Generic Track-to-Cluster Matching in Hps-Java

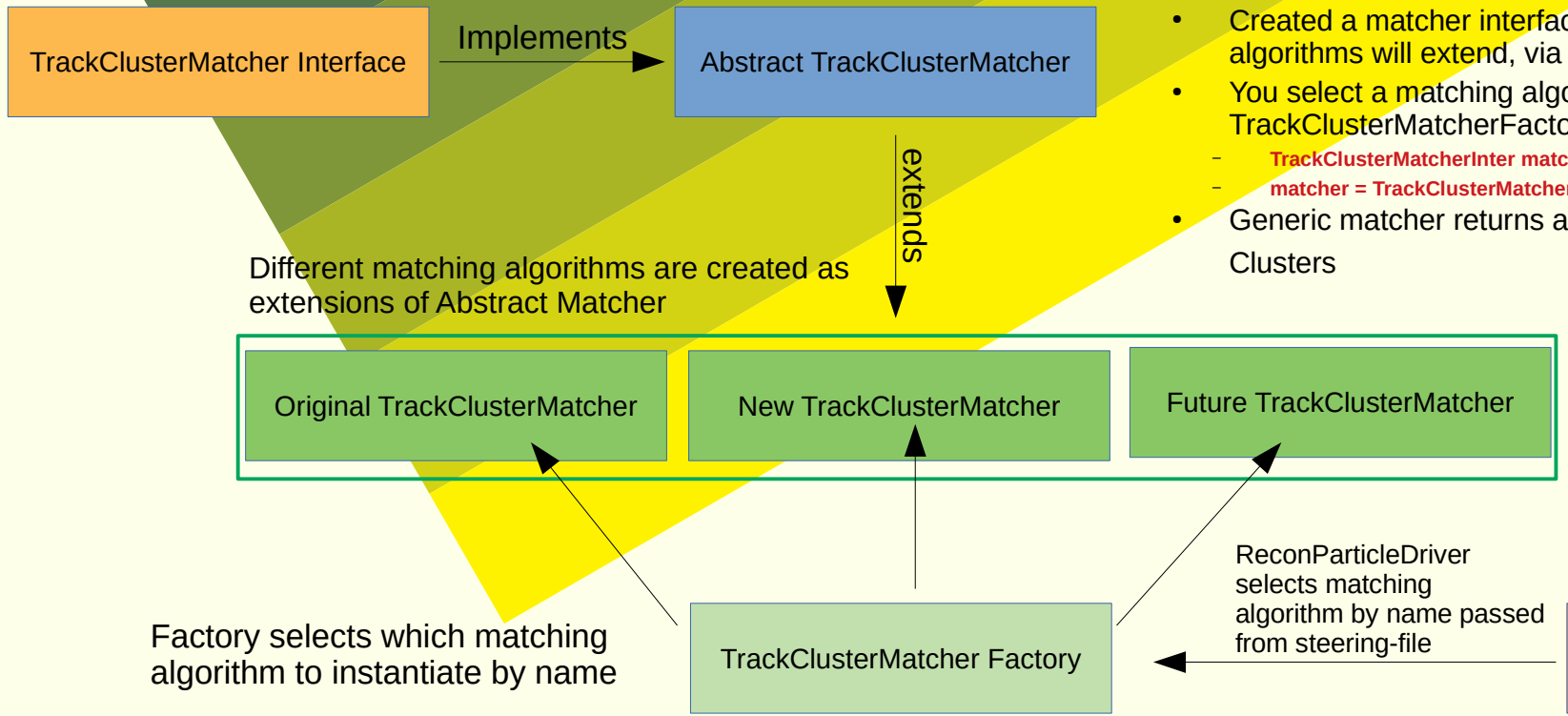
Alic Spellman

02/16/2021

Intro

- Need to implement the new track-to-cluster matching algorithm (that supports both GBL and KF Tracks) in ReconParticleDriver...
- Current matcher implementation is specific to original GBL algorithm and **does not support choosing different algorithm classes**
- To support the ability to use a variety of matching algorithms in reconstruction, I have…
  - created a generic TrackClusterMatcher Interface class that all matching algorithms extend
  - created a TrackClusterMatcher Factory that allows users to select matching algorithm by name
  - edited matcher implementation in ReconParticleDriver to be as generic as possible, **supporting selection of different matching algorithms by a simple steering-file setting**
- I will show the general organization of the new TrackClusterMatcher Interface
- I will show that these updates have not changed the output of hps-java via hps-java Tests

Generic Track-to-Cluster Matching Implementation



- Created a matcher interface class from which all matching algorithms will extend, via an Abstract matcher class
- You select a matching algorithm through the TrackClusterMatcherFactory
  - `TrackClusterMatcherInter matcher;`
  - `matcher = TrackClusterMatcherFactory.create(<algorithm_name>);`
- Generic matcher returns a map of Tracks to matched Clusters

Integration Testing New Matcher Implementation

- **Ran full suite of hps-java tests** to make sure new implementation does not change reconstruction results as compared to Master branch
  - Updated my branch and ran “mvn clean install” to perform all tests
  - Ran these same tests for the current (as of 02/15/21) master branch, for comparison
- Found 6 failures in testing my branch, **but also found 6 failures when testing the master branch itself**
- Perhaps my errors are not indicative of integration issue, as we see them in the master as well…so let’s investigate to make sure

Master

```
<?xml version="1.0" encoding="UTF-8"?>
<failsafe-summary result="255" timeout="false">
  <completed>18</completed>
  <errors>0</errors>
  <failures>6</failures>
  <skipped>0</skipped>
  <failureMessage/>
</failsafe-summary>
```

My Branch

```
<?xml version="1.0" encoding="UTF-8"?>
<failsafe-summary result="255" timeout="false">
  <completed>18</completed>
  <errors>0</errors>
  <failures>6</failures>
  <skipped>0</skipped>
  <failureMessage/>
</failsafe-summary>
```

Integration Testing New Matcher Implementation

- Running hps-java tests for both Master and my branch show the same set of "failures"
  - PhysRun2016V0ReconTest
    - Expected 4892, but was 4884
    - Expected 4892, but was 4884
  - PhysRun2016FeeReconTest
    - Expected 124, but was 115
    - Expected 124, but was 115
  - PhysRun2016MollerReconTest
    - Expected 2223, but was 2126
    - Expected 2223, but was 2126
  - EngRun2015V0ReconTest
    - Expected 1950, but was 1936
    - Expected 1950, but was 1936
  - EngRun2015FeeReconTest
    - Expected 1.278 but was 1.2612
    - Expected 1.278, but was 1.2612
  - EngRun2015MollerReconTest
    - Expected 417, but was 384
    - Expected 417, but was 384

Master

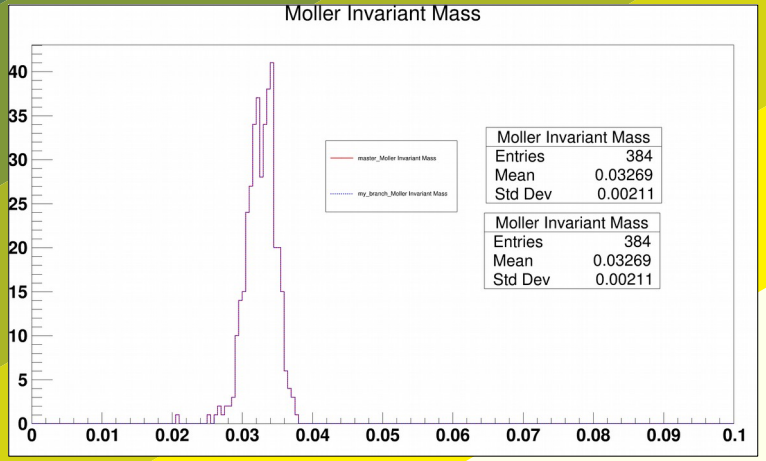
```
TEST-org.hps.test.it.EngRun2015FeeReconTest.xml:    <failure message="expected:&lt;1.2783774159754366&gt; but was:&lt;1.26121912033525778&
gt;" type="junit.framework.AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<1.2783774159754366> but was:<1.
2612191203352578>
TEST-org.hps.test.it.EngRun2015MollerReconTest.xml:    <failure message="expected:&lt;417&gt; but was:&lt;384&gt;" type="junit.framework.
AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<417> but was:<384>
TEST-org.hps.test.it.EngRun2015V0ReconTest.xml:    <failure message="expected:&lt;1950&gt; but was:&lt;1936&gt;" type="junit.framework.As
sertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<1950> but was:<1936>
TEST-org.hps.test.it.PhysRun2016FeeReconTest.xml:    <failure message="expected:&lt;124&gt; but was:&lt;115&gt;" type="junit.framework.As
sertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<124> but was:<115>
TEST-org.hps.test.it.PhysRun2016MollerReconTest.xml:    <failure message="expected:&lt;2223&gt; but was:&lt;2126&gt;" type="junit.framewo
rk.AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<2223> but was:<2126>
TEST-org.hps.test.it.PhysRun2016V0ReconTest.xml:    <failure message="expected:&lt;4892&gt; but was:&lt;4884&gt;" type="junit.framework.A
ssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<4892> but was:<4884>
```

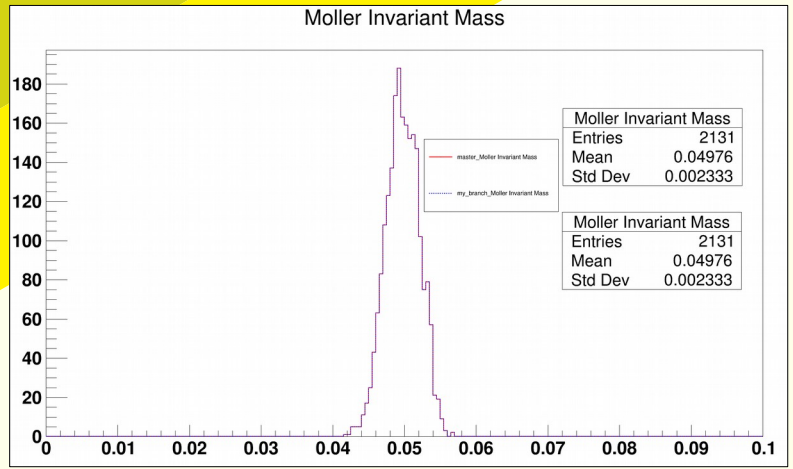
My Branch

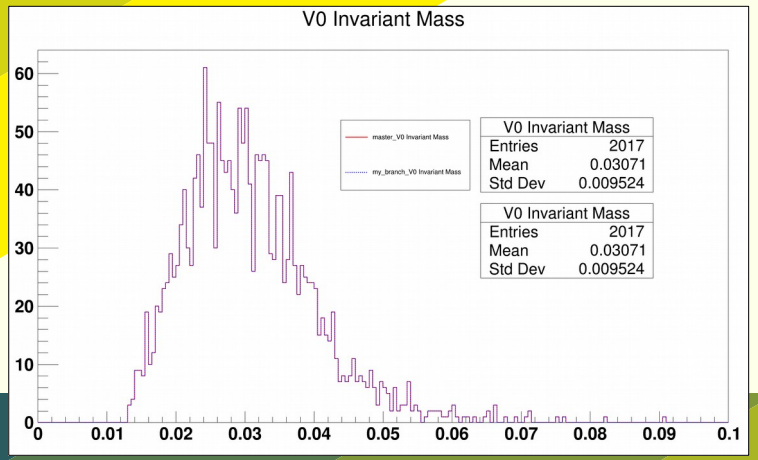
```
TEST-org.hps.test.it.PhysRun2016V0ReconTest.xml:    <failure message="expected:&lt;4892&gt; but was:&lt;4884&gt;" type="junit.framework.A
ssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<4892> but was:<4884>
TEST-org.hps.test.it.EngRun2015V0ReconTest.xml:    <failure message="expected:&lt;1950&gt; but was:&lt;1936&gt;" type="junit.framework.As
sertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<1950> but was:<1936>
TEST-org.hps.test.it.EngRun2015MollerReconTest.xml:    <failure message="expected:&lt;417&gt; but was:&lt;384&gt;" type="junit.framework.
AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<417> but was:<384>
TEST-org.hps.test.it.PhysRun2016MollerReconTest.xml:    <failure message="expected:&lt;2223&gt; but was:&lt;2126&gt;" type="junit.framewo
rk.AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<2223> but was:<2126>
TEST-org.hps.test.it.PhysRun2016FeeReconTest.xml:    <failure message="expected:&lt;124&gt; but was:&lt;115&gt;" type="junit.framework.As
sertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<124> but was:<115>
TEST-org.hps.test.it.EngRun2015FeeReconTest.xml:    <failure message="expected:&lt;1.2783774159754366&gt; but was:&lt;1.26121912033525778&
gt;" type="junit.framework.AssertionFailedError"><![CDATA[junit.framework.AssertionFailedError: expected:<1.2783774159754366> but was:<1.
2612191203352578>
```

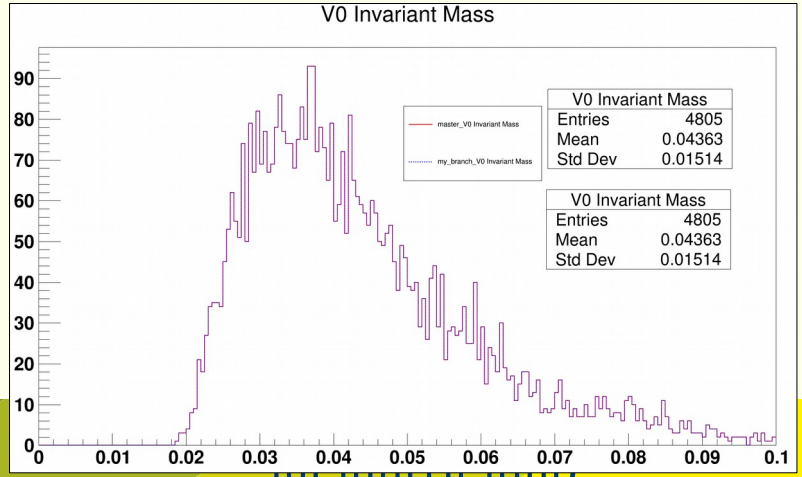
Integration Test Comparisons

- Comparing the integration test output plots
- Plots show that the distributions are exactly the same
- New implementation of Track Cluster matching performs as expected
- All comparison plots stored here: /nfs/slac/g/hps3/users/alspellm/projects/kalman/kf2016mc/testTrackClusterMatching/pass1DevFix/integration_test_comparisons









Summary

- Hps-Java tests show that this new generic implementation of Track-to-Cluster matching has not changed the reconstruction output in any way
- This change will allow us to move forward using the new KF supported matching algorithm, and any future matching algorithms, in a very user-friendly way
- **I think this is ready to PR**