

# Ssd Embedding status

Jonathan Bouchet  
S&C 02/20

# Steps

- (taken from StSvtSimulationMaker)
- 1) loop over GEANT hits and take the track associated
- 2) IdealToReal() method : project the GEANT track in the real wafer geometry
- If the localPosition of the tHelix projection is not in the wafer active area, loop over neighboring ladders and wafers
- 3) Fill the StSsdPointList with these global projection of track
- 4) Convert the hits into fired strips (method from St\_sls\_Maker)
- 5) Fill slsStrip table containing these information

# Reco. Chain

**Sls** : convert the hits from Geant into fired strips



Spa : do the daq behaviour



Spt : hit reconstruction



BFChain :  
ssd : **sls** spa spt

## plain Simulation



Will be included in macro bfcMixer\_TpcSvt.C

**SsdEmbed** : convert the hits from Geant into fired strips with real geom



Spa : do the daq behaviour



**SsdAdd** : merge datasets from simu+real

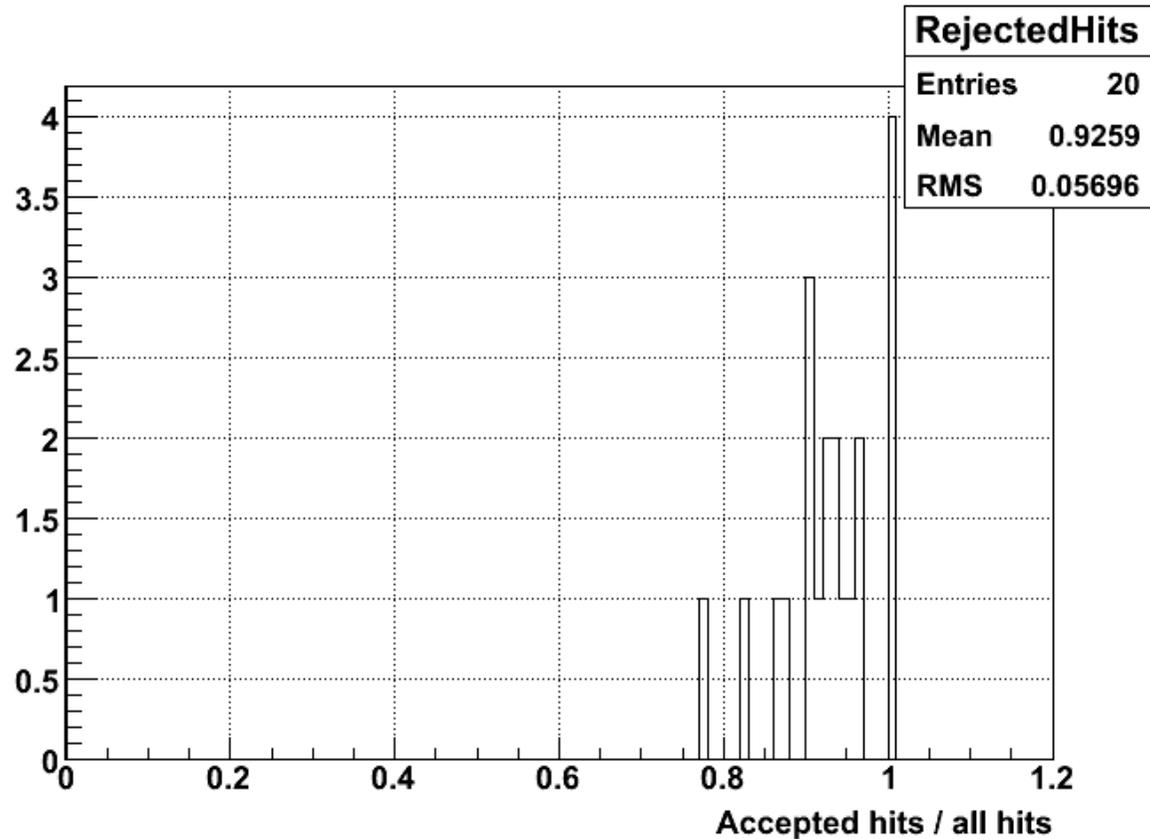


BFChain :  
ssdE : **SsdEmbed** spa **SsdAd** spt

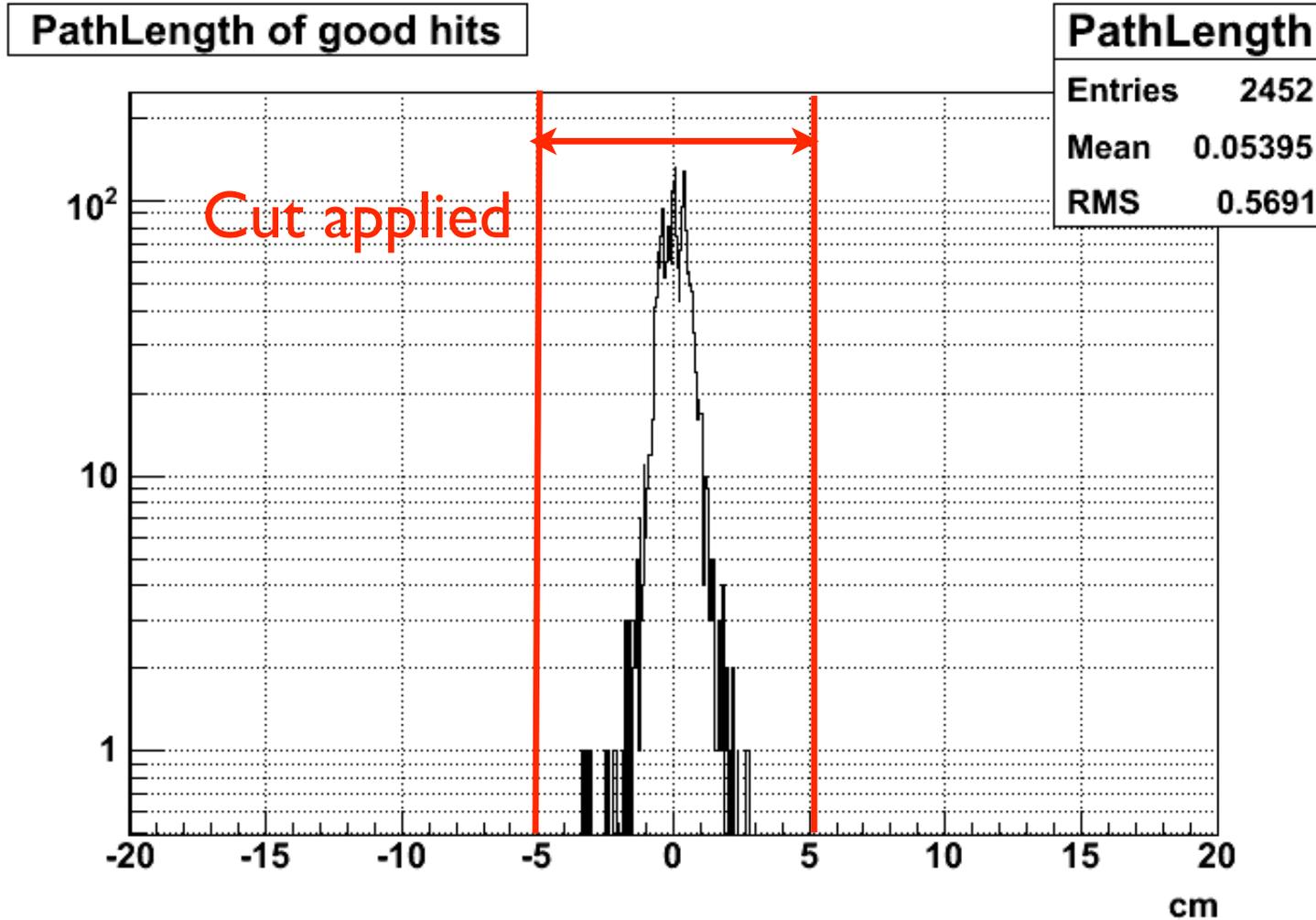
## Embedding

# Projection of the helix to the real geom

- Even after the loop on the neighboring wafers, I don't get the requirement that the projection is in the active area
- I don't keep these hits
- less than 10 % are rejected



# Pathlength found for the good hits



## 2nd part of the chain : merging of 2 datasets (StSsdAddMaker)

- Both datasets(=list of strips) are zero & pedestal subtracted
- from the simulation :
  - ➡ `St_sls_strip *sls_strip = (St_sls_strip *)GetDataSet("SsdEmbedding/.data/sls_strip");`
  - ➡ Then Spa takes this list coming from the EmbeddingMaker
  - ➡ And produces a spa\_strip list (ie, zero & pedestal subtracted)
- from the real data :
  - ➡ `St_spa_strip *spa_strip = dynamic_cast<St_spa_strip *> (SpaStrip->Find("spa_strip"));`
- Still working on the merging function

# Inactive areas in wafer

