

new hittree soft Document

Hiroki Yoneda

2017 年 7 月 31 日

これまでの hittree では、一挙に行われていた merge や reconstructなどを分解して解析することや、それらのアルゴリズムをより柔軟に実装できるのを目標として、”new hittree soft”を作成中である。この Document では、database や生成ファイルについて、説明する。

1 Database

検出器に与えるデータベースは、detector_map、detector_profile、cal 関数の 3 種類必要である。

1.1 detector_map

ASICID, ASICCH と DETID, DETCH との対応付をするデータである。”detector_map”という TTree 形式で与えられる。以下のような Branch を持っている。また、asic は、64ch 持っていることを前提している。

Branch 名	型	説明
asicid	Int_t	使用していない ch であっても記述する必要あり 検出器全体での ch に対する通し番号。使用していない ch は、”-1”を記入。 検出器の ID。Si を 0-9 に、CdTe を 10 以上にすることを推奨。 検出器内での Ch。隣接 Ch は、detch も隣接する。使用していない ch は、”-1”を記入。
asicch	Int_t	
remapch	Int_t	
detid	Int_t	
detch	Int_t	

1.2 detector_profile

検出器は、xy 面に平行であり、各ストリップは、x 軸、または、y 軸に平行であることを前提としている。検出器が 3 次元的に複雑に配置される場合は、hittree_lv3 から変換する必要あり。

1.3 cal 関数

”calfunc_DETID_DETCH”という名前の TSpline3 を持っている root file で与えられる。

2 生成ファイル

データの生成は、eventtree → hittree_lv1 → hittree_lv2 → hittree_lv3 の 3 段階に分かれて行う。

Branch 名	型	説明
detid	Int_t	検出器の ID。Si を 0-9 に、CdTe を 10 以上にすることを推奨。 使用していない ch は、ここでは記入する必要なし。 0: Si、1: CdTe 0: Ground 1: HVside 位置情報を持たない場合は、適当な値 (0 など) を入れる。 位置情報を持たない場合は、適当な値 (0 など) を入れる。 位置情報を持たない場合は、"-1"を入れる。負値から、ストリップ方向を認識する。 位置情報を持たない場合は、"-1"を入れる。負値から、ストリップ方向を認識する。 0: Good Channel 1: Bad Channel
detch	Int_t	
detector_material	Int_t	
detector_HV	Int_t	
pos_x	Double_t	
pos_y	Double_t	
pos_z	Double_t	
delta_x	Double_t	
delta_y	Double_t	
delta_z	Double_t	
badch	Int_t	

hittree_lv1 は、データベースの適用を行うのみとする。hittree_lv2 は、ストリップのマージを基本的には行う。hittree_lv3 は、マージされたストリップシグナルを元に、ヒットを再構成する。各段階で、新しい Branch が元の tree に追加されていく形を採用する。したがって、hittree_lv3 は、eventtree, hittree_lv1, hittree_lv2 の情報も保持している。

2.1 hittree_lv1

エネルギーレシヨルドは適用せず、データベースを当てるのみ。各ブランチは、データベースの値をコピーした値になっている。1 番目以外は、全て可変長配列。

Branch 名	型	説明
nsignal_lv1	Int_t	シグナル数
detid_lv1[nsignal_lv1]	Int_t	
detch_lv1[nsignal_lv1]	Int_t	
remapch_lv1[nsignal_lv1]	Int_t	
detector_material_lv1[nsignal_lv1]	Int_t	
detector_HV_lv1[nsignal_lv1]	Int_t	
epi_lv1[nsignal_lv1]	Double_t	
pos_x_lv1[nsignal_lv1]	Double_t	
pos_y_lv1[nsignal_lv1]	Double_t	
pos_z_lv1[nsignal_lv1]	Double_t	
delta_x_lv1[nsignal_lv1]	Double_t	
delta_y_lv1[nsignal_lv1]	Double_t	
delta_z_lv1[nsignal_lv1]	Double_t	

2.2 hittree_lv2

マージアルゴリズムを適用した後のシグナル情報。pos や delta は、マージアルゴリズム内で計算する必要がある。

Branch 名	型	説明
nsignal_lv2	Int_t	各シグナルにおいて、マージに使用した元のシグナル数
detid_lv2[nsignal_lv2]	Int_t	
detector_material_lv2[nsignal_lv2]	Int_t	
detector_HV_lv2[nsignal_lv2]	Int_t	
n_merged_signal_lv2[nsignal_lv2]	Int_t	
pos_x_lv2[nsignal_lv2]	Double_t	
pos_y_lv2[nsignal_lv2]	Double_t	
pos_z_lv2[nsignal_lv2]	Double_t	
delta_x_lv2[nsignal_lv2]	Double_t	
delta_y_lv2[nsignal_lv2]	Double_t	
delta_z_lv2[nsignal_lv2]	Double_t	
epi_lv2[nsignal_lv2]	Double_t	

2.3 hittree_lv3

両面情報からヒット情報に変換した後のデータ。pos や delta は、再構成アルゴリズム内で計算する必要がある。

Branch 名	型	説明
nhit	Int_t	reconstuct 後のヒット数
detid_lv3[nhit]	Int_t	
detector_material_lv3[nhit]	Int_t	
pos_x_lv3[nhit]	Double_t	
pos_y_lv3[nhit]	Double_t	
pos_z_lv3[nhit]	Double_t	
delta_x_lv3[nhit]	Double_t	
delta_y_lv3[nhit]	Double_t	
delta_z_lv3[nhit]	Double_t	
epi_lv3[nhit]	Double_t	

3 その他

asic の最大数は、100 としている。