Dual Tuples in Pass8

Alex Drlica-Wagner

June 14, 2012

SLAC KIPAC STANFORD UNIVERSITY

Motivation

- **The MeritTuple is large and unwieldy**
  - **During development, it's useful to have a sandbox to play in**
  - **It's difficult to remove variables after the fact**
  - **Documentation is incomplete at best (incorrect at worst)**
- **The goal of the PrimeTuple is to begin cleaning the MeritTuple without destroying the sandbox**
  - **"Important" variables are promoted to the PrimeTuple**
  - **When development is complete, the PrimeTuple will replace (and be renamed) the MeritTuple**
  - **In the meantime, the PrimeTuple should be a subset of the MeritTuple***
- **Dual tuples are a transient phenomenon!**

***There is a caveat for GR-v20r3p1 which is covered later**

Current Implementation

- **The PrimeTuple is currently implemented through a second instance of NtupleMaker in the Gleam job options.**
- **Currently, a TTree friendship link is created between the MeritTuple and PrimeTuple (*not required*)**

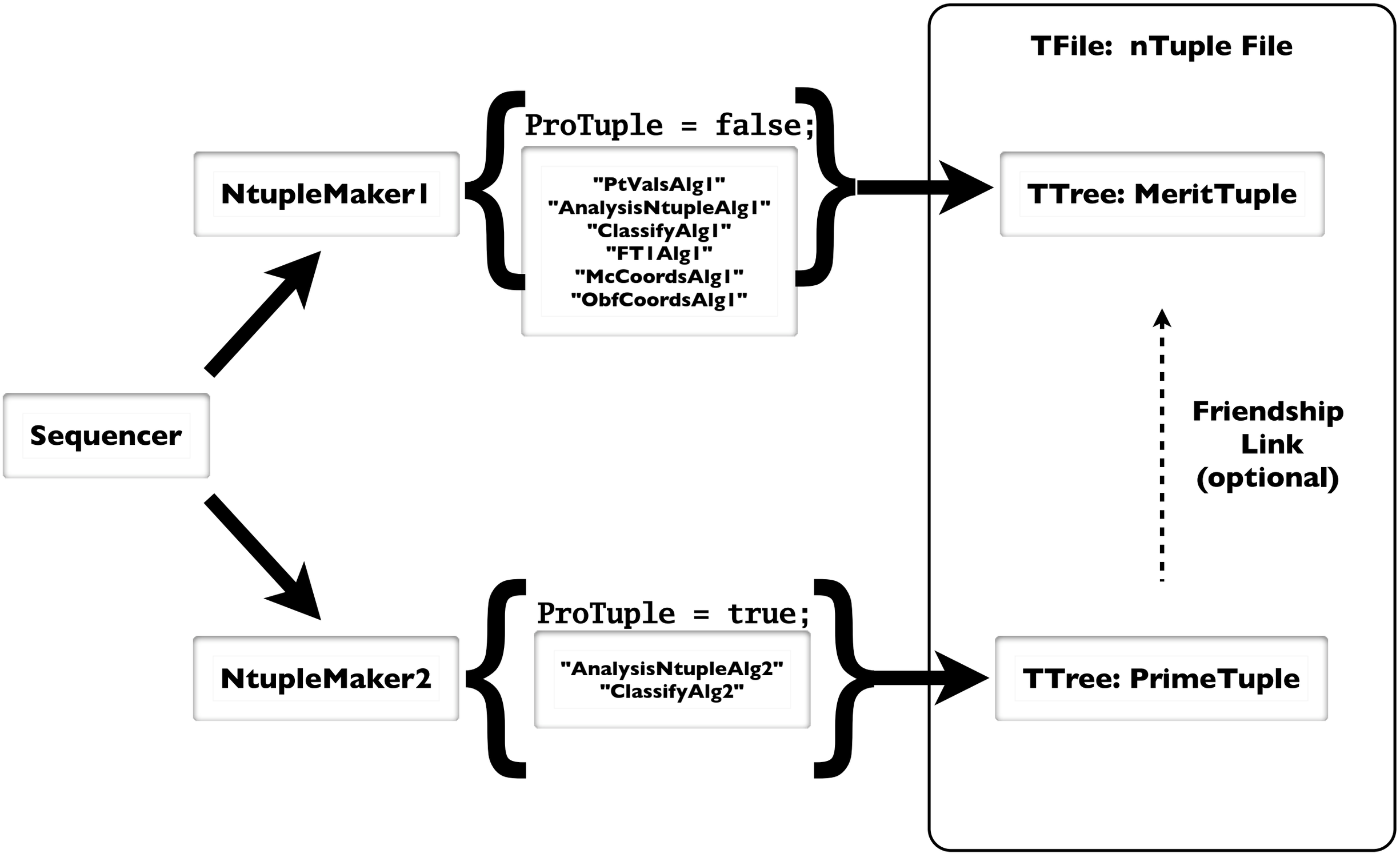
```
/// ADW: Make production tuple named PrimeTuple
NtupleMaker2.Members = {
    "AnalysisNtupleAlg/AnalysisNtupleAlg2",
    "ClassifyAlg/ClassifyAlg2"
 };
AnalysisNtupleAlg2.tupleName = "PrimeTuple";
AnalysisNtupleAlg2.proTuple = true;

ClassifyAlg2.TreeName = "PrimeTuple";
ClassifyAlg2.UseTMine = true;
ClassifyAlg2.TMineCacheFile = "PrimeTuple_cache.root";
ClassifyAlg2.xmlFileName = "$(GLASTCLASSIFYXMLPATH)/Pass8_Analysis_Open.xml";

// default output to GLEAMDATAPATH
RootTupleSvc.filename            = "${GLEAMDATAPATH}/merit.root";
RootTupleSvc.RejectIfBad = false;


/// ADW: Friend the two tuples
RootTupleSvc.treeFriends = {"MeritTuple","PrimeTuple"};
```

Variable Promotion

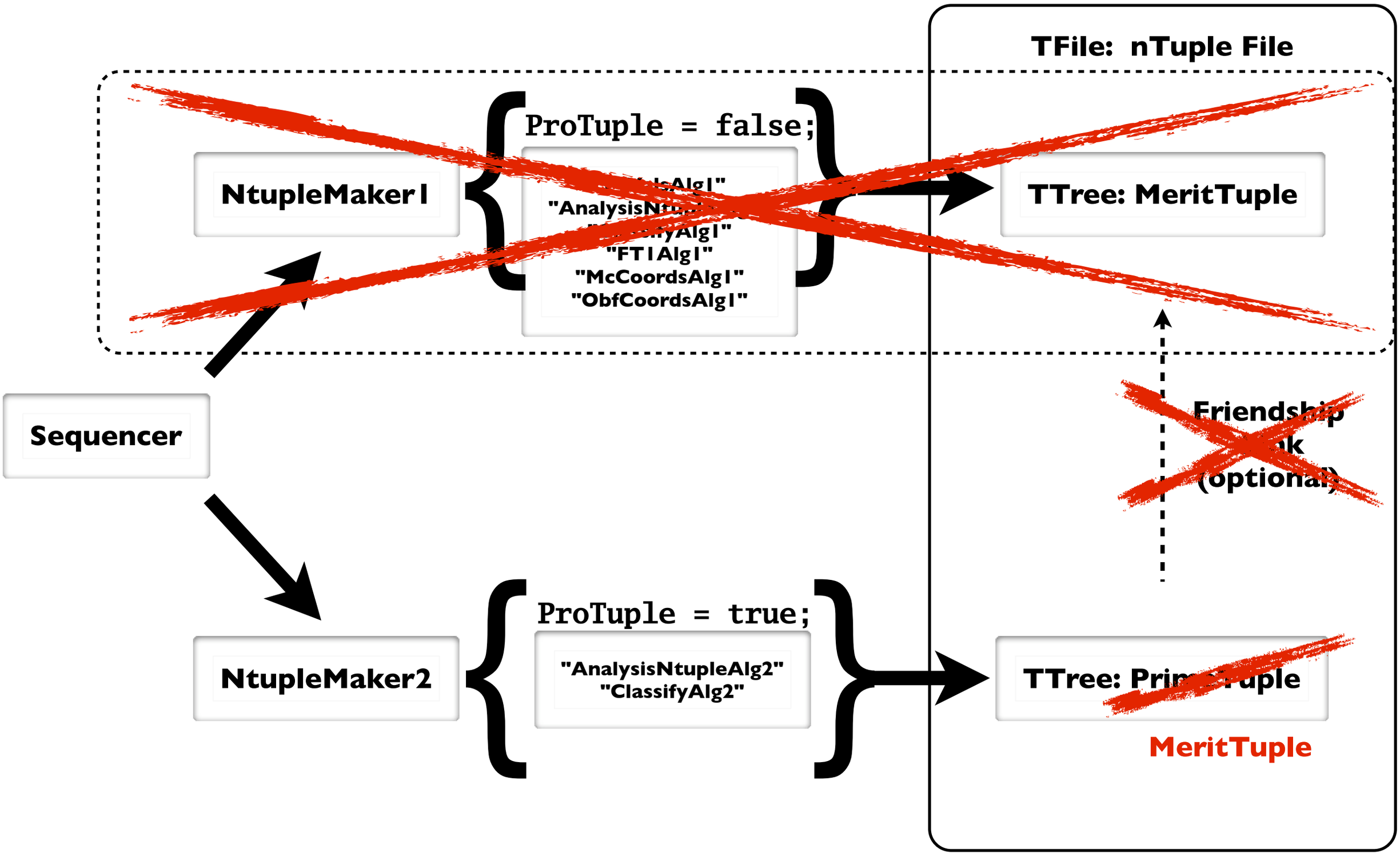
- **Variable promotion is hardcoded into GlastRelease (e.g., AnalysisNtuple and similar packages)**
- **Suggested criteria for promotion:**
  - **Check of Data-MC agreement**
  - **Documentation**
  - **Naming convention**
  - **Consent of the Pass8 group**
- **Example of variable promotion in McValsTool:**

```
addItem("McDirErr",     &MC_dir_err,          ); // Development variable
addItem("McTkr1DirErr"  &MC_TKR1_dir_err, true); // Promoted variable
```

Current Scheme



Production Scheme



Merit Skimming

- By default, the PrimeTuple variables should be a subset of the MeritTuple variables.*
- Thus, skimming the MeritTuple should provide all the information (and more) that is in the PrimeTuple.
- Friendship linking the two tuples in xrootd is causing problems, but friendship linking should be easy to turn off in the job options.
- It should NOT be necessary to change the skimmer.
  - Ensure PrimeTuple is a subset of the MeritTuple
  - No friendship linking in xrootd

*Caveat on the next slide

Caveat in GR-v20r3p1

- **Since the jobOptions for each member of NtupleMaker2 can be altered independently of NtupleMaker1, the PrimeTuple is not required to be a subset of the MeritTuple.**
- **Specifically the case in GR-v20r3p1:**
  - **Keep the old Pass7 worksheet variables around while beginning work on the new Pass8 worksheet**
    - **ClassifyAlg1 was run on the old Pass7 worksheet**
    - **ClassifyAlg2 was run on the new Pass8 worksheet**
- **Variables exist in PrimeTuple that are not in MeritTuple (and vice versa).**
- **Skimming on the MeritTuple will not access all information.**
- **Easily fixed by running the Pass8 worksheet on both tuples.**

Path Forward

- **Proposed fixes for next GR:**
  - Skimming should be fixed by remove friendship linking.
  - Run the MeritTuple on the Pass8 worksheet (lose Pass7 variables, but include all PrimeTuple info)
- **Further down the road:**
  - Continue to produce the PrimeTuple to insure that variable promotion is enforced
  - Keep PrimeTuple in xrootd files but not accessed by skimmer
- **Nearing production:**
  - Turn of the old (development) MeritTuple
  - Rename the PrimeTuple to MeritTuple
  - Produce files with one tuple (name MeritTuple) but with many fewer variables

Foreseeable Issues

- The size of merit will increase before it decreases
  - As a sandbox, the number of variables will continue to grow
  - The PrimeTuple will duplicate info in the MeritTuple
- Creating the large development tuple after transition:
  - Eventually we will be making a smaller production tuple by default
  - If we then want to go back to creating a development tuple, we might have trouble skimming it
  - A simple fix would be to only create a development tuple and name it MeritTuple (though this would probably lead to confusion)
- In Pass8, some recon algs. are writing directly to merit; this could cause problems with dual tuples...

Some of Tom's Questions

A list of some practical issues of interest to this group include:

1) an overview of what is planned with approx schedule (your web page is a good start)

Have merit files contain two trees until the Pass8 tuple is stabilized. The goal is to have this done by next year...?

2) a description of how this is implemented in the ROOT file(s), maybe with a diagram

See slide 5 and 6, but really it is just another TTree in the ROOT files with an optional friend link between the two TTrees.

3) what sorts of new functionality will be needed, e.g., viewing 'new' vs 'old' variables, ability to handle dual-tuples -- at least during a transition phase, etc.

Hopefully no new functionality will be needed. The idea would be that the skimmer would continue to skim only the MeritTuple, which would include all of the information in the PrimeTuple.

4) any new externals dependencies (if any), e.g., a particular version of ROOT

There should be no external dependencies and any version of ROOT (within reason?) should work.

5) how this scheme might evolve between today and the time when Pass8 is released to production

This scheme will hopefully evolve towards the elimination of dual tuples before Pass 8 moves into production.

6) is there a way to make these dual-ntuple MERIT files behave like existing MERIT files?

This is certainly the goal and I believe it will be the case as long as the friendship link is eliminated and that the PrimeTuple is a subset of the MeritTuple.