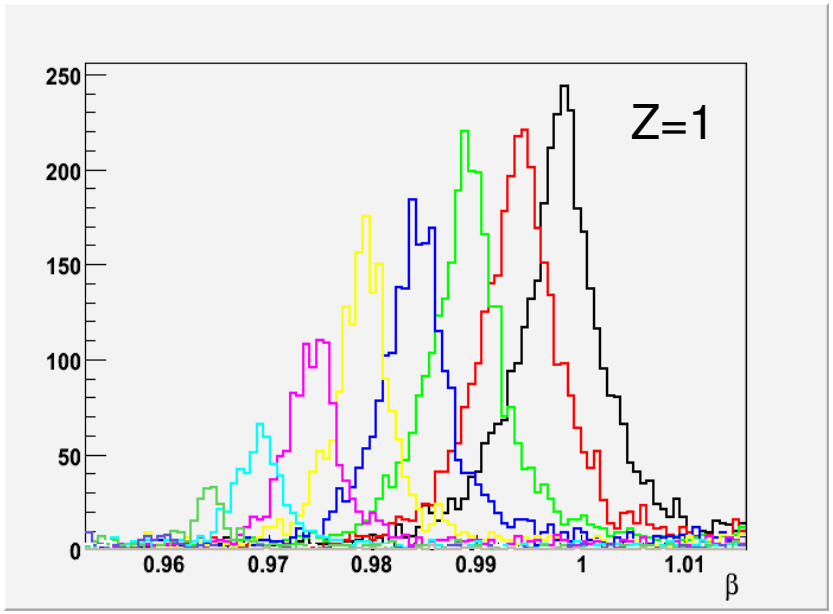
RICH software update

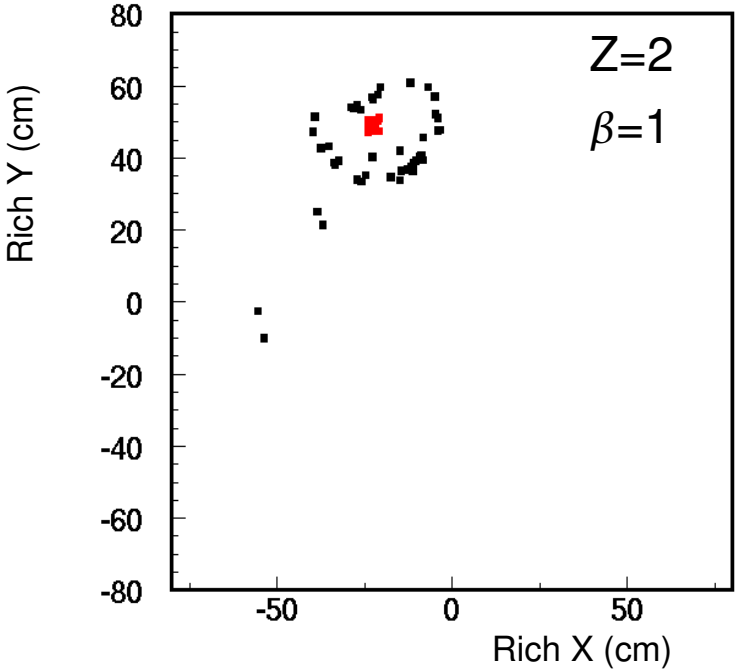
Outline

- Standalone software updates
- Full AMS software updates
  - Geometry
  - Radiator tiles management
- On going/pending work
  - PMT plane
  - Reconstruction
  - Raw event input/output
  - Calibration

Standalone software: standalone reconstruction

- Stand alone reconstruction commited (including hot-spots identification)
- My new hot spots identification is only enabled if the my standalone is enabled.





- **To use it add 1000 to the datacard RSWI (it takes very long...)**

Standalone software: standalone reconstruction

Ntuple

B4BETA:	Reconstructed beta
B4NHITS	Number of hits flagged as good ones during reconstruction
B4USED	Number of hits which belong to the ring
B4GOF	Goodness of fit
B4PMTS	Number of PMTs flagged as good ones
B4THETA	Reconstructed theta angle
B4PHI	Reconstructed phi angle
B4R(3)	Reconstructed impact position
B4B1BETA	Not yet implemented (standard reconstruction using beta4 track)
B4B1NUSED	Not yet implemented
B4B1PROBKL	Not yet implemented

- **Moreover, signal for hits within crossed PMTs have negative signal (let me know if this disturbs you)**

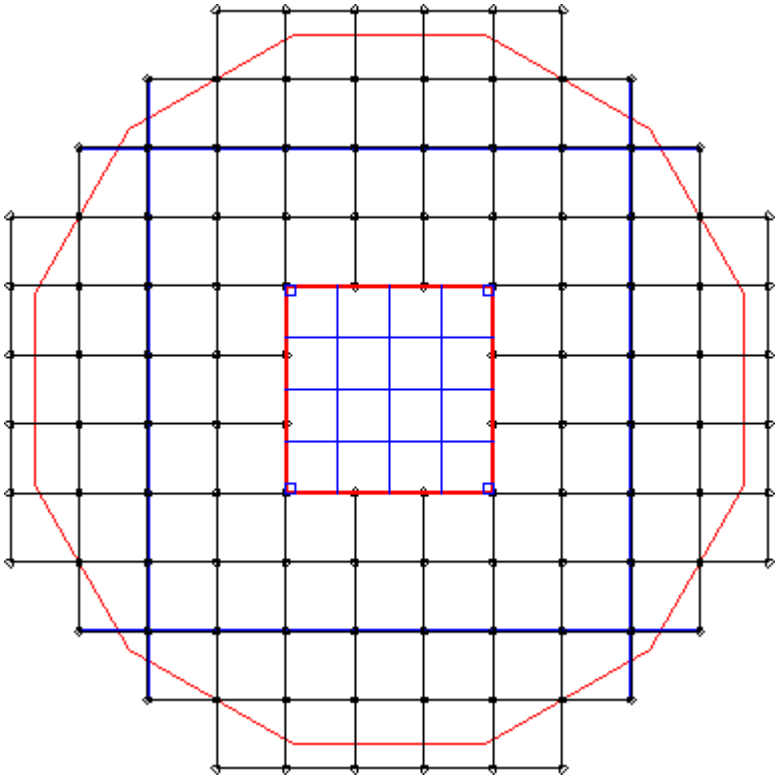
Standalone software: radiator geometry

- Partial update of radiator geometry (full update for full AMS software):
  - Added radiator container geometry.
  - Aerogel tiles in the border have the correct shape.
  - NAF radiator not yet updated.

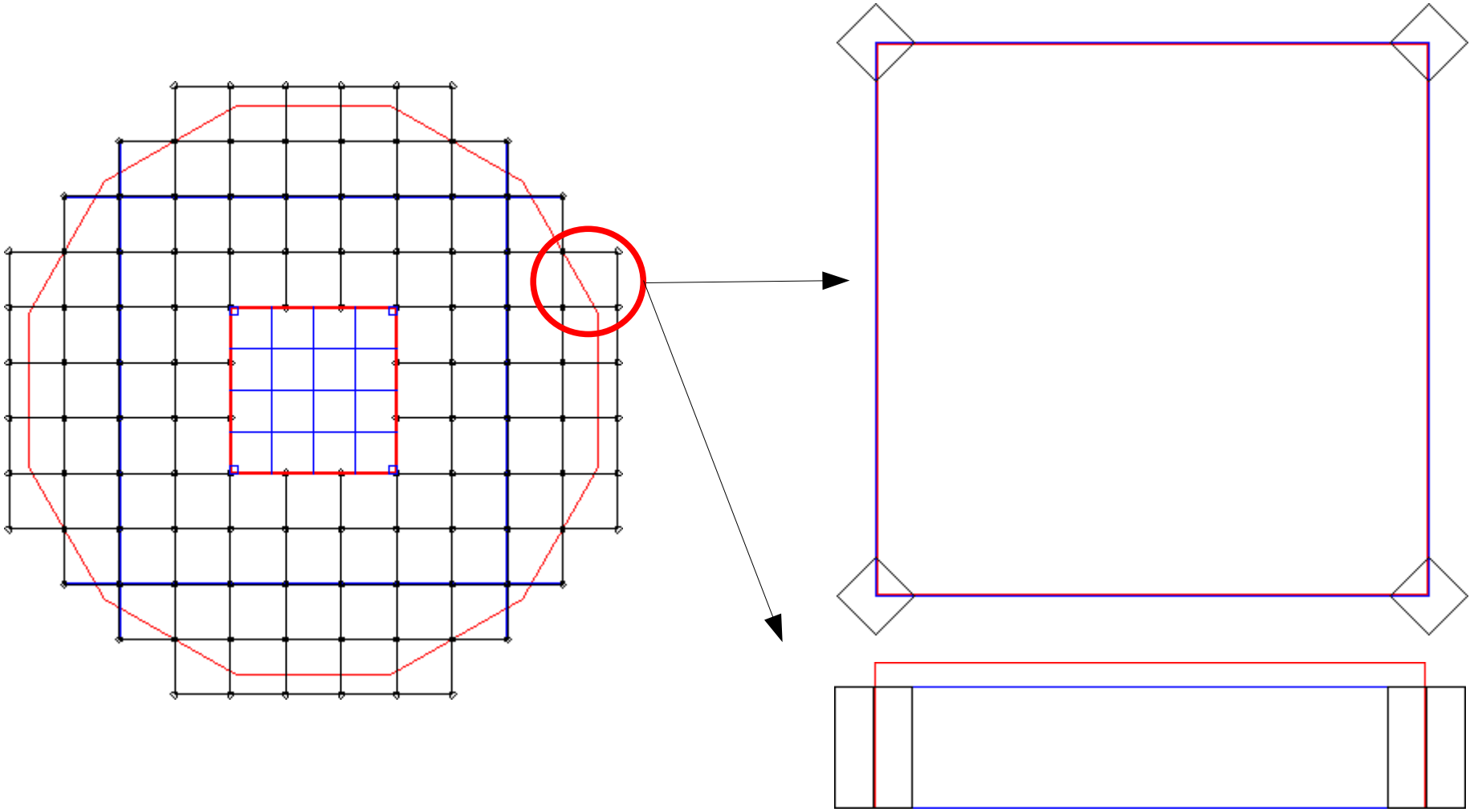
Full AMS software: radiator geometry

- Added radiator container geometry.
- Aerogel tiles in the border have the correct shape.
- Added corner holes for screw (filled with PORON-like material)
- Updated NAF radiator tiles at the border.
- Added holes in the NAF radiator.
- Included correct NAF radiator tiling (still 0.2 mm disagreement between drawings and current simulation)
- Simulation still not working as it should
  - I have to decide how I implement completely different optical properties for each tile (implementing different tracking media, custom tracking algorithm...)

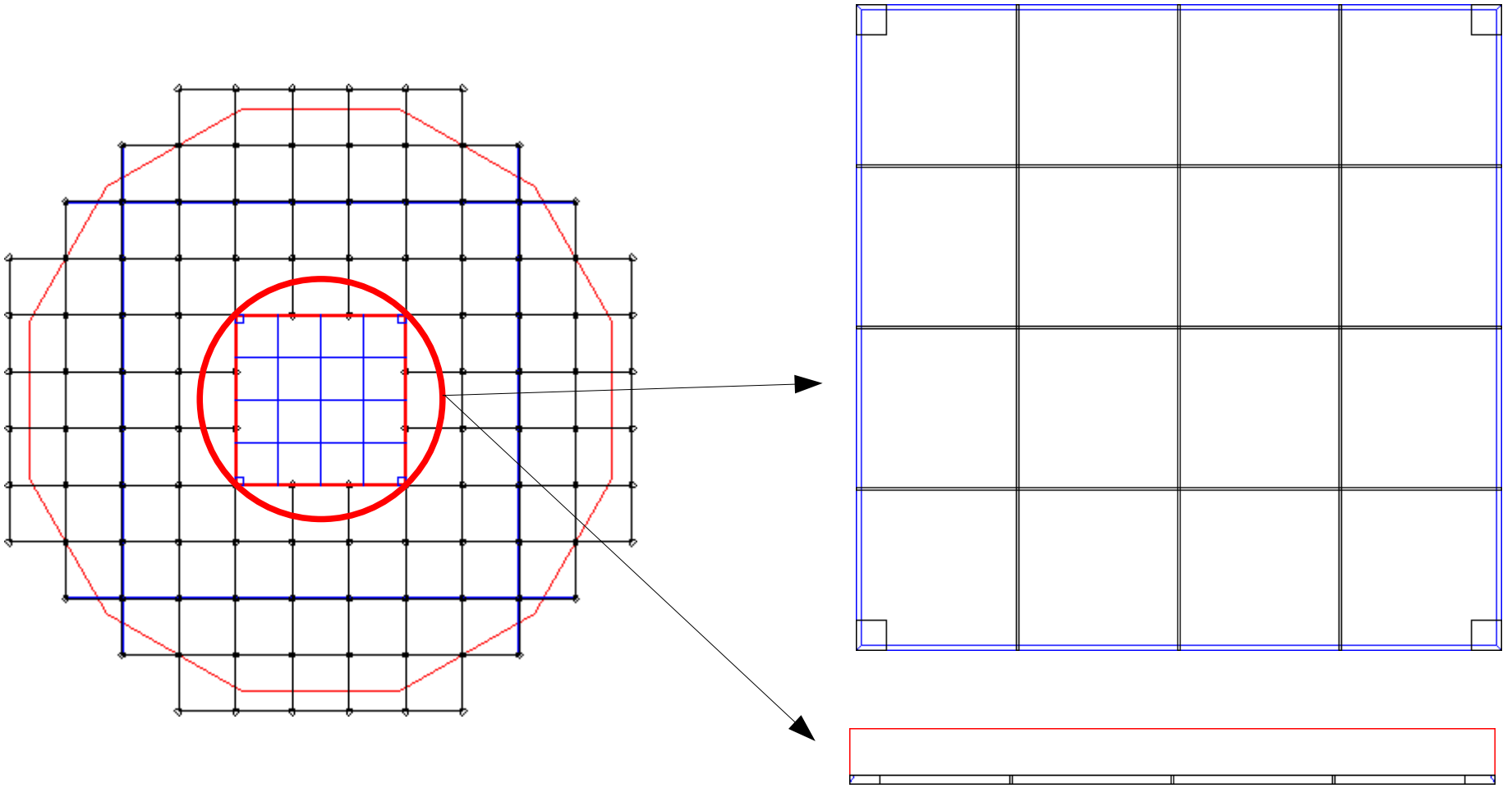
Full AMS software: radiator geometry



Full AMS software: radiator geometry



Full AMS software: radiator geometry



Full AMS software: radiator tiles management

- Big changes in how radiator information is accessed/stored within software.
- New code is in richradid.[C,h]
- New classes:
  - RichRadiatorTile: store information for a single tile
  - RichRadiatorTileManager:
    - Access to information in RichRadiatorTile.
    - Modify it as needed.

Full AMS software: radiator tiles management

```
Class RichRadiatorTile{
 private:
  double position[2];                    // Position wrt container center
  double bounding_box[3][2];             // Bounding box (3 coordinates, 2 boundary values)
  int kind;                              // Flag with the kind of tile
  int id;                                // A unique identificative number
  double mean_refractive_index;          // Mean refractive index of the tile for reconstruction
  double mean_height;                    // Mean emission height for reconstruction
  double clarity;                        // Clarity
  double effective_scattering_probability;   // Scattering model parameters
  double effective_scattering_angle;
  geant *abs_length;                     // Array with absorption lengths
  geant *index_table;                    // Array with chromatic dispersion
 public:
  RichRadiatorTile(){}; ~RichRadiatorTile(){}

  geant getheight(){return fabs(bounding_box[2][0]-bounding_box[2][1]);}

  // Here we can add geometric models and maps for everything
  // Here we should add histograms and whatever is needed for calibration purposes


  friend class RichRadiatorTileManager;  // Only the tile managercan modify those guys
};
```

Full AMS software: radiator tiles management

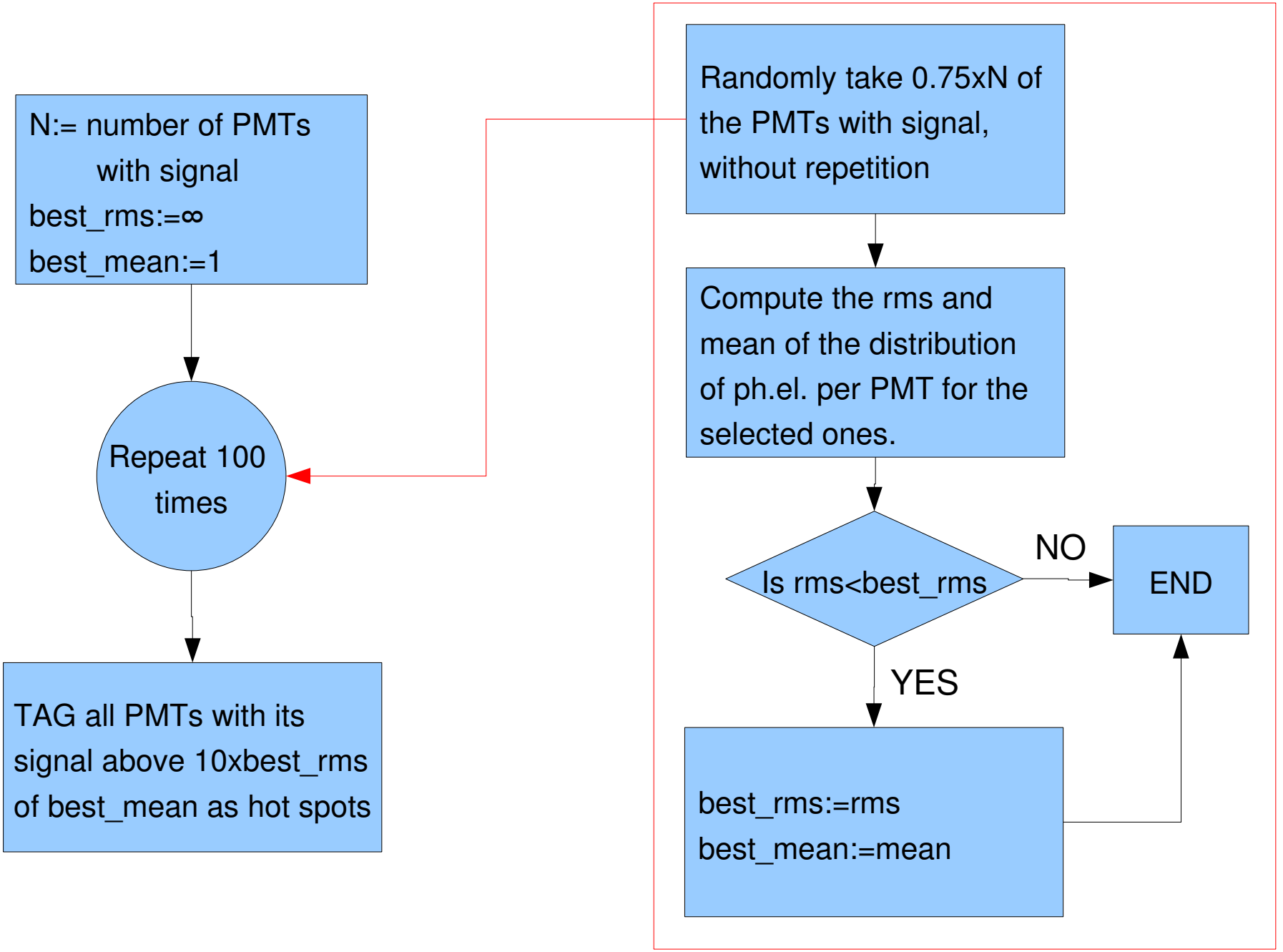
```
class RichRadiatorTileManager{
 private:
  int _current_tile;
  AMSPoint _p_direct;
  AMSPoint _p_reflected;
  AMSDir _d_direct;
  AMSDir _d_reflected;
  AMSPoint _p_entrance;
  AMSDir   _d_entrance;

  // All the necessary numbers
  static integer _number_of_rad_tiles;       // Number of radiator tiles
  static RichRadiatorTile **_tiles;          // The tiles themselves

 public:
  static void Init();                 // Init geometry and kinds and so on
  static void Init_Default();                 // Init geometry and kinds and so on
  static void Finish();
  static void Finish_Default();

  RichRadiatorTileManager(AMSTrTrack *track); // Constructor given a track
  ~RichRadiatorTileManager(){};
}
```

Full AMS software: hot spots identification *a la* standalone rec.



Full AMS software on going work: PMT plane

- Big changes in how PMTs information is accessed/stored within software.
- New code not yet released.
- Include raw data interfaces.
- Include slots for calibration data and methods.
- Calibration output will be stored when job finishes.
- New classes implementation reflects raw data structure:
  - A class to store all information for a single channel in a single gain mode.
  - A class to manage them.
  - Accessing by channel number, PMT and anode number, CDP card number and position within it...

Full AMS software on going work: reconstruction

- (Nearly)Stand-alone reconstruction not yet implemented.
- Track direction reconstruction is being implemented now, with the constrains of using tracker impact point, as a fast cross-check method.
- Full stand-alone reconstruction will be enabled with a datacard.