

Reaction: $\gamma p \rightarrow K^0 \Sigma^+$ plots and code snippets

Isabella Illari¹

¹Physics Department, Columbian College of Arts & Sciences, The George Washington University, Washington, D.C. 20052

Reaction: $K^0\Sigma^+$ photoproduction

- for K^0 examine the following reaction: $\gamma p \rightarrow K^0\Sigma^+$

$$K^0 \rightarrow \pi^0\pi^0 \quad (\text{BR} = 69\%)$$

$$\rightarrow \pi^+\pi^- \quad (\text{BR} = 31\%)$$

$$\Sigma^+ \rightarrow p\pi^0 \quad (\text{BR} = 52\%)$$

$$\rightarrow n\pi^+ \quad (\text{BR} = 48\%)$$

$$\pi^0 \rightarrow 2\gamma \quad (\text{BR} = 99\%)$$

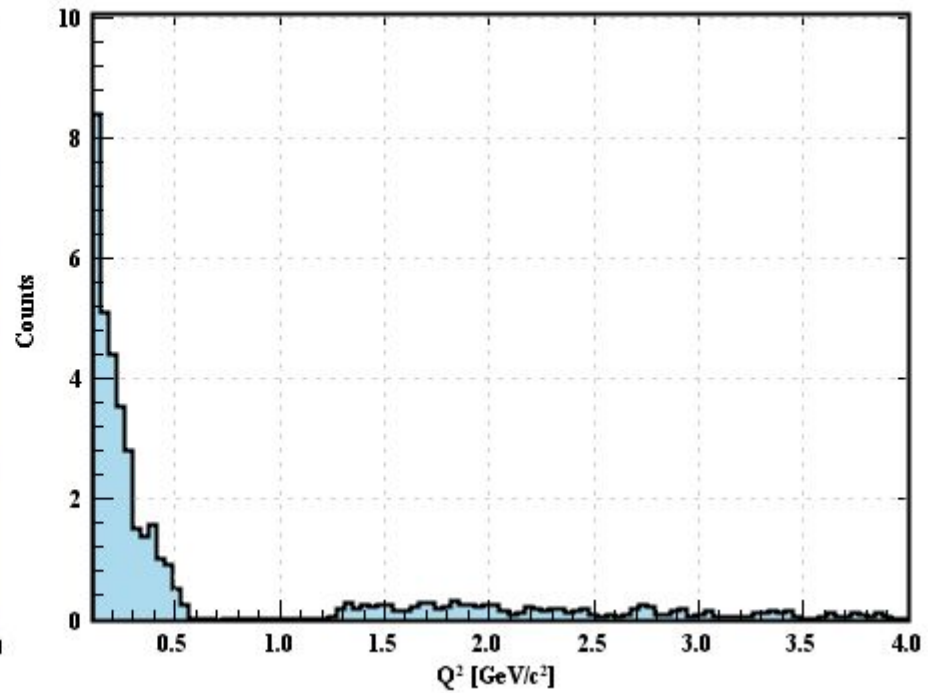
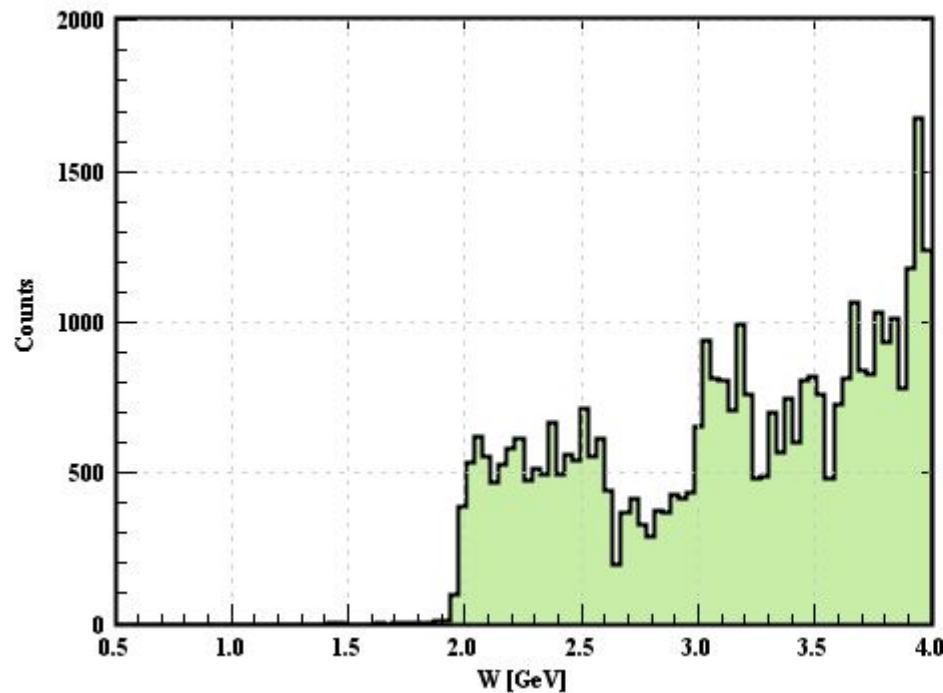
$$\text{final state} : \gamma\gamma p\pi^+\pi^-$$

MC: PYTHIA (Dr. Harut Avagyan)

initial state: γp

set: $Q^2_{\min} = 0.00001$ & $Q^2_{\max} = 15.0$

987425 events



PYTHIA MC: generates lund file
lund file then put thru GEMC, create a evio file, convert to hipo, analyze
said hipo file

Analysis Code

- **Event Selection/Filters:**

- $\gamma\gamma p\pi^+\pi^-$ (and any other p/m or n particle)
- $\gamma\gamma$ (and any other p/m or n particle)
- $\gamma\gamma p\pi^+\pi^-$ (ONLY)

```
//e:gamma:gamma:p:pip:pim explicitly detected + any other positive/negative or neutral particle
EventFilter eventFilter = new EventFilter("11:22:22:2212:211:-211:Xn:X+;X-");

// e:gamma:gamma explicitly detected + any other positive/negative or neutral particle
EventFilter eventFilter1 = new EventFilter("11:22:22:X+:X-:Xn");
//EventFilter eventFilter1 = new EventFilter("22:22:X+:X-:Xn");

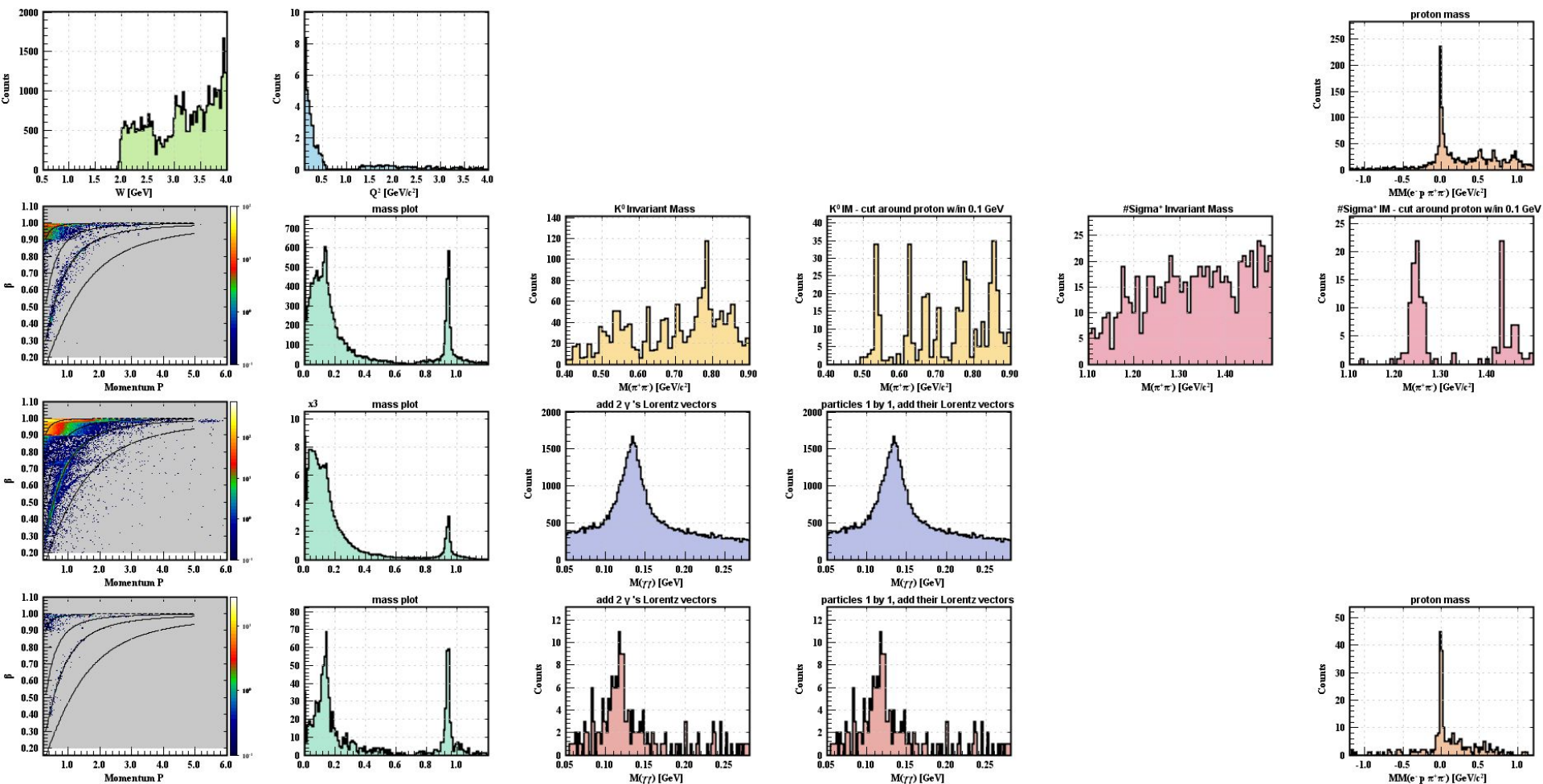
// e:gamma:gamma:p:pip:pim:gamma:gamma NOTHING else
EventFilter eventFilter2 = new EventFilter("11:22:22:2212:211:-211");
```

- **Shema/Banks:**

- REC::Particle
- REC:Scintillator
- REC::Event
- REC:ForwardTagger

```
//----- Banks-----//
Event event = new Event();
Bank particles = new Bank(reader.getSchemaFactory().getSchema("REC::Particle"));
Bank scintillator = new Bank(reader.getSchemaFactory().getSchema("REC::Scintillator"));
Bank even = new Bank(reader.getSchemaFactory().getSchema("REC::Event"));
Bank ftagger = new Bank(reader.getSchemaFactory().getSchema("REC::ForwardTagger"));
```

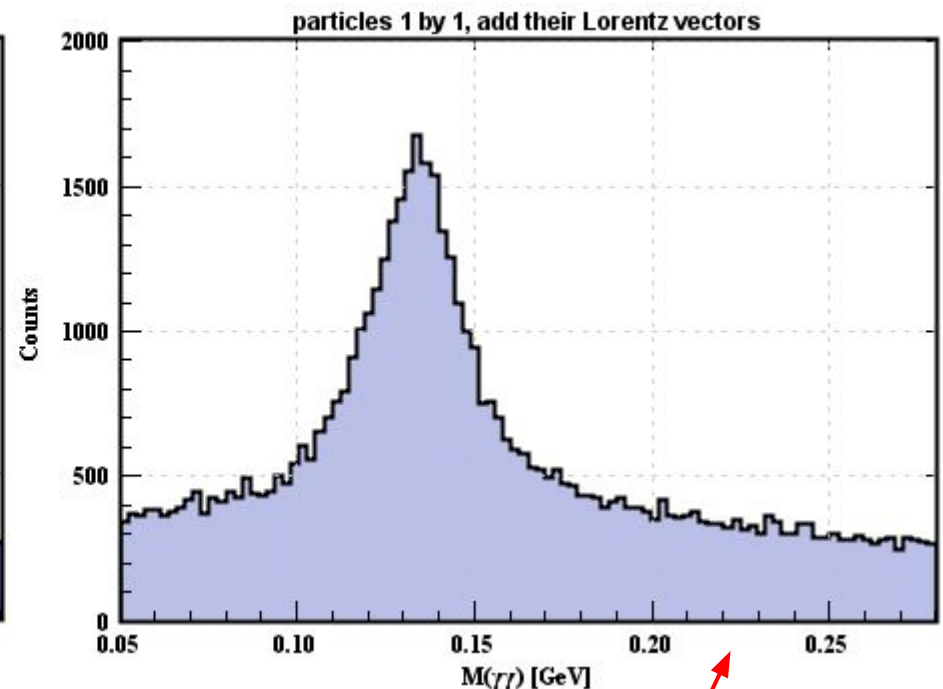
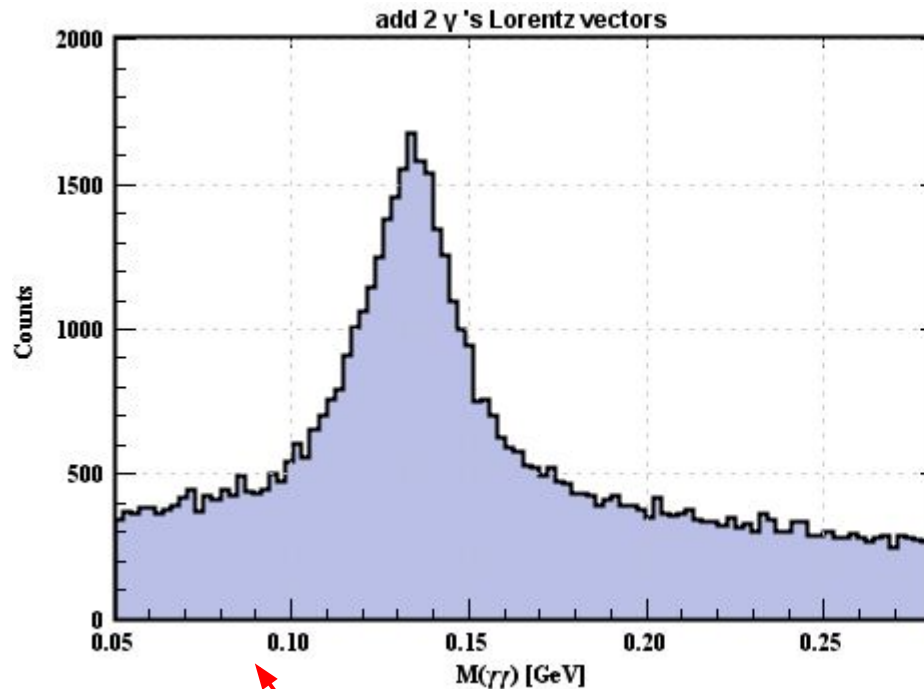
All plots at a glance (MC data)



Yellow: K^0 (Filter 1); Red: Σ^+ (Filter 1); Purple: π^0 (Filter 2); RedOrange: π^0 (Filter 3); BlueGreen: mass (all 3 Filters); Orange: proton mass (Filter 1 & 3)

Compare Invariant Mass Calc (MC)

Second Filter; π^0 mass: 0.13498 GeV



```
//----- Second Filter -----//
if(eventFilter1.isValid(physEvent)==true) {
    filterCounter1++;

    // The particle Lorentz vector is constructed by adding 2 photon's Lorentz vectors
    // [22,0] - first photons (or 0'th) and [22,1] - second photons (or index 1)
    org.jlab.jnp.physics.Particle pi0 = physEvent.getParticle("[22,0]+[22,1]");
    hpi0.fill(pi0.mass());

    // Another way to calculate invariant mass:
    // Get particles one by one, and add their Lorentz vectors
    org.jlab.jnp.physics.Particle gamma0 = physEvent.getParticleByPid(22, 0);
    org.jlab.jnp.physics.Particle gamma1 = physEvent.getParticleByPid(22, 1);
    LorentzVector vL_pi0 = LorentzVector.from(gamma0.vector());
    vL_pi0.add(gamma1.vector());
    hpi0vec.fill(vL_pi0.mass());
}
```

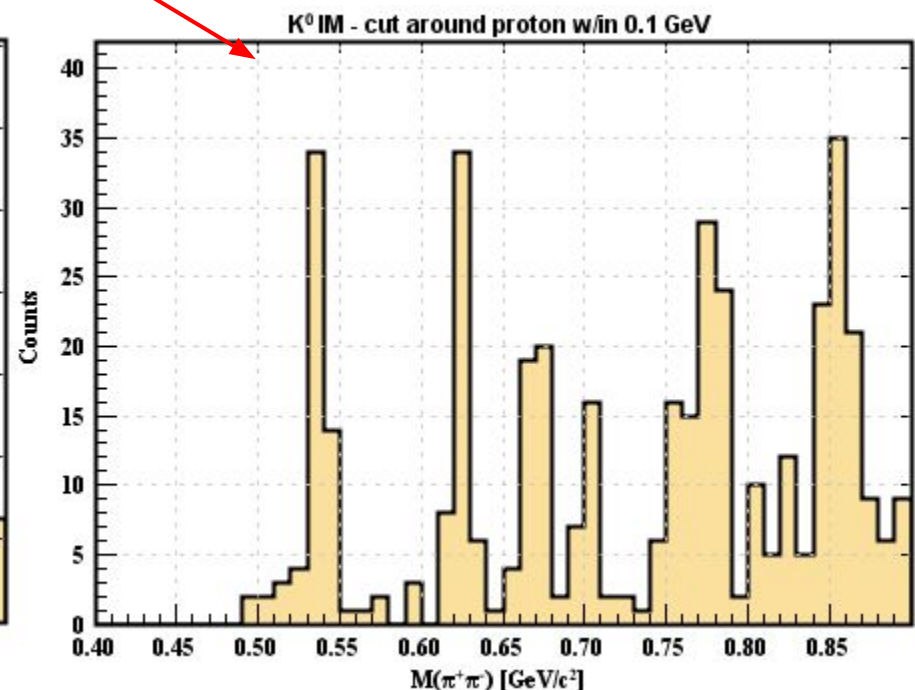
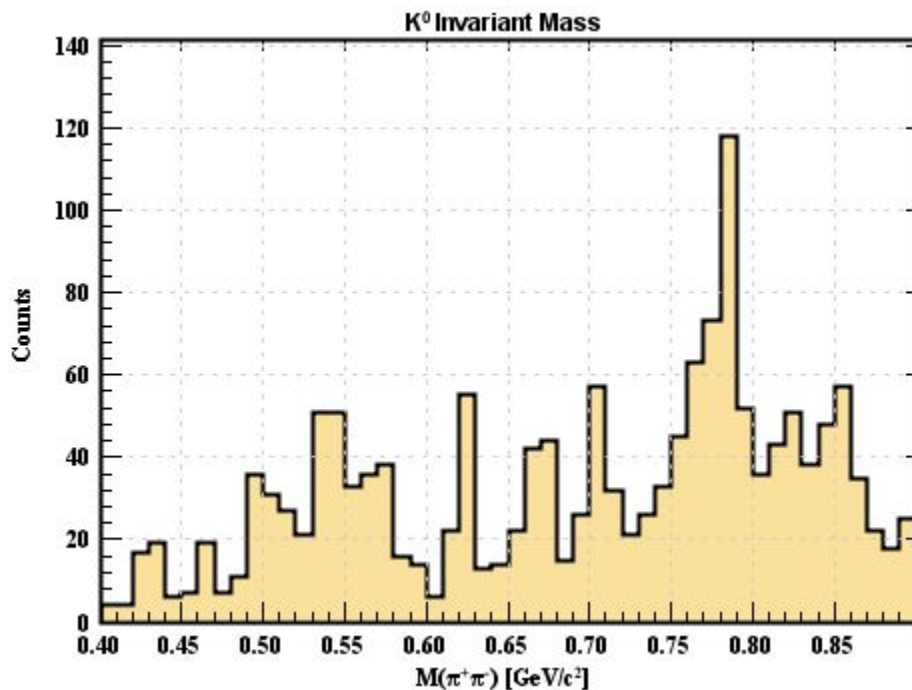
K^0 Invariant Mass (MC)

Filter 1; K^0_s mass = 0.49761 GeV, visible shift to right

Right: **cut** around "pro" particle w/in 0.1 GeV

org.jlab.jnp.physics.Particle pro =

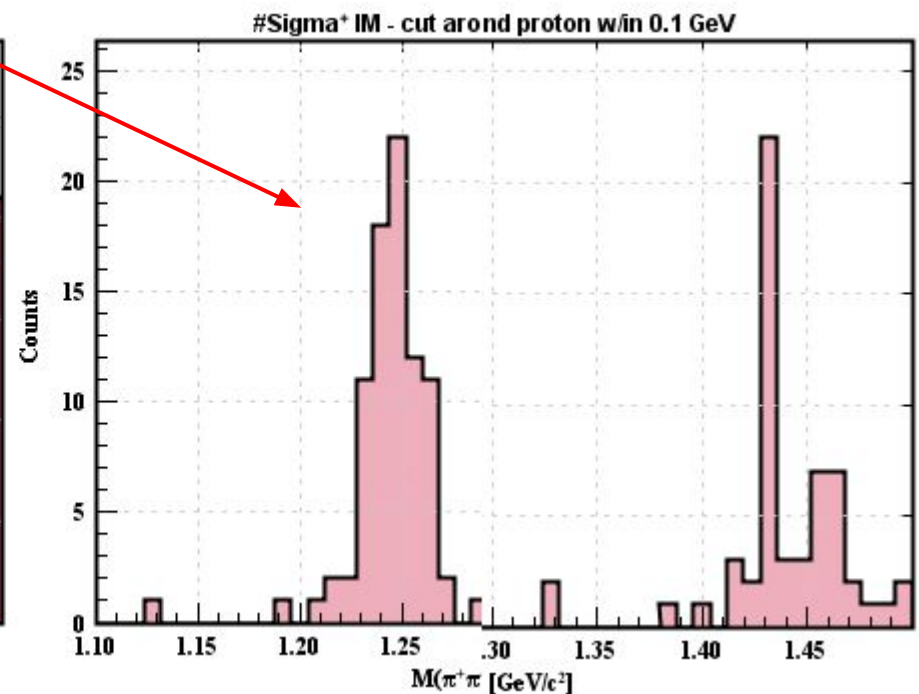
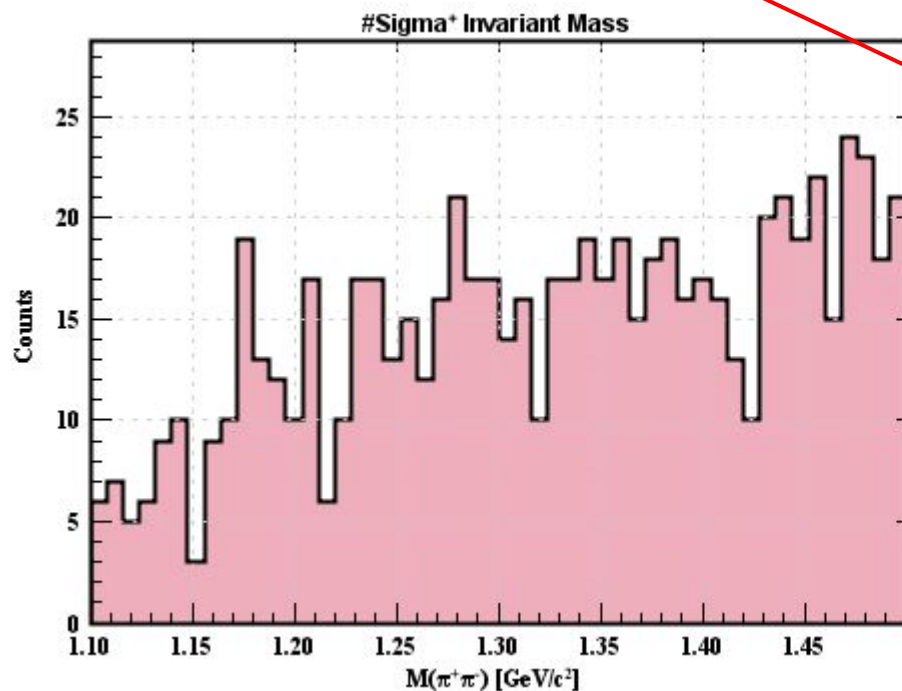
```
physEvent.getParticle("[b]+[t]-[11,0]-[2212,0]-[211,0]-[-211,0]");
```



Σ^+ Invariant Mass (MC)

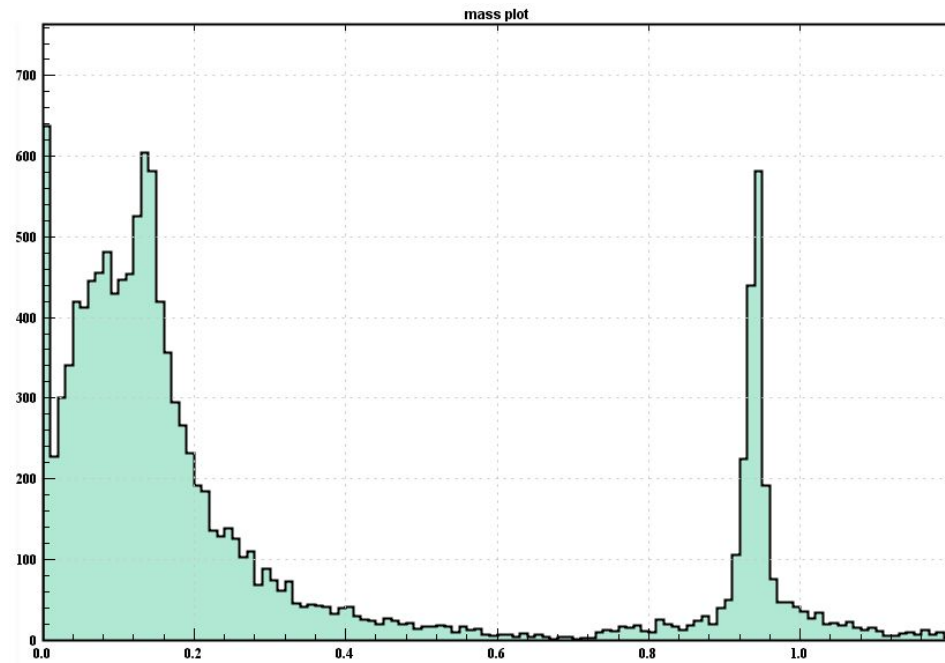
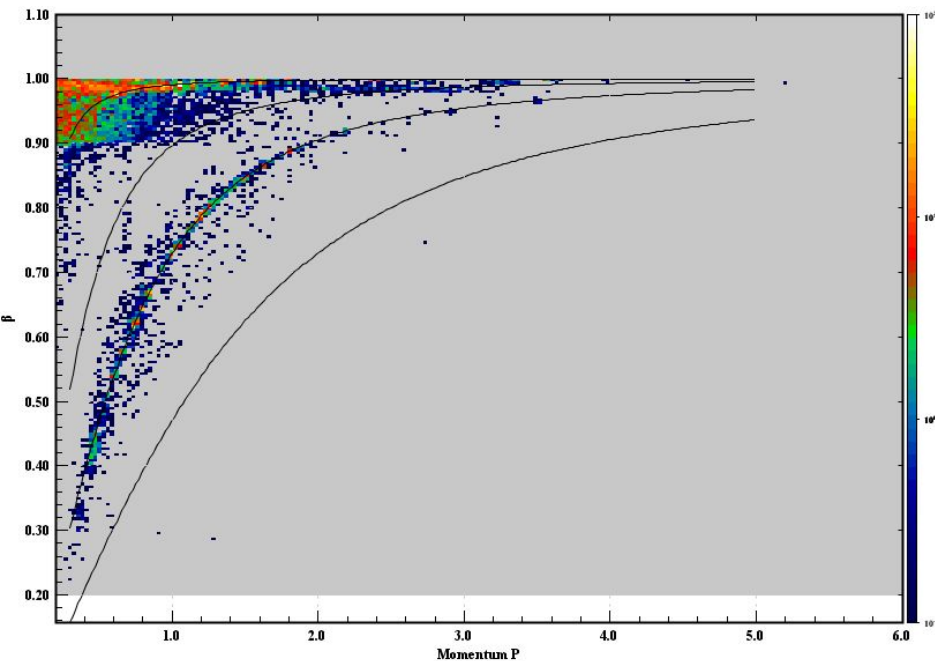
Filter 1; Σ^+ mass = 1.18900 GeV, visible shift to right

Right: same cut as previous slide



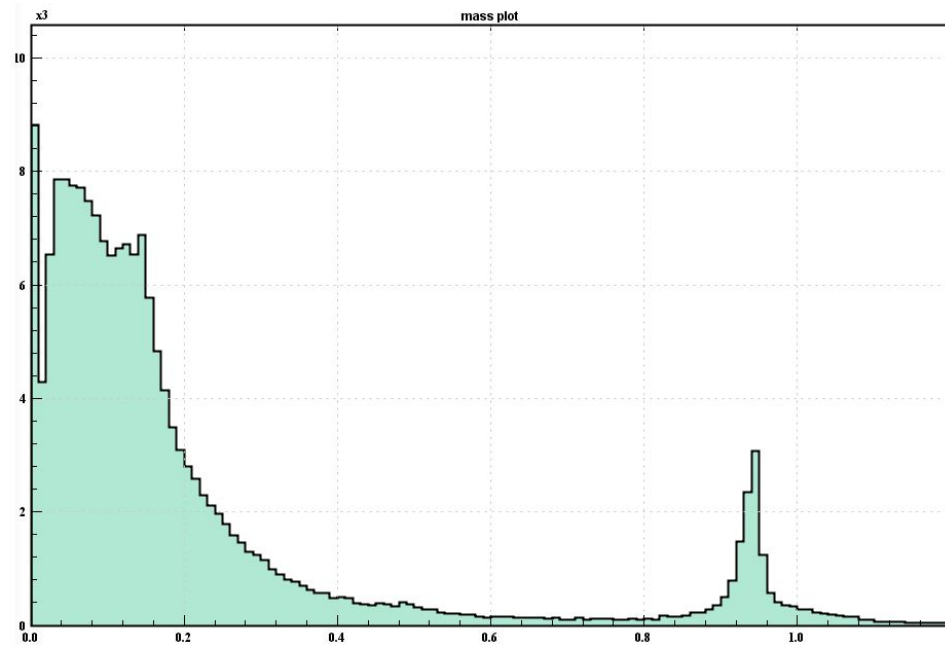
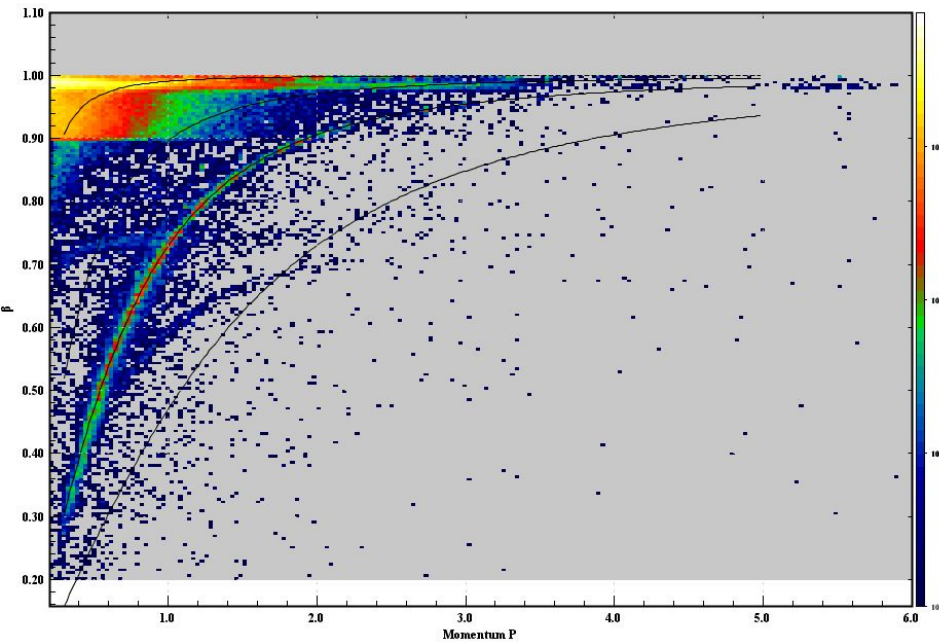
Beta, momentum, mass calc (MC)

Filter 1: $\gamma\gamma p \pi^+ \pi^-$ (and any other p/m or n particle)



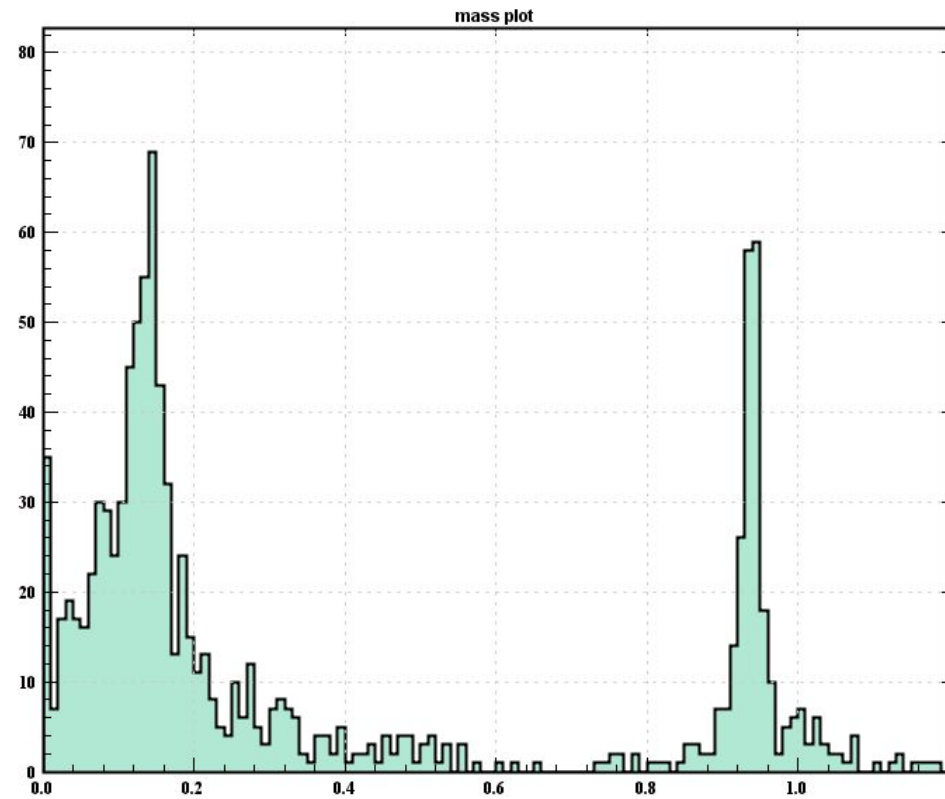
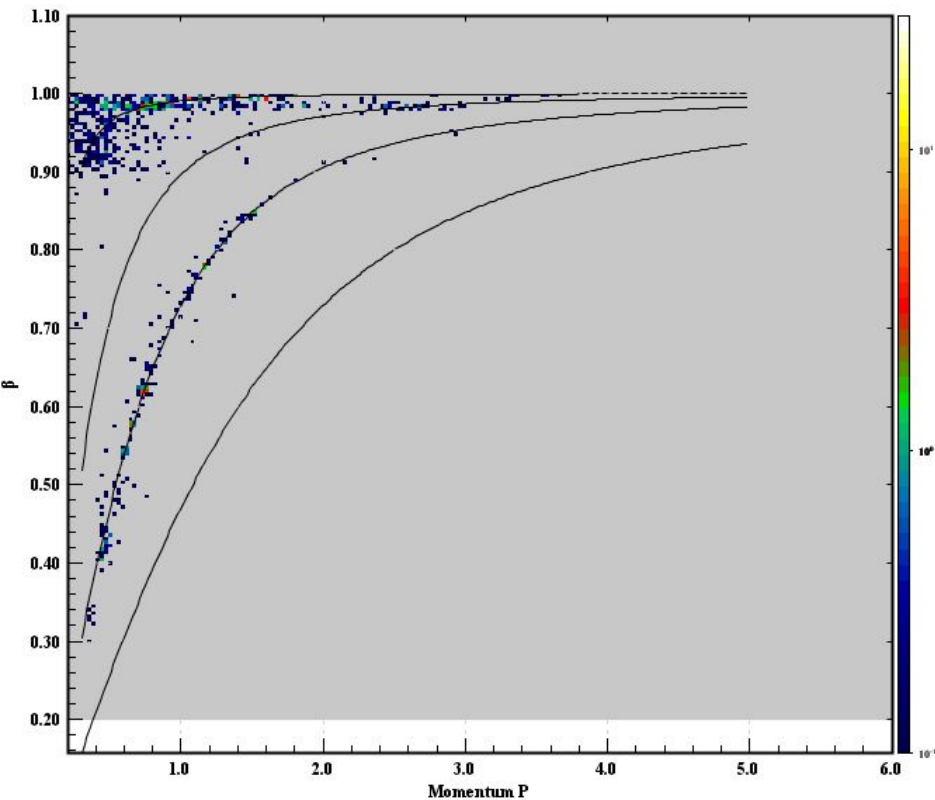
Beta, momentum, mass calc (MC)

Filter 2: $\gamma\gamma$ (and any other p/m or n particle)



Beta, momentum, mass calc (MC)

Filter 3: $\gamma\gamma p\pi^+\pi^-$ (ONLY)



Beta, momentum, mass calc

```
// get # of columns in REC::Particle bank
int prows = particles.getRows();
//Read every val, plot its beta entry in the bank
for(int i=0; i<prows; i++) {
    //Beta vals
    float b = particles.getFloat("beta", i);
    //Momentum vals px, py, pz and calculates P
    float px = particles.getFloat("px", i);
    float py = particles.getFloat("py", i);
    float pz = particles.getFloat("pz", i);
    double p = Math.sqrt(px*px + py*py + pz*pz);

    //if beta less than 1 then calculate the particle mass
    if(b<1.0/*&&cpid<2.0&&cpid>-1.0*/) {
        double m = (p / b)* Math.sqrt(1 - b*b);
        //Cut for spikes, pi masses, proton masses
        if(Math.abs(m-0.939)>0.001&&Math.abs(m-0.1395)>0.0005) {
            mass.fill(m);
            BvsP.fill(p, b);
        }
    }
}
```

Further Results

- issues w/ REC:ForwardTagger timing?
- RG-A check skim4
 - electron in ForwardDetector
 - compare to MC results