- Photon reconstruction requires:
  - Energy, cluster location (from calorimetry)
  - Vertex location (from tracking)
- Previously I only looked at step 1

DChargedTrack -> DVertex -> “DPhoton”

DVertex_factory

- Input: Each track has a position = position of closest approach to beamline
- Group tracks together if they are within X sigma's of each other in (t,z)

```
GROUP_NUM_SIGMAS_TIME = 100.0; // originally was 3, but changed as temporary measure
GROUP_NUM_SIGMAS_Z    = 100.0; // originally was 3, but changed as temporary measure
```

- Each group gets a DVertex with position equal to average of track positions

Thrown particles

DMCThrown:

q:	x(cm):	y(cm):	z(cm):	E(GeV):	t(ns):	p(GeV/c):	theta(deg):	phi(deg):	type:
+1	0.0	0.0	72.4	0.917	-999.000	0.906	26.237	117.080	8
-1	0.0	0.0	72.4	1.737	-999.000	1.731	16.350	175.330	9
+0	0.0	0.0	72.4	0.049	-999.000	0.049	54.473	0.377	1
+0	0.0	0.0	72.4	0.384	-999.000	0.384	5.307	-149.683	1
+1	0.0	0.0	72.4	1.749	-999.000	1.743	6.583	29.965	8
-1	0.0	0.0	72.4	4.060	-999.000	4.057	5.390	-46.139	9
+1	0.0	0.0	72.4	1.044	-999.000	0.458	41.102	-42.565	14

Reconstructed tracks

DTrackTimeBased:

q:	x(cm):	y(cm):	z(cm):	E(GeV):	t(ns):	p(GeV/c):	theta(deg):	phi(deg):
-1	47.6	-16.9	61.4	0.713	0.000	0.699	37.952	70.447
+1	12.7	-2.3	136.5	0.969	0.000	0.241	87.045	-100.189
-1	6.9	7.4	94.3	0.668	0.000	0.653	40.322	137.259
-1	-0.9	5.4	102.8	0.233	0.000	0.187	119.488	-170.181
+1	-6.0	-8.5	79.3	691.726	0.000	691.726	16.252	144.725
-1	0.0	0.0	72.5	4.098	0.000	4.095	5.386	-46.480
+1	0.1	-0.1	70.6	1.730	0.000	1.724	6.390	30.869
+1	0.1	-0.1	70.6	1.964	0.000	1.726	6.390	30.875

Reconstructed vertex

DVertex:

x:	y:	z:	t:	Ntracks:
7.94	-2.12	80.16	0.00	6

Thrown photons

DMCThrown:

q:	x(cm):	y(cm):	z(cm):	E(GeV):	t(ns):	p(GeV/c):	theta(deg):	phi(deg):	type:
+1	0.0	0.0	72.4	0.917	-999.000	0.906	26.237	117.080	8
-1	0.0	0.0	72.4	1.737	-999.000	1.731	16.350	175.330	9
+0	0.0	0.0	72.4	0.049	-999.000	0.049	54.473	0.377	1
+0	0.0	0.0	72.4	0.384	-999.000	0.384	5.307	-149.683	1
+1	0.0	0.0	72.4	1.749	-999.000	1.743	6.583	29.965	8
-1	0.0	0.0	72.4	4.060	-999.000	4.057	5.390	-46.139	9
+1	0.0	0.0	72.4	1.044	-999.000	0.458	41.102	-42.565	14

Reconstructed photons

DNeutralParticleHypothesis:

PID:	tProj:	q:	x(cm):	y(cm):	z(cm):	E(GeV):	t(ns):	p(GeV/c):	theta(deg):	phi(deg):
1	1.250253	+0	7.9	-2.1	80.2	0.029	0.000	0.029	47.979	3.067
1	2.679564	+0	7.9	-2.1	80.2	0.034	0.000	0.034	16.116	135.583
1	0.684761	+0	7.9	-2.1	80.2	0.454	0.000	0.454	19.045	148.358
1	1.250681	+0	7.9	-2.1	80.2	0.034	0.000	0.034	29.356	147.405
1	8.394413	+0	7.9	-2.1	80.2	0.277	0.000	0.277	16.518	148.995
1	5.947721	+0	7.9	-2.1	80.2	0.043	0.000	0.043	16.306	-170.764
1	0.229491	+0	7.9	-2.1	80.2	0.365	0.000	0.365	5.891	-154.980
1	0.308300	+0	7.9	-2.1	80.2	0.377	0.000	0.377	6.123	-50.343

Bad vertex reconstruction leads to poor angular resolution

But what if we don't just group all the tracks together?

Try:

```
GROUP_NUM_SIGMAS_TIME = 5.0;
GROUP_NUM_SIGMAS_Z    = 5.0;
```

we get 4 vertices

```
DVertex:
      x:     y:     z:     t: Ntracks:
-------------------------------------
   0.96   0.63  77.60  -2.64        2
   0.45   0.25  65.41  -0.21        1
  -0.18  -1.25  71.60   9.01        1
 -17.57   5.26  65.25   0.20        2
```

too many vertices -> too many photons

Currently:

DChargedTrack -> DVertex -> “DPhoton”

One bad track makes everything else bad

Need:

DChargedTrack -> (track quality cuts
PID) -> DVertex -> “DPhoton”

Second step done by user

How?