HF jets analysis

08.09.2020 ALICE@IFJ meeting

Sebastian Bysiak

Outline

1. Kinematic variables
2. Selection of ranges
3. Detector efficiency correction

Kinematic variables

- jet p_T
- splittings:

  E_{rad} – emitting particle

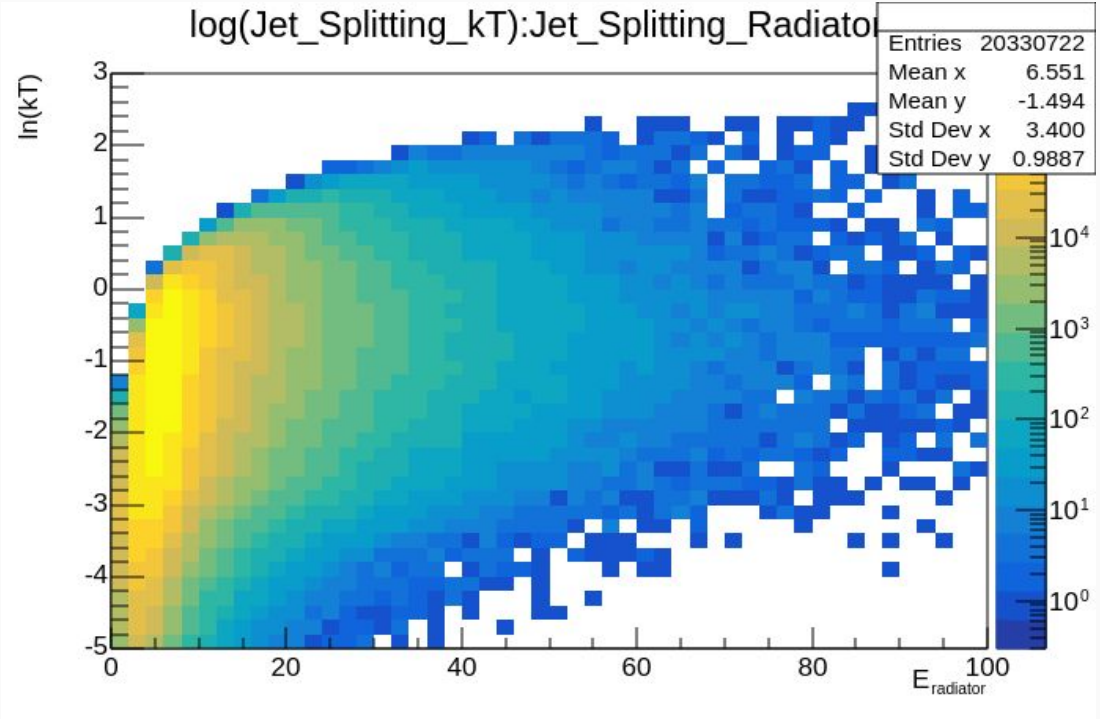
  θ – emission angle

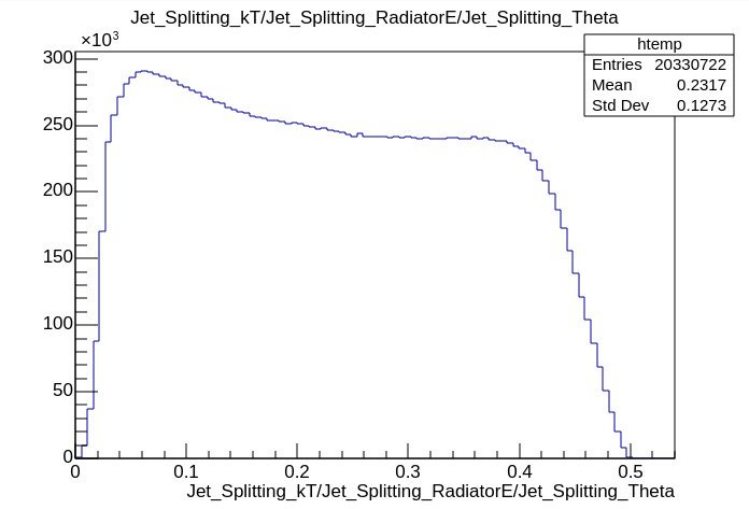
  kT – momentum of the emission transverse to emitter direction

  z – fraction of emitter momentum carried by emission

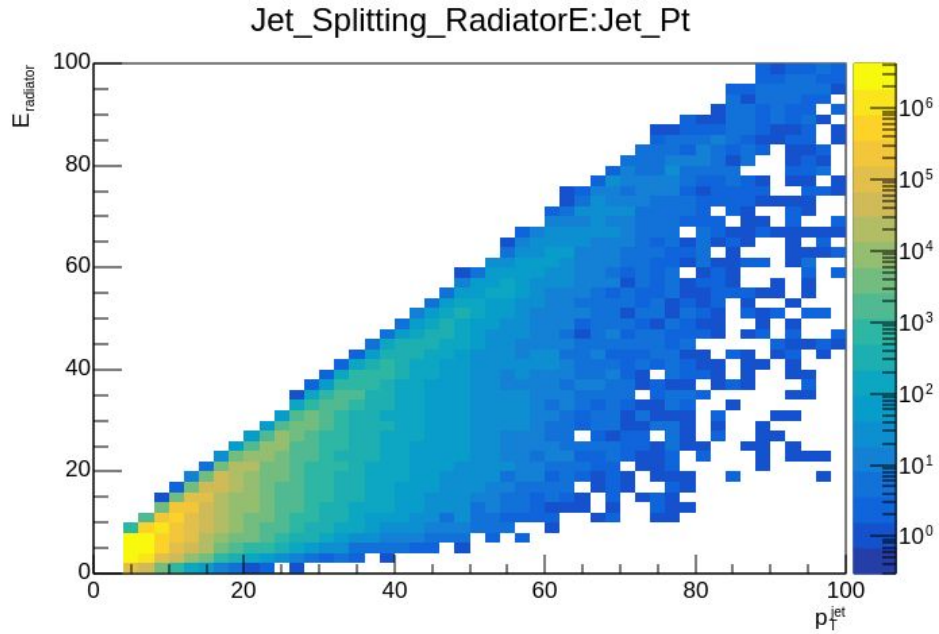
  $$kT = E_{rad} * z * \theta$$

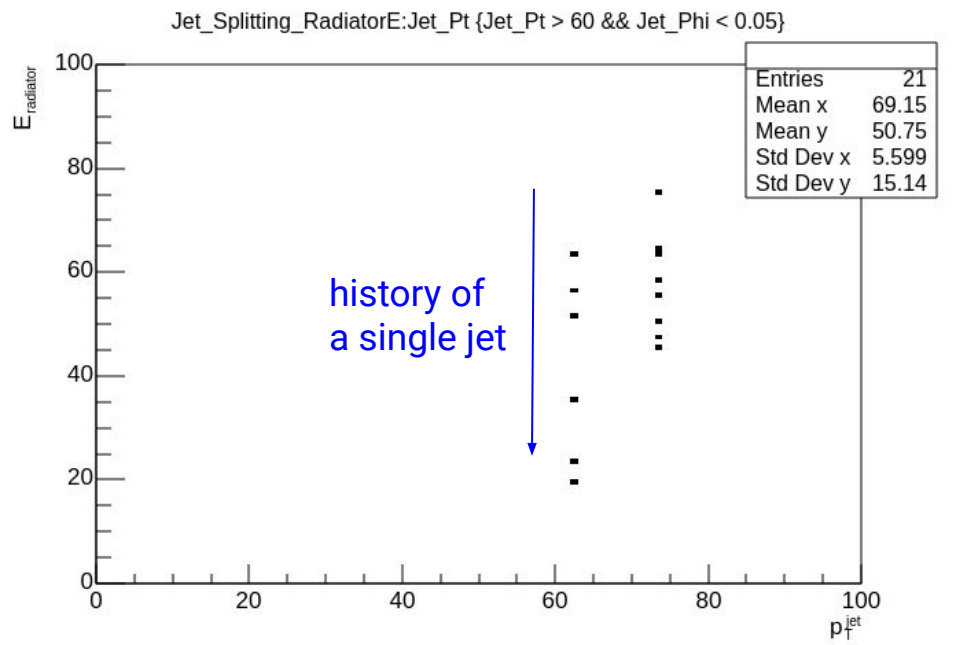
kT vs E_{rad} and z



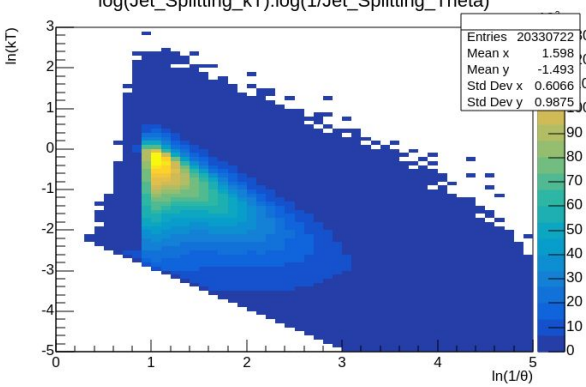


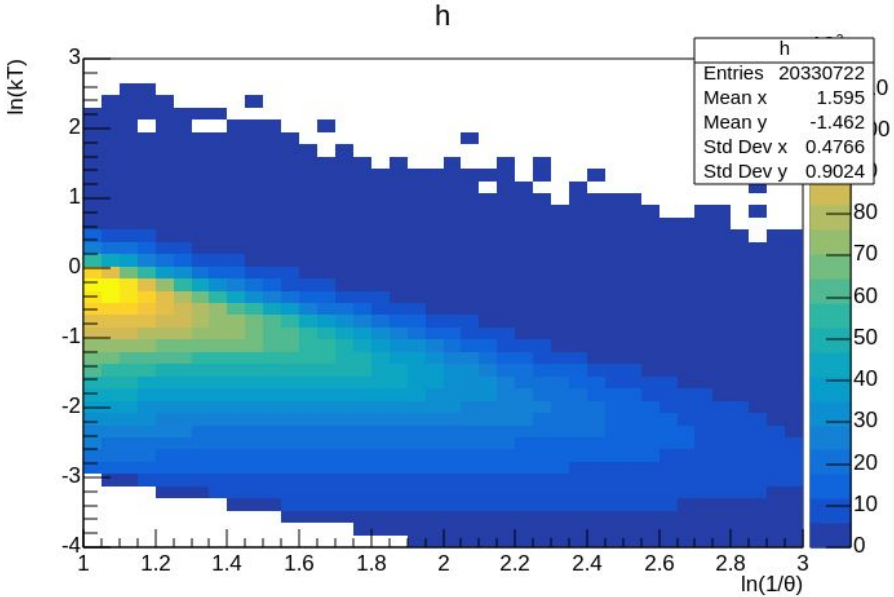
E_{rad} vs p_T

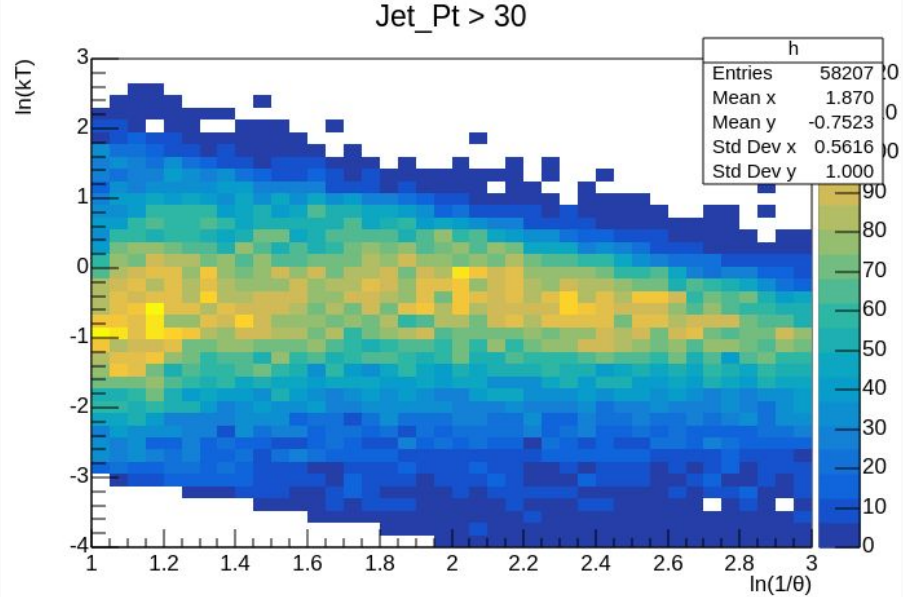

Jet_Splitting_RadiatorE:Jet_Pt
E_radiator
p_T^jet
0
20
40
60
80
100
10^0
10^1
10^2
10^3
10^4
10^5
10^6


Jet_Splitting_RadiatorE:Jet_Pt {Jet_Pt > 60 && Jet_Phi < 0.05}
Entries 21
Mean x 69.15
Mean y 50.75
Std Dev x 5.599
Std Dev y 15.14
history of
a single jet
E_radiator
p_T^jet
0
20
40
60
80
100

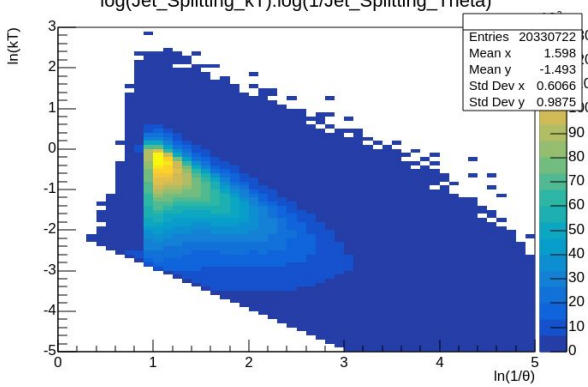
kT vs θ – “Lund Plane”

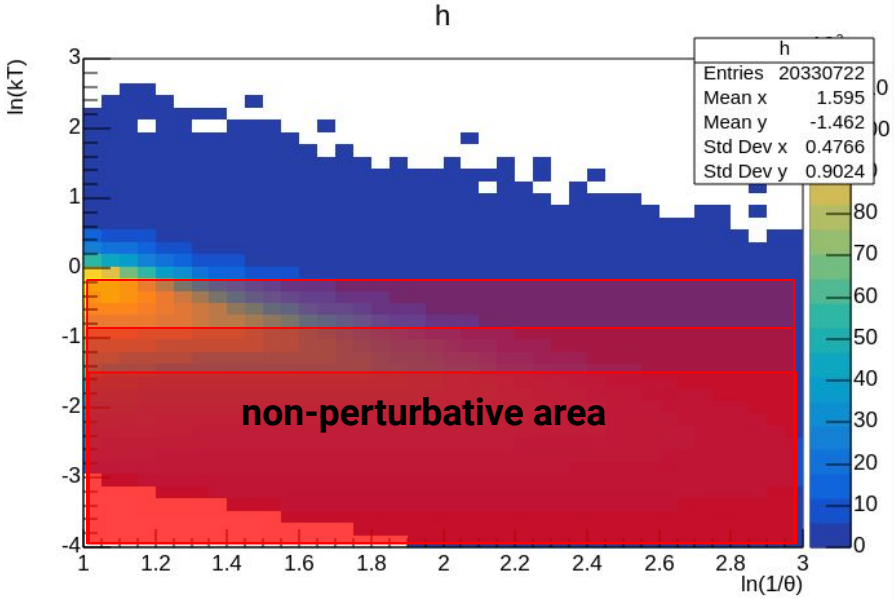


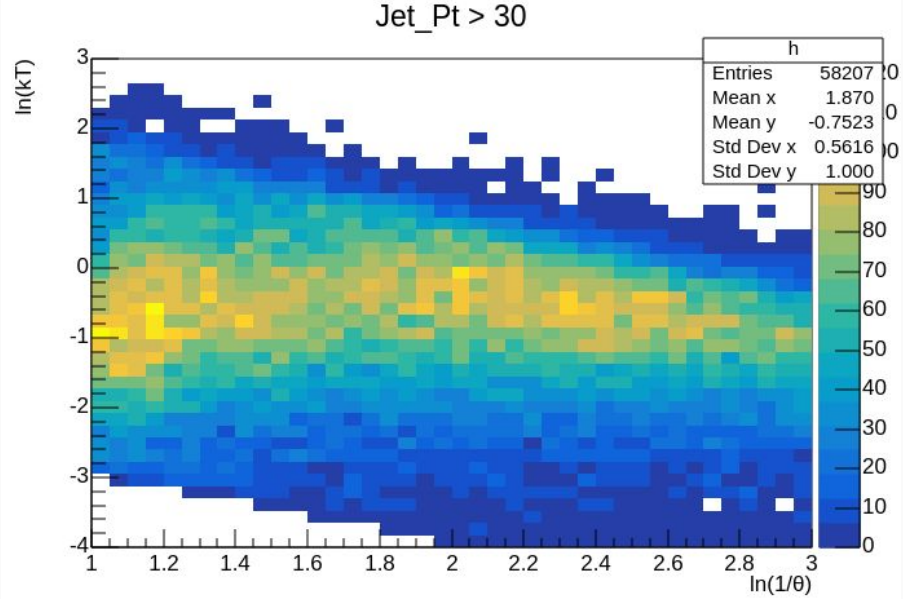




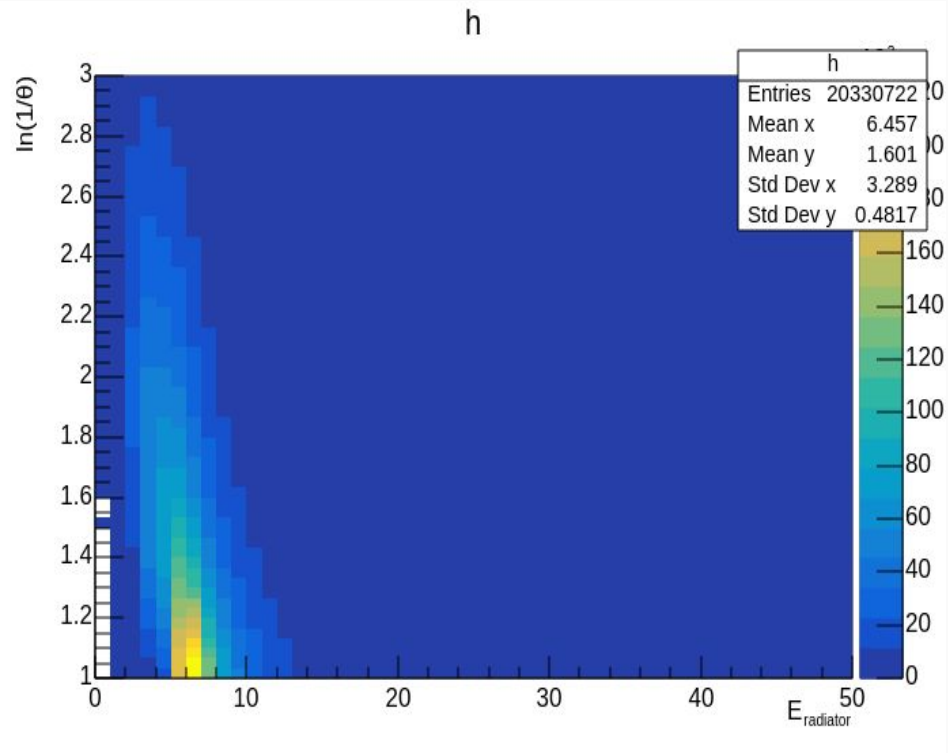
kT vs θ – “Lund Plane”

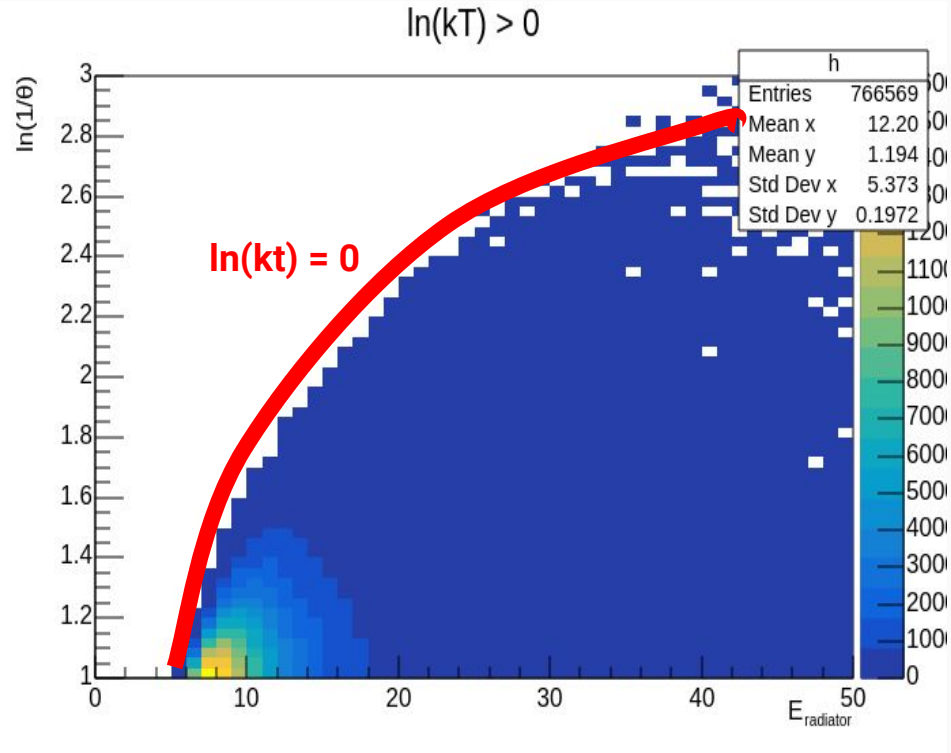






θ vs E_{rad}

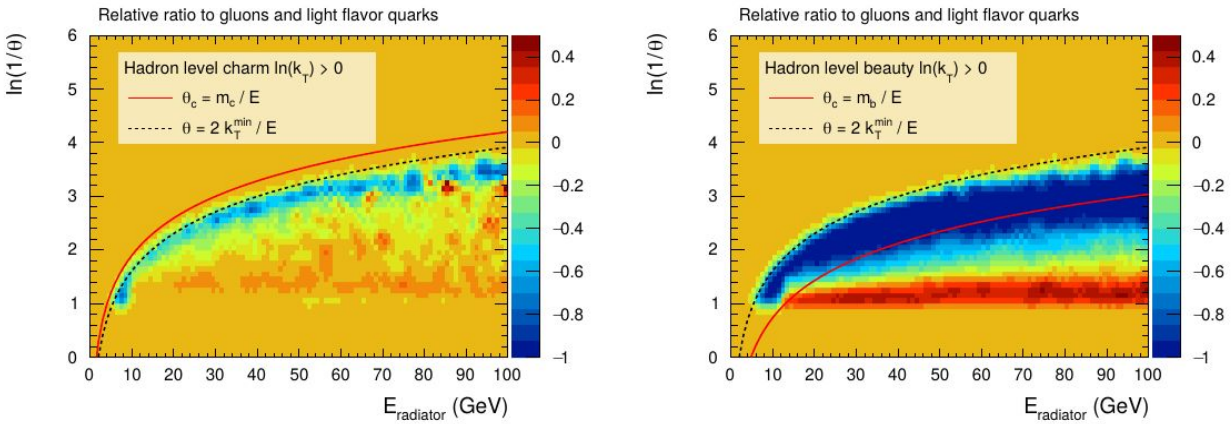

h
Entries 20330722
Mean x 6.457
Mean y 1.601
Std Dev x 3.289
Std Dev y 0.4817
ln(1/θ)
E radiator


ln(kT) > 0
h
Entries 766569
Mean x 12.20
Mean y 1.194
Std Dev x 5.373
Std Dev y 0.1972
ln(kt) = 0
ln(1/θ)
E radiator

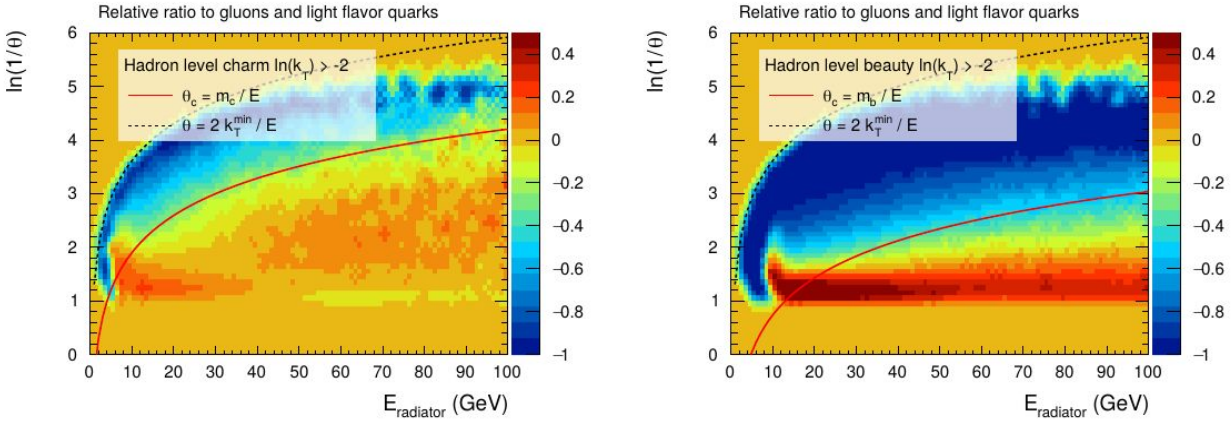
- **Smaller E_{rad} does not implies stronger effect**

- $1 < \ln(1/\theta) < 3$
  is not optimal

- kT cuts:
  (½ , 1, 2) * Λ_{QCD} (=200 MeV)
  which corresponds to $\ln(k_T/\mathrm{GeV})$ = (-2.3, -1.6, -0.9) D0-tagged jets
  or
  $\ln(k_T/\mathrm{GeV})$ = (-2 , 0)
  ⅔ , 5 * Λ_{QCD} dead-cone w/ declustring



b) Hadron level c-jets (left) and b-jets (right).

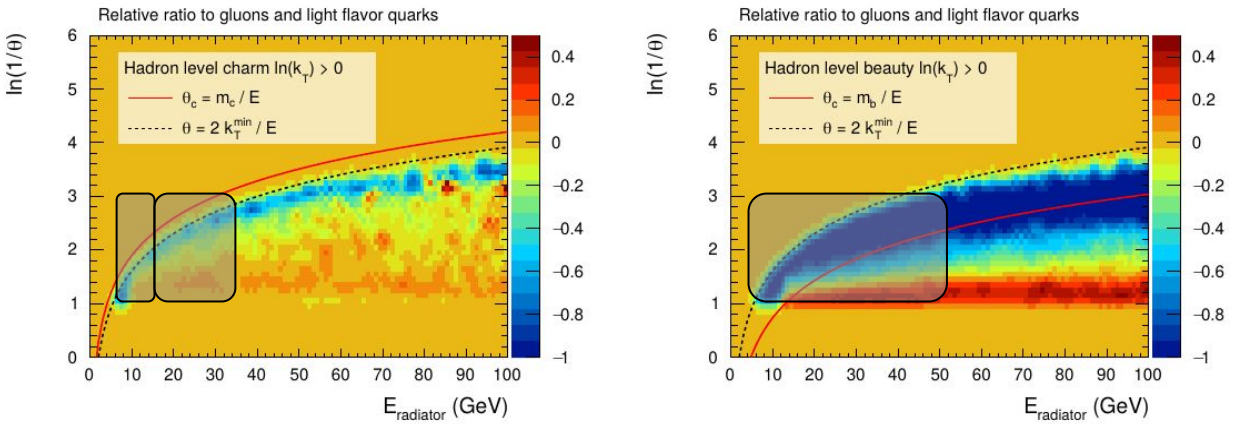


c) Hadron level with a relaxed cut on k_T - demonstration of the impact of non-perturbative effects

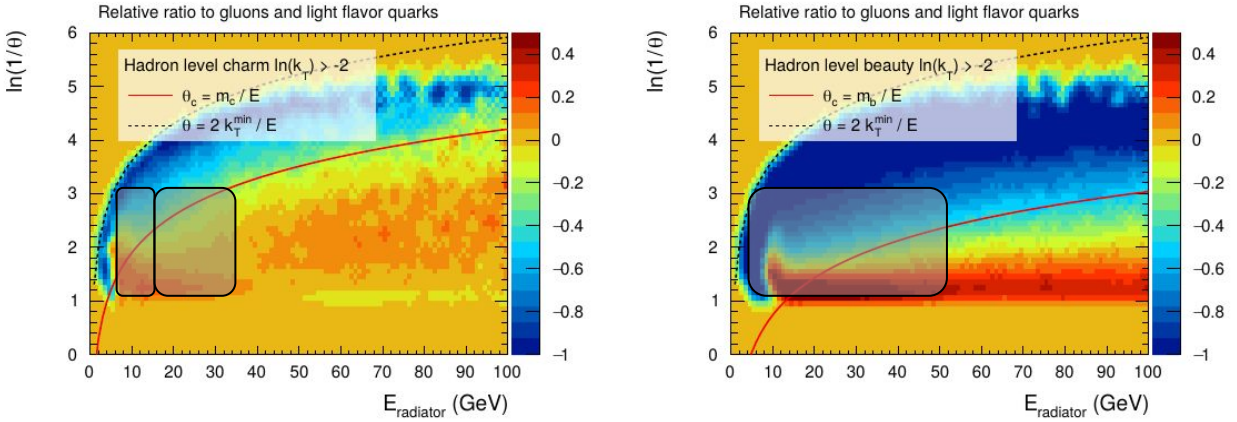
- **Smaller E_{rad} does not implies stronger effect**

- $1 < \ln(1/\theta) < 3$ is not optimal

- kT cuts:
  (½ , 1, 2) * Λ_{QCD} (=200 MeV)
  which corresponds to $\ln(k_T/\mathrm{GeV})$ = (-2.3, -1.6, -0.9) D0-tagged jets
  or
  $\ln(k_T/\mathrm{GeV})$ = (-2 , 0)
  ⅔ , 5 * Λ_{QCD} dead-cone w/ declustring



b) Hadron level c-jets (left) and b-jets (right).



c) Hadron level with a relaxed cut on k_T - demonstration of the impact of non-perturbative effects

Detector efficiency correction

How to select particles findable in the final state from `TreeK` in the `Kinematics.root`?

```
df.head(10)
```

executed in 781ms, finished 09:47:22 2020-09-08

	fUniqueID	fBits	fLineColor	fLineStyle	fLineWidth	fPdgCode	fStatusCode	fMother[2][0]	fMother[2][1]	fDaughter[2][0]	...	fPy	fPz	fE	fVx	fVy	fVz	fVt	fPolarTheta
entry																			
0	5	33570823	1	1	1	-11	0	413	-1	422	...	-0.006435	-0.008449	0.014375	-126.314026	-76.794815	-99.857407	5.903354e-09	-99.0
1	5	33570823	1	1	1	11	0	413	-1	416	...	-0.007330	-0.010033	0.017904	-126.314026	-76.794815	-99.857407	5.903354e-09	-99.0
2	9	33570823	1	1	1	11	0	415	-1	419	...	0.000426	-0.000074	0.000673	-119.343674	-94.417297	-162.076736	9.452287e-09	-99.0

```
df.query('fStatusCode == 1 and fPdgCode != 22').query('fPt > 0.15')
```

yields too much (~50-100 particles per event in pp)

fUniqueID, fBits, fLineColor, fLineStyle, fLineWidth, fPdgCode, fStatusCode, fMother[2][0], fMother[2][1], fDaughter[2][0], fDaughter[2][1], fWeight, fCalcMass, fPx, fPy, fPz, fE, fVx, fVy, fVz, fVt, fPolarTheta, fPolarPhi, fPt