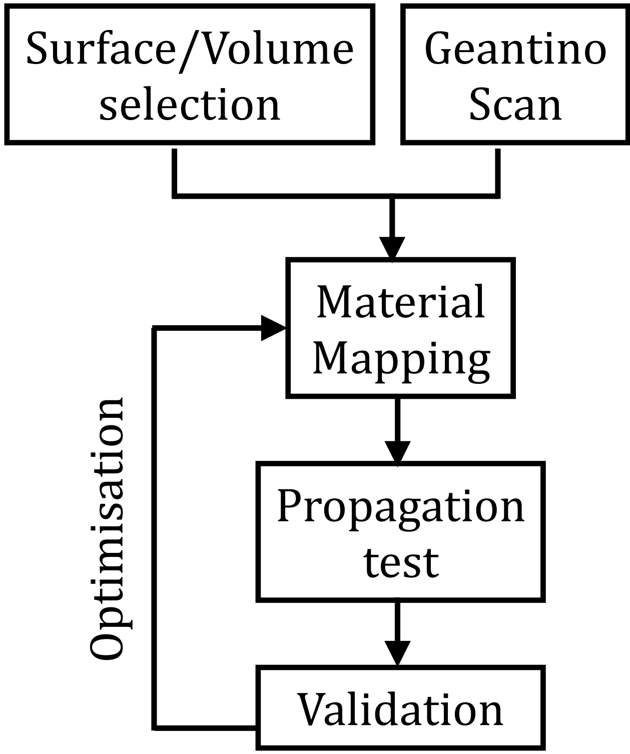
Material Mapping optimisation with Orion

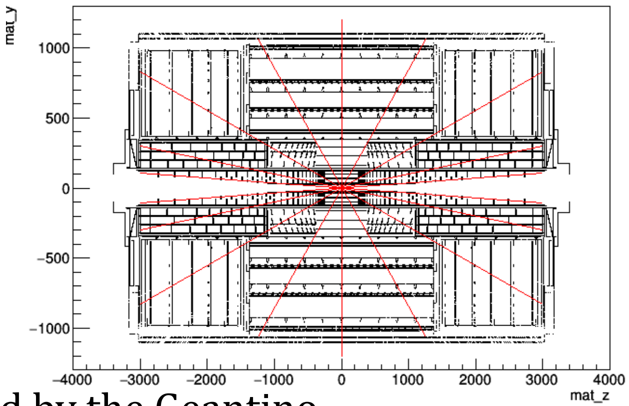
Corentin Allaire

Material Mapping

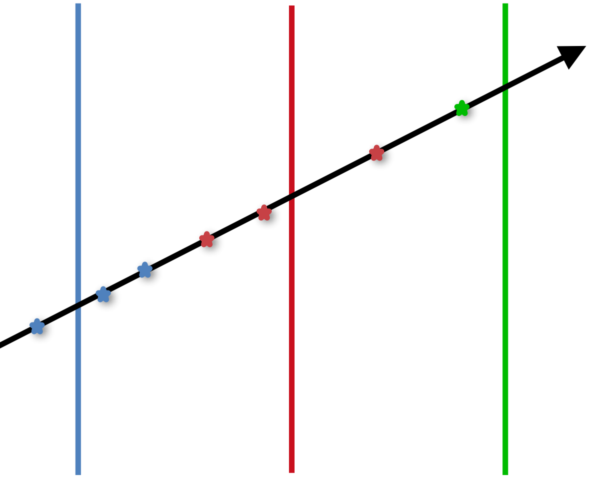
- Interaction with material need to be accounted in tracking
- The usual implementation of the geometry and the material of detector (G4) is extremely detail and is not convenient to be used
- The material thus need to be average and store in a more convenient format before being used in the tracking
- Two type of mapping :
  - **Surface mapping**, used to describe tracking volumes and gap material. The material is average onto surfaces, the material interaction is then taken into account punctually when the surface is crossed.
  - Volume mapping, used to describe dense material volume (for example calorimeters). The material interaction is taken into account at each propagation step.



Surface Material Mapping



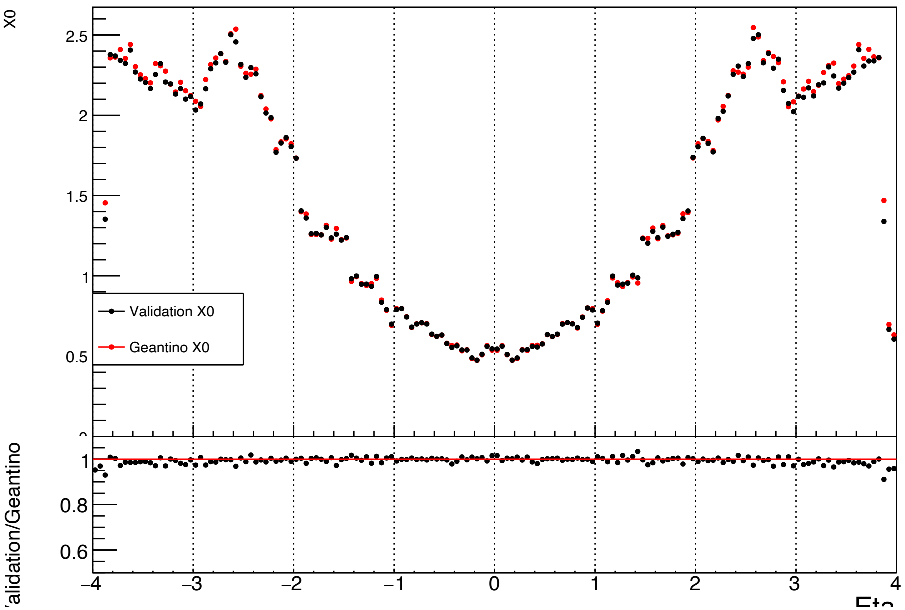
- Material collected from the detector with a Geantino scan
- Store Material Track :
  - Information on the simulated Geantino
  - Position and material at all the interaction points
- Mapping : for each track we iterate through all the surfaces crossed by the Geantino
- The material of each interaction is mapped onto the closest surface

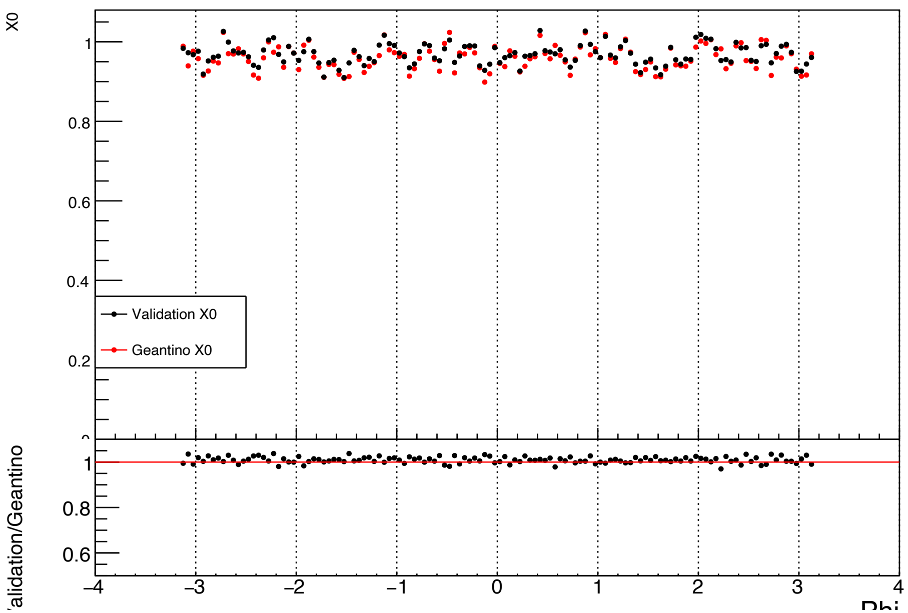


- Each surface is binned in 1 or 2D with a binning chosen by the user
- The material collected is average over all the tracks for each bin
- The resulting material map is then stored as a root or json file
- In addition a new Material Track collection is created in which the associated surface is added to the interaction information for validation purpose

Material Validation

- Once the map has been created we generate some validation plots
- Use the Acts extrapolation to collect the material and do comparison with the geantino scan
- Lots of other plots available : distance between the surface and material, material ratio per surface, …
- Great for manual optimisation but take a lot of time to do manually -> Could we automatise ?





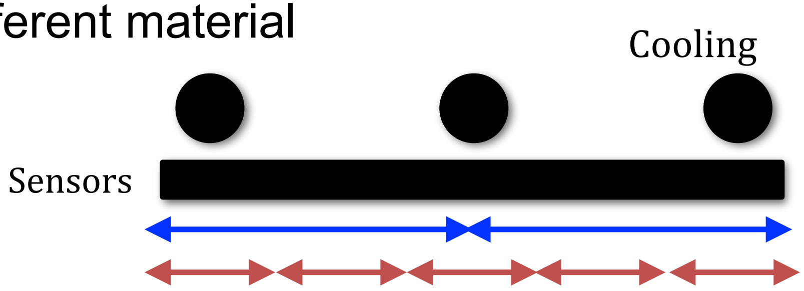
Automatic Optimisation ?

- What is the main goal of the material mapping ?

  ➡Create an easy to read binned map of the material
  ➡Must be a good representation of the original material
  ➡Should use too many bins or it will be hard to store in memory

- A good proxy for those criteria is the material variance in each bin :

  ➡With bin too large : we average a lot of different material -> large variance -> poor performance
  ➡ With smaller bins : each bin correspond to one material combination -> small variance good performance



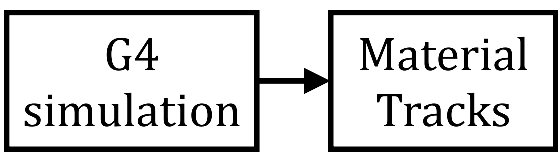
- The number of track use in each bin can the also be used to limit the creation of bin that are too small

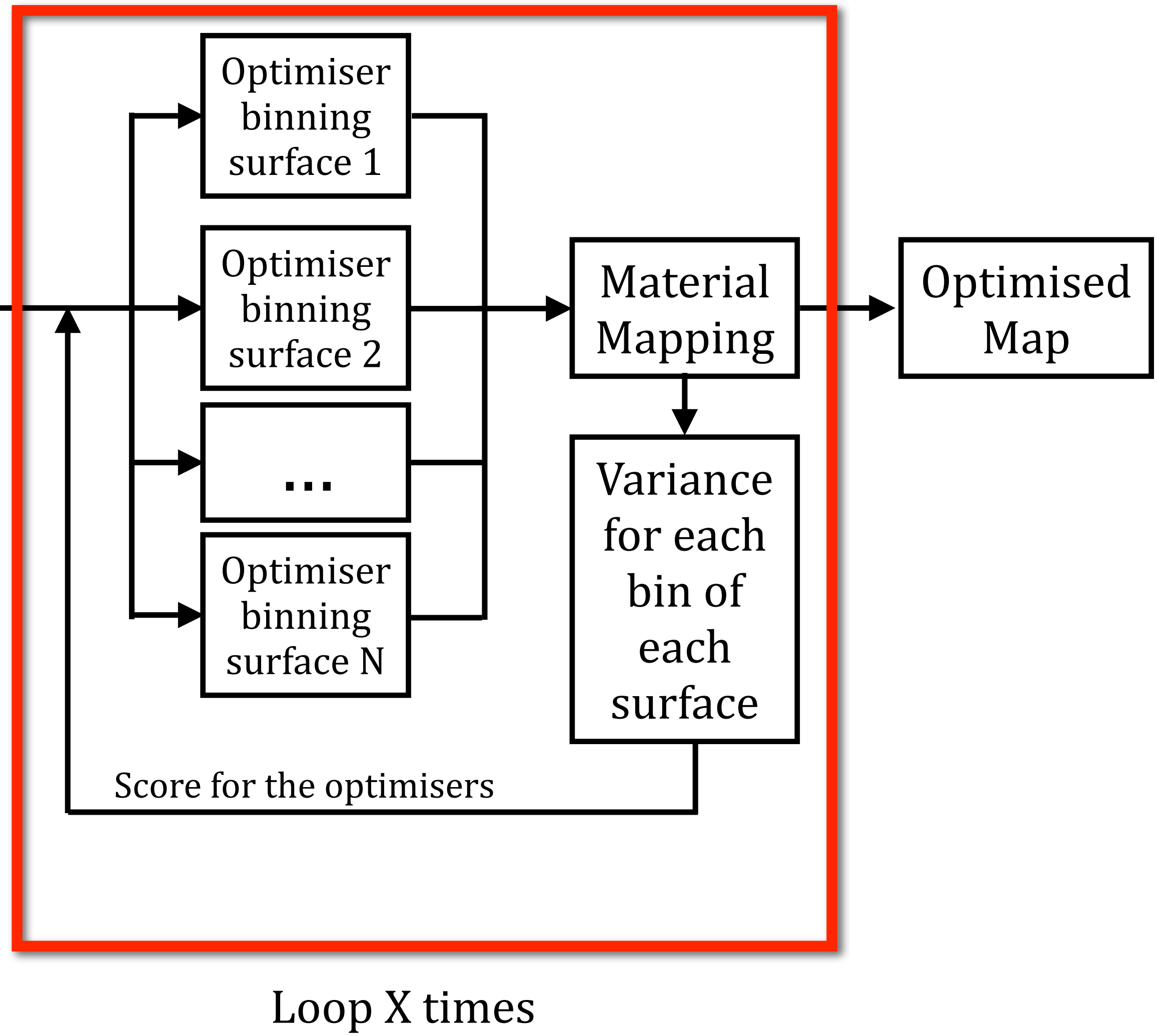
Material Variance

- To compute the variance we need to already know the average material in each bin
- We need to run the material mapping twice to compute it :
  - On the first path we create the material map that contain the average material for each bin
  - On the second we use the map as an input to be able to compute the variance for each bin
- I updated the material mapping to support the computation of the variance, a PR should follow soon
- The method can be a bit slow as it requires performing the mapping twice (for the ODD that is 2 x 12 ms per track)

Optimisation of the binning

$$Score_{Surface} = \sum_{bin} \left(\frac{variance_{mat}}{\sqrt{tracks}}\right)$$





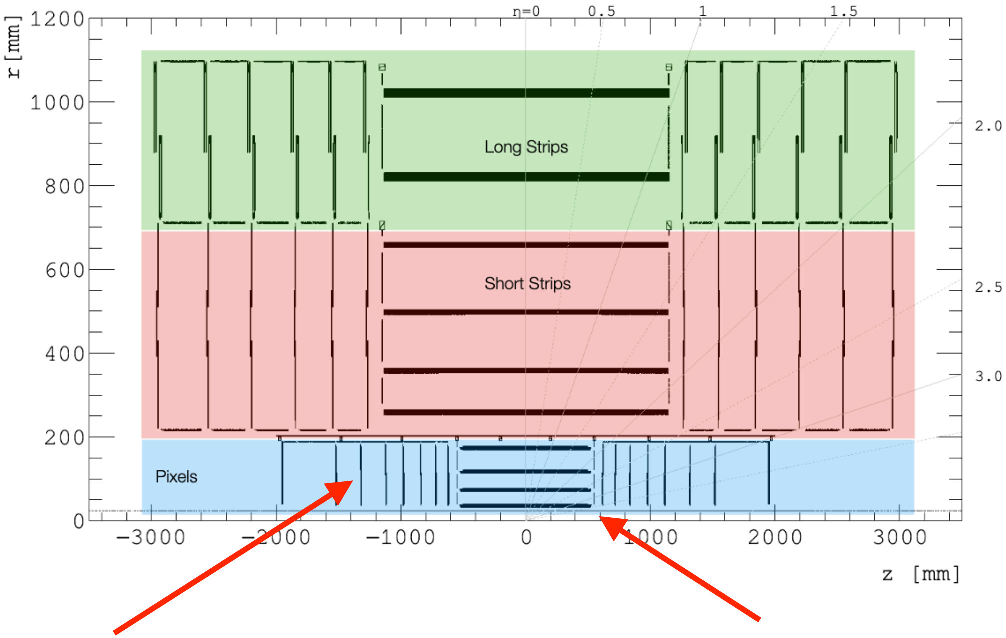
- Use orion to optimise the binning of each surface
- Binning are independent (for fixed surfaces)
- Each optimiser act independently -> one last mapping with all the best score need to be performed at the end

Orion

- This optimisation as been integrated in the Acts python binding :
https://github.com/Corentin-Allaire/acts/blob/Mapping_variance/Examples/Scripts/Python/material_mapping_optimisation.py
- The material mapping jobs can be quite long if you want to run on a lot of tracks (~4h)
- Support parallel execution to speed up the optimisation but a dedicated machine running over the night is recommended
- This is made quite easy with the way orion use and access database (can stop the optimisation, restart, add more trial, …)
- Provide a lot of interesting validation plots at the end

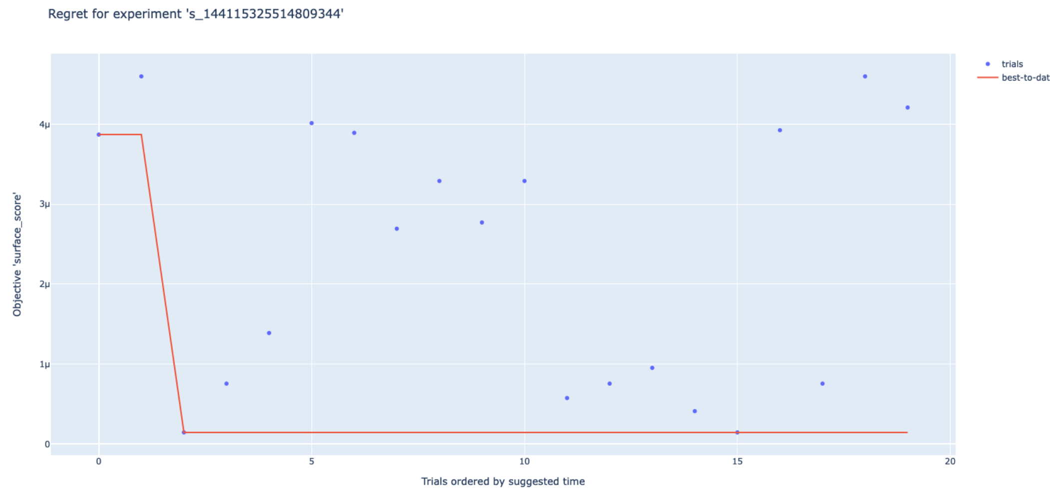
ODD optimisation

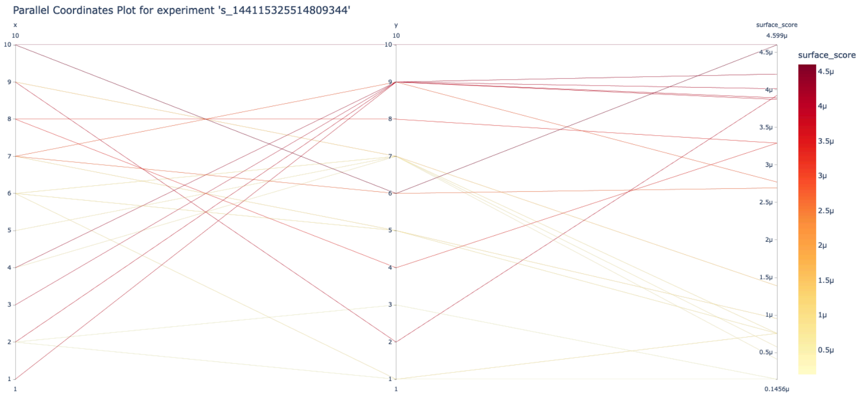
- No full material mapping optimisation yet
- First trial with the ODD :
  - 100000 tracks
  - Bin range between 1 and 10
  - 20 iterations of a random search (4 parallel jobs of 5 trials each
- Binning : [Phi-R] for disk (end-caps) or [Phi-Z] for cylinder (barrel)
- Results with two surfaces next slides
- Haven't look at the full map yet

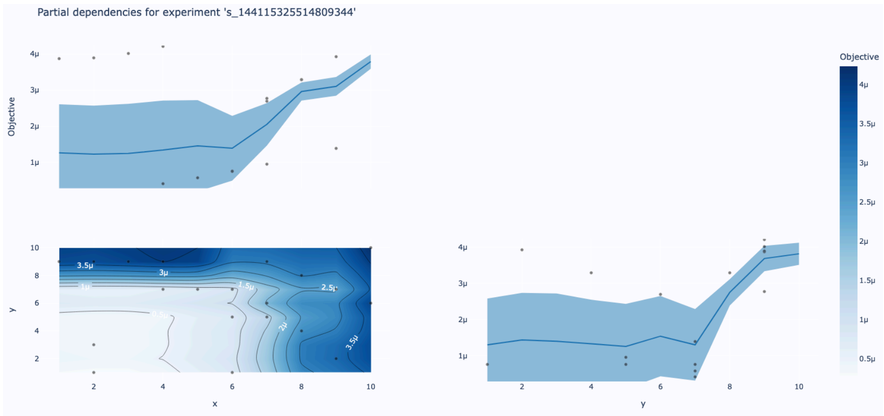


Binning BeamPipe

- Beam pipe (first cylinder) homogeneous material
- The binning converge to (1,1)
- Maybe a bit more bin could be helpful but due to the low stat we will tend to converge to small score

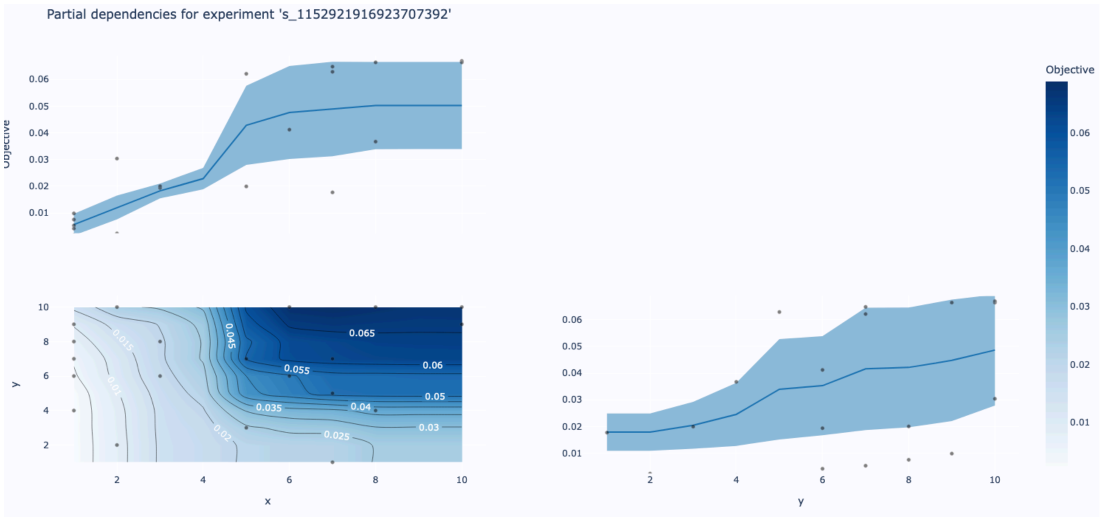


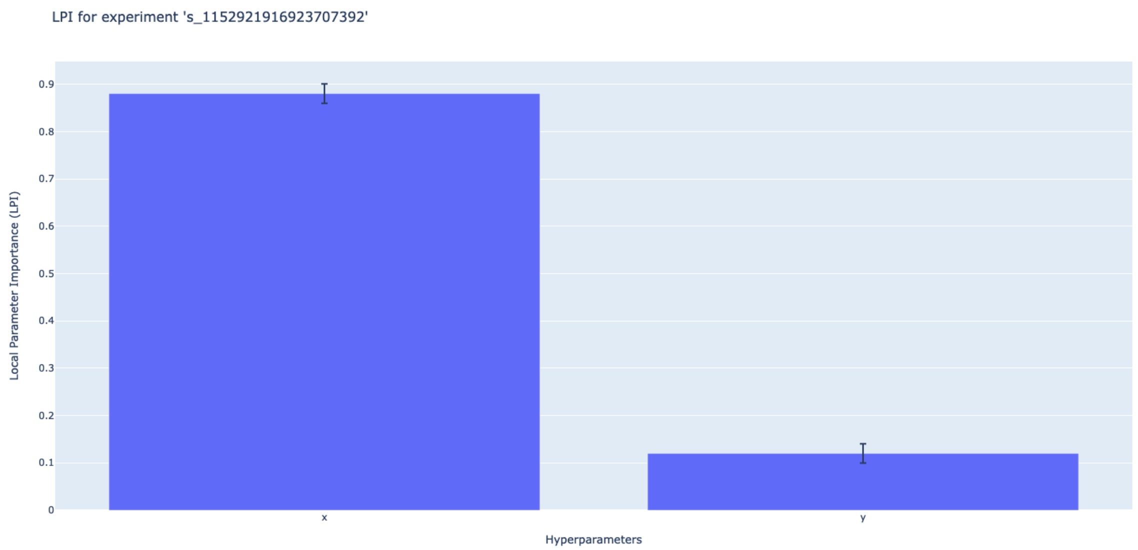




Binning end-cap

- This endcap layer is homogenous in Phi but the material varies in R
- Exactly what we observe with the optimisation
- Small score for low X value





Next steps

- A framework for the automatic optimisation of the material mapping has been put in place
- The optimisation might be a bit slow we might in the future try to speed it up
- Should perform a full test with the ODD with larger bin range and full stat -> compare the resulting map with the current one.
- Try other optimisation algorithm to see which one is optimal

BACKUP