Introduction to Jet Finding and Jetography (1)

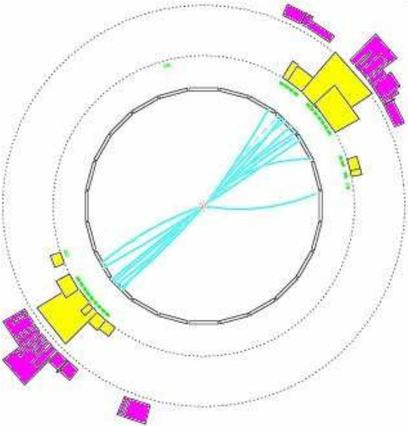
Gavin Salam

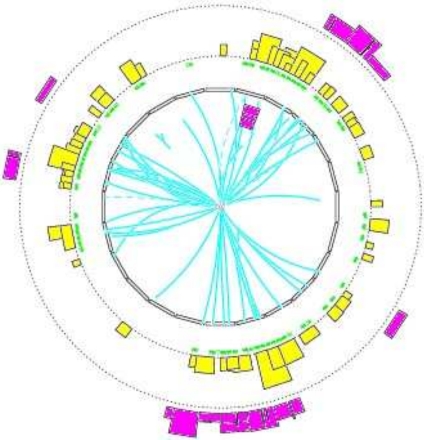
CERN, Princeton University and LPTHE/Paris (CNRS)

2011 IPMU-YITP School and Workshop
on Monte Carlo Tools for LHC

Yukawa Institute for Theoretical Physics, Kyoto University, Japan
September 2011

Jets

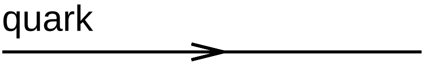




Jets are everywhere in QCD
Our *window on partons*

But *not* the same as partons:
Partons ill-defined; jets *well-definable*

Why do we see jets? Parton fragmentation



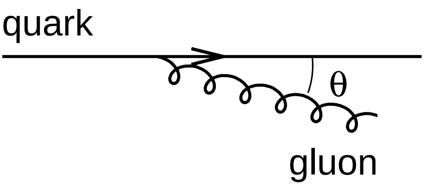
Gluon emission:

$$\int \alpha_s \frac{dE}{E}\frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



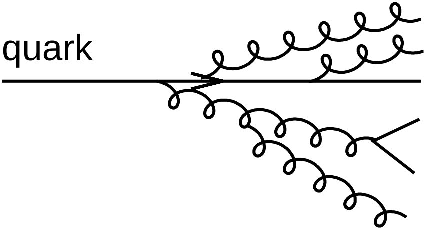
Gluon emission:

$$\int \alpha_s \frac{dE}{E}\frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



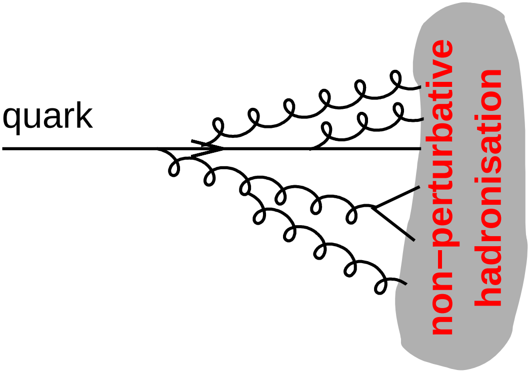
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



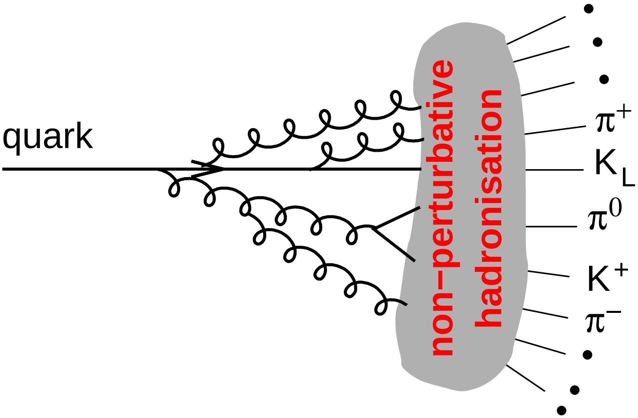
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



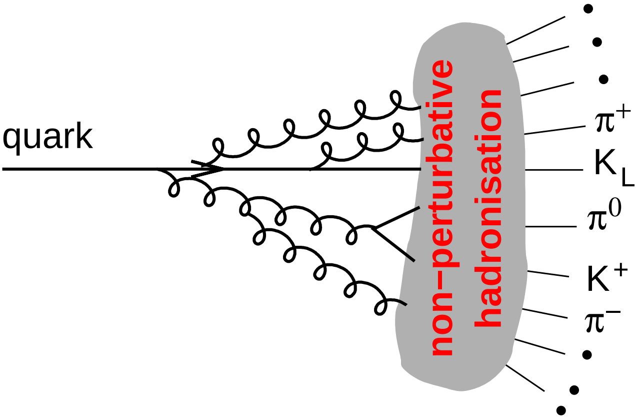
Gluon emission:

$$\int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1$$

At low scales:

$$\alpha_s \to 1$$

Why do we see jets? Parton fragmentation



Gluon emission:

$$\int \alpha_s \frac{dE}{E}\frac{d\theta}{\theta} \gg 1$$

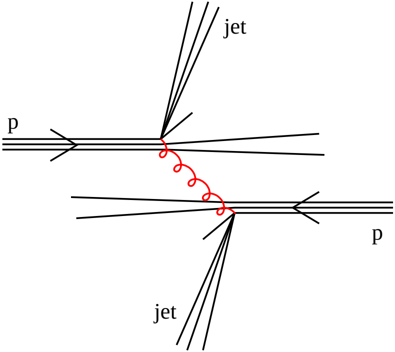
At low scales:

$$\alpha_s \to 1$$

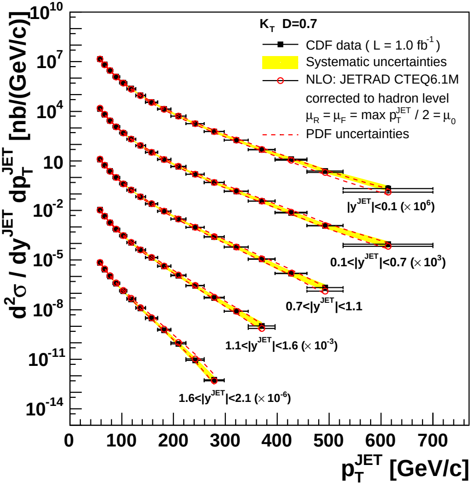
High-energy partons unavoidably lead to collimated bunches of hadrons

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering



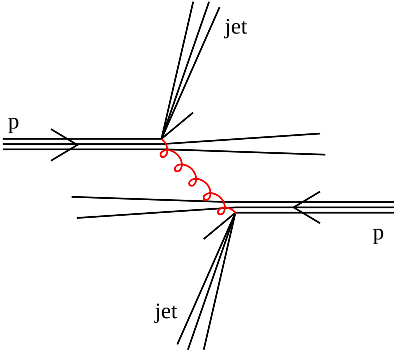
Tevatron results



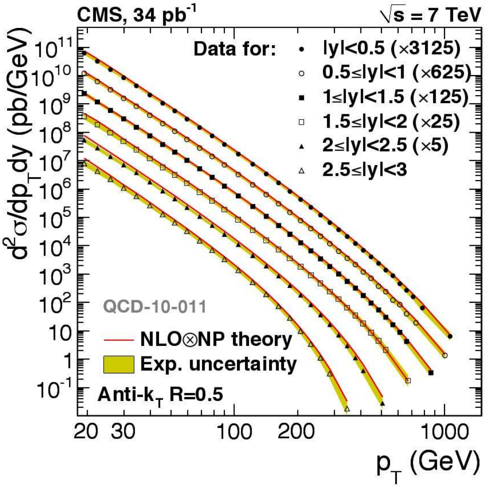
Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering



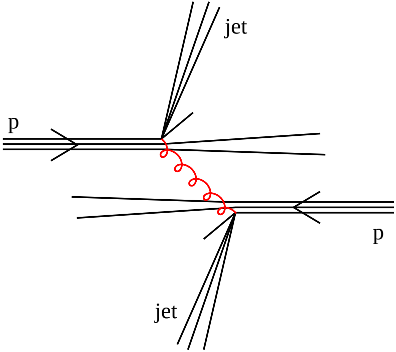
CMS results



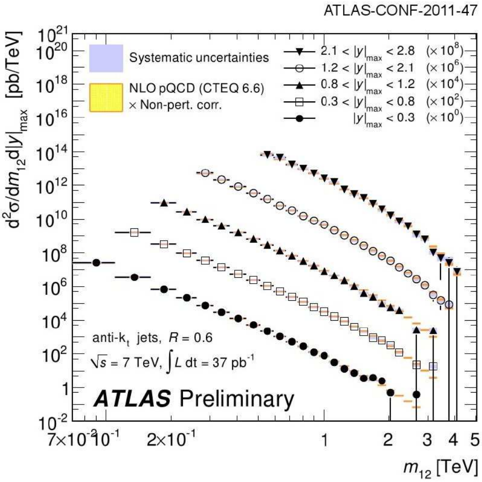
Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from scattering of partons

Jets are unavoidable at hadron colliders, e.g. from parton scattering

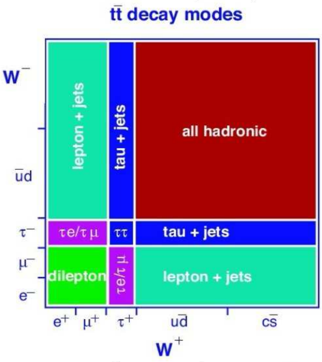


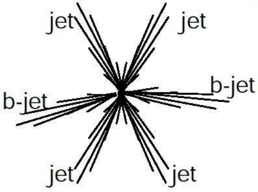
ATLAS results



Jet cross section: data and theory agree over many orders of magnitude ⇔ probe of underlying interaction

Jets from heavy decays





All-hadronic
(BR~46%, huge bckg)

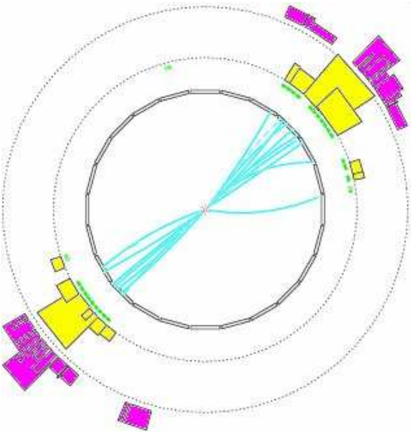
Heavy objects: multi-jet final-states

- 10^7 $t\bar{t}$ pairs for 1 fb^{-1} @ 14 TeV
- Vast # of QCD multijet events

# jets	# events for 1 fb^{-1}
3	$2 \cdot 10^{10}$
4	$5 \cdot 10^{9}$
5	$1 \cdot 10^{9}$
6	$3 \cdot 10^{8}$
7	$1 \cdot 10^{8}$
8	$4 \cdot 10^{7}$

Tree level
$p_t(\text{jet}) > 20$ GeV, $\Delta R_{ij} > 0.4$, $|y_{ij}| < 2.5$
Gleisman & Höche '08

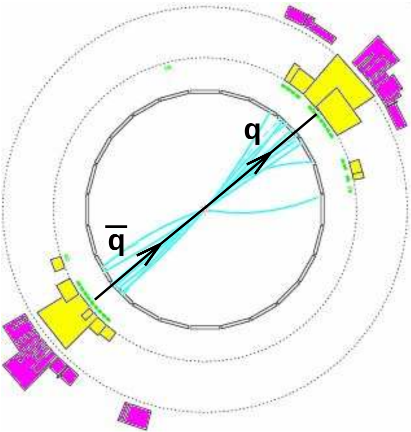
Seeing v. defining jets



Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

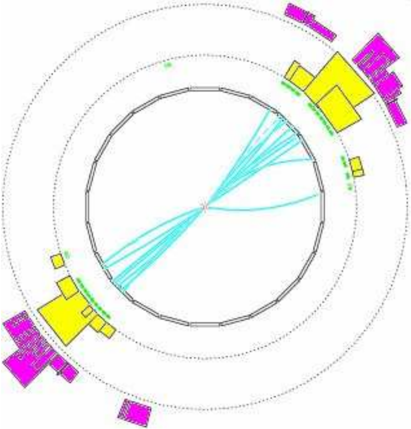
Seeing v. defining jets



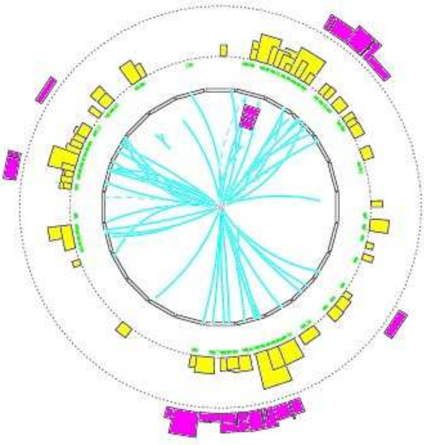
Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets



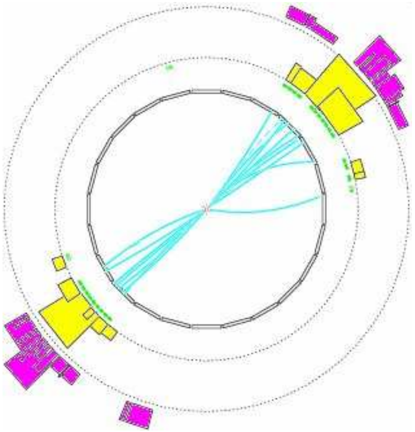
Jets are what we see.
Clearly(?) 2 jets here



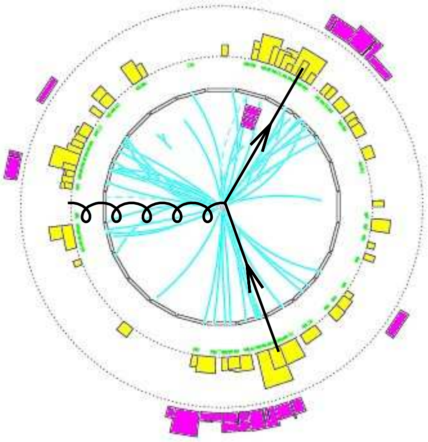
How many jets do you see?

Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets

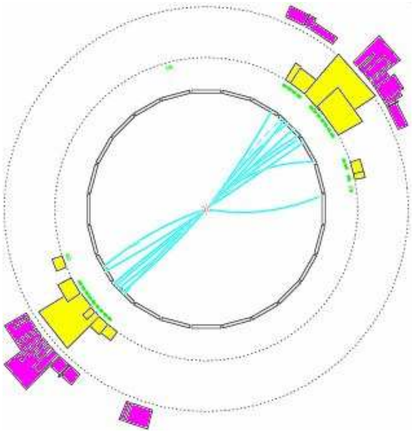


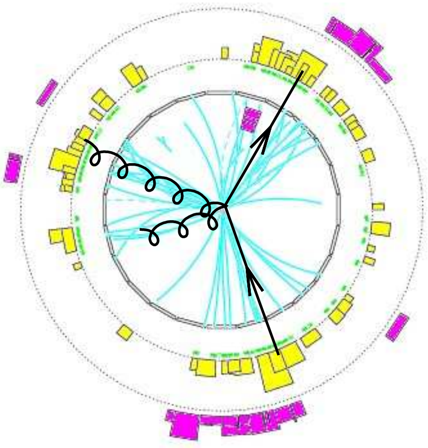
Jets are what we see.
Clearly(?) 2 jets here



How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets

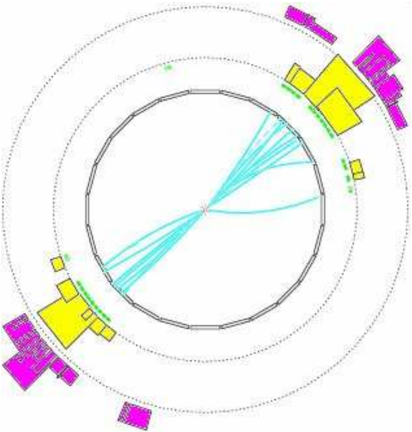




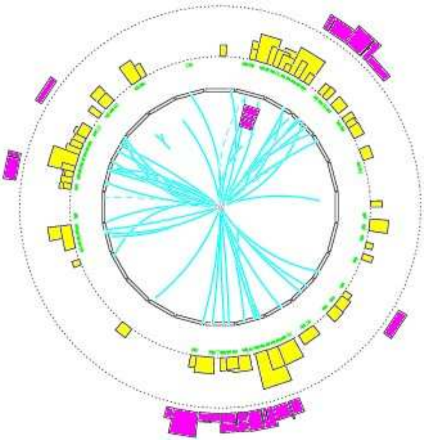
Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

Seeing v. defining jets



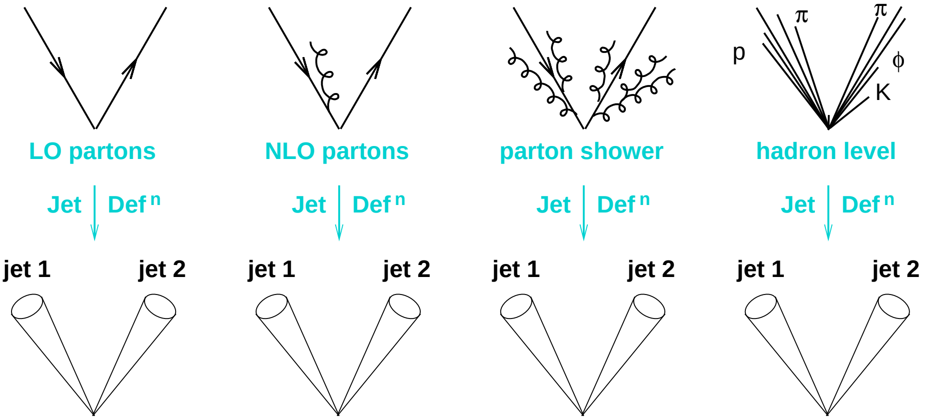
Jets are what we see.
Clearly(?) 2 jets here



How many jets do you see?
Do you really want to ask yourself this question for 10^9 events?

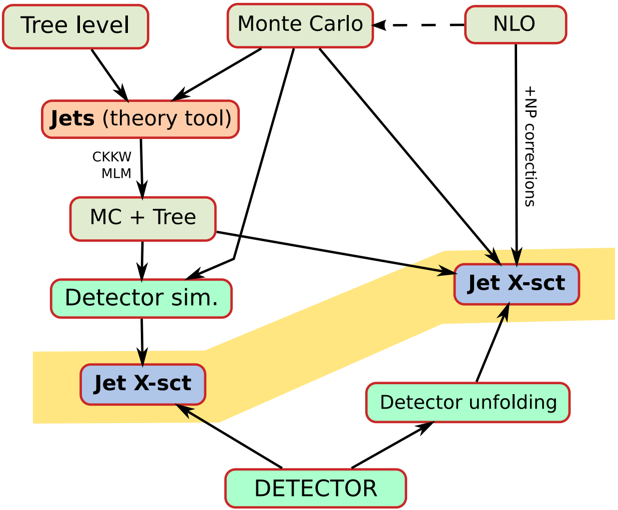
Seeing v. defining jets

A jet definition is a fully specified set of rules for *projecting* information from an event's partons or hadrons onto a handful of *parton-like* objects (jets):



Projection to jets should be resilient to QCD effects

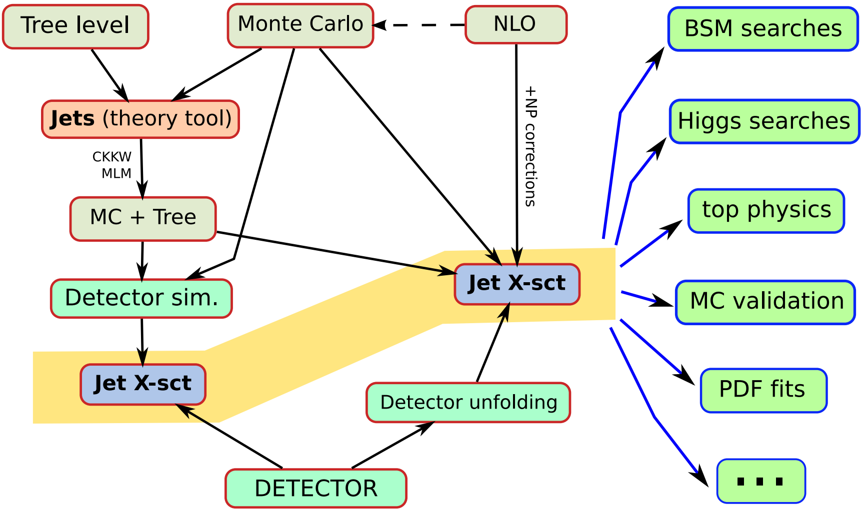
QCD jets flowchart



Jet (definitions) provide central link between expt., "theory" and theory

And jets are an input to almost all analyses

QCD jets flowchart



Jet (definitions) provide central link between expt., “theory” and theory

And jets are an input to almost all analyses

These lectures

Aims: to provide you with

- the "basics" needed to understand what goes into current jet-based measurements;
- some insight into the issues that are relevant when thinking about a jet measurement

Structure:

- General considerations
- Common jet definitions at LHC — Today
- Physics with jets — Tomorrow

Defining jets

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? — Jet algorithm + parameters
2. how do you combine their momenta? — Recombination scheme

   Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? Jet algorithm + parameters
2. how do you combine their momenta? Recombination scheme

   Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations $\to$ divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Not all ambiguity is allowed

Jets should be **invariant** with respect to certain modifications of the event:

- collinear splitting
- infrared emission

Why?

- Because otherwise lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ divergent results
- Hadron-level 'jets' would become fundamentally non-perturbative
- Detectors can resolve neither full collinear nor full infrared event structure

Known as **infrared and collinear safety**

Two main classes of jet alg.

Sequential recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Cone

- top-down
- centred around idea of an 'invariant', directed energy flow

Cones: most widely used at Tevatron
Seq. rec.: most widely used at LHC and HERA

In this lecture we'll concentrate on the sequential recombination algorithms

Two main classes of jet alg.

Sequential recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Cone

- top-down
- centred around idea of an 'invariant', directed energy flow

Cones: most widely used at Tevatron
Seq. rec.: most widely used at LHC and HERA

In this lecture we'll concentrate on the sequential recombination algorithms

Sequential recombination jet algorithms

starting with a classic e^+e^- algorithm

Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

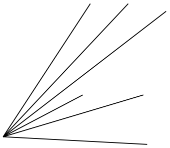
Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

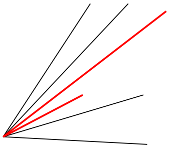


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

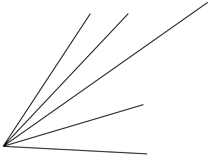


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

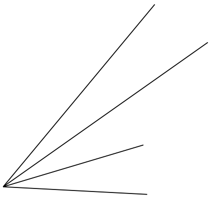


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

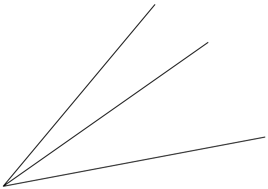


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.

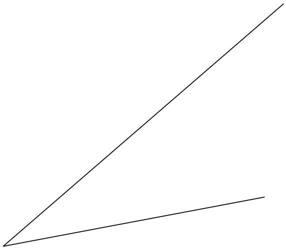


Motivating sequential recombination algorithms

It's a good approximation to think of the development of a jet as a consequence of the repeated $1 \to 2$ branching of quarks and gluons.

E.g. this is how Pythia and Herwig have long modelled events

Sequential recombination algorithms try to work their way backwards through this branching, repeatedly combining pairs of particles into a single one.



The main questions are:

- How do you choose which pair of particles to combine at any given stage?
- When do you stop combining them?

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}, \qquad (E_j \ll E_i\,,\ \theta_{ij} \ll 1)\,.$$

To invert branching process, take pair with strongest divergence between them — they're the most *likely* to belong together.

This is basis of **k_t/Durham algorithm** (e^+e^-):

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

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

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\ \ \theta_{ij} \ll 1)\,.$$

To invert branching process, take pair with strongest divergence between them — they're the most *likely* to belong together.

This is basis of **$\mathbf{k_t}$/Durham algorithm** (e^+e^-):

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

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

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

[dk ... 1) .

To inv ... een
them

The algorithm has one parameter

- y_{cut}: sets minimal relative transverse momentum between any pair of jets

This is basis of **$\mathbf{k_t}$/Durham algorithm** (e^+e^-):

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

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

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

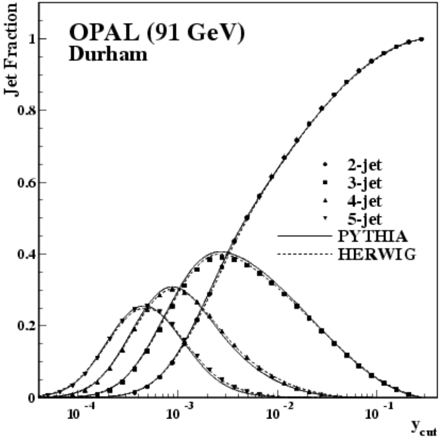
Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm features

- Gives hierarchy to event and jets

  Event can be charaterised by y_{23}, y_{34}, y_{45}.

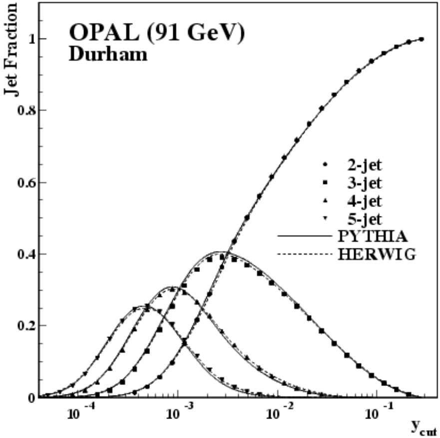
- Resolution parameter related to minimal transverse momentum between jets



Most widely-used jet algorithm in e^+e^-

- Collinear safe: collinear particles recombined early on
- Infrared safe: soft particles have no impact on rest of clustering seq.

k_t/Durham algorithm features



- Gives hierarchy to event and jets

  Event can be charaterised by y_{23}, y_{34}, y_{45}.

- Resolution parameter related to minimal transverse momentum between jets

Most widely-used jet algorithm in e^+e^-

- Collinear safe: collinear particles recombined early on
- Infrared safe: soft particles have no impact on rest of clustering seq.

k_t alg. at hadron colliders

1st attempt

- Lose absolute normalisation scale Q. So use unnormalised d_{ij} rather than y_{ij}:

$$d_{ij} = 2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})$$

- Now also have *beam remnants* (go down beam-pipe, not measured)
  Account for this with particle-beam distance

$$d_{iB} = 2E_i^2(1-\cos\theta_{iB})$$

squared transv. mom. wrt beam

k_t alg. at hadron colliders

2nd attempt: make it longitudinally boost-invariant

Catani, Dokshitzer, Seymour & Webber '93

- Formulate in terms of rapidity (y), azimuth (ϕ), p_t

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2\,, \qquad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

NB: not η_i, E_{ti}

- Beam distance becomes

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

squared transv. mom. wrt beam

Apart from measures, just like e^+e^- alg.
Known as **exclusive k_t algorithm**.

Problem: at hadron collider, no single fixed scale (as in Q in e^+e^-). So how do you choose d_{cut}?

See e.g. Seymour & Tevlin '06

k_t alg. at hadron colliders

3rd attempt: **inclusive k_t algorithm**

- Introduce angular radius R (NB: dimensionless!)

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

- 1. Find smallest of d_{ij}, d_{iB}
  2. if ij, recombine them
  3. if iB, call i a jet and remove from list of particles
  4. repeat from step 1 until no particles left.

S.D. Ellis & Soper, '93; the simplest to use

Jets all separated by at least R on y, ϕ cylinder.

NB: number of jets not IR safe (soft jets near beam); number of jets above p_t cut **is** IR safe.

k_t alg. at hadron colliders

3rd attempt: **inclusive k_t algorithm**

- Introduce angular radius R (NB: dimensionless!)

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

- 1. F
  2. if
  3. if
  4. re

to use

Jets a

Two parameters to remember

- **R**: sets $y-\phi$ reach of the jet; minimal interjet separation
- **$\mathbf{p}_t$ cut** on the jets

These parameters are common to all widely used hadron-collider jet algorithms.

NB: number of jets not IR safe (soft jets near beam); number of jets above p_t cut **is** IR safe.

k_t in action

k_t alg.: Find smallest of

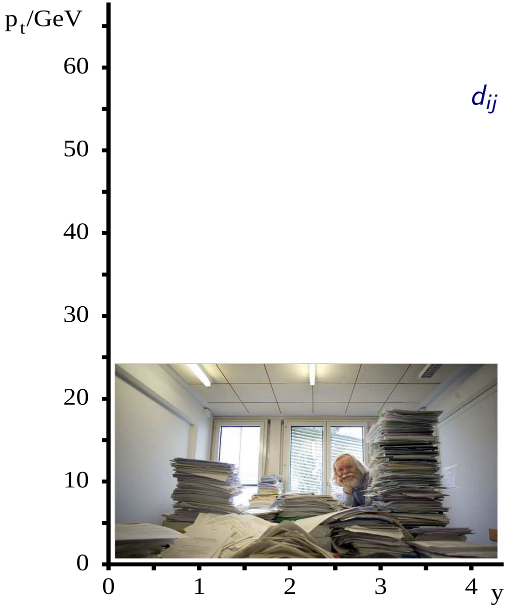
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

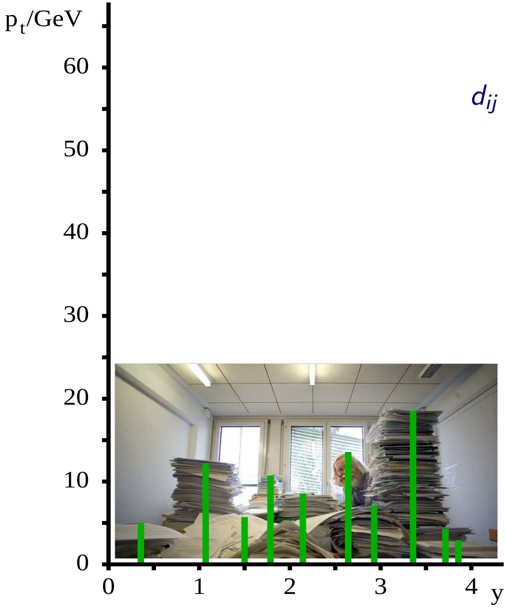
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

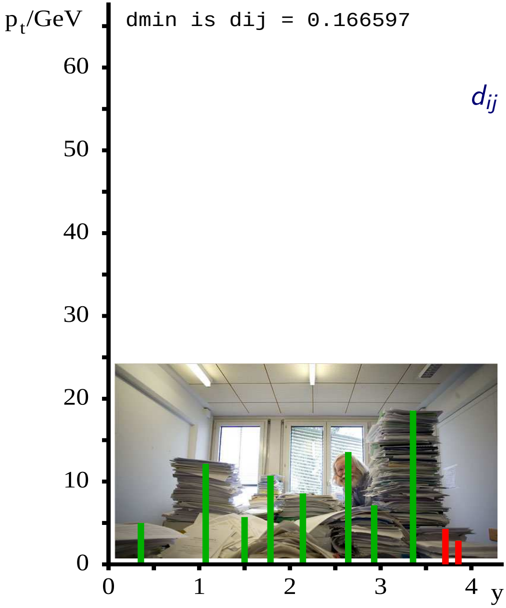
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

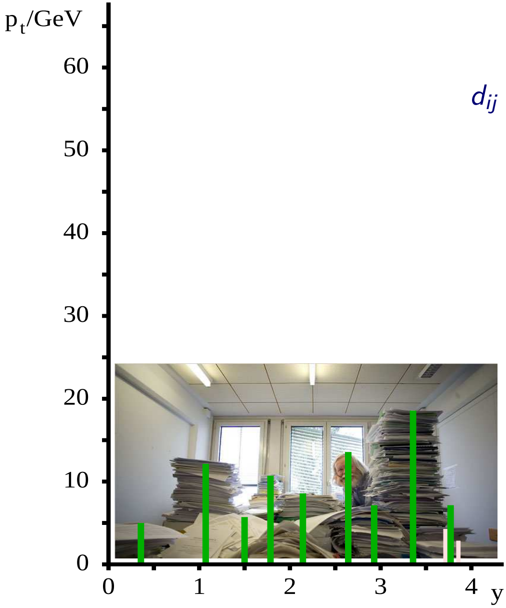
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

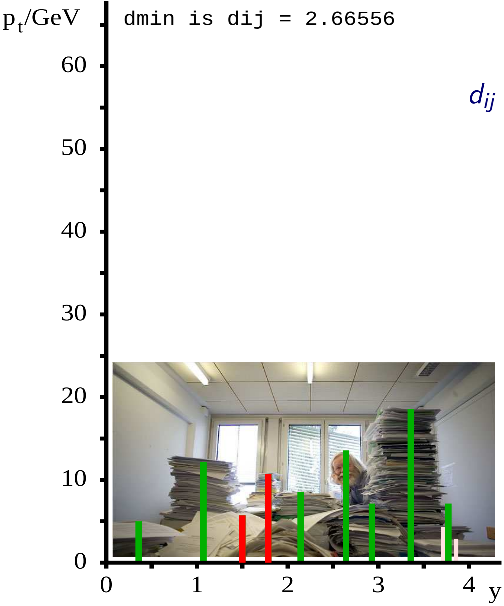
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

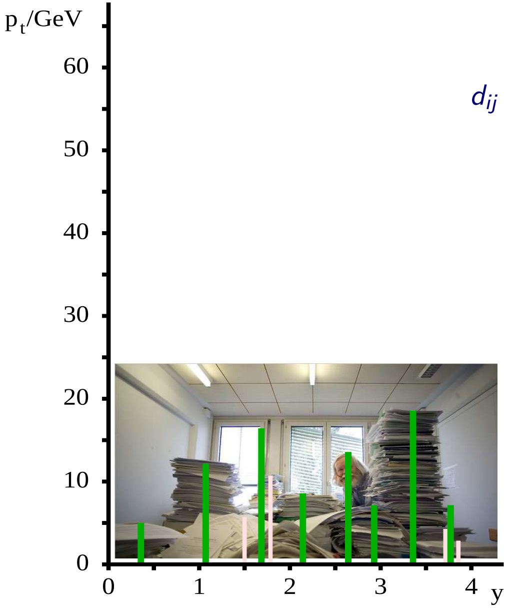
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

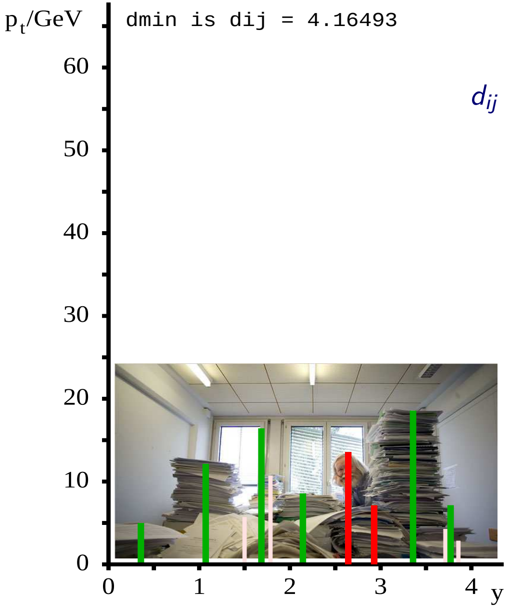
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

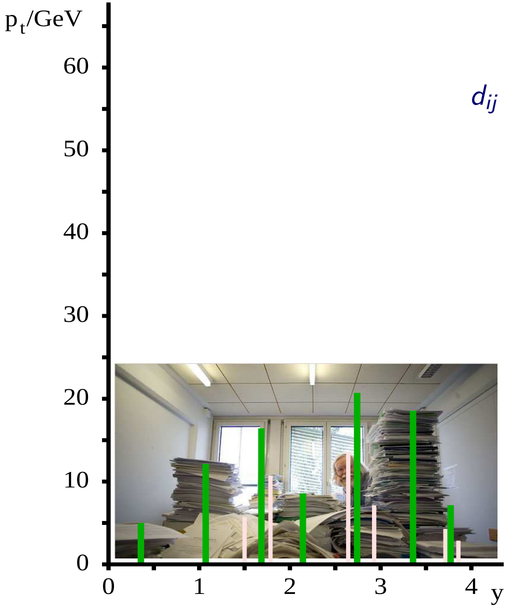
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

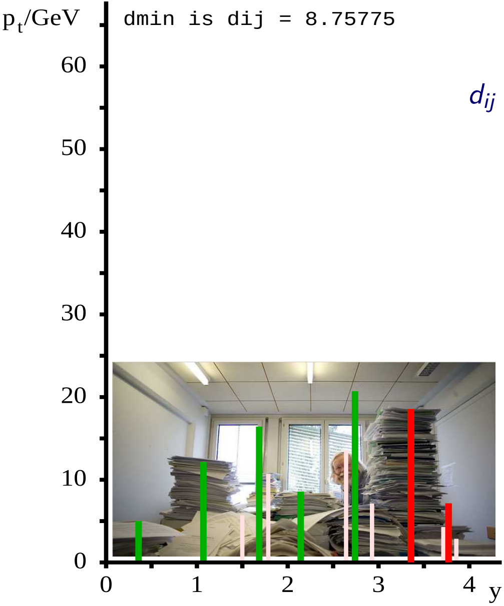
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

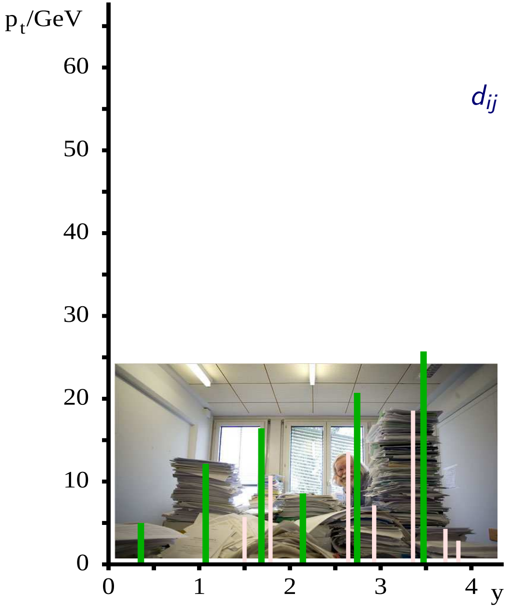
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2) \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

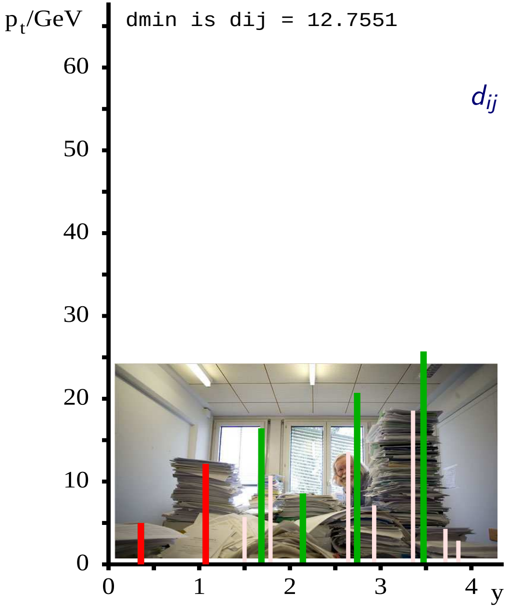
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

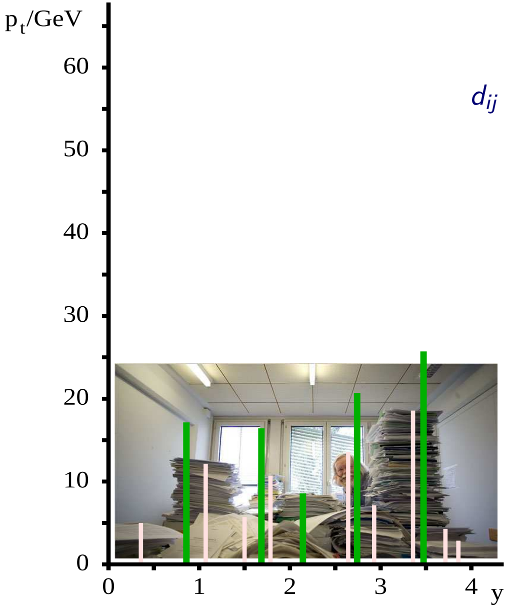
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

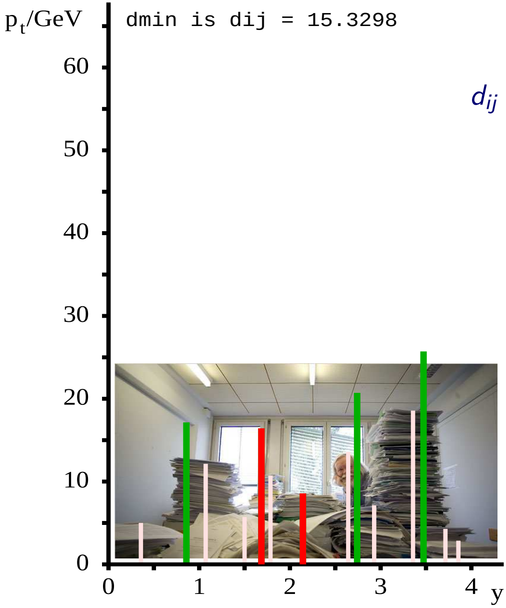
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

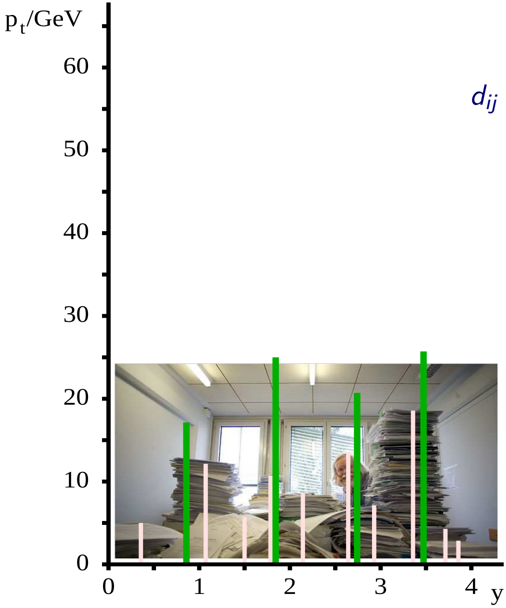
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

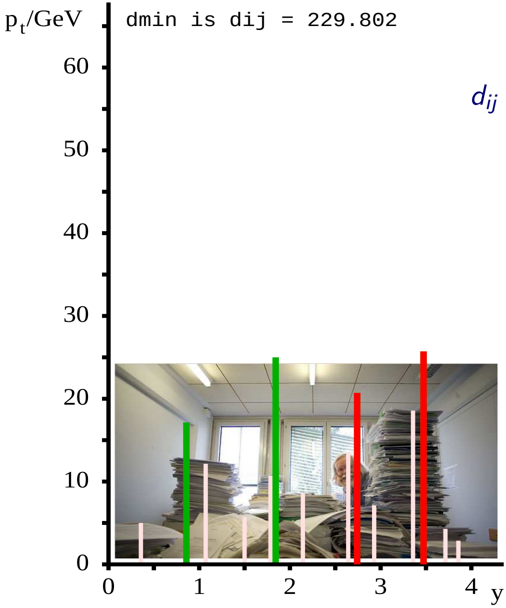
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

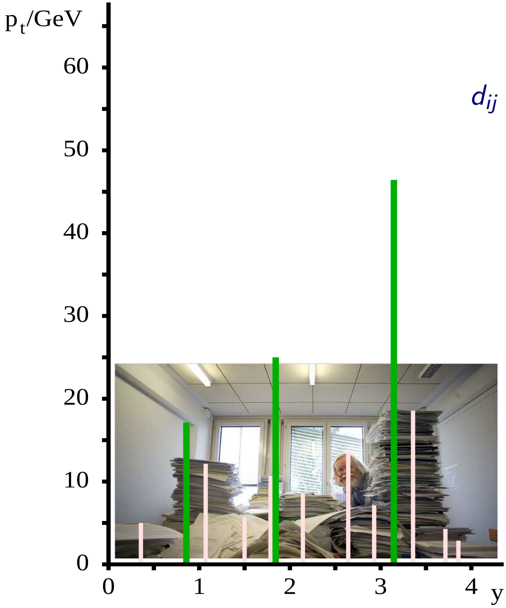
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

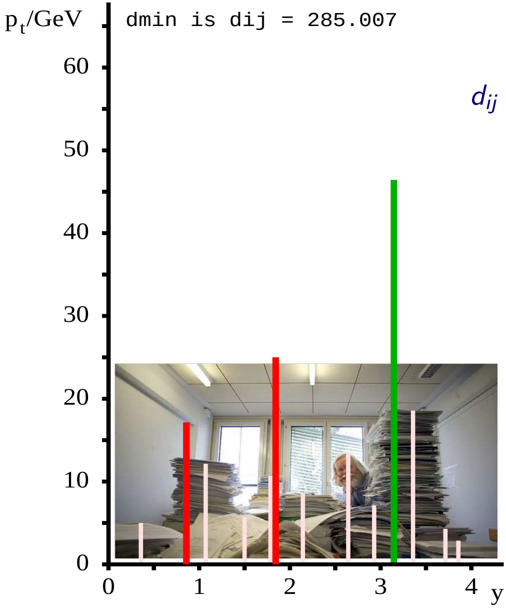
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

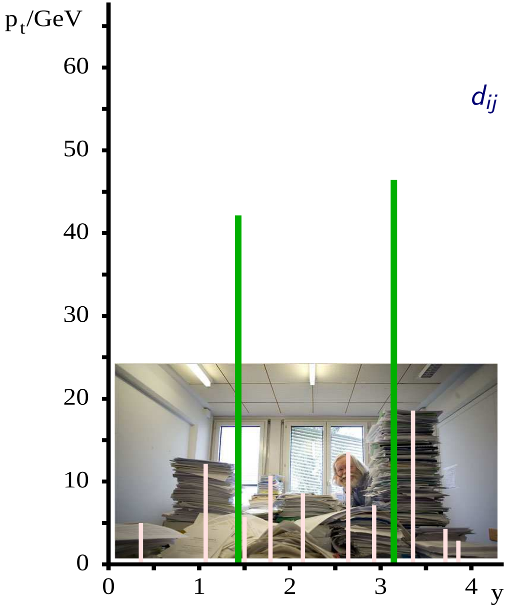
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



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

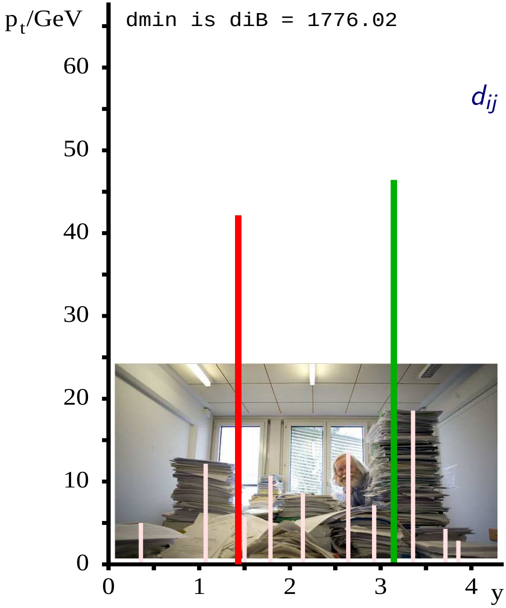
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

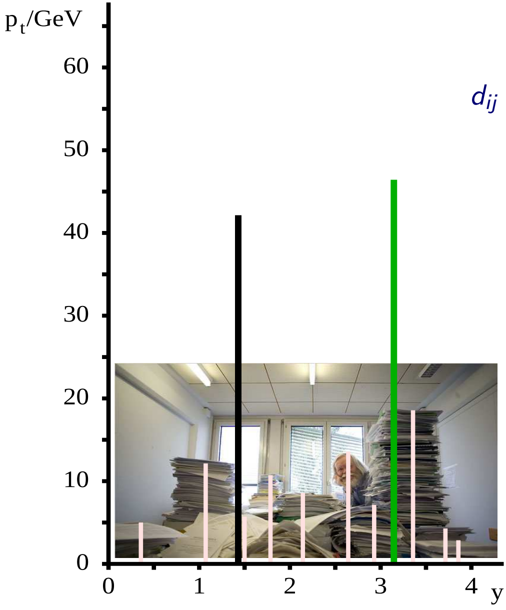
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

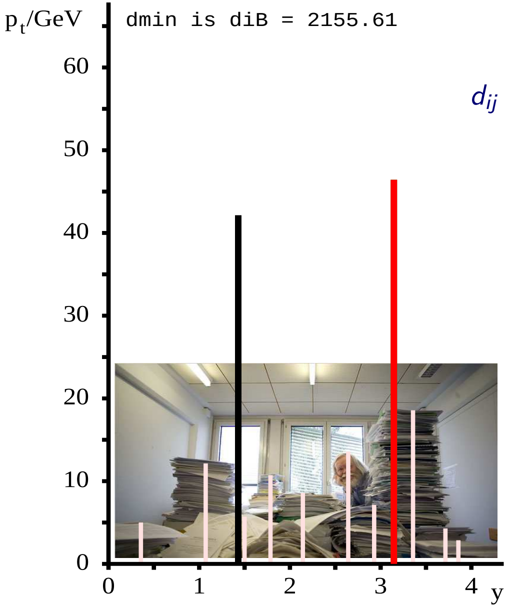
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

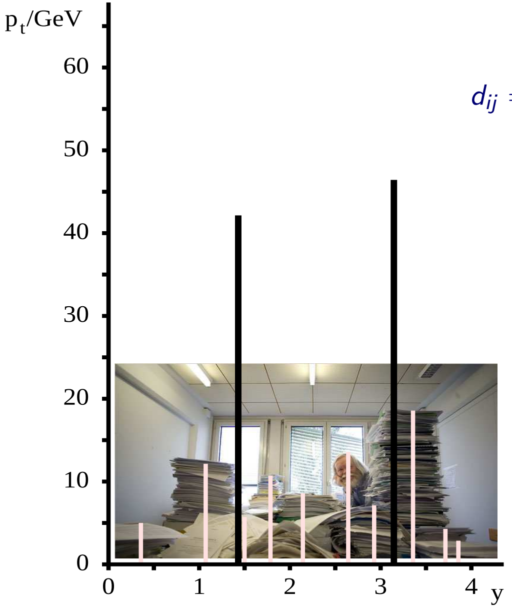
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = k_{ti}^2$$

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

k_t in action



k_t alg.: Find smallest of

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

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

Example clustering with k_t algorithm, $R = 1.0$

ϕ assumed 0 for all towers

The k_t algorithms form one of several "families" of sequential recombination jet algorithm

Others differ in:

1. the choice distance measure between pairs of particles
[i.e. the relative priority given to soft and collinear divergences, e.g. Cambridge/Aachen (C/A), which uses just angular distance]

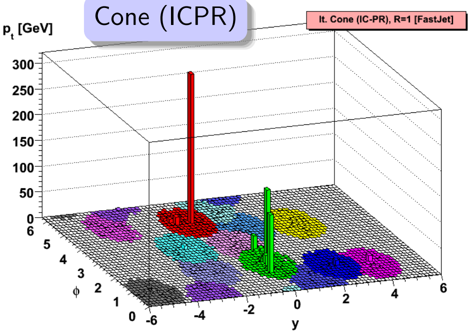
2. using $3 \to 2$ clustering rather than $2 \to 1$
[ARCLUS; not used at hadron colliders, so won't discuss it more]

The k_t family of algorithms was widely used at LEP (e^+e^-) and HERA (ep and γp).

Tevatron instead used **cone** algorithms, as did the LHC experiments during the design and planning stages.

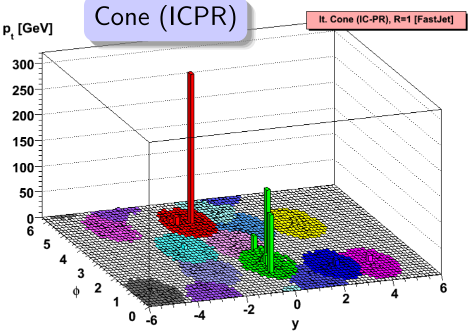
Essential characteristic of cones?

Cone (ICPR)



Essential characteristic of cones?

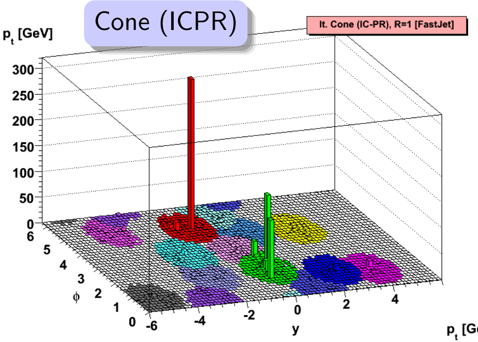
Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

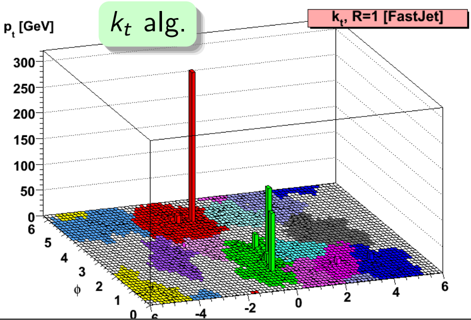
Much appreciated by experiments e.g. for acceptance corrections

Essential characteristic of cones?



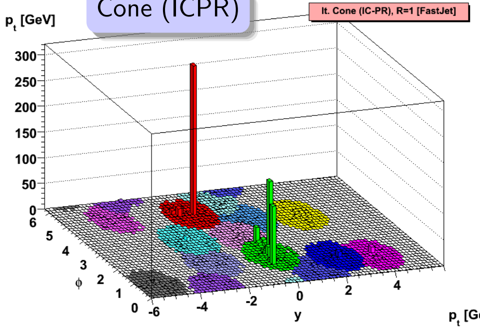
(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections



Essential characteristic of cones?

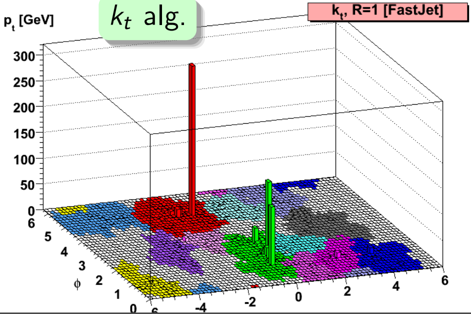
Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections

k_t alg.



k_t jets are **irregular**

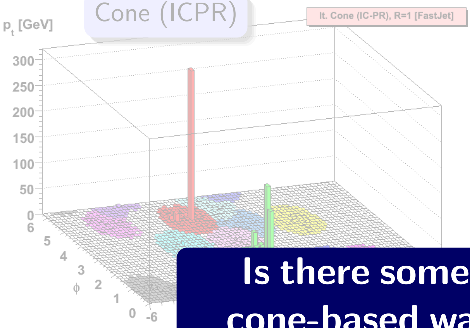
Because soft junk clusters together first:

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

Regularly held against k_t

Essential characteristic of cones?

Cone (ICPR)



(Some) cone algorithms give **circular** jets in $y - \phi$ plane

Much appreciated by experiments e.g. for acceptance corrections

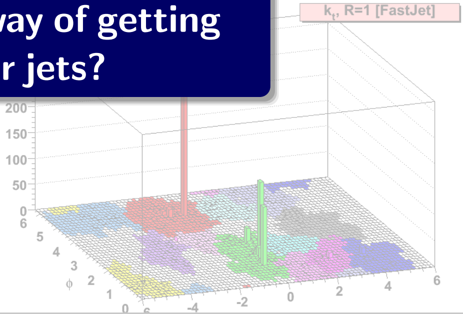
Is there some other, non cone-based way of getting circular jets?

k_t jets are **irregular**

Because soft junk clusters together first:

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

Regularly held against k_t



Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

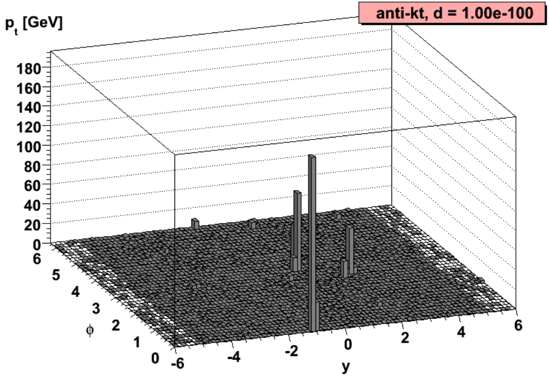
Hard stuff clusters with nearest neighbour

Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

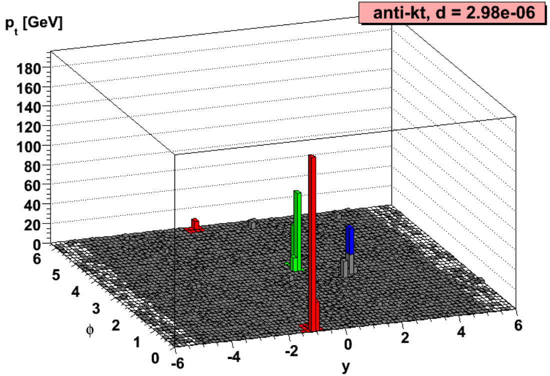


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

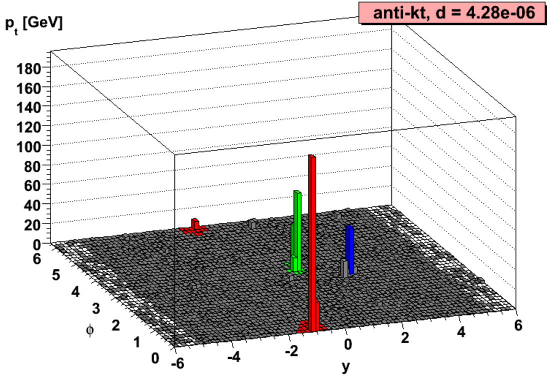


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_{\mathbf{t}}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

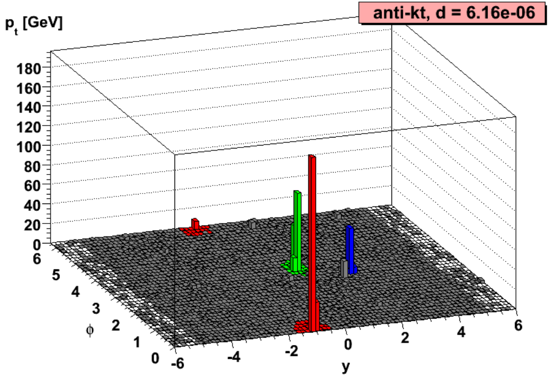


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\text{: } d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\text{: } d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

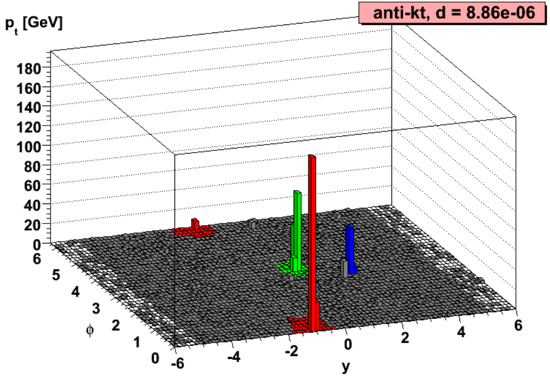


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

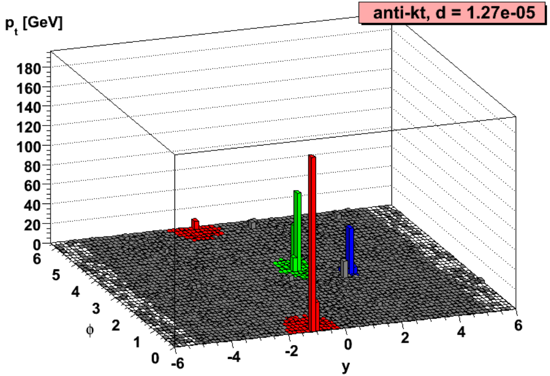


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

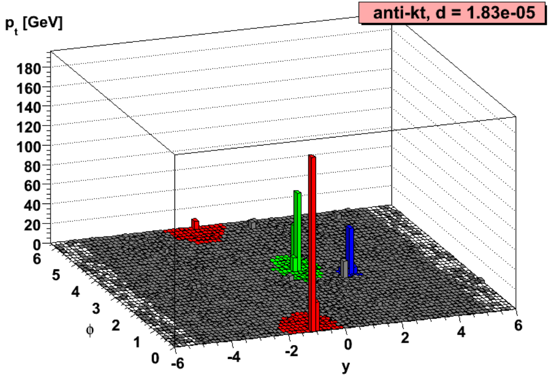


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

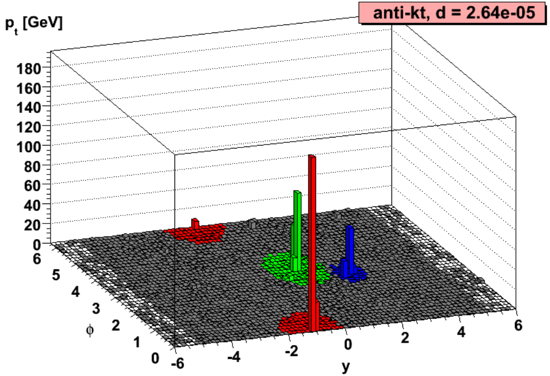


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

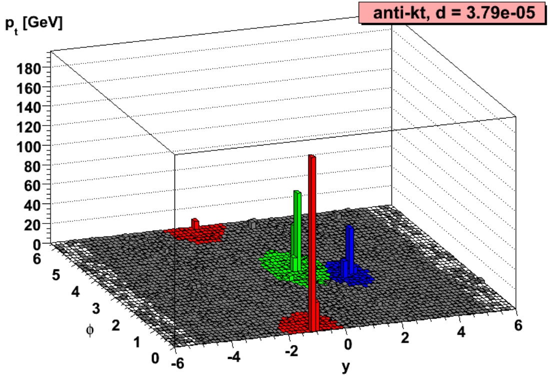


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

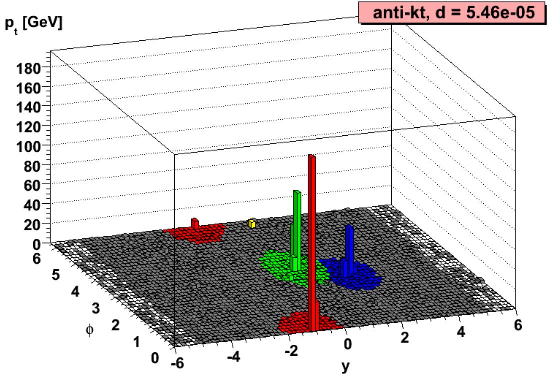


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t\colon\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}\colon\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

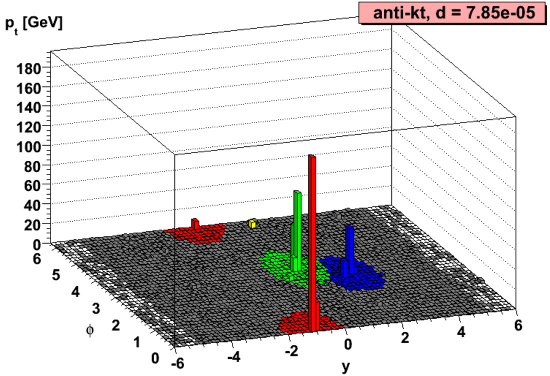


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour

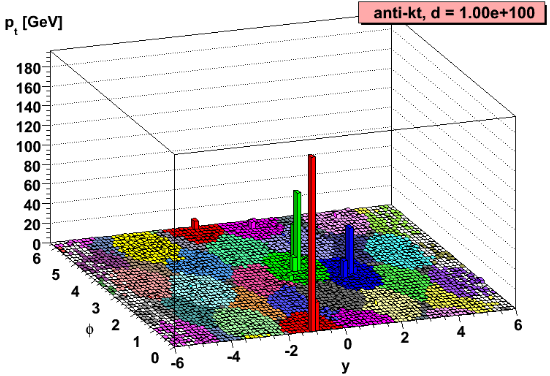


Adapting seq. rec. to give circular jets

Soft stuff clusters with nearest neighbour

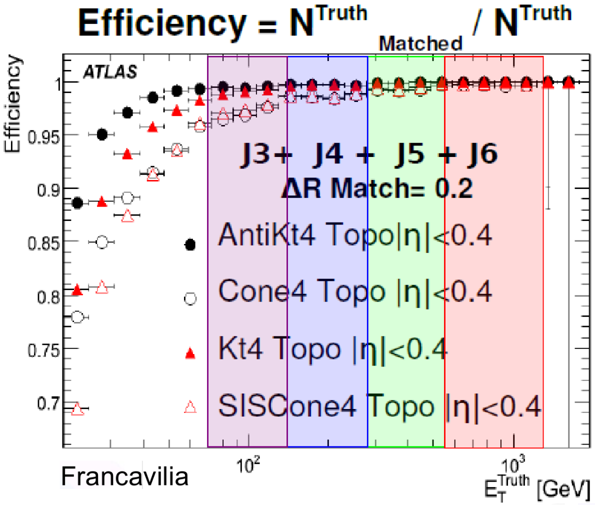
$$k_t:\ d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2 \longrightarrow \textbf{anti-k}_\mathbf{t}:\ d_{ij} = \frac{\Delta R_{ij}^2}{\max(k_{ti}^2, k_{tj}^2)}$$

Hard stuff clusters with nearest neighbour



anti-k_t gives cone-like jets without using stable cones

Anti-k_t experimental performance

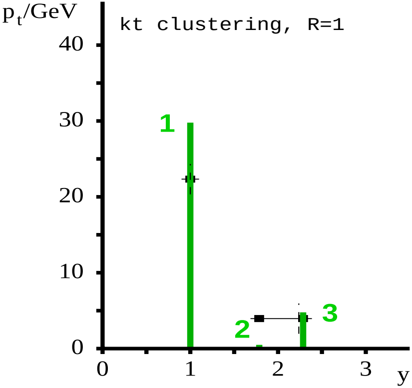


As good as, or better than all previous experimentally-favoured algorithms.

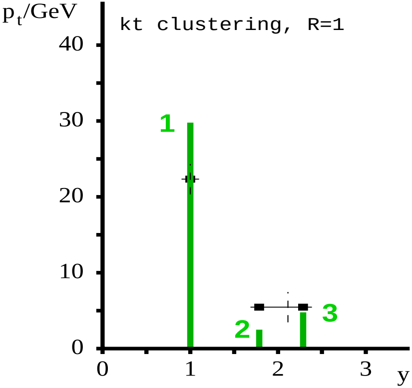
Essentially because anti-k_t has linear response to soft particles.

And it's also infrared and collinear safe (needed for theory calcs.)

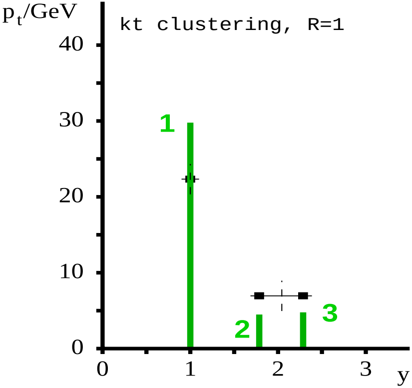
Linearity: k_t v. anti-k_t



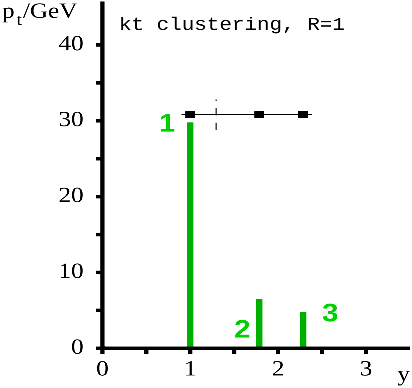
Linearity: k_t v. anti-k_t



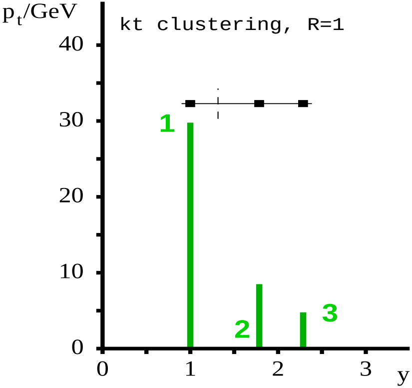
Linearity: k_t v. anti-k_t



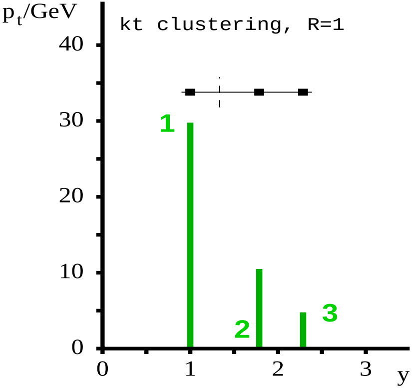
Linearity: k_t v. anti-k_t



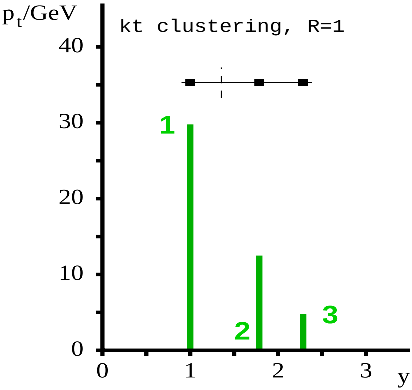
Linearity: k_t v. anti-k_t



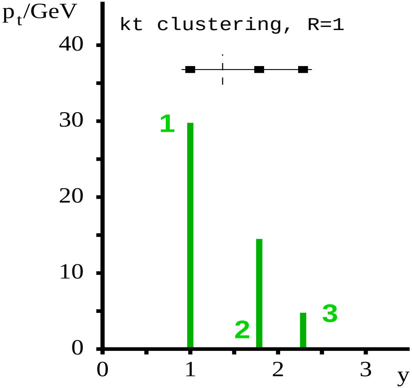
Linearity: k_t v. anti-k_t

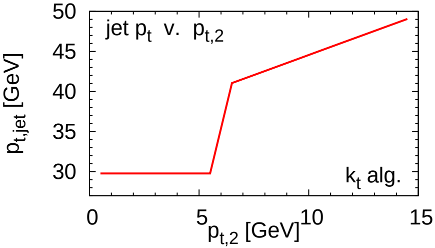


Linearity: k_t v. anti-k_t

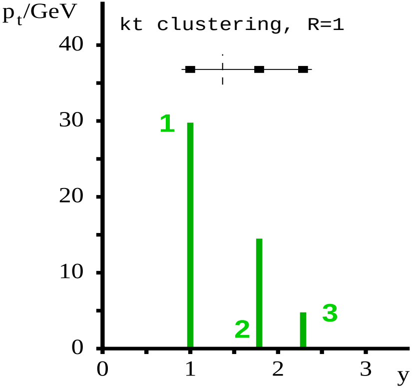


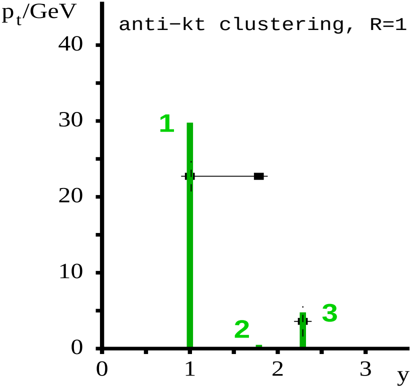
Linearity: k_t v. anti-k_t

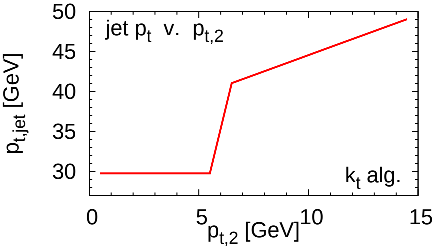




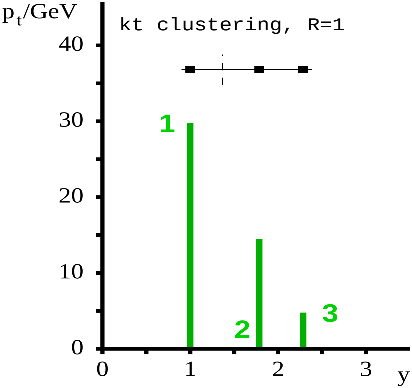
Linearity: k_t v. anti-k_t

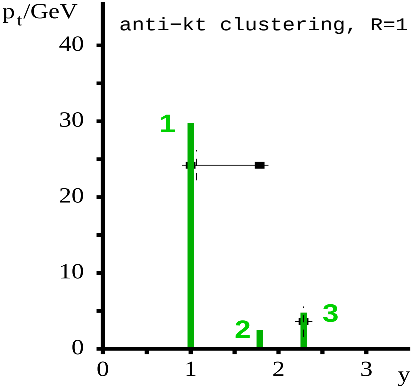


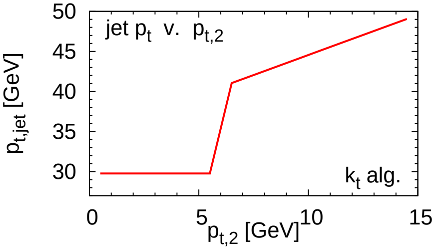




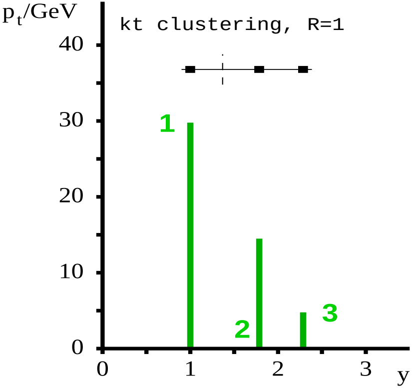
Linearity: k_t v. anti-k_t

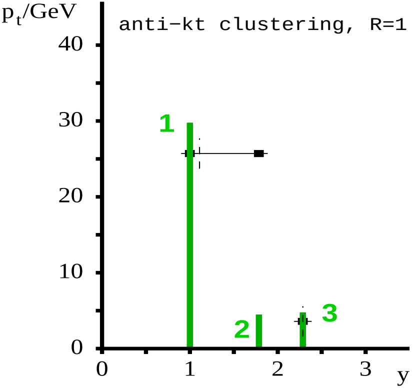


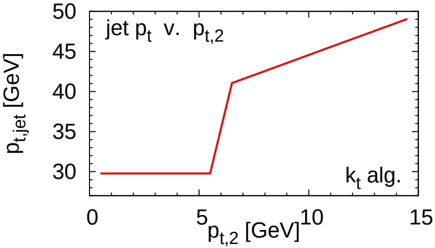




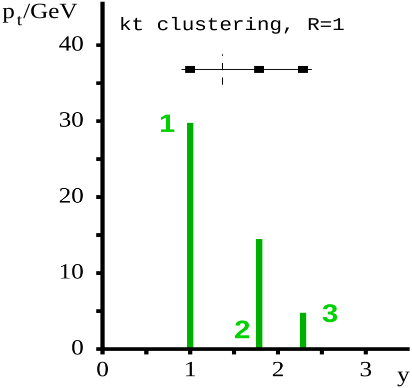
Linearity: k_t v. anti-k_t

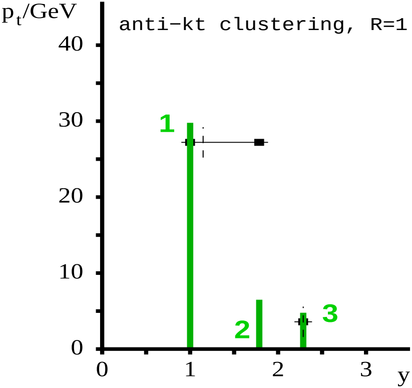


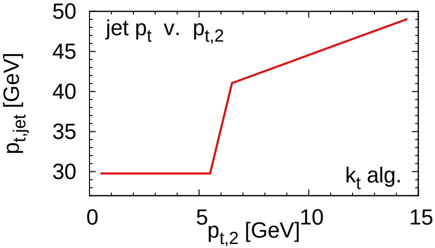




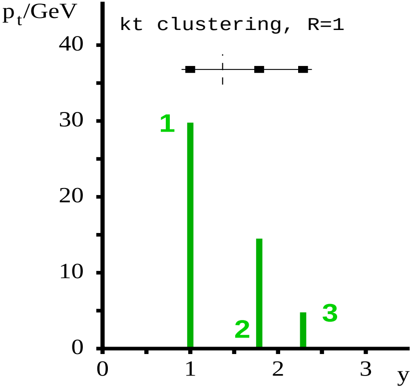
Linearity: k_t v. anti-k_t

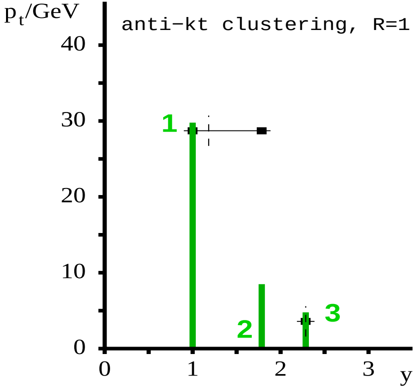


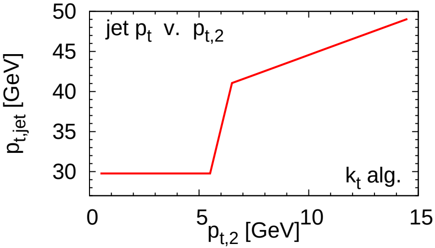




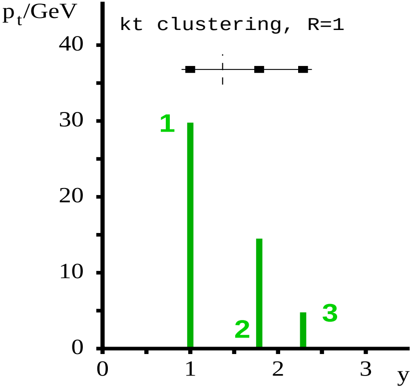
Linearity: k_t v. anti-k_t

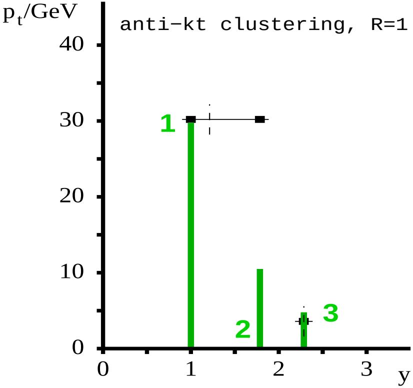


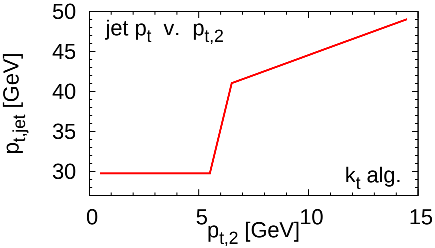




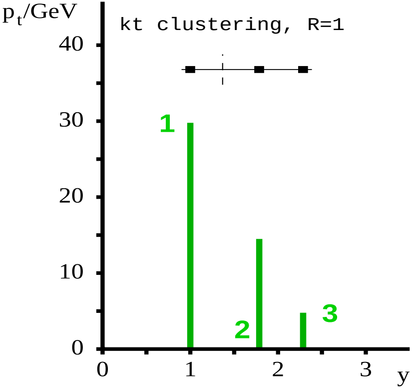
Linearity: k_t v. anti-k_t

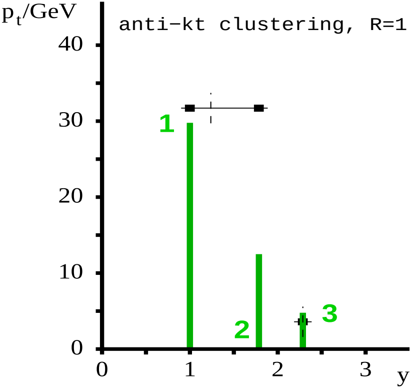


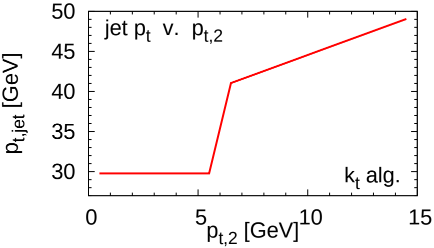




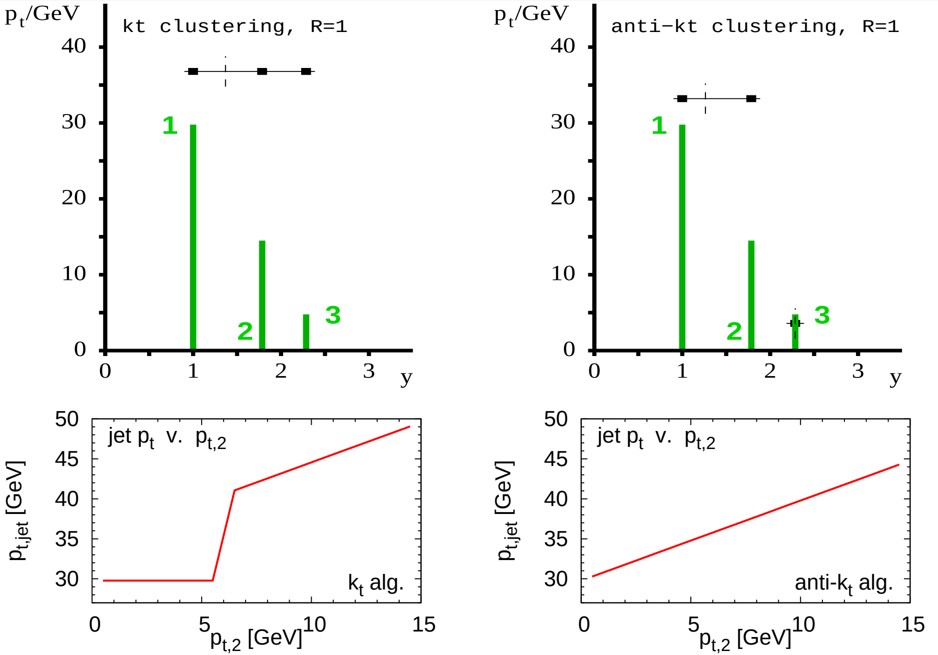
Linearity: k_t v. anti-k_t



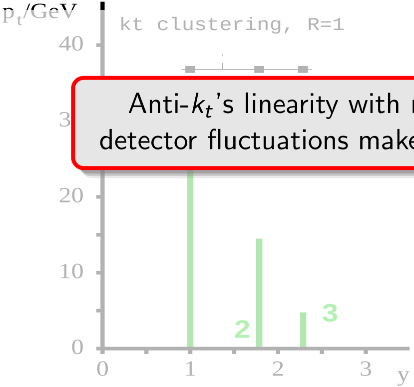


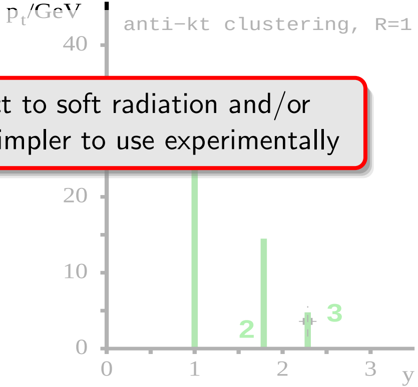


Linearity: k_t v. anti-k_t

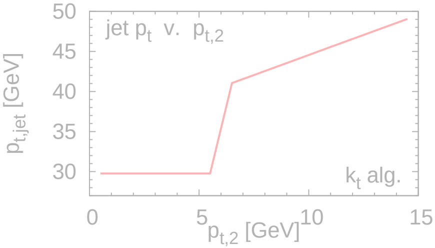


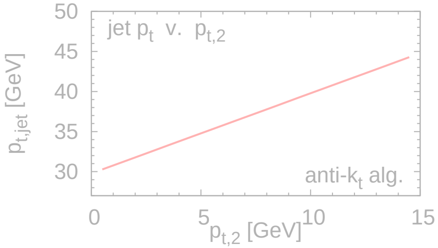
Linearity: k_t v. anti-k_t



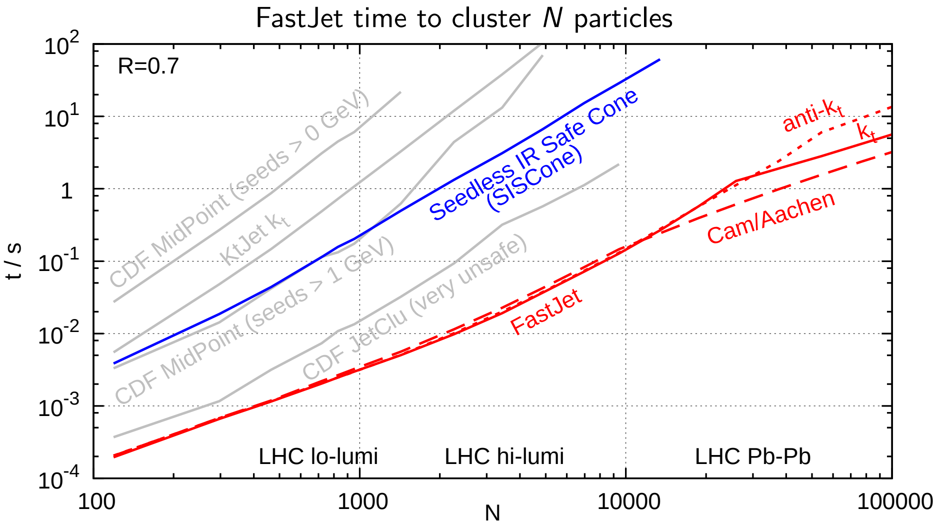


Anti-k_t's linearity with respect to soft radiation and/or detector fluctuations makes it simpler to use experimentally





Nearly all algorithms available in FastJet



Used by all the LHC experiments

Aavailable from `http://fastjet.fr/`

Summary

Today we've examined why we need jets and looked at some of the logic behind the way they're defined.

Of the different algorithms we've discussed, the one that's most widely used at LHC today is anti-k_t.

But the other algorithms we've seen will also play a role at the LHC.

Tomorrow's subject will be *making the best use of jets*.