FUNCTIONAL QC Pipeline for Human Connectome Acquisitions

Contents

Introduction

A “FUNCTIONAL_QC” pipeline, which executes after the acquisition validation pipeline and hence is called the Level2QC pipeline, was implemented to (i) perform rapid quality control (QC) to identify scans that are potential candidates for re-acquisition (i.e., before the subject leaves on Day 2), (ii) facilitate retrospective review of large numbers of fMRI scans within the IntraDB, and (iii) eventually support queries and subject filtering within the public ConnectomeDB based on scan quality. Immediately after conversion of the raw DICOMs to NIFTI format, the FUNCTIONAL_QC pipeline is launched for QC of both task and resting-state fMRI scans. [Ref: 1]

The Level2QC pipeline has been implemented so that each scan can be processed in parallel. The pipeline uses the Sun Grid Engine as the queuing system. The command used to submit the jobs is qsub with appropriate options. (If you have a queuing system other than SGE/PBS/OpenGrid, see Step 7 below).

Setting up Level2QC pipeline

In order to set up the Level2QC Pipeline, some tools need to be installed. The Level2QC Pipeline can be launched from your customized XNAT using pipeline launch user interface or via REST calls.

Enabling Level2QC pipeline consists for four main tasks:

- Installing various tools which are needed to run the QC pipeline,
- Setting up Grid Engine,
- Setting up XNAT to launch the QC pipeline,
- Setting up XNAT to view the QC results.

The following steps need to be followed to be able to launch the Level2QC Pipeline.

STEP 1: Install tools:

- EPD Python (tested version - Python 2.7.3 -- EPD 7.3-2 (64-bit))
- gnuplot (tested version - gnuplot 4.2 patchlevel 6)
- FSL
- BIRN Tools
- AFNI
- Tomcat
- XNAT version 1.6.4
- XNAT Pipeline Engine (the folder where XNAT Pipeline engine is installed would be referred to as PIPELINE_HOME)
- Install processing tools from https://github.com/Washington-University/hcp_functional_qc_tools
- Install OpenGrid or PBS/Torque or SGE as the queuing system.

STEP 2: Download the Level2QC Pipeline and its dependents

https://github.com/Washington-University/hcp_functional_qc_pipeline_customizations is the repository to use

(You will need the subfolders:

templates/misc/catalog/HCP_QC_PARALLEL

templates/misc/catalog/ToolsHCP)

Copy the above the above two folders to PIPELINE_HOME and re-run PIPELINE_HOME/setup.sh

Launching the Level2QC pipeline

STEP 3: Add the pipeline to XNAT site

- Login to XNAT as Admin.
- Navigate to Administer -> Pipelines
- Click on Add Pipeline to Repository

- Set the path to the pipeline XML

Enter Path to Pipeline descriptor xml: PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/Level2QCLauncher_v2.0.xml

Enter Name of the custom webpage to launch this pipeline: LEAVE THIS BLANK

Click on Add

You will get a message "... was added successfully"

If you did everything correct, you will see Level2QCLauncher listed in the pipeline repository.

STEP 4: Setup Level2QC Pipeline for XNAT Project

- As a project owner, navigate to your project, say PROJECTA.
- Click on Pipelines tab.
- Click on Add More Pipelines
- Click on Add (against Level2QCLauncher_v2.0 pipeline)
- Set the value for the pipeline parameters:
  functional_scan_type : the value of this variable should be the scan-type of the FMRI scans

- Click on Submit

STEP 5: Configure the pipeline and the tools

From the file,

PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/config/level2qc.config

copy the templates (lines 40-78) to create the following setup scripts in a folder, whose location is set to the value of the parameter SETUP_SCRIPTS:

- epd-python_setup.sh
- bxh_xcede_tools_setup.sh
- AFNI_setup.sh
- fsl5_setup.sh

Edit the file
PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/config/level2qc.config

and set the values of the following parameters for your site:

- PATH_TO_PYTHON_SCRIPT (set the value to the location where you install hcp_functional _qc_tools)
- SETUP_SCRIPTS
- FSL_DIR

STEP 6: Launch the pipeline

- Navigate to a session in the ProjectA.
- Click on Build
- Select Level2QC Pipeline
- Enter submit

STEP 6a: Launch the pipeline via REST

Use any tool (curl, XnatRestClient, XnatDataClient) to POST to the URI

data/archive/projects/PROJECT_ID_HERE/pipelines/Level2QCLauncher_v2.0/experiments/XNAT_EXPERIMENT_ID_HERE

If you use XnatRestClient (located in PIPELINE_HOME/xnat-tools), the command to launch the pipeline via REST would be:

PIPELINE_HOME/xnat-tools/XnatRestClient \

-u XNAT_USER_NAME_HERE \

-p XNAT_USER_PASSWORD_HERE \

-host XNAT_HOST_URL_HERE \

-m POST \

-remote \
"data/archive/projects/PROJECT_ID_HERE/pipelines/Level2QCLauncher_v2.0/experiments/XNAT_EXPERIMENT_ID_HERE"

STEP 7: Adapt the pipeline to a different queuing system

The file

PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/***functional_batch_script_generator.sh***

is a wrapper script which takes care of job dependencies and is responsible for submitting the jobs. This script sources

PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/config/***level2qc.config***

which contains queuing system specific parameters and other site specific configuration parameters.

The above two files would need to be modified if you use a different scheduling system.

More details about the Level2 QC Pipeline:

For a given scan, the order of commands in *functional_batch_script_generator.sh* file is:

- Get the scan details
- Check if the number of frames in the scan is more than the minimum expected (configured in *level2qc.config* file)
- Get the scan DICOM files
- Get the scan NIFTI files
- Launch the script to compute Fourier Statistics (Mean, Std per volume) and generate plots
- Launch the script to compute the motion outliers (RMS, DVAR, tSNR, SD etc)

- Launch the script to invoke BIRN fMRI QA script
- Launch the script to generate the Wavelet Kurtosis based measures.
- Launch the script which generated the QC Assessor and uploads the output files back to XNAT.

Apart from the above steps, the Level2QC pipeline can check for outliers in the QC Assessor. The outlier test is done based on a schematron file which lists the values to be tested. The outlier component of the pipeline, sends an email with the outlier report ONLY if it detects outliers.

Pipeline Output

After the pipeline completes (successfully), you are expected to have the following subfolders (within ASSESSORS) for EACH functional scan:

- BIRN_DATA
- FOURIER_COEFFICIENTS_DATA
- MOTIONOUTLIER_DATA
- WAVELETSTATISTICS_DATA

Files in output folders:

BIRN_DATA: would contains all output generated by the BIRN tool, fmriqa_generate.pl, and some convenience files so that plots can be displayed on the XNAT UI

FOURIER_COEFFICIENTS_DATA: would contain output generated by FourierSlope.py and plots generated by generate_plots_functional.csh. FourierSlope.py generates the files:

*_FourierSlopeStatistics_4_MeanStd.txt

(contains per volume Mean and Std)

*_FourierSlopeStatistics_4.txt

(contains are per volume per slice measures of Slope Intercept r-value p-value stderr)

*_FourierSlopeStatistics_4_MeanStd_stripped.dat

(*_FourierSlopeStatistics_4_MeanStd_stripped.dat is used by the plot script and is nothing but *_FourierSlopeStatistics_4_MeanStd.txt with first header row removed)

MOTIONOUTLIER_DATA: would contain output generated by motion_qc.sh script and would contains various flies with measures of DVARS, tSNR etc. and some plots.

WAVELETSTATISTICS_DATA would contain output generated by WaveletStatistics.py and some plots generated by generate_plots.csh.

Scripts which generate various outputs

The wrapper scripts which generate the output files mentioned above are:

BIRN:
PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/functional_batch_birn_human.sh

FOURIER STATS:
PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/functional_batch_fourierstatistics.sh

MOTION OUTLIER STAT:
PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/functional_batch_motionoutlier.sh

Wavelet Stats:
PIPELINE_HOME/catalog/HCP_QC_PARALLEL/Wrapper_QC/resources/functional_batch_wavelet.sh

Displaying QC Assessor results in XNAT

Velocity and Java files required to display the QC results in XNAT are being made available as a module [Ref 2]. The module is available at:

https://github.com/Washington-University/hcp_functional_qc_xnat_module

Files which control the display of the QC Statistics:

qc_summary.vm : responsible for summary display on MR Session report page

QCPopup.vm : responsible for popup window display of various scan specific plots and measures.

(both the files are located at src/templates/screens)

REFERENCES

1. Human Connectome Project informatics: quality control, database services, and data visualization. Daniel S Marcus, Michael P Harms, Abraham Z Snyder, Mark Jenkinson, J Anthony Wilson, Matthew F Glasser, Deanna M Barch, Kevin A Archie, Gregory C Burgess, Mohana Ramaratnam, Michael Hodge, William Horton, Rick Herrick, Timothy Olsen, Michael McKay, Matthew House, Michael Hileman, Erin Reid, John Harwell, Timothy Coalson, Jon Schindler, Jennifer S Elam, Sandra W Curtiss, David C Van Essen, WU-Minn HCP Consortium

NeuroImage, 2013-05-28 | PMID: 23707591

2. https://wiki.xnat.org/display/Marketplace/How+to+Create+an+XNAT+Module+-+Developer+Guide