Metadata creation

Review of how metadata are created in

1. FD reco 2 campaign
2. PD-HD MC campaign (testing now)
3. PD-VD MC campaign (testing now)

AH, EP, production meeting March 21, 2024

For reference, in POMS metadata file was created by extracting metadata (SAM) from output root file by using ***extractor_prod.py*** (in duneutils)

```
dune_mc.mixerconfig + "  $mx  " ) >> vd_mc.json
postscript_7 = extractor_prod.py --infile $(ls
%(outname)s_${CLUSTER}_${PROCESS}_${FILETIMESTAMP}_%(tailname)s.root) --strip_parents --no_crc --appname reco
--appversion %(version)s --campaign %(campaign)s --appfamily art --requestid %(requestid)s --input_json vd_mc.json
> $(ls %(outname)s_${CLUSTER}_${PROCESS}_${FILETIMESTAMP}_%(tailname)s.root).json
postscript_8 = sed -i -e "s/DUNE_MC/dune_mc/g" -e "s/DUNE/dune/g" $(ls
%(outname)s_${CLUSTER}_${PROCESS}_${FILETIMESTAMP}_%(tailname)s.root).json
```

FD campaign summer 2023

pandora_metadata.py is also available (for .pndr file)

reco 2 (justIN)

MD creation is done in two steps:
1)metadata extraction with extraction_prod.py
2)metadata translation (SAM→ metacat)

```
extractor_prod.py --infile ${OUTFILE}.root --campaign ${CAMPAIGN:-fd_mc_2023a_reco2} \
                  --requestid ritm1780305 --no_crc > ${OUTFILE}.root.json
extractorExit=$?
if [ $extractorExit -eq 0 ] ; then
  # Success !
  echo "Extracted metadata"
else
  # Error -- exit immediately
  jobscriptExit=1
  echo "Failed to extract md"
  exit $extractorExit
fi

### Convert the metadata to metacat
##TODO -- make sure the RCDS behavior is correct
python $CONVERT_DIR/convert_metadata.py -i ${OUTFILE}.root.json \
                                        -c ${RECOFCL} -j old_md.json \
                                        --app "art.reco2" \
                                        --app_ver "${DUNE_VERSION:-v09_81_00d02}" \
                                        --det "${MDDETTYPE}" \
                                        --parent ${DID}
converterExit=$?
```

1

2

campaigns in preparation now: PD-HD and PD-VD, two different ways to produce metadata

1) PD-HD

```
# Make metadata
overrides="core.data_tier=full-reconstructed \
 core.application.version=${DUNE_VERSION} \
 dune.config_file=standard_reco_protodunehd.fcl \
 core.start_time=${recostart} \
 core.end_time=${recoend} \
 core.application.name=reco \
 core.application=art.reco \
 dune_mc.h4_input_file=H4_v34b_1GeV_-27.7_10M_${filenum}.root \
"

namespace=${JUSTIN_SCOPE:-"usertests"}

#-- ${filenum} \
python ${INPUT_DIR}/pdhd_meta_writer.py \
       --json ${INPUT_DIR}/pdhd_base_meta.json \
       --overrides ${overrides} \
       --event ${eventnum} \
       --nevents ${nevents} \
       --filenum ${filenum} \
       --jobid ${JUSTIN_JOBSUB_ID} \
       --past_fcls prod_beam_cosmics_1GeV_protodunehd.fcl \
                   standard_g4_protodunehd_stage1.fcl \
                   standard_g4_protodunehd_stage2.fcl \
                   standard_detsim_protodunehd.fcl \
       --past_apps gen g4_stage1 g4_stage2 detsim \
       -o ${reco_name}.root.json
```

- the script ***pdhd_meta_writer.py*** is used to write the json file
- values for different fields taken from the jobscript itself

2) PD-VD

```
extractor_prod.py --infile ${reco_name}.root  --appfamily art --appname reco --appversion  v09_82_02d01  --no_crc --
requestid ritm1998918  --input_json ${INPUT_DIR}/pdvd_input.json > ${reco_name}.root.ext.json  && sed -i -e 's/
stepfcl/protodunevd_reco/g' ${reco_name}.root.ext.json
/cvmfs/dune.opensciencegrid.org/products/dune/justin/pro/NULL/jobutils/pdjson2metadata ${reco_name}.root.ext.json
all-input-dids.txt > ${reco_name}.root.json
```

metadata are extracted (where possible) from output file with *extractor_prod*, then converted in metacat using *pdjson2metadata* . Testing now

it is possible to merge the 2 steps in a extractor_prod.py “speaking metacat”.
Work in progress (same work is still needed to produce a json in the right format, thanks Andrew)

- Both approaches are valid, but we have to choose one and **document it**, so that it can be used for these and next campaigns
- Identify the person/group in charge of developing and maintaining the scripts to generate the metadata file