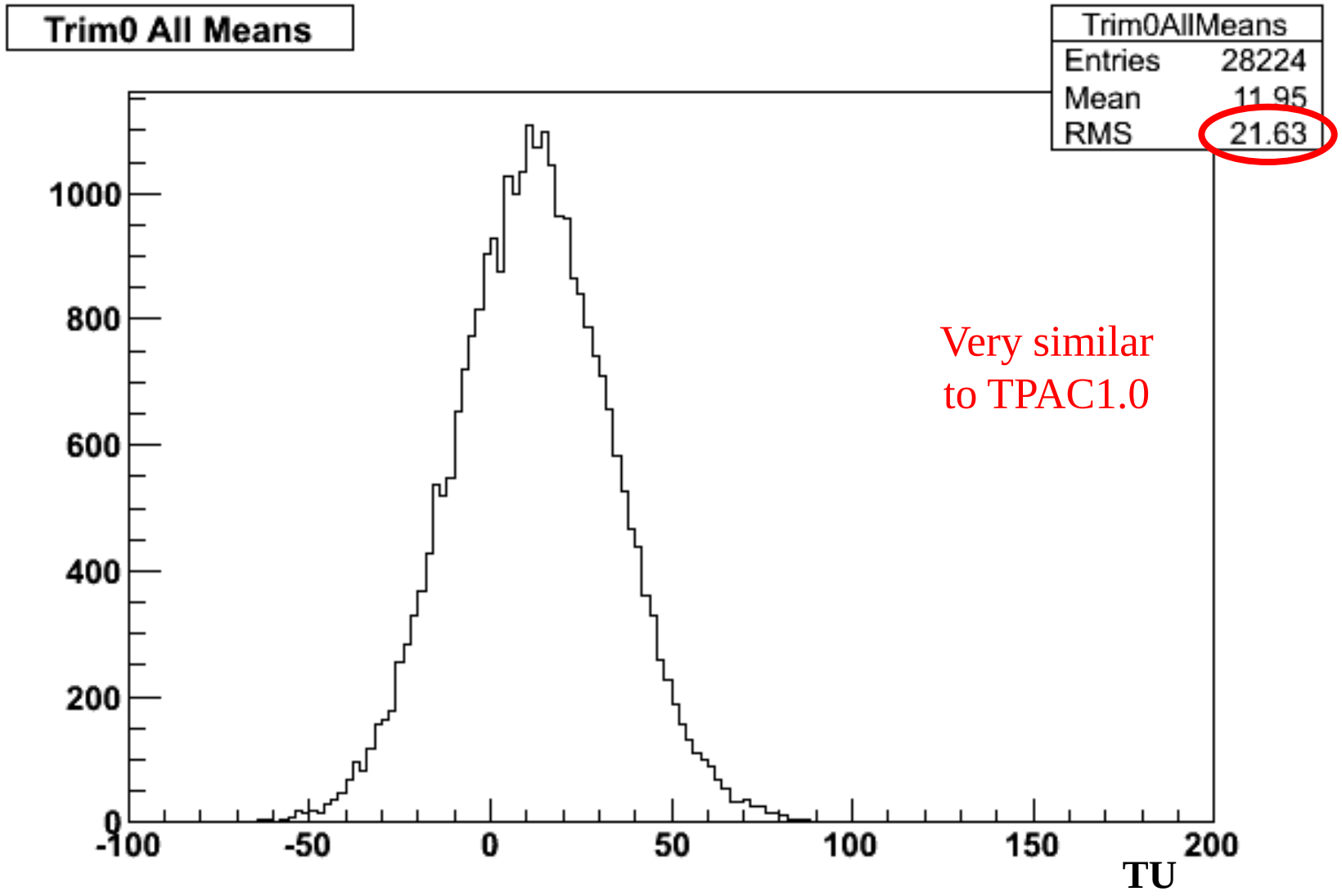
TPAC1.2 trimming

Paul Dauncey

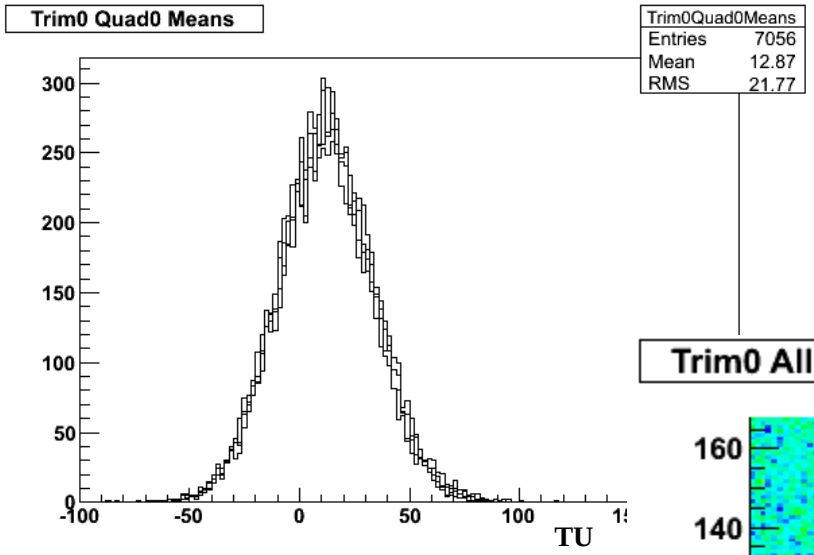
Trimming Sensor #29

- Same method as for TPAC1.0
  - Fix trims to some values
  - Unmask 42 pixels per readout region
  - 4 regions = 168 pixels total in each run
  - Do threshold scan to determine mean and width
  - Do 168 runs to cover complete sensor
- Adjust trim to narrow mean distribution
  - Can only go up from trim=0
  - Need to pick target value near top of range of means
  - Trim each pixel to get mean as close as possible to target
  - Need to iterate; not yet complete for this sensor

Set all trims=0 initially: mean

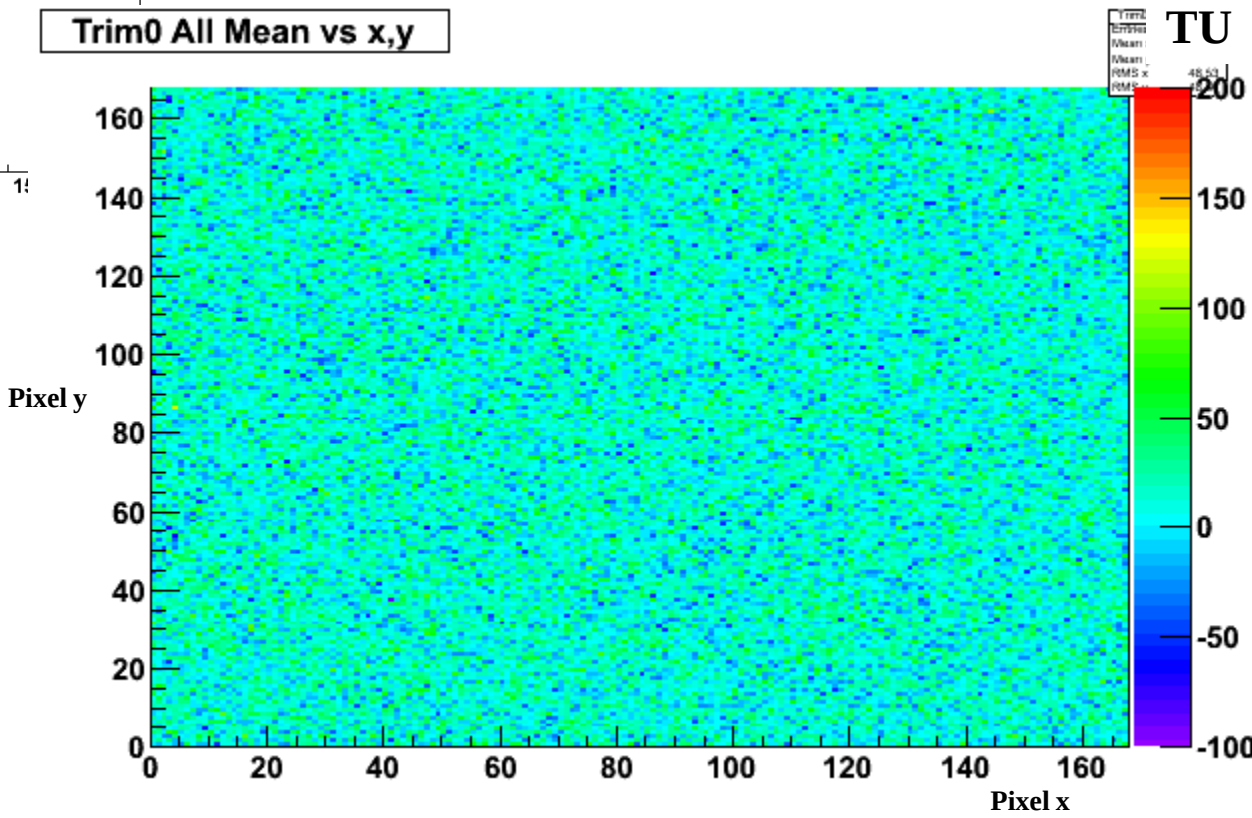


No obvious position dependence for mean

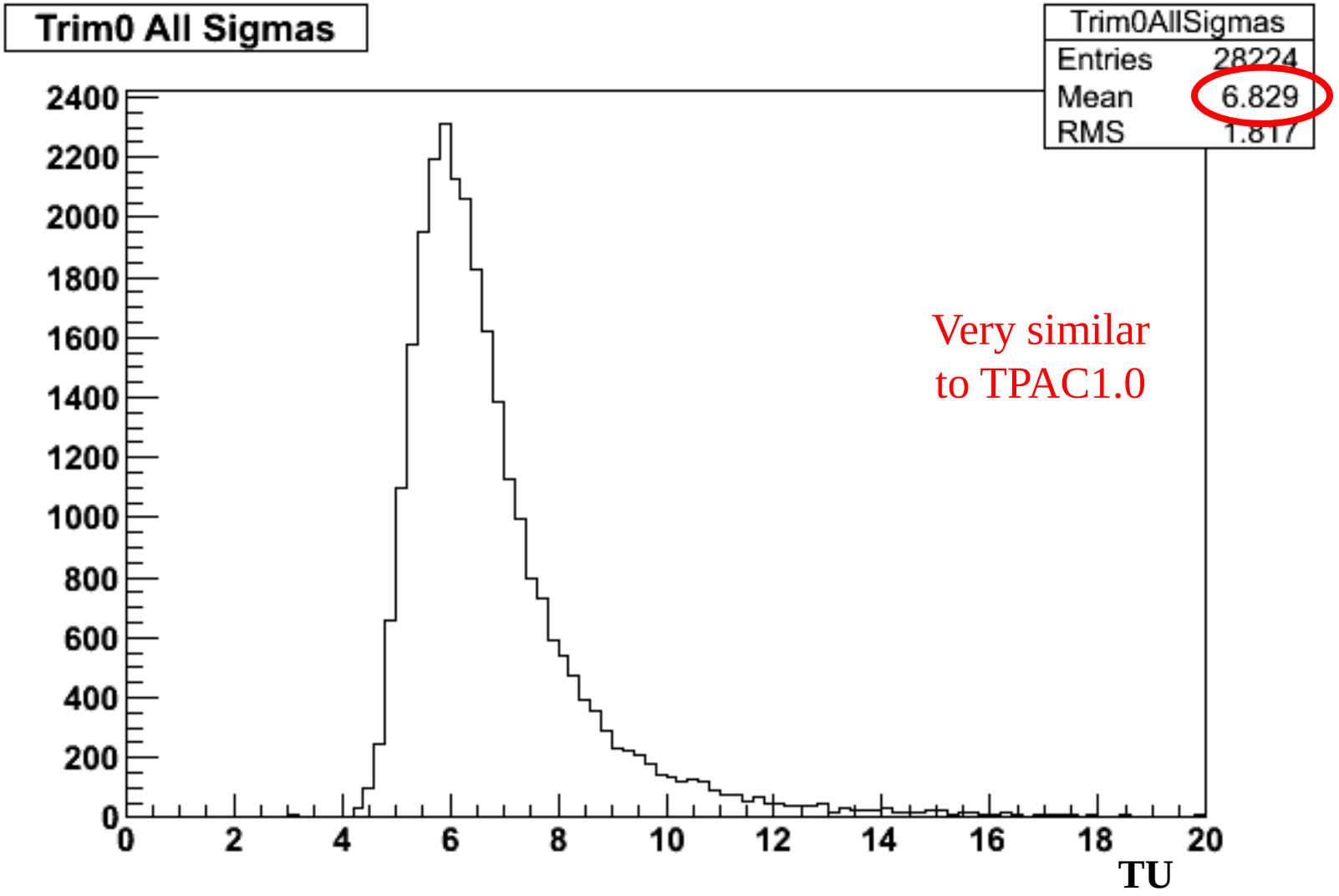


Each of four quadrants

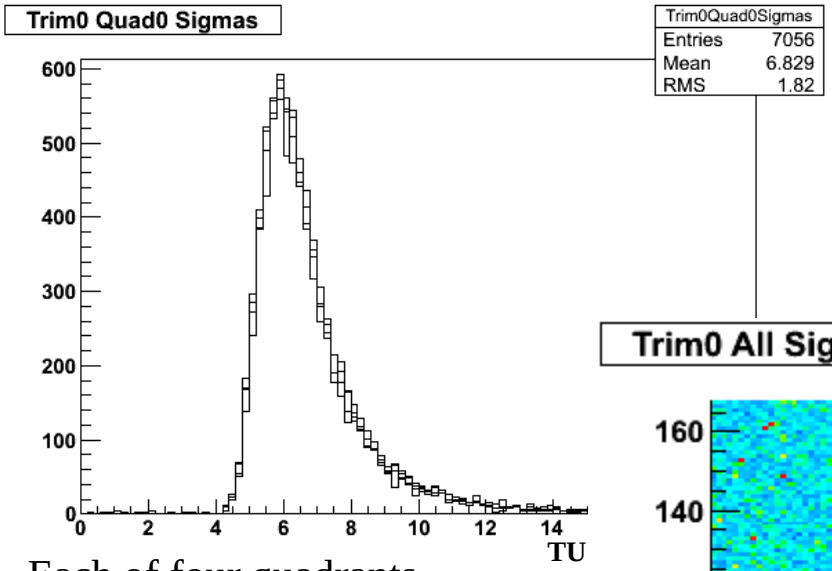
Physical position on sensor



Trims=0: width

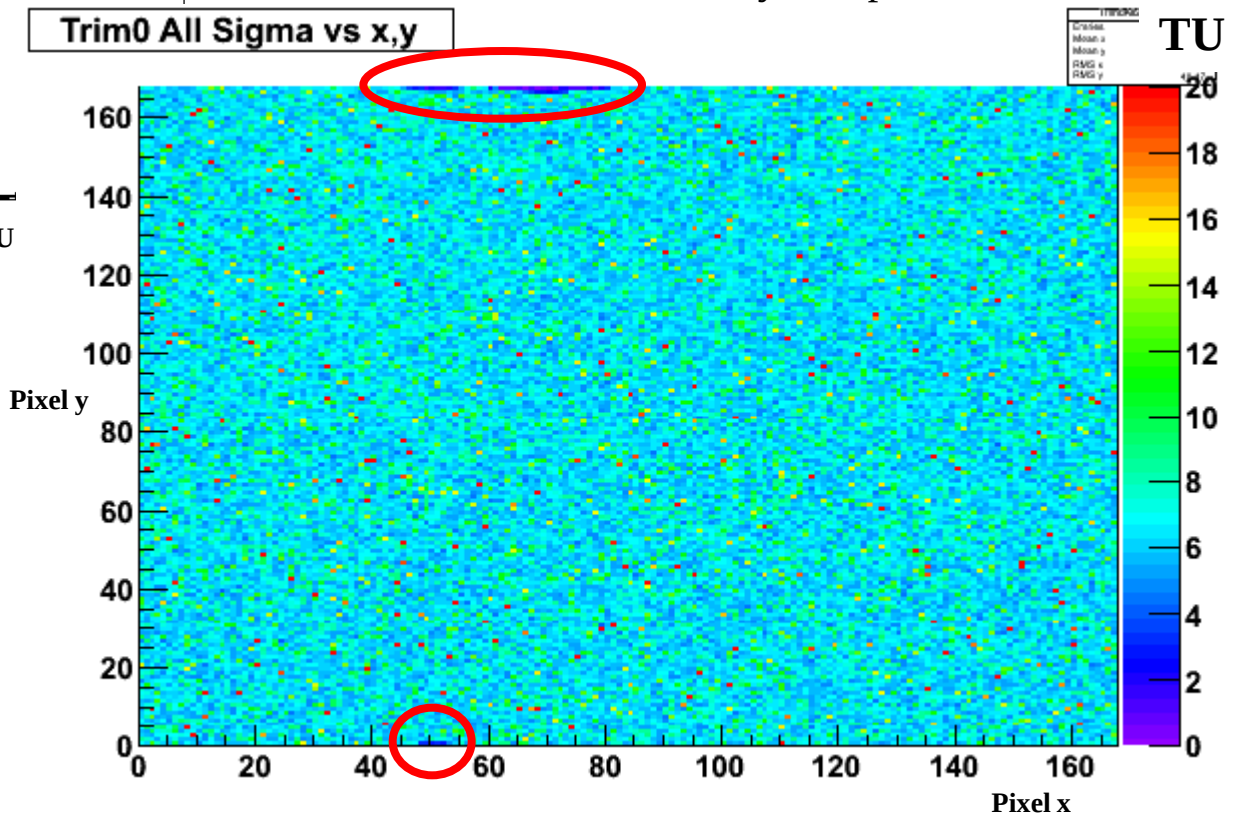


No obvious position dependence for width

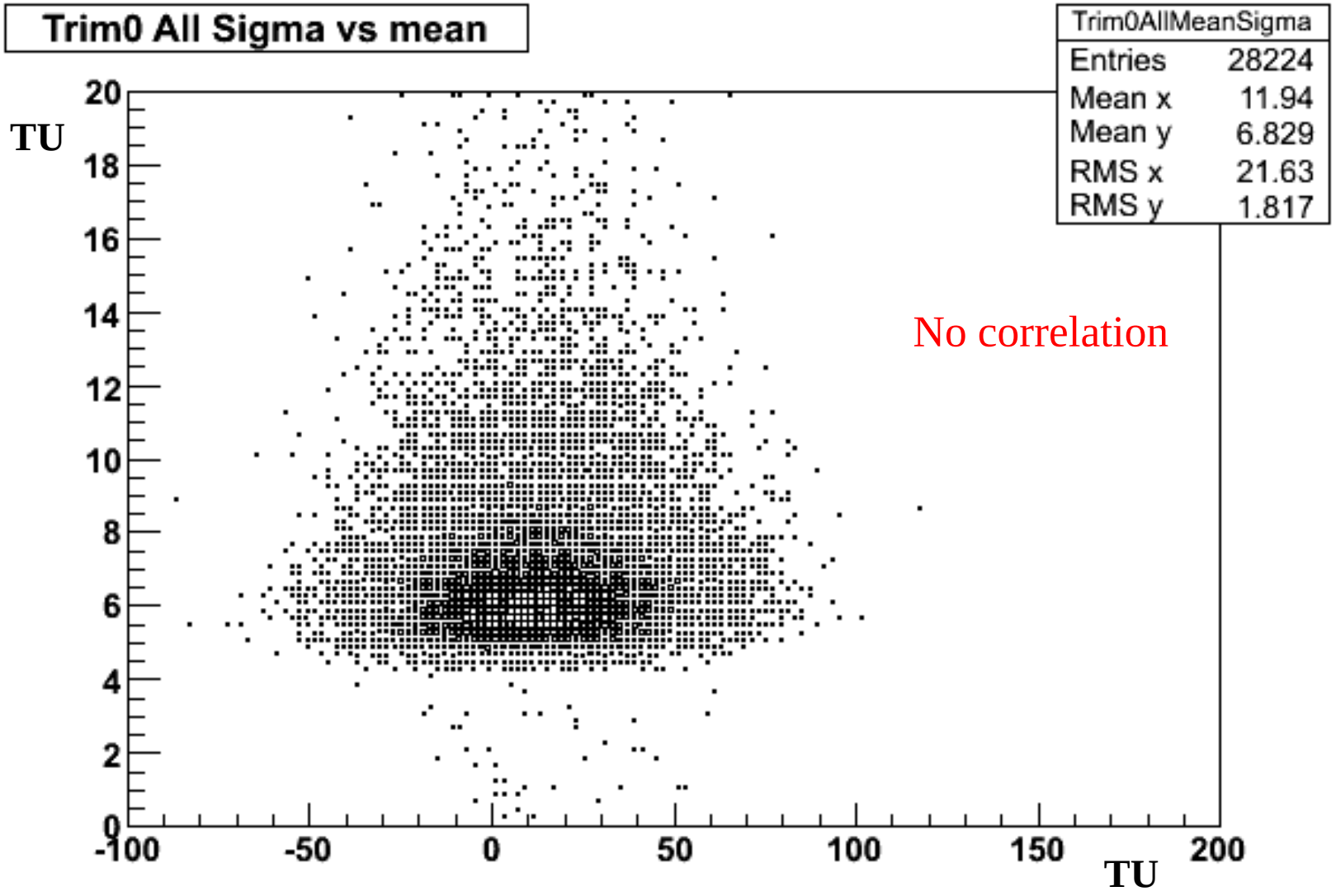


Each of four quadrants

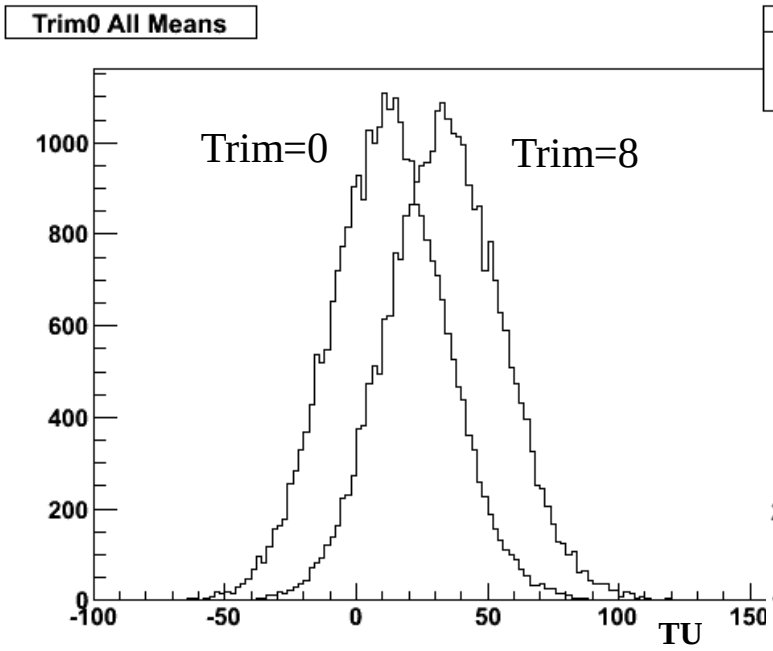
Physical position on sensor



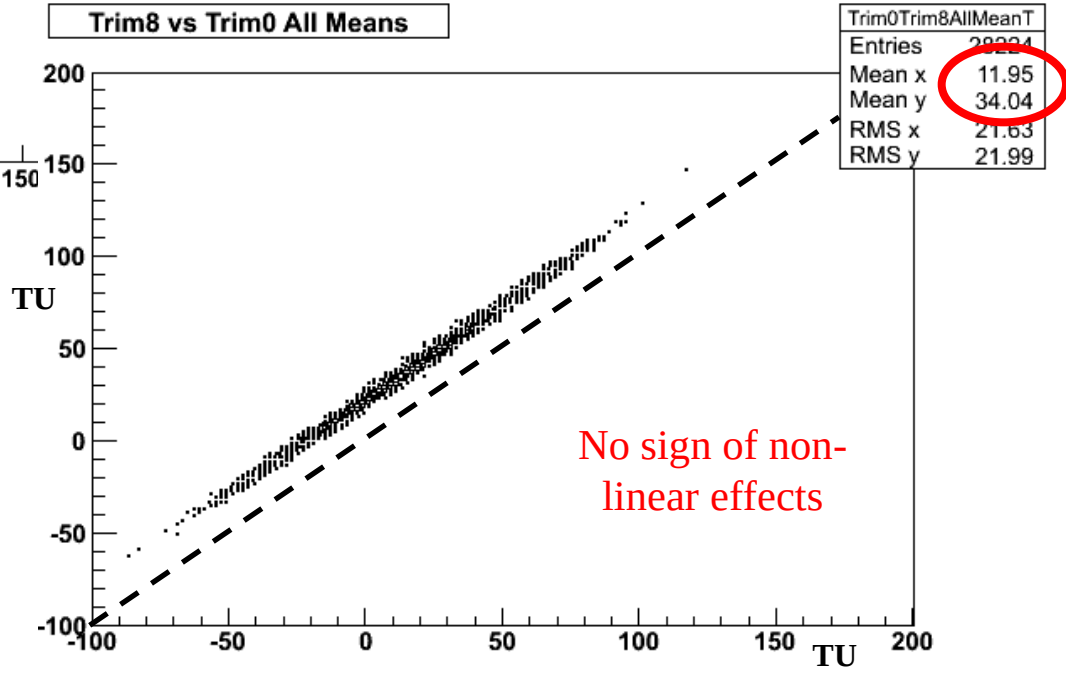
Trim=0: width vs mean



Trim=8: Shift of means

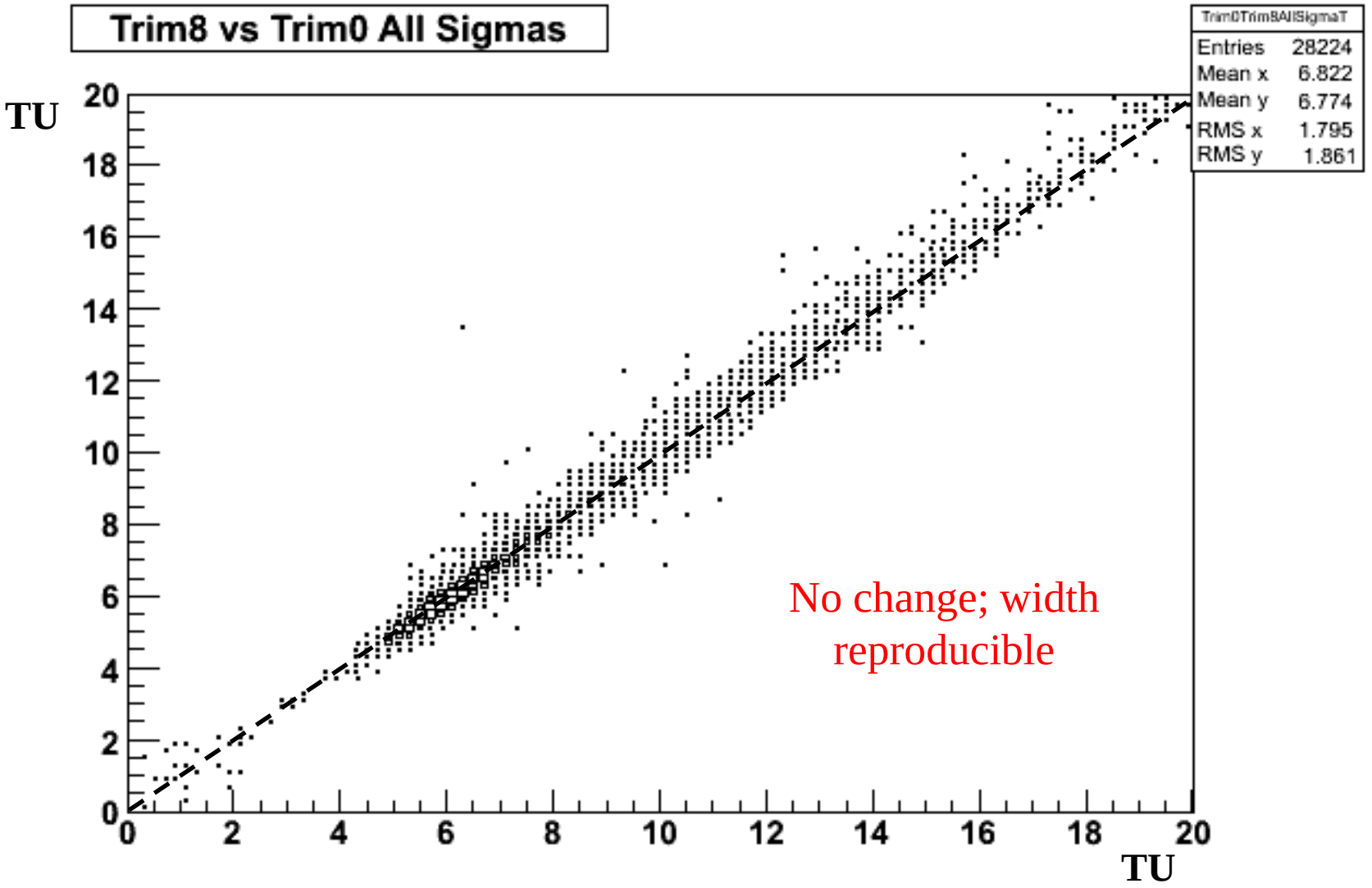


Shift of average = 22TU

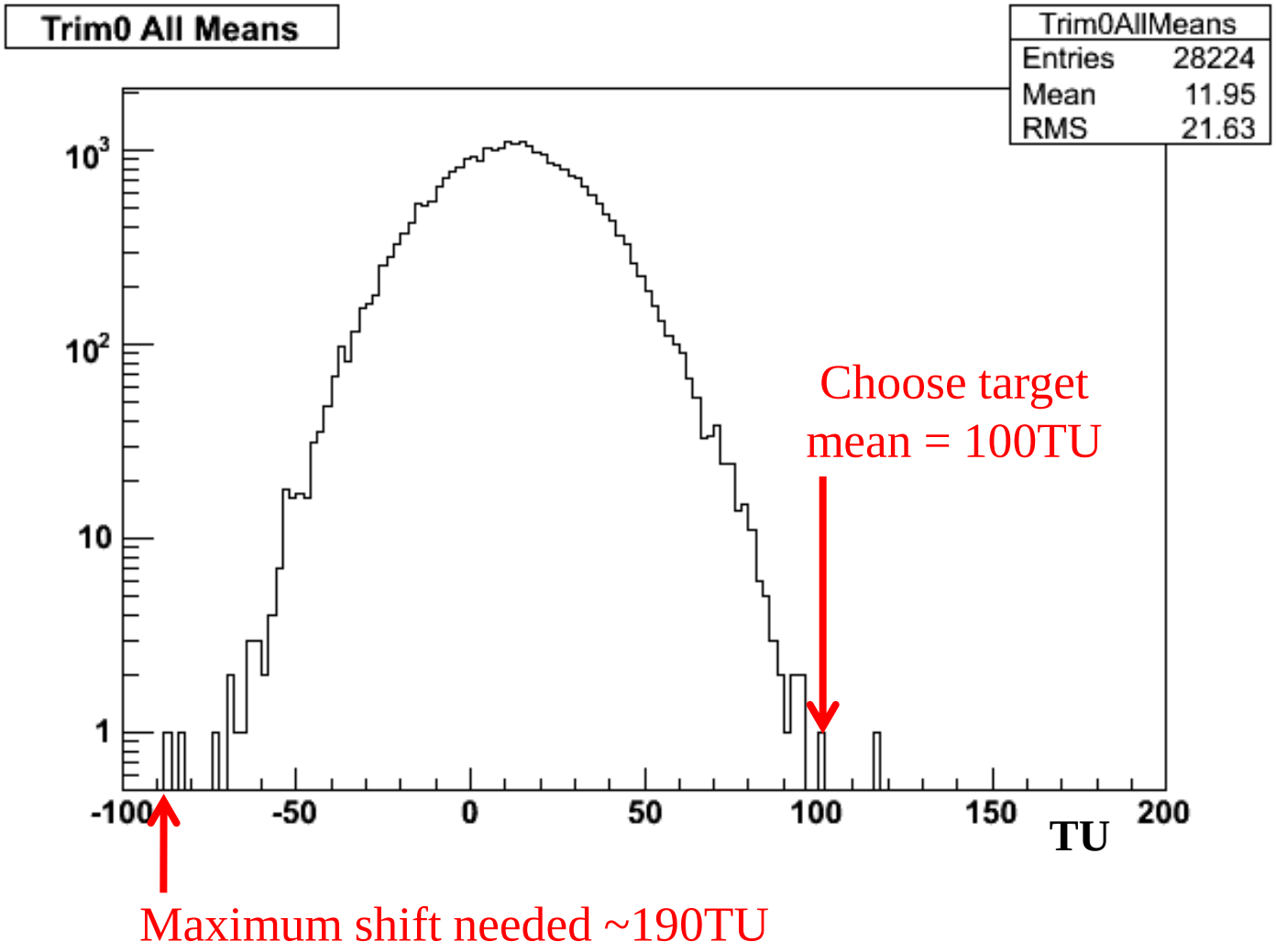


No sign of non-linear effects

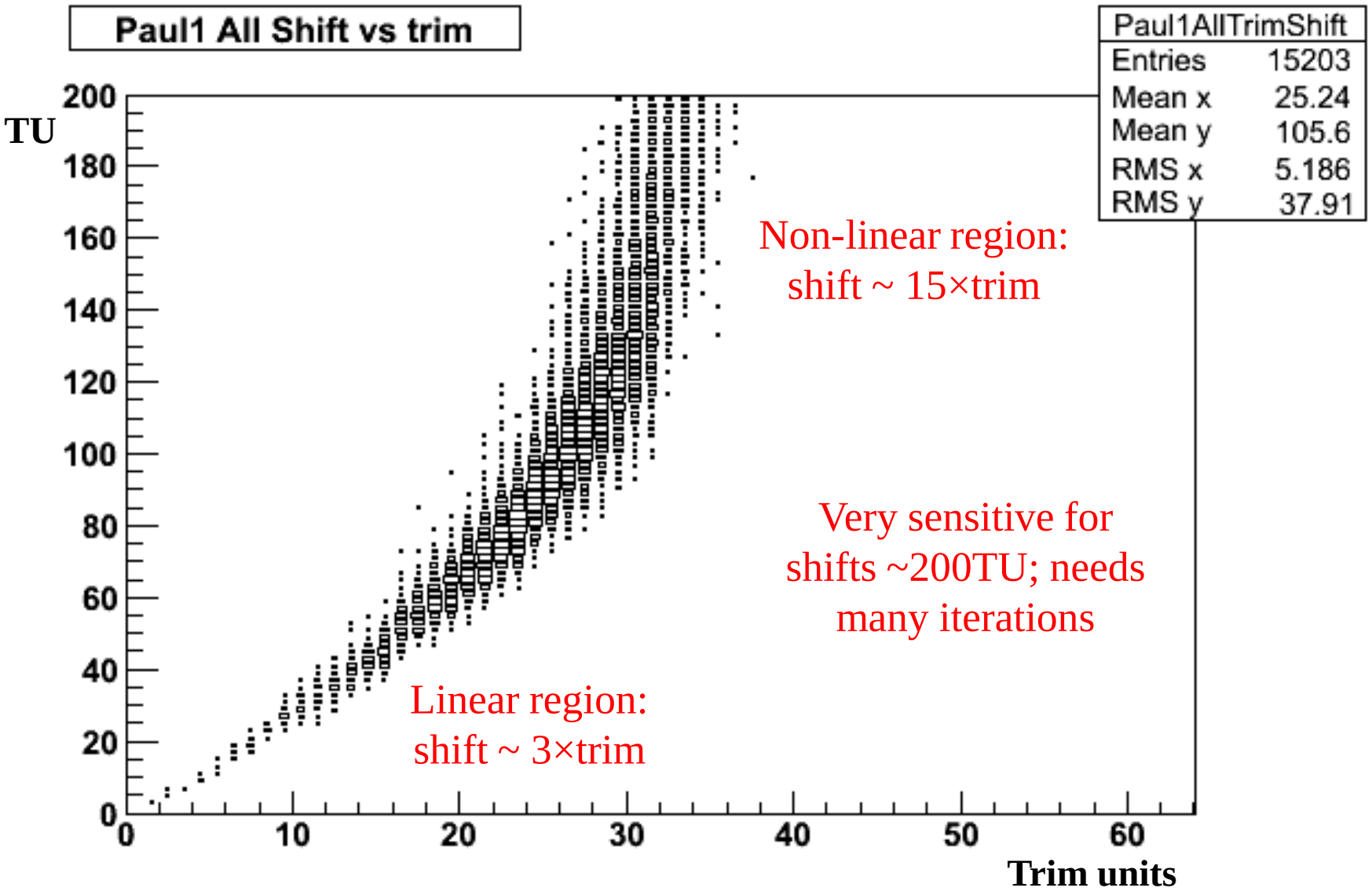
Trim=8: Comparison of widths



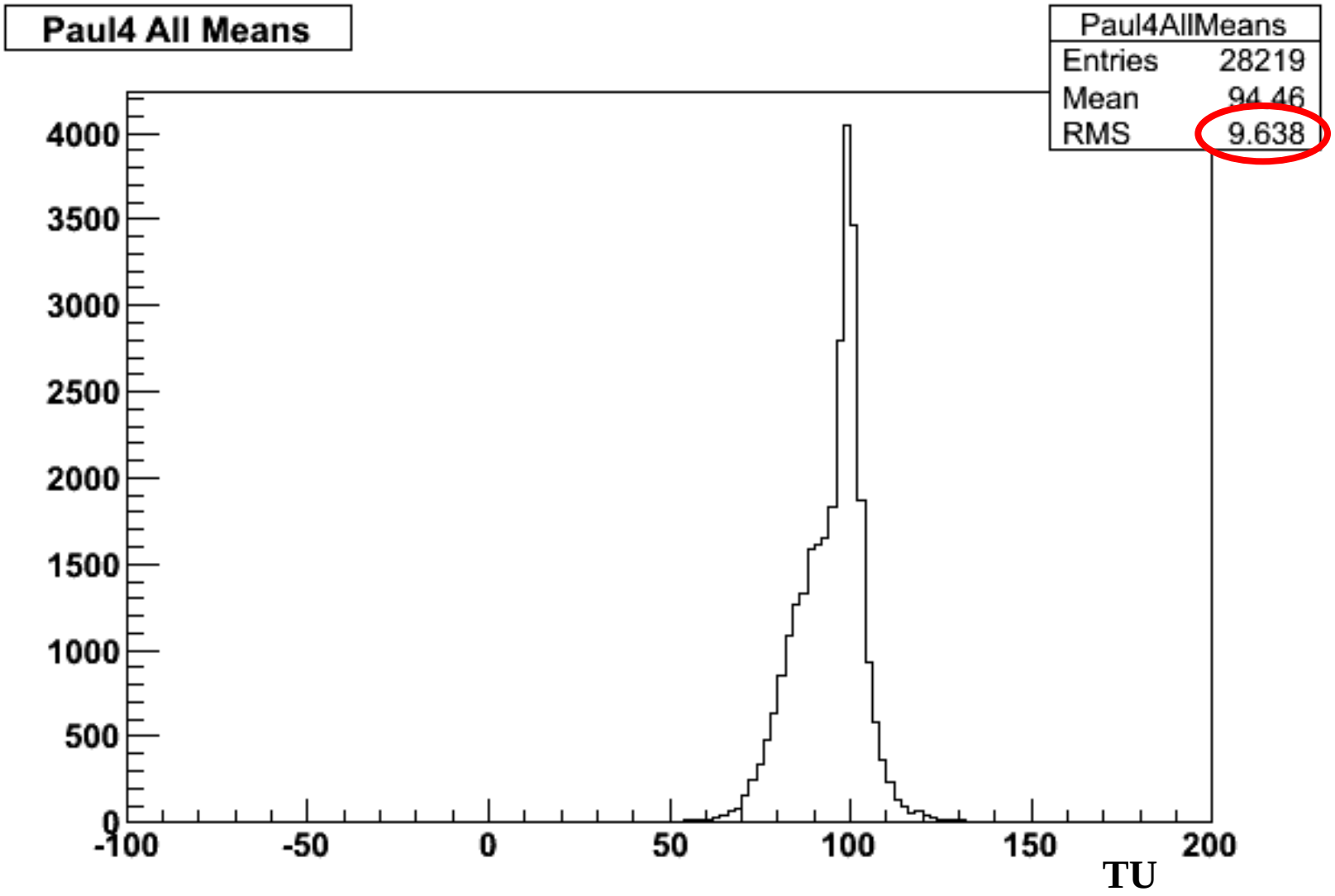
Need to pick target mean value



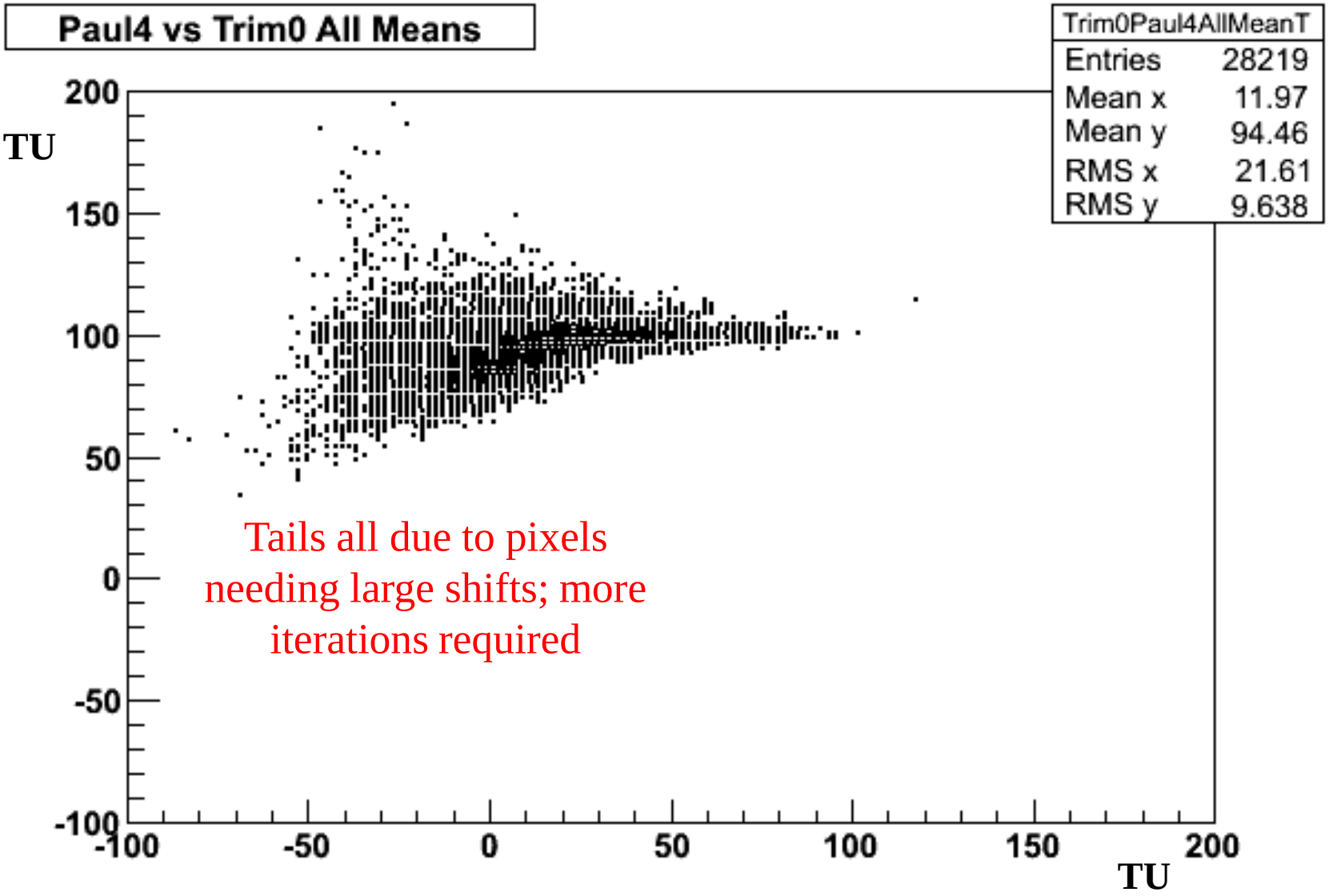
Shift of mean vs trim value



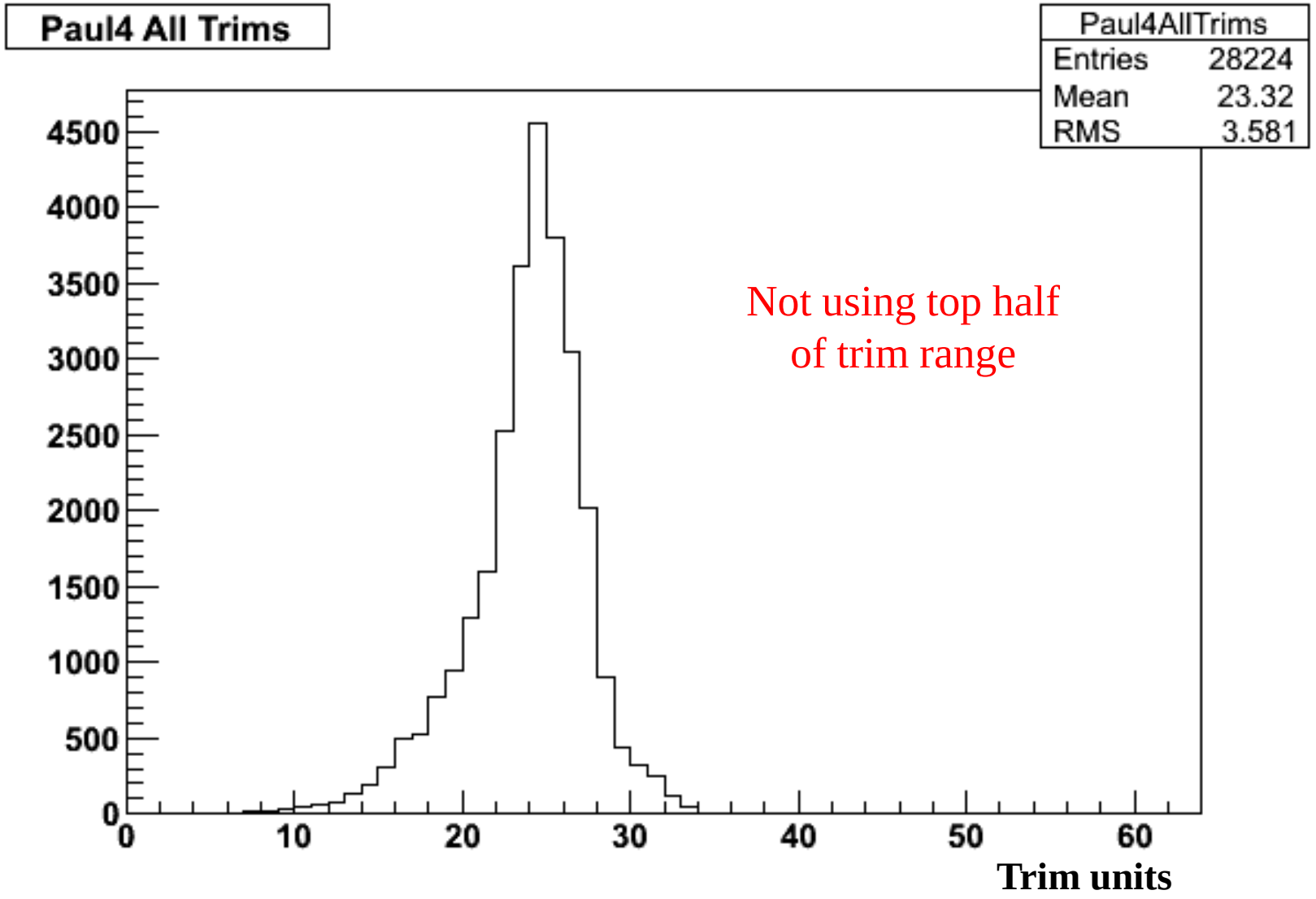
Current status of trim: means



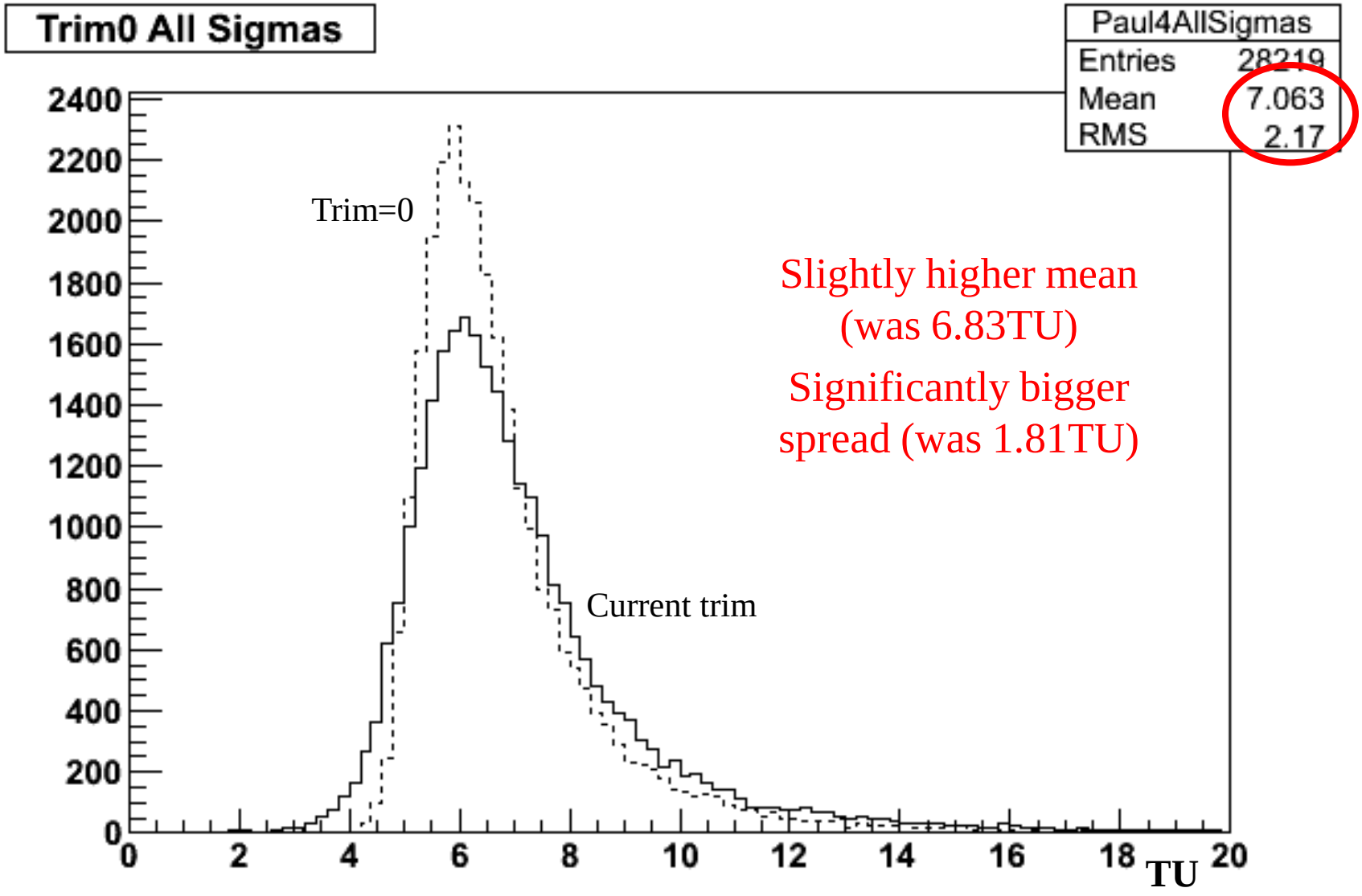
Current status: mean vs trim=0 mean



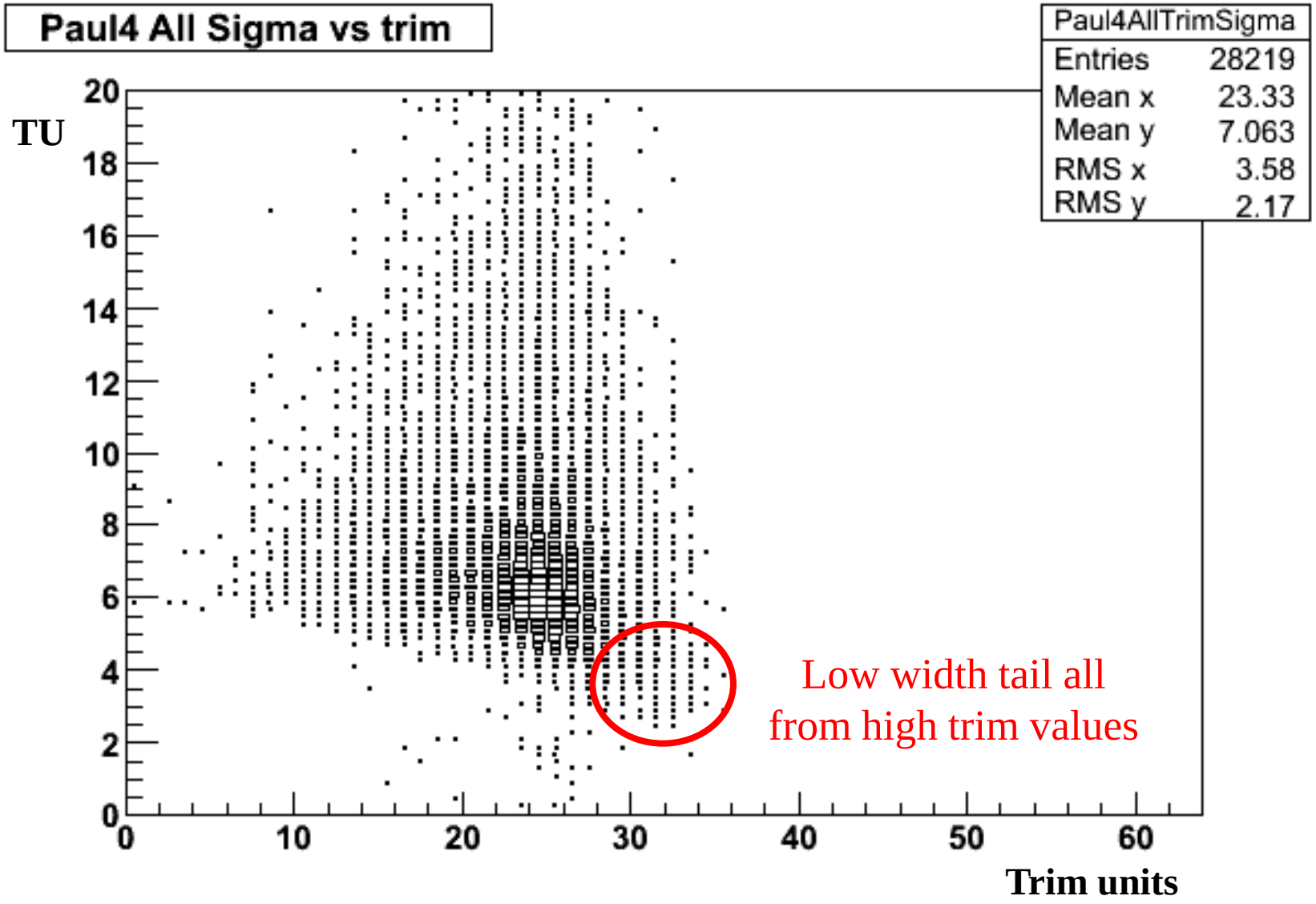
Current status: trim values



Current status: widths



Current status: width vs trim



Conclusions

- TPAC1.2 seems to be working very like TPAC1.0
  - Very similar means and widths
  - In terms of both central values and spread
  - This is very good news
- Trim is working and sensor can be trimmed
  - Non-linear trim response makes mean very sensitive
  - Can this be reduced?
  - Can whole trim range be used?
- Width distribution degraded after trimming
  - Not seen in TPAC1.0 (I think)
  - Due to change of gain?
  - 15% variation comparable with normal spread