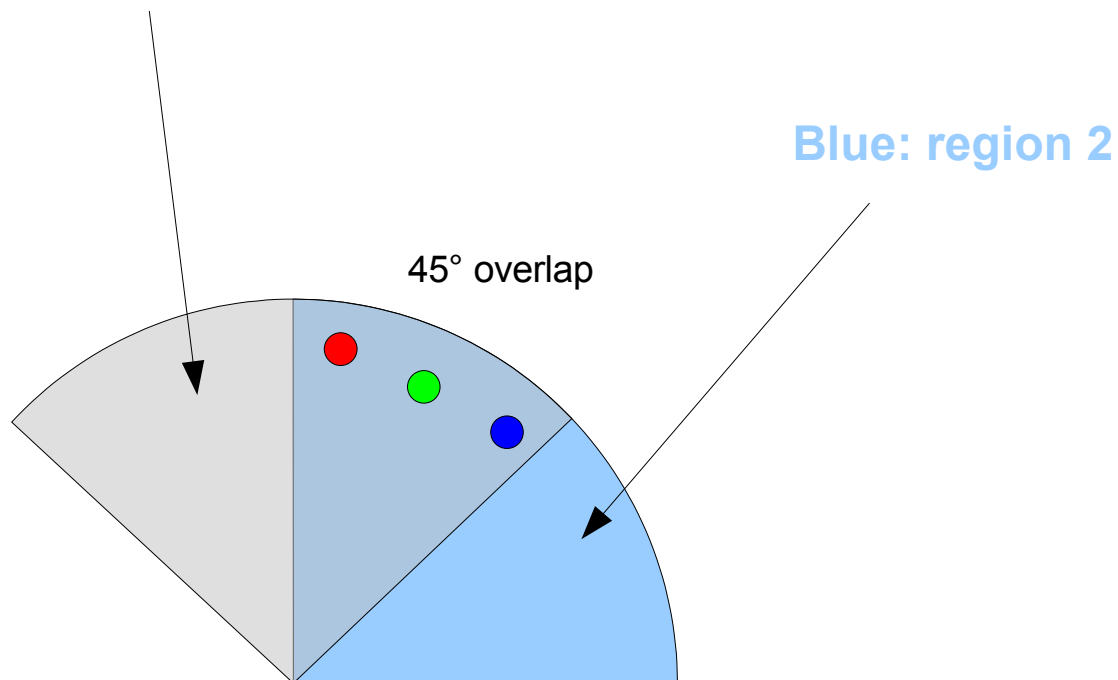


For simplicity, consider only one layer. Shown are region bounds for region 1 and region 2, with the presently used 45-degree overlap. Tracks that leave the three indicated hits will produce sectors (and patterns) in region 2 only (see below for a detailed explanation).

Gray: region 1



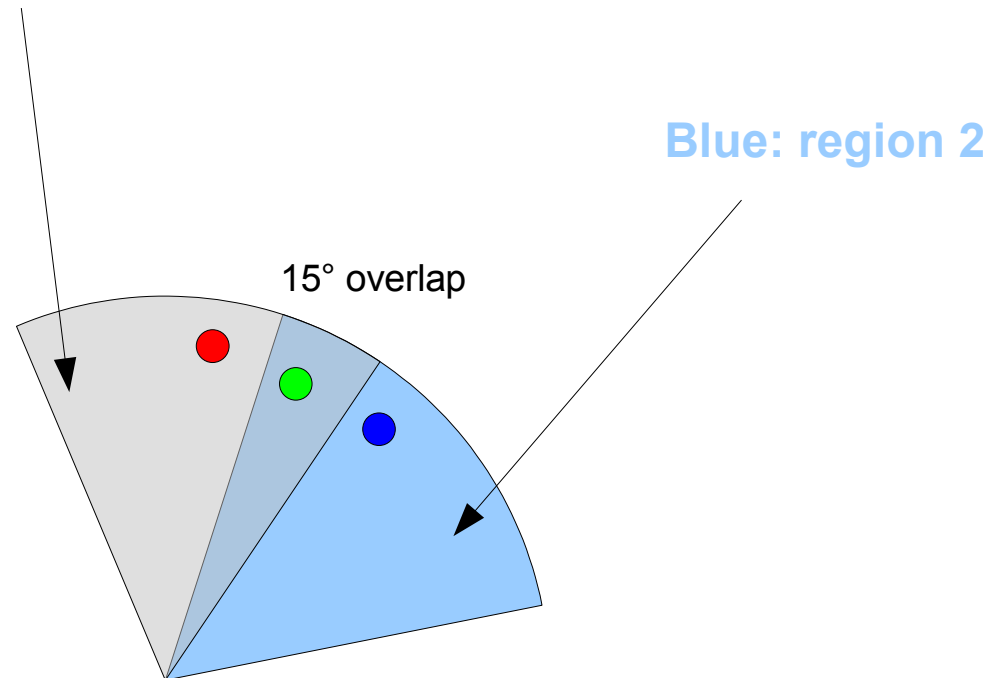
Suppose training tracks leave the three hits shown in the picture. All three will produce a pattern assigned to region 2 (because `whichRegionAlt()` puts sectors and patterns into the the most clockwise of the matching regions)

Now suppose we load the above patterns into AM, and now three physics tracks leave the above hits. In TrigFTKSim (and in DataFormatter in hardware), each hit will get duplicated to both regions 1 and 2. In region 2, they will match to the three corresponding patterns.

But in region 1, they are useless – that's why we would like to have smaller overlaps

Now suppose we reduce region sizes from 90 degrees to something smaller, so that the overlap is around 15 percent:

Gray: region 1



What happens if we try to use old patterns with the new region map?

All three patterns are still in region 2. However, due to smaller overlap, the same hits are duplicated differently:

- **Blue** hit only goes to region 2, and will match to a pattern there.
- **Green** hit goes to both region 1 and 2. In region 2, it will match to a pattern. In region 1, it is useless
- **Red** hit will only go to region 1. But the matching pattern is in region 2! it won't match, and we lose efficiency.

This means that we have to produce patterns using a new region map. In this case, patterns compatible with the red hit will (correctly) be stored in region 1.

In principle, it is possible to take the old patterns and carefully go through those that are in region 2, but should be in region 1 under a new, smaller-overlap region map. Then we can move such patterns to region 1. However, that's an error-prone procedure, and it certainly more complicated than simply producing new patterns using the new region map.