Covariance Matrix Fit Users' Guide

Tom Carroll, Adam Lister and Brian Rebel

May 2019

Abstract

This technote details a new method for performing fits for oscillation parameters by using covariance matrices.

Contents

1 Introduction

This technote is intended to be a Users' guide documenting how the the Covariance Matrix Fit (CMF) analysis is run start-to-finish such that it is reproducible.

2 Structure

The CMF code lives within `nova/CovarianceMatrixFit`. Within this directory there are several directories:

```
core   : the core classes that CMF is built on: EventLists, VarVals,
   ↪ ShifterAndWeighter
data   : contains root files for calibration systematic uncertainties
dataProducts: data products and structs commonly used in CMF analysis
fhicl  : contains all fhicl files
macros   : useful .C files
modules  : art modules and plugins which are run with fhicl files
scripts  : scripts for, i.e. submitting to the grid
utilities : only the bin utility lives here, this may be removed in the
   ↪ future
```

3 Generating Covariance Matrices

This section contains information on how to generate covariance matrices for a given systematic uncertainty both locally.

3.1 Locally

In order to generate a covariance matrix locally, a single fhicl file can be run [1],

```
cmf_covariancematrixmakerjob.fcl
```

[2]

Inside this fhicl file, there are three options that a user should configure:

```
TREEFILE : Path to EventList file. For now these are located in /nova/ana/
    ↪ users/brebel/skimmed
SYSTPAR  : Systematic parameter to vary
NUMITER  : Number of iterations of the systematic to run (i.e. number of
    ↪ universes)
```

The options for which systematics you can choose can be found in `CMF_SystematicParameters` ↪ `.fcl`

Once these substitutions have been made, a covariance matrix can be generated with the following command

```
art -c cmf_covariancematrixmakerjob.fcl
```

[1] I'd recommend copying this to your `SER/nova/ana/users/$U` area and running from there.

[2] Ensure you're using the CMF version of this file. Another version, `covariancematrixmakerjob.fcl` exists, but is related to the FNEX framework and is not what you should be running.

3.2 On The Grid

Running on the grid is made easy by the existence of a bash script, located in

```
CovarianceMatrixFit/scripts/CMF_Run_Covariance_Grid.sh
```

which has a usage:

```
usage: CMF_Run_Covariance_Grid.sh <systematics> <eventlist file> <local
   ↪ products> <output dir>
<systematics>: specify one of calib, genie, mec, nue, norm, reco, xsec1,
   ↪ xsec2, xsec3
<eventlist file>: full path to event list tree file in pnfs
<local products>: name of the local products directory
<output dir>: top level directory for output matrix root files
```

where the `<local products>` is the name of a tar file (without the `.tar.bz2`) which should be stored in your scratch area (`/pnfs/scratch/users/${USER}`).

This script by default will run a set of several uncertainties defined by the `<systematics>` tag. Each systematic uncertainty will have by default 2000 systematic universes across 200 grid jobs. This can be modified by changing the variables in the `setVariables()` function in the bash script.

4 Generating Event Lists

This section details how to generate a CMF-style tree of selected events using the CAF selections and how to compare the generated tree to the CAF selection.

4.1 Producing EventListTree Files

The CovarianceMatrixFit package provides an art module to read the decaf or concatenated CAF files from a SAM dataset and produce output files containing the necessary TTrees to

use as event lists in CMF. It also has a set of scripts that can be used to run that process.

4.1.1 art Module: CMFCAFToEventLists

The module is located at `CovarianceMatrixFit/modules/CMFCAFToEventLists\_module ↪ .cc`. The module is an EDAnalyzer, but does not actually do anything in the analyze method because the CAFs are not art files. We use a module here to be able to configure the conversion job and because art provides a nice state machine that processes jobs in a particular order. All the work of the module is done from methods called by the module's `endJob` method. The `CMFCAFToEventLists::FillVariables()` method calls other methods to fill the `cmf::DataVarVals`, `cmf::EventId` and `cmf::MCVarVals` objects. It also calls methods to determine if the event passes the event selection determined through the configuration. The code is the best resource for understanding how these steps are accomplished and details will not be given here. Once the events have been selected they are written into ROOT TTrees based on the metadata information for each event.

4.1.2 script: makeEventListsFromDecaf.sh

The script that runs the above module is located at

```
CovarianceMatrixFit/scripts/makeEventListsFromDecaf.sh.
```

It has six functions; one function prints the necessary command line inputs to the terminal when the script is called without inputs. The output from that function is

```
Usage: makeEventLists.sh <outdir> <detector> <file type> <selection> <horn
    ↪ current> <files per iteration> <tag> <systematic name>
<outdir>: directory where the output root files will go
<detector>: Near or Far
<file type>: data, nonswap, fluxswap, tauswap, cosmicbackground,
    ↪ rockfluxswap, rocknonswap
```

```
<selection>: NuESel, NCSel, or NuMuSel
<horn current>: fhc or rhc
<files per iteration>: number of files for each subjob
<tag>: analysis tag, eg 2018, 2019
<systematic name>: optional name for systematically shifted set
```

The last argument is optional and is only used when making event lists for systematically shifted datasets.

The remaining functions handle converting files from SAM datasets containing the concatenated CAF files. The most important function for converting the concatenated CAF files is `makeXROOTDFile` which creates a text file containing the xrootd location of the concatenated CAF files. This method checks which combination of detector, file type, selection, and horn current is being requested and then creates the corresponding SAM dataset name from those inputs. **NB:** the SAM dataset name is not dynamically determined based on the production run desired. The base of the name must be changed by hand for each production run.

The script configures the art job to produce a text file containing the event identification information, ie run, subrun, event, slice, and cycle numbers, the event energy, and the CVN value for the event. This text file has a name of the form `selectedEvents_<detector>_<selection>_<horn current>_<file type>_<systematic>_<analysis tag>.txt`.

4.1.3 script: startEventLists.sh

The script that calls the `makeEventListsFromDecaf.sh` script is located at `CovarianceMatrixFit/scripts/startEventLists.sh`. If the script is called without arguments it prints the necessary command line inputs to the terminal,

```
Usage: startEventLists.sh <detector> <file type> <files per iteration> <tag> <systematic\_name>
```

```
<detector>: Near or Far
<file type>: data, nonswap, fluxswap, tauswap, cosmicbackground,
   ↪ rockfluxswap, rocknonswap
<files per iteration>: number of files for each subjob
<tag>: analysis tag, eg 2018, 2019
<outdir>: location where the output root files should go
<systematic name>: optional name for systematically shifted set
```

The script will start a series of jobs that run locally and in the background to convert each of the possible event selections and horn currents available in the `makeEventListsFromDecaf ↪ .sh` script.

4.2 Producing CAF Text Lists

4.2.1 cafe macro: get_eventlist.C

The `CovarianceMatrixFit/macros/cafe/get\_eventlist.C` macro creates a text file list of events from a given SAM CAF concatenated dataset. The macro must be run with the `cafe` executable. The macro has to be provided with the desired SAM dataset name, the name of the output text file, the selection to use and the detector. Those values are given nonsense default values which can be replaced easily using a `sed` command as is done in the `CovarianceMatrixFit/scripts/compareEventLists.sh` script discussed next. If one wants to change the values at run time they can be given as arguments to the macro at run time as well. The output file name has the form `cafEvents_<detector>_<selection>_< ↪ horn current>_<file type>_<systematic>_<analysis tag>.txt`.

This macro must be run from a terminal that is set up to run novasoft in an SRT environment. One cannot run the SRT and ups/CMake environments at the same time in the same terminal.

4.2.2 macro: compareEvents.C

The `CovarianceMattrixFit/macros/compareEvents.C` macro compares the output text files of selected events produced by the `CovarianceMatrixFit/macros/cafe/get\_eventlist` ↪ `.C` macro and the `CovarianceMatrixFit/modules/CMFCAFToEventLists\_module.cc` module. The file names for the CAF and CMF text files are assumed to follow the forms indicated above.

This macro makes histograms comparing the selected number of events by CAF and CMF, including any events that are in one sample but not the other. It also produces histograms showing the differences between the reconstructed energy, product of PPFX and cross section central value weights, and the CVN score. The macro produces comparisons of the weighted and unweighted spectra as well.

4.2.3 script: compareEventLists.sh

The `CovarianceMatrixFit/scripts/compareEventLists.sh` script will run the `CovarianceMatrixFit` ↪ `/macros/cafe/get_eventlist.C` macro and compare the output from that macro to the output text file from the `CovarianceMatrixFit/modules/CMFCAFToEventLists_module.` ↪ `cc`. This script also runs the `CovarianceMattrixFit/macros/compareEvents.C` macro described above. If the script is called without arguments it prints the necessary command line arguments to the screen,

```
Usage: compareEventLists.sh $<selection> <filetype> <tag> <systematic name
   ↪ >
<selection>: nue, numu, nus
<filetype>: data, nonswap, fluxswap, tauswap, cosmicbackground,
   ↪ rockfluxswap, rocknonswap
<tag>: analysis tag, eg 2018, 2019
<systematic name>: optional name for systematically shifted set
```

Both the SRT build of novasoft and samweb must be set up or the script will exit with a corresponding message. The script will compare all combinations of horn current and detector for the given set of selection and file type chosen at run time.

4.3 Looking for Duplicate Events among Selections

4.3.1 macro: compareSelectedSets.C

The `CovarianceMatrixFit/macros/compareSelectSets.C` macro compares the event lists in the text files produced by the `CovarianceMatrixFit/modules/CMFCAFToEventListsd\_module.cc` module to see if any events in one PID selection, ie ν_μ, ν_e or NC, overlaps with any other selection. It requires the detector, the selections to be compared, the file type, horn current and analysis tag be specified. The macro produces a text file containing the duplicated event identification information. The produced file name has the form `duplicate_list_<detector>_<selection 1>_<selection 2>_<horn current>_<file type>_<analysis tag>.txt`. If one wanted to compare the CAF selected events, the macro would have to be edited such that the input lists to compare have file names starting with `cafEvents` rather than `selectedEvents`.

4.4 script: compareSelections.sh

The `CovarianceMatrixFit/scripts/compareSelections.sh` script calls the `CovarianceMatrixFit/macros/compareSelectSets.C` macro for all combinations of detector, horn current and file type for the analysis tag and set of selections to compare. If the script is called without arguments it prints the necessary command line arguments to the screen,

```
Usage: compareSelections.sh $<tag> <selection 1> <selection 2>$
<tag>: analysis tag, eg 2018, 2019
<selection X>: nue, numu, nus
```