



Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Iterative Algorithm Development

Completing an algorithm and improving it

Sessions 17

What we will be doing

- Complete MakeCombinations module, making CombinationCollection available in the event for downstream consumers and for writing to a file.
- We will use the things listed in our strategy to make this algorithm reasonably good.
- Procedure
 - Working in “art-workbook / IterativeAlgDevel”
 - Will be working in three phases
 - See README_phase1, 2, and 3 in this unit
 - There is a README_extra, if you can make it there.

Phase 1 - finish the functionality

- The starting point - what we are given
- Complete the first version of the module
 - Invent helper class that encapsulates track representation and functions to manipulate it
 - Test the new track helper class, which is now independent of the framework.
 - Verify that the module is doing the right thing

Phase 2 - algorithm independence

- Revisit the histogram filling
- Extract an algorithm from the module
 - extract combinations block into an algorithm function
 - Test the algorithm outside the module
 - Think about what features need testing and why

Phase 3 - polish the algorithm

- Factorization
 - Make the algorithm readable and good for maintenance
- Using standard C++ features in the algorithm
- Optional advanced exercises