LPNHE Paris
June 20, 2008

Jet Algorithms

Matteo Cacciari
LPTHE Paris

Jet ‘Finder’ v. Jet ‘Algorithm’

A jet **finder** can be thought of as something which gets as close as possible to some initial parton, collecting (most of) its energy. You could just eyeball it

The purpose (and role) of a jet **algorithm** is to **reduce the complexity** of the final state, simplifying many hadrons to **simpler objects** that one can hope to **calculate**

It’s just semantics, but it sets the stage...

Jet Algorithm

A **jet algorithm** maps the momenta of the final state particles into the momenta of a certain number of jets:

$$\{p_i\} \xrightarrow{\text{jet algorithm}} \{j_k\}$$

particles,
4-momenta,
calorimeter towers, ...

jets

Most algorithms contain a resolution parameter, **R**, which controls the extension of the jet

Jet Definition

Les Houches 2007 proceedings, arXiv:0803.0678

jet definition

$\{p_i\}$ $\xrightarrow{\text{jet algorithm}}$ $\{j_k\}$

particles,
4-momenta,
calorimeter towers, ...

jets

+ parameters (usually at least the radius R)

+ recombination scheme

Reminder: running a jet definition gives a well defined physical observable, which we can measure and, hopefully, calculate

Two main classes of jet algorithms

Sequential recombination algorithms

bottom-up approach: combine particles starting from **closest one** in some distance measure. Repeat until few left: call them jets

Work because of mapping closeness ⇔ QCD divergence
Loved by e+e-, ep and theorists

Infrared safe but naively slow with many particles ($O(N^3)$)

[One day to cluster 30000 particles, a single heavy ions collision at LHC]

Cone algorithms

top-down approach: find coarse regions of energy flow. How? Find **stable** cones (i.e. their axis coincides with sum of momenta of particles in it)

Work because QCD only modifies energy flow on small scales
Loved by pp and (fewer) theorists

Not Infrared/Collinear safe in approximate implementations
Apparently impossibly slow ($O(N2^N)$) if done exactly

[10^{17} years to cluster just 100 particles]

Jet Algorithm

In order to work properly jet algorithms should be

- ❑ Infrared safe (for the calculation to converge)
  soft emission shouldn't change jets
  collinear splitting shouldn't change jets

- ❑ fast (if many particles have to be clustered)
  In order to be realistically
  applicable at detector level

These (and other) requirements where widely acknowledged as early as 1990 (Snowmass accord)

Perhaps quite surprisingly, they had not yet been met in 2005

A calorimeter is of course always infrared safe.
You need these properties not so much to find jets, but to be able to calculate them in pQCD

A (long!) list of cones

Les Houches 2007 proceedings, arXiv:0803.0678

CDF JetClu	IC_r-SM
CDF MidPoint cone	IC_{mp}-SM
CDF MidPoint searchcone	$IC_{se,mp}$-SM
D0 Run II cone	IC_{mp}-SM
ATLAS Cone	IC-SM
PxCone	IC_{mp}-SD
CMS Iterative Cone	IC-PR
PyCell/CellJet (from Pythia)	FC-PR
GetJet (from ISAJET)	FC-PR

IC = Iterative Cone
SM = Split-Merge
SD = Split-Drop
FC = Fixed Cone
PR = Progressive Removal

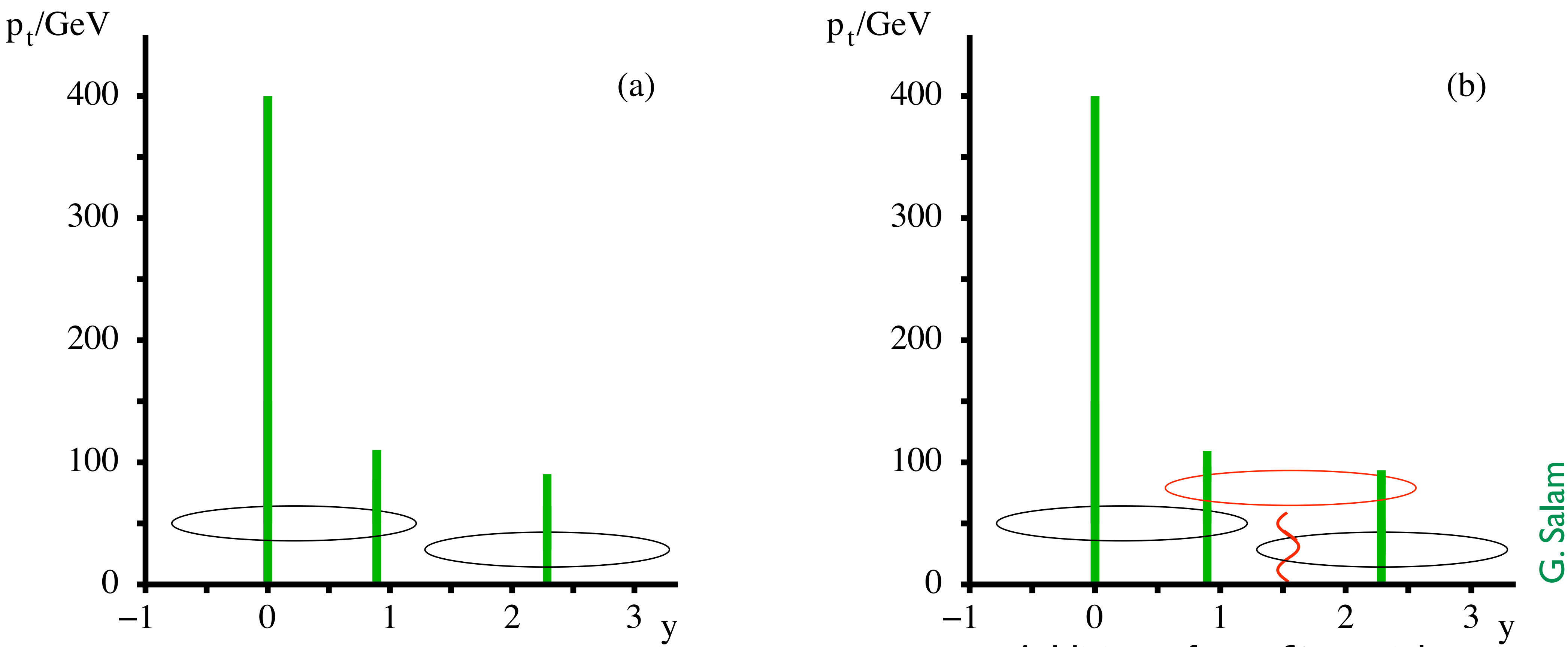
type of algorithm

Example of IC-SM: **Midpoint** cone

- Begin with seed particles
- Cluster particles into cone if $\Delta R < R$
- Iterate until stable (i.e. axis coincide with sum of momenta) cones found
- Start new search cones at midpoint of stable cones
- Merge jets if overlapping energy is > f times the energy of the smaller jet

Use of seeds is the most problematic issue

MidPoint Cone Infrared Unsafety



Three hard particles clustered into **two** cones by the MidPoint algorithm

Addition of a **soft** particle changes the hard jets: **three** stable cones are now found

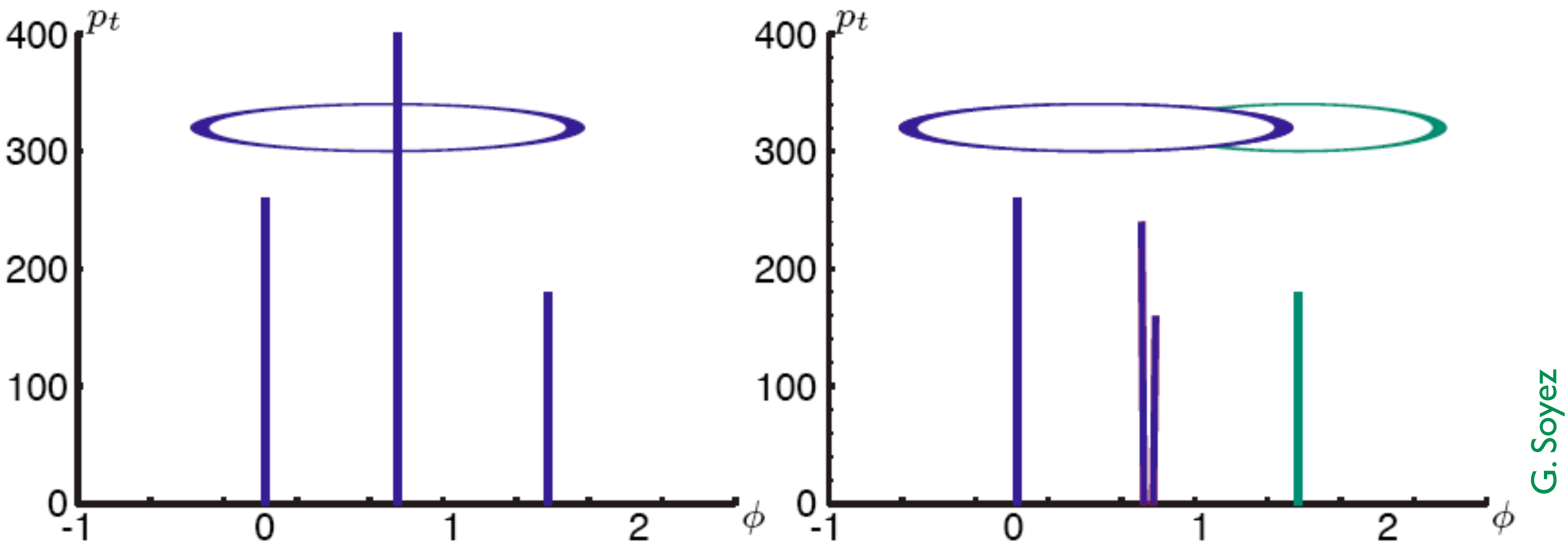
The problem is that the specific stable-cone search procedure used by MidPoint cannot find **all** possible stable cones

Example of IC-PR (e.g. CMS cone)

- Begin with hardest particle as seed
- Cluster particles into cone if $\Delta R < R$
- Iterate until stable (i.e. axis coincide with sum of momenta) cones found
- Eliminate constituents of jet and start over from hardest remaining particle

NB. This is a very different algorithm from previous one. Many physics aspects differ.

IC-PR Cone Collinear Unsafety



Splitting the hardest particle **collinearly** changes the number of final jets

A (long!) list of cones (all eventually unsafe)

Les Houches 2007 proceedings, arXiv:0803.0678

CDF JetClu	IC_r-SM	IR_{2+1}
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}
CDF MidPoint searchcone	$IC_{se,mp}$-SM	IR_{2+1}
D0 Run II cone	IC_{mp}-SM	IR_{3+1}
ATLAS Cone	IC-SM	IR_{2+1}
PxCone	IC_{mp}-SD	IR_{3+1}
CMS Iterative Cone	IC-PR	$Coll_{3+1}$
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$

IC = Iterative Cone
SM = Split-Merge
SD = Split-Drop
FC = Fixed Cone
PR = Progressive Removal

type of algorithm

safety issue

IR_{n+1} : unsafe when a soft particle is added to n hard particles in a common neighbourhood

$Coll_{n+1}$: unsafe when one of n hard particles in a common neighbourhood is split collinearly

Lessons

- There isn't **one** cone algorithm, but rather many different cones, which can behave quite distinctly from one another
- Essentially all of the cones commonly used are unsafe at some point. The best ones only fail at NNLO (3+1), others already at NLO (2+1)

Examples:

	Last meaningful order		
	ATLAS cone [IC-SM]	MidPoint [IC_{mp}-SM]	CMS it. cone [IC-PR]
Inclusive jets	LO	NLO	NLO
$W/Z + 1$ jet	LO	NLO	NLO
3 jets	none	LO	LO
$W/Z + 2$ jets	none	LO	LO
m_{jet} in $2j + X$	none	none	none

G. Salam

Calculations cost real money:

~ 100 theorists ×15 years ≈100 M€

Using unsafe jet tools essentially renders them useless

Solutions?

SISCone

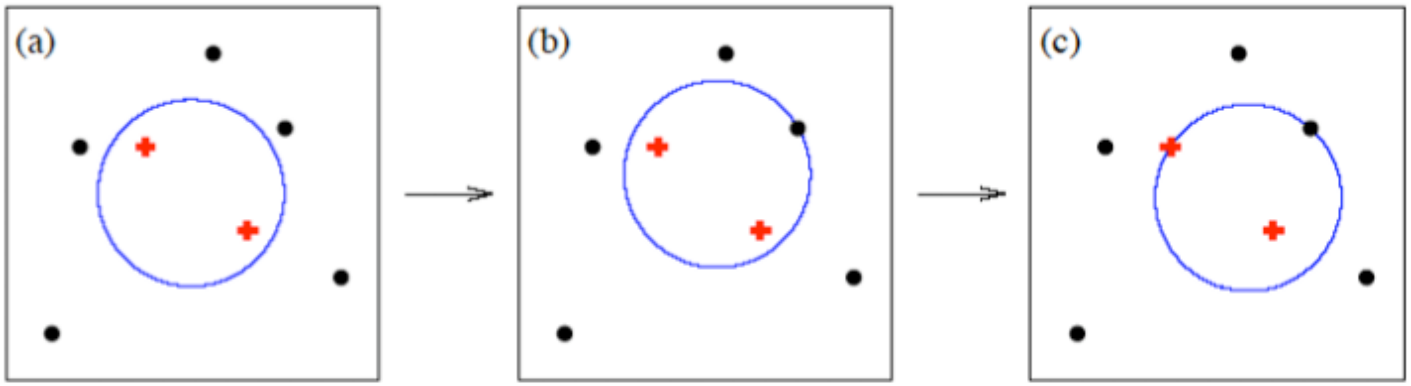
Checking **all particles** in an event to test for stable combinations

(i.e. the axis of the cone containing a subset of particles coincides with the momentum sum)

takes $O(N2^N)$ time

Solution: once more, transform into a geometrical problem

1. Find all distinct way of enclosing a set of particles in a y-ϕ circle
2. Check, for each enclosure, if it corresponds to a stable cone



Finding all distinct circular enclosures of a set of points is geometry:
move it until you hit a point, then rotate it until one of the points hits the edge

Result: Seedles Infrared Safe Cone (SISCone)

[runs in $O(N^2 \ln N)$ time, similar to MidPoint (N^3)]

[Salam, Soyez, arXiv: 0704.0292]

Infrared (un)safety

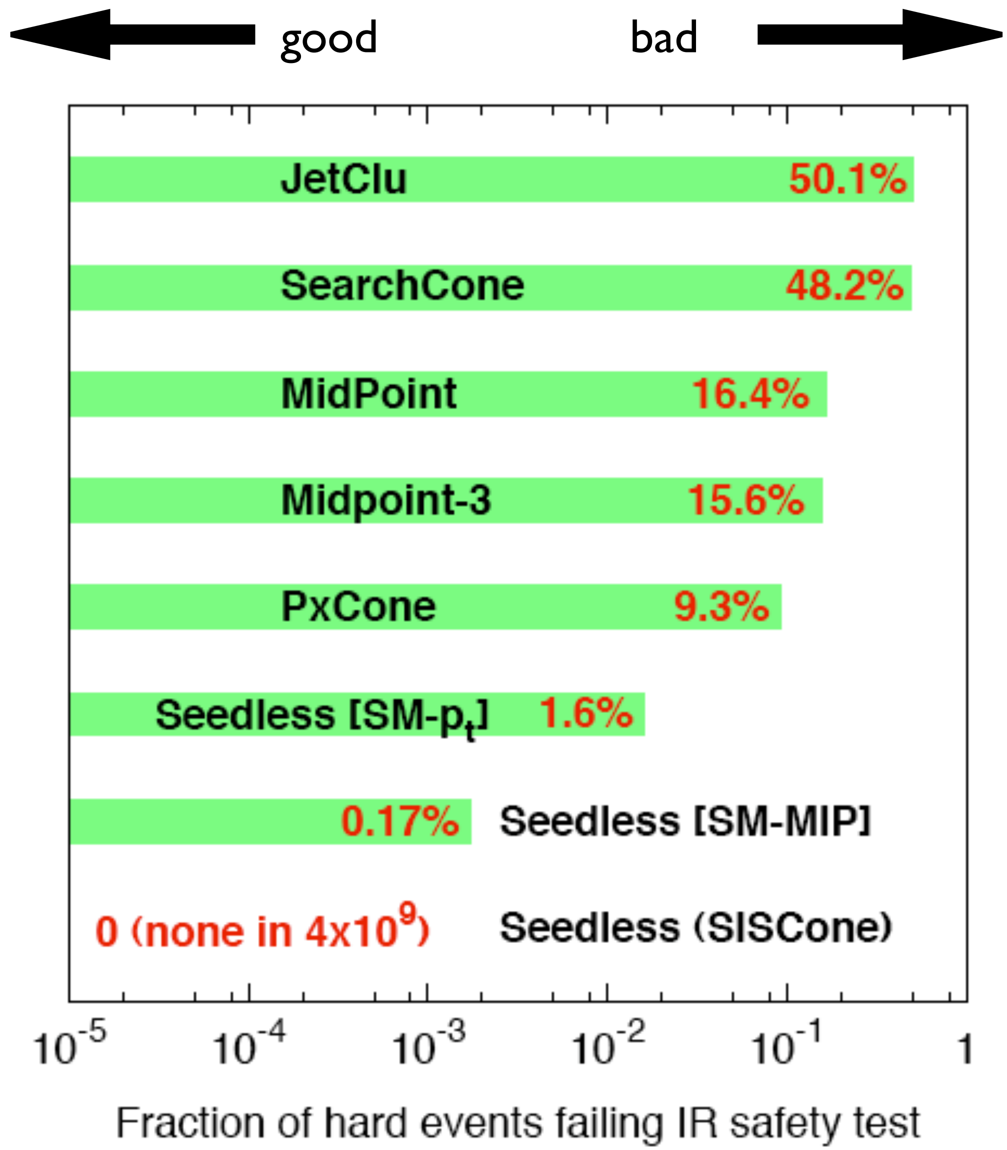
Q: How often are the hard jets changed by the addition of a soft particle?

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

A:

Unsafety level	failure rate
2 hard + 1 soft	$\sim 50\%$
3 hard + 1 soft	$\sim 15\%$
SISCone	IR safe !

Be careful with split–merge too



Salam & Soyez

A sequential recombination jet algorithm

- Calculate distances between particles $d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta y^2 + \Delta\phi^2}{R^2}$
- Calculate the beam distances: $d_{iB} = k_{ti}^{2p}$
- Combine particles with smallest distance or, if d_{iB} is smallest, call it a jet
- Find again smallest distance and repeat procedure until no particles are left

This is always infrared and collinear safe

Best known example: the k_t algorithm

The Cambridge/Aachen and the anti-k_t algorithm

One can generalise the k_t distance measure:

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta y^2 + \Delta\phi^2}{R^2} \qquad\qquad d_{iB} = k_{ti}^{2p}$$

p = 1 k_t algorithm

S. Catani, Y. Dokshitzer, M. Seymour and B. Webber, Nucl. Phys. B406 (1993) 187
S.D. Ellis and D.E. Soper, Phys. Rev. D48 (1993) 3160

p = 0 Cambridge/Aachen algorithm

Y. Dokshitzer, G. Leder, S.Moretti and B. Webber, JHEP 08 (1997) 001
M.Wobisch and T.Wengler, hep-ph/9907280

p = -1 anti-k_t algorithm

MC, G. Salam and G. Soyez, arXiv:0802.1189

NB: in anti-kt pairs with a **hard** particle with cluster first: if no other hard particles are close by, the algorithm will give **perfect cones**

'[Equivalent to the reverse-k_t coded in ATLAS software, P.-A. Delsart and P. Loch]'

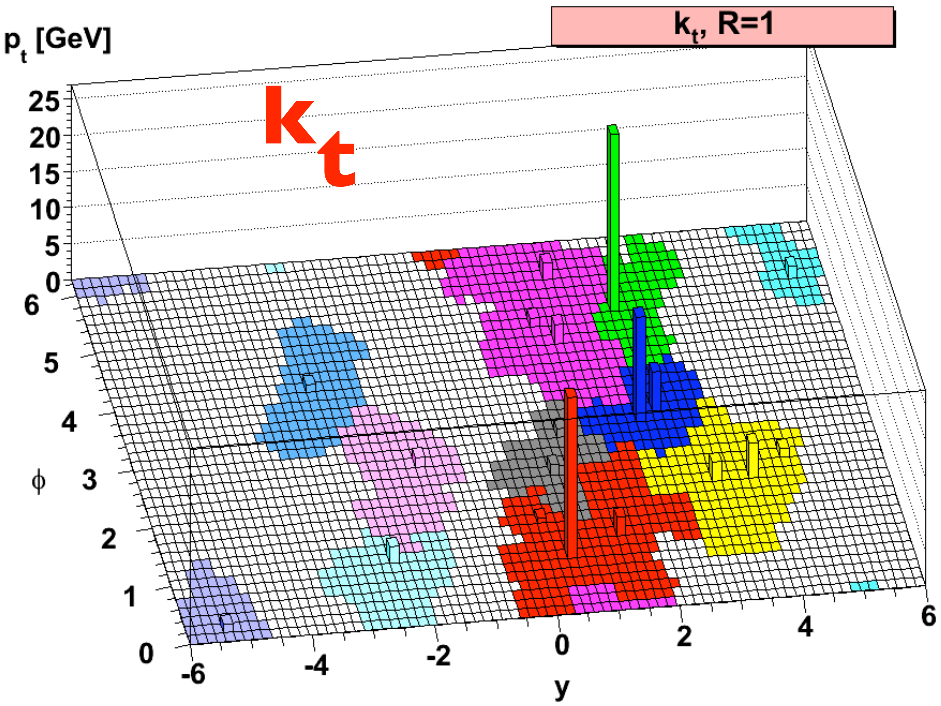
Quite ironically, a sequential recombination algorithm is the perfect cone algorithm

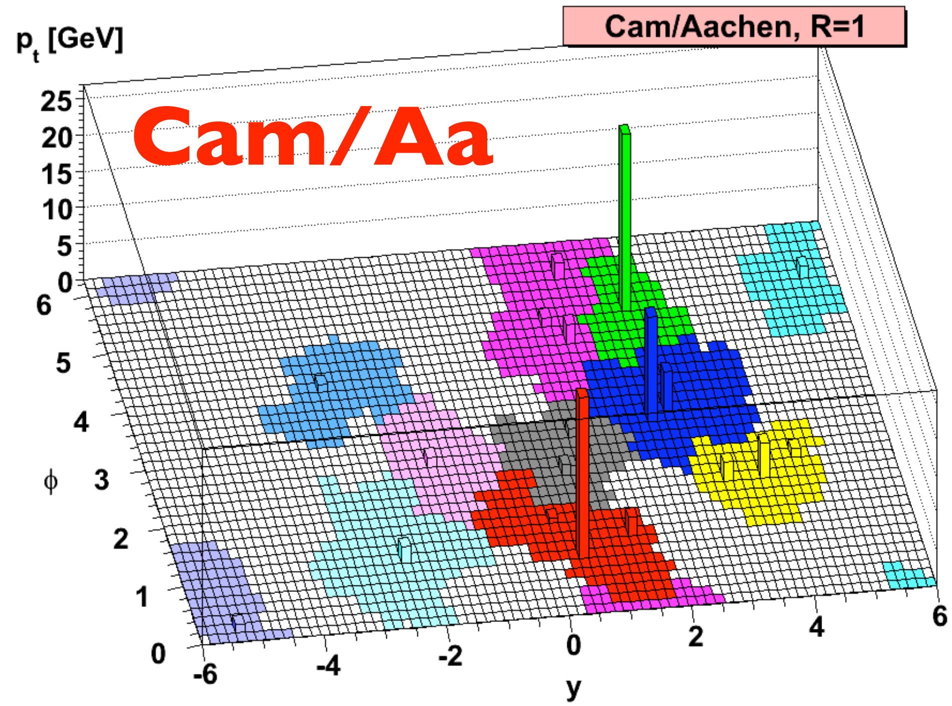
The IRC safe algorithms

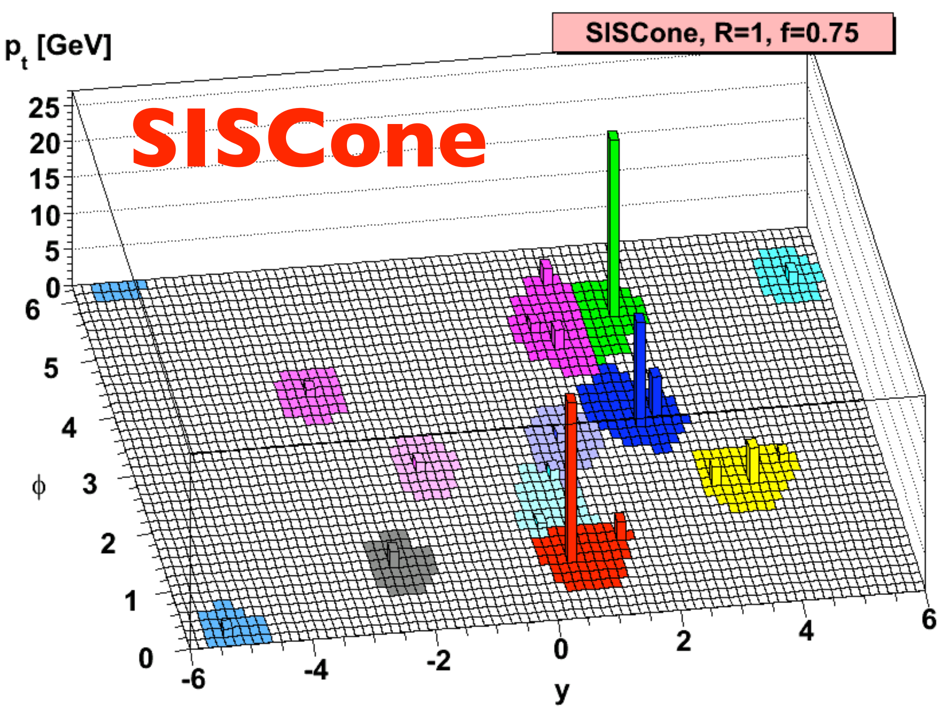
k_t SR, $d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2$ hierarchical in rel $\perp$ momenta	Cambridge/Aachen SR, $d_{ij} = \Delta R_{ij}^2/R^2$ hierarchical in angle
anti-k_t SR, $d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2$ gives perfectly conical jets	SISCone Seedless Infrared Safe cone +SM gives "economical" jets

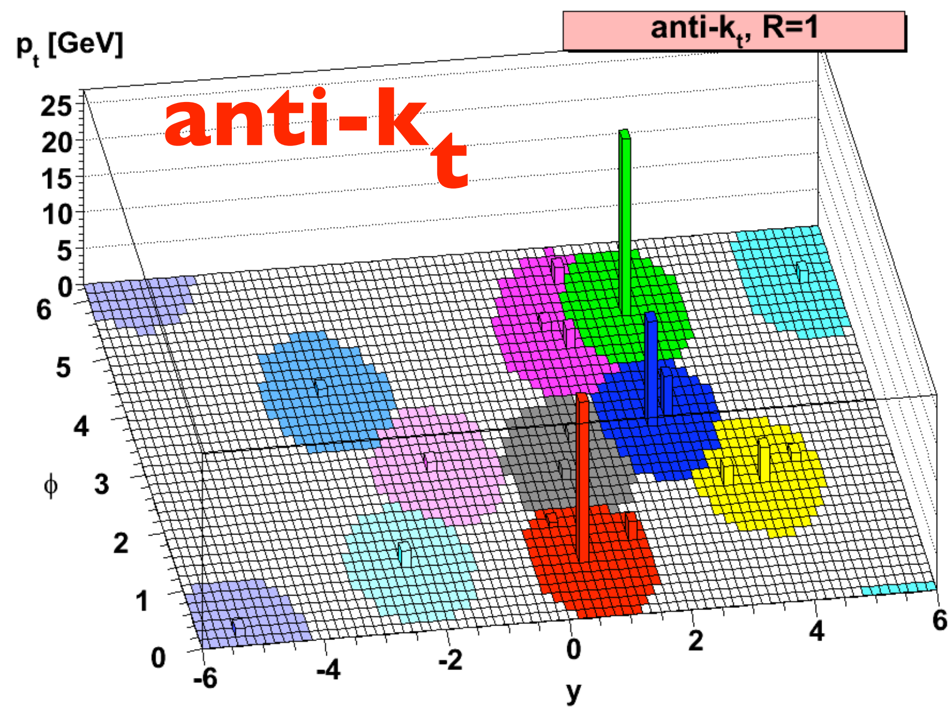
G. Salam

All are available in FastJet, http://www.lpthe.jussieu.fr/~salam/fastjet


k$_t$, R=1
k$_t$
p$_t$ [GeV]
ϕ
y

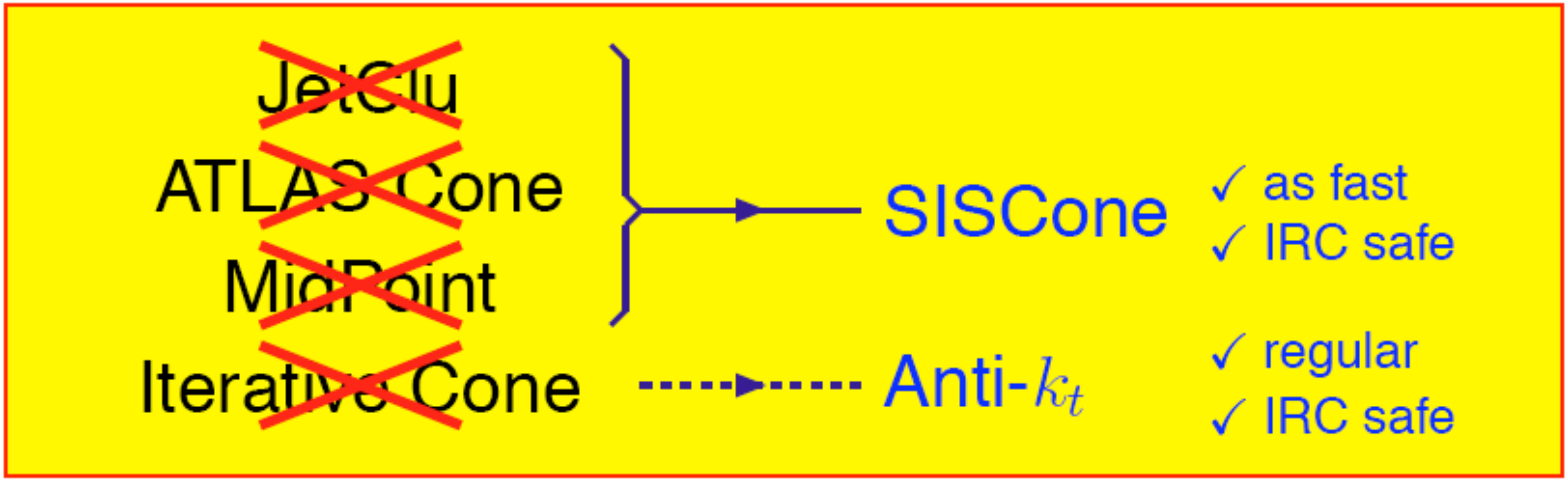

Cam/Aachen, R=1
Cam/Aa
p$_t$ [GeV]
ϕ
y


SISCone, R=1, f=0.75
SISCone
p$_t$ [GeV]
ϕ
y


anti-k$_t$, R=1
anti-k$_t$
p$_t$ [GeV]
ϕ
y

Replacements

If you care about IRC safety but don't want to stray too far from algorithms used so far, these are possible replacements:



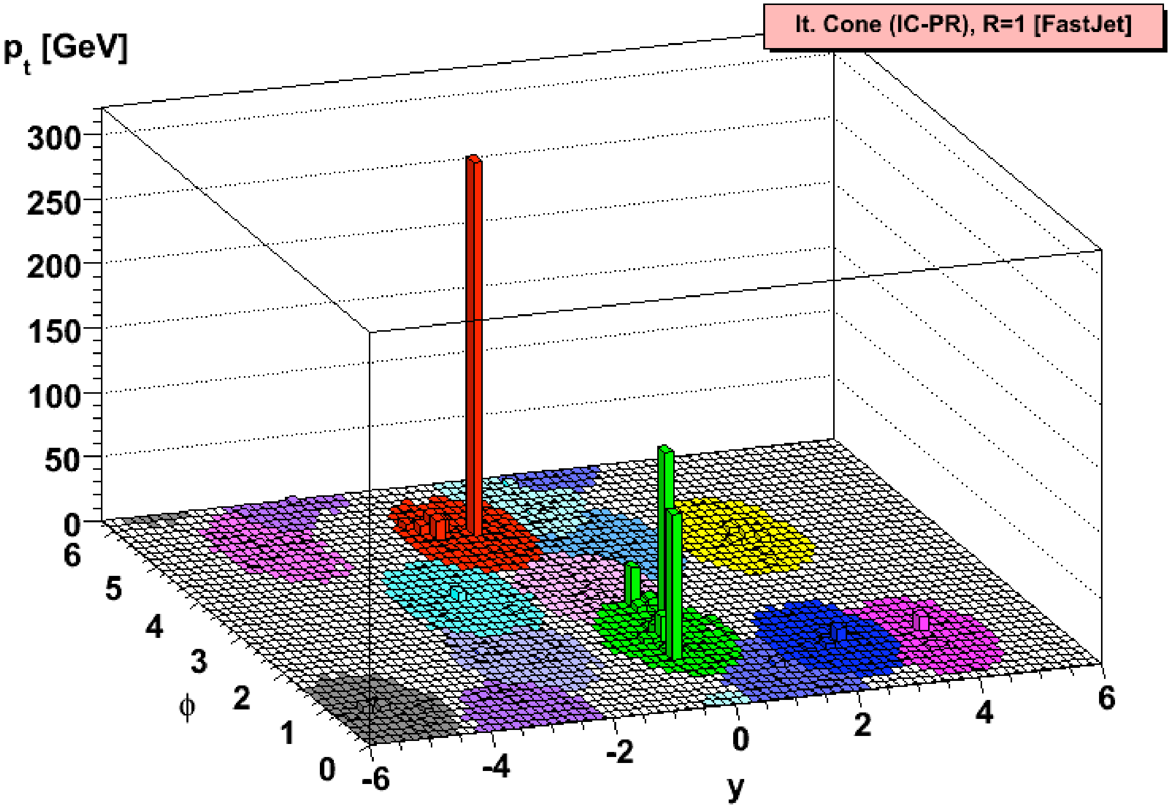
In addition, kt and Cambridg/Aachen will provide further flexibility

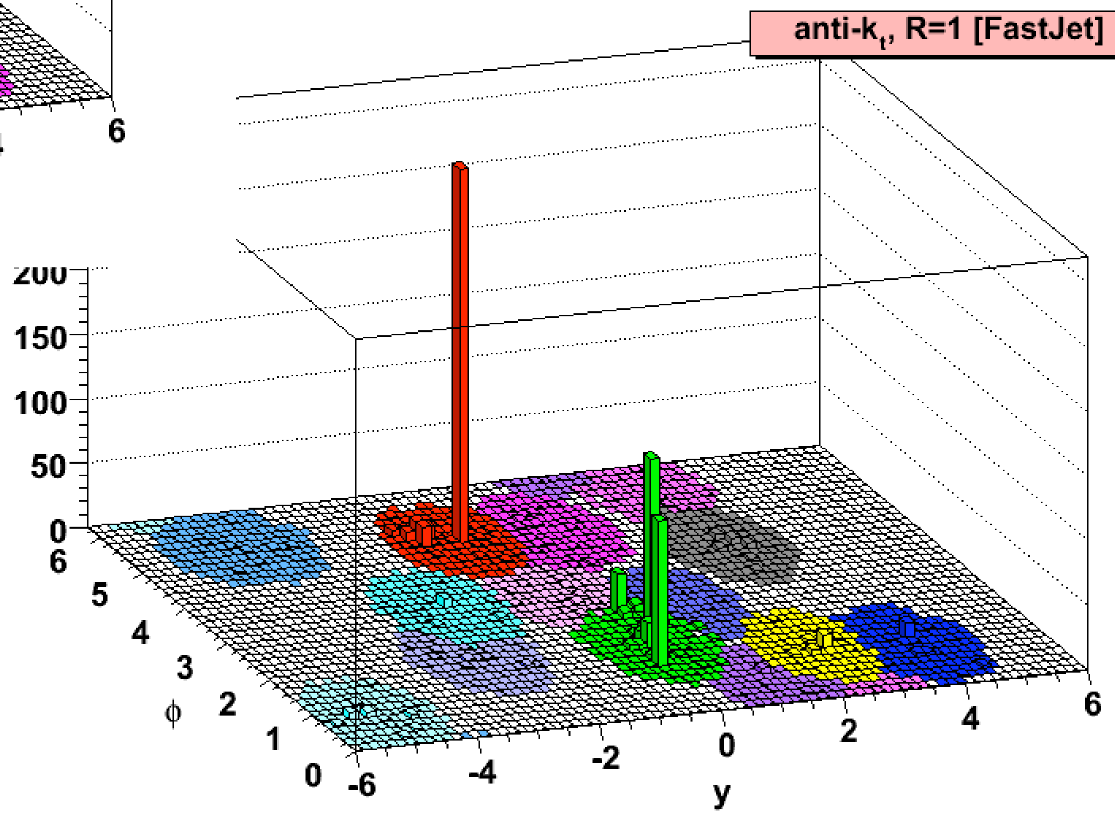
G. Soyez

Different algorithms spanning a series of different and complementary characteristics: should be enough for most purposes

Our proposal is to concentrate on these, both for analytical understanding and practical use in experiments, rather than using IRC unsafe ones

Example: IC-PR v. anti-k_t





Hardest jets are similarly regular

Why many different algorithms?

Advantages of flexibility

Resolution of mass peaks

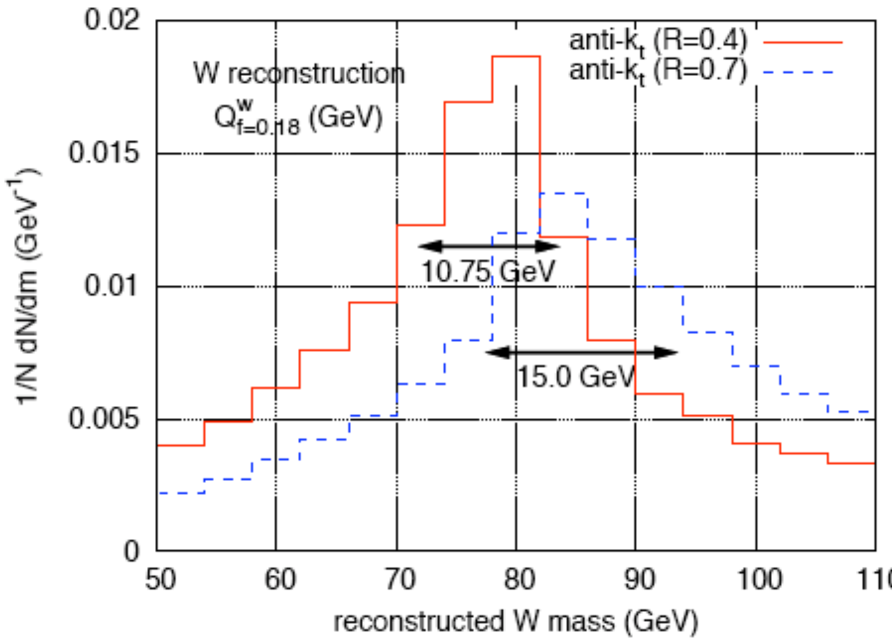
To compare different algorithms, define **figures of merit**:

Les Houches 2007 proceedings, arXiv:0803:0678

1. $Q^{w}_{f=z}(R)$

Smallest width of an histogram window containing a fraction f=z of the generated objects, i.e.

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

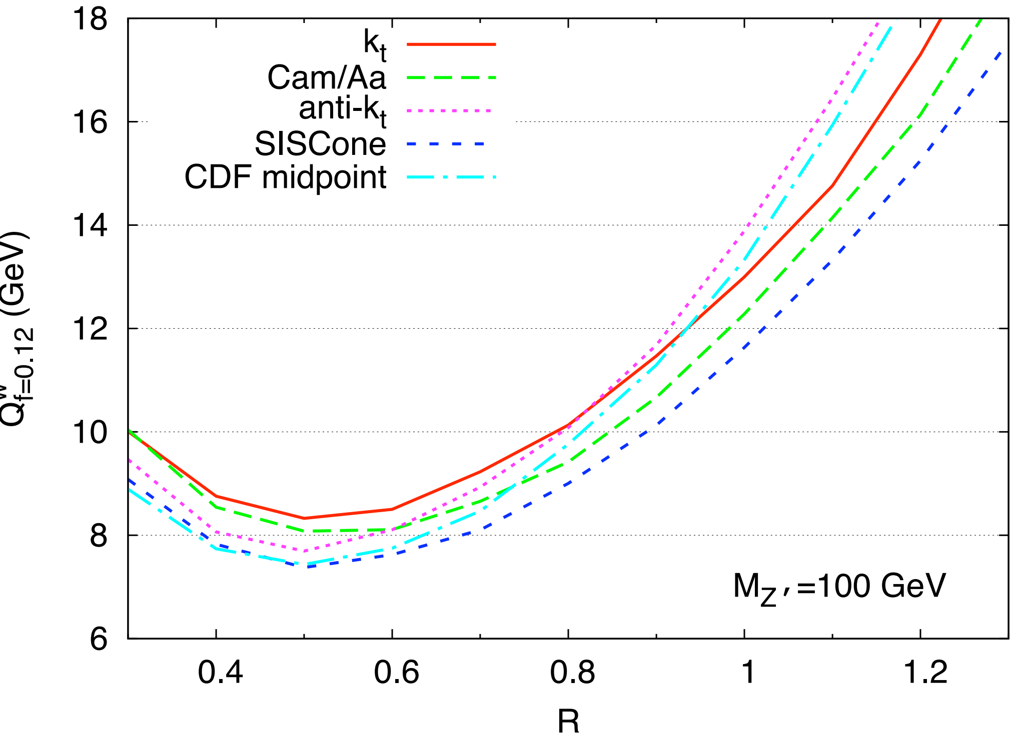


2. Maximum fraction of events in window of given width:

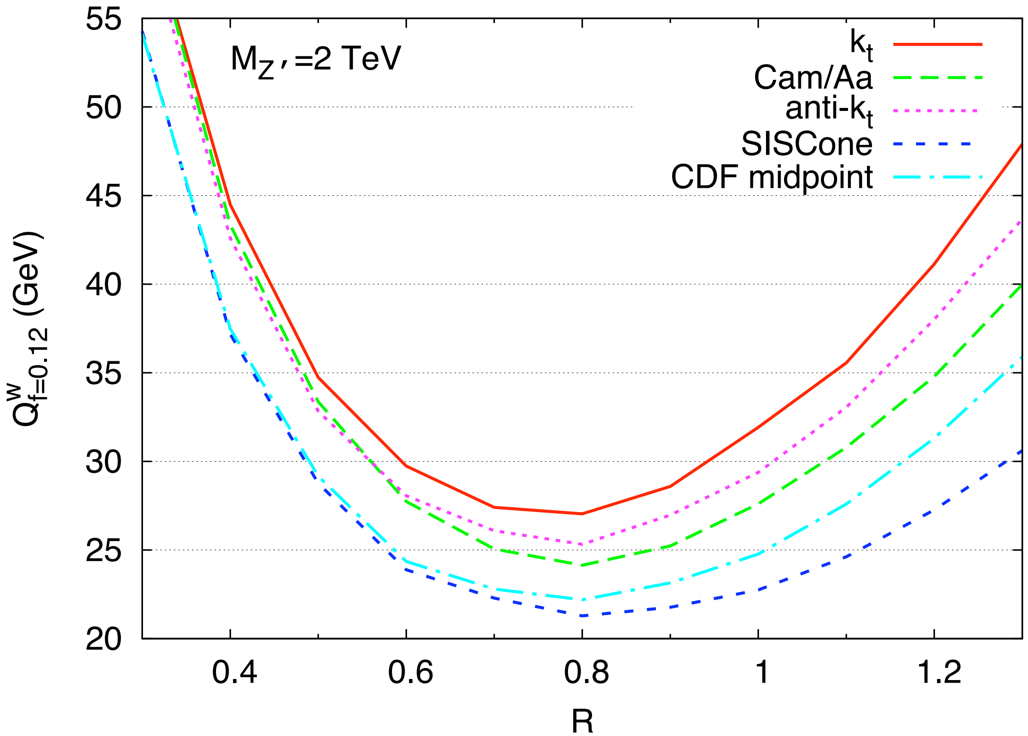
$$Q^{f}_{w=x\sqrt{M}}(R) \equiv \left(\frac{\text{Max }\#\text{ reconstructed massive objects in window of width } w = x\sqrt{M}}{\text{Total }\#\text{ generated massive objects}}\right)^{-1}$$

Z’ → qqbar

M = 100 GeV

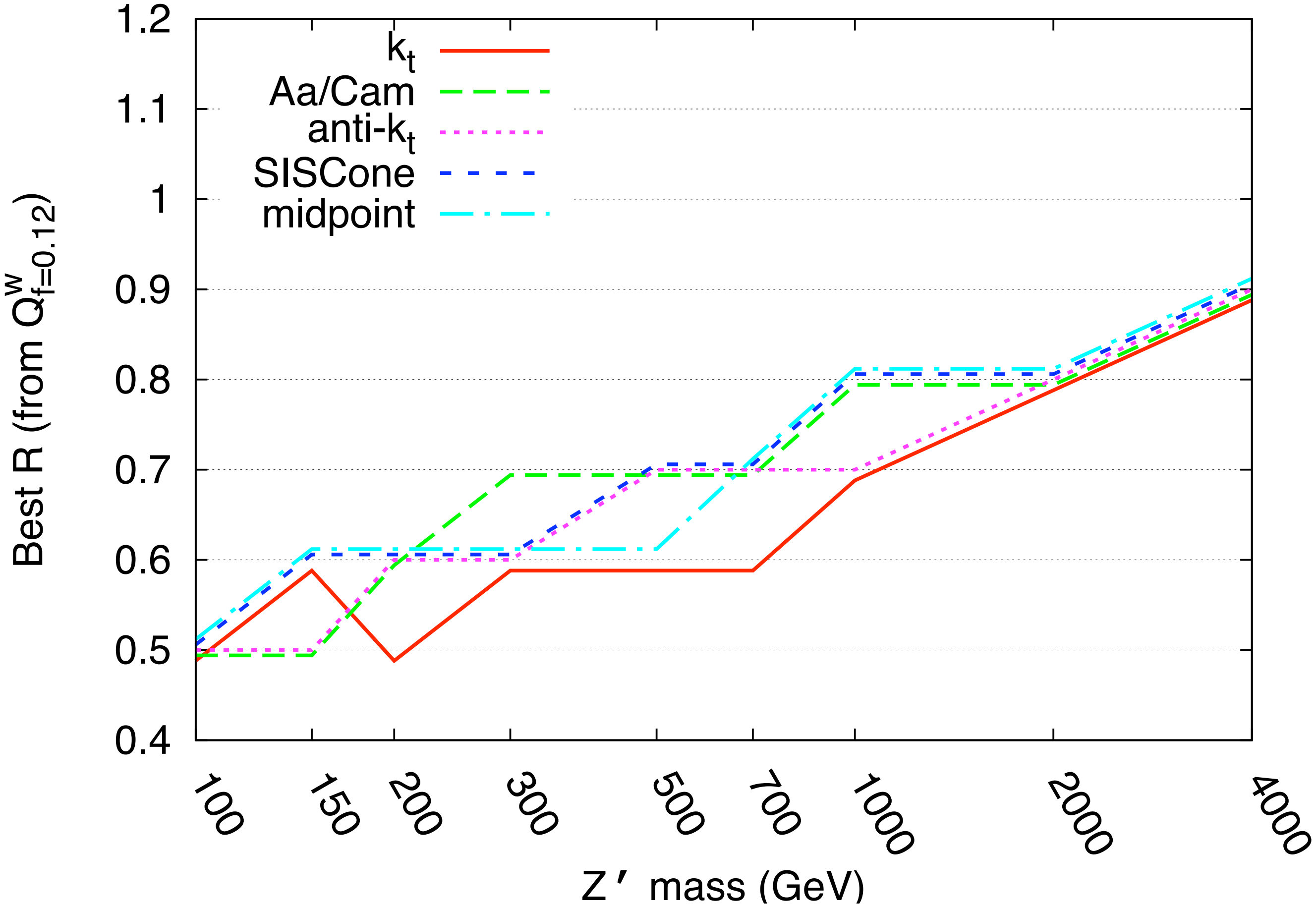


M = 2 TeV



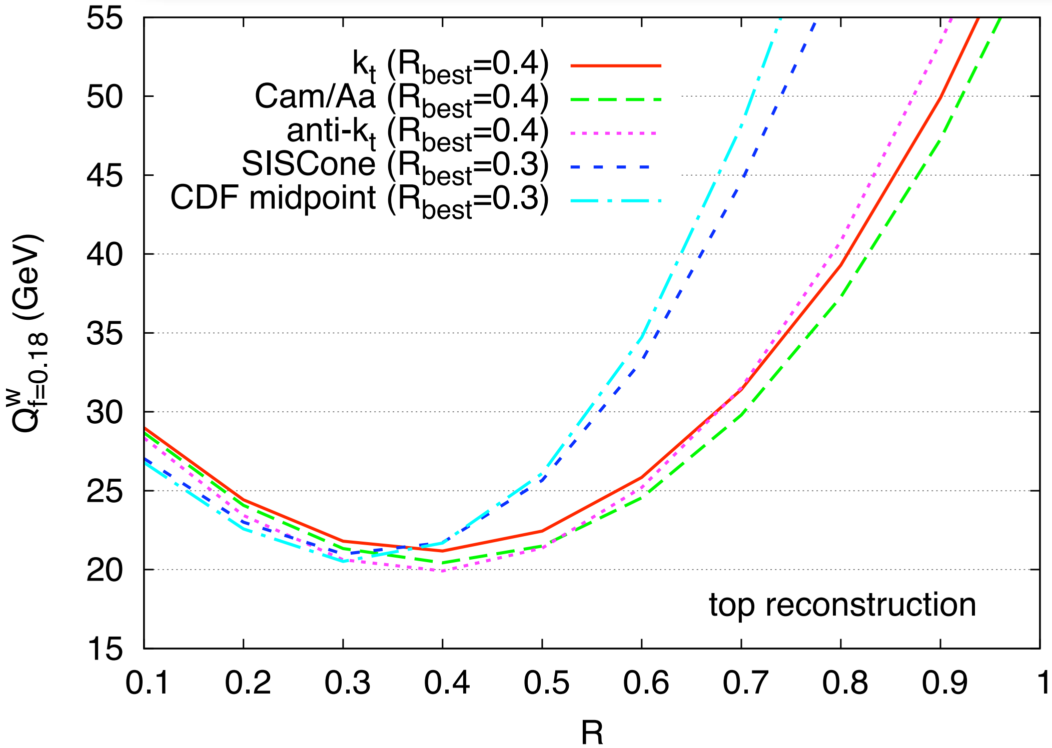
- The value of R which optimizes the figure of merit is quite different
- Some algorithms are better than others

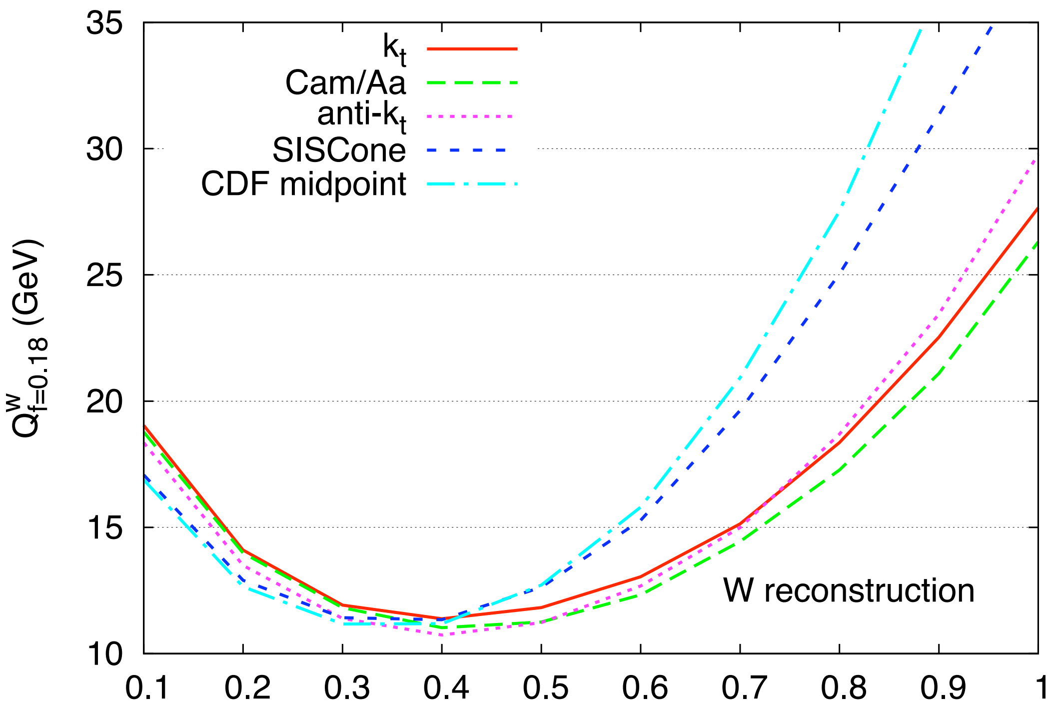
Z' → qqbar



Best R as a function of the Z' mass

ttbar





A non-optimal choice (e.g SISCone with R=0.5 rather than 0.3) corresponds to a figure of merit higher by 20%

In order to reach the same discriminating power, the luminosity will have to be higher by the same amount

Conclusions

- Jet algorithms, both the IRC unsafe and the safe ones, are well understood

- For every unsafe algorithm, there exists a safe replacement with similar characteristics

- Flexibility given by different R and different algorithms allows one to improve many analyses