ACD Recon Upgrade -- Pass 8 First Look

⚠️ Note: The pre-filters and MC samples in this analysis are consistent with the analysis Eric was performing; however, they are significantly different from the data sets and pre-filters Bill applied in designing the Pass6 and Pass7 analyses. For example, these data sets have no cut on FswGamState == 0 and the IrrFilter is different (and almost certainly having unintended consequences).

Introduction

I've gone back to Eric's old page with the hope of picking up where he left off. Of course, this leads to the always enjoyable task of replicating the work that he has already done. The working directory for the previous analysis is:

```
/afs/slac/u/ek/echarles/vol2/glast_workdirs/GR_v15r50p3
```

From there, I have been able to more or less reconstruct the analysis that Eric was performing. Building my own version of GR v15r50p3, I used the runme.csh and runme_ag.csh scripts (with minor modifications) in order to generate a set of background and allGamma MC. Key points about these MC simulations:

- The bkg sims are on orbit using a mix of particle types...
- Other ???

Looking in gamma_ana.txt, I found the exact cuts that Eric was using. Namely, the prefilters that were being applied before the cut efficiencies were evaluated. I've listed them here for completeness. One slightly troubling detail is the dependence on CTBCore > 0.1 (which has not yet been defined in the Pass8 analysis). Another is that the irreducible background filter does not seem to cut out the Earth10 photons which come standard in the background mix.

nSig	nBkg
~9x10^5	~4x10^6

Name	Purpose	Gamma Efficiency	Bkg Efficiency	Cut

IrrFilter	Remove irreducible bkg events	0%	13%	`((McZDir < -.2 && McId < 20) | (McZDir < .1 & McId > 20))`
				`&& McAcdZEnter > 100 && McCharge !=0 &&`
				`McTHPosHitOthers < 6`

TkrCalCore	Take only "good" events	6.2%	2.2%	`TkrNumTracks>0 && CalEnergyRaw > 5 && CalCsIRLn>4 && CTBCORE>0.1`
AcdEngFilter	Only events with moderate signal in the ACD	0%	47%	`(AcdTotalEnergy+AcdRibbonEnergy<0.05) && McZDir > 0`
Prefilter	Only select relevant events ⚠️ !(IrrFilter) is wrong ⚠️	6.2%	1.2%	`(TkrCalCore && !(IrrFilter) && !(AcdEngFilter)`

	nSig	nBkg
Before Prefilters	**~9x10^5**	**~4x10^6**
After Prefilters	**55015**	**48614**

One very important thing to note here is how confusing the convention of "Cuts" and "Filters" is. There needs to be some standard convention for either keeping or rejecting events that return TRUE for a selection (I believe that Bill uses "Cuts" and "Vetos").

Pass 6 Performance

First defining some ACD variables:

```
"AcdCornerDocaENorm" : "AcdCornerDoca*(min(1000, max(30,CTBBestEnergy)))^.5/10.",
"AcdTileEventEnergyRatio" : "AcdTkr1ActDistTileEnergy/max(1., CTBBestEnergy)*100",
"AcdTkr1ActiveDistENorm" : "AcdTkr1ActiveDist * sqrt(min(3000., max(10, CTBBestEnergy)))/10.",
"AcdTotalTileEventEnergyRatio" : "AcdTotalEnergy/max(1., CTBBestEnergy) * 100",
```

Then, applying the prefilters (which cut out a large portion of un-reconstructable events), I finally apply the ACD background rejection cuts, omitting the CT analysis to the files that I generated with Eric's scripts. I've also listed, in parentheses, the background efficiency removing the Earth10 photon contamination. The efficiency for each cut is listed stacked with the preceding cuts.

Name	Purpose	Gamma Efficiency	Bkg Efficiency	Cut
BasicTileCut	Reject events with track pointing at struck tile.	95.2%	7.85 (7.37)%	`(Tkr1SSDVeto == 0 && AcdTkr1ActiveDist > -16 && AcdTkr1ActDistTileEnergy > .4)`
RibbonCut	Reject events with track pointing at struck ribbon.	94.9%	7.43 (6.95)%	`(AcdRibbonActDist > -(2 +350/sqrt(max(20, CTBBestEnergy))) && Tkr1SSDVeto < 3 && AcdRibbonEnergy > .05 )`
TotalTileEnergyCut	Reject events with excess ACD total energy.	89.3%	1.17 (1.04)%	`(AcdTotalTileEventEnergyRatio > .8 || (AcdTkr1ActiveDistENorm > -300 && AcdTotalTileEventEnergyRatio > max(.005, .1 - .0001*AcdTkr1ActiveDistENorm)))`
CornerCut	Reject events in the corner gap of the ACD.	88.4%	1.15 (1.03)%	`((Tkr1LATEdge/1.5) ^ 2 + (AcdCornerDocaENorm - 10)^2 < 3800 || (Tkr1LATEdge < 80 && abs(AcdCornerDocaENorm-2) < 4)) && Tkr1SSDVeto < 3`
TileEdgeCut	Reject events at tile edges with decreased signal.	88.3%	1.15 (1.03)%	`(abs(AcdTkr1ActiveDistENorm) < 15 && AcdTotalTileEventEnergyRatio > .005)`

These numbers seem to agree well with Eric's results. However, it naively seems like there are still some photons sneaking into the bkg sample. This is because selecting !(IrrFilter) will return True if McCharge == 0 (leaving about ~50 gamma events out of ~550). Putting a cut on McCharge, we now get a background rejection of 1.03%.

Pass 7 Performance

Moving on to Pass7, the ACD analysis has be modified a fair amount. I rebuilt GR v17r35p10 and generated roughly the same number of signal and background events (again using Eric allGamma.txt and background.txt scripts). Since Pass7 was just a re-design of the worksheet, the prefilters should have the same efficiency (Irreducible 13%, AcdEngCut 47%) with a slight change in CalTkrCore since it depends on CTBCore (2.2% for bkg and 6.4% for sig) leaving a total of 1.3% of the bkg and 6.4% of the gamma-ray events.

Again defining some ACD variables:

```
    "AcdCornerDocaENorm" : "AcdCornerDoca*(min(1000, max(30, CTBBestEnergy)))^.5/10." ,
    "AcdTkr1RibbonActDistENorm":"AcdTkr1RibbonActDist * sqrt(min(3000., max(10, CTBBestEnergy)))/10.",
    "AcdTkr1ActiveDistENorm":"AcdTkr1ActiveDist * sqrt(min(3000., max(10, CTBBestEnergy)))/10." ,
    "Tkr1ACDTopX":"Tkr1X0 + Tkr1XDir*(755-Tkr1Z0)/Tkr1ZDir" ,
    "Tkr1ACDTopY":"Tkr1Y0 + Tkr1YDir*(755-Tkr1Z0)/Tkr1ZDir" ,
    "AcdTileEventEnergyRatio":"100*AcdTkr1ActDistTileEnergy/max(10., CTBBestEnergy)" ,
    "AcdTotalTileEventEnergyRatio":"100.*AcdTotalEnergy/max(10., CTBBestEnergy)" ,
    "Tkr1ACDSideZ":"min((Tkr1Z0 + abs(Tkr1ZDir*(840 - abs(Tkr1Y0))/Tkr1YDir)),(Tkr1Z0 + abs(Tkr1ZDir*(840 - abs
(Tkr1X0))/Tkr1XDir)))" ,
    "AcdTkrVActiveDistENorm":"AcdActiveDist3D  * sqrt(min(3000., max(10, CTBBestEnergy)))/10." ,
    # Why this 1* is necessary, we may never know...
    "AcdTkr1RibbonActDistMaxTileEnergy":"1*max(AcdTkr1RibbonActEnergyPmtA, AcdTkr1RibbonActEnergyPmtB)",
```

And looking at cut efficiencies:

Name	Purpose	Gamma Efficiency	Bkg Efficiency	Cut
BasicTileCut	Reject events with track pointing at struck tile.	94.2%	6.18%	`Tkr1SSDVeto< 5 && AcdTkr1ActDistTileEnergy > .7 && AcdTkr1ActiveDistENorm> -350`
RibbonCut	Reject events with track pointing at struck ribbon.	93%	5.85%	`(AcdTkr1RibbonActDistENorm > -40 && Tkr1SSDVeto < 3 && AcdTkr1RibbonActDistMaxTileEnergy > .04) || (AcdTkr1RibbonDist > -1/(CTBBestEnergy/100) && Tkr1SSDVeto < 2)`
TotalTileEnergyCut	Reject events with excess ACD total energy.	88.1%	2.08%	`AcdTotalTileEventEnergyRatio > .8 || AcdTkr1ActiveDistENorm > -200 && AcdTotalTileEventEnergyRatio > max(.005, .1 - . 0001*AcdTkr1ActiveDistENorm) * max(1., CTBBestLogEnergy/2.5)`
CornerCut	Reject events in the corner gap of the ACD.	86.9%	2.06%	`((Tkr1LATEdge/1.5)^2 + (AcdCornerDocaENorm - 10)^2 < 6400 && Tkr1SSDVeto < 3) || (Tkr1LATEdge < 300 && abs(AcdCornerDocaENorm-2) < 4)`
TileEdgeCut	Reject events at tile edges with decreased signal.	86.9%	2.06%	`Tkr1SSDVeto == 0 & abs(AcdTkr1ActiveDistENorm) < 10 & AcdTkr1ActDistTileEnergy > .025`
VetoTileCut	Reject events with ...	86.6%	1.71%	`(AcdTkrVActiveDistENorm > -100 && AcdActDistTileEnergy / sqrt(max(1., CTBBestLogEnergy-3.5)) > .9 +.15* TkrVSSDVeto) || (abs(AcdTkrVActiveDistENorm) < 15 && AcdActDistTileEnergy > .25 && TkrVSSDVeto < 2)`

Pass 8 Performance

Generating merit files off of the GlastRelease-HEAD1.1365 version (which came shortly after GlastRelease-v19r3p4), I started by looking at the effect of the Pass7 ACD cuts. Here the prefilters could have changed (since track finding etc. has changed). Using the same cut definitions, I find fairly similar results with slightly worse background rejection.

Prefilter	Gamma Efficiency	Bkg Efficiency
IrrFilter	0%	12%
TkrCalCore	6%	48%
AcdEngFilter	0%	2.4%
Total	6%	1.6%

Name	Gamma Efficiency	Bkg Efficiency
BasicTileCut	94.2%	11%
RibbonCut	93.3%	10.6%
TotalTileEnergyCut	87.1%	2.47%
CornerCut	86.1%	2.46%
TileEdgeCut	86.1%	2.45%
VetoTileCut	85.7%	2.01%

Next, I look at implementing the AcdReconV2 cuts that Eric originally suggested. First defining some new AcdReconV2 variables:

```
"Acd2TileEventEnergyRatio" : "Acd2Tkr1ActDistTileEnergy/max(1., CTBBestEnergy)*100",
"Acd2Tkr1ActiveDistENorm" : "Acd2Tkr1ActiveDist * sqrt(min(3000., max(10, CTBBestEnergy)))/10.",
"Acd2TotalTileEventEnergyRatio" : "Acd2TotalEnergy/max(1., CTBBestEnergy) * 100",
"Acd2VetoHit" : "sqrt(Acd2TkrVetoSigmaHit*Acd2TkrVetoSigmaHit + 1.5*Tkr1SSDVeto*Tkr1SSDVeto)",
"Acd2VetoHit1" : "sqrt(Acd2Tkr1VetoSigmaHit*Acd2Tkr1VetoSigmaHit + 1.5*Tkr1SSDVeto*Tkr1SSDVeto)",
"Acd2VetoGap1" : "sqrt(Acd2Tkr1VetoSigmaGap*Acd2Tkr1VetoSigmaGap + 1.5*Tkr1SSDVeto*Tkr1SSDVeto)",
```

Applying the same prefilters, here are the cuts and their efficiencies.

Name	Purpose	Gamma Efficiency	Bkg Efficiency	Cut
VetoHit1Cut	Reject events with best track pointing a struck tile/ribbon	94.5%	14.2%	`Acd2VetoHit1 < 5.`
TotalTileEnergyCut	Reject events with excess ACD total energy	86.5%	2.44%	`(Acd2TotalTileEventEnergyRatio > .8 || (Acd2Tkr1ActiveDistENorm > -300 && Acd2TotalTileEventEnergyRatio > max(.005, .1 - .0001*Acd2Tkr1ActiveDistENorm)))`
VetoGap1Cut	Reject events in gaps of the ACD	86.2%	2.42%	`Acd2VetoGap1 < 2.`
VetoHitCut	Reject events if other track points at hit tile	86.1%	2.35%	`Acd2VetoHit < 2.`

Conclusions

Well, it doesn't seem like we are where we want to be (not a surprise). It looks like we would like to roughly double the background rejection (1/2 the efficiency) for the Pass 8 analysis with the AcdReconV2 variables. However, it appears that the simple cuts with AcdReconV2 are roughly comparable to those with the original AcdRecon, meaning that the degradation in background rejection may be in part due to upstream reconstruction changes (which in any case are not yet complete).

Looking Forward

- Can we easily modify the AcdV2 cuts to increase background rejection with the current reconstruction? Basically, can we optimize these cuts?
- Right now AcdV2 is using the G4 propagated covariance matrix, which really isn't right for cosmic-rays (since it uses electron hypothesis). Would changing this make an improvement?
- How do the CalOnly events look? Can we develop some rough background rejection for them?
- Obviously this will improve as the upstream reconstruction improves. Can we provide any guidance for that?
- CTs are always the after-burner to get the boosted rejection power that we need.