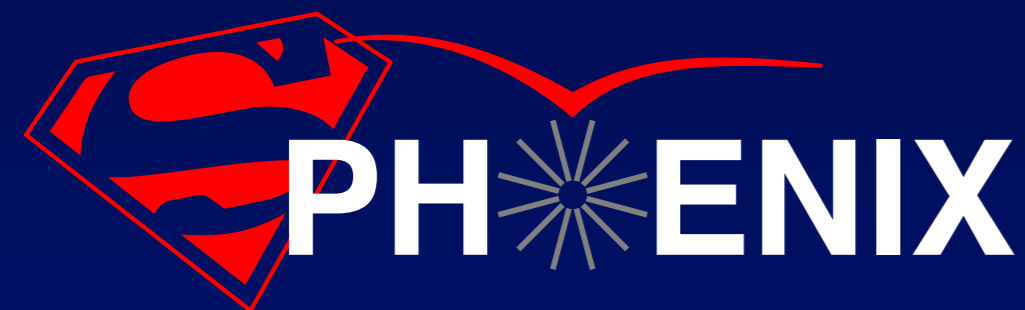# *MINIMAL FULL MAPS TRACKER*

MICHAEL P. MCCUMBER
LOS ALAMOS NATIONAL LABORATORY

SPHENIX TRACKING MEETING
2/19/2016

MAPS C&S WORKFEST

Dear Colleagues,

We are pleased to announce the "sPHENIX MAPS Cost and Schedule Workfest" which will be held in Santa Fe, New Mexico, March 30th - April 1st 2016. The purpose of this 3 day workshop is to define and document the cost and schedule for the MAPS based tracking options under consideration for the sPHENIX detector. The interactive workfest format will be organized into topical breakout teams with MAPS, engineering, and C&S experts we are gathering from ALICE, sPHENIX, and other projects and will minimize time spent in presentations.

Additional details will be given in the next announcement and posted on the following meeting webpage: https://indico.bnl.gov/conferenceDisplay.py?confId=1741
There is no registration fee, but we ask that you register as a participant if you plan on attending.

We look forward to seeing you in Santa Fe.

Sincerely,
Mike McCumber,
Ming Liu

Go to day ▾

Wednesday, March 30, 2016

09:00 - 10:50	Morning Plenary	
	09:00	**Welcome** *10'*
	09:10	**MAPS in sPHENIX** *20'*
	09:30	**Draft Cost & Schedule Document** *20'*
	09:50	**Key Issues from Expert #1** *30'*
	10:20	**Key Issues from Expert #2** *30'*
10:50 - 12:00	Morning Breakout	
12:00 - 13:30	Lunch Break	
13:30 - 17:30	Afternoon Breakout	

C&S PREPARATION SIMULATIONS

In order to C&S a full MAPS tracker, we need to optimize that design for minimal surface area (material cost) while preserving the physics output.

I've started that effort, here is an initial look at removing the 5th of 7 layers (r=24.5 cm) and shows no degradation in performance.

solid = 6 layer MAPS, open = MIE full efficiency default
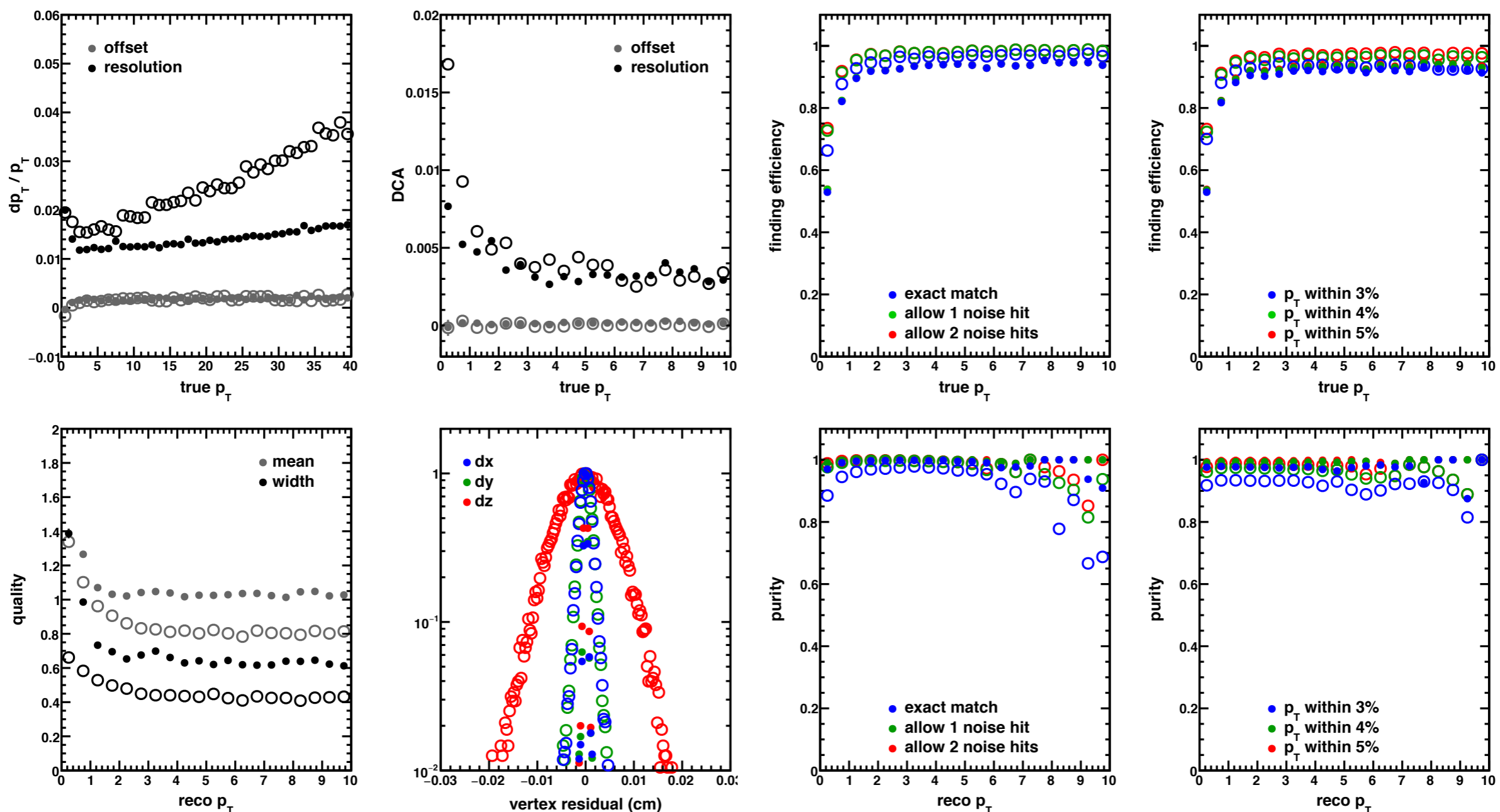
The DCA plot doesn't show the improvement from the much better vertexing because that isn't how the tracking is setup right now. The DCA is computed against (x,y) = (0,0) and not the vertex position. I need to fix this and the vertex resolution plot (easy) so that we can start tossing against more realistic distributions of collision points.

I toyed with implementing RAVE but ran into some issues with the current build. I may just have the tracking slide to the reconstructed vertex and redo the tracking (but this struck me as wasteful on the CPU, so I initially tried a RAVE implementation). I just need to find some time to sort out a working approach…