# Tag & probe updates


Kalanand Mishra

*Fermilab*

(*on behalf of T&P team: N. Adam, J.Berryhill, D. Bjergaard, Z. Gecse, V.Halyo, A. Hunt, M. Makouski, K. Mishra, G. Petrucciani, ... and with help from PhysicsTools group*)


*Analysis Tools* meeting
(March 16, 2010)

# Overview of the new functionality

There are two steps to run tag & probe sequence

- ✦Step I: Make TTree storing tag, probe, and passing probe information
- ✦Step II: compute efficiency as function of any probe variables (e.g., $E_T$, η, ..)

Details at: https://twiki.cern.ch/twiki/bin/view/CMS/TagAndProbe
https://twiki.cern.ch/twiki/bin/view/CMS/ElectronTagAndProbe
https://twiki.cern.ch/twiki/bin/view/CMS/MuonTagAndProbe

Main Customers:

[POG] Egamma: https://twiki.cern.ch/twiki/bin/view/CMS/EgammaOctoberExercise
Muon Z: https://twiki.cern.ch/twiki/bin/view/CMS/OctoberXMuonsTnPEWK
Muon J/ψ: https://twiki.cern.ch/twiki/bin/view/CMS/OctoberXMuonsTnPB
http://indicoprev.cern.ch/getFile.py/access?contribId=4&resId=0&materialId=slides&confId=83367

[PAG] Electroweak: Quarkonia, QCD photon
(SUSY, Top, Exotica)

Hypernews: *Physics Analysis Tools:* hn-cms-physTools@cern.ch

Savannah: https://savannah.cern.ch/projects/physicstool/

# New step-1: Tree-based processing functionality

- ✦ Stores the output as ROOT Tree
- ✦ User can specify which variables to store in the Tree
  - •user has full control over efficiency steps
  - •you no longer need to remember the numeric codes for the efficiency step etc.

- ✦ Two ways to specify which probes are passing the selection criteria
  - string-based cut parser
  - using edm::ValueMap (a generic approach)

- ✦ User can easily store external variables
  - nJets, ΔR(lepton, jet), MET, Ht, ..
  - and use them as binning variable for efficiency

- ✦Store provenance information in the header of the Tree

# New step-2: fitting/SBS functionality

✦User can do fit in any number of probe variables, not just two at a time (as was the case in the old interface)
- both real and discrete
- non-uniform binning supported
- support constrained fit (e.g., μ trigger for probes which are standAlone)

✦PDF can be defined "on the fly" by string parsing to RooFactory
- support for different PDFs (with their own initial values) in different bins

✦User can specify the efficiency categories
- e.g., can compute efficiency for different trigger paths

✦We now save efficiency as RooDataSet as functions of the user-specified probe binning variables
- also automatically save 1D efficiency plots as function of each individual probe variables (efficiency is reported at the mean of the probe variable distribution)

# Validation of the new functionality

✦Extensive testing of the new functionality/configurations in all different analysis cases that we were using the old interface
- done for $Z \to ee$, $Z \to \mu\mu$, quarkonia

✦Bin-by-bin comparison of the efficiencies derived from the "old" and "new" interface using exact same data sample
- a relatively complicated task
- for $Z \to ee$, get essentially the same bin-by-bin efficiency using new interface, number of probe/passing probe candidates sometimes differs a bit but is close
- for quarkonia, the efficiencies at each step and bin-by-bin match w.r.t OCTX.

✦Update of documentation, user example scripts, ... etc.
- already in place for e/μ efficiency from $Z \to ee$, $Z \to \mu\mu$, $J/psi \to \mu\mu$
- I am working on example scripts for computing e/μ efficiency in presence of jets (e.g., as a function of jet multiplicity, jet $\Delta R$, etc).
    - Needed for W/Z/γ + jets, top, exotica, SUSY analyses.

Message: The new interface is validated in CMSSW 3_5_X. All the new changes are queued up for the next 3_5_X and 3_6_X releases.

# Clear, concise, and user-friendly documentation

- Step 0:

```
source /uscmst1/prod/sw/cms/setup/cshrc prod    (at Fermilab LPC only)
scram project CMSSW CMSSW_3_5_3
cd CMSSW_3_5_3/src/
cmsenv
cmscvsroot CMSSW
cvs co -r V02-02-03 PhysicsTools/TagAndProbe
cvs co -r V07-05-00 PhysicsTools/Utilities
scram b -j4
```

- Step 1:

| Parameter name | Meaning | Default value | Comment |
|---|---|---|---|
| | A parameter set containing the names of probe variables to store. | None | User also needs to specify how to get these variables using either a member access function or an external plugin. |
| | A Boolean to specify if one is running over simulation events instead of real data. | "False" | If this parameter is set true then one can do additional MC-related analysis, e.g., truth-matching. |
| airs | A composite candidate whose daughters are tag and probe. | None | |
| | The entire probe collection (i.e., the one that goes into the denominator of the efficiency calculation) | None | Must inherit from a reco::Candidate base class. |
| | A parameter set specifying the passing probe criteria. | None | Can be either a collection of reference vectors of passing probes or a string-based selection. |
| | In case of multiple probe candidates in the same event, one needs to define how to choose the best candidate. | "None" | The available options are: "None" (keep all multiple candidates), "OneProbe" (select one probe randomly), "BestMass" (select the one with tag-probe invariant mass closest to the nominal resonance mass) |

or can depend on event topology. In the
eds to define an external plugin to specify
eight.

-matched efficiency only.

-matched efficiency only. Some useful
3, photon/γ* = 22, J/ψ = 443, Υ = 553

-matched efficiency only.

-matched efficiency only.

-matched efficiency only.

- step 2:

| Parameter name | Meaning | Default value | Comment |
|---|---|---|---|
| InputFileName | Name of the input ROOT file | None | |
| InputDirectoryName | Name of the relevant directory in the input ROOT file | None | This information is needed because the input file may contain several tag-probe directories |
| InputTreeName | Name of the relevant input tree within the above directory in the input ROOT file | None | |
| OutputFileName | Name of the output file to store efficiency results. | None | |
| Variables | Name of the variables needed for efficiency computation. | None | Defines all probe variables available in the input tree and intended for use in the efficiency computation. |
| Categories | Name of the variables needed for efficiency computation. | None | Defines all Boolean probe variables available in the input tree and intended for use in the efficiency computation (e.g., isMCTruth, isPassingHLT, ..). |
| PDFs | Signal and background shapes for likelihood fit. | None | Uses RooFit's "factory" syntax. Each PDF needs to define "signal", "backgroundPass", "backgroundFail" components. |
| Efficiencies | Defines details of efficiency calculation: what PDF to use for fitting and how to bin the data. | None | There will be a separate output directory for each calculation that includes a simultaneous fit, side band subtraction, and counting. |
| NumCPU | Number of CPU on the host machine that can be used for simultaneous fit. | 1 | Don't put a number higher than the available number of CPU's on the host machine ! |
| SaveWorkspace | Do you want to save the RooFit "work space" containing information about signal+background PDFs and their parameters ? | False | Don't bother about this parameter if you have no idea what a "work space" is ! |

## The goal

User should not have to ask about common functionality options in hypernews !

# Tag & probe in FWLite ("TPLite")

TPLite has been available since CMSSW 3.1.X. The current effort is directed at integrating into the new functionality and porting to the main trunk of the package.

Details at:
http://indico.cern.ch/contributionDisplay.py?contribId=19&confId=68267
https://twiki.cern.ch/twiki/bin/viewauth/CMS/PhysicsToolsDevTagAndProbe

**SideBandSubtract status:**

- Validated and integrated into PhysicsTools/Utilities
- Intelligent enough to construct histograms from the *RooDataSet* rather than having the user specify them
- Plans to incorporate *RooWorkspace* into the object

**Integration Status:**

- FWLite SBS code is ported into CMSSW
- Working on integrating *RooSimultaneous* PDF in the workspace in FWLite
- Need to validate changes against old FWLite code
- Additional work under progress

FWLite code and Full Framework code will be integrated and factorized to present a cohesive package that can run in both places

# Interface with POGs

## Egamma electron efficiency sequence from Z→ee

**Tag Selection**

- GsfElectrons.
- $|\eta| < 1.4442$ OR $1.56 < |\eta| < 2.5$
- $E_T > 20$ GeV/c.
- Isolation
- Electron ID: Loose
- Trigger: Electron Ele15

**Probe Selection**

- $|\eta| < 1.4442$ OR $1.56 < |\eta| < 2.5$
- $E_T > 20$ GeV/c.
- Fit the tag-probe invariant mass to get the number of signal events.

Obtain factorized efficiencies for passing probes:

SuperCluster → GsfElectron → Isolation → ID → HLT

offline electron reconstruction efficiency with respect to acceptance

electron trigger efficiency w.r.t. offline selection

## Muon efficiency sequence from Z→μμ

**Tag Selection**

- Global Muons.
- $p_T > 20$ GeV/c.
- Track isolation
- Trigger: Mu9

**Probe Selection**

- $p_T > 20$ GeV/c.
- Fit the tag-probe invariant mass to get the number of signal events.

Obtain factorized efficiencies for passing probes:

Track → Standalone Muon → Isolation → HLT

offline muon reconstruction efficiency w.r.t. acceptance

muon trigger efficiency w.r.t. offline selection

## Low $p_T$ μ efficiency from J/ψ→μμ

tag = global muon passing Mu0+Track or Mu3+Track

- Muon reconstruction efficiency:
  track → global or tracker muon using that track
- Trigger efficiency:
  global or tracker muon → muon firing HLT trigger
- Inner tracker efficiency:
  standalone muon → standalone matched to a track

We have standard scripts and work flow in the *TagAndProbe* package to compute these efficiencies for *Egamma* and *Muon* POGs.

# Interface with PAGs

Delivering efficiency for the first CMS papers using leptons. Areas of active work:

**Electroweak**

- ✦Assuming a few hundred Z events, compute efficiency in unbiased way
- ✦Premium on robustness of the result: need to have unbinned likelihood fit and simple sideband-subtraction method work at better than ~% level in presence of high background.
- ✦Need to improve Z lineshape and background shape parametrizations
- ✦There are ideas to improve on uncertainty (e.g., by using random cone to estimate e/μ isolation efficiency - if this works then we can start with a cleaner probe selection in the denominator of the T&P efficiency).

**Quarkonia**

- ✦Correlations between the two μ efficiencies and propagation to xsec.
- ✦Systematical uncertainties due to the averaging within each bin:
  - •we're on the turn on curve, so the efficiency varies much more within a bin than for Z case ➜ map out the efficiency turn-on curve
  - •working with very low $p_T$ standalone muons, fine bins in eta and $p_T$
- ✦Resolution effects important: common resolution function for Upsilon 1S, 2S, 3S
- ✦Crystal Ball lineshape for signal to accommodate FSR

# Summary

✓ Completed the porting of new interface and functionalities.
- was much awaited and long overdue
- save tag, probe, and passing probe information in a ROOT tree
  → much easier to handle across different platforms ("laptop")
- Robust and highly configurable fitting & bkg subtraction interface
  → save output as *RooDataSets* for use in physics analyses by users

✦ We are now focussing on the user support part
- since CMS is getting fully into physics analysis mode
- first focus on delivering efficiency for Ewk W/Z analysis (**V**ector **B**oson **T**ask **F**orce), quarkonia analysis (QTF)
- already have infrastructure to store efficiency tables and stat errors
- work with POG/PAGs on logistics, support for evaluating syst. errors

✓ Inputs, ideas, and volunteers are always welcome !