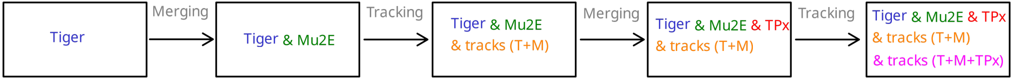
"Combined" format proposal

Straw TB team

October 20, 2023

Combined format history

- First iteration of the combined format created for using in Renat's software (with format simplicity requirement). Initially, it worked with APV and Mu2E data, which was merged without saving scintillator time in output format.
- For Tiger data the full time hit was saved for merging with timepix (mu2e, later) possibility.
- For the merged Tiger & Timepix data current (purely technical) format exists, but it is difficult to use and to share.
- I want to propose new format, which can be used with our current setup and any additional 1/2D coordinate subdetector and/or straw plane(s).
  Additionaly, track parameters along with alignment information can be saved
- The main goal of the new format is to allow to use **iterative procedure** to add new data to existed events without removing old ones, to have possibility to use all from event.



Tiger

Merging

Tiger & Mu2E

Tracking

Tiger & Mu2E
& tracks (T+M)

Merging

Tiger & Mu2E & TPx
& tracks (T+M)

Tracking

Tiger & Mu2E & TPx
& tracks (T+M)
& tracks (T+M+TPx)

Format 1 (TB 2022)

- **root file**

Branch "scintEntry"

- Type: *Long64_t*
- Values: Hit entry in data ROOT file for the scinttillator coinsidence hit

Branch "mm_clusters"

- Type: *std::map<int, std::pair<double, double»*
  - Keys in map: MicroMegas layer (see setup description)
  - Values in map: pair cluster center and uncertainty (only one cluster per layer) Cluster center was calculated as mean for strips in cluster weighted with hit charge. Uncertainty calculated as square root of dispersion weighted with hit charge.

Branch "straw_hits"

- Type: *map<int, double>*
  - Keys: straw number
  - Values: Time difference between straw hit and scinttillator coinsidence hit, in ns

Format 1 (TB 2022)

Branch "mm_clusters_all"

- Values: Additional information for clusters (currently, the same clusters as in mm_clusters branch)
- Type: *structure*:
  - *int* mm_clusters_all.layer: MicroMegas layer
  - *double* mm_clusters_all.center: cluster center
  - *double* mm_clusters_all.centerE: uncertainty of cluster center position
  - /map<double, double> mm_clusters_all.hits:
    - Keys in map: MicroMegas strip
    - Values in map: Charge in strip
  - *int* mm_clusters_all.quality: currently, not used

Format 2 (TB 2023)

- Format 1 with changes:

Branch "straw_hits" **changed**

- Type: *map<int, pair<double, double»*
- Keys: straw number
- Values:
  - Time difference between straw hit and scinttillator coinsidence hit, in ns
  - Straw hit charge

Branch "scintTimeS" **new**

- Type: *Long64_t*
- Values: Scintillator hit time from file start, s

Branch "scintTimeNS" **new**

- Type: *double*
- Values: Scintillator hit time from file start (withing one second), ns

Branch "scintTimeCorrectionNS" **new**

- Type: *double*
- Values: Scintillator hit time correction, ns

Format "Tiger-Timepix"

- **Text file**
- [MM0 cluster center] [MM2 cluster center] [MM0 cluster center] [Number timepix hits] [Number straw hits]
- Next, per line for each timepix hit:
  - [timepix hit positionX] [timepix hit positionY]
- Next, per line for each straw hit:
  - [Straw number] [straw hit time to scintillator] [straw hit charge]

Format 3 (proposal) – "straw_combined" tree

- **root file**

Tree "straw_combined", Branch "time"

- Value: Scintillator time data
- Type: *structure*:
  - *Long64_t* timeS: Scintillator hit time from file start, s
  - *double* timeNS: Scintillator hit time from file start (withing one second), ns
  - *double* timeCorrectionNS: Scintillator hit time correction, ns

Tree "straw_combined", Branch "straw"

- Value: Straw hit data
- Type: *map<string*[1], *map<int*[2], *structure*[3]»:
  1. *string* straw type (possible formats "20mm", "10mm", "5mm", "na64x-0", etc)
  2. *int* straw number (inside straw plane)
  3. Structure for straw hits:
     - *double* time: Time from scintillator hit to straw hit, ns
     - *double* charge: Some type of charge
     - *int* sublayer: Sub-layer number (to separate odd/even layers, for example)

It is important to stay straw numeration constant during all testbeam. For now it is, but it is important to save that.

Format 3 (proposal) – "straw_combined" tree

Tree "straw_combined", Branch "clusters"

- Value: Clusterized coordinates data (micromegas, timepix)
- Type: *map<string*^{1}, *vector<structure*^{2}»:
  1. *string* sub-detector type (possible formats "MM0", "TPx0", etc)
  2. *Vector of structure* for cluster information (sorted in descending order by charge/size):
     - *double* x: X axis coordinate, **in strips/pixels**
     - *double* xE: X axis coordinate uncertainty, **in strips/pixels**
     - *vector<structure*^{3}> hits: information for hits in cluster
     - *int* quality: quality value, may be useful later
  3. *structure* for hit information:
     - *int* x: X axis hit position **(maybe double?)**
     - *double* charge: hit charge
     - **(maybe)** *int* Y: Y axis hit position **(maybe double?)**

We decided to split all hits to one-dimentional subdetectors, even in case of two-dementional detectors (e.g. timepix)
The main problem: we should use second coord in case of 2-dimentional detectors

Format 3 (proposal) – "straw_combined" tree

Tree "straw_combined", Branch "track"

- Value: constructed track information
- Type: *map<straw*[1]*, structure*[2]*>*:
  1. *straw* track version (possible formats: "v0-x", "v0-y", "v1-y", etc)
  2. *structure* for track information (one-dimentional track):
     - *double* p0: p0 coefficient in "p0 + x * p1" equation
     - *double* p1: p1 coefficient in "p0 + x * p1" equation
     - *int* alignment: used alignment (index in "straw_alignment" tree)
     - set<string> used: set of sub-detector names used for constructing track (see "clusters/sub-detector type")
     - *int* quality: quality value, may be useful later
     - *double* chi2: χ^2 for track (do we need?)
     - *double* ndf: NDF for track (do we need?)

Format 3 (proposal) – "straw_alignment" tree

Tree "straw_alignment", Branch "alignment_clusters"

- Value: Alignment information for micromegas/timepix, with strips/pixels in XY plane. We suppose, Z coordinate is constant
- Type: *map<string[1], structure[2]>*:
  1. *string* sub-detector type (see "clusters/sub-detector type")
  2. *structure* for alignment information
     - *int* axis: axis for detector **(maybe char?)**
     - *double* x: coordinates of strip/pixel (one axis)
     - *double* rotXY: rotation angle (in XY plane)
     - *double* rotXZ: rotation angle (in XY plane)
     - *double* rotYZ: rotation angle (in XY plane)
     - *double* stepX: strip/pixel/straw step by X axis, mm

Tree "straw_alignment", Branch "alignment_straw" – **do we need it?**

- Value: Alignment information for straw planes
- Type: *map<string[1], structure[2]>*:
  1. *string* straw type (see "straw/straw type")
  2. *structure* for alignment information
     - *int* axis_wire: wire along X(0), Y(1) axis **(maybe char?)**
     - *double* rot: rotation angle to Z axis
     - *double* step_size: step size along plane, mm

Format 3 (proposal) – "z_coord" tree

Tree "z_coord", Branch "z_coord"

- Value: Alignment information for micromegas/timepix, with strips/pixels in XY plane. We suppose, Z coordinate is constant
- Type: *map<string*[1]*, double*[2]*>*:
  1. *string* sub-detector type or straw type (see "clusters/sub-detector type" and "straw/straw type")
  2. *double* z coordinates (Z axis)