University of Sussex

Overview and status of the validation framework

Reco. forum, Sep. 2016

Luke Vinton, University of Sussex

Outline

- Framework designed by Matthew Tamsett
- More details found here: https://cdcvs.fnal.gov/redmine/projects/novaart/repository/entry/trunk/Validation/analysis/README
- Last update given by Bruno Zamorano in DocDB-14824
  - useful overview of the website
- Introduction to the validation framework
- Brief guide on how the framework works
- Histogram naming convention
- Current state
- To-do list

The validation framework

- The validation framework was designed to lessen the load of validation plot making
- Instructions on installing and running the framework can be found here: **https://cdcvs.fnal.gov/redmine/projects/novaart/repository/entry/trunk/Validation/analysis/README**

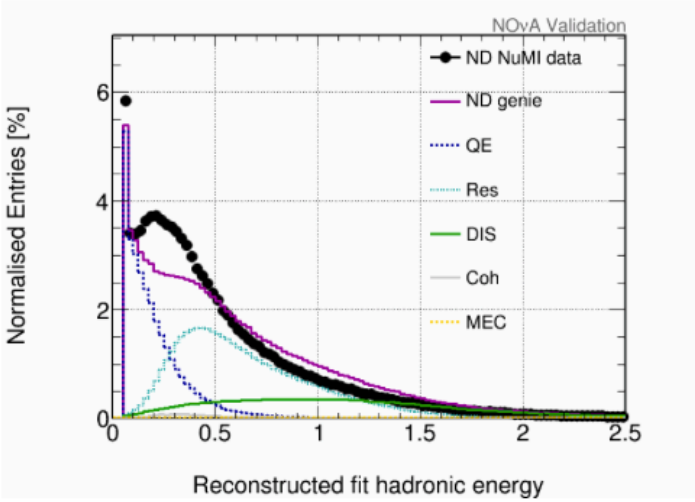
- Validation framework works by running a python script which points to a CAFAna macro via json configuration file
  - The json file contains configurations for the novasoft release, validation plot version, datasets and shell scripts to be run
  - Each run over a macro produces a root file containing histograms defined by the macro and dataset

- Example:
  - ***python $NOVAANAVALID_DIR/core/controller/analysis_validation.py –c /nova/app/users/vinton/ana_valid/nu_mu/configurations/caf_numu_nd_cutflow.json***

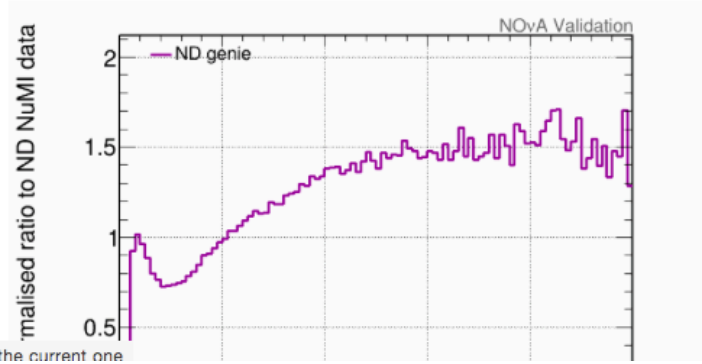
- The shell scripts each point to a CAFAna macro
  - For example, one of the shell scripts pointed to by caf_numu_nd_cutflow.json is nu_mu/components/run_caf_numu_nd_cutflow.sh, which runs: ***cafe -bq caf_numu_nd_cutflow.C***

The validation framework

- A website displaying all validation components created by a user can be generated with: ***python $NOVAANAVALID_DIR/core/view/makeHTMLView.py***
  - The website is created at: **http://nusoft.fnal.gov/nova/users/<USERNAME>/validation/**
- The website user can select the datasets they want to compare
- Cuts can be selected from a dropdown menu next to a plot (If the CAFAna macro produced hists for different cuts, see “naming convention” slide)



png



nd the current one

Reconstructed fit hadronic energy

ND NuMI data: prod_decaf_S15-05-22a_nd_numi_numu_contain_goodruns

ND genie: rocco_decaf_S15-05-22_nd_genie_fhc_nonswap_genierw_numu_contain

Dataset	Entries	Mean	Mean diff [%]
ND NuMI data	501,010	0.52	0.00
ND genie	1,510,188	0.61	17.07

Controls

Log y: disable

Normalisation: none

Truth: By component

Ratio zoom: none

Reco cuts:

Top of page

Aside: naming convention

- All histograms must follow the naming convention:
  - <**category**>-<**variable**>-<*reco_level*>-<*true_level*>

    - **category**: logical category the plots fall into (reco, true, mixed, sam)
    - **variable**: observable
    - *reco_level*: reconstruction cuts
    - *true_level*: QE,Res,DIS,Coh

  - both *reco_level* and *true_level* are optional

  - Examples of acceptable names:
    - reco-reco_numu_energy
    - reco-reco_numu_energy-QE
    - true-numu_energy
    - true-numu_energy-n_minus_1_preselection

- The website user can choose the truth and reco level for each plot, provided there are multiple reco_levels and/or true_levels per <**category**>-<**variable**> combination

Changes needed to run art modules

- The validation CAFAna jobs currently dump output histograms in a flat format directly to a directory
    - However, Art modules dump the histograms into a subdirectory within a file. To enable use of the framework with art modules we should choose one of the two output file structures and apply to both art and CAFAna validation jobs

- Both the output file structure and the use of multiple module Art jobs need attention before art modules can be run with the framework

Current state

- Website populated with image files (eps)
- Can produce plots comparing variables in multiple datasets (data or MC)
  - validation with CAFAna working smoothly
  - work needed to enable running art modules

- Unfortunately, I don't have the time to develop this any further, framework needs a new champion
- further development will require some understanding of json, python and bash

Future plans

- framework needs adjusting to run art jobs
- I don't plan to have the time to develop this further

- **framework needs a new champion!**

  - Sussex group can provide support to get started
  - Framework is not complicated, get stuck in!
  - **Great** way to learn and hone **transferable** computing **skills**

Backup

The validation framework

- The validation framework was designed to lessen the load of validation plot making
- Instructions on installing and running the framework can be found here: **https://cdcvs.fnal.gov/redmine/projects/novaart/repository/entry/trunk/Validation/analysis/README**

- Validation framework works by running a python script which points to a CAFAna macro via json configuration file
  - Each macro is run once for each dataset defined in the json file
  - Each run over a macro produces a root file containing histograms defined by the macro and dataset

- The website is created and populated with image files by running "makeHTMLView.py"
  - Mulitple image files are created for each histogram type
    - Image files are made for all combinations of data/MC/Cuts for each detector
- The website user can select the datasets they want to compare
- Cuts can be selected from a dropdown menu next to a plot (If the CAFAna macro produced hists for different cuts, see "naming convention" slide)

The validation framework

- Validation jobs are started by running the python validation script with a json file as argument, for example:
  - ***python $NOVAANAVALID_DIR/core/controller/analysis_validation.py -c /nova/app/users/vinton/ana_valid/nu_mu/configurations/caf_numu_nd_cutflow.json***
  - The json file contains configurations for the novasoft release, validation plot version, datasets and shell scripts to be run

- The shell scripts each point to a CAFAna macro
  - For example, one of the shell scripts pointed to by caf_numu_nd_cutflow.json is nu_mu/components/run_caf_numu_nd_cutflow.sh, which runs: ***cafe -bq caf_numu_nd_cutflow.C***

- A website displaying all validation components created by a user can be generated with: ***python $NOVAANAVALID_DIR/core/view/makeHTMLView.py***
  - The website created at: **http://nusoft.fnal.gov/nova/users/<USERNAME>/validation/**