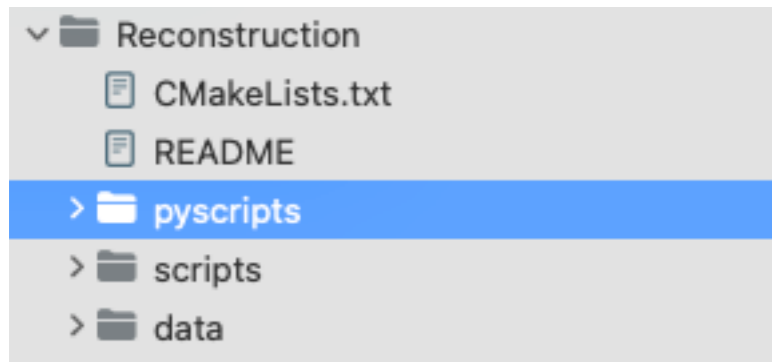


PyFOOT Framework

New Scripts

❑ Pyscript folder:



❑ CMakeLists.txt

```
#-----  
# build py-script  
#  
configure_file("${CMAKE_CURRENT_SOURCE_DIR}/pyscripts/load_libs.py.cmake" "${CMAKE_BINARY_DIR}/Reconstruction/load_libs.py"  
@ONLY)  
file(COPY ${CMAKE_CURRENT_SOURCE_DIR}/pyscripts/ DESTINATION ${CMAKE_BINARY_DIR}/Reconstruction/ PATTERN "*.cmake" EXCLUDE)
```

Loading libraries

□ load_libs.py(.cmake)

```
#!/usr/bin/env python3

print("*****")
print("          Load libraries for Reconstruction \n")
print("*****")

from ROOT import gSystem

# comes from configure
gSystem.Load("/Users/cfinck/Hadron/F00T/build/Libraries/lib/libTAGbase.so")

gSystem.Load("/Users/cfinck/Hadron/F00T/build/Libraries/lib/libTAMCbase.so")

. . .

gSystem.Load("/Users/cfinck/Hadron/F00T/build/Libraries/lib/libTACAbase.so")
gSystem.Load("/Users/cfinck/Hadron/F00T/build/Libraries/lib/libTAEDbase.so")
```

- ➔ Script for loading all libraries of F00T in python via ROOT
- ➔ Need to use the same version of python than for compiling Root

Example (i)

□ decode_analysis.py (translating from DecodeGlbAnalysis.cc)

```
# import libraries
import load_libs
import sys

# import required classes
from ROOT import TAGroot
from ROOT import TAGrecoManager
. . .

# looking for arguments
for i in range(0, len(argv)):
    if argv[i] == "-out":
        i += 1
        outFileNamg =str(argv[I])
. . .

TAGrecoManager.Instance(exp)
TAGrecoManager.GetPar().FromFile()
TAGrecoManager.GetPar().Print()

glbAna = GlobalAna(exp, runNb, fileIn, fileOut, mc)
glbAna.BeforeEventLoop()

if nSkipEv > 0:
    glbAna.GoEvent(nSkipEv)

glbAna.LoopEvent(nTotEv)
glbAna.AfterEventLoop()
```

➔ Simply translate C++ style into python

Example (ii)

Execute:

```
./decode_analysis.py -in run4287_GlbS_70kEvts_Out.root -out MassAnalysis.root -exp GSI2021 -run 4287
```

```
*****
                          Load libraries for Reconstruction
*****

===== Input Parameters =====

Global debug level: 0
Detectors included:
- Start Counter - Beam Monitor - Target - Vertex - MicroStrip Detector - ToF Wall - Calorimeter -

Info in <TAGanaManager::FromFile(): Open file ./config/GSI2021/TANAdetector.cfg for analysis configuration

Info in <TAGgeoTrafo::FromFile(): Open file ./geomaps/GSI2021/F00T_4271.geo for geometry
Info in <TGeoManager::TGeoManager>: Geometry F00T, F00T Geometry created

In file ./geomaps/GSI2021/TAGdetector_4271.geo the following beam parameters for a 160 beam impinging on C target have been set:
BeamEnergy:      0.400 GeV/u
BeamAtomicMass:  16
BeamAtomicNumber: 8
BeamMaterial:    "0"

TargetMaterial:  "C"
TargetThickness: 0.500 cm

Info in <TASTparGeo::FromFile(): Open file ./geomaps/GSI2021/TASTdetector.geo for geometry

. . .
```

➔ Same way as for C++ executable

Conclusion

- Simple interface (thanks to Root)
- Only need to load the different FOOT libraries (load_libs.py)
- Call all different methods (as in C++)