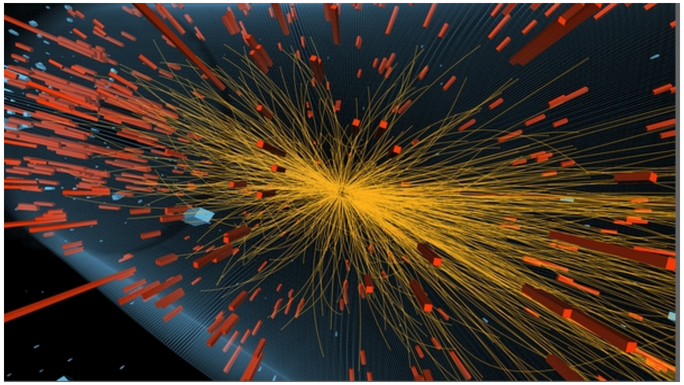
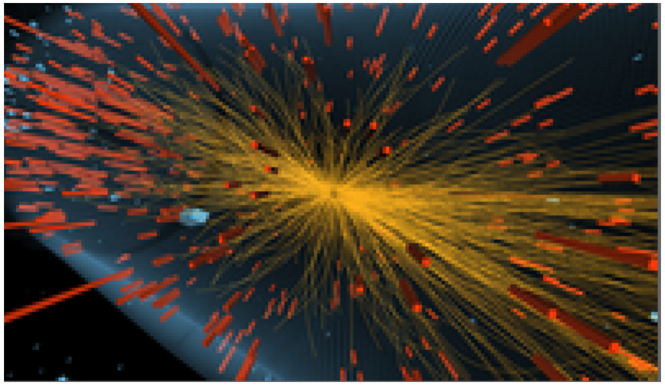
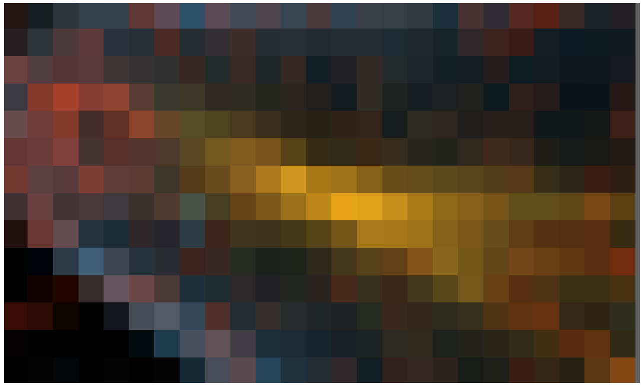
A further reduction in CMS event data for analysis: the NANOAOD format

Andrea Rizzi University and INFN Pisa

CHEP, Sofia, Bulgaria







MINI

NANO

Outline

- CMS Analysis flow from Run1 to today
- Challenges for current and future runs
- MiniAOD event content
- NanoAOD event content
- Status and prospects

Analysis flow

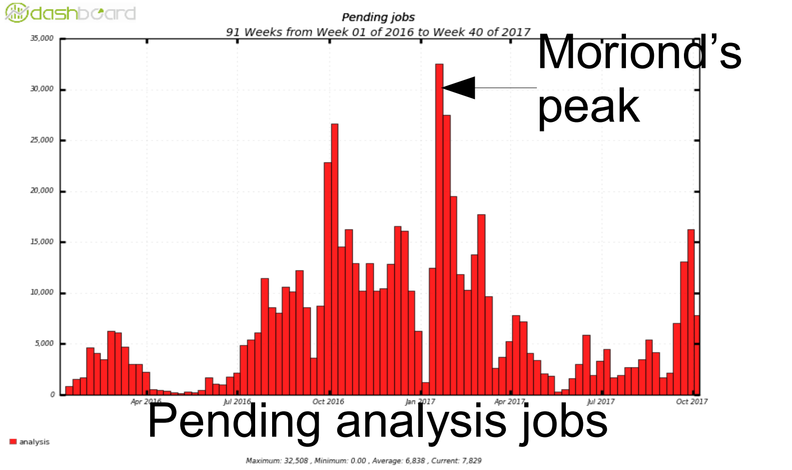
- Several data reduction/elaboration steps are present in CMS analysis workflow
- Organized **central** processing was originally planned only for the most expensive steps:
  - Data-taking + trigger + storage of RAW data
  - Prompt calibration
  - Event reconstruction and storage of analysis objects
- Downstream processing, **left to user implementations**, may include
  - Selection of relevant analysis objects
  - Further calibration or correction of measured/simulated quantities
  - Solving ambiguities/event interpretation (is this a jet or an electron?)
  - Reduction of per object information
  - Reduction of number of objects per event
  - Reduction of number of events

CMS Run1 to Run2

- Run1 model
  - Analysis groups/institutes privately process “AOD” and produce some “large ntuples” to be used by “many”
    - Small groups typically borrow large ntuples from larger groups that maintained their own ntuplizing code, and further reduce from there
    - Size per event of the large ntuples ~100kb/ev
  - A complex analysis would typically access 500 M events MC and 500 M events Data per year
- Run2 model
  - Introduction of “MiniAOD” as a common “large ntuple” format
    - Actually still in CMS EDM framework
    - Smaller and more rational than the typical ntuple (~40 kb/ev)
    - Originally foresee to satisfy 80% of use cases but today exceeding 95% coverage
  - A complex analysis typically access ~2B events per year

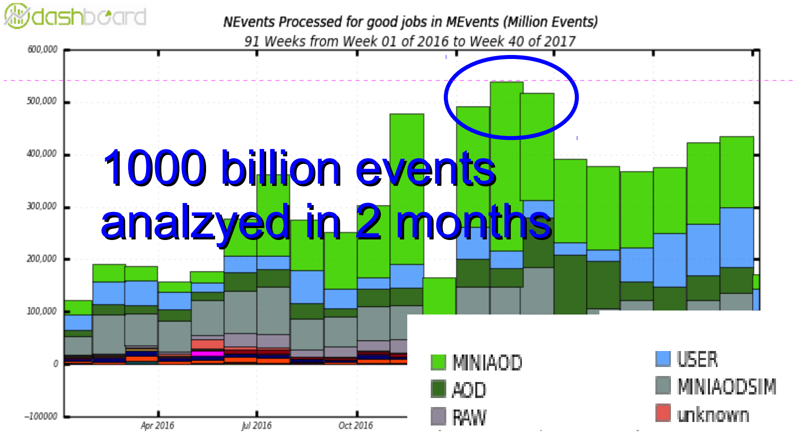
Run2 experience and Run3

- In 2017 we noticed that analysis queues in hot periods were “full”
- We loop over our ~10B events about 100 times in a month
  - Different groups of course, for different analyses
  - Were all 100 loops doing completely different things?
- Total number of events
  - Higher in Run1 → Run2 because of trigger rates (and will keep increasing with Run3, Run4, etc..)
  - Rate “must” increase because the EWK scale is still at 100GeV and the lumi grows… we simply cannot “cut harder” (or there is no point for more lumi in many analyses)

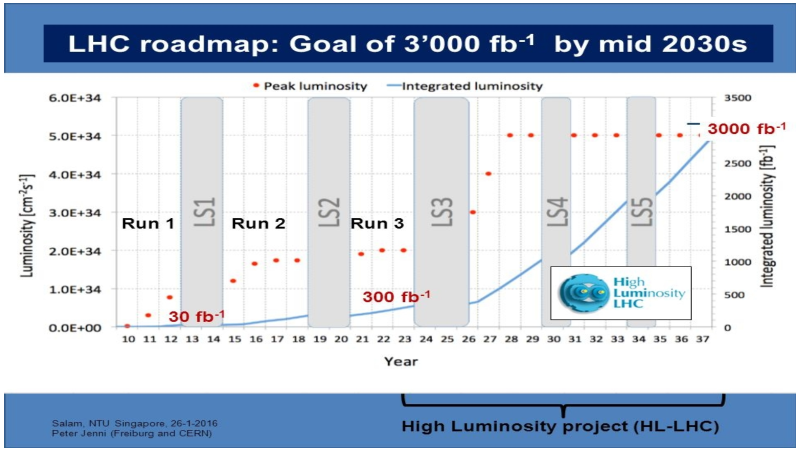


Moriond’s peak

Pending analysis jobs



1000 billion events analzyed in 2 months


LHC roadmap: Goal of 3’000 fb^{-1} by mid 2030s

Salam, NTU Singapore, 26-1-2016
Peter Jenni (Freiburg and CERN)

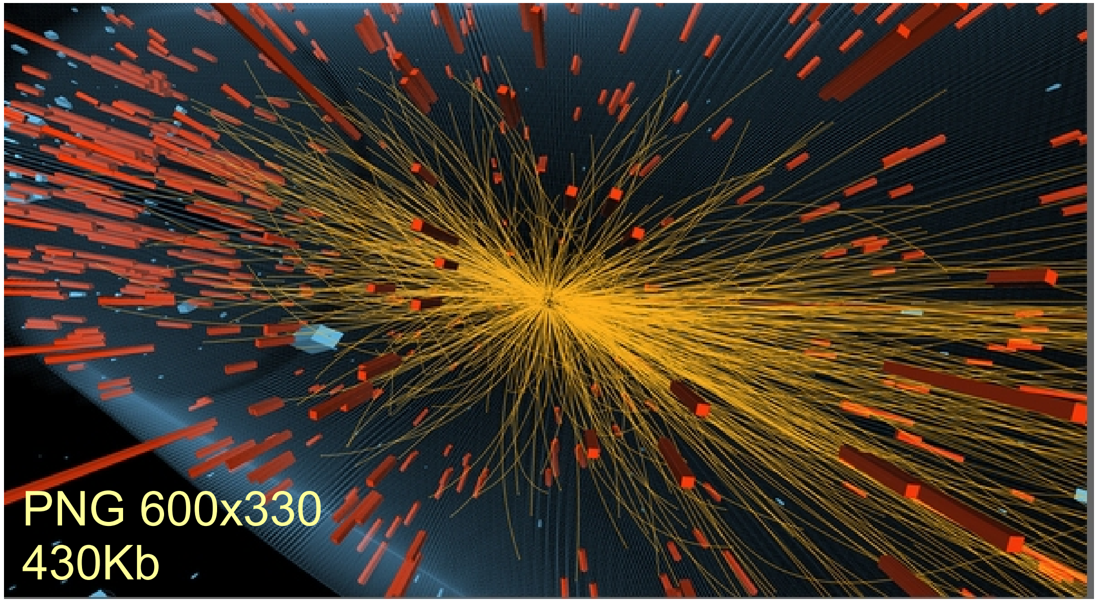
High Luminosity project (HL-LHC)

Further review of analysis model

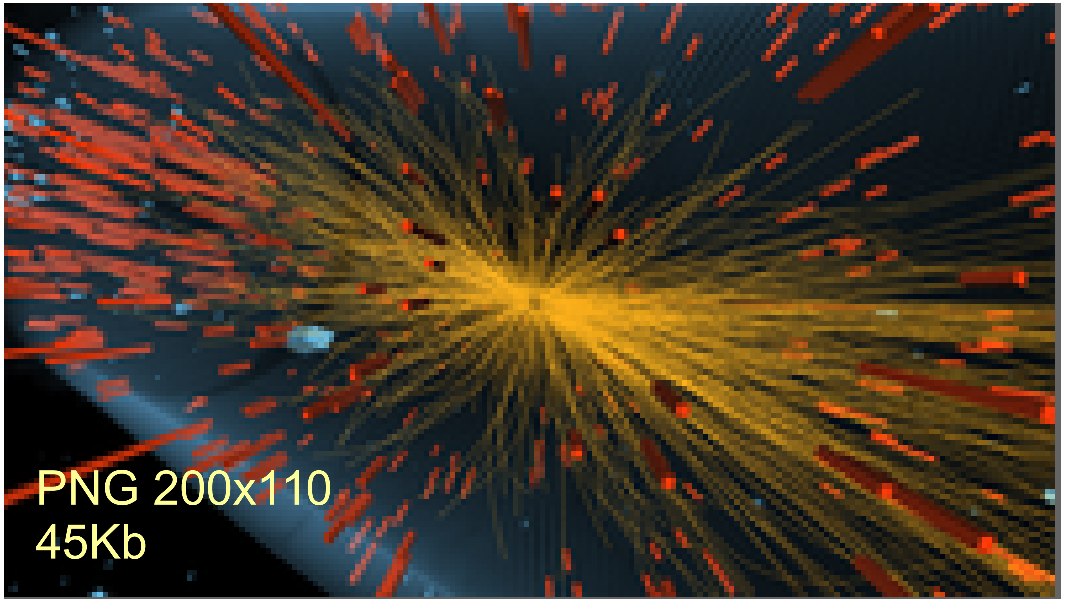
- Datasets size can be reduced vertically/horizontally
  - Reduce event size (RECO->AOD->MiniAOD->Ntuples)
  - Reduce number of events (Trigger, Skims, analysis selection)
- Can we do central low efficiency skim for each analysis with arbitrary event content?
  - Not really, if we could select tighter, most likely we would do it at HLT
  - Historically skims had very little success in CMS (only analyses that need very rich event content forced to do it)
- Is further event size reduction possible with no skimming ?
  - Many groups have **aggressive reduction** steps in the flow and while those steps are implemented for specific goals they are often **reused** by other groups doing something different (aka “can we use your ntuples?”)
  - **Physics object** information is often **standardized with “recipes”** shared across the groups, reimplemented in each analysis framework but effectively doing the same task with un-reviewd, emergency mode written, cut&pasted code
- Hint that **there is room for a single common ntuple serving a large fraction** of use cases
  - How big would such ntuple be?

Next Step: *nano*AOD

- It makes a difference if we reduce event size by another order of magnitude:
  - AOD: 450kb/ev
  - MINIAOD: 45kb/ev
  - NANOAOD (target): 4kb/ev



PNG 600x330
430Kb

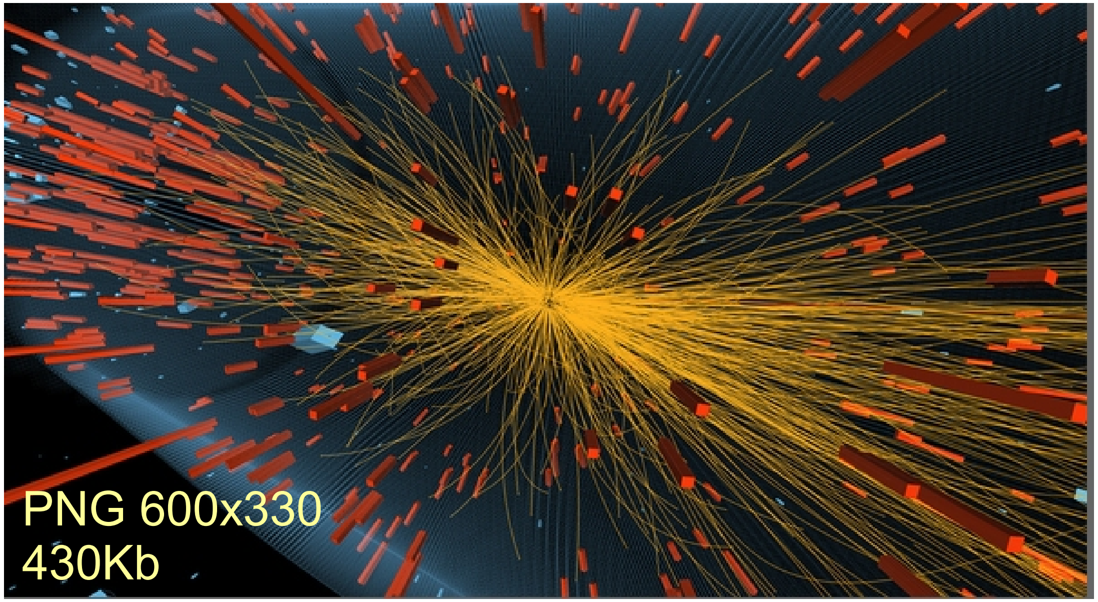


PNG 200x110
45Kb

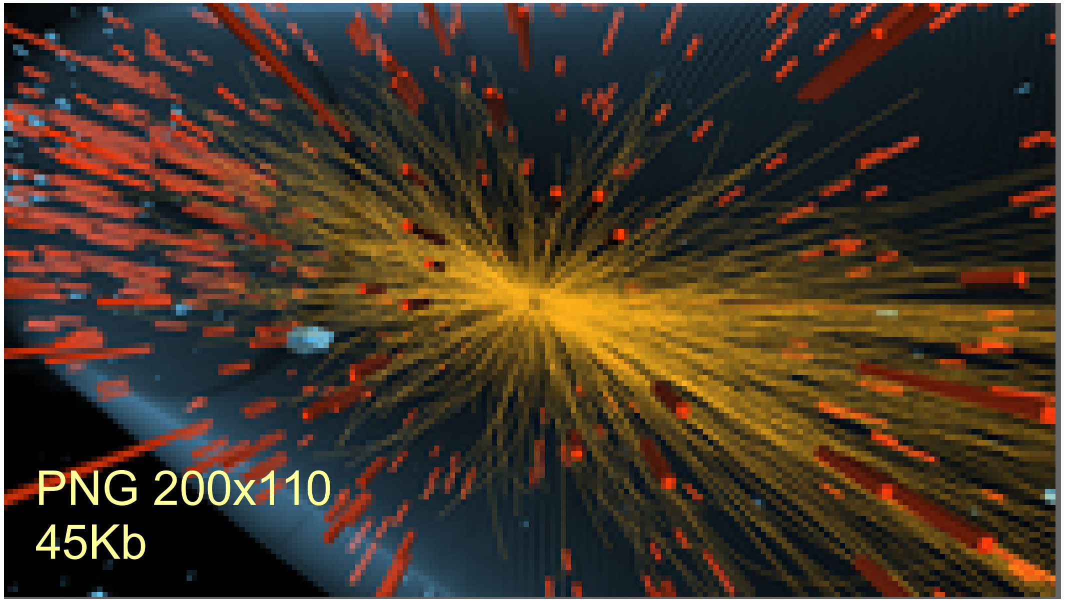
PNG 25x14
1.2Kb (~300bytes header)

Next Step: *nano*AOD

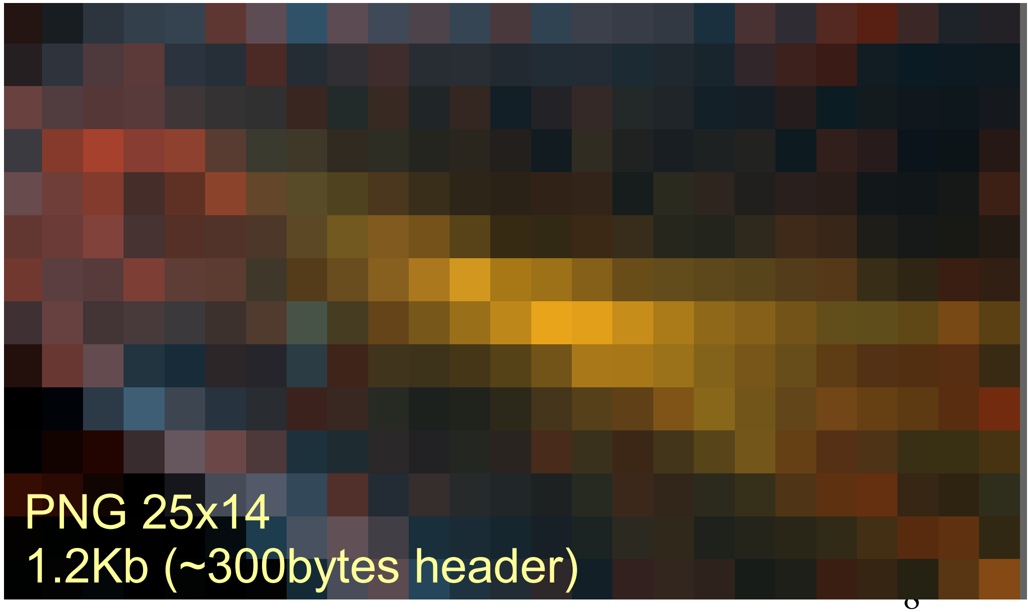
- It makes a difference if we reduce event size by another order of magnitude:
  - AOD: 450kb/ev
  - MINIAOD: 45kb/ev
  - NANOAOD (target): ~~4kb/ev~~ 1kb/ev
- First version:
  - 0.8Kb on data, 1.5Kb/ev MC



PNG 600x330
430Kb



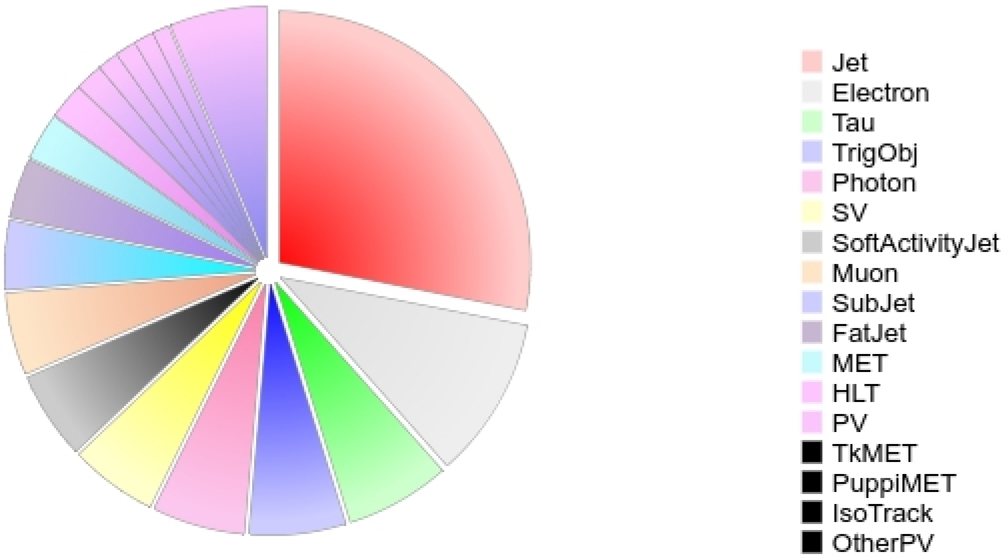
PNG 200x110
45Kb

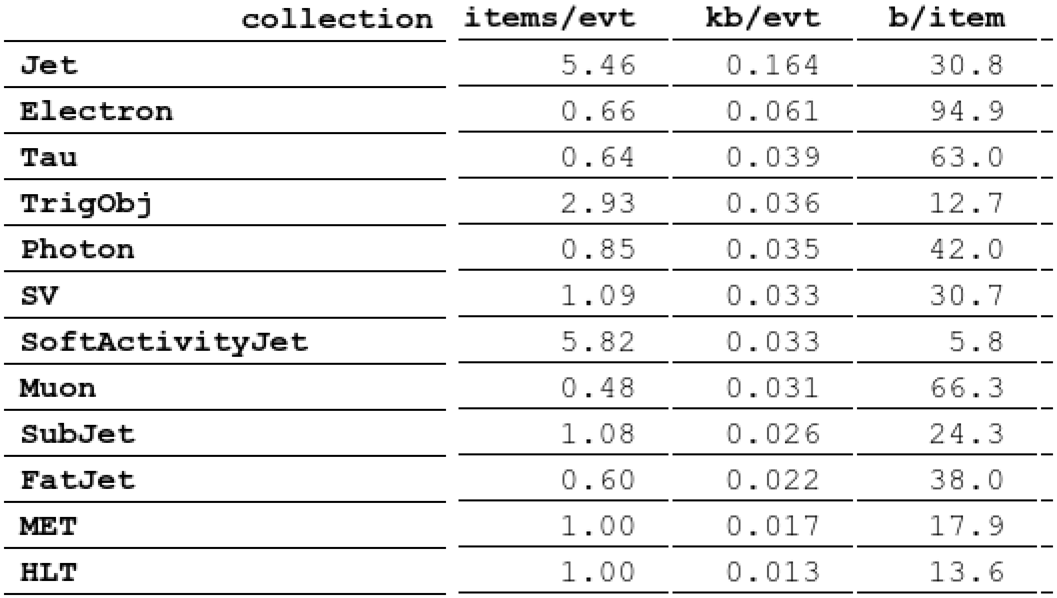


PNG 25x14
1.2Kb (~300bytes header)

Current prototype size/content

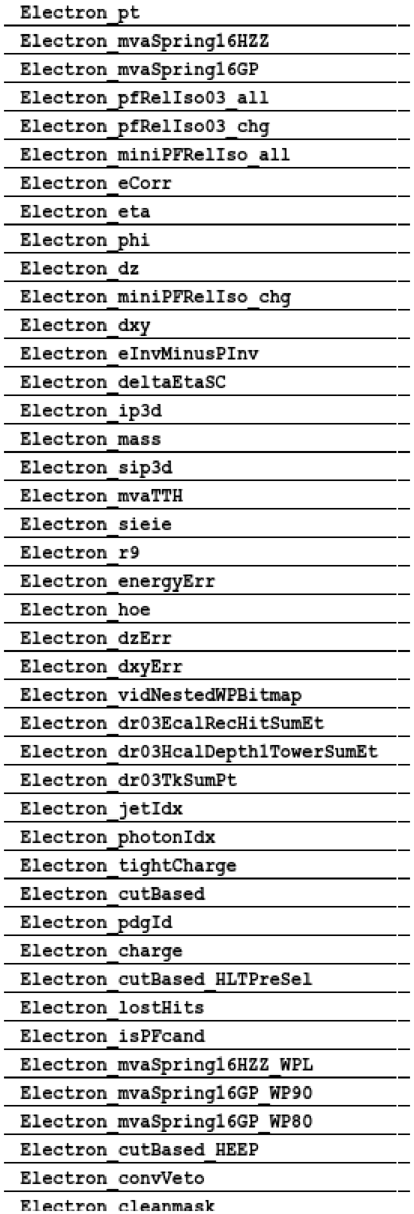
t (8.379 Mb, 10000 events, 0.86 kb/event)





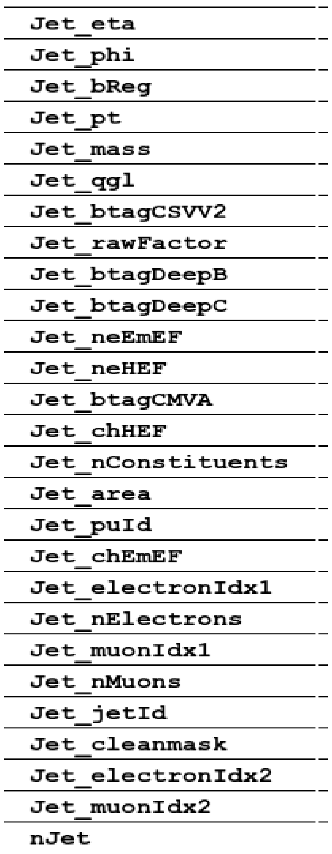
collection	items/evt	kb/evt	b/item
Jet	5.46	0.164	30.8
Electron	0.66	0.061	94.9
Tau	0.64	0.039	63.0
TrigObj	2.93	0.036	12.7
Photon	0.85	0.035	42.0
SV	1.09	0.033	30.7
SoftActivityJet	5.82	0.033	5.8
Muon	0.48	0.031	66.3
SubJet	1.08	0.026	24.3
FatJet	0.60	0.022	38.0
MET	1.00	0.017	17.9
HLT	1.00	0.013	13.6

Electron



Electron_pt
Electron_mvaSpring16HZZ
Electron_mvaSpring16GP
Electron_pfRelIso03_all
Electron_pfRelIso03_chg
Electron_miniPFRelIso_all
Electron_eCorr
Electron_eta
Electron_phi
Electron_dz
Electron_miniPFRelIso_chg
Electron_dxy
Electron_eInvMinusPInv
Electron_deltaEtaSC
Electron_ip3d
Electron_mass
Electron_sip3d
Electron_mvaTTH
Electron_sieie
Electron_r9
Electron_energyErr
Electron_hoe
Electron_dzErr
Electron_dxyErr
Electron_vidNestedWPBitmap
Electron_dr03EcalRecHitSumEt
Electron_dr03HcalDepth1TowerSumEt
Electron_dr03TkSumPt
Electron_jetIdx
Electron_photonIdx
Electron_tightCharge
Electron_cutBased
Electron_pdgId
Electron_charge
Electron_cutBased_HLTPreSel
Electron_lostHits
Electron_isPFcand
Electron_mvaSpring16HZZ_WPL
Electron_mvaSpring16GP_WP90
Electron_mvaSpring16GP_WP80
Electron_cutBased_HEEP
Electron_convVeto
Electron_cleanmask

Jet



Jet_eta
Jet_phi
Jet_bReg
Jet_pt
Jet_mass
Jet_qgl
Jet_btagCSVV2
Jet_rawFactor
Jet_btagDeepB
Jet_btagDeepC
Jet_neEmEF
Jet_neHEF
Jet_btagCMVA
Jet_chHEF
Jet_nConstituents
Jet_area
Jet_puId
Jet_chEmEF
Jet_electronIdx1
Jet_nElectrons
Jet_muonIdx1
Jet_nMuons
Jet_jetId
Jet_cleanmask
Jet_electronIdx2
Jet_muonIdx2
nJet

How to fit everything in 1kb/ev ?

- No tracks / individual particle candidate
- No detector details for objects (no calo cells, rechits, etc..)
- Prefer **precomputed** obj IDs to "variables needed for ID"
- Complex event quantities should be stored rather than providing the needed inputs (even if used by few analyses)
- Limit information in collections with many entries (e.g. jets)
- **systematic variations not persistently stored** (Jet energy corrections, b-tagging data/MC scale factors, etc...)
  - They can be computed later with a simple function

    f_corr(pt,eta, and few other variables we can store)
- Do not store **32bit** precision floats (1e-7 relative precision) because **we do not measure with this precision**!

NanoAOD Format

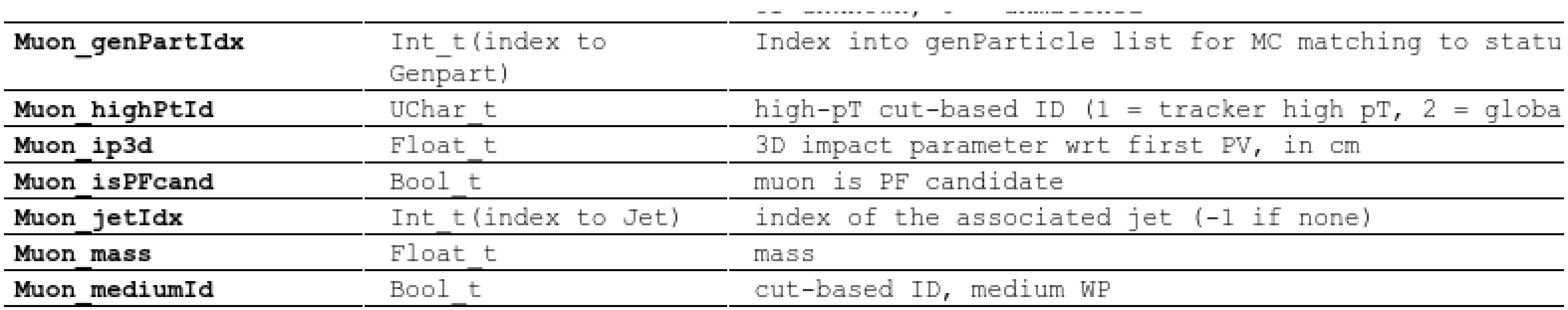
- NANOAOD format is a bare root ntuple
  - Typical reasonable ntuple format (Muon_pt[nMuons], Muon_eta[nMuons] etc...)
  - Simple to export to modern machine learning and non-HEP analysis frameworks
- Even if it is a bare root ntuple, it has some additional goodies
  - Contains multiple trees to store non Events information
  - has “provenance” information (I.e config used to process up to this point)
  - It is compatible with most CMS “EDM tools”
  - ...and especially, it can be produced by central production

Cross ~~cleaning~~ linking

- MiniAOD collections are not cross cleaned
  - i.e. leptons appears as jets, both a photon and an electron can originate from a single ECAL cluster, almost all jets are tau candidates etc...
- Cross collection cleaning is a typical example of “analysis dependent choice”
  - We do not want to enter “analysis freedom”
  - So NanoAOD are not cross clean.
- … but we cross link!
  - With ParticleFlow/GlobalEventDescription we have an obvious way to know what should be cleaned (are two objects sharing the input PFCandidates? Then your analysis should decide where the candidate belong...)
  - We save links (i.e. just indices) among physics objects in the final format

NanoAOD (more) features

- NanoAOD-Tools (useful to process nanoaod)
  - Fast and efficient skimming or friend-trees creation
  - JEC uncertainties, jet smearing, btag uncertainties
  - Lepton scale factors
  - Central location for any additional analysis “recipe”
- Auto generated documentation:



Muon_genPartIdx	Int_t(index to Genpart)	Index into genParticle list for MC matching to statu
Muon_highPtId	UChar_t	high-pT cut-based ID (1 = tracker high pT, 2 = globa
Muon_ip3d	Float_t	3D impact parameter wrt first PV, in cm
Muon_isPFcand	Bool_t	muon is PF candidate
Muon_jetIdx	Int_t(index to Jet)	index of the associated jet (-1 if none)
Muon_mass	Float_t	mass
Muon_mediumId	Bool_t	cut-based ID, medium WP

https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookNanoAOD

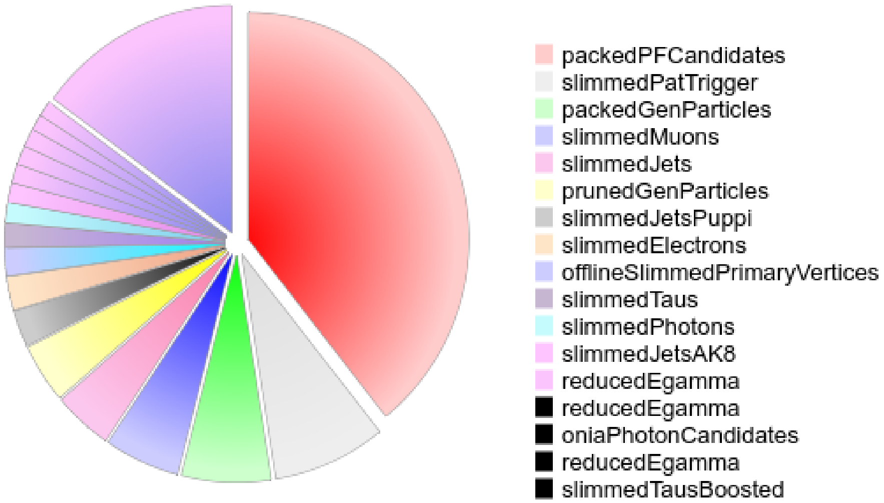
Conclusions

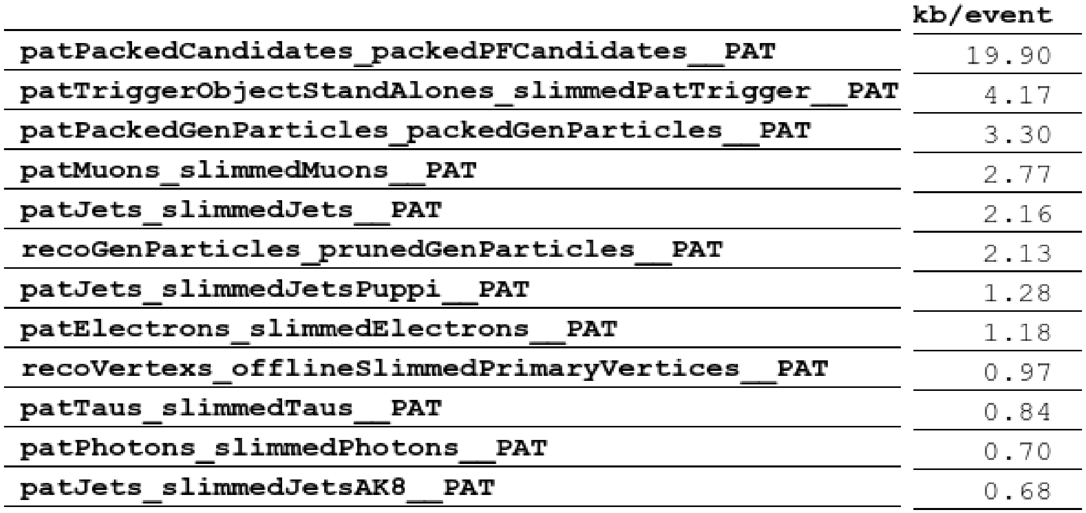
- We introduced MiniAOD as a new analysis format between Run1 and Run2 to rationalize resource usage
- The current analysis model could be problematic in coming LHC Run scenarios
- We now introduce a 1Kb-per-event format (NanoAOD) as a possible way forward
- Deployment of NanoAOD ongoing in CMS
  - Central production automatically creating NanoAOD
  - many analysis switching to the new 1kb/event format

- backup

MINIAOD

- MiniAOD was a first step to unify one of the “middle layers” (the ~50kb/ev steps that every analysis had) between AOD and final histos
  - It worked well and more analyses switch to using it
- MINIAOD content:
  - All single particles information (in some more or less compressed form)
    - Track + PFcandidate information unified
    - Details up to ECAL rechits or muon segments
  - Allow to recalculate all POG quantities that need fine tuning after the data has been taken
    - ID, isolation, energy corrections, pu rejection
    - All so called “POG recipes”
  - Allow some “special analysis needs”
    - Different jet clustering, ….





	kb/event
patPackedCandidates_packedPFCandidates__PAT	19.90
patTriggerObjectStandAlones_slimmedPatTrigger__PAT	4.17
patPackedGenParticles_packedGenParticles__PAT	3.30
patMuons_slimmedMuons__PAT	2.77
patJets_slimmedJets__PAT	2.16
recoGenParticles_prunedGenParticles__PAT	2.13
patJets_slimmedJetsPuppi__PAT	1.28
patElectrons_slimmedElectrons__PAT	1.18
recoVertexs_offlineSlimmedPrimaryVertices__PAT	0.97
patTaus_slimmedTaus__PAT	0.84
patPhotons_slimmedPhotons__PAT	0.70
patJets_slimmedJetsAK8__PAT	0.68