

代码功能添加 scaling

RecNeutralTool.cc:

```
/// default variable for resolution scale;
```

```
m_scaleResEne = 1.0;
```

```
m_scaleResThe = 1.0;
```

m_scaleResThe:

```
m_scaleResPhi = 1.0;
```

```
m_scaleRecEff = 1.0;
```

使用新脚本产生The_scaling Map，光子动量范围0-3.5GeV，角度范围 (-1, 1)，能量70个bin，角度80个bin

构造TH2D，对m_scaleResThe = 1.0调整，scaling暂时赋值为：

```
if (cos< -0.8) scaling= 0.7;
```

```
    else if (cos< -0.4) scaling= 1.0;
```

```
    else if (cos< 0.0) scaling= 1.3;
```

```
    else if (cos< 0.4) scaling= 1.7;
```

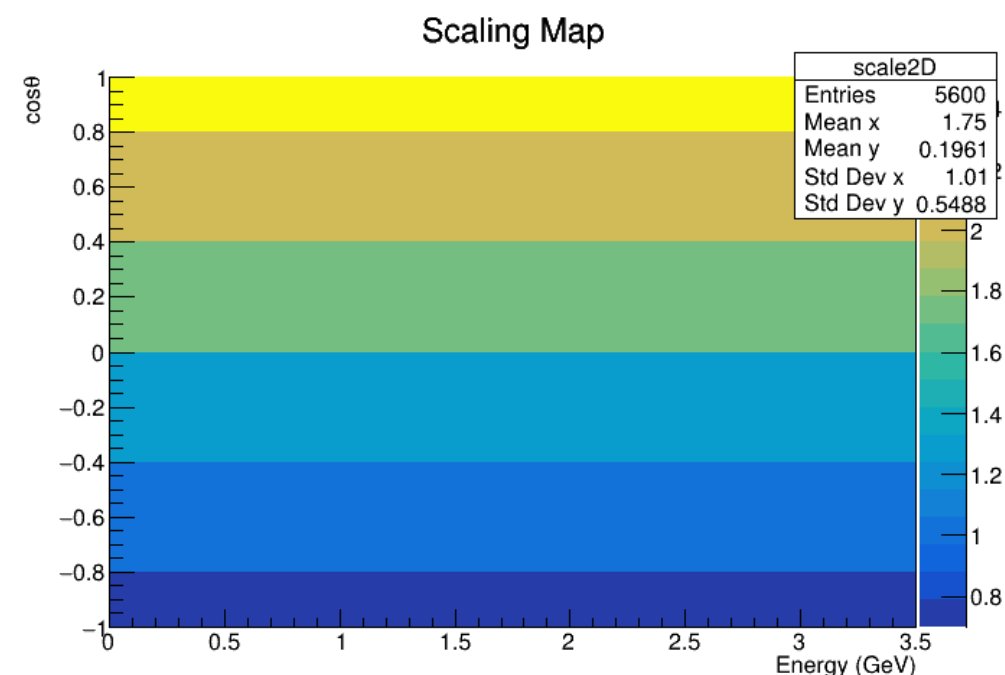
```
    else if (cos< 0.8) scaling = 2.0;
```

```
    else scaling= 2.5;
```

保存为CosThetaScaleTable.root

```
在bool RecNeutralTool::recGamEmcShower( trutrak& m_truTrk, recgam& m_recGam ){
```

中读取CosThetaScaleTable.root并改变m_scaleResThe 的值



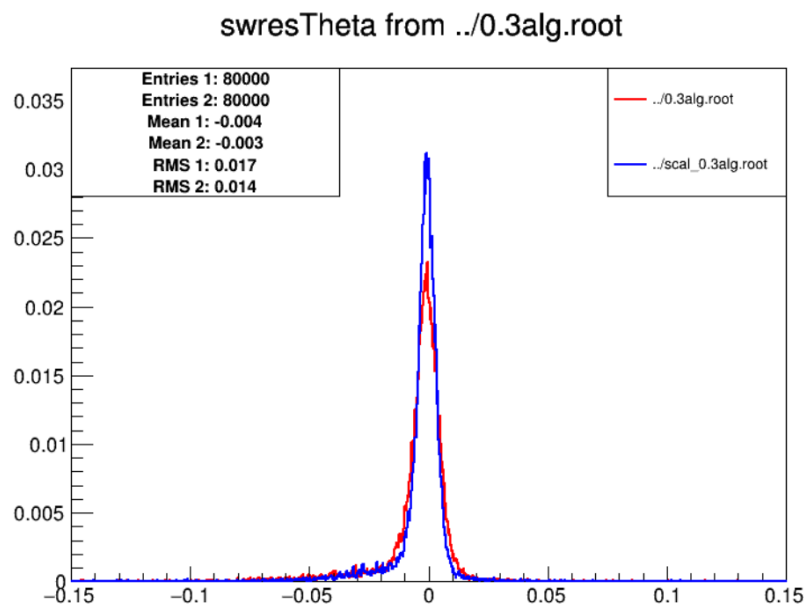
代码功能添加 scaling

RecNeutralTool.cc:

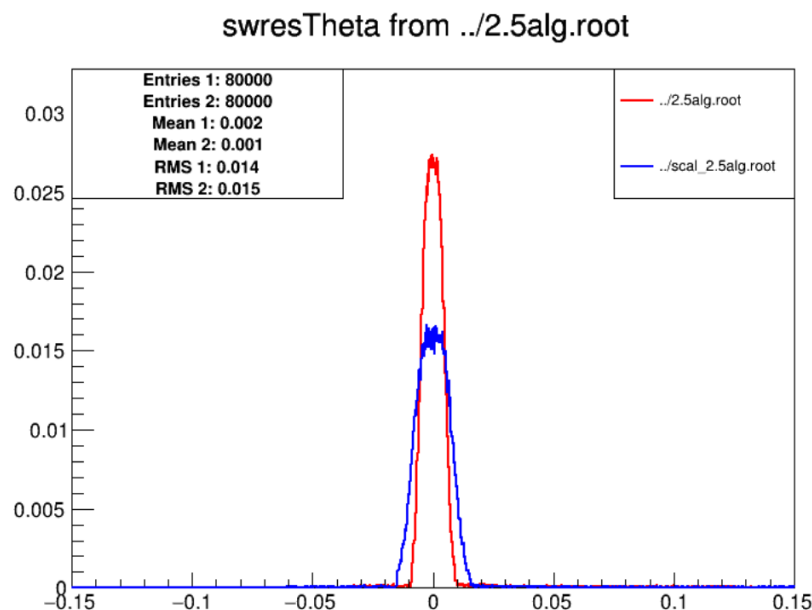
生成cosTheta为-1 to -0.8(0.3GeV) scaling=0.7, 0to0.4(2.5GeV)scaling=1.7, 0.8to1(3.3GeV)scaling=2.5的光子事例各80000个, 在ene_scaling的基础上添加the_scaling功能

跑分析算法FastTestAlg 生成 fastsim_alg.root 和 scal_fastsim_alg.root 对比swresThe分辨

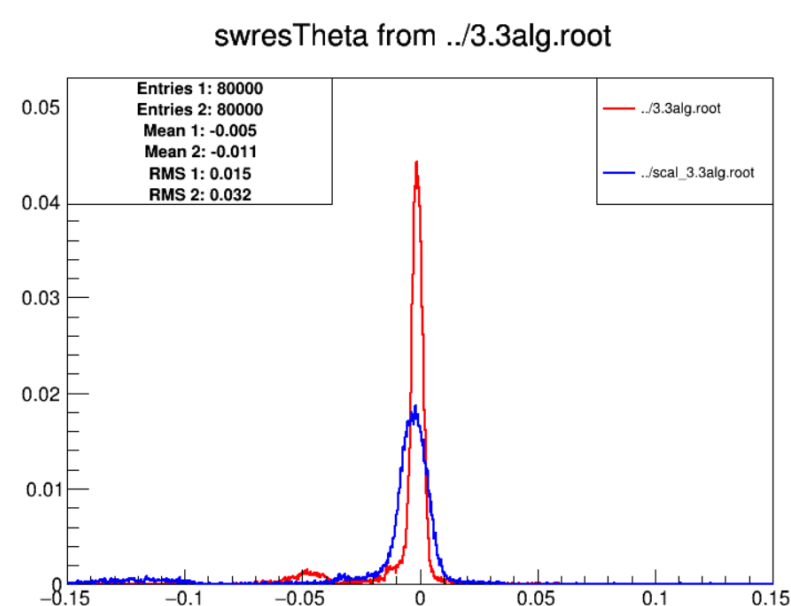
```
// 直方图设置
int nbins = 1000;
float xmin = -0.15;
float xmax = 0.15;
```



0.3GeV, the_scal=0.7



2.5GeV, the_scal=1.7



3.3GeV, the_scal=2.5

代码功能添加 scaling

RecNeutralTool.cc:

m_scaleResPhi:

使用新脚本产生Phi_scaling Map，光子动量范围0-3.5GeV，角度范围 (-1, 1)，能量70个bin，角度80个bin

构造TH2D，对m_scaleResPhi = 1.0调整，scaling暂时赋值为：

```
if (cos< -0.3) scaling= 0.5;
```

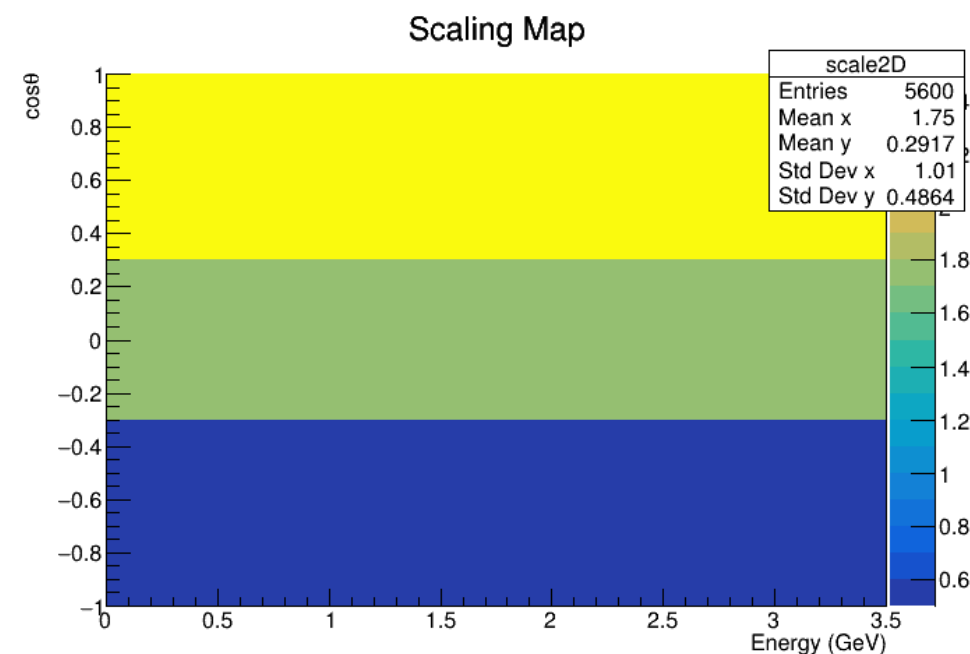
```
    else if (cos< 0.3) scaling= 1.7;
```

```
    else scaling= 2.5;
```

保存为PhiScaleTable.root

在bool RecNeutralTool::recGamEmcShower(trutrk& m_truTrk, recgam& m_recGam){

中读取PhiScaleTable.root并改变m_scaleResPhi 的值



代码功能添加 scaling

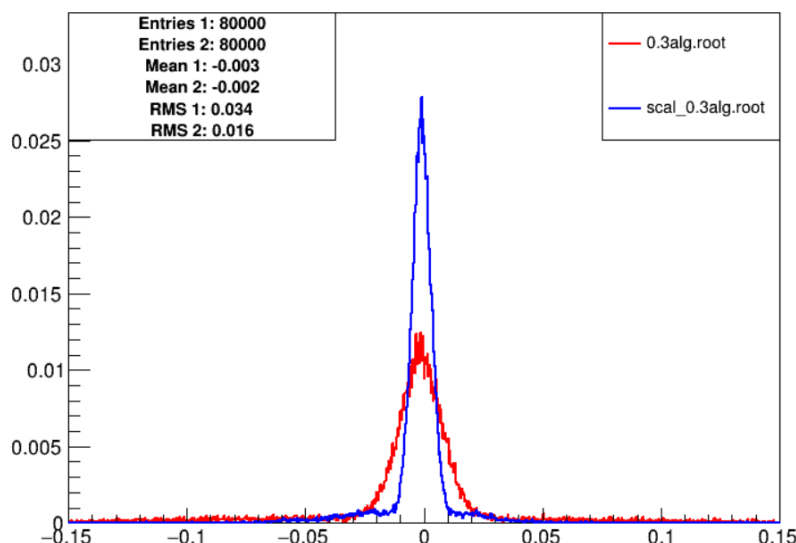
RecNeutralTool.cc:

生成cosTheta为-1 to -0.3(0.3GeV) scaling=0.5, -0.3to0.3(2.5GeV)scaling=1.7, 0.3to1(3.3GeV)scaling=2.5的光子事例各80000个, 在ene_scaling的基础上添加phi_scaling功能

跑分析算法FastTestAlg 生成 fastsim_alg.root 和 scal_fastsim_alg.root 对比swresPhi分辨

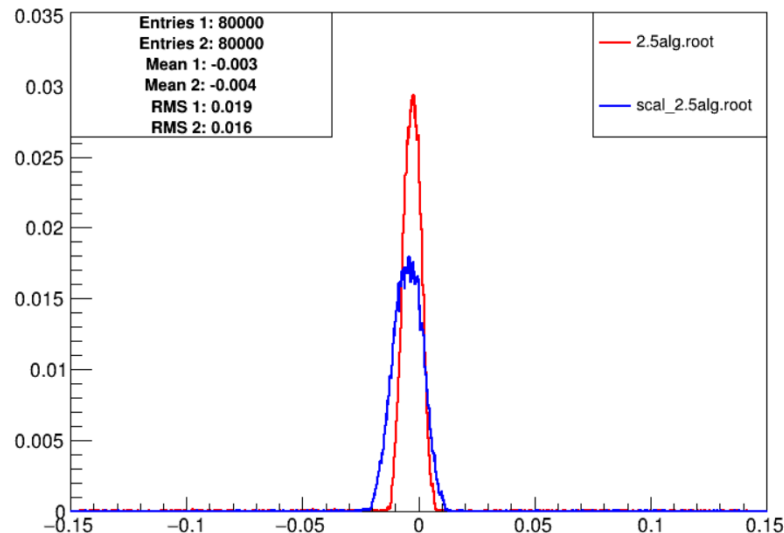
```
// 直方图设置
int nbins = 1000;
float xmin = -0.15;
float xmax = 0.15;
```

swresPhi from 0.3alg.root



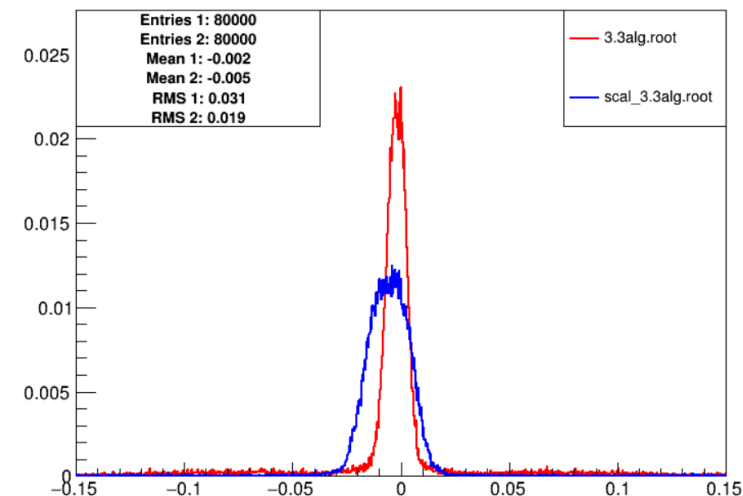
0.3GeV, phi_scal=0.5

swresPhi from 2.5alg.root



2.5GeV, phi_scal=1.7

swresPhi from 3.3alg.root



3.3GeV, phi_scal=2.5

代码功能添加 scaling

RecNeutralTool.cc:

M_scaleRecEff :

对比cos-1to1 , 能量0-3.5GeV , Scaling前后重建效率,光子样本8w个

if ($E < 2.0$) scaling = 0.5;

else if ($E < 3.0$) scaling = 1.7;

else scaling = 2.5;

