

# Overview of Common Analysis Tools

D0 Workshop

Michigan State University

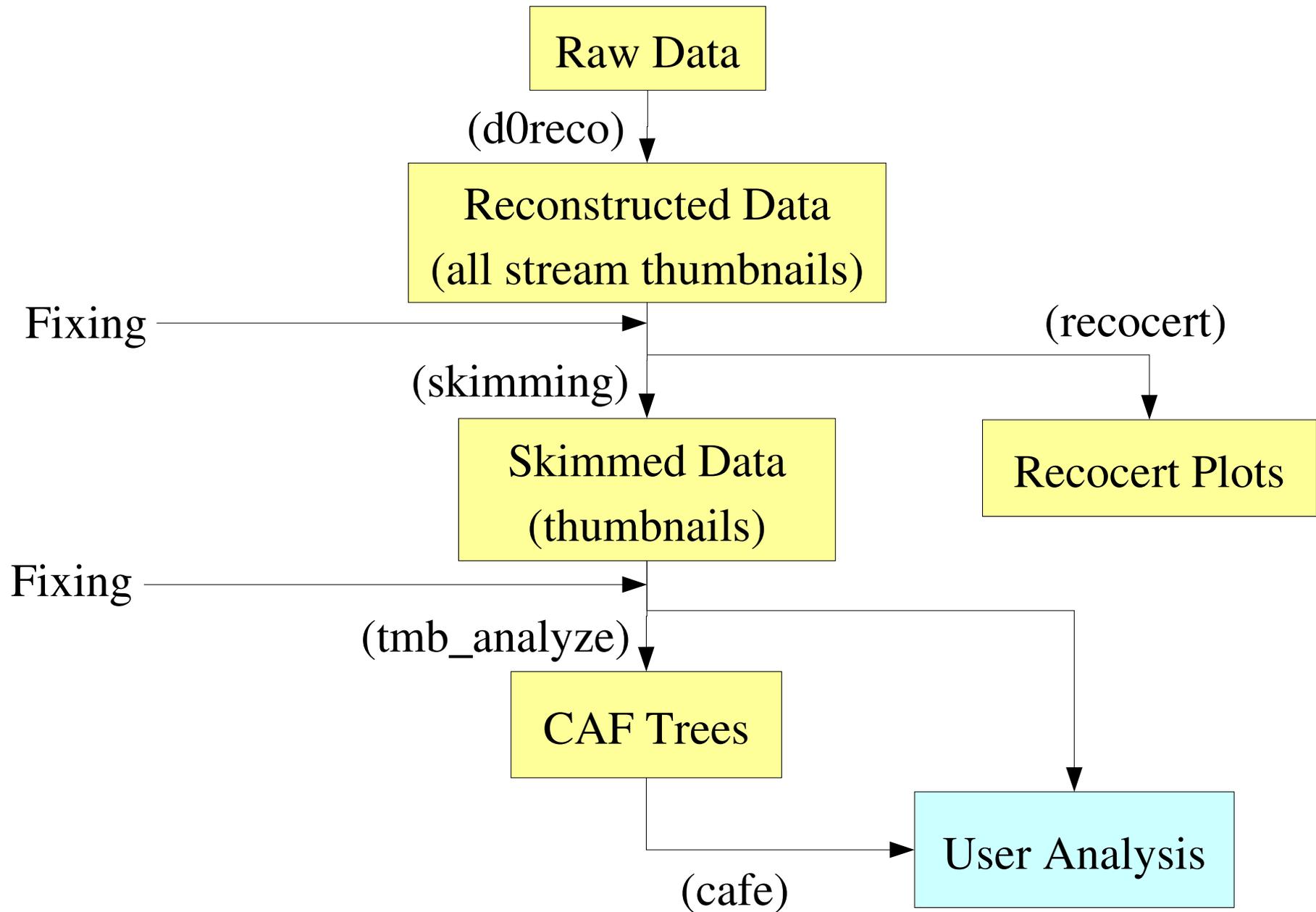
June 19, 2007

H. Greenlee

# Outline

- Standard Datasets.
- Framework (tmb) object id. and efficiency tools.
- CAF object id. and efficiency tools.
- Trigger tools.
- Data Quality tools.
- Luminosity tools.

# Data Processing Chain

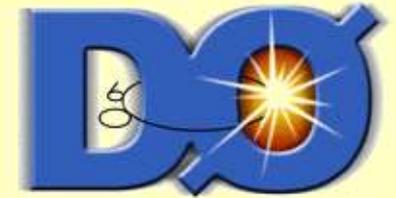


# Standard Datasets

- Input for most analyses are skimmed datasets (tmb or caf).
  - Standard skimmed datasets exist for run 2a and run 2b.
  - Tmb and caf datasets contain the same events, just different formats.
  - There are 14 physical skims in run 2a (13 in run 2b).
  - Logical skims (event flags) can be used at either tmb or caf level to quickly select a subset of physical skims.
    - Tmb level – framework package **FilterEventFlags**.
    - Caf level – processor **EventFlags**.
  - Refer to Common Sample Group web page for definitions of physical skims and event flags for run 2a and run 2b data.

# Standard Datasets (cont.)

- Run 2a (p17 d0reco).
  - Only one standard dataset, includes fixed and unfixed data.
- Run 2b (p20 d0reco).
  - Pass 1 (p20.04.01 d0reco).
    - First version of d0reco used with run 2b data.
    - Now mostly obsolete. Does not include any 2007 data.
  - Pass 2 (p20.07.01/p20.08.xx d0reco).
    - Latest and greatest run 2b data.
    - Frozen summer 2007 datasets are subset of this pass.
  - Pass 3 (fixed/p20.10.xx d0reco?).
    - Doesn't exist yet.



## Meetings:

Week A, Wednesday 10:30am to 12:30pm in the Hurricane Deck.

Transparencies are posted on the [D0 Agenda Server](#).

## Mailing List:

Send e-mail to the entire group [d0\\_csg@fnal.gov](mailto:d0_csg@fnal.gov) or browse the [archive](#)

## Conveners:

The current [conveners](#) are [Viatcheslav Sharyy](#) (x6830 in the Outback or 011-33-169086129 at Saclay) and [Herbert Greenlee](#) (x3618).

## News

- May 24, 2007: caf\_util interface updated to the new version of the jetcorr. See [new list of packages](#). More details about new JES can be found in [this](#) and [this](#) announces.
- May 8, 2007: Frozen run 2b pass 2 skimmed datasets for summer 2007 physics results have been defined. Refer to [P20 Pass 2 Data and MC](#) page for details.
- Feb. 28, 2007: Metadata for newly generated p17 and p20 mc caf trees no longer contains the "runType='monte carlo'" field. Predefined sam datasets will be defined to reflect this change. However, people who are querying mc caf files from sam directly (e.g. using "sam translate constraints") should modify their queries to drop the "runType" field.
- Feb. 23, 2007: new version of muon efficiencies available (muid\_eff v03-00-00). For more detail see [this web page](#).
- Feb. 2, 2007: Skimming and caf production of p17.09.00 data from the early part of Run IIa is finished.
- Jan. 30, 2007: Newly reconstructed data from the early part of Run IIa (runs 151817-166493) are being skimmed and caf

## Data Sets

[p20 Pass 2 Data and MC](#)

[p20 Pass 1 Data and MC](#)

[p17 Data and MC](#)

[p14 Pass 1](#)

[p14 Pass 2](#)

## Topics

[Common Analysis Format](#)

[Skimmed Data](#)

[D0Correct](#)

[Howtos](#)

[Raw Data Samples](#)

[D0Correct Requests](#)

[Comments to CSG Conveners](#)

# D0correct Framework Package

- Framework package **d0correct** should always be the first step in any analysis.
  - Included in `tmb_analyze` executable and `rcp` files.
  - User `tmb`-based analyses should also include **d0correct**.
- What's in **d0correct**.
  - Muon post-processing (**MuoCandidate**).
  - EM post-processing.
  - Jet id and jet energy scale correction.
  - Missing ET correction.
  - MTC (muon tracks in calorimeter).

# D0correct (cont).

- What is included in **d0correct** post-processing?
  - Standard quality calculation (loose/medium/tight) stored in object.
  - Object corrections.
  - MC smearing.
- What is not included in **d0correct** – efficiency corrections.
- Where are updated objects stored.
  - New chunk replaces old chunk of same type (**MuonParticleChunk**, **EMParticleChunk**, **MissingETChunk**)
  - New chunk of different type (**CorrJetChunk** supersedes **JetChunk**).
  - Entirely new chunk (**MuoCalTrackChunk**).

# Efficiency Calculation at TMB Level

- Tools for calculating efficiencies.
  - Cvs package **muo\_cert** – Executable and framework package for calculating high pT, isolated, muon-related efficiencies.
    - Results for standard selections stored in cvs package **muid\_eff**.
  - Cvs package **em\_cert** – Executable and framework package for calculating high pT, isolated, electron-based efficiencies.
    - Results for standard selections stored in cvs package **emid\_eff**.
  - Cvs package **photon\_id\_tools** – For photons.
  - You only need to run these packages if you are using nonstandard object ids, or if you are an expert/maintainer.
- There is no tool for applying precalculated efficiencies at tmb level.
  - Do this by hand at tmb level, or use caf level tools.

# Efficiency Calculation at CAF Level

- Tools for calculating efficiencies.
  - Cvs package **trigeff\_cafe** – Contains processors for calculating jet trigger and id efficiency.
    - Results for standard selections stored in **jetid\_eff**.

# Tools for Selecting Events at Caf Level

- **Select** processor.
  - Select events based on arbitrary cuts on caf objects.
  - Manchester tutorial shows how to select run range.
- **Trigger** processor.
  - Select events based on trigger.
- **EventFlags** processor.
  - Select events based on event flags (very fast).
- **CafeDataQualityProcessor** (cvs package **caf\_dq**).
  - Reject bad runs and bad lbns.

# Caf Level Analysis Tools – Selectors

- The main processors for most kinds of caf objects are various “selectors.”
  - **MuonSelector.**
  - **ElectronSelector/PhotonSelector.**
  - **VertexSelector.**
  - **TrackSelector/IsoTrackSelector.**
  - **JetSelector.**
  - **METSelector.**
  - **TauSelector.**
  - **BTagJetSelector.**
  - **Others.**

# Caf Selectors

- **Selectors** –
  - Live in cvs package **caf\_util**.
  - Inherit from template class **SelectUserObject<T>**.
  - Override method **selectObjects**.
  - Copy selected objects from a specified input branch to output branch. Rejected objects may be stored in a third branch.
    - Selection based on quality, kinematic cuts, or other criteria.
  - May or may not not modify selected objects.
    - Caf objects already include d0correct corrections.

# Object Corrections at the CAF Level

- In principle, objects corrections are supposed to be handled by `d0correct` (i.e. before `caf`). However, some corrections can be (re-)calculated at the `caf` level.
  - Corrections may have changed since `caf` trees were made.
- Muon objects.
  - **ApplyMuonSmear** processor (`caf_mc_util`).
    - Recalculate MC muon smearing (same as `d0correct`).
- EM objects.
  - **ApplyElectronSmear/ApplyEMSmear** processor.
    - Recalculate MC electron smearing (doesn't exist yet). 😞

# Track Corrections at the CAF Level

- **TrackSmear** processor (**caf\_util**).
  - MC track smearing.
  - Not available in d0correct or as framework package. 😞

# Jet Corrections at the CAF Level

- **ApplyJES** processor (**caf\_util**).
  - Recalculate jes corrections for data or mc (same as d0correct).
- **RunJetSSR** processor (**caf\_mc\_util**).
  - Jet smearing, shifting, removal (mc jet corrections).
  - Not available in d0correct or as framework package. 😞
- **EMJetMatching** processor (**caf\_util**).
  - Recalculate EM-jet matching for EM jet removal.
- Above processors should be run before **JetSelector** processor.

# Missing ET Correction at CAF Level

- **ReComputeMET** processor (**caf\_util**).
  - Recalculate missing ET including jes corrections.
  - Run after **ApplyJES** (& **RunJetSSR**) but before **METSelector**.
  - You can choose which jet correction to propagate to MET.
    - JES or JESMU.
  - You can also choose to propagate muon correction to MET.
    - Do not choose both muon correction and JESMU jet correction (double correction).
  - Recent bug fixes and improvements:
    - Fix propagation of jet smearing.
    - Add tau correction (p18 only currently).

# Missing ET Correction (cont.)

- In principle, only jet response correction is supposed to be propagated to missing ET. This is so far not being done correctly in Run II. Comment from Patrice Verdier: 😞
  - The procedure to propagate the JES correction to MET is not really correct. We are still following a RunI prescription which is to propagate only the Response part to MET, but the formula implemented in jetcorr is not valid anymore with the RunIIA MPF method (which is different from the runI MPF method). The effect of showering and offset is however a small effect. We could therefore want to propagate the entire JES correction to MET (not yet implemented).

# Missing ET Selectors

- **METSelector** processor (**caf\_util**).
  - Apply global missing ET cuts.
  - Store missing ET related values in event (not branch) using **Event::put** (access using **Event::get**).
- **METsigAlg** processor (**caf\_util**).
  - Calculate missing ET significance.
  - Apply cut on missing ET significance.
  - Store missing ET significance in Event (not branch).
  - Must run after **METSelector** processor.

# Tau Id

- Tau candidates are created in d0reco, but tau NN usually needs to be recalculated later with newer version.
  - Tau NN can be recalculated in `tmb_analyze` or at caf level.
- **TauSelector** processor (`caf_util`).
  - Acts as normal selector, and can recalculate tau NN.

# B-Tagging at the CAF Level

- **BTagProcessor (btags\_cert\_caf).**
  - Recalculate b-tags and trfs. Same as framework bid (but framework bid is usually not final when caf trees are made).
  - Can also be used to recalculate just trfs.
- **BTagJetSelector (caf\_util)**
  - Select tagged jets to a different branch.
- **BTagPermuter (btags\_cert\_caf).**
  - Apply trfs to different combinations of mc jets.
  - In caf terms, BTagPermuter is a Controller (can call subprocessors).

# Stat Processor

- **Stat** processor in package **cafe** keeps track of event statistics associated with cut flows.
- **Stat** processor also manages non-unit event weights.
- There should be one **Stat** processor, and it should be instantiated first in processor list.
  - One single **Stat** processor can handle multiple samples or cut flows.
- **StatBbranch** processor makes global event weight persistent in **EventWeight** branch.
  - Currently **EventWeight** is in **cafe** cvs package, not **tmb\_tree**.

# MC Efficiency Corrections

- MC efficiency corrections corresponding to some predefined standard object selections are stored in .spc files in cvs.
  - Cvs package **emid\_eff** – Standard em efficiency corrections.
  - Cvs package **muid\_eff** – Standard muon efficiency corrections.
  - Cvs package **jetid\_eff** – Standard jet efficiency corrections.
  - Both packages only have head branch, not released (you have to check them out).
- Object definition metadata also stored in .spc files.
  - Automatically choose correct efficiency for different object definitions.

# Applying MC Efficiency Corrections

- Various packages can be used to calculate efficiency corrections or scale factors in conjunction with **Stat** processor.
  - **MuonCorr** processor (**caf\_eff\_util**).
    - Apply muon id efficiencies from **muid\_eff** to event weight.
  - **EMCorr** processor (**caf\_eff\_util**).
    - Apply em id efficiencies from **emid\_eff** to event weight.
  - **JetCorr** processor (**caf\_eff\_util**).
    - Apply jet id efficiencies from **jetid\_eff** to event weight.
  - **Bid**.
    - No processor (yet) to apply trfs or taggability to global event weight. ☹️

# Other Reweighting Processors

- There are a number of processors that do various reweighting functions in conjunction with **Stat** processor.
  - **LumiReWeighting** (`caf_mc_util`).
    - Reweight events to change luminosity profile.
  - **ZptReWeighting** (`caf_mc_util`).
    - Reweight Z mc events to fix pT and jet multiplicity distributions.
  - **PDFReweight** (`caf_pdfreweight`).
    - Change pdf function by reweighting.
  - **bFragReweightProc** (`caf_mc_util`).
    - Change b quark fragmentation by reweighting.

# Trigger Tools

- Single object turn-on curves are stored in cvs packages **muid\_eff**, **emid\_eff**, **jetid\_eff**.
- Cvs package **caf\_trigger**.
  - Global trigger efficiency processors.
    - **TriggerProbability** processor.
    - **TriggerORProbability** processor.
  - Reco/trigger object matching processors.
- For additional information, refer to Trigger Studies Group wiki.  
<https://plone4.fnal.gov/P1/D0Wiki/tdaq/tsg/>

# Data Quality Basic Tools

- **recocert** (cvs package **recocert**).
  - Stand alone executable for generating performance plots from tmb.
    - Used for checking d0reco and data quality.
- **physics\_cert** (cvs package **caf\_physics\_cert**).
  - A collection of caf processors for generating standard plots.
- Run & Data Quality Databases (web interface).
  - <http://d0db.fnal.gov/run/runQuery.html>
  - <http://d0db.fnal.gov/qualitygrabber/qualQueries.html>
- **CafeDataQualityProcessor** (cvs package **caf\_dq**).
  - Main dq caf tool for removing events via bad runs and bad lbns.

# Apply Data Quality Cuts at Caf Level

- Use **CafeDataQualityProcessor** (cvs package **caf\_dq**).
  - Requires as input a collection of files known as a **DQDEF** or a **LUMIDEF**.
    - **LUMIDEF** includes **DQDEF** as a subset.
  - Standard **DQDEF**s are centrally maintained in cvs package **dq\_defs**.
  - You make your own **LUMIDEF**, which includes a standard **DQDEF**.
- More information on Data Quality web page.

[http://www-d0.fnal.gov/computing/data\\_quality/d0\\_private/forusers.html](http://www-d0.fnal.gov/computing/data_quality/d0_private/forusers.html)

# Luminosity Tools (Instantaneous)

- At TMB level, instantaneous luminosity comes from **LumiInfoChunk**, which can be (re)generated by **LumiChunkPkg**.
  - **LumiInfoChunk** interface requires you to obtain the tick number externally from **TMBTriggerChunk** (data) or **MinBiasChunk** (mc).
  - **LumiChunkPkg** access luminosity database.
  - **LumiInfoChunk** is not (yet?) implemented in p18. For p18, use old **LumPerTick** class, which accesses luminosity from stage3 files. 😞
- At CAF level, instantaneous luminosity is stored in **TMBGlobal** branch (data and mc) and **TMBMCevtInfo** branch (mc).
  - Run 2b mc p21.05.00 or earlier had bug – no inst. luminosity.

# Luminosity Tools (Integrated)

- The basic tool for obtaining integrated luminosity is program **getLuminosity** program from `lm_tools` product (setup `lm_tools`).
  - Calculates integrated luminosity by trigger, run range, excluding bad runs, bad lbns.
- Combined DQ and integrated luminosity determination.
  - The DQ group supports a higher level interface (on top of **getLuminosity** program) that combines DQ and luminosity.
  - You are required to supply a configuration file known as a **LUMIDEF**, which includes a **DQDEF**, plus extra information about triggers, run range, etc.
  - You analyze your **LUMIDEF** using script `dq_util/util/lumitool.py`, which writes its output into a **LUMICACHE**.

# Summary

- Standard TMB and CAF Datasets.
- Framework level tools -- **d0correct**.
- Efficiency tools – **em\_cert**, **muo\_cert**, **photon\_id\_tools**, **trigeff\_cafe**.
- CAF tools.
  - Object id – selectors.
  - Efficiency correction – weighted events.
- Trigger tools – **caf\_trigger**.
- Data Quality – **caf\_dq**.
- Luminosity.