ATLAS SUSYTools

DAWG Technology and Innovation Survey

K.Yoshihara (Penn), H.AbouZeid (Copenhagen)

February 13th, 2019

Outline

- **What is SUSYTools?**
  - And what it is not!
- **Extra Tools**
  - PRW autoconfigure
  - Cross-section tools (not discussed here)
- **Management Benefits**
  - Modelling feedback
  - Direct link between physics modelling and code development
  - Metadata collection
  - Monitoring

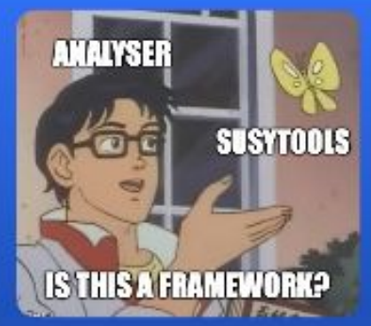
SUSYTools is maintained by the subconvenors of the SUSY Background Forum -- some of the advantages and use cases here are a result of this ‘special’ relationship

About SUSYTools: Fundamental Philosophy

- **Make things simple for the user!**
  - The multiple different ways to grab an electron from a collection is uninteresting -- provide a single, *flexible*, correct way to do it, and allow analyzers to do something else with their time
  - Update and push recent object performance updates to code base w/o manual intervention from groups
    - Most up to date recommendations are the default, groups can 'opt out' with manual intervention
  - Easily retrieve list of systematics which are relevant for an object, apply that variation, and return the result
  - Supplement the framework proper with a tools to 'make life easier' (more on this later)
  - Provide centralized patches and workaround for *upstream* bugs or calibration issues

About SUSYTools: From last time… (link to talk)

ONE APPROACH



- "SUSYTools" originally developed in Run 1 to provide recommended configuration(s) and object preparation logic for SUSY analyses
- Provides:
  - GetObjects() methods that apply all needed corrections and attach selection flags, optimised systematics handling
  - Trigger decision and navigation access
  - Metadata reading
- Analysers free to choose/develop:
  - Event loop, I/O management
  - Analysis steering
- Associated toolkits for querying database information (progressively being centralised)

T.J. Khoo

About SUSYTools

- Who uses it?
  - Most SUSY searches on ATLAS use SUSYTools
    - Some exceptions for RPV/LL searches, due to special signatures/reconstruction needs
  - **Outside of SUSY:**
    - MET performance groups
    - Select analysis within ATLAS Exotics subgroup

- The Basics:
  - SUSYTools is ingrained as part of the ATLAS Athena project
  - (D)xAODs are the input
  - Almost exclusively C++, but with python helpers, wrappers and *Job Options* (for integration with Athena)
  - Dual use: Accessible from Athena, or the EventLoop
    - Just to say: ST shouldn't care which parent ATLAS framework is calling it
  - At the most basic -- provides an interface to:
    - GetObjects (and calibrate/correct/reject them)
    - Common Metadata (cross-sections, etc.)
    - Apply proper scale factors, and easily iterate through uncertainties
    - Apply algorithmic corrections for upstream problems

About SUSYTools

- Configuration:
  - Plain text configuration file
  - Defines:
    - Object definitions and working points
    - Overlap strategies
    - Special corrections turned on/off
  - Defaults are defined based on studies shown in the background forum

SUSYTools_Default.conf 3.78 KB

```
1  ##################################################
2  # SUSYTools configuration file
3  ##################################################
4  EleBaseline.Pt: 10000.
5  EleBaseline.Eta: 2.47
6  EleBaseline.Id: LooseAndBLayerLLH
7  EleBaseline.CrackVeto: false
8  EleBaseline.z0: 0.5
```

- Optimizations and parallelism:
  - Generally outside the purview of SUSYTools (although simple scalar operations are optimized)
  - Parallelism expressed at the job level

- What SUSYTools is NOT:
  - A full fledged framework for statistics, ML, plotting, etc.
    - It leaves the rest of up to the analyses
  - Define output:
    - Output can be histograms, tuples, pngs, or just a really huge text file

Example Use

Initialization (other settings controlled by configuration file)

```
//--- ST config and retrieval
ATH_CHECK( m_SUSYTools.setProperty("DataSource",m_dataSource) );
ATH_CHECK(m_SUSYTools.setProperty("PRWLumiCalcFiles", m_PRWLumiCalcFiles) );
if (m_usePRWAutoconfig){
  ATH_CHECK(m_SUSYTools.setProperty("AutoconfigurePRWTool", true) );
  ATH_CHECK(m_SUSYTools.setProperty("mcCampaign", "mc16a") );
}else{
  ATH_CHECK(m_SUSYTools.setProperty("PRWConfigFiles", m_PRWConfigs) );
}
```

Get Objects, decoration of SFs (electrons)

```
xAOD::ElectronContainer* electrons_nominal(0);
xAOD::ShallowAuxContainer* electrons_nominal_aux(0);
ATH_CHECK( m_SUSYTools->GetElectrons(electrons_nominal, electrons_nominal_aux) );
ATH_MSG_DEBUG( "Number of electrons: " << electrons_nominal->size() );
```

Get Missing Energy

```
ATH_CHECK( m_SUSYTools->GetMET(*metcst_nominal,
                               jets_nominal,
                               electrons_nominal,
                               muons_nominal,
                               photons_nominal, 0, false, false) );
```

Object Overlap Removal

```
//--- Overlap Removal
ATH_CHECK( m_SUSYTools->OverlapRemoval(electrons_nominal, muons_nominal, jets_nominal, 0, taus_nominal) );
```

Loop over systematics (making ‘shallow’ copies)

```
for (const auto& sysInfo : sysInfoList) {
  CP::SystematicSet sys = sysInfo.systset;
  if (m_SUSYTools->applySystematicVariation(sys) != CP::SystematicCode::Ok) {
```

SUSYTools handles all of the interface with all of the (many!) performance packages which are needed for:

- Particle ID
- Calibration
- Scale factors
- Isolation

Systematics:

- Give SUSYTools a full list of systematics: ST will determine which systematics are relevant for each object/SF

Example Use

Initialization (other settings controlled by configuration file)

```
//--- ST config and retrieval
ATH_CHECK( m_SUSYTools.setProperty("DataSource",m_dataSource) );
ATH_CHECK(m_SUSYTools.setProperty("PRWLumiCalcFiles", m_PRWLumiCalcFiles) );
if (m_usePRWAutoconfig){
  ATH_CHECK(m_SUSYTools.setProperty("AutoconfigurePRWTool", true) );
  ATH_CHECK(m_SUSYTools.setProperty("mcCampaign", "mc16a") );
}else{
  ATH_CHECK(m_SUSYTools.setProperty("PRWConfigFiles", m_PRWConfigs) );
}
```

SUSYTools handles all of the interface with all of the (many!) performance packages which are needed for:

- Particle ID
- Calibration
- ...tors

Get Objects, dec...

```
xAOD::ElectronConta
xAOD::ShallowAuxCon
ATH_CHECK( m_SUSYTo
ATH_MSG_DEBUG( "Num
```

...ools a full list of
... ST will
...hich
... are relevant for
...SF

Get Missing Ene...

```
ATH_CHECK( m_SUSYTools
```

SUSYTools monitoring algorithms are stored in the same gitlab repository

Allows new groups/users to use these as a skeleton when making their first SUSYTools analysis

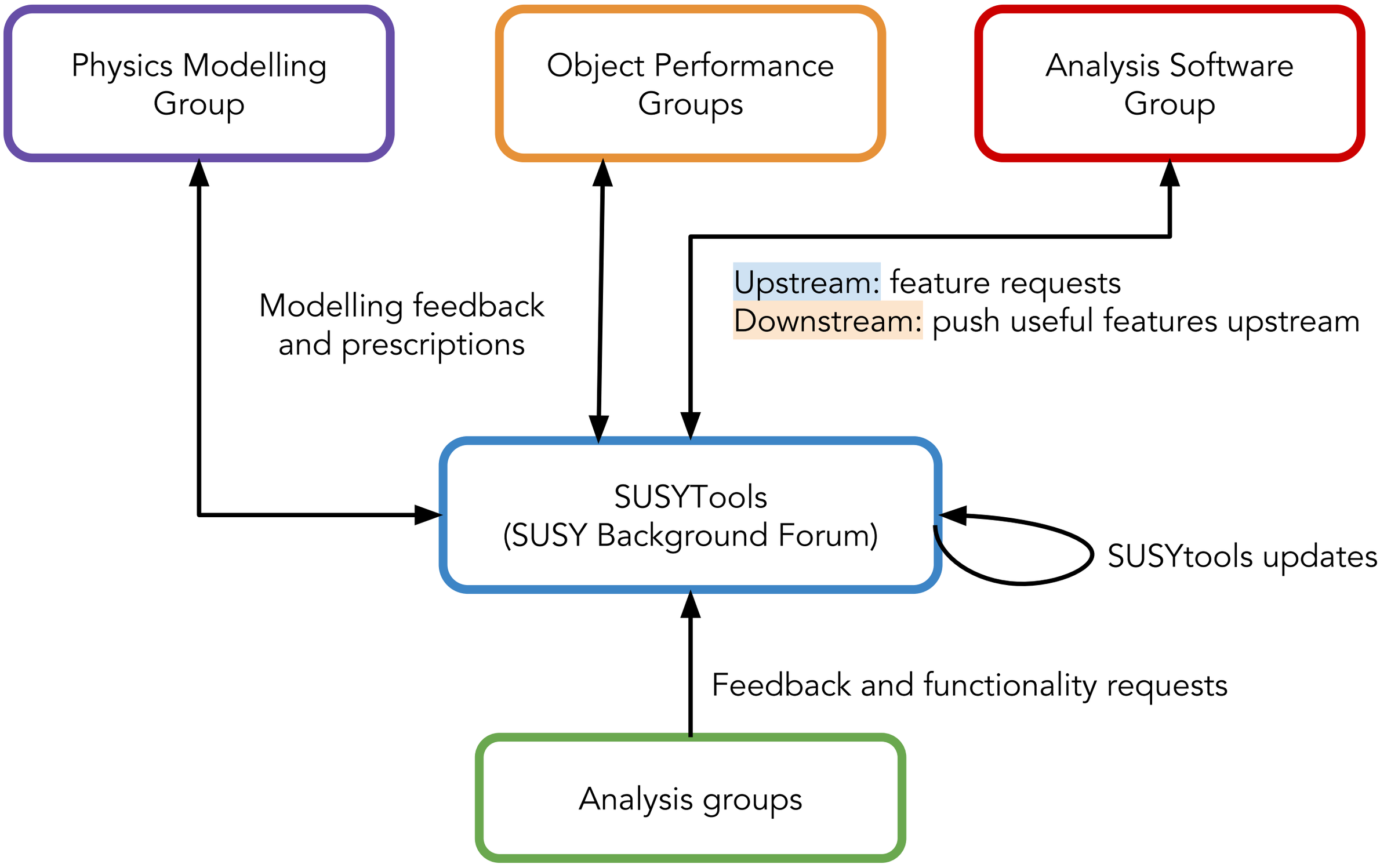
Object Overlap R...

```
//--- Overlap Removal
ATH_CHECK( m_SUSYTools->OverlapRemoval(electrons_nominal, muons_nominal, jets_nominal, 0, taus_nominal) );
```

Loop over systematics (making 'shallow' copies)

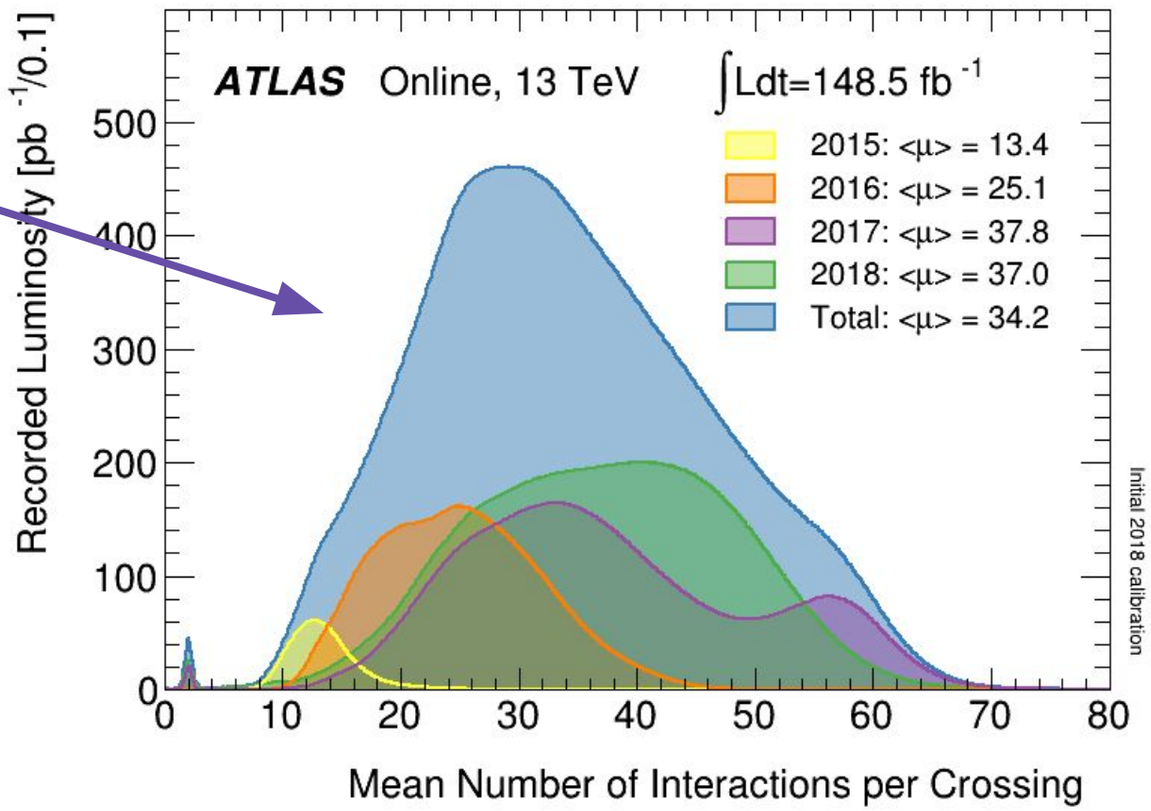
```
for (const auto& sysInfo : sysInfoList) {
  CP::SystematicSet sys = sysInfo.systset;
  if (m_SUSYTools->applySystematicVariation(sys) != CP::SystematicCode::Ok) {
```

Group and Framework Structure



Helpers: Autoconf of Pile-up Reweighting

- Need to reweight the MC samples in order to exactly match the data period's profile



- **Ordinarily:**
  - User's download locally the pileup profile for all MC data-sets used (or make them, if they do not exist)
  - User jobs must correctly pick the correct profile based on the MC sample, and the type of MC generation
  - In case of regeneration, new pileup profiles need to be generated, downloaded, etc.

- **In SUSYTools:**
  - All existing pile-up profiles are downloaded automatically every night
  - Stored in a GRID accessible storage location
  - If configured to be used, SUSYTools automatically selects the correct profile based on the MC input and data period

- This is not a SUSY specific operation -- making efforts to port this tool upstream

Management Benefits

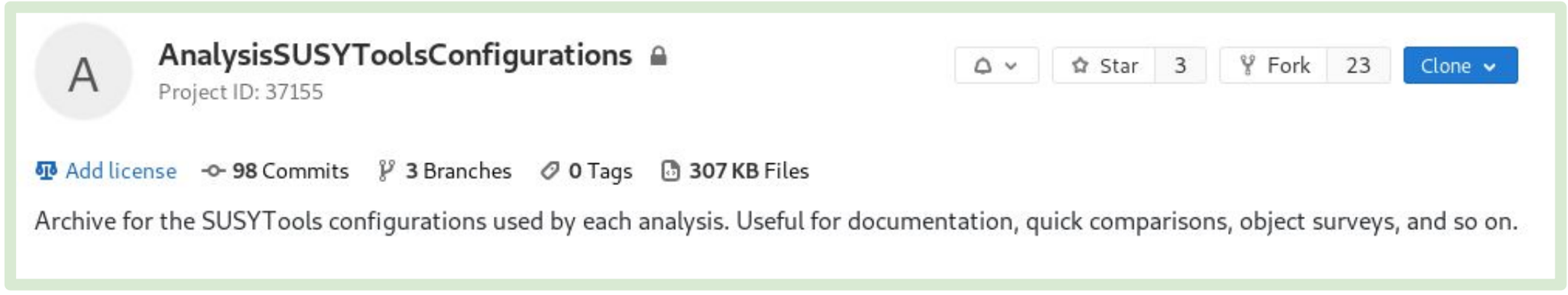
- **Direct link between physics modelling and code development**
  - Issues seen in modelling can immediately be addressed in the code as necessary -- no intermediary, and upstream meetings to be held
  - Updates to code (including upstream changes) are discussed in the same venue as general modelling, and physics background studies
    - More people know what the code is doing, how it is being updated -- there isn't one person in an analysis group who is 'in charge' or the code, managing dependencies, and giving updates to the analysis group

- **Modelling feedback**
  - Easy switches can be advertised in order to get other groups to collaborate on the use of new algorithms/taggers/WPs and possible workarounds and hot fixes

- **Metadata collection** (see next slide)

- **Monitoring** (see next slides)

Collection of Meta-data



- Consistent configuration method across **all** analyses which use SUSYTools
  - Configuration is detailed enough to have a snapshot of the entire analysis
- Once analysis are far enough along, they are required to upload their ST configurations to gitlab

- Having all analysis configurations in one place allows for:
  - Easy overlap checks for combinations
  - Check for use of outdated or conflicting configurations
  - Summary of WPs/configurations used by SUSY group as a whole, allowing for easy upstream feedback on use, preferences, and performance

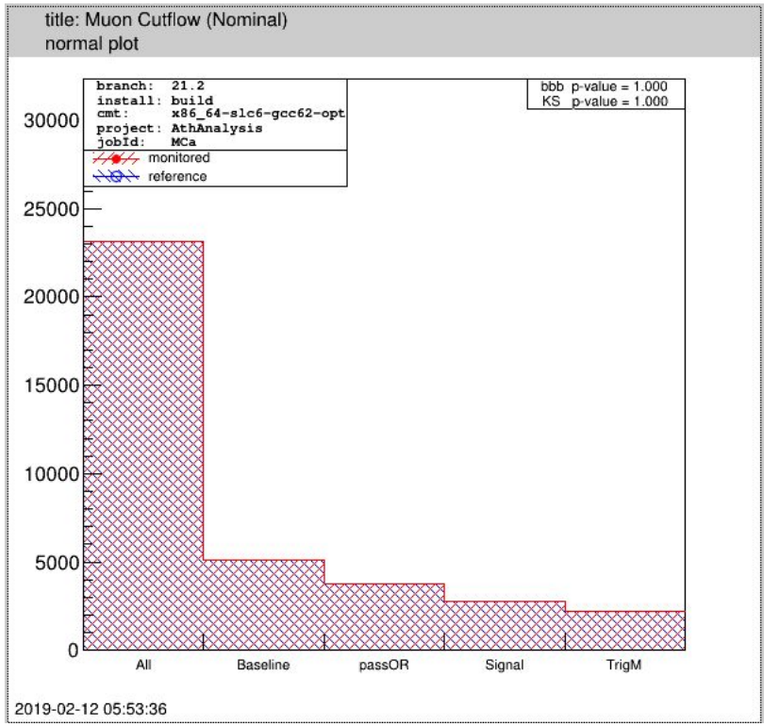
	Default	Analysis 1	Analysis 2
EleBaseline.Pt	10000.	10000.	7000.
EleBaseline.Eta	2.47	2.47	2.47
EleBaseline.Id	LooseAndBLayerLLH	LooseAndBLayerLLH	VeryLooseLLH
EleBaseline.CrackVeto	false	true	false
EleBaseline.z0	0.5	0.5	
Ele.Et	25000.	10000.	20000.
Ele.Eta	2.47	2.47	2.47
Ele.CrackVeto	false	true	false
Ele.Iso	Gradient	FCTight	Gradient

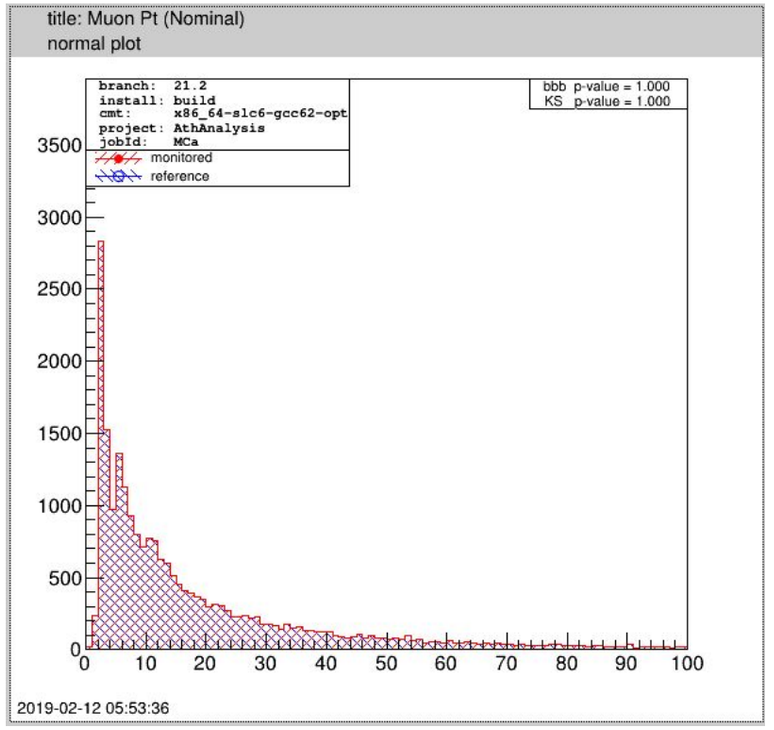
Key	Colour
same as default	
different to default, but same as at least two other configurations	
different to default, but same as at least one other configuration	
different to default and all other analysis configurations	

Monitoring

- Many analysis groups have their own *continuous integration* tests running for each new MR for their local group's code
  - Helpful for checking internal consistency
  - *Misses* upstream changes
- Every night: checks the consistency of upstream changes -- also looks out for bugs, variables disappearing, etc.
  - Run over all input types (all MC periods, datatypes and skimmed data types)

- Future plans: Can run over all group's configurations (as collected in gitlab project on previous slide)





Reference histogram updated automatically every night

Pipeline/Continuous integration

Discussion in BGF about need for new feature

Merge Request from users or background forum convenors

CI Result SUCCESS

	AnalysisBase	AnalysisTop	AthAnalysis	AthDerivation
externals	✅	✅	✅	✅
cmake	✅	✅	✅	✅
make	✅	✅	✅	✅
required tests	✅	✅	✅	✅
optional tests	✅	✅	✅	✅

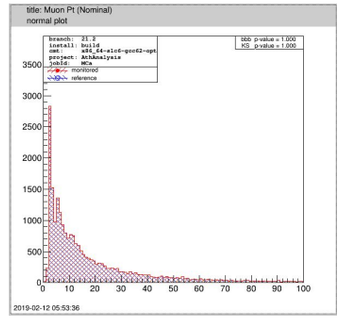
Pipeline checks:

- All combination of possible WPs can be initialized
- No change in dummy cutflow for all data-input types
- No change in cutflow for all configuration options
- [Note: very low stats!]

Continuous integration

MR accepted

Detailed monitoring results available next day



Summary

- **Potential Drawbacks**
  - Yet ‘another’ black box for some users
  - Optimized for ‘working’ analyses
    - Not set up for testing in parallel an arbitrary number of working points
    - Some R&D type checks which need more detailed information are not supported ‘out of the box’
      - Where reasonably achievable, these are implemented on a case by case basis in ‘hidden’ configuration options

- Otherwise:
  - SUSYTools otherwise is a robust and mature ‘framework’ aimed at helping user’s with common tasks needed for analyses
  - It is **not** a full or complete framework in the traditional sense
  - Focuses on ‘what makes life easier’
  - Integration with SUSY Background forum is a strong advantage
    - Allows for more direct feedback on codebase
    - Discussions on modelling and supporting infrastructure occur in the same meeting


BACKUP
BUDDY