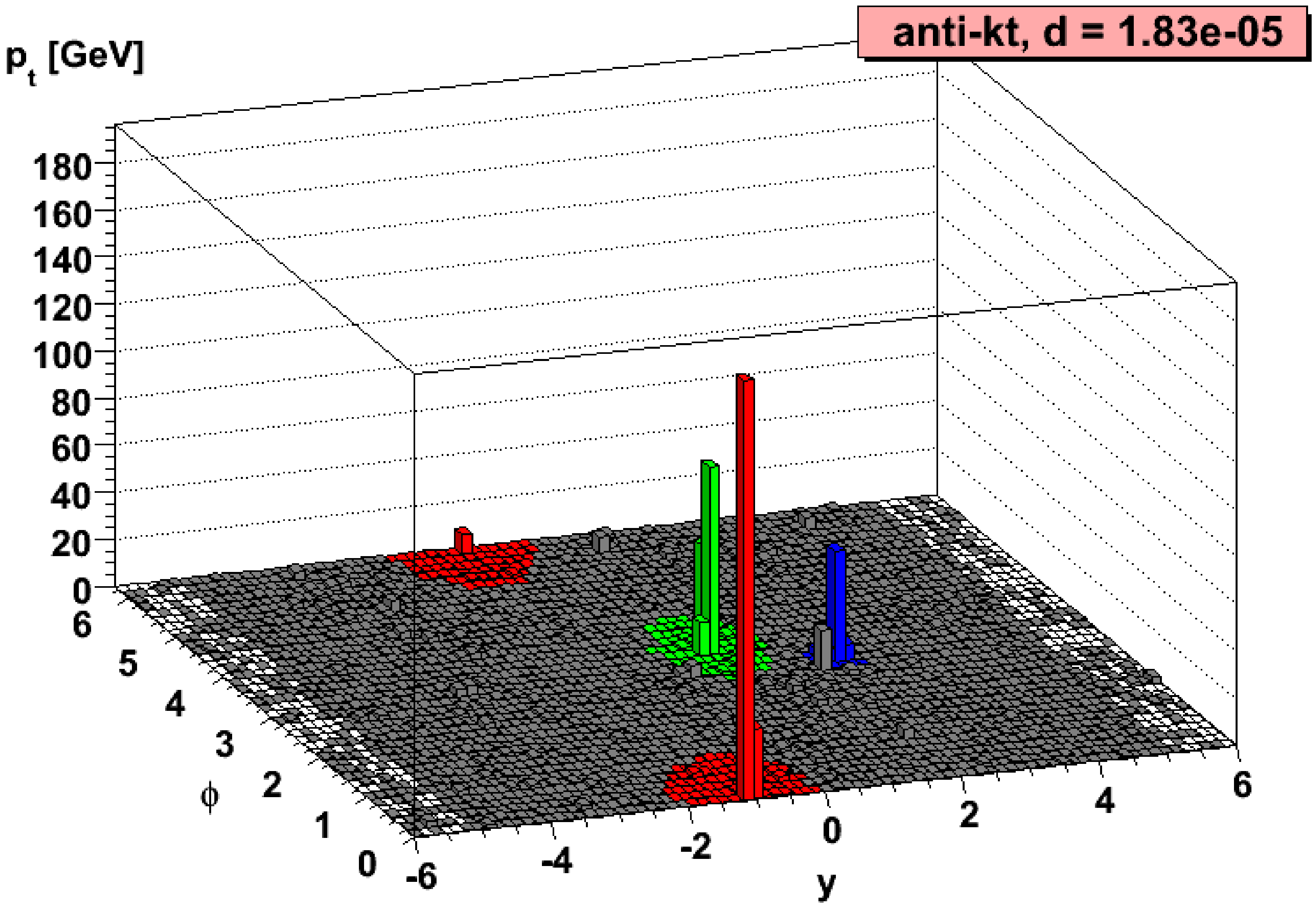
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \qquad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

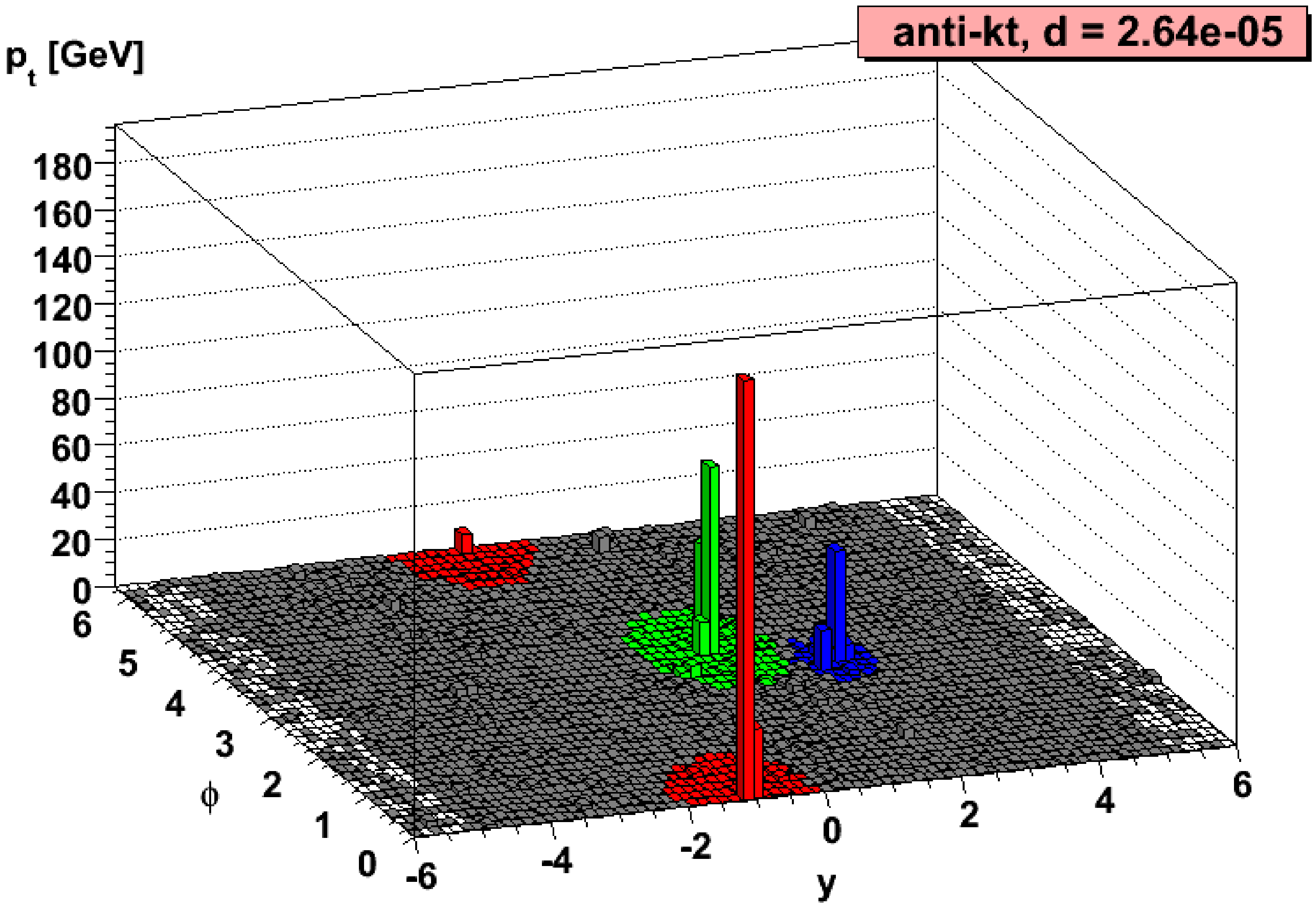
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

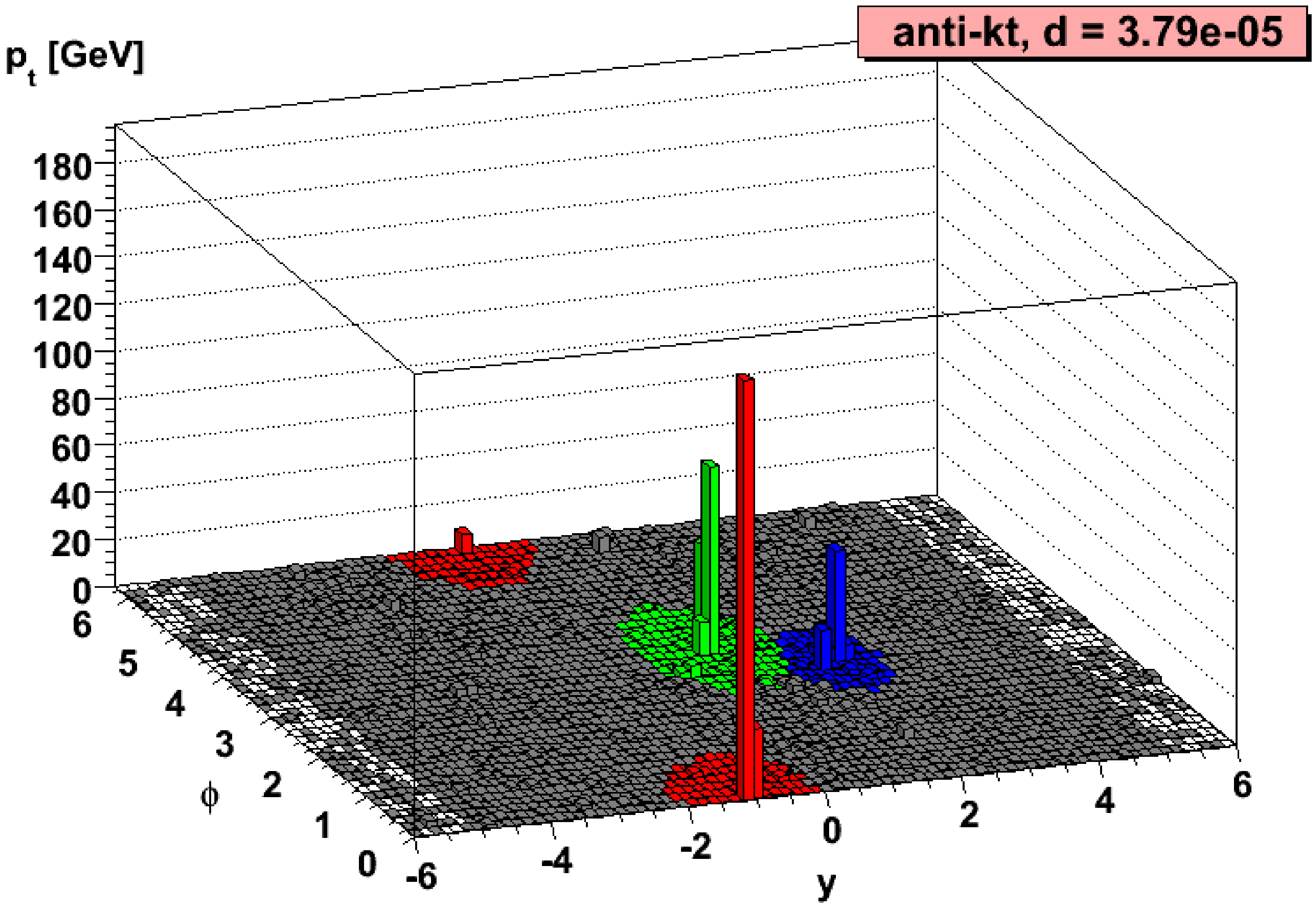
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

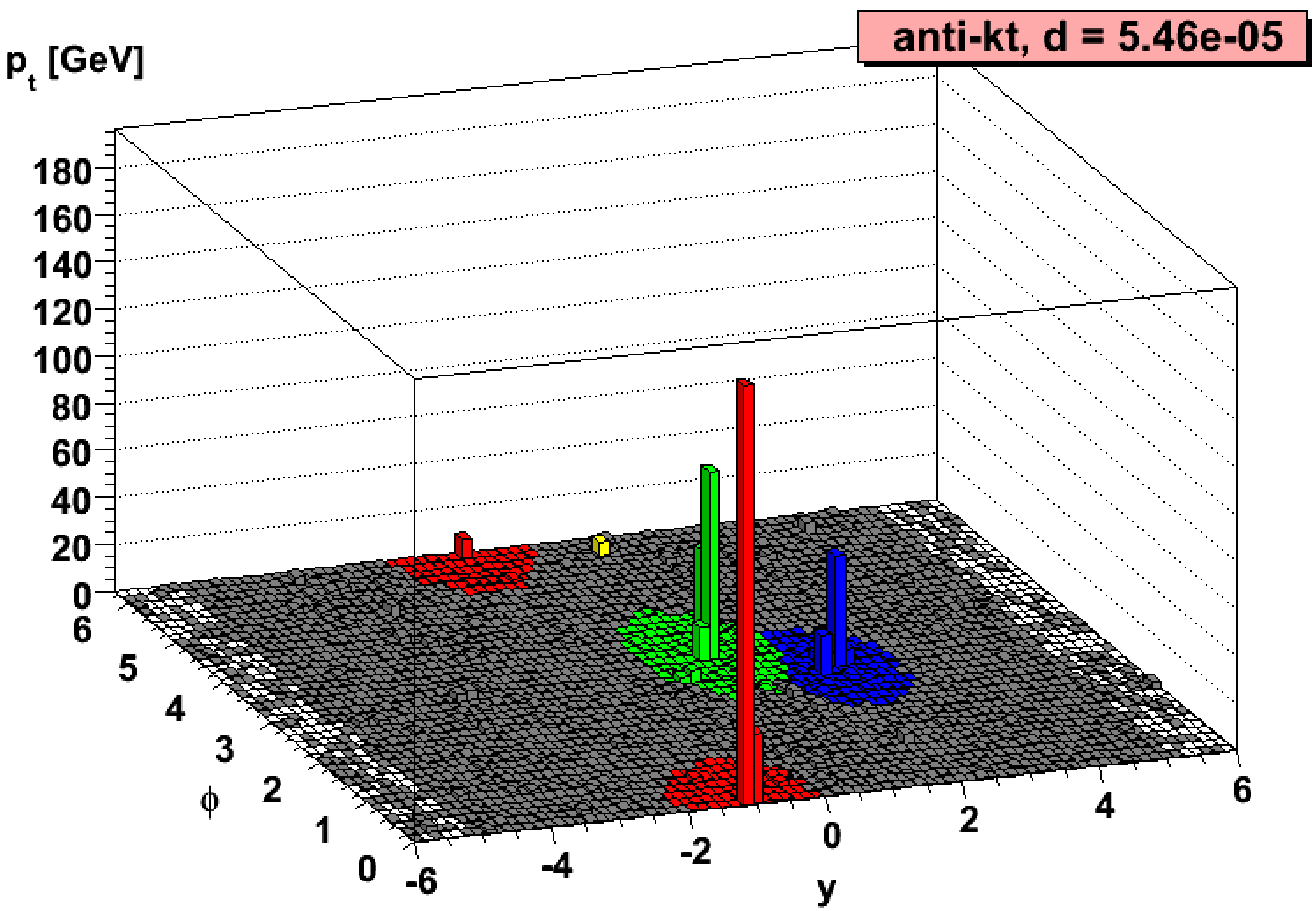
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k$_t$ in action

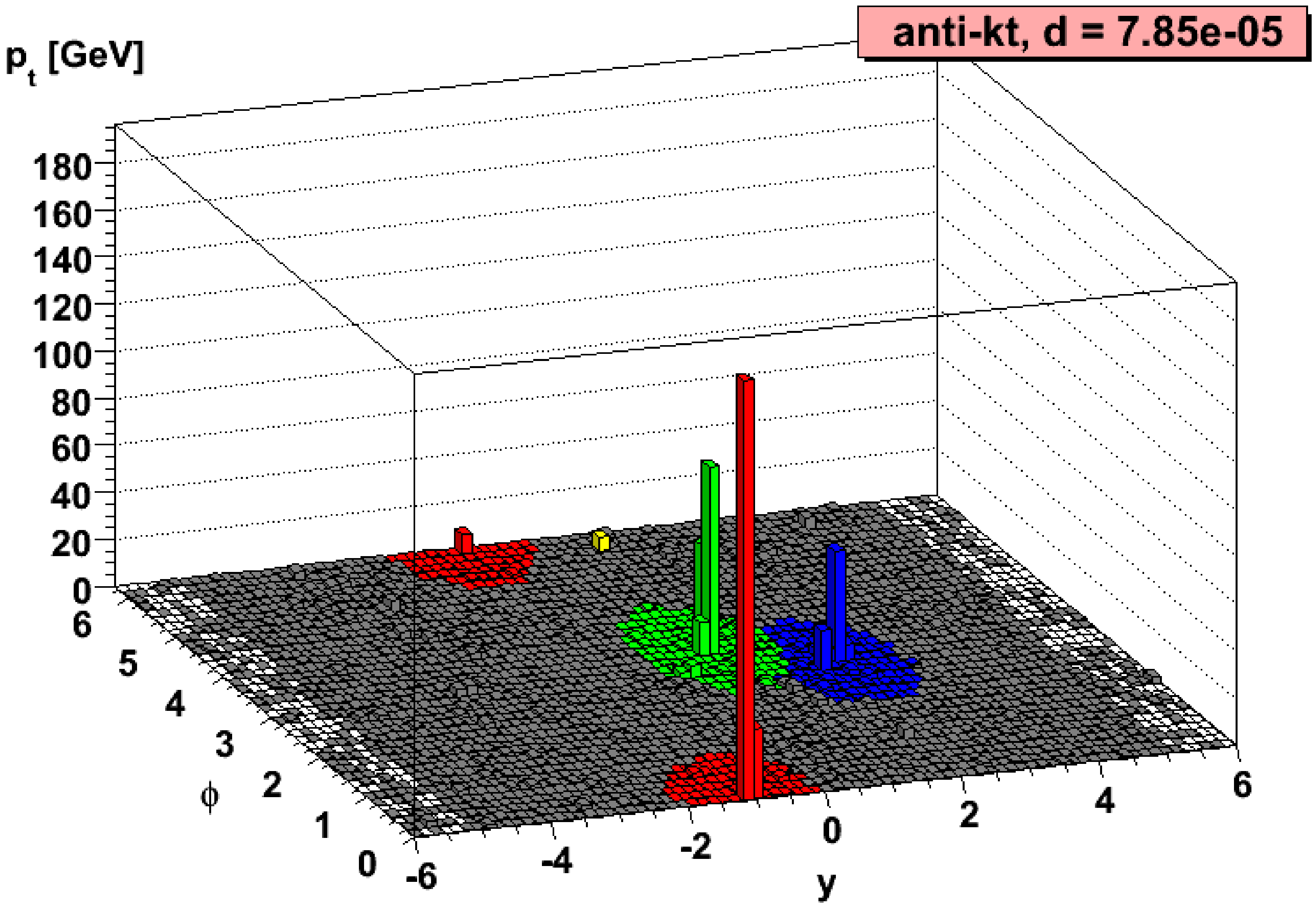
Clustering grows around hard cores

$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Clustering grows around hard cores

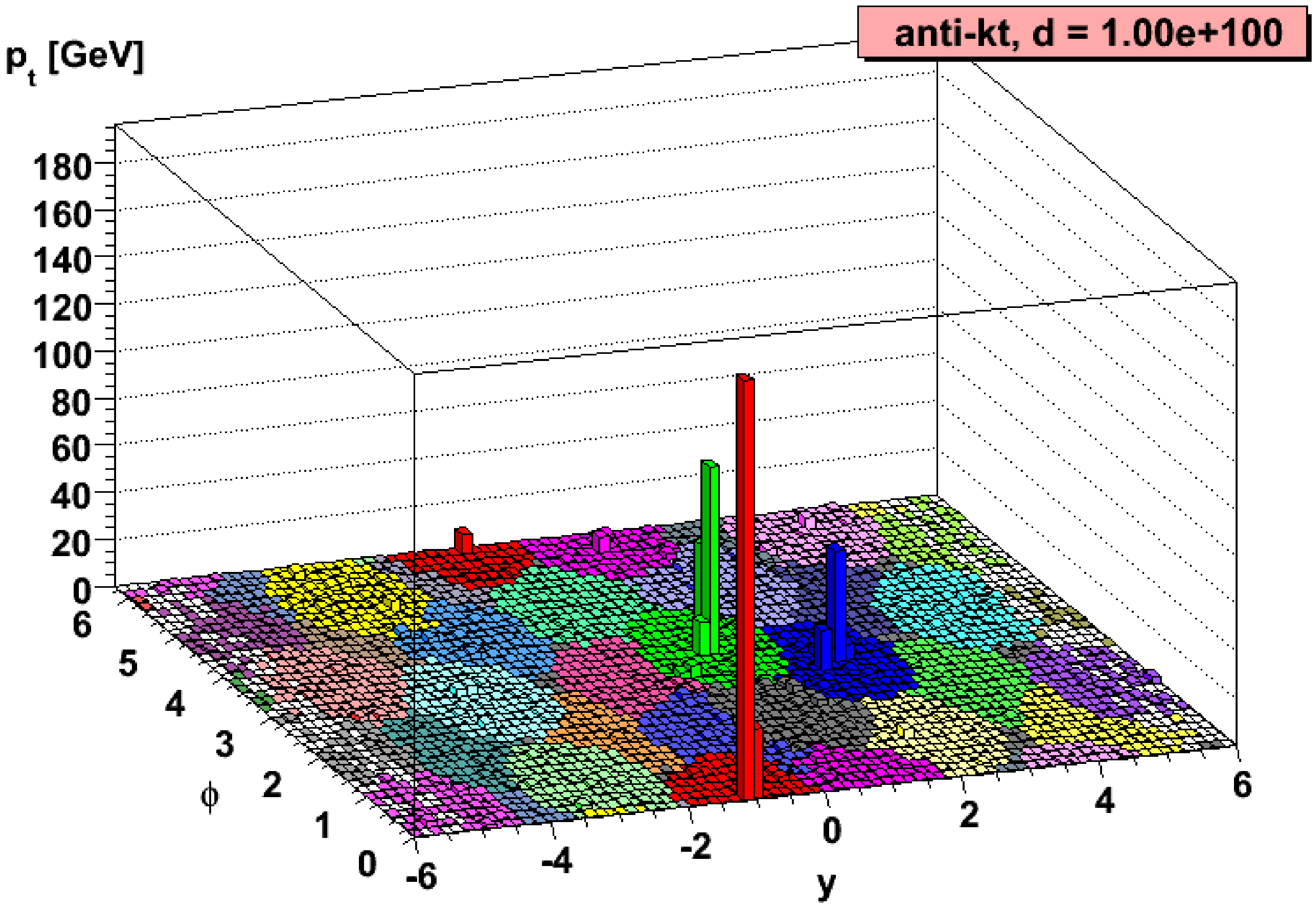
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t in action

Clustering grows around hard cores

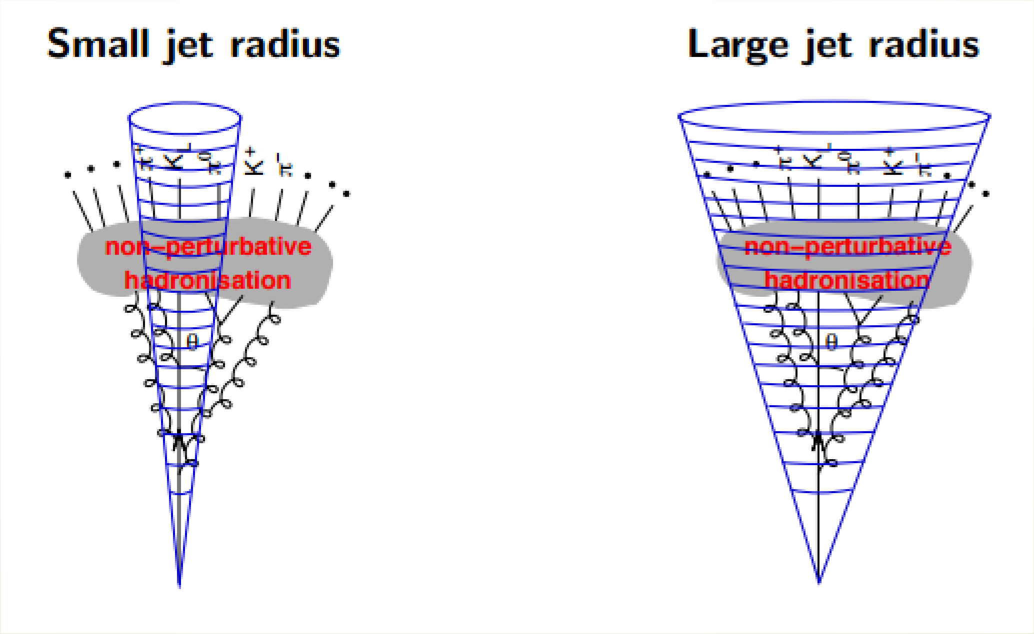
$$d_{ij} = \frac{1}{\max(p_{ti}^2, p_{tj}^2)} \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = \frac{1}{p_{ti}^2}$$



Anti-k_t gives circular jets ("cone-like") in a way that's infrared safe

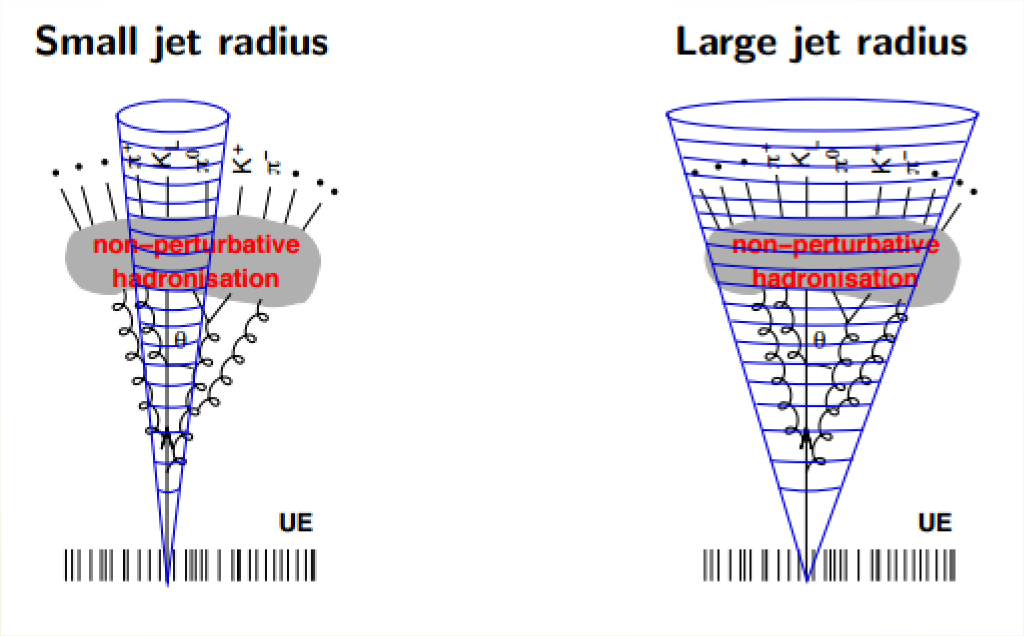
Sequential Clustering Algorithms

- How do we choose jet radius R?
- Answer: it depends on what effects are most important to us!
- Suppose the dominant effect we want to study is non-perturbative fragmentation—then large jet radius is good because it captures more



Sequential Clustering Algorithms

- How do we choose jet radius R?
- Usually, two requirements that we desire in a jet-finding algorithm (not listed earlier) are minimal sensitivity to pileup and underlying event
- In that case, small jet radius is good because it captures less
- Analyses have shown anti-k_T algorithm is insensitive to pileup and UE

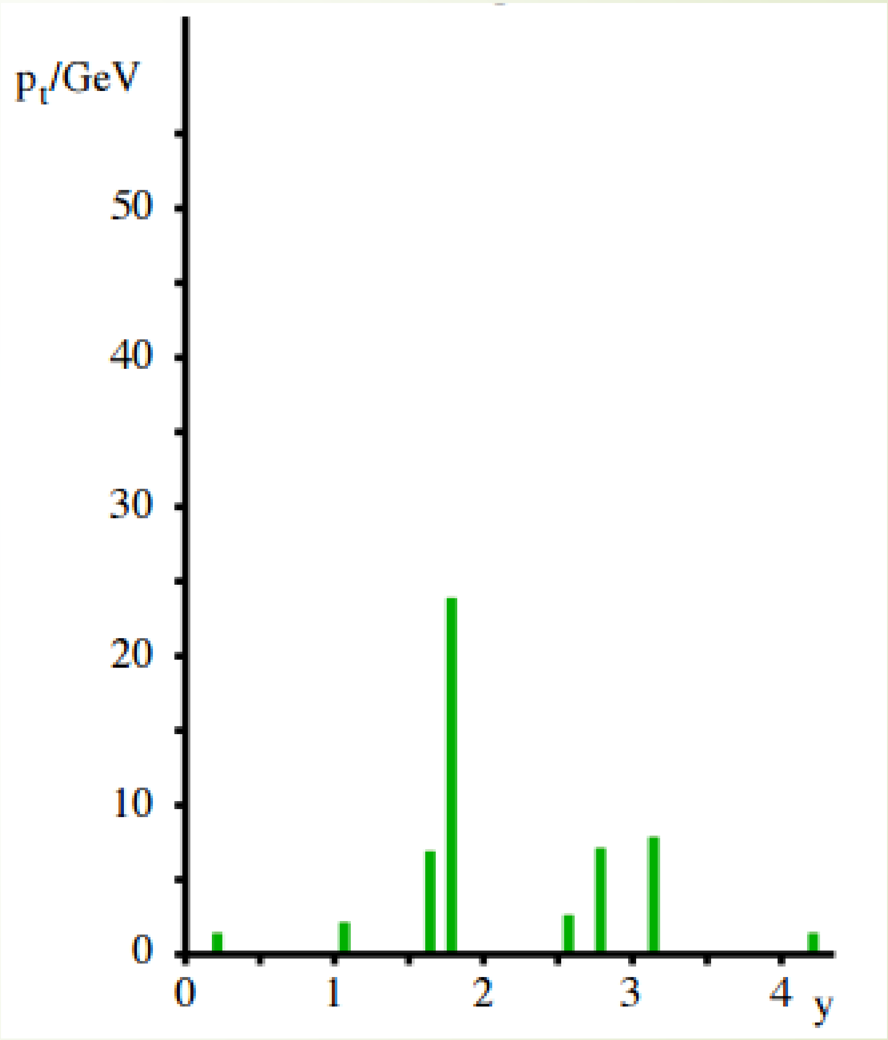


Jet Substructure

- Jet-finding algorithms give us information about energies and momenta of jets as a whole
- Can they also help us find information about partons *within* a jet?
  - Yes, but some algorithms are better than other
- All algorithms give the same final jet, but can they be undone to give information about partons?
- Consider sequential clustering algorithms because cone algorithms do not build up jets piece-by-piece

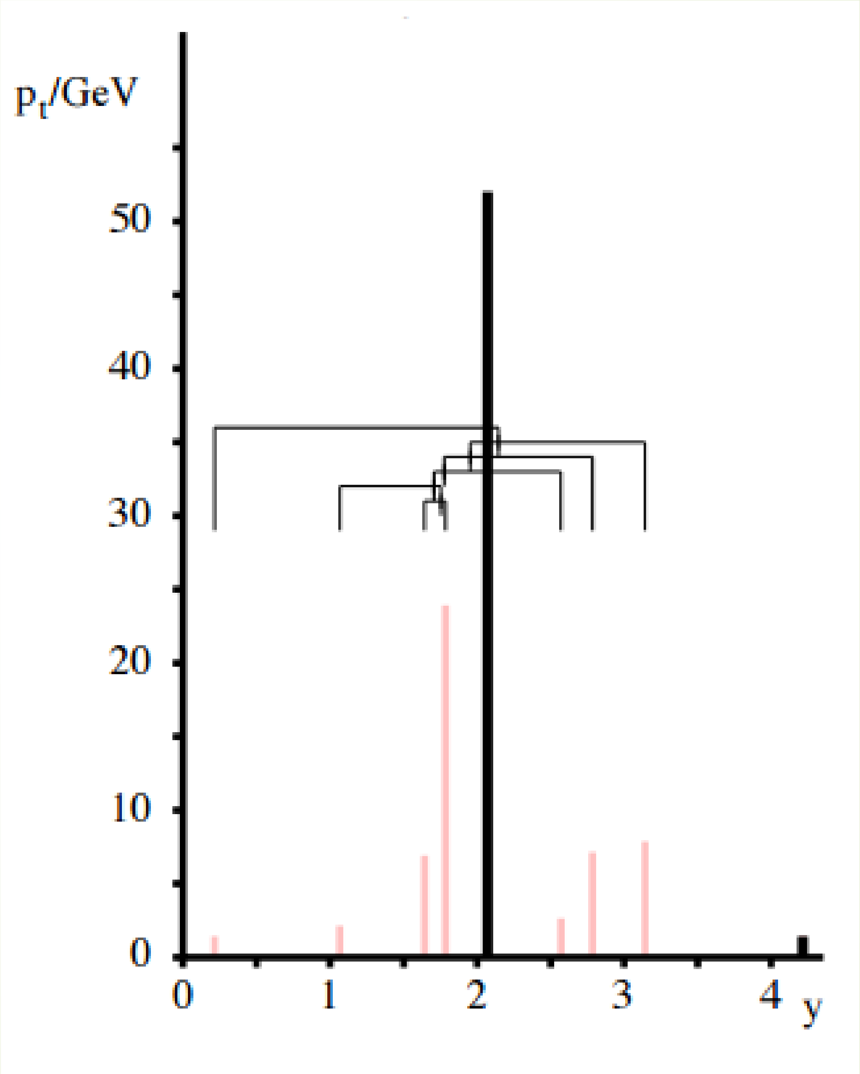
Jet Substructure

- Consider the following data at right:
- We would probably like to consider this as two partons at y = 1.75 and 3, and some soft junk



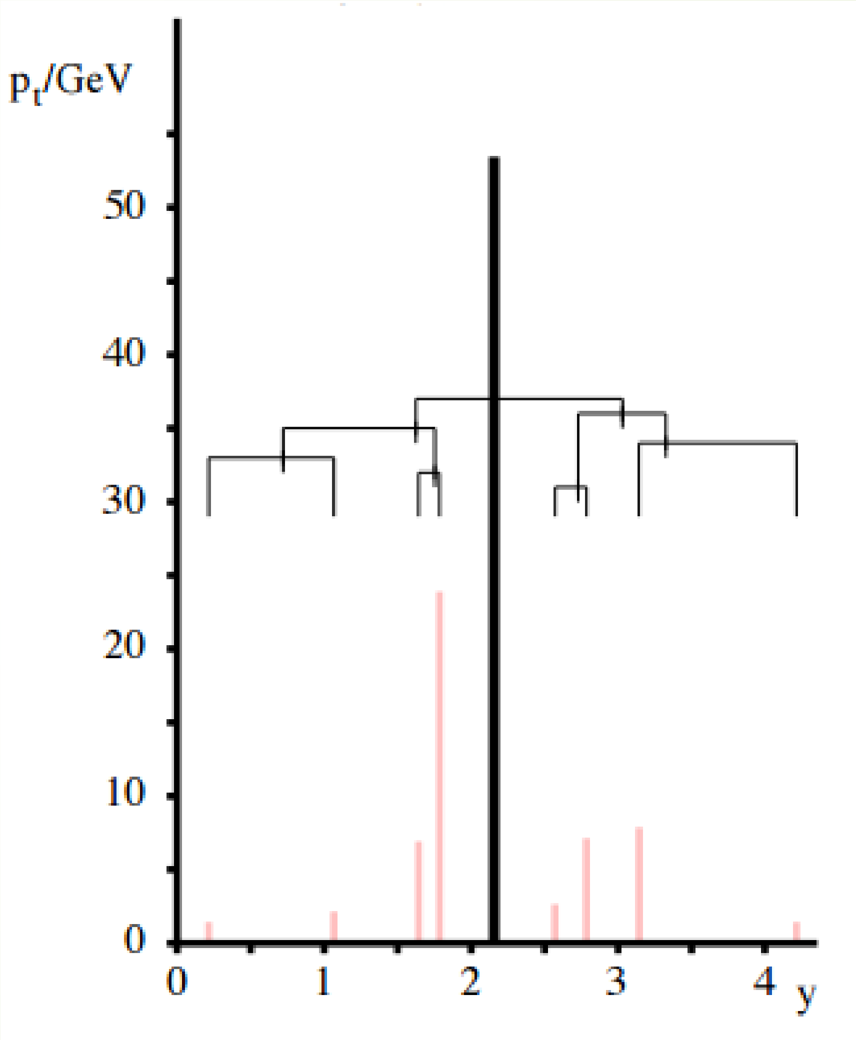
Jet Substructure

- Consider the following data at right:
- We would probably like to consider this as two partons at y = 1.75 and 3, and some soft junk
- Use anti-k_T algorithm
- Algorithm works through both "blobs" without considering them as separate partons
- Not very useful



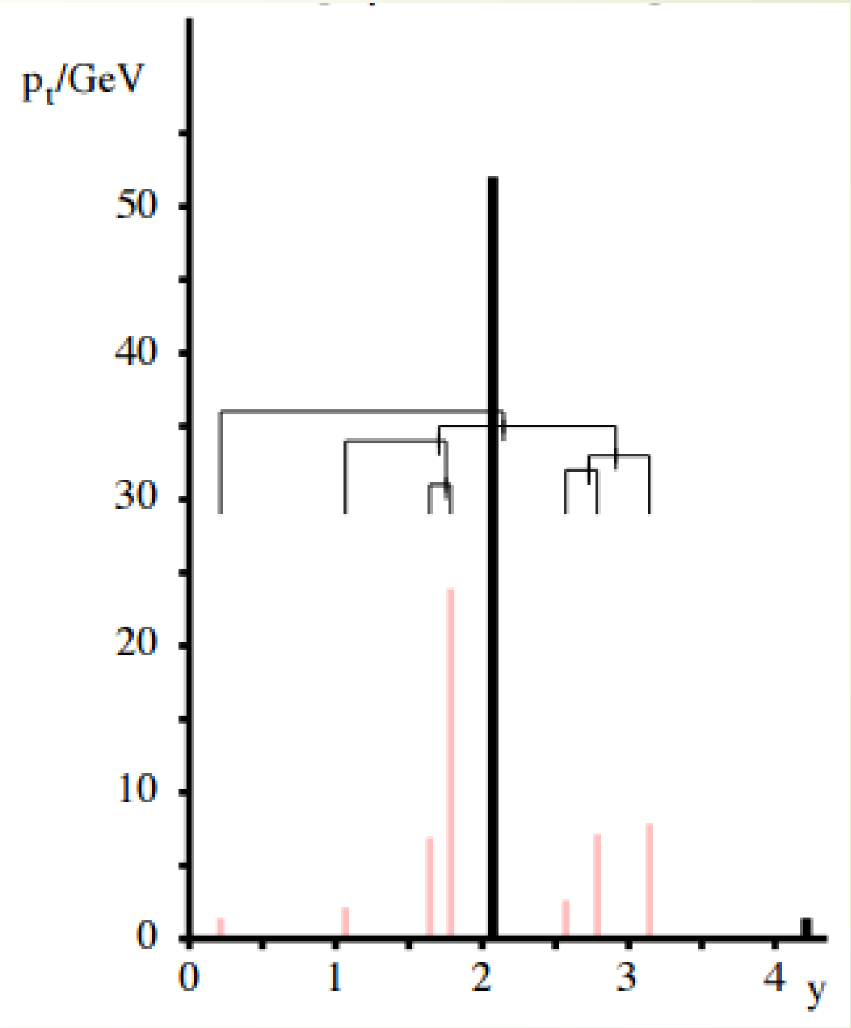
Jet Substructure

- Consider the following data at right:
- We would probably like to consider this as two partons at y = 1.75 and 3, and some soft junk
- Use k_T algorithm
- Clusters soft particles early in the algorithm
- Last step is to merge two hard pieces
- This is good! We have two partons. But could the soft "junk" be removed?

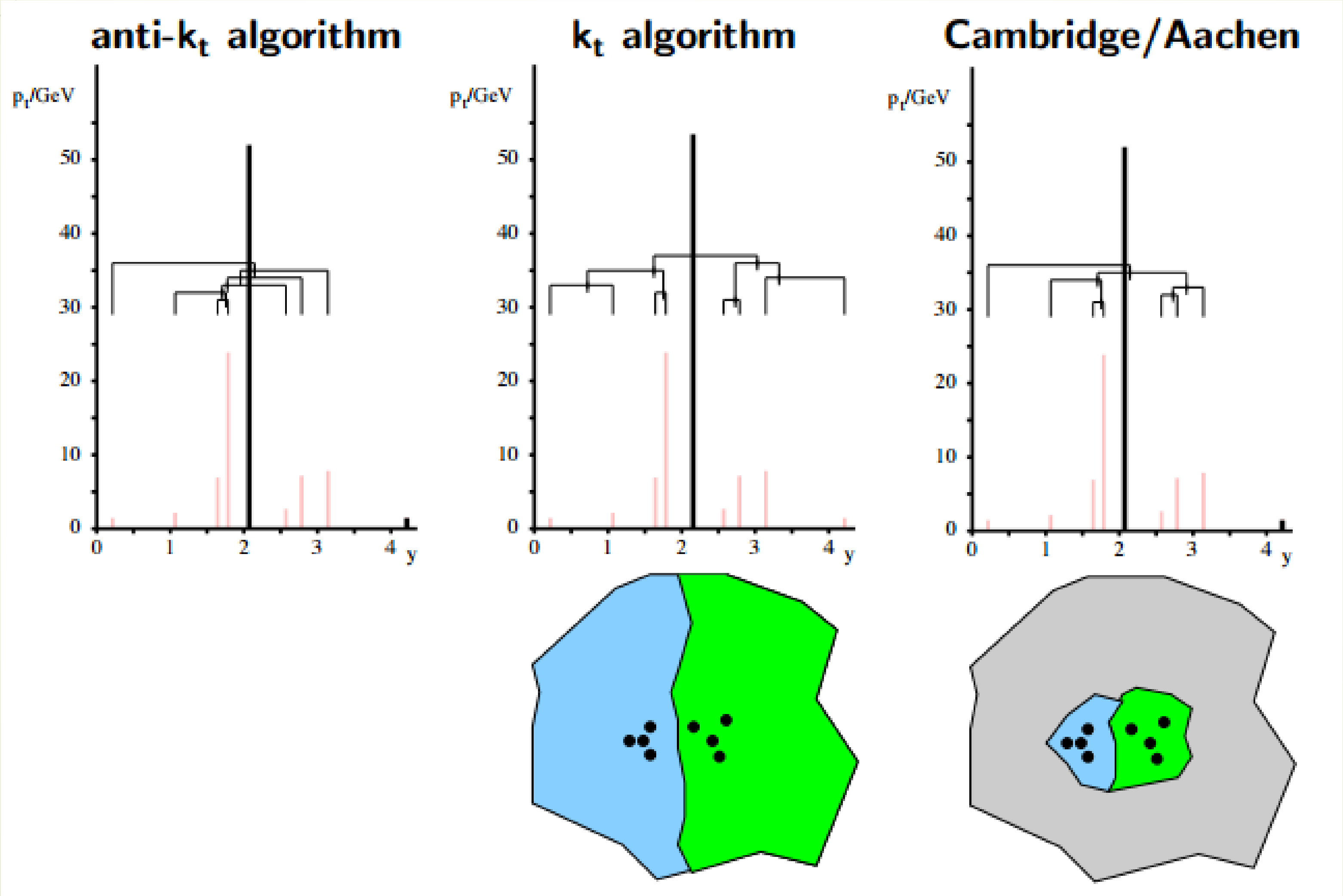


Jet Substructure

- Consider the following data at right:
- We would probably like to consider this as two partons at y = 1.75 and 3, and some soft junk
- Use Cambridge/Aachen algorithm
- Identifies hard "blobs" without soft particles, then throws in soft particles at the end
- Running backwards, we reject soft particles, then separate data into partons



Jet Substructure



Summary

- Physical jets do not always correspond to jets in a computational analysis; need to create jet definitions
- Jet-finding algorithms actually define jets and sort data according to these rules
- Cone algorithms: nice for experimental purposes (circular jets), but until recently, not IR safe
- Sequential clustering algorithms: IR safe, fairly simple to implement, and are now computationally fast
- Anti-k_T algorithm seems to be the best choice for many hadron collider purposes, but to study jet substructure, Cambridge/Aachen offers the best results

References

1. G. Salam. “Basics of QCD: Jets and Jet Substructure”. ICTP-SAIFR school on QCD/LHC Physics. July 2015, Sao Paulo, Brazil. https://gsalam.web.cern.ch/gsalam/repository/talks/2015-SaoPaulo-lecture4.pdf
2. G. Salam. “QCD, Lecture 4”. European School of High-Energy Physics. June 2009, Bautzen, Germany. https://gsalam.web.cern.ch/gsalam/repository/talks/2009-Bautzen-lecture4.pdf
3. P. Schieferdecker. “Jet Algorithms”. April 2009. https://twiki.cern.ch/twiki/bin/viewfile/Sandbox/Lecture?rev=1;filename=Philipp_Schieferdeckers_Lecture.pdf
4. A. Banfi. “Jet Algorithms”. October 2011. https://people.phys.ethz.ch/~banfi/Lectures/jets/jet_algorithms_1.pdf
5. G. Salam. “QCD at Hadron Colliders, Lecture 2”. Maria Laach Herbtschule für Hochenenergiephysik. September 2010. https://gsalam.web.cern.ch/gsalam/repository/talks/2010-MariaLaach-lecture2.pdf

Note: Many slides taken from sources [1] and [2]

Further Reading

List of talks (lectures, summer schools, etc.) by Gavin Salam:
https://gsalam.web.cern.ch/gsalam/teaching/PhD-courses.html

Snowmass Accord: http://inspirehep.net/record/303065/files/fermilab-conf-90-249.pdf

Original proposal of anti-k_T algorithm: http://arxiv.org/pdf/0802.1189v2.pdf

Resources on SISCone Algorithm: http://arxiv.org/pdf/0704.0292v2.pdf
https://siscone.hepforge.org/algorithm.html

New J_{E_T} algorithm: http://arxiv.org/pdf/1411.3705v1.pdf