

# ATLAS実験におけるGMSBモデルの 長寿命超対称性粒子の探索に関する研究

2009年3月30日(月)

日本物理学会 第64回年次大会

(@立教大学)

神戸大学大学院 理学研究科 物理学専攻

岡田勝吾, 山崎祐司, 藏重久弥, 松下崇, 川越清以

# Outline

## 1. Introduction

## 2. 長寿命超対称性粒子の速度・質量の再構成

## 3. SingleMuonサンプルを用いたバックグラウンドの概算

## 4. まとめ

# 研究の背景

## 長寿命超対称性粒子について

- ✓ Gauge Mediation SUSY-Breaking (GMSB) モデルによると、スレプトンがATLASのミュオン検出器を抜けるまで崩壊せず、「重いミュオン」の様に振る舞う可能性がある。
- ✓ ミュオン検出器で測定した粒子の運動量Pと速度βから質量mを計算し、質量から長寿命スレプトンを探す手法の研究を行っている。

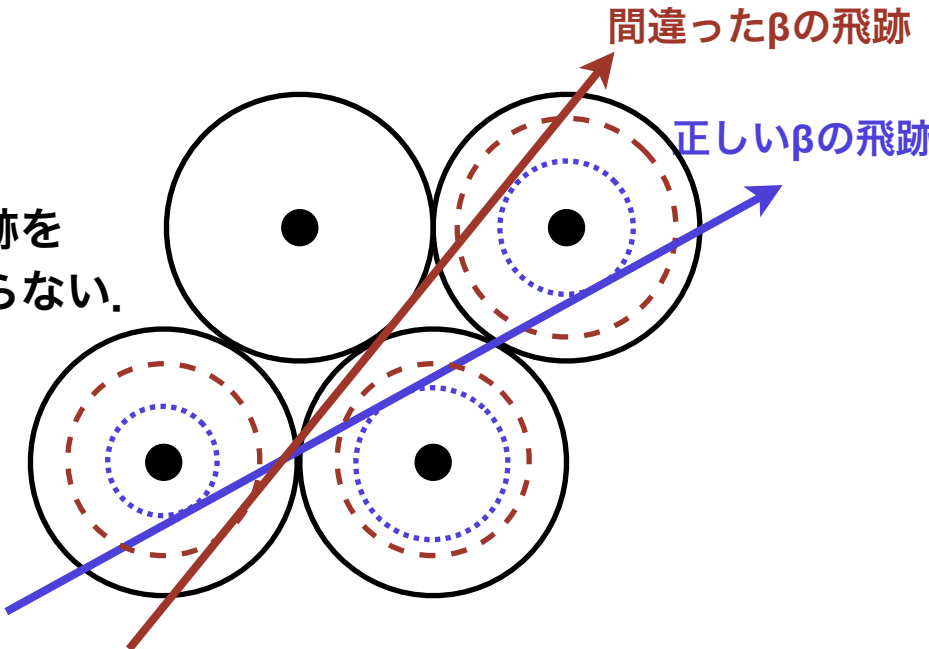
$$m = P/\beta\gamma \quad (\beta = v/c, \gamma = 1/\sqrt{1-\beta^2})$$

--> 質量がO(100GeV)であれば標準模型では存在し得ない新粒子(長寿命超対称性粒子)。

## 問題点

- ✓ 長寿命スレプトンは質量が大きく遅い(β<1)。
- ✓ β=1を仮定したMDTのdrift circleで、スレプトンの飛跡をフィットしようとしても上手く行かず、運動量が決まらない。

--> スレプトンの飛跡を再構成する際、βをfree parameterとして飛跡のフィットを行い、最適なβの値を決める。



Reference: 野本裕史氏 他  
(2007年物理学会@北大 21aZE6)

# 長寿命超対称性粒子探索に必要なミュオン検出器

TGC (Thin Gap Chamber)  
RPC (Resistive Plate Chamber)

ミュオントリガー用検出器。

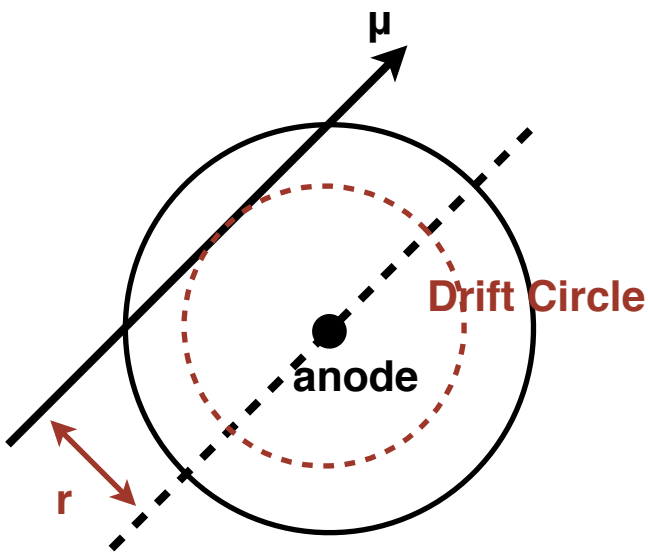
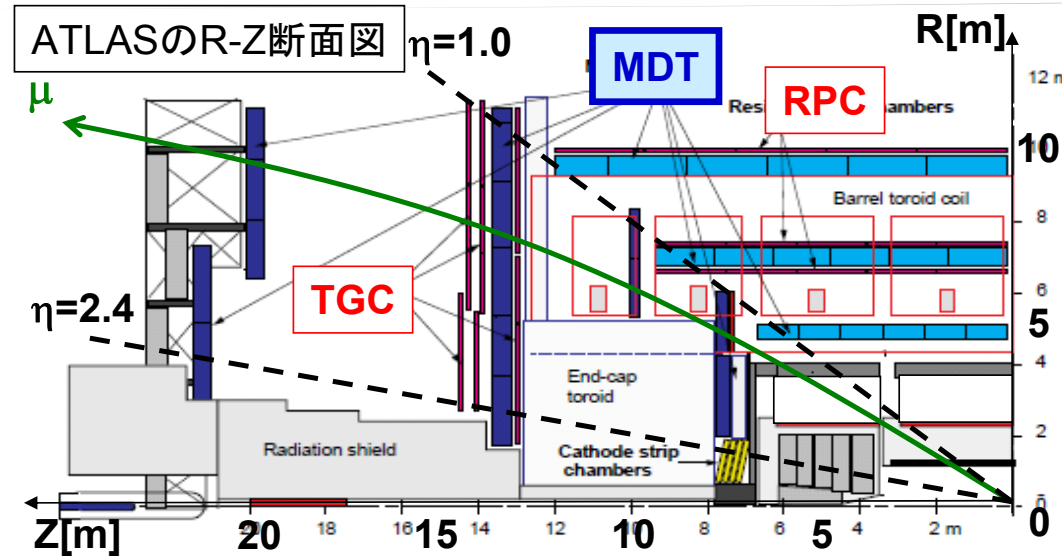
--> 粒子の通過した位置の $\phi$ 座標がわかる。

LHCのBunch Crossing (BC) の時間間隔は25nsec

-->  $\beta > 0.5$ の長寿命粒子は同じBCでトリガーされる。

MDT (Monitored Drift Tube)

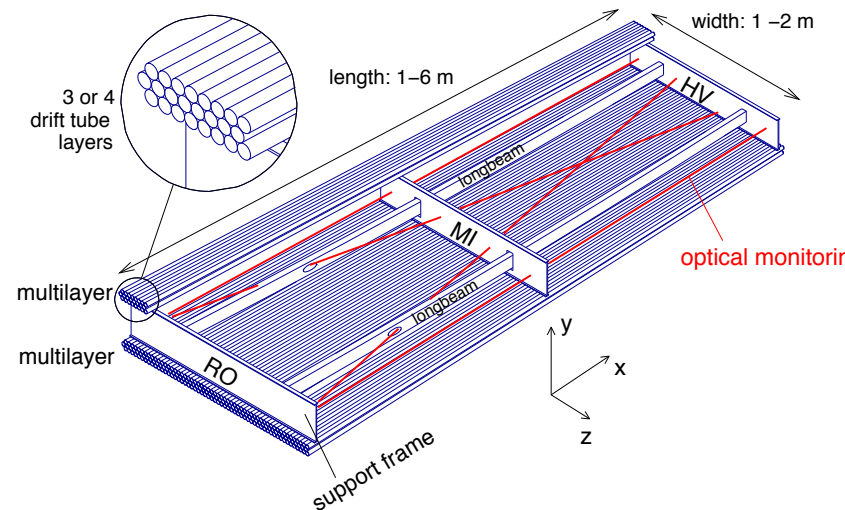
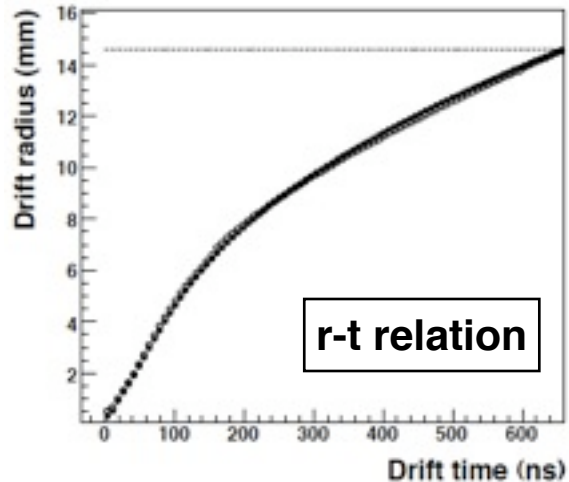
ミュオン検出器内の粒子の位置(R,Z)の精密測定。



★ t: Tubeからの電気信号の測定時刻。

★ t0: ミュオンが衝突点からTubeに到達する時刻。 (-->  $\beta$ の関数)

--> drift time = t - t0 --> drift radiusが決まる。



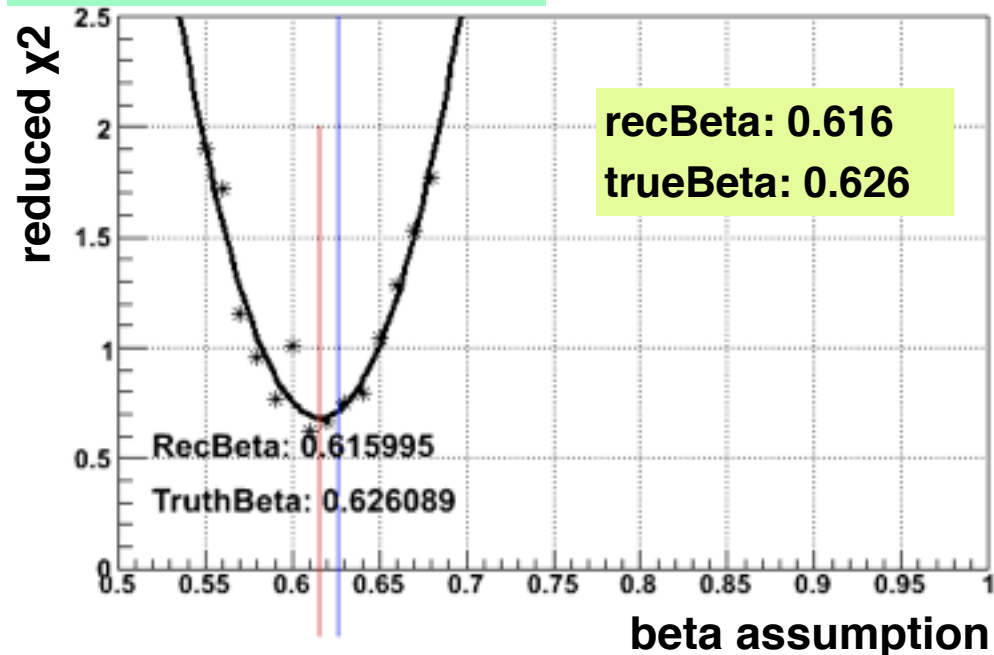
各検出器で測定した(R, $\phi$ ,Z)は粒子の飛跡の再構成で使う。

# 長寿命超対称性粒子の速度 $\beta$ の決定方法

- (1)  $\beta$ の値を変えて ( $\beta=0.50\sim 0.99$ までの間を0.01のステップで) 粒子の飛跡を再構成し、各 $\beta$ での飛跡の $\chi^2$ を計算する。
- (2)  $\chi^2$ をプロットし、二次関数でフィットして最小値を取る時の $\beta$ の値を、その粒子の $\beta$ とする。
- (3)  $\beta$ と運動量から粒子の質量を計算する。

$$m = P/\beta\gamma$$

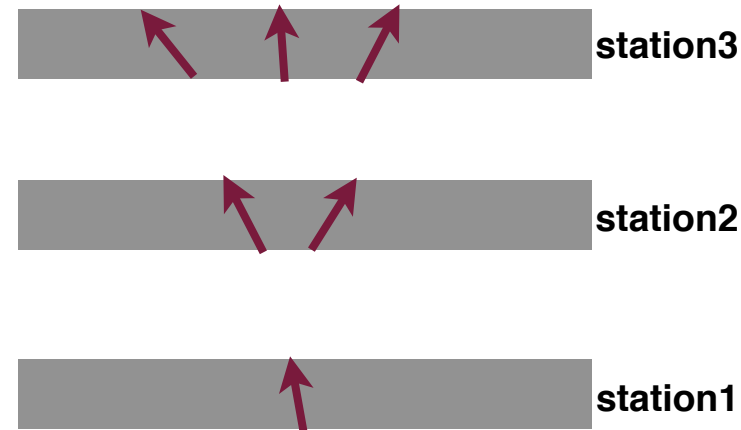
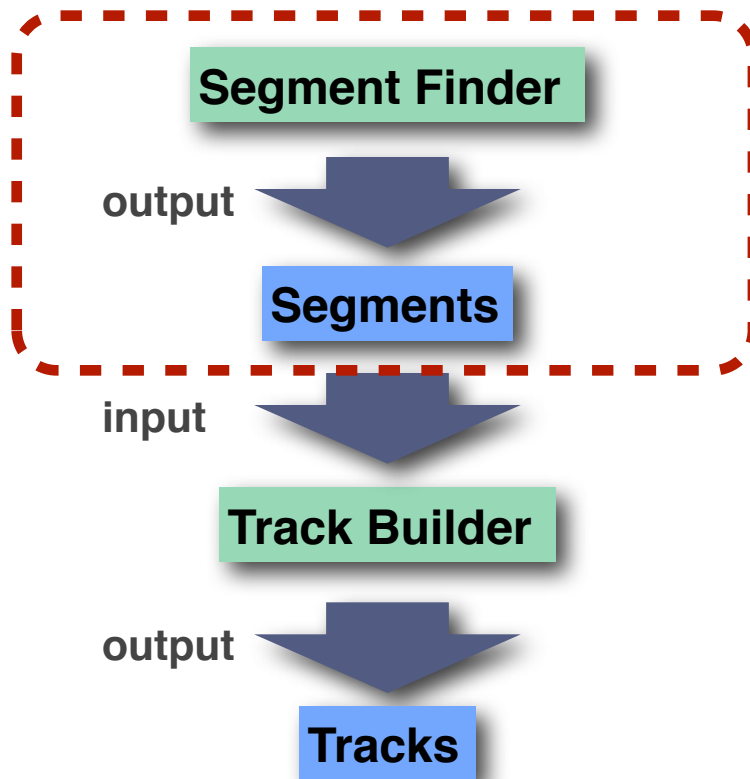
スレプトンの $\chi^2/Ndof$  分布



--> 上記の方法で粒子の速度・質量を再構成するため、Event Filter (LVL3 Trigger) 用のツールで  
ミュオン検出器内の飛跡を再構成するアルゴリズム (TrigMuonEF) を改良した。

# Beta Reconstruction Algorithm with TrigMuonEF

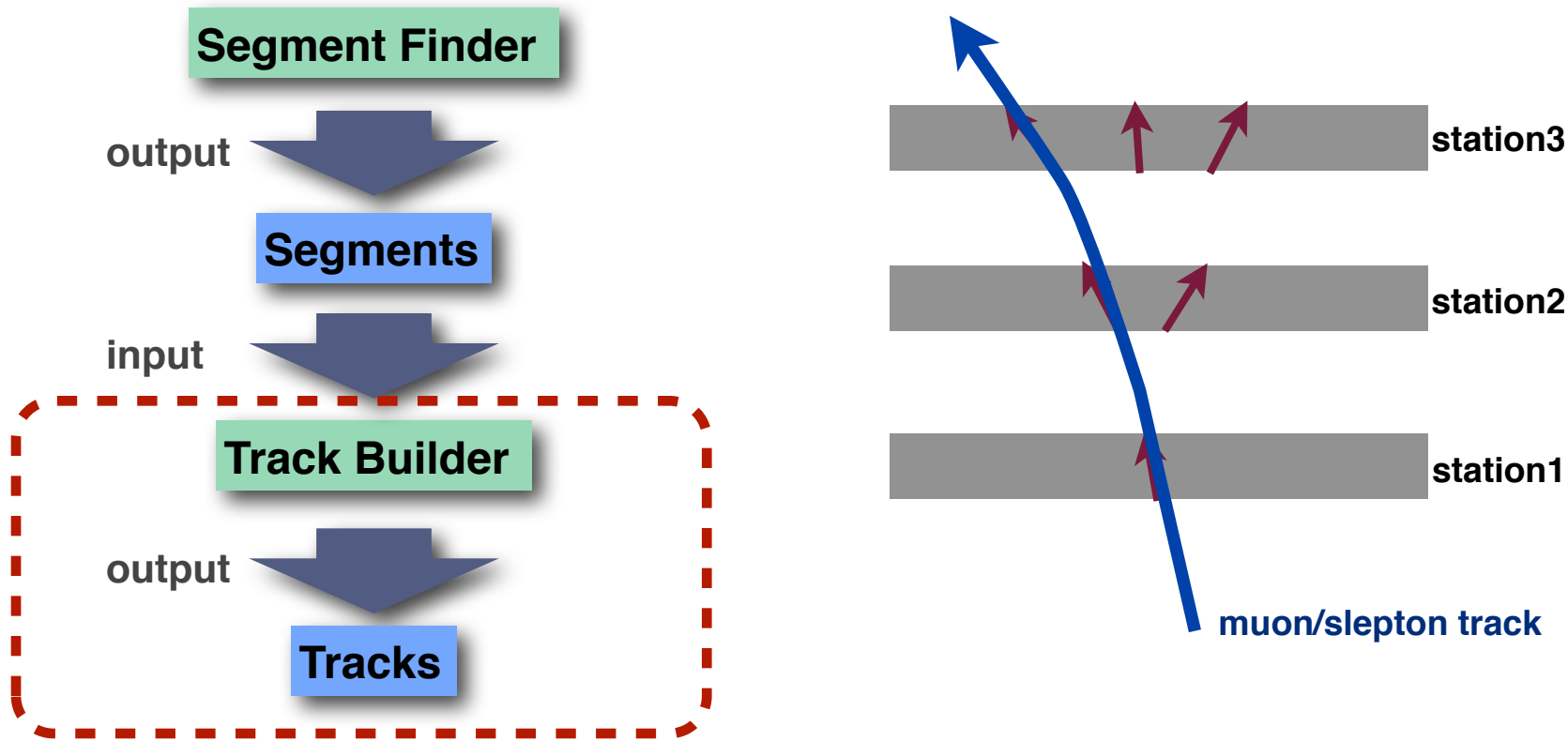
## TrigMuonEFによる飛跡の再構成の流れ



- (1) Segment Finderにより、飛跡のヒット情報を元に、各ミュオン検出器の各station内でパターン認識 (pattern recognition) を行い、各station毎で segmentを作る。

# Beta Reconstruction Algorithm with TrigMuonEF

## TrigMuonEFによる飛跡の再構成の流れ



(2) Track Builderでsegmentをseedとして、station間でパターン認識を行い、  
ミュオン検出器内のミュオン/スレプトンの飛跡を再構成する。

パターン認識の段階で $\beta$ の値を変えて飛跡を再構成するやり方

Segment Finder

Segments

Track Builder

Tracks for each beta assumption

--> 1イベントに対する処理時間が長い.

$\beta$ の値を0.50~0.99の間を0.01のステップで変更

$\beta=1$ で再構成した飛跡を、 $\beta$ の値を変えてドリフト円を計算し直してrefitするやり方

Segment Finder

Segments

Track Builder

Tracks

\*\*Track refitting tool

Refitted tracks with beta

$\beta=1$ で飛跡を再構成する.

$\beta$ の値を変えて先に作った飛跡が通るMDTのドリフト半径を計算し直し、飛跡をRefit.

1イベントあたりの平均処理時間

CPU: AMD Phenom(tm) 9600B

Quad-Core Processor 1.150GHz

	default	パターン認識を行う場合	飛跡をrefitする場合
TrigMuonEFSegmentFinder	14.3 msec/event	559 msec/event	14.3 msec/event
TrigMuonEFTrackBuilder	119 msec/event	6140 msec/event	802 msec/event
Total	133 msec/event	6699 msec/event	816 msec/event

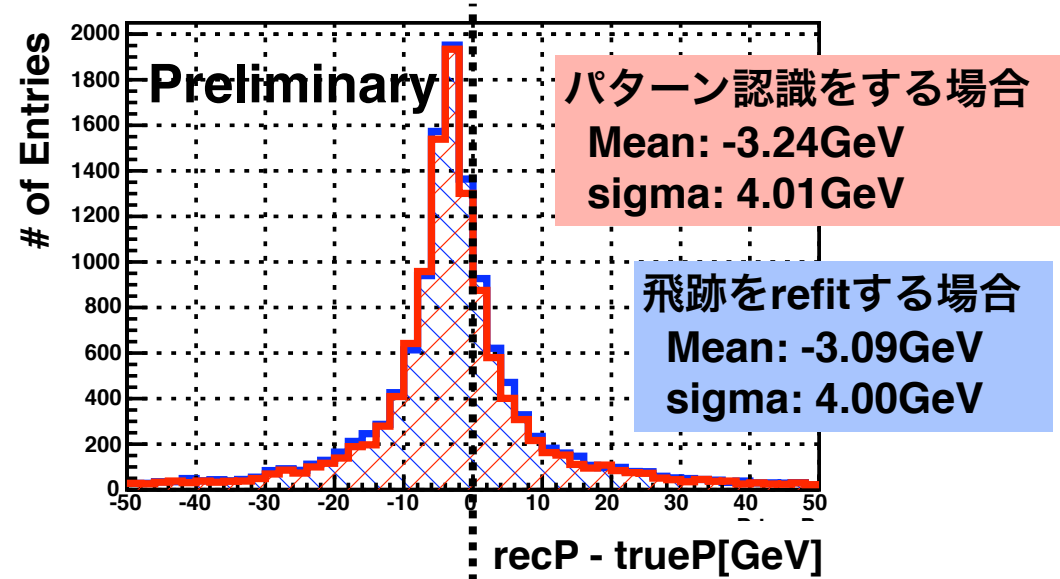
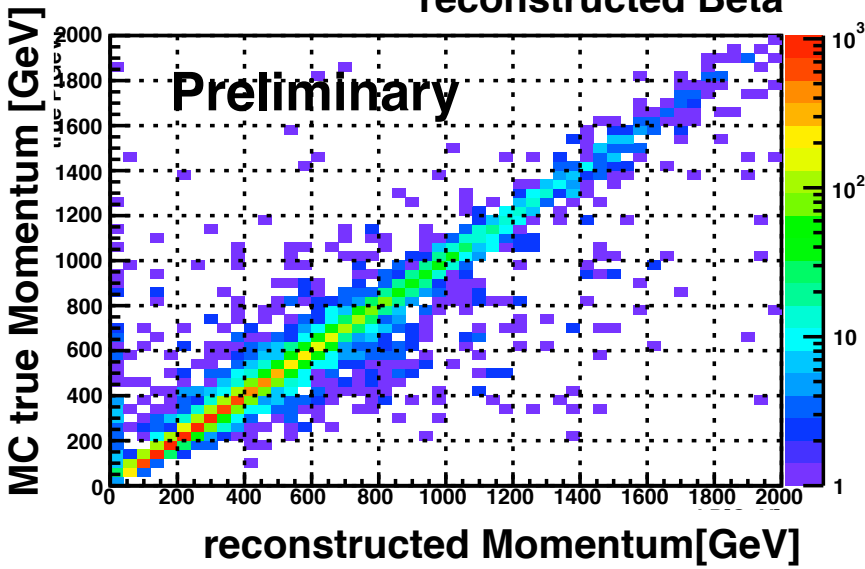
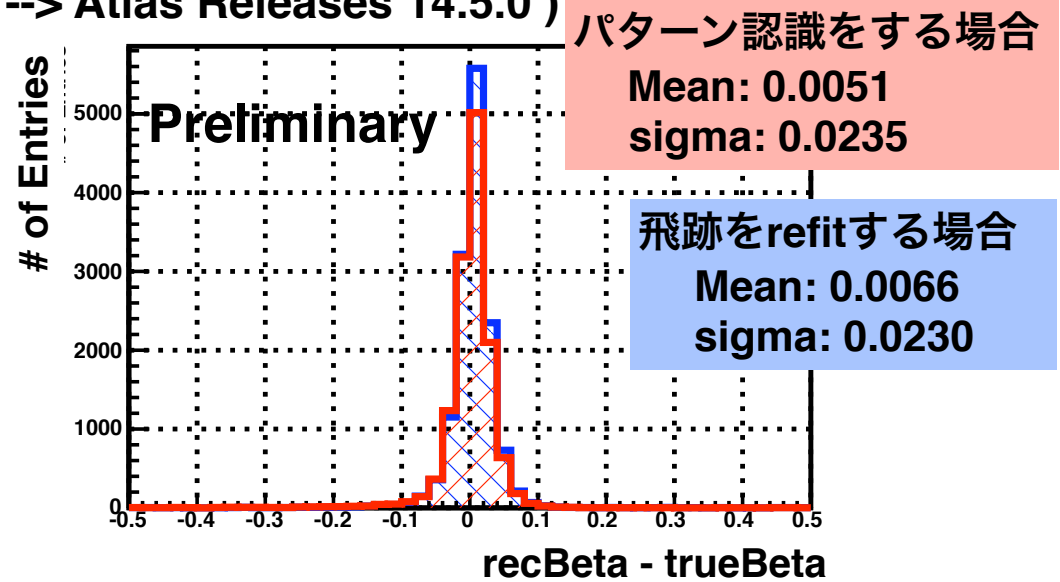
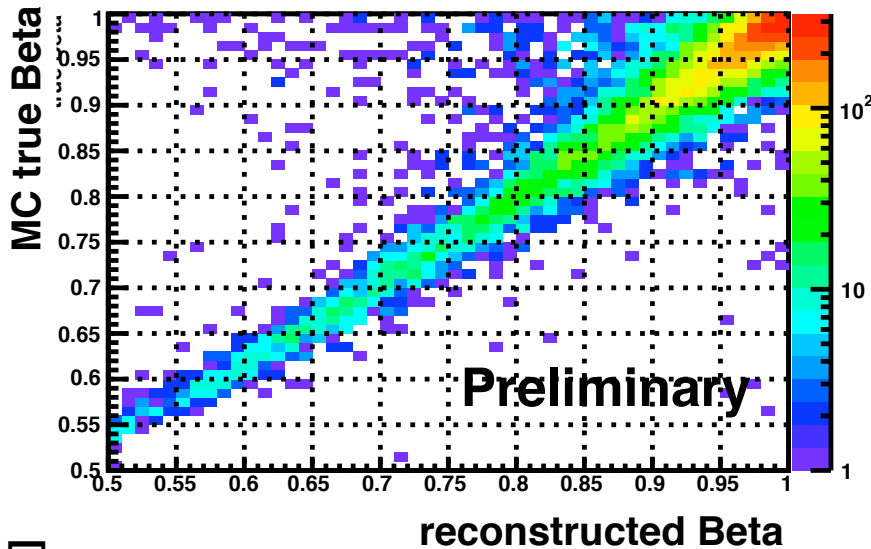
✓ パターン認識を省くことで、処理速度が8~9倍程度速くなった.

Preliminary



# sleptonの速度 $\beta$ と運動量の測定

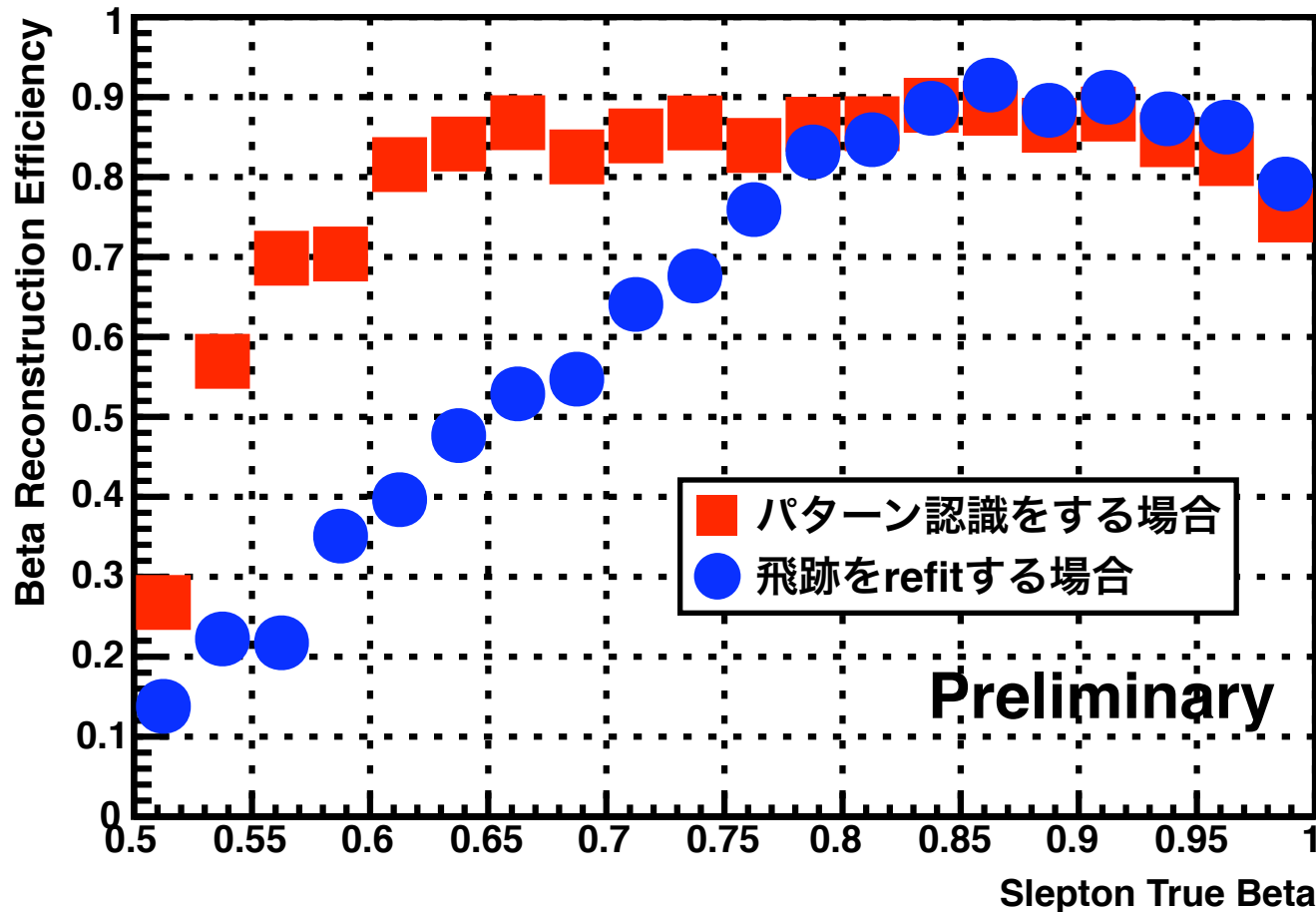
1イベントにつき2個のslepton ( $M_{\tilde{R}}=100.34\text{GeV}$ ,  $M_{\tilde{L}}=102.29\text{GeV}$ )を含む約9,000の SUSY (GMSB5) イベントを再構成した. ( --> Atlas Releases 14.5.0 )



(2D plotsは飛跡をrefitする場合での結果)

— パターン認識をする場合  
— 飛跡をrefitする場合

# $\beta$ Reconstruction Efficiency



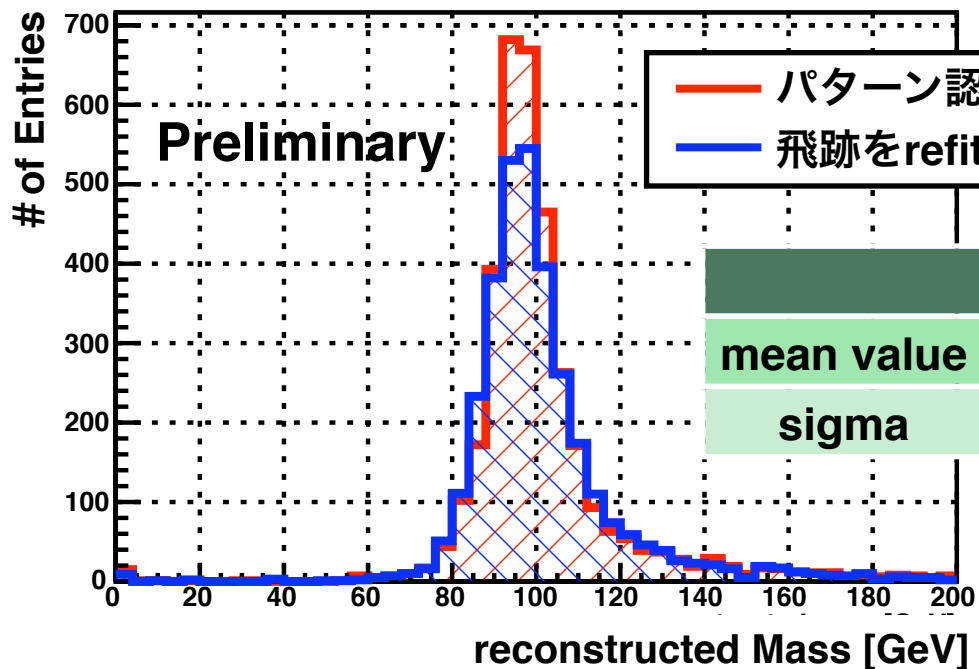
$$\text{Eff.} = \frac{\text{そのうち}\beta\text{が再構成されたsleptonの数}}{\text{生成したsleptonの総数}}$$

(※ $\beta$ が正しいか否かは問わない。)

✓ Beta Reconstruction Efficiencyは、パターン認識を行ったものの方が、広い $\beta$ 領域において70~90%と高い。

# sleptonの質量の再構成

## Reconstructed Mass (recBeta<0.90)



GMSB5:  
 $\tilde{e}_R, \tilde{\mu}_R = 100.34 \text{ GeV}$   
 $\tilde{\tau}_1 = 102.29 \text{ GeV}$

	パターン認識をする場合	飛跡をrefitする場合
mean value	97.1 GeV	96.9 GeV
sigma	7.7 GeV	8.8 GeV

飛跡をrefitする場合の方の、  
 スレプトンの質量の分解能が悪い。

### ★ パターン認識をする場合：

beta	0.50~0.60	0.60~0.65	0.65~0.70	0.70~0.75	0.75~0.80	0.80~0.85	0.85~0.90
mean	94.2GeV	96.1GeV	96.5GeV	97.7GeV	98.2GeV	98.2GeV	97.1GeV
sigma	3.8GeV	4.2GeV	5.0GeV	6.2GeV	6.2GeV	8.4GeV	10.2GeV

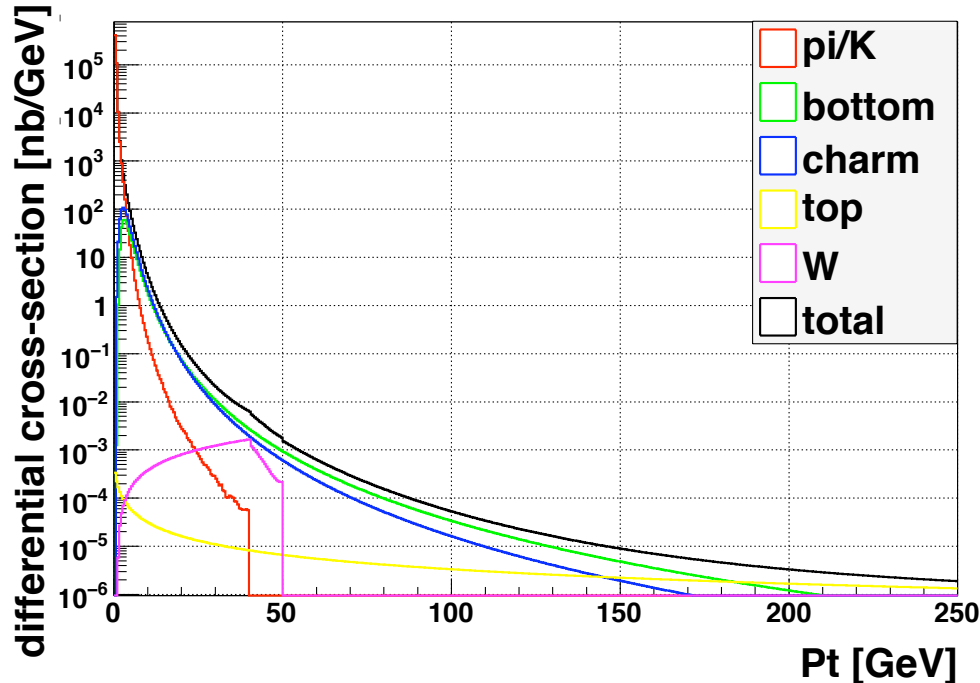
### ★ 飛跡をrefitする場合：

beta	0.50~0.60	0.60~0.65	0.65~0.70	0.70~0.75	0.75~0.80	0.80~0.85	0.85~0.90
mean	92.9GeV	96.0GeV	95.6GeV	96.8GeV	97.4GeV	97.9GeV	96.8GeV
sigma	5.9GeV	6.6GeV	7.0GeV	7.6GeV	7.1GeV	8.8GeV	10.1GeV

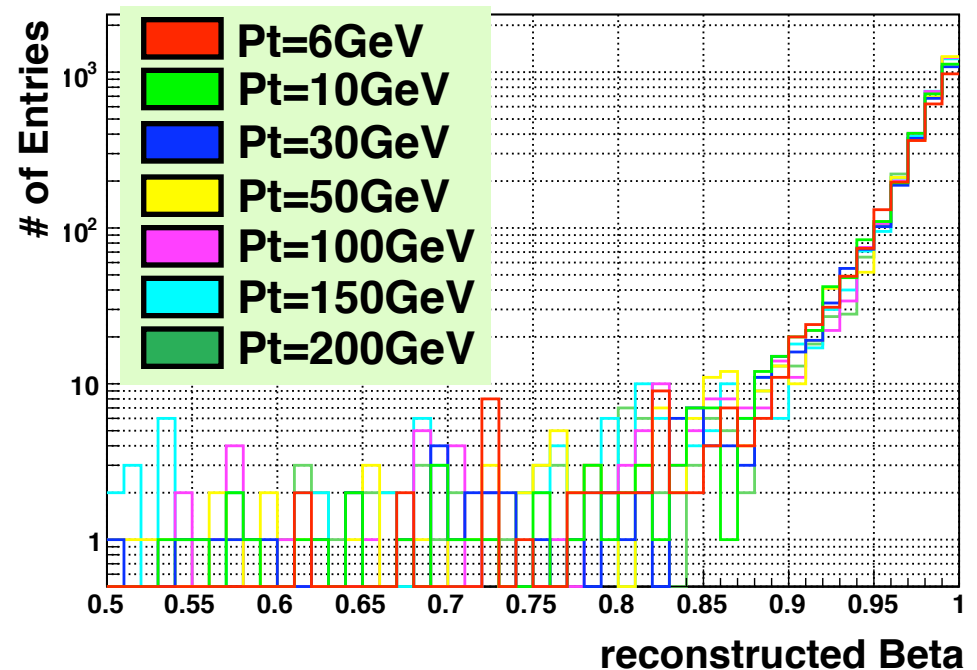
# SingleMuonを用いたBackgroundの量の概算

第一近似として、Pt=6,10,30,50,100,150,200[GeV]のSingleMuonを用いて、スレプトン探索でのBackgroundの量を、SingleMuonを出す物理過程の断面積に焼き直して概算した。

single muonを出す物理事象の微分断面積[nb/GeV]

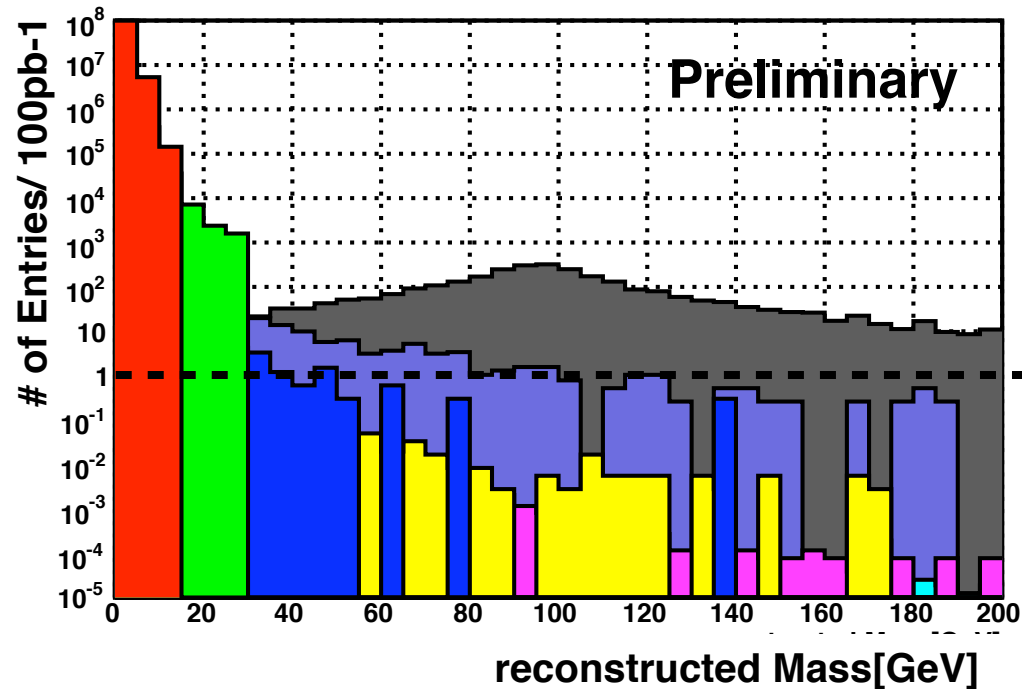
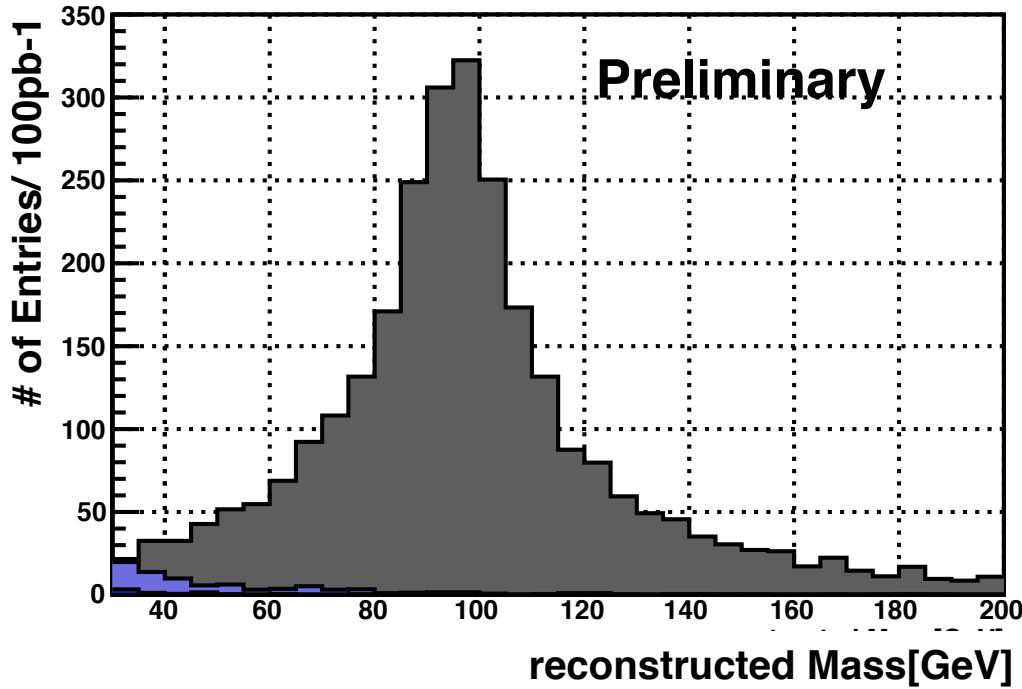


reconstructed Beta分布



Pt	6 GeV	10 GeV	30 GeV	50 GeV
生成断面積	$2.64 \times 10^5$ nb	23.2 nb	2.32 nb	$4.83 \times 10^{-2}$ nb
100 GeV	150 GeV	200 GeV		
$4.72 \times 10^{-3}$ nb	$5.77 \times 10^{-4}$ nb	$1.59 \times 10^{-4}$ nb		

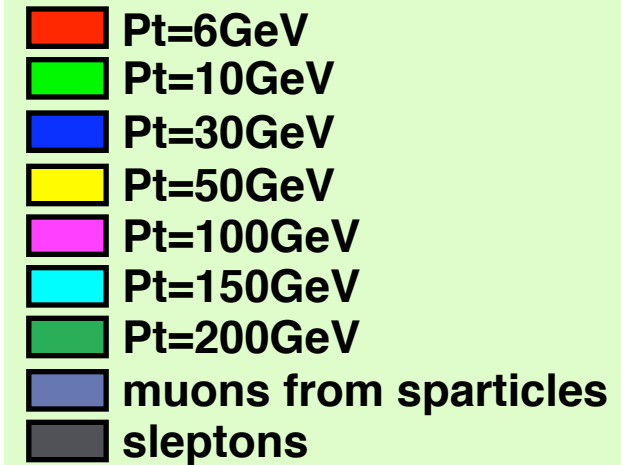
(飛跡をrefitする場合での結果)



Preliminary

✓ SingleMuon Eventsを用いてバックグラウンドの量を見積もった結果, Sleptonの質量(~100GeV)を含む広い質量領域でミュオンバックグラウンドによる影響はほぼ無い.

✓ 次はttbar, W, Zの標準模型事象を使って, ミュオンバックグラウンドの影響を精度良く調べる.



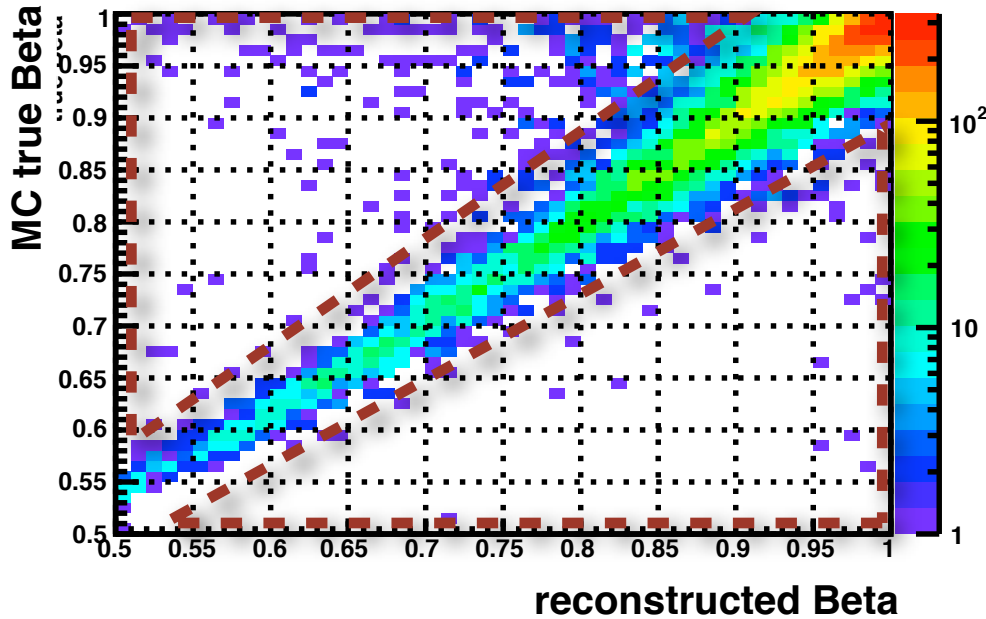
# Summary & Outlook

- ✓ ATLAS実験のEvent Filter用のツール (TrigMuonEF) をベースに長寿命スレプトンの速度 $\beta$ を再構成するプログラムを作った.  $\beta$ は正しく, 高いefficiencyで再構成できた.
- ✓ しかし, Triggerとしてこのプログラムを使うには時間が掛かりすぎるので, パターン認識を省くアルゴリズムを作成したところ, 処理速度は8~9倍速くなった.
- ✓ だが, スレプトンの質量の分解能が悪くなり, また, low  $\beta$ のスレプトンに対する,  $\beta$  reconstruction efficiencyが低くなった.
  - > efficiencyの向上と更なる処理時間短縮のための改良をする必要がある.
- ✓ 第一近似として, SingleMuonサンプルを用いたBackgroundの量を概算したところ, スレプトンによるシグナルの方が多く, ミューオンバックグラウンドによる影響は少ない.
  - > 今後, ttbar, W, Zの標準模型事象を用いて精度良く確かめる.

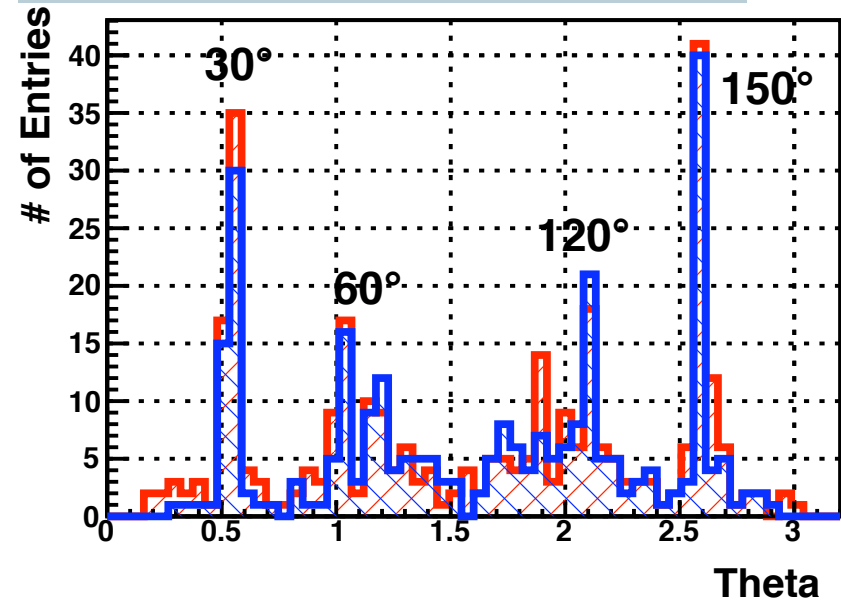
**Backup Slides**

# About $\beta$ mis-reconstruction

## recBeta vs trueBeta



## Theta (|recBeta-trueBeta|>0.1)



— with pattern rec.  
— without pattern rec.

✓  $\beta$ のmis-reconstructionについて詳しいstudyを行った.

-->  $\beta$ のmis-reconstructionはsleptonの質量の再構成に影響を与える.

✓  $|\text{recBeta} - \text{trueBeta}| > 0.1$ を満たすslepton trackの角度 $\theta$  (ビーム軸とtrackの間の角度)の分布を見たところ,  $\theta=30^\circ, 60^\circ, 120^\circ, 150^\circ$ にピークを持つ分布が見られた. (--> 右図)

✓ MDTの構造上, これらの角度方向に進んだsleptonの $\beta$ を間違えることがある.



- 例えば, Barrel部でmuon/sleptonが $\theta=60^\circ$ の方向に進んだとき, 飛跡と粒子が通ったMDTの関係は(団子や焼き鳥の様な)「串刺し」みたいになる.
- この場合, 粒子の通った全てのMDTのドリフト半径(アノードと飛跡の間の距離)は同じ.
- また, この様な場合では,  $\beta$ が変化しても上手く(小さい $\chi^2$ で)飛跡がフィットできるので, (→串刺し状態では無い場合と大きく違う点) $\chi^2$ の値に大きな違いが見られない(右下図).
- 結果として,  $\beta$ を間違えることがある.

