

Proposal for New ND-GAr Analysis Scheme

Chris Hilgenberg
ND-GAr Meeting
22 February 2021



UNIVERSITY OF MINNESOTA



Content

- Reminder: Motivation & requirements
- Propose implementation for new analysis scheme
- Moving forward

Reminder: analysis infrastructure needs

- Starting point: Mike Kordosky's [talk](#)
- Cross section analyses
- ν_μ and ν_e CC selection
- Calibration analyses
- Sim/reco R&D
- GENIE reweighting
- Overlay samples
- Crude visualization
- Portability
- Useable by newbies and useful to experts
 - Framework for simplifying analysis
 - Support for C++ and Python

Proposal

- New repository, [garana](#)
 - Condensed versions of [garsoft](#) products
 - Tools for reading class-based or flat trees
 - Tools for class-based → flat tree conversion, consistency checking
 - Standalone, lightweight event display
 - Backtracking
 - Support for C++ or Python analysis with ROOT
- Make [garsoft](#) depend on [garana](#)
 - Extensions for [garana](#) products to facilitate construction from [nusimdata](#) objects
 - Class-based trees produced by new analyzer module, [StructuredTree_module.cc](#)
 - [garana](#) takes care of the rest

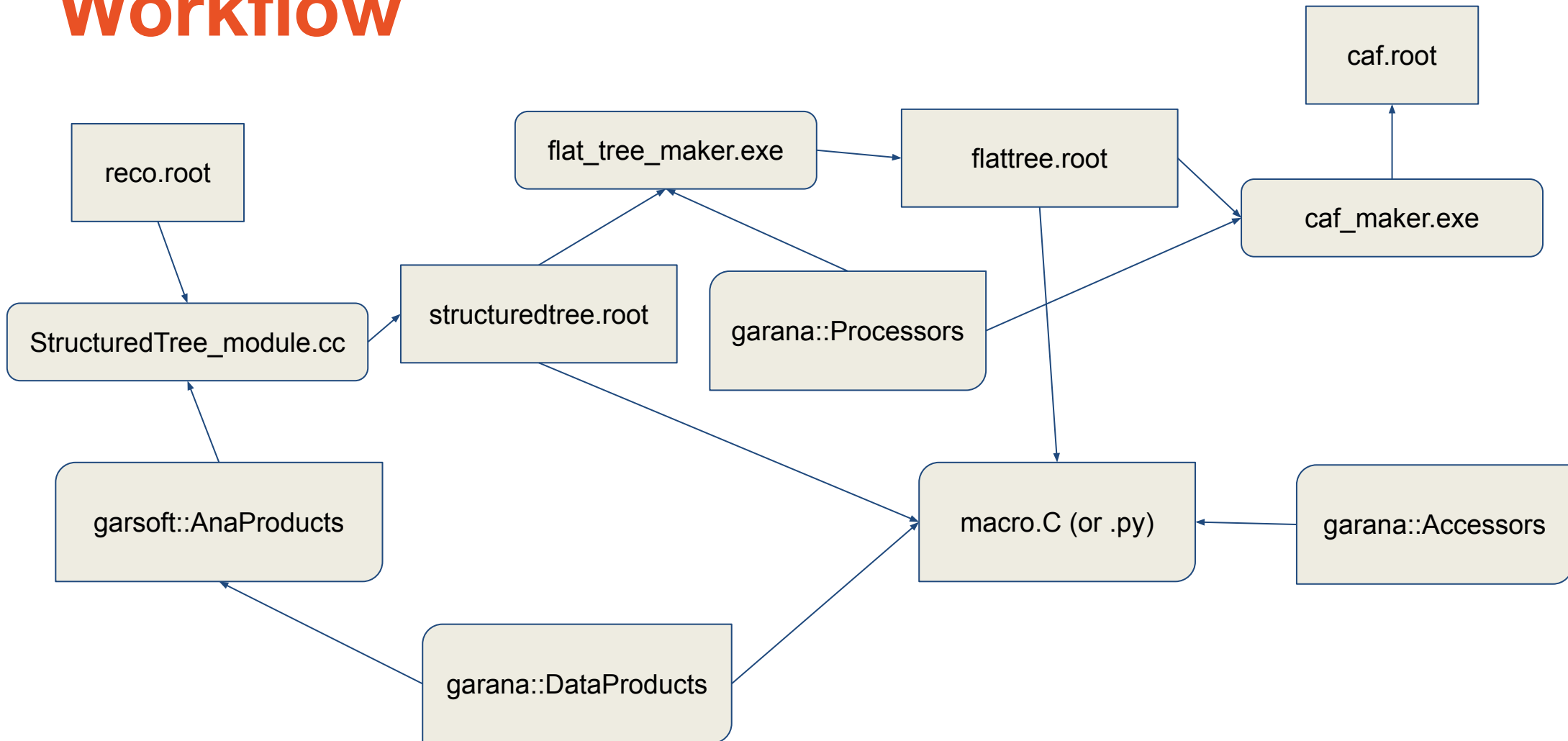
Trees (all configs)

- Construct multiple trees in single file
- Intratree associations via indexing (also considering using TRefs)
- **headerTree** (1 entry per file)
 - Flux component weights
 - Geometry used
 - Run-level info
 - Other relevant config values
- **genTree**
 - Generator-level info
 - Full GHEP record if applicable
 - Final state particle 4-vectors
- **g4Tree**
 - G4-level info
 - MCParticle initial/final 4-vectors
 - MCParticle parentage
- **detTree**
 - Readout sim info, e.g. RawDigits
- **recoTree**
 - Info from final reco pass
 - TPC Track, Vertex ECal (or MuID) clusters
- **displayTree**
 - Info for event display
 - Sparsified trajectory points and reco hits

Configurability

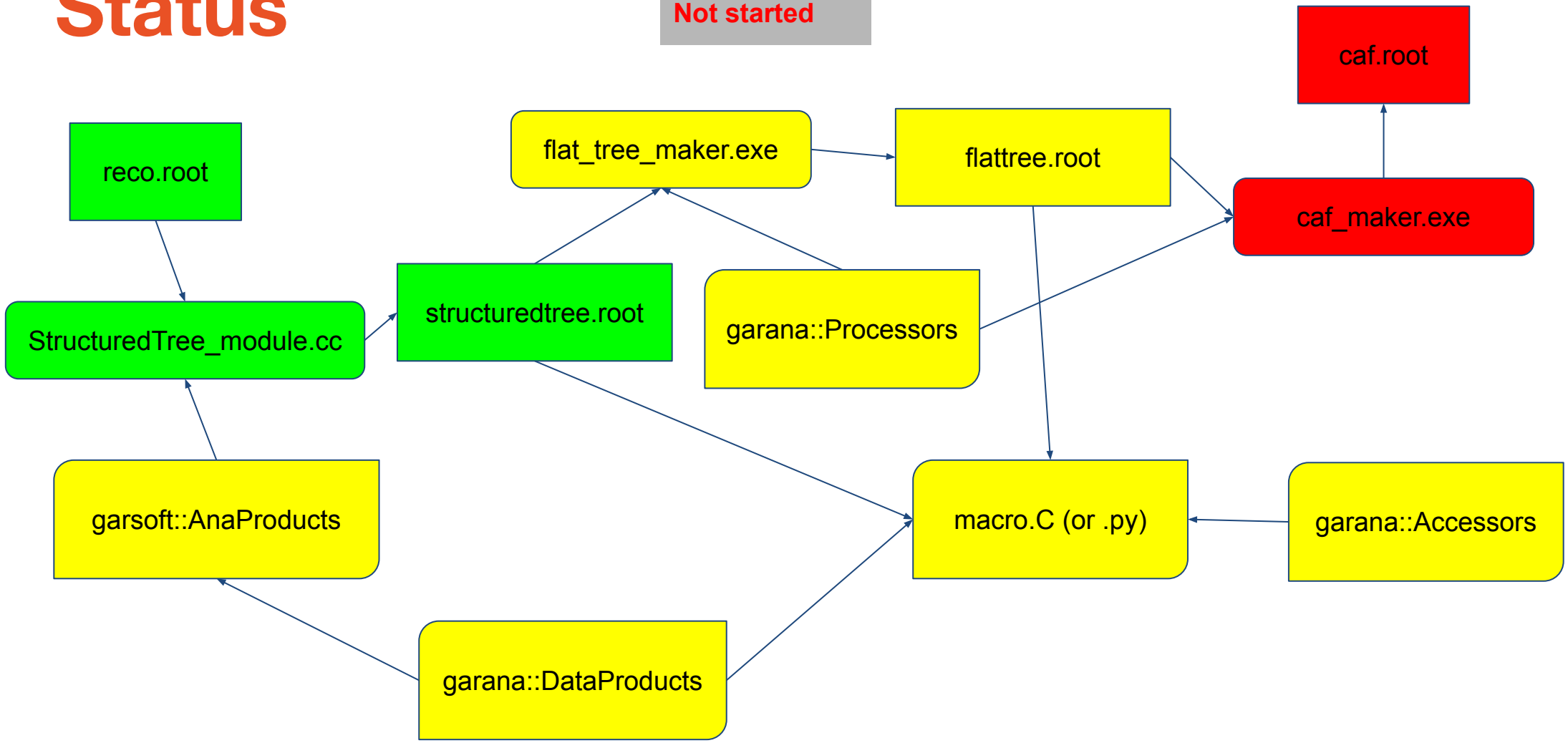
- Avoid too much configurability
- Offer a few configurations by task
 - General (default)
 - Event selection
 - Reco performance (resolution, efficiency)
 - Cross section analysis
 - Reco development (largest files)
 - Need access to each reconstruction step
 - TPC ↔ ECal matching
 - Algorithm/fitter development
 - Track info for each track fitting stage
 - Readout simulation development
 - Gen/G4 level info
 - Raw digits from each subsystem
- displayTree turned on or off
- Highly customized configurations can still be used in [anatree_module.cc](#)

Workflow



Status

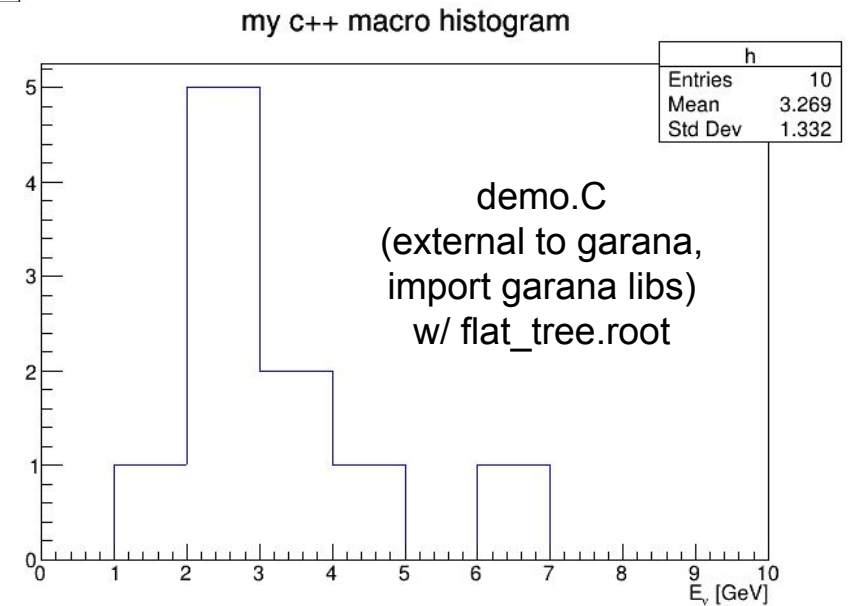
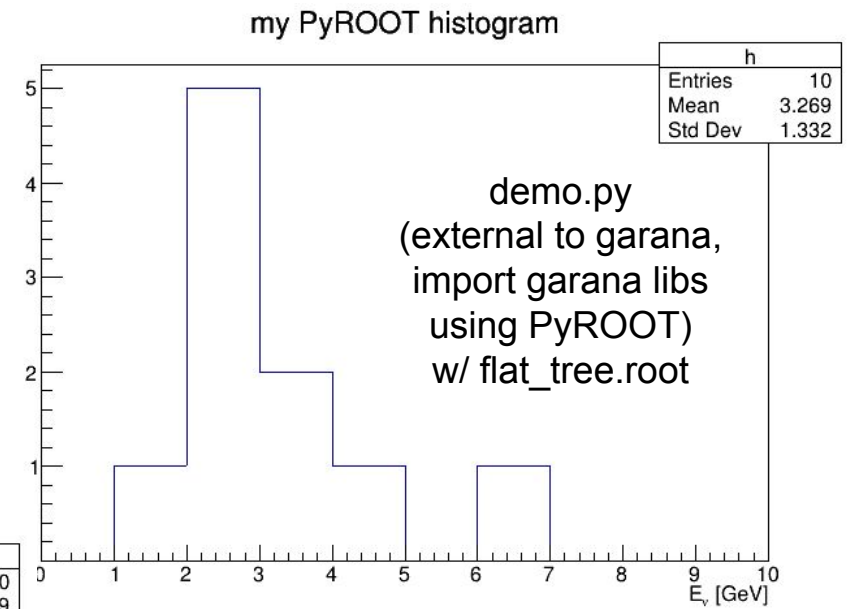
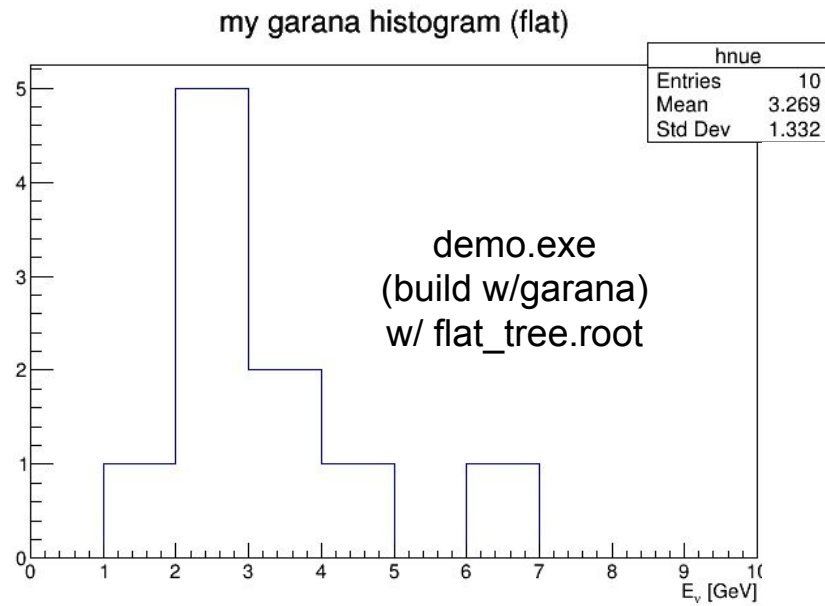
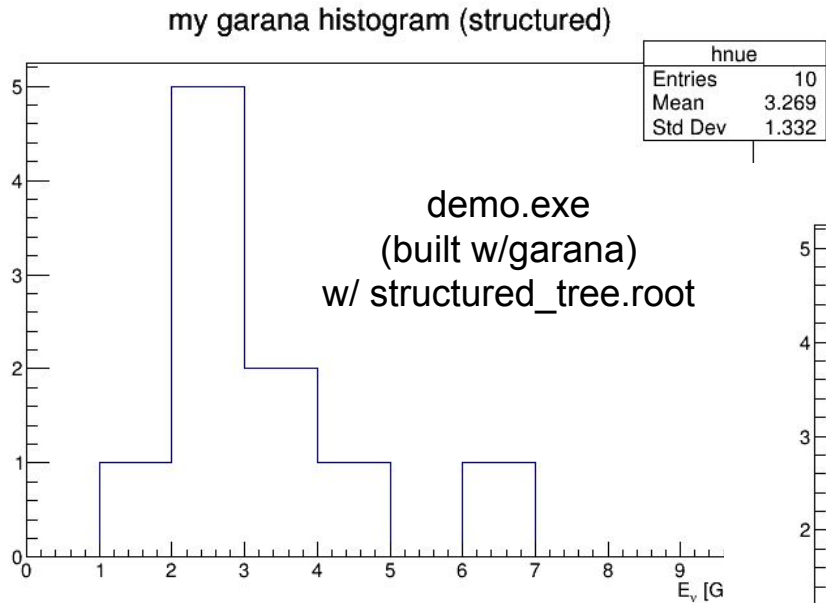
Done
Partially done
Not started



Portability

- Analysis output file size
 - Use same file for input to `anmtree_module` and `StructuredTree_module`
 - 1000 event GENIE sample (1 nu/event)
 - `anmtree`
 - default configuration: 160 kB/event
 - all write options enabled: 360 kB/event
 - `garana` w/default configuration
 - structured: 60 kB/event
 - flat: 3 kB/event
- Repository size < 200 kB, build size < 3 MB
- CMake based install (tested so far only on Ubuntu 20.04 and SLF 7.9)
- Dependencies
 - ROOT (v6_22_00 +)
 - Python 3
 - C++17

Demo: consistency check



Open question: CAF

- CAFAna used for oscillation analyses (e.g. TDR, ND CDR)
- A CAF file maker exists in garsoft
 - Takes anatree.root as input
 - Outputs CAF-ready flat ntuple
- CAF files input to CAFAna, outside garsoft
- We can add more info in CAF files at cost of longer processing time in CAFAna
 - Makes writing general purpose CAF files difficult
 - Tanaz is working on including xsec systematics, nu_mu and nu_e selections, full instead of parameterized reco info in CAFMaker module

Next steps

- **garana** repo currently at: <https://github.com/chilge/garana>
- **garsoft** changes on branch **feature/chilgenb_NewAnaScheme**
- If the group likes the **garana** solution as proposed
 - **garana** builds on my local machine and dunegpvm's (add to UPS eventually)
 - Migration of **garana** → **Redmine** (cdcvcs) or **garsoft** → **Github**
 - **LArSoft**, SBN code have migrated to **GitHub**
 - **GitHub** has a lot of nice features w.r.t. **Redmine**
 - Write useful documentation
 - Import existing analysis tools already developed inside **garsoft** into **garana**
 - E.g. Tanaz's event selection tools

BEGIN BACKUP SLIDES

Rewighting

- MC truth at the interaction level and genie final state level. Neutrino type, 4-vector, target, process type, kinematics (q^2 , etc) neutrino vertex, track PIDs and 4-vectors for final state particles
- May need full GENIE record
 - Looking into making a minimal list of GENIE params that could facilitate reweighting
 - A similar problem has been addressed in MicroBooNE
- In the overlay case, what samples should be made reweightable?
 - Of course, reweight interactions in TPC
 - ECal - probably
 - “Dirt” background - probably not

Visualization

- Need MCTParticle trajectory points and reco clusters at a minimum
 - For crude visualization, we do not need all of these points
 - One option could be to “downsample” in some way that depends on the particle momentum, particle type, and range
- Could use simple 2D/3D histograms for visualizing particle trajectories or comparing truth and reco objects
- Could move beyond histograms and employ a minimal standalone TEve based event display
 - Could be a nice tool for newbies or people who don't want or need to learn GArSoft