Data Preservation at MINERvA

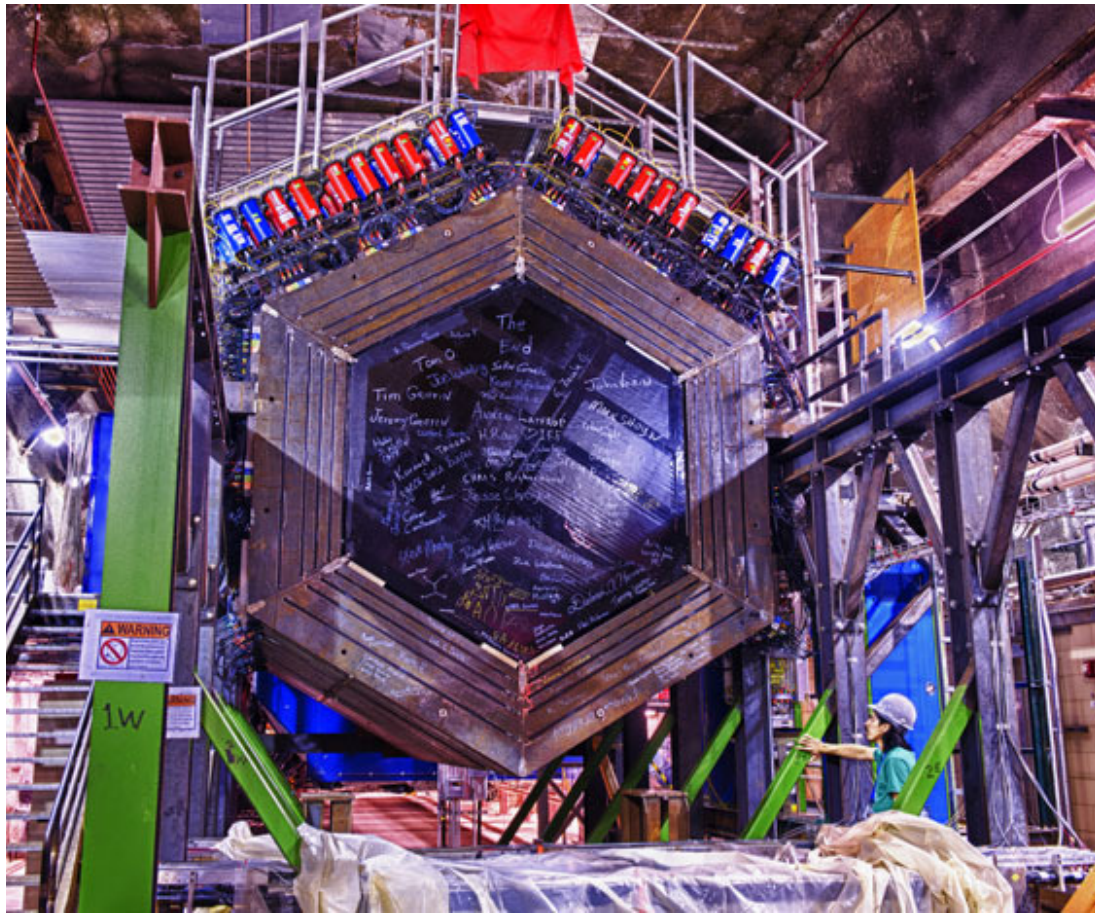
Noah Harvey Vaughan *(they/them)*
PhD Candidate, Oregon State University

NuXTract 2023
Wednesday October 6th, 2023

Oregon State University

Outline

- Overview of MINERvA's strategy
- Master Ana Dev (MAD) Tuples
- MINERvA's systematics handling
- MINERvA Analysis Toolkit (MAT)
- Example of user development



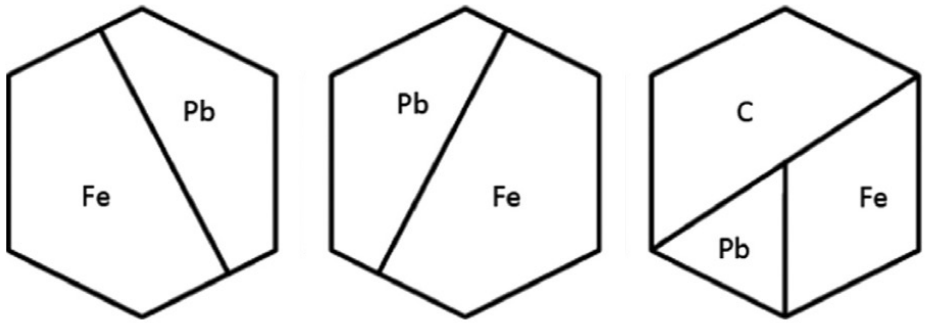
MINERvA Experiment

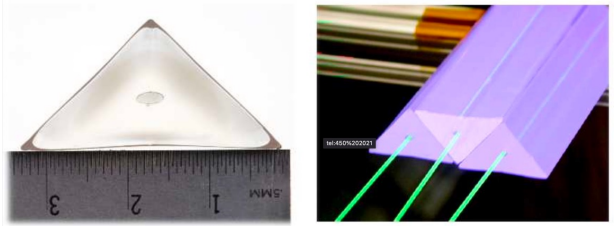
MINERvA Experiment

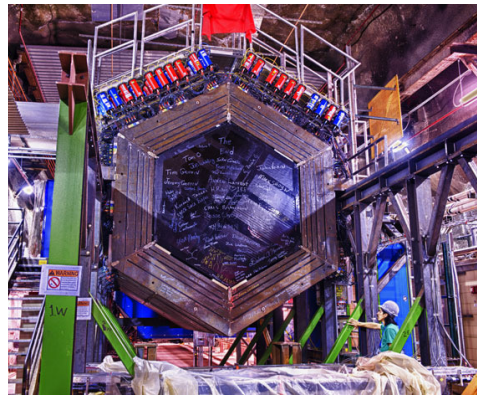
MINERvA = **M**ain **IN**jector **ExpeR**iment on **v** (nu) **A** (atom)

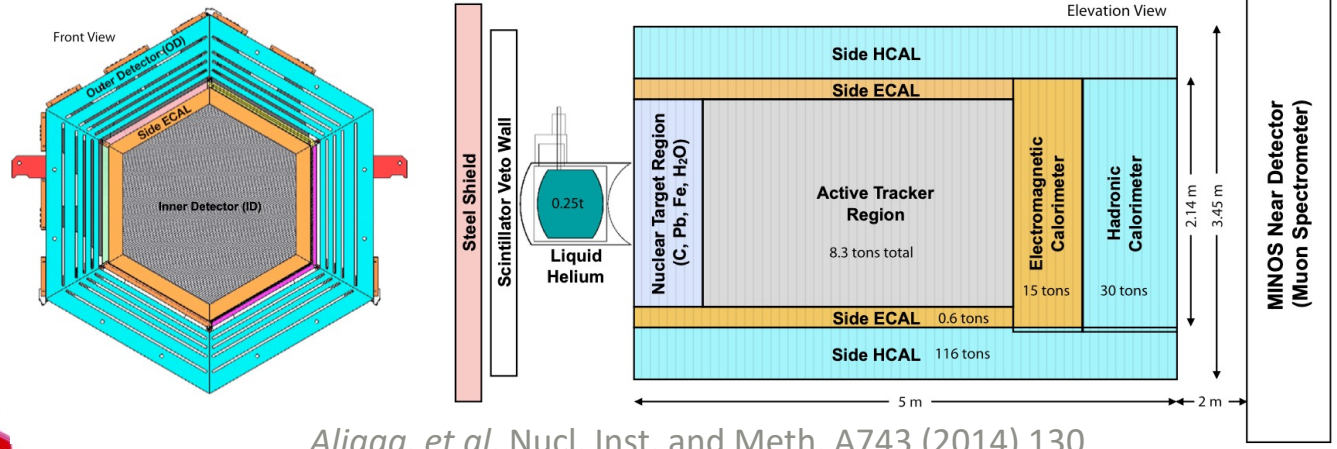
- Dedicated x-section experiment at FNAL
- Data runs from 2009 – 2019:
  - NuMI LE POT: 4.0×10^{20} ν, 1.7×10^{20} $\bar{\nu}$
  - NuMI ME POT: $\sim 3 \times$ LE ν, $\sim 7 \times$ LE $\bar{\nu}$

Geometry of nuclear target planes









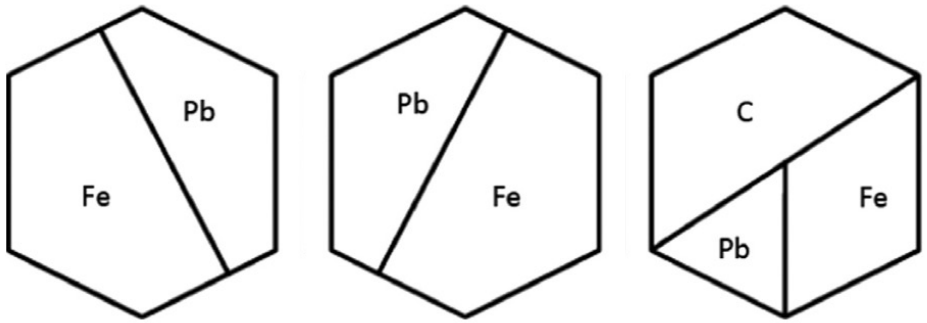
Aliaga, et al. Nucl. Inst. and Meth. A743 (2014) 130

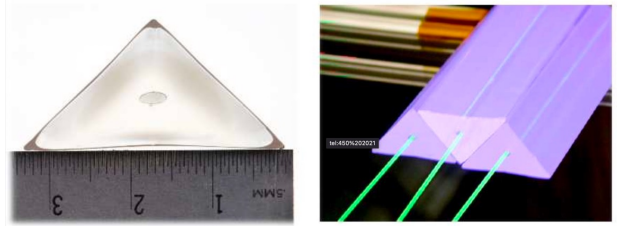
MINERvA Experiment

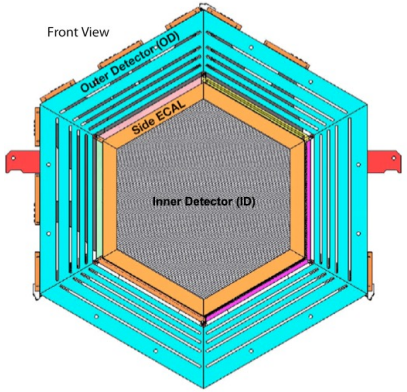
MINERvA = **M**ain **IN**jector **ExpeR**iment on **v** (nu) **A** (atom)

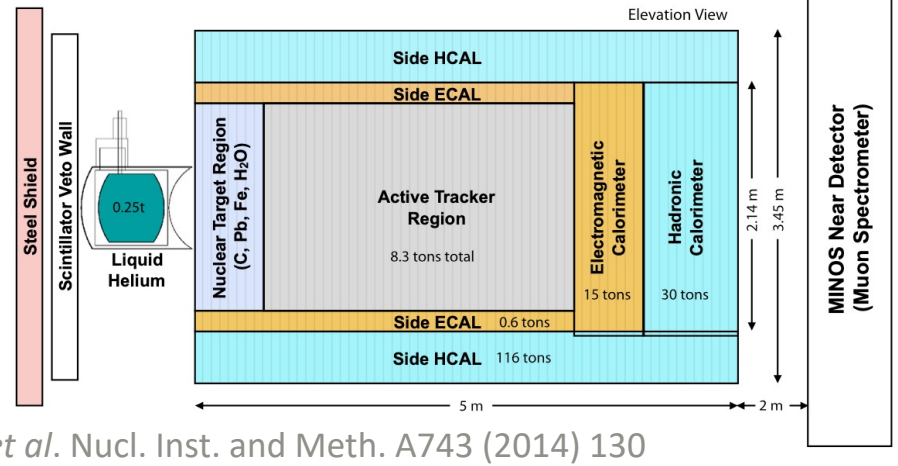
- Dedicated x-section experiment at FNAL
- Data runs from 2009 – 2019:
  - NuMI LE POT: 4.0×10^{20} ν, 1.7×10^{20} $\bar{\nu}$
  - NuMI ME POT: $\sim 3 \times$ LE ν, $\sim 7 \times$ LE $\bar{\nu}$

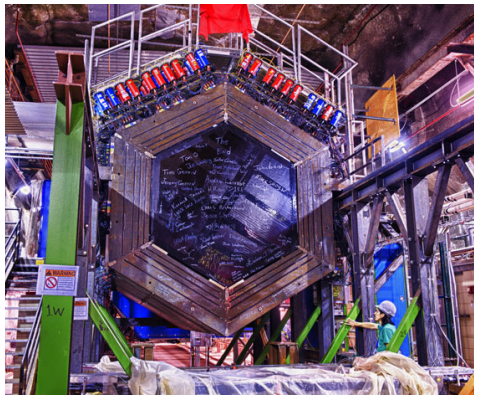
Geometry of nuclear target planes







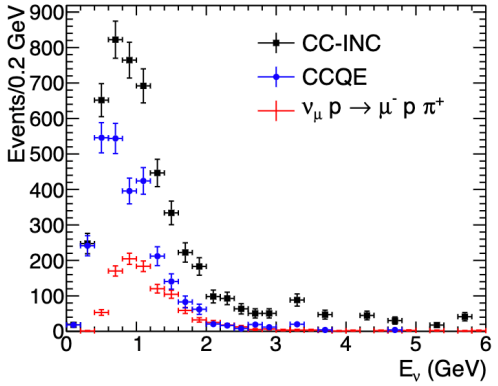




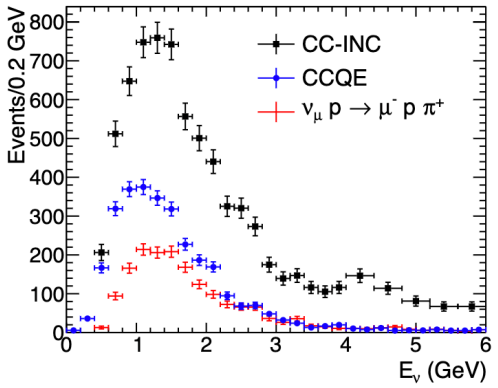
Aliaga, et al. Nucl. Inst. and Meth. A743 (2014) 130

Motivation for Data Preservation

- Models still need constraint for future experiments ⇒ further analysis still required
- MINERvA data is rich and unique
  - More physics than analyzers to look at it!
- Ability to train new analyzers limited, expertise will become dispersed
- Lessons learned from bubble chamber data reanalysis
  - Older data is useful, even if newer detector technologies come out



(a) ANL



(b) BNL

Digitized ANL and BNL bubble chamber data

C. Wilkinson et al., Phys. Rev. D 90, 112017 (2014)

Strategy

Open development of tools used within collaboration, available for other experiments

- Creates a framework for future (and current) analyzers/reanalyzers to build from

Centrally develop detailed and easy to use/access data

- Support necessary reanalysis as possible

Strategy

Open development of tools used within collaboration, available for other experiments

- Creates a framework for future (and current) analyzers/reanalyzers to build from

Centrally develop detailed and easy to use/access data

- Support necessary reanalysis as possible

Solution:

- Master Ana Dev Tuples (MADTuples): Complete data tuples for long term storage and use
- MINERvA Analysis Toolkit (`MAT`) and `MnvHND`'s: Software for analysis and systematics handling

An Error Analysis Toolkit for Binned Counting Experiments

B. Messerly,[1,*] R. Fine,[2,†] A. Olivier,[2] Z. Ahmad Dar,[3,4] F. Akbar,[4] M. V. Ascencio,[5] A. Bashyal,[6] L. Bellantoni,[7] A. Bercellie,[2] J. L. Bonilla,[8] G. Caceres,[9] T. Cai,[2] M.F. Carneiro,[6,‡] G.A. Díaz,[2] J. Felix,[8] L. Fields,[7] A. Filkins,[3] A. Ghosh,[10,9] S. Gilligan,[6] R. Gran,[11] H. Haider,[4] D.A. Harris,[12,7] S. Henry,[2] S. Jena,[13] D. Jena,[7] J. Kleykamp,[2] M. Kordosky,[3] D. Last,[14] A. Lozano,[9] X.-G. Lu,[15] K.S. McFarland,[2] C. Nguyen,[16] V. Paolone,[1] G.N. Perdue,[7,2] M.A. Ramírez,[14,8] H. Ray,[16] D. Ruterbories,[2] H. Schellman,[6] C.J. Solano Salinas,[17] H. Su,[1] E. Valencia,[3,8] N.H. Vaughan,[6] B. Yaeggy,[10] K. Yang,[15] and L. Zazueta[3]
(The MINERvA Collaboration)[§]

[1] *Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA*
[2] *University of Rochester, Rochester, New York 14627 USA*
[3] *Department of Physics, College of William & Mary, Williamsburg, Virginia 23187, USA*
[4] *AMU Campus, Aligarh, Uttar Pradesh 202001, India*
[5] *Sección Física, Departamento de Ciencias, Pontificia Universidad Católica del Perú, Apartado 1761, Lima, Perú*
[6] *Department of Physics, Oregon State University, Corvallis, Oregon 97331, USA*
[7] *Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*
[8] *Campus León y Campus Guanajuato, Universidad de Guanajuato, Lascurain de Retana No. 5, Colonia Centro, Guanajuato 36000, Guanajuato México.*
[9] *Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud 150, Urca, Rio de Janeiro, Rio de Janeiro, 22290-180, Brazil*
[10] *Departamento de Física, Universidad Técnica Federico Santa María, Avenida España 1680 Casilla 110-V, Valparaíso, Chile*
[11] *Department of Physics, University of Minnesota – Duluth, Duluth, Minnesota 55812, USA*
[12] *York University, Department of Physics and Astronomy, Toronto, Ontario, M3J 1P3 Canada*
[13] *Department of Physical Sciences, IISER Mohali, Knowledge City, SAS Nagar, Mohali - 140306, Punjab, India*
[14] *Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104*
[15] *Oxford University, Department of Physics, Oxford, OX1 3PJ United Kingdom*
[16] *University of Florida, Department of Physics, Gainesville, FL 32611*
[17] *Facultad de Ciencias, Universidad Nacional de Ingeniería, Apartado 31139, Lima, Perú*

(Dated: March 17, 2021)

We introduce the MINERvA Analysis Toolkit (MAT), a utility for centralizing the handling of systematic uncertainties in HEP analyses. The fundamental utilities of the toolkit are the `MnvH1D`, a powerful histogram container class, and the systematic `Universe` classes, which provide a modular implementation of the many universe error analysis approach. These products can be used stand-alone or as part of a complete error analysis prescription. They support the propagation of systematic uncertainty through all stages of analysis, and provide flexibility for an arbitrary level of user customization. This extensible solution to error analysis enables the standardization of systematic uncertainty definitions across an experiment and a transparent user interface to lower the barrier to entry for new analyzers.

arXiv:2103.08677v1 [hep-ex] 15 Mar 2021

* Now at University of Minnesota
† Now at Los Alamos National Laboratory
‡ Now at Brookhaven National Laboratory
§ Please direct correspondence to mess@umn.edu, finer@fnal.gov, and aolivier@ur.rochester.edu

Master Ana Dev (MAD) Tuples

Analysis tuples storing data and simulation for the Data Preservation Era

New production

- Preserve CV's and systematics for the full MINERvA sample
  - Still quite large in total (~10 TB)
- Consolidate data, MC into singular tuple for analyses at MINERvA
- Includes both low- and high-level reco information
  - Gives options for future analyzers of new physics
- Implement more modern reconstruction methods, systematics
- Backwards compatibility
  - Many older analyses can be easily adapted to work with new tuples
- Robust documentation in development
  - Maintain institutional memory, even after expertise is gone

In use

Standard methods to access datatypes in MADTuples

- E.g., `GetInt(branch_name)`, `GetDouble(branch_name)`, `GetVecElem(branch_name)`

Central methods for more common variables, accessing standard named branches

- E.g., `GetPmu()`, `GetElepTrue()`, `GetGenieWeight()`

User methods for analysis specific calculations

- E.g., `GetRecoilEnergy()`, `GetVertex()`, `GetNMichel()`

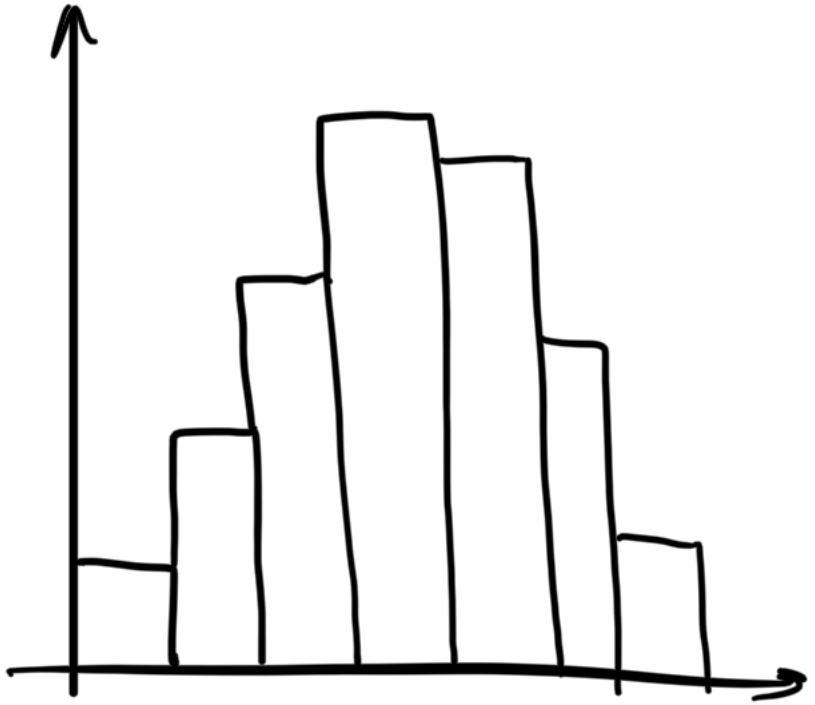
Compactly Handling Systematics

MnvHND histogram objects, `MnvVertErrorBand`, `MnvLatErrorBand`

ROOT Histograms

`TH1D`, `TH2D`, `TH3D` histogram objects

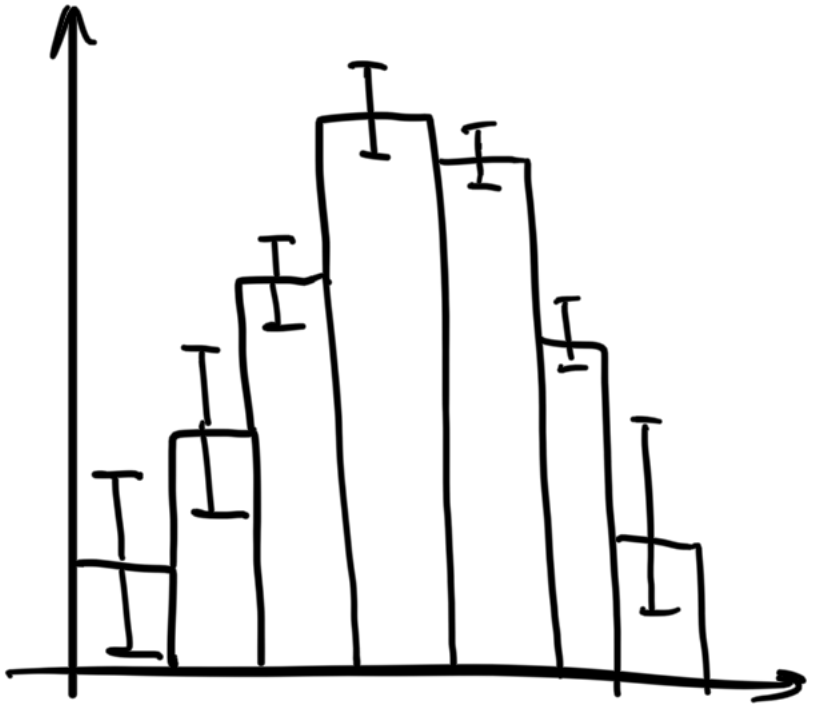
- `Double` central values w/ stat error
- Operations on histograms w/ error propagation
  - Scale, Add, Divide, Multiply, Project
- Can do unfolding using `RooUnfold`
- No built-in systematics handling, left to user to figure it out



ROOT Histograms

`TH1D`, `TH2D`, `TH3D` histogram objects

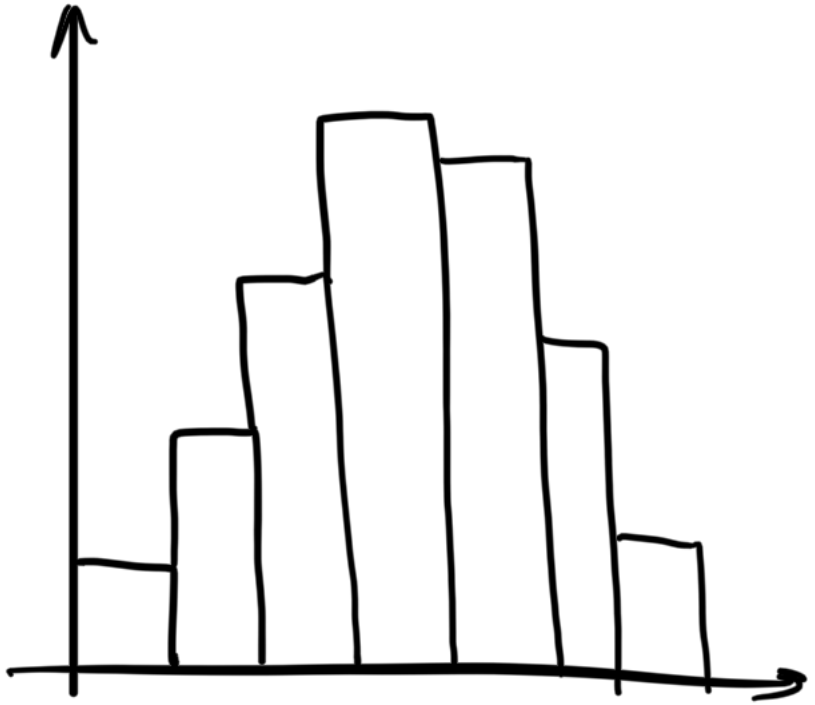
- `Double` central values w/ stat error
- Operations on histograms w/ error propagation
  - Scale, Add, Divide, Multiply, Project
- Can do unfolding using `RooUnfold`
- No built-in systematics handling, left to user to figure it out



MINERvA Histograms

`MnvH1D`, `MnvH2D`, `MnvH3D` inherit from `THND`

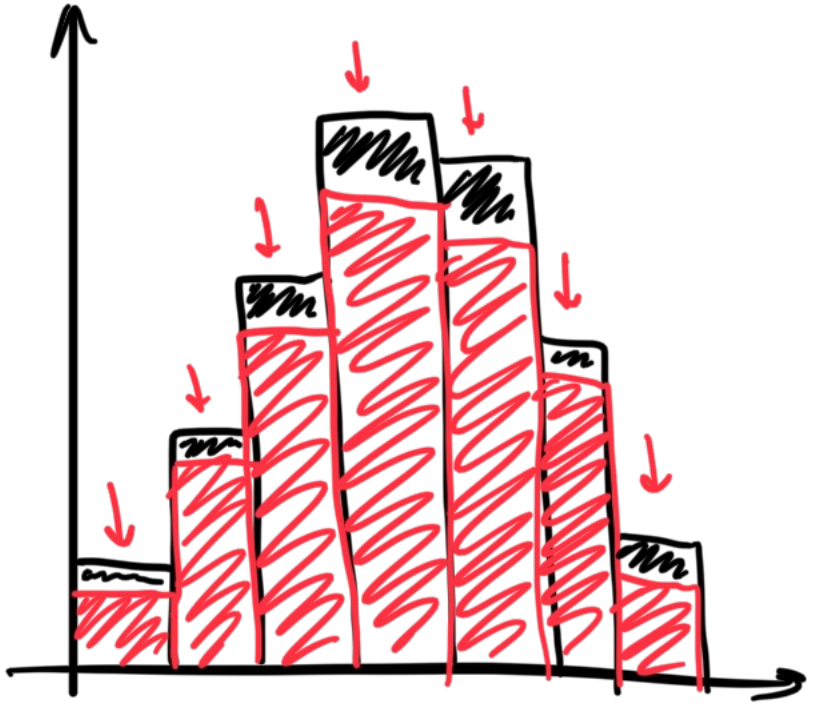
- Central value universe: `CVUniverse`, stored as `ROOT` histogram
- Systematic universes (error bands) shifts and migrations on `CVUniverse`



MINERvA Histograms

`MnvH1D`, `MnvH2D`, `MnvH3D` inherit from `THND`

- Central value universe: `CVUniverse`, stored as `ROOT` histogram
- Systematic universes (error bands) shifts and migrations on `CVUniverse`
  - `MnvVertErrorBand`: vertical shifts on bins, weights on CV

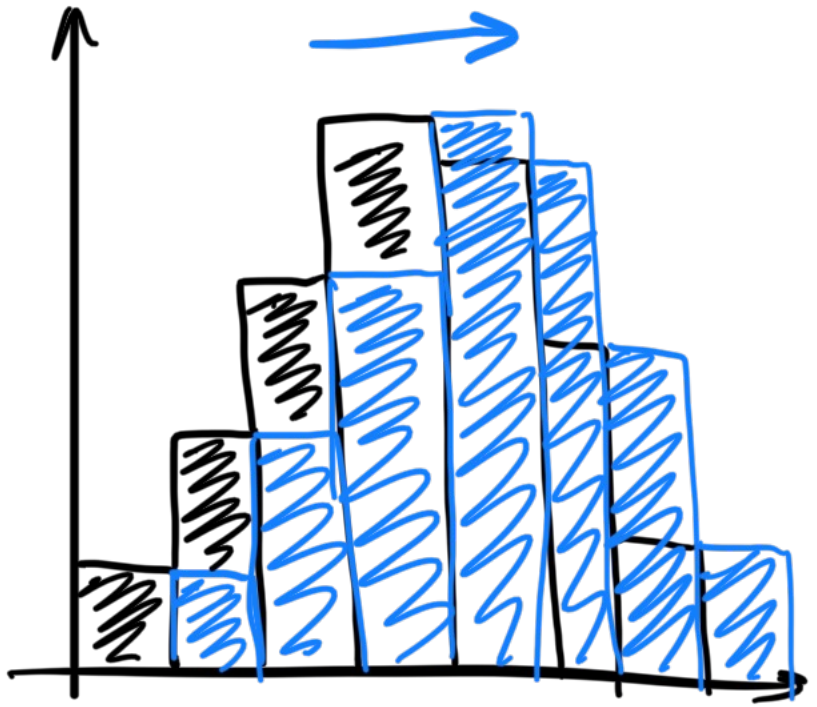


*Vertical universes don't change statistical value, just weights on events

MINERvA Histograms

`MnvH1D`, `MnvH2D`, `MnvH3D` inherit from `THND`

- Central value universe: `CVUniverse`, stored as `ROOT` histogram
- Systematic universes (error bands) shifts and migrations on `CVUniverse`
  - `MnvVertErrorBand`: vertical shifts on bins, weights on CV
  - `MnvLatErrorBand`: lateral shifts, migration from one bin to another

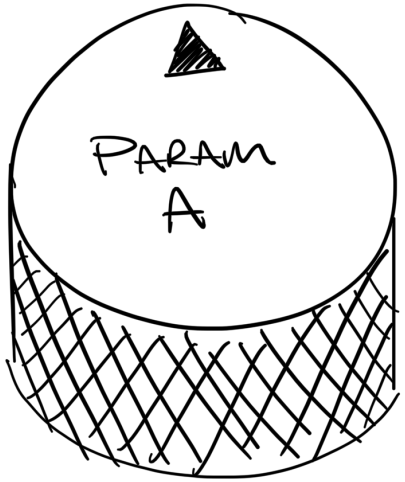


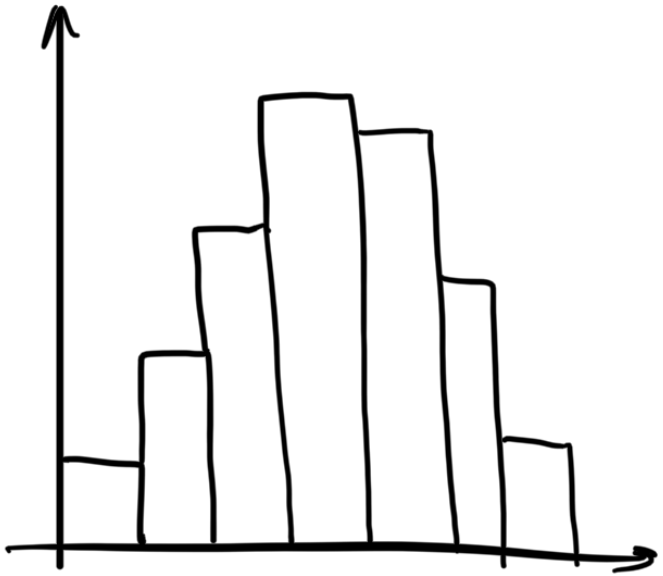
Ex. energy scale on variable can shift laterally

How theses shifts are handled

For model/MC uncertainties, "Two-universe" approach

- Shifts $\pm 1\sigma$ around CV
- Independent parameters, w/o correlation

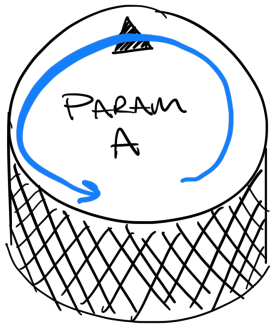


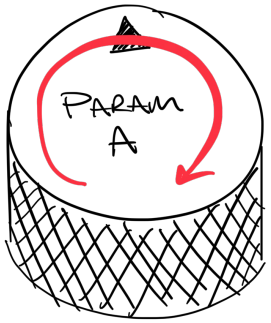


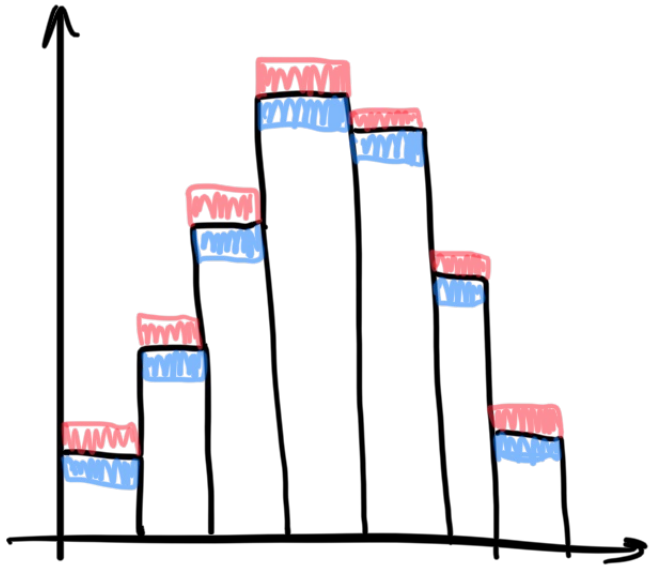
How theses shifts are handled

For model/MC uncertainties, "Two-universe" approach

- Shifts $\pm 1\sigma$ around CV
- Independent parameters, w/o correlation



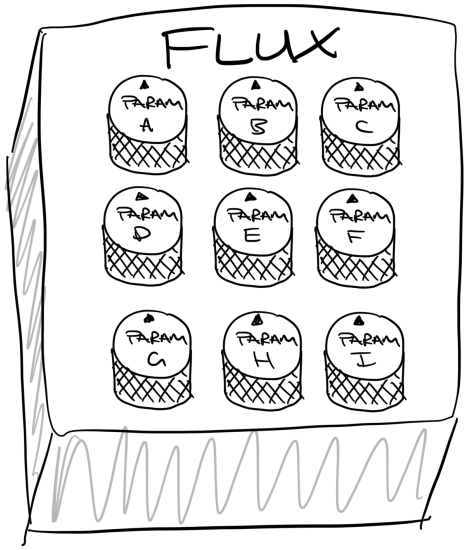




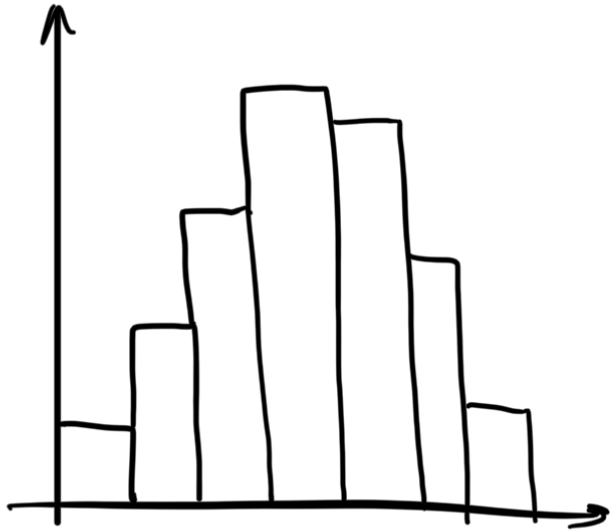
How theses shifts are handled

For more complicated uncertainties, “Multi-universe” approach

- Multiple params w/ correlated uncertainties (e.g., flux, reconstruction)
- Correlations within universe carried to covariance matrix
- Specified number of universes w/ randomly shifted parameters (gaussian distributed) accounting for correlations



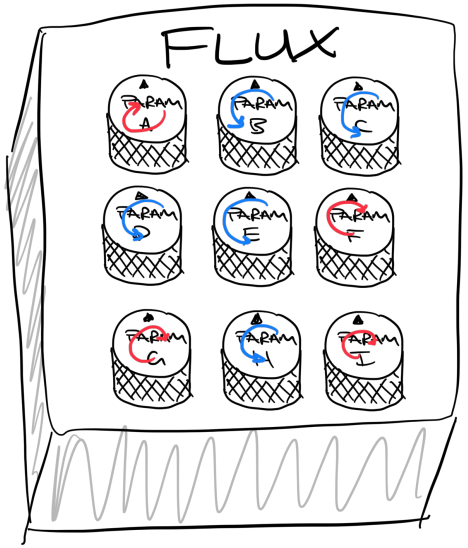
Knobs on focusing parameters, hadron production in beamline, etc.



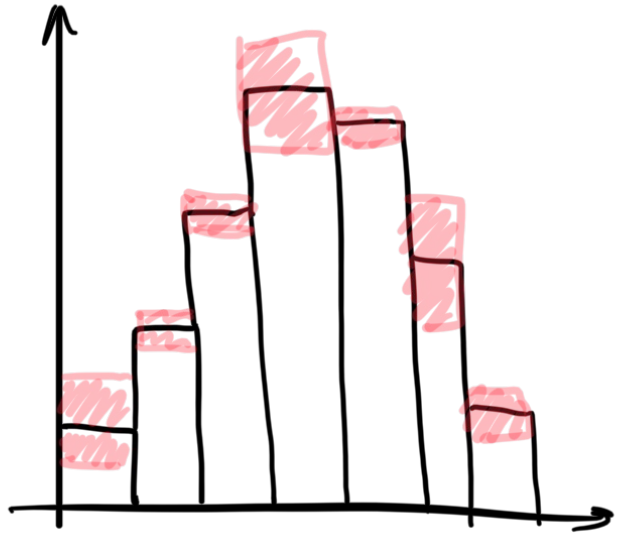
How theses shifts are handled

For more complicated uncertainties, “Multi-universe” approach

- Multiple params w/ correlated uncertainties (e.g., flux, reconstruction)
- Correlations within universe carried to covariance matrix
- Specified number of universes w/ randomly shifted parameters (gaussian distributed) accounting for correlations



$\times N_{Flux\ universes}$ times



Varied params:

$$\vec{v} = \vec{v}_0 + \boldsymbol{L}\vec{r}$$

$\vec{v}_0$: CV on params

$\boldsymbol{L}$: Cholesky decomp of cov

$\vec{r}$: random numbers

Handling Systematic Universes

- Perform full analysis on each universe independently
- Standard systematics developed centrally by collaboration
- New error bands can be added *ad hoc* by users
- Single `MnvHND` histogram object makes systematic error propagation in calculations trivial for the user
  - Cross section extraction is done through one `MnvHND` for each sample

MINERvA Analysis Toolkit

Central tools for handling MINERvA histograms for analysis

Development Hierarchy

Wider community: MAT

- MINERvA Analysis Toolkit (MAT) open and designed for more general use (even outside MINERvA)
- Contains tools to handle `MnvHND` histograms, build simple analyses
- MAT github repo
- μBooNE NC π^0 analysis using MAT

Development Hierarchy

Wider community: MAT

- MINERvA Analysis Toolkit (MAT) open and designed for more general use (even outside MINERvA)
- Contains tools to handle MnvHND histograms, build simple analyses
- MAT github repo
- μBooNE NC π^0 analysis using MAT

Collaboration: MAT-MINERvA

- Extension of MAT with collaboration specific tools
- MParamFiles contains MINERvA-specific parameters
- Useful as guide for other collaborations' extension of MAT
- MAT-MINERvA github repo

Development Hierarchy

Wider community: MAT

- MINERvA Analysis Toolkit (MAT) open and designed for more general use (even outside MINERvA)
- Contains tools to handle MnvHND histograms, build simple analyses
- MAT github repo
- μBooNE NC π^0 analysis using MAT

Collaboration: MAT-MINERvA

- Extension of MAT with collaboration specific tools
- MParamFiles contains MINERvA-specific parameters
- Useful as guide for other collaborations' extension of MAT
- MAT-MINERvA github repo

MAT/MAT-MINERvA is meant to be a toolkit for analyzers to build from

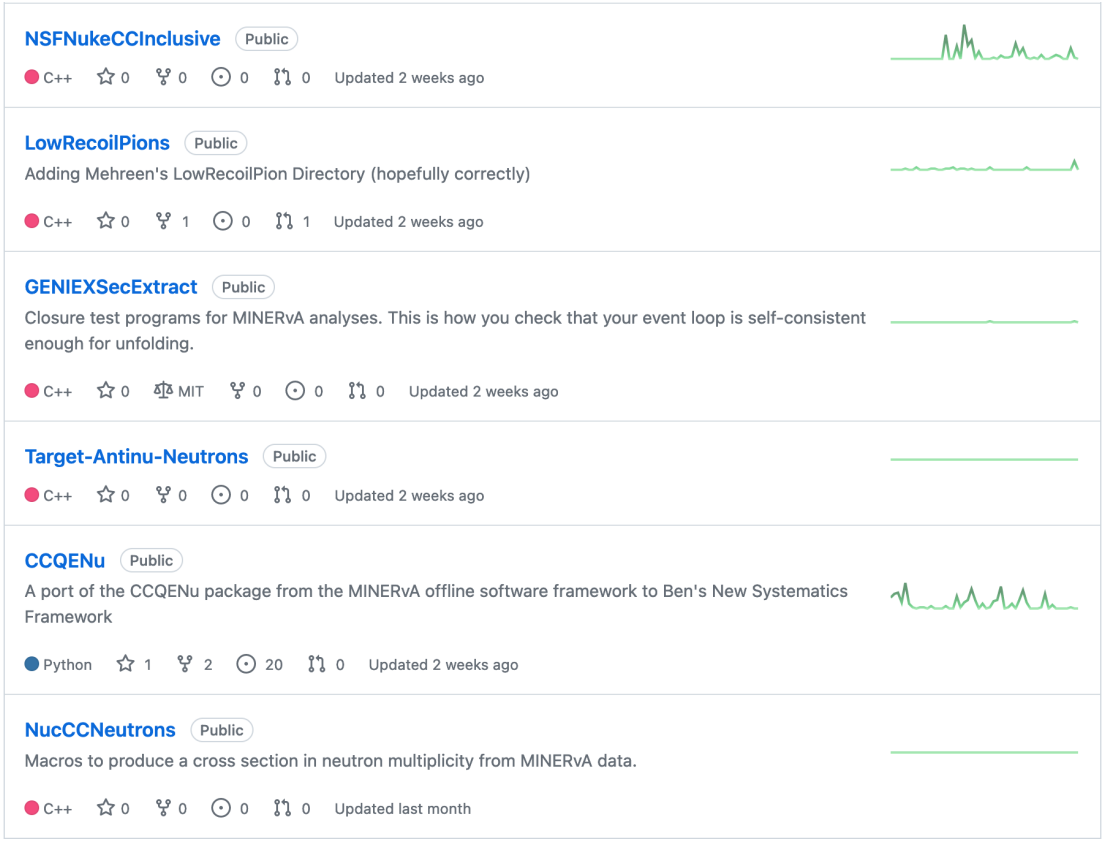
User Development

How to build an analysis, and an example of one in development

User Development

MAT/MAT-MINERvA in use

- Every new MINERvA analyses required to use MAT/MAT-MINERvA
- Example cross section code exists, many analyzers develop from this
  - MINERvA101CrossSection github
- Users often extend tools in MAT to better suit needs specific to analysis
- MINERvA Expt github organization



Analysis Structure

Two macros (usually C++):

- Event loop:
  - **Input** data and MC tuple lists
  - “3 loop structure”: data, selected MC (reco, response), MC truth
  - User specified cuts and weights applied event-by-event
  - **Output** user defined histograms, filled with CV and *systematic universes*
- Cross section extraction:
  - Manipulate histograms, unfold, etc. to extract cross section
  - Relatively lightweight (python possible), though unfolding can be bulky

In practice, usually more complex, but that’s the gist…

Analysis Structure

Two macros (usually C++):

- Event loop:
  - **Input** data and MC tuple lists
  - "3 loop structure": data, selected MC (reco, response), MC truth
  - User specified cuts and weights applied event-by-event
  - **Output** user defined histograms, filled with CV and *systematic universes*
- Cross section extraction:
  - Manipulate histograms, unfold, etc. to extract cross section
  - Relatively lightweight (python possible), though unfolding can be bulky

Recall from yesterday: unfolding is done *universe-by-universe*

In practice, usually more complex, but that's the gist…

Example: CCQENuMAT

First analysis (re)developed entirely using the MAT

- *2D ME QElike* $\overline{\nu}_\mu$ analysis software from pre-MAT era ported to new system
- Used for ongoing *QElike* ν_μ and $\overline{\nu}_\mu$ analyses
- Can be ran locally e.g., on my mac using conda install of ROOT

Extensions from MAT/MAT-MINERvA

- Fitting macro (including systematics)
- Event-by-event MC tuning for background constraint
- Support for higher dimensional (>2D) analyses
- Declarative configuration system for self documentation

Cuts config example


```
"QElike":[
{"name":"NoDeadTime","max":1,"variable":"DeadTime"},
{"name":"ApothemX","max":850,"variable":"ApothemX"},
{"name":"ApothemY","max":850,"variable":"ApothemY"},
{"name":"Z Vertex","min":5980,"max":8422,"variable":"ZVertex"},
{"name":"Has MINOS Match","equals":-1,"variable":"IsMinosMatchTrack"},
{"name":"NuHelicity","equals":2,"variable":"NuHelicity"},
{"name":"Max Muon Angle in Degrees","max":20.0,"variable":"ThetamuDegrees"},
{"name":"Min MuonMomentumPar ","min":1.5,"variable":"PparMuGeV"},
{"name":"Max MuonMomentumPar ","max":15.0,"variable":"PparMuGeV"},
{"name":"Max Multiplicity","max":1,"variable":"Multiplicity"},
{"name":"GoodRecoil","equals":1,"variable":"GoodRecoil"}
],
```

Cuts config example


```
"QElike":[
{"name":"NoDeadTime","max":1,"variable":"DeadTime"},
{"name":"ApothemX","max":850,"variable":"ApothemX"},
{"name":"ApothemY","max":850,"variable":"ApothemY"},
{"name":"Z Vertex","min":5980,"max":8422,"variable":"ZVertex"},
{"name":"Has MINOS Match","equals":-1,"variable":"IsMinosMatchTrack"},
{"name":"NuHelicity","equals":2,"variable":"NuHelicity"},
{"name":"Max Muon Angle in Degrees","max":20.0,"variable":"ThetamuDegrees"},
{"name":"Min MuonMomentumPar ","min":1.5,"variable":"PparMuGeV"},
{"name":"Max MuonMomentumPar ","max":15.0,"variable":"PparMuGeV"},
{"name":"Max Multiplicity","max":1,"variable":"Multiplicity"},
{"name":"GoodRecoil","equals":1,"variable":"GoodRecoil"}
],
```

Name for sanity check cut summary printed during event loop

Cuts config example


```
"QElike":[
{"name":"NoDeadTime","max":1,"variable":"DeadTime"},
{"name":"ApothemX","max":850,"variable":"ApothemX"},
{"name":"ApothemY","max":850,"variable":"ApothemY"},
{"name":"Z Vertex","min":5980,"max":8422,"variable":"ZVertex"},
{"name":"Has MINOS Match","equals":-1,"variable":"IsMinosMatchTrack"},
{"name":"NuHelicity","equals":2,"variable":"NuHelicity"},
{"name":"Max Muon Angle in Degrees","max":20.0 "variable":"ThetamuDegrees"},
{"name":"Min MuonMomentumPar ","min":1.5,"variable":"PparMuGeV"},
{"name":"Max MuonMomentumPar ","max":15.0,"variable":"PparMuGeV"},
{"name":"Max Multiplicity","max":1,"variable":"Multiplicity"},
{"name":"GoodRecoil","equals":1,"variable":"GoodRecoil"}
],
```

Name for sanity check cut summary printed during event loop

Threshold and for the cut (limited to “equals”, “max”, “min”)

Cuts config example


```
"QElike":[
{"name":"NoDeadTime","max":1,"variable":"DeadTime"},
{"name":"ApothemX","max":850,"variable":"ApothemX"},
{"name":"ApothemY","max":850,"variable":"ApothemY"},
{"name":"Z Vertex","min":5980,"max":8422,"variable":"ZVertex"},
{"name":"Has MINOS Match","equals":-1,"variable":"IsMinosMatchTrack"},
{"name":"NuHelicity","equals":2,"variable":"NuHelicity"},
{"name":"Max Muon Angle in Degrees","max":20.0,"variable":"ThetamuDegrees"},
{"name":"Min MuonMomentumPar ","min":1.5,"variable":"PparMuGeV"},
{"name":"Max MuonMomentumPar ","max":15.0,"variable":"PparMuGeV"},
{"name":"Max Multiplicity","max":1,"variable":"Multiplicity"},
{"name":"GoodRecoil","equals":1,"variable":"GoodRecoil"}
],
```

Name for sanity check cut summary printed during event loop

Name of function to access variable used for the cut

Threshold and for the cut (limited to “equals”, “max”, “min”)

Cuts config example

Sample name (can be selected in main “driver” config)

```
'QElike':[
{"name":"NoDeadTime","max":1,"variable":"DeadTime"},
{"name":"ApothemX","max":850,"variable":"ApothemX"},
{"name":"ApothemY","max":850,"variable":"ApothemY"},
{"name":"Z Vertex","min":5980,"max":8422,"variable":"ZVertex"},
{"name":"Has MINOS Match","equals":-1,"variable":"IsMinosMatchTrack"},
{"name":"NuHelicity","equals":2,"variable":"NuHelicity"},
{"name":"Max Muon Angle in Degrees","max":20.0,"variable":"ThetamuDegrees"},
{"name":"Min MuonMomentumPar ","min":1.5,"variable":"PparMuGeV"},
{"name":"Max MuonMomentumPar ","max":15.0,"variable":"PparMuGeV"},
{"name":"Max Multiplicity","max":1,"variable":"Multiplicity"},
{"name":"GoodRecoil","equals":1,"variable":"GoodRecoil"}
],
```

Name for sanity check cut summary printed during event loop

Name of function to access variable used for the cut

Threshold and for the cut (limited to “equals”, “max”, “min”)

Variable config example


```
"ptmu": {
  "reco": "PperpMuGeV",
  "true": "TruePperpMuGeV",
  "name": "ptmu",
  "title": "muon p_{T} (GeV/c)",
  "bins": [0, 0.075, 0.15, 0.25, 0.325, 0.4, 0.475, 0.55, 0.7, 0.85, 1, 1.25, 1.5, 2.5],
  "fluxnorm": false
},
"pzmu": {
  "reco": "PparMuGeV",
  "true": "TruePparMuGeV",
  "name": "pzmu",
  "title": "muon p_{||} (GeV/c)",
  "bins": [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 15],
  "fluxnorm": false
},
```

Variable config example


```
"ptmu": {
  "reco": "PperpMuGeV",
  "true": "TruePperpMuGeV",
  "name": "ptmu",
  "title": "muon p_{T} (GeV/c)",
  "bins": [0, 0.075, 0.15, 0.25, 0.325, 0.4, 0.475, 0.55, 0.7, 0.85, 1, 1.25, 1.5, 2.5]
  "fluxnorm": false
},
"pzmu": {
  "reco": "PparMuGeV",
  "true": "TruePparMuGeV",
  "name": "pzmu",
  "title": "muon p_{||} (GeV/c)",
  "bins": [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 15],
  "fluxnorm": false
},
```

Binning for the histogram

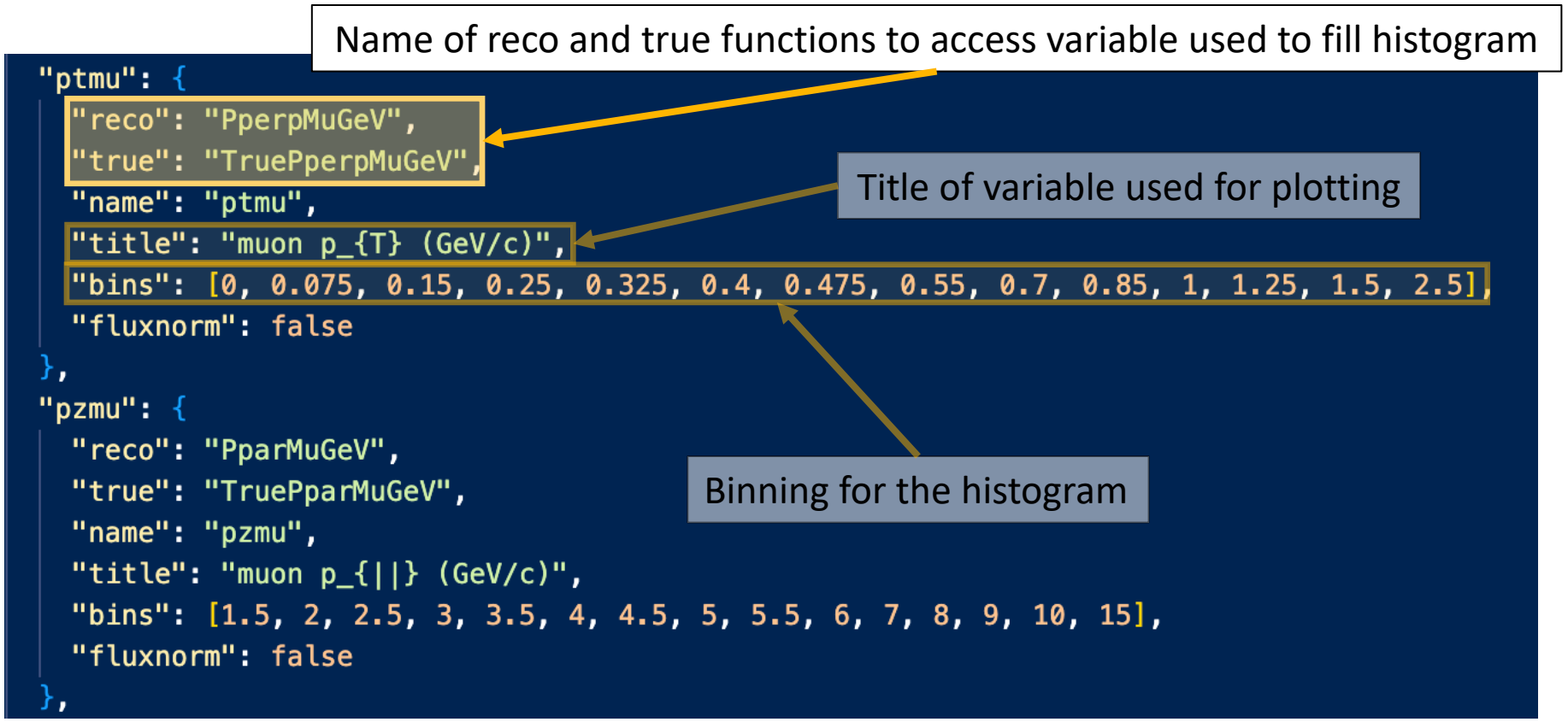
Variable config example


```
"ptmu": {
  "reco": "PperpMuGeV",
  "true": "TruePperpMuGeV",
  "name": "ptmu",
  "title": "muon p_{T} (GeV/c)",
  "bins": [0, 0.075, 0.15, 0.25, 0.325, 0.4, 0.475, 0.55, 0.7, 0.85, 1, 1.25, 1.5, 2.5],
  "fluxnorm": false
},
"pzmu": {
  "reco": "PparMuGeV",
  "true": "TruePparMuGeV",
  "name": "pzmu",
  "title": "muon p_{||} (GeV/c)",
  "bins": [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 15],
  "fluxnorm": false
},
```

Title of variable used for plotting

Binning for the histogram

Variable config example




```
"ptmu": {
  "reco": "PperpMuGeV",
  "true": "TruePperpMuGeV",
  "name": "ptmu",
  "title": "muon p_{T} (GeV/c)",
  "bins": [0, 0.075, 0.15, 0.25, 0.325, 0.4, 0.475, 0.55, 0.7, 0.85, 1, 1.25, 1.5, 2.5],
  "fluxnorm": false
},
"pzmu": {
  "reco": "PparMuGeV",
  "true": "TruePparMuGeV",
  "name": "pzmu",
  "title": "muon p_{||} (GeV/c)",
  "bins": [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 15],
  "fluxnorm": false
},
```

Variable config example

Naming to select the variable in a main “driver” config

Name of reco and true functions to access variable used to fill histogram

Title of variable used for plotting

Binning for the histogram

```
"ptmu": {
    "reco": "PperpMuGeV",
    "true": "TruePperpMuGeV",
    "name": "ptmu",
    "title": "muon p_{T} (GeV/c)",
    "bins": [0, 0.075, 0.15, 0.25, 0.325, 0.4, 0.475, 0.55, 0.7, 0.85, 1, 1.25, 1.5, 2.5],
    "fluxnorm": false
},
"pzmu": {
    "reco": "PparMuGeV",
    "true": "TruePparMuGeV",
    "name": "pzmu",
    "title": "muon p_{||} (GeV/c)",
    "bins": [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 15],
    "fluxnorm": false
},
```

Closing

Data Preservation at MINERvA

MINERvA has a unique dataset, robust analysis tools worth preserving

- Efforts ongoing to further develop MAT
- Early production of MADTuples availabe for public use (reach out if interested)
- Further validation, final production of MADTuples coming soon
- Continued analysis and reanalysis of MINERvA data made possible through these efforts

Final shutdown of MINERvA

Celebrating 30e20 POT

Acknowledgements

Backup

Hopefully something will answer your question here

Handling `MnvHND`'s

- `BaseUniverse/MinervaUniverse/CVUniverse`
  - Defines functions that for the central value universe
  - Interfaces with branches in tuples to calculate values for events to fill histograms
- `HistWrapper`
  - Allows for easier user access to build `MnvHND`'s
- `VariableBase` class
  - How users define binning, reco/truth functions (from CVUniverse) for variables of interest
  - Plotting specifications (e.g. axis labels, titles, units)
- `Cut/Cutter`
  - Cut contains cut values of functions (from CVUniverse) specified by user
  - Cutter applies these cuts to events

Higher Dimensions

- Full support for `MnvH1D` & `MnvH2D`, partial support for `MnvH3D` (no unfolding)
- `HyperDimLinearizer` tool projects higher dimensional histograms to 1D or 2D
  - Created for 3D *QElike neutrino* result, further for 3D *QElike antineutrino* result currently under development
  - Allows 3D (or higher) analyses using tools for 1D and 2D (at least for D'Agostini unfolding)
  - Memory hungry, though work to mitigate this is underway

Validation

Validating new tuples is an ongoing process

- Some branches have changed from older legacy analysis tuples
- Compare to tuples used in recent publications
- Ensure new methods can be used to produce similar results well within uncertainty
- Check backwards compatibility to older analyses
  - Make old branches accessible or replace with a suitable, easy to implement alternative