

1. build calypso
 - a. Checkout calypso
 - b. git clone --recursive ssh://[git@gitlab.cern.ch](https://gitlab.cern.ch):7999/faser/calypso.git or from your fork
 - c. Add ssh://[git@gitlab.cern.ch](https://gitlab.cern.ch):7999/keli/calypso.git
 - d. Checkout the branch: "skim"
 - e. Make "build" and "run" folder
 - f. Compile calypso:
 - g. In build folder, do:
 - i. cmake -DINSTALL_CONDB=ON -DCMAKE_INSTALL_PREFIX=./run ../calypso
 - h. Then make & make install
2. Make new condition database with mis-alignment
 - a. In run folder, copy "/afs/cern.ch/work/k/keli/public/ForAli/aligndb_copy.sh" to your run folder
 - b. Line 9 indicate the mis-alignment parameters for station/layer/modules for Faser-02
 - i. For example, "012", [shift_x, shift_y, shift_z, rotation_x, rotation_y, rotation_z], the "012" means the ID for station 0 layer 1 and module 2, unit for shift is mm for rotation is rad
 - c. source ./setup.sh
 - d. mkdir data/sqlite200/ in your run folder
 - e. Copy the official database: cp /cvmfs/faser.cern.ch/repo/sw/database/DBRelease/current/sqlite200/ALLP200.db ./run/data/sqlite200/
 - f. ./aligndb_copy.sh
 - g. Set the correct environment "export ATLAS_POOLCOND_PATH= /your run path/data", "your run path" is the full path
 - h. **Don't need to run .h, skip directly to condor jobs**
 Modify the input/output in bin/FaserCKF2Alignment.py and run it with python

Running Condor Jobs:

The following process uses Ke's calculated alignment parameters directly to go through all iterations.

1. First run maketemplate_daod.sh (this makes all the scripts you need for the jobs, **change this file for different data input**)
 - a. Input to this is faserkf_alignment_template_head.py, and faserkf_alignment_template_tail.py, run_faserkf_data_template.sh (adjust paths in this file)
 - b. Also input kfaligntemplate.sub
2. adjust path: mergeandextract_test_0.sh,
3. condor_submit kfalign_9073.sub (can run one .sh file to test)
 This will output a .txt file containing alignment parameters for station/layer/modules , and a Ntuple root file containing aligned track data.

**Note: make empty directories named “0”~”20”
and manually set output directory from 0 to 20 each time you run it**

Run one .sh file to Test:

1. change the eos and afs path to your corresponding path
2. change py file around line150: ConfigFlags.Exec.MaxEvents = -1, to 1000 to reduce the running number of events.

The following process will NOT use Ke’s generated alignment parameters, but rather generate each iteration and alignment parameter ourselves.

Running Alignment (iteration):

1. condor_submit_dag data_iter.dag

To check on condor jobs do: condor_q username

Things need to copy (for Yen-Ting):

In your Public Directory:

/afs/cern.ch/user/j/jlai/public/align_template

run_faserkf_data_template.sh

aligndb_copy.sh

aligndb_template_head.sh

aligndb_template_tail.sh

faserkf_alignment_template_head.py

faserkf_alignment_template_tail.py

extract_align_fix2layers_combine.C

makenewdb_fix2layers.C

kfaligntemplate.sub

maketemplate_data.sh

mergeandextract.sub

mergeandextract_test_0.sh

WriteAlignmentConfig_Faser01.py

WriteAlignmentConfig_Faser02.py

maketemplate_daod.sh

/eos/home-k/keli/Faser/alignment/results/inputforalign.txt

In your Skim Directory:

aligndb_copy.sh

aligndb_template_head.sh

aligndb_template_tail.sh

extract_align_fix2layers_combine.C

makenewdb_fix2layers.C

RDO: raw data object

AOD: analysis object data