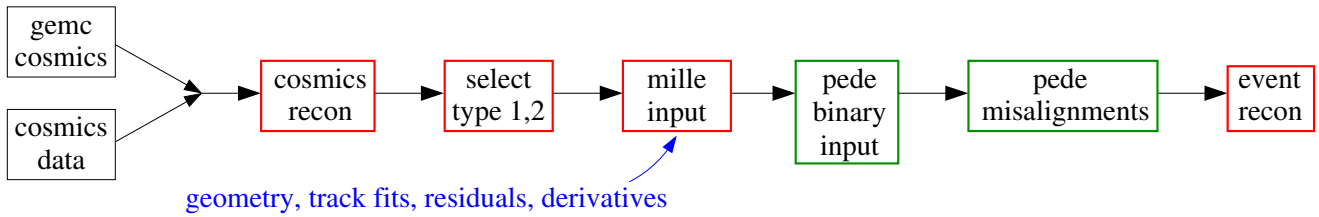
SVT Track-Based Alignment With Millepede

1. Analysis chain:

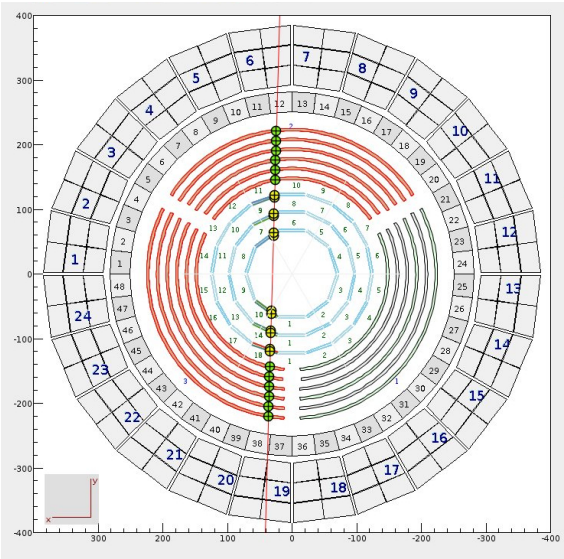
   Red - C++

   Green - java

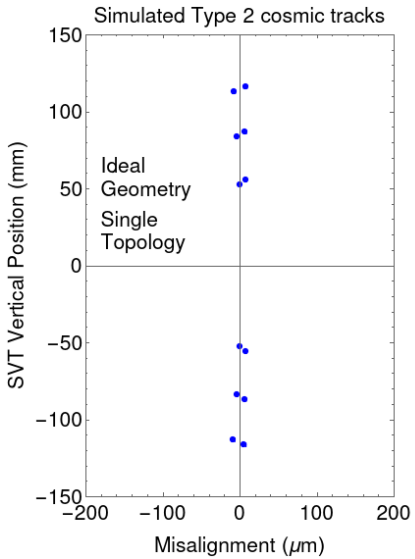


2. Chain worked with type-1 events within $\lesssim 10\ \mu m$
3. Applying Tracker reconstruction and the full chain to type-2 events.

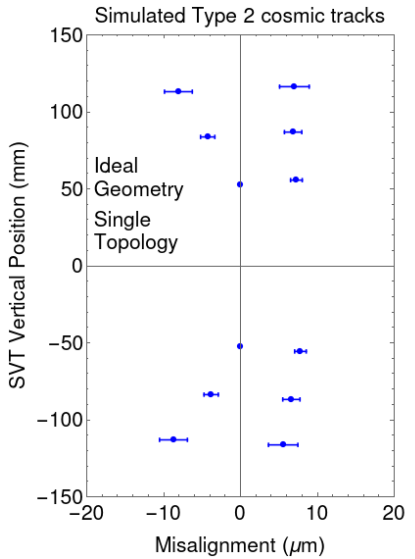
1. Single topology, type-2 event.



2. Ideal geometry



3. Zoom in.

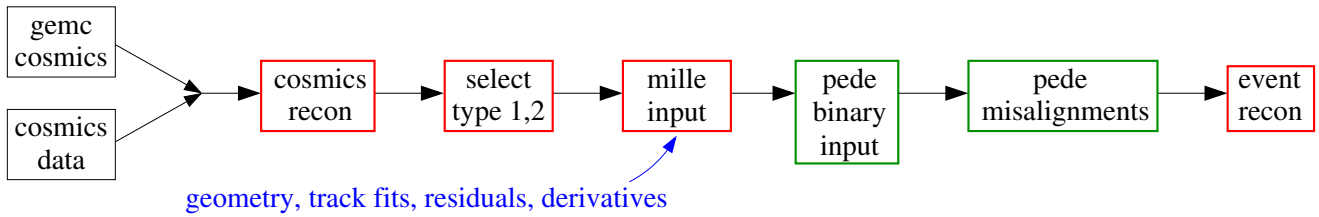


SVT Track-Based Alignment With Millepede

1. Analysis chain:

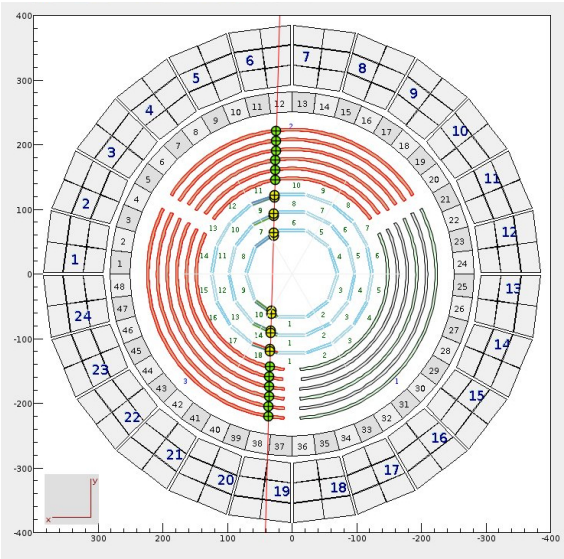
   Red - C++

   Green - java

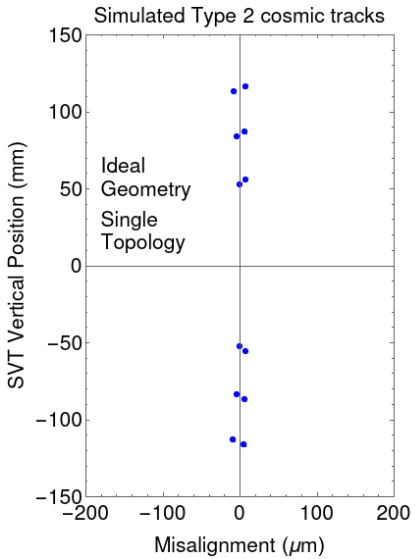


2. Chain worked with type-1 events within $\lesssim 10\ \mu m$
3. Applying Tracker reconstruction and the full chain to type-2 events.

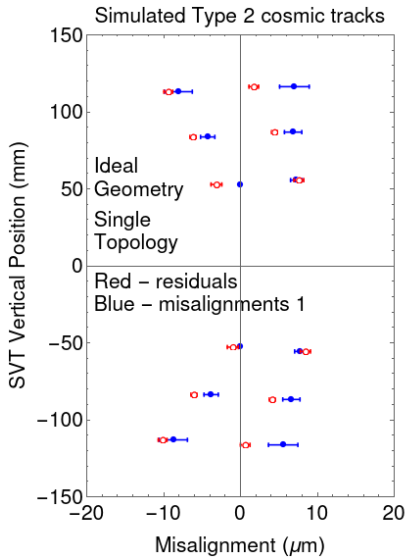
1. Single topology, type-2 event.



2. Ideal geometry

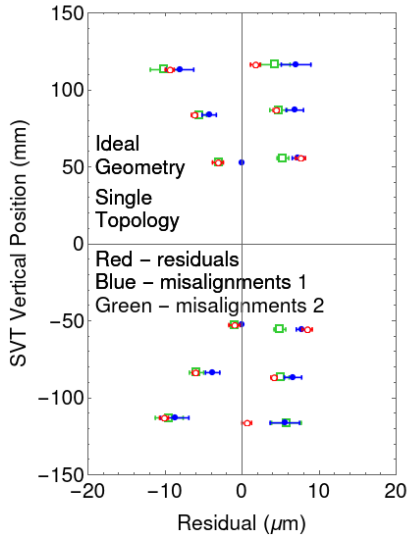


3. Zoom in.

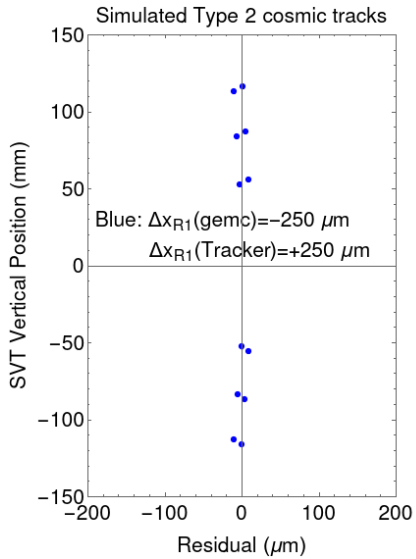


SVT Track-Based Alignment With Millepede

4. Tune the constraints on millepede to match the L1 residuals. See modest improvement in match between new misalignments(green) and residuals (red).

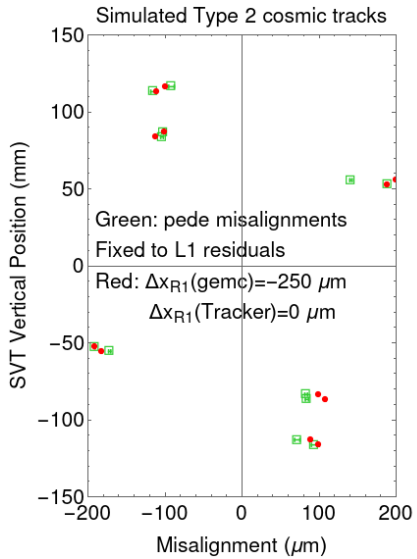


5. Test larger shifts: use $\Delta x_{R1} = -250\mu m$ in *gemc* and $\Delta x_{R1} = +250\mu m$ in Tracker. Get good agreement.



SVT Track-Based Alignment With Millepede

6. Now apply millepede to shifted *gemc* and unshifted Tracker to get misalignments. Set constraints to L1 residuals. Get good agreement.



7. Now set the millepede constraints to the known value (zero) and get the misalignments. Should see L3-L6 with small misalignments and L1-L2 with large ones near $\approx \pm 250\ \mu m$. Some fits fail. Other get poor agreement.