v01 - May 2024

LocaliteQA toolbox Instructions

ATTN: When using this toolbox, please cite Burns & Hermiller and the toolbox GitHub
https://github.com/HermillerLab/LocaliteQA

The LocaliteQA toolbox assumes that the user is familiar with the terminology and experienced with the functionality of the Localite neuronavigation software. It assumes that in one TMS session, a single TMS protocol (i.e., sequence train) is delivered, targeting the same location. The current version of the toolbox is not intended to evaluate single pulses. Aspects of the LocaliteQA toolbox will differ based on how the users create TMS sessions in Localite (e.g., “Create a new session” versus “Continue session” versus “Continue session as copy”). Users should familiarize themselves with their own Localite data before attempting to use the LocaliteQA toolbox. We offer suggestions for optimal use below.

LocaliteQA toolbox structure. The LocaliteQA toolbox directory contains the LocaliteQA.m ‘wrapper script’ which the user will launch and execute in MATLAB. The wrapper script will read Localite data copied into the raw_data directory, call the functions scripts saved in the QA_Functions directory, and save the output in the QA_Results directory (**Figure 1)**. Do not change any directory or file names in the toolbox. Please follow the instructions below (also provided as a pdf in the toolbox directory) when using the LocaliteQA. Any future updates to these instructions corresponding to amended toolbox code will be posted in the github (https://github.com/HermillerLab/LocaliteQA).

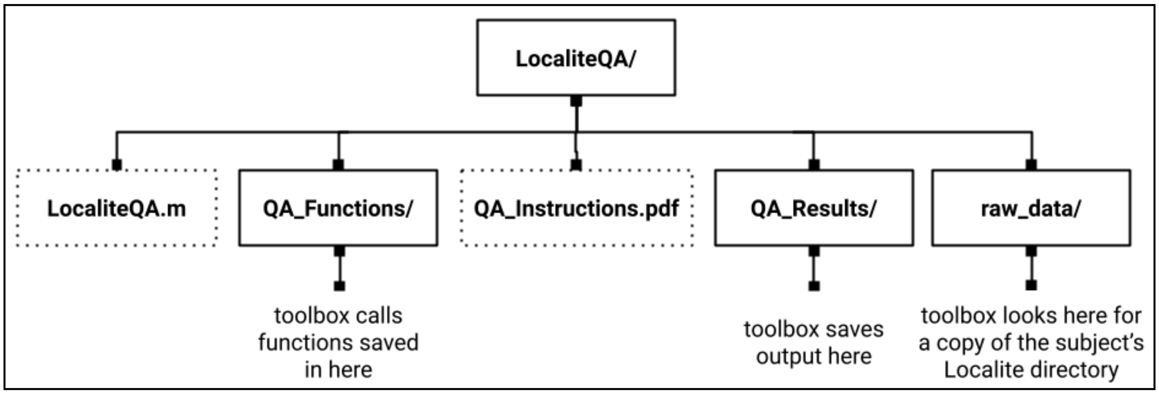

LocaliteQA/
LocaliteQA.m
QA_Functions/
QA_Instructions.pdf
QA_Results/
raw_data/
toolbox calls functions saved in here
toolbox saves output here
toolbox looks here for a copy of the subject’s Localite directory

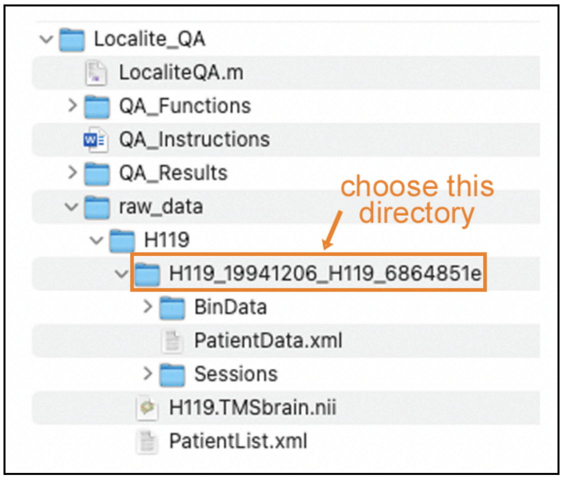
Figure 1. LocaliteQA toolbox directory structure. See text for complete details.

Ensure pulse locations are recorded in Localite during the TMS session. Prior to using the Localite QA toolbox, users must first collect the TMS data in the Localite software. Importantly, the user must initiate the recording of each pulse in Localite (see **Tips for data collection** section below). Prior to beginning the stimulation protocol, open the Stimulation panel in Localite. Under Record Stimulation Markers, click “Start”; the system will now record the location of the pulses. At the end of the sequence, click “Stop” to end the recording. All pulses delivered between this time will be saved in the same trigger file.

Transfer Localite data to LocaliteQA toolbox. After TMS data collection, copy the subject’s entire directory from the Localite computer and transfer the folder into your LocaliteQA/raw_data directory. Do not change any folder or file names. Toolbox users may make additional study-specific directories within the raw_data directory to further organize their folder.

Launch and execute the MATLAB LocaliteQA.m script. This is a wrapper script, which will call the function scripts saved in the QA_Functions directory. The analysis is interactive and requires the user to respond to prompts in the terminal, following these steps:

1) **Choose your subject’s session data.** You will be prompted to select the subject’s directory you are analyzing via a popup menu. Direct the prompt to the LocaliteQA/raw_data directory and into your subject’s folder. You must select a folder that was created by Localite; this folder’s name begins with the label (TMS operator provided to Localite during session setup), followed by letters/numbers (see **Figure 2** for example). This selected folder must contain (at minimum) a BinData directory, Sessions directory, and a PatientData.xml file, all of which are generated by Localite. Note that how the TMS sessions are initiated in Localite by the TMS operator may influence how Localite structures the contents of the subject directory.



```
Localite_QA
  LocaliteQA.m
  QA_Functions
  QA_Instructions
  QA_Results
  raw_data
    H119
      H119_19941206_H119_6864851e
        BinData
        PatientData.xml
        Sessions
      H119.TMSbrain.nii
      PatientList.xml
```

choose this directory

Figure 2. The LocaliteQA toolbox is looking for the Localite-generated folder (shown here) inside the subject’s directory.

2) **Enter label for results folder.** In the MATLAB command window, you will be prompted to provide a name (e.g., subject ID) for the folder that will be created in the QA_Results directory. A timestamp will be appended to the folder name. All toolbox output for this subject will be saved in this folder, where the automatically generated results will be saved.

3) **Indicate which sessions to analyze.** In the MATLAB command window, the label (provided to Localite during the session creation) and date of each session found in the selected directory will be listed (see below). You will be prompted to type the corresponding number(s) for the session(s) you intend the toolbox to assess.

```
SESSIONS
        {‘C13_MT’}
        {‘C13_A1’}
        {‘C13_A2’}
        {‘C13_A3’}

DATES
        {‘20190709’}
        {‘20190710’}
        {‘20190713’}
        {‘20190715’}

Enter session(s) (e.g., 1,2):
```

User response: `2,3`

4) **Indicate which trigger file to read.** In the MATLAB command window, the size of each trigger file created in each selected session will be listed. Localite will create several small trigger files that are empty and the toolbox will ignore; here, the toolbox user will tell LocaliteQA which files to assess. To automatically select the largest file, hit enter. Depending on how the user creates the Localite session, there may be multiple files to select from. You can only choose one trigger file per session.

```
C13_A2:
     {[1192]}    {[192844]}

Press ENTER to use last file in session (or type file number "{[####]}"):
```

User response: `<enter key>`

```
C13_A3:
     {[1192]}       {[2150]}       {[1192]}       {[1192]}       {[1191]}
     {[192582]} {[1191]}

Press ENTER to use last file in session (or type file number "{[####]}"):
```

User response: `{{192582]}`

5) **Provide the requested thresholds.** In the MATLAB command window, you will be prompted to type the value of two thresholds. The first threshold sets the allowable distance (mm) between the location of the TMS pulses and the position of the entry/target and instrument marker. This is used to test accuracy as the graph will show how many pulses fall above and below the threshold. The second threshold sets the allowable distance (mm) between the coordinates of the predefined target and the coordinates that the toolbox estimates was actually targeted based on the actual location of the TMS pulses (see below). The toolbox uses these thresholds to identify off-target pulses.

```
Enter the “off-target” threshold (distance in mm) for each pulse location
relative to the Entry and Instrument Marker (e.g., 3):
```

User response: `3`

```
Enter the “off-target” threshold (distance in mm) for each pulse location
relative to the estimated actual stimulation site (e.g., 1):
```

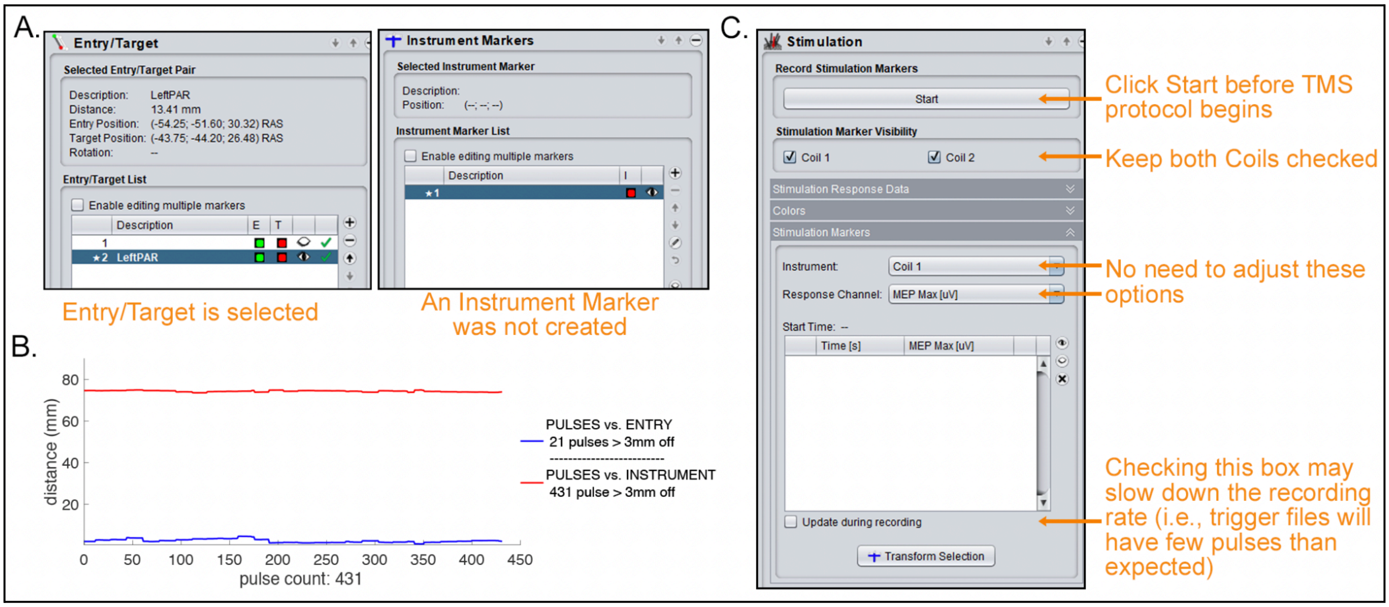
User response: `1`

6) **Review the output.** The LocaliteQA toolbox creates a folder in the QA_Results directory using the user-provided label (step 2) and timestamp, and saves all output into this folder. Output files include the extracted data from Localite in .mat format (e.g., session info, di/dt, entry and target coordinates, instrument marker coordinates, coordinates of each TMS pulse, and RMS deviation from Localite registration), a .txt file containing all the output to the MATLAB command window from the toolbox, and corresponding plots in .png and .fig format. See Figure 2 in the manuscript by Burns and Hermiller for example output.

Tips for data collection:

- If running multiple TMS sessions for a given subject, create each session in Localite as “Continue session as copy”. This will ensure that previously set entry/target and instrument markers are carried over, but will create a new session folder.

- Do not use spaces when naming sessions in Localite. This includes names such as “Copy of subj_123” (as generated automatically by Localite). If spaces are included in the name, an error will occur.
- Make note of the session name (and date), as this will help you identify it in the LocaliteQA prompts
- If you have multiple entry/target pairs and/or instrument markers, make sure you have the correct one selected before the stimulation session (see **Figure 3A**); LocaliteQA will use what is selected as the set that it assess pulse locations against. **Figure 3B** shows an example of a Localite stimulation session with an instrument marker selected that did not correspond to the entry/target that was used.
- Click “Start” in the Stimulation tab (“Record Stimulation Markers”) - if this is not done, then pulse locations will not be recorded. Keep both coils (Coil 0, Coil 1) selected (i.e., do not uncheck a box); the LocaliteQA toolbox expects data files from both coils (**Figure 3C**)
- Do not rename directories or files created by Localite when copying and transferring the data; the LocalteQA toolbox relies on the original directory structure and the files created by Localite.



Description: LeftPAR
Distance: 13.41 mm
Entry Position: (-54.25; -51.60; 30.32) RAS
Target Position: (-43.75; -44.20; 26.48) RAS
Rotation: --

□ Enable editing multiple markers

Description	E	T
1	🟩	🟥
★2 LeftPAR	🟩	🟥

Entry/Target is selected

Description:
Position: (--; --; --)

□ Enable editing multiple markers

Description	I
★1	🟥

An Instrument Marker was not created

PULSES vs. ENTRY
21 pulses > 3mm off

PULSES vs. INSTRUMENT
431 pulse > 3mm off

☑ Coil 1 ☑ Coil 2

Instrument: Coil 1
Response Channel: MEP Max [uV]
Start Time: --

Time [s]	MEP Max [uV]

□ Update during recording

Click Start before TMS protocol begins

Keep both Coils checked

No need to adjust these options

Checking this box may slow down the recording rate (i.e., trigger files will have few pulses than expected)

Figure 3. Suggested Localite session operations to ensure compatibility with LocaliteQA toolbox functionality. A) Example of an Entry/Target that is selected by the TMS user (left panel) and an example of an empty Instrument Marker (right panel). **B)** Example output where the Instrument Marker did not correspond to the selected Entry/Target (see Fig. 4A). Here, the operator used the entry/target for real-time coil location monitoring. **C)** Localite operations to ensure that each pulse location is recorded during a TMS protocol.

LocaliteQA Toolbox Functions:

- **parseXML.m -** reads an XML file and converts its contents into a MATLAB structure, preserving the original hierarchy and details for easy analysis and modification in MATLAB. The original function can be found at http://www.mathworks.com/help/matlab/ref/xmlread.html#zmw57dd0e797590.
- **read_dir.m -** processes subject session data by locating session timestamps, parsing associated XML files, and organizing the extracted information into a MATLAB structure for analysis.
- **read_entry.m -** extracts and organizes entry and target coordinates from XML files for subject sessions, iterating over session indices, parsing the XML files, and storing the data in MATLAB structures for further analysis and verification.
- **read_instrmt.m -** reads XML files containing instrument markers, extracts and organizes the data into a matrix for specified sessions, and saves the data and timestamp into a .mat file.

- **read_RMSdev.m -** processes RMS deviation data from XML and text files, extracts and organizes transformation matrices and RMS deviations for each session, handles missing data, and saves the results in a .mat file.
- **rem_NaN_rotmax.m -** replaces NaN values in the last three columns of the final row of a reshaped rotation matrix with zeros, ensuring compatibility with various matrix dimensions and returning the modified matrix.
- **read_triggers.m -** extracts and processes trigger markers from Localite TMS session XML files, detecting session breaks and distinguishing between different coils. It then displays the average trigger locations and recorded di/dt values, saving the data and timestamps into .mat files.
- **calc_dev.m -** calculates and visualizes deviations and variances in triggers and pulse data, removing problematic triggers and processing the data to display results and generate plots with reference thresholds. These results and plots, which illustrate the distances between pulses, entry points, instrument markers, and targets, are organized in the Command Window and saved for further analysis.