Using Orion to tune the material mapping

Corentin Allaire

Link

https://acts.readthedocs.io/en/latest/howto/run_material_mapping.html

https://acts.readthedocs.io/en/latest/howto/run_material_mapping_auto_tuning.html#material-validation

Setup

```
pip install orion==0.2.2

source ../ACTS-PR/acts/build/python/setup.sh

export DD4HEP_LIBRARY_PATH+=:/Users/allaire/Desktop/ACTS-PR/acts/
build/thirdparty/OpenDataDetector/factory

export DYLD_LIBRARY_PATH+=:/Users/allaire/Desktop/ACTS-PR/acts/
build/thirdparty/OpenDataDetector/factory
```

Get the json geometry

```
../ACTS-PR/acts/build/bin/ActsExampleGeometryDD4hep -n1 -j1 --mat-output-
file geometry-map --dd4hep-input ../ACTS-PR/acts/thirdparty/
OpenDataDetector/xml/OpenDataDetector.xml --output-json --mat-output-
allmaterial true --mat-output-sensitives false
```

- Return a json file than can be edited to select the surface the material will be mapped on
- In the ODD case this is define at the DD4Hep level so we don't need to do it
- Favour approach 1 surface close to your sensor layer and boundary surfaces

Record the Geantino

```
../ACTS-PR/acts/build/bin/ActsExampleMaterialRecordingDD4hep -n100 -j1 --
dd4hep-input ../ACTS-PR/acts/thirdparty/OpenDataDetector/xml/
OpenDataDetector.xml --output-root --gen-nparticles 10
```

- Return a root file with all the material interaction used in the mapping
- Increasing the nparticles above 100 will speed up the file reading
- For a good quality map n =10000; gen-nparticles=1000

Auto tuning with Orion

```
python3 ../ACTS-PR/acts/Examples/Scripts/Python/
material_mapping_optimisation.py  --numberOfJobs 10 --topNumberOfEvents 100
--inputPath "." --outputPath "." --doPloting

python3 ../ACTS-PR/acts/Examples/Scripts/Python/
material_mapping_optimisation.py  --numberOfJobs 10 --topNumberOfEvents 100
--inputPath "." --outputPath "." --readCachedSurfaceInformation
```

- **numberOfJobs** the number of simultaneous jobs executed (this can be as high as you want, but if you have enough Core 40 works quite well).
- **topNumberOfEvents** the number of events from the input material track file used in the mapping can be changed (by default : 10000) it needs to be the same in all the calls to material_mapping_optimisation.
- **inputPath** the path to the input directory
- **outputPath** the path to the output directory
- **doPloting** if added at the end of the optimisation script, the optimal material will be computed and result plots will be created. (Should be used at the end of the optimisation).
- **readCachedSurfaceInformation** if added the material-surface association will be taken from the input material track file (doesn't work with geantino file, you need to use the material track file obtained from running the material mapping).

Validation

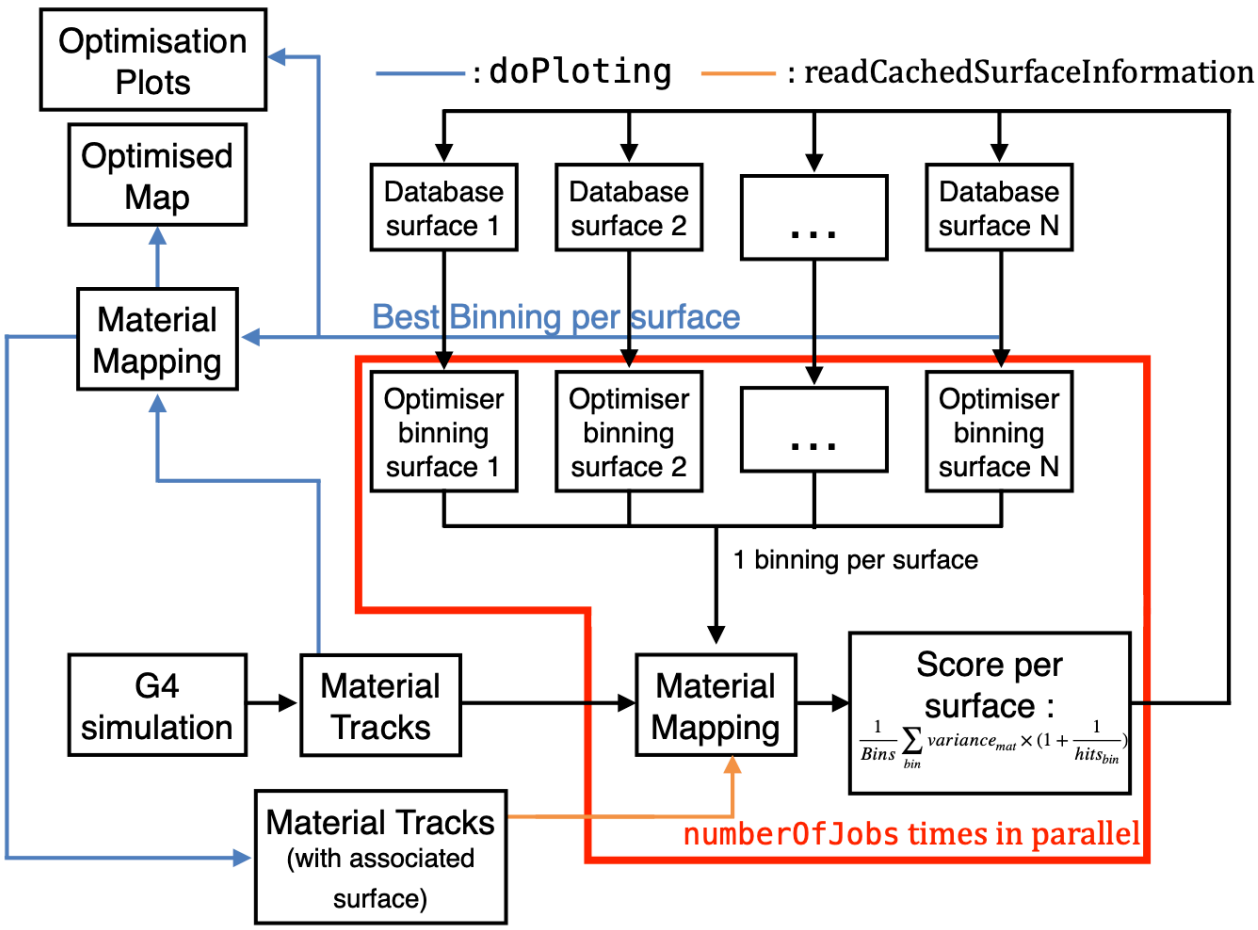
```
../ACTS-PR/acts/build/bin/ActsExampleMaterialValidationDD4hep -n1000 --
mat-input-type file --mat-input-file optimised-material-map.json --output-
root --mat-output-file propagation-material --dd4hep-input ../ACTS-PR/
acts/thirdparty/OpenDataDetector/xml/OpenDataDetector.xml --prop-z0-sigma
0.0 --prop-d0-sigma 0.0


mkdir Validation
root -l -b ../ACTS-PR/acts/Examples/Scripts/MaterialMapping/
Mat_map.C'("propagation-material.root","optimised-material-
map_tracks.root","Validation")'
.q
```

- The Validation propagate though the detector with the map and collect the material
- Don’t forget to set z0-sigma and d0-sigma to 0
- Many root macro available to evaluate the map quality (see the read the doc)

Auto-tuning of the material mapping

- One optimiser instance per surface we want to map onto
- Combined together to parametrise one mapping job
- N mapping running in parallel
- After the N finishes update the optimiser : score for each binning (variance in each bin)
- After enough iteration : get the best binning per surface -> run one last mapping



BACKUP