Atlas-based Imaging Data Analysis pipeline for Quality Control of Animal MRI Data

AIDAqc

v1.1

Code: Aref Kalantari, Michael Diedenhofen, Markus Aswendt

Manual: Aref Kalantari

February 2022

Department of Neurology
University Hospital Cologne

Contents

1 Introduction

It can be challenging to acquire MR images of consistent quality or to decide in the screening of large databases, which dataset is of sufficient quality for further processing. Manual screening without quantitative criteria is strictly user-dependent and not feasible for huge databases. In contrast to clinical MRI, in preclinical, animal imaging, there is no consensus on standardization of quality control measures or categorization of good vs. bad quality images.

The Atlas-based Processing Pipeline for Quality Control of Animal MRI Data(AIDAqc) was developed for measuring and standardizing the quality of mouse brain MRI in a dynamic and novel way. AIDAqc works with T2-weighted MRI (T2w), diffusion-weighted MRI or diffusion tensor imaging (DTI), and resting-state functional MRI (fMRI).

Here, we developed a tool in Python to create a basic overview of MR image datasets including information about the SNR, temporal SNR, spatial resolution, and movement severity (Figure 1). Currently, this tool covers T2w, DWI, and fMRI sequences.

a) Data Parsing:

The user sets the input path and the program will parse iteratively through all subfolders with a list of all raw MR data as its result. After parsing, only those MR files chosen by the user between the options of T2w, DTI, or fMRI data are selected, and duplicates are eliminated. Finally, an excel file is created with the storage path of every selected file, which will be the input of the next step.

b) Parameter Extraction:

In this step, SNR is calculated for T2w and DTI images, tSNR, and movement severity for fMRI images. Mutual information was used as a metric to calculate movement severity.

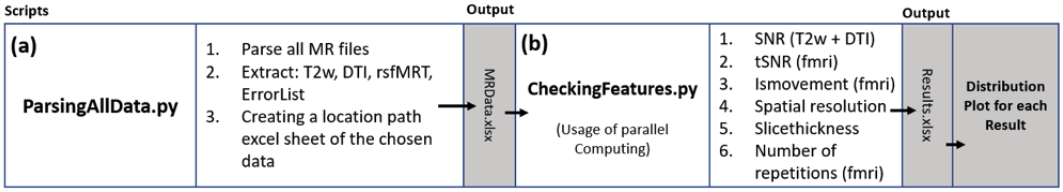

Figure 1: Pipeline workflow: All of the necessary functions are implemented in this module, SNR is calculated by using the Chang method. (a) In this stage, all of the available MR files are parsed and located. The final result is saved in an Excel document called "MRData" which contains separate sheets for T2w, DWI, or fMRI measurements. (b) The main block, here all of the parameters are calculated: SNR, tSNR, and MS. Spatial Resolution, slice thickness, number of repetitions are also extracted. The final output is a second Excel document called "QuiC_Data_Result_Processed_featurs.xlsx".

2 Installation

1. Download repository by using this link. The project folder contains the python scripts necessary for the quality measurement.

2. Download & Install Python 3.6 or higher using Anaconda.

3. Importing AIDAqc environment: After the installation of anaconda navigator, we have to import the necessary environment. This can be done by importing the "aidaqc.yaml" file at the *Environments tab*, *import* and choosing *local drive* if the file is downloaded from GitHub to a local drive.

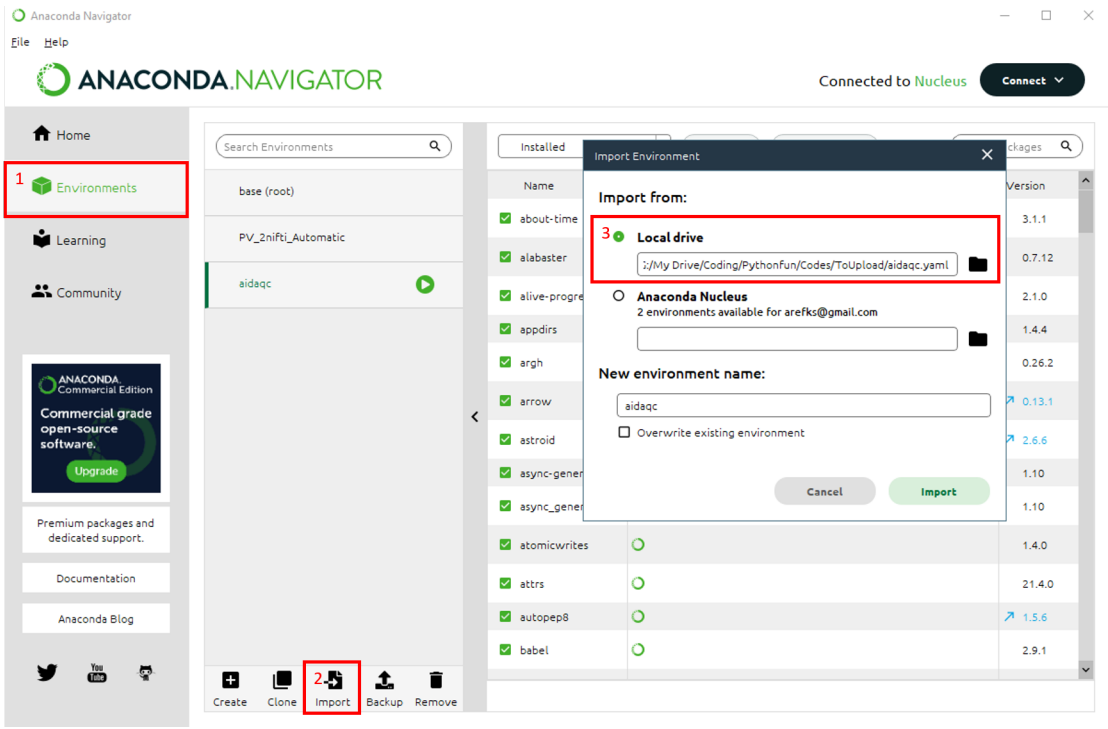

Figure 2: Importing the environment downloaded from GitHub: Environments/Import/Local drive .

3 Functions

List of functions:

- **ParsingAllrawData:** Parser for identifying the location of T2w, DTI and fMRI files based on their sequence name. The inputs of this function are two strings. An initial path which can be set by the user. The program will use this path as a starting point and search for MR files in every possible subsequent folder after this initial path. The second input is a saving path which is the location where the results should be saved. Figure 8 shows an exemplary use of the author.

- **CheckingFeatures_final:** After *ParsingAllrawData.py* has been used. An Excel file with the corresponding addresses is created at the defined

location from the user. This Excel file can be used as an input for this function.

- **QCPlotting:** In this function, the produced Excel file from the last step is used as an input to calculate corresponding histograms of the measured quality features. The resulting illustrations are saved in one pdf file at the same location as the excel file (see 3).

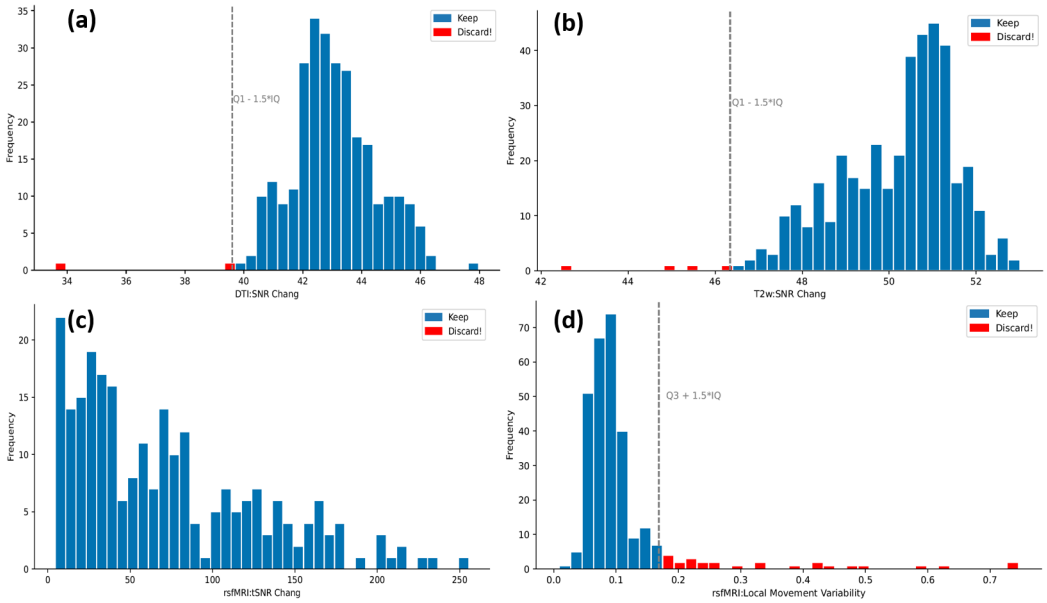

Figure 3: Exemplary statistical plots. The grey vertical line indicates the threshold of good vs bad data based on the statistical definition of "outliers". The bars with the red color indicate those files which should be discarded. (a) Histogram of SNR values of the DTI dataset (b) Histogram of SNR values of the T2w dataset (c) Histogram of tSNR values of the rsfMRI dataset (d) Histogram of the movement variability of the rsfMRI dataset, calculated based on mutual information

Attention: All program examples are only listed with the mandatory input parameters. For more details/help, call python `.../python <command> -h`. After a successful download, you can choose to either process batches of files step by step or automate the processing in a cascading way. Processing steps can simply be divided into three stages:

- **Stage I:**
  1) Parsing[1]
  2) Extraction (of wanted sequences)
  3) Cleansing (Eliminating duplicate files)
  4) Saving (`QuiC_Data_Result.xlsx`)
- **Stage II:**
  1) Reading parsed files
  2) Conversion to nifti format
  3) Calculating quality parameters
  4) Saving (`QuiC_Data_Result_Processed_featurs.xlsx`)
- **Stage III:**
  1) Reading calculated qc features
  2) Statistical evaluation
  3) Creating corresponding plots
  4) Saving (`QCfigures.pdf`)

4 Exemplary workflow

Here we want to show how the pipeline can be used. The steps are explained subsequently.

1) Download the sample data set from GIN via this link.
2) If not already done, download the repository from this link and follow the installation steps described in the Installation chapter.

[1]Parsing can be translated to searching. Here it means the program *searches* or *parses* all of the available directories for a wanted file.

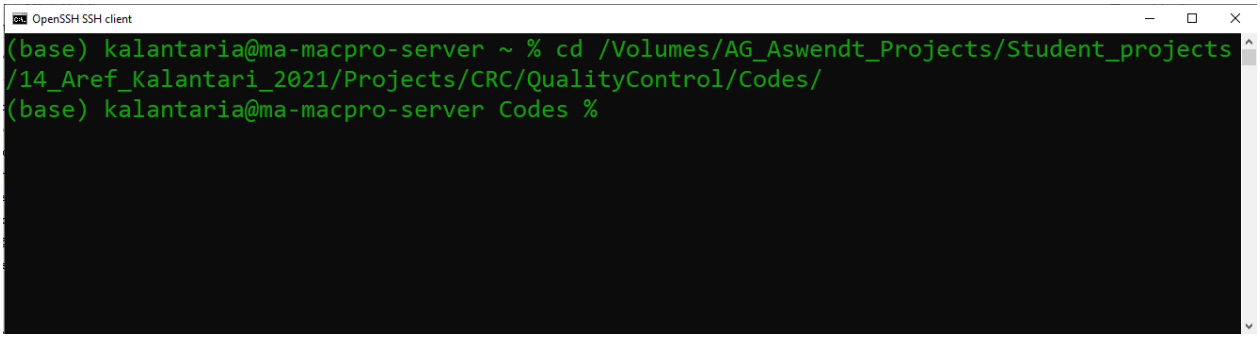

Figure 4: Changing the terminal's directory to the folder containing the python scripts downloaded from GitHub.

4) activate the aidaqc environment by typing in the following command (figure 5).

```
conda activate aidaqc
```

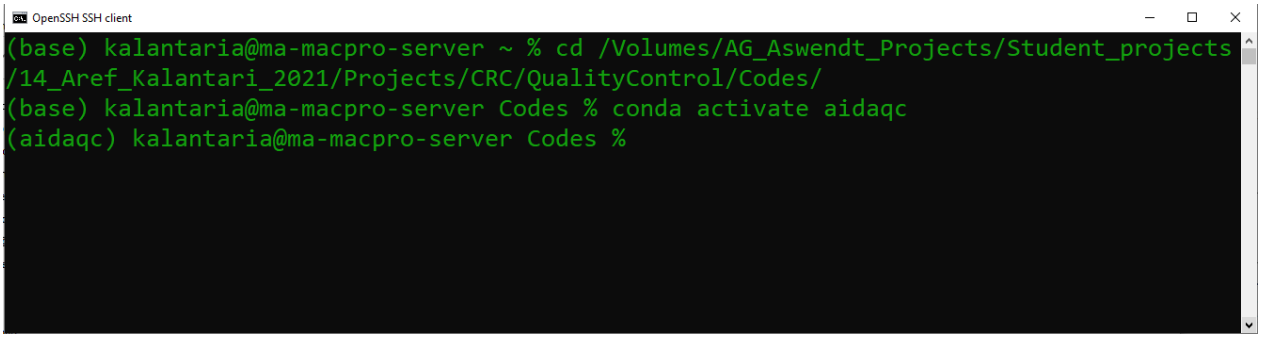

Figure 5: Activating the aidaqc environment. Note that the installation of anaconda and loading the `.yaml` *file is a prerequisite for this step to work (see 3).*

5) After activating the environment it is best to check if the environment has been installed correctly and to see if there are any errors accruing in the script. This can be done by using the help option of the function by using the following command. Additionally, a short explanation can also be seen.

```
python ParsingAllrawData.py -h
```

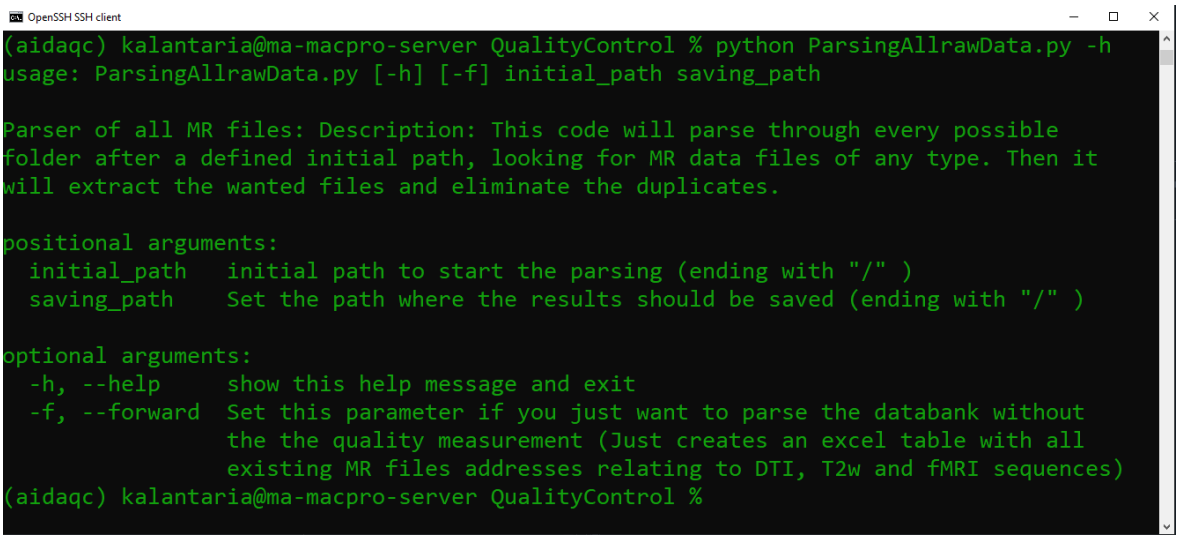

Figure 6: Using ParsingAllrawData.py with the help option.

6) After activating the environment the process can be started by typing in the following command using the script *ParsingAllrawData.py*. Pay attention to the space that is between the two inputs and note that both initial and saving paths have to end with a forward slash (/).

```
✓python ParsingAllrawData.py <!!initial_path!!>/ <!!saving_path!!>/
✗python ParsingAllrawData.py <!!initial_path!!>/<!!saving_path!!>/
✗python ParsingAllrawData.py <!!initial_path!!> <!!saving_path!!>
```

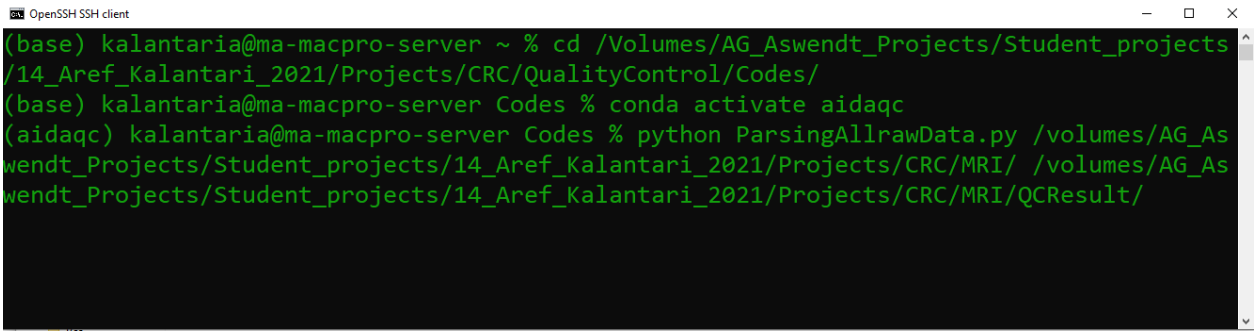

Figure 7: Using ParsingAllrawData.py with its two inputs.

7) After the program has parsed the files and calculated the QC features, three excel files and one pdf file will be created at the defined location set by the user. As can be seen in figure 8 the pipeline informs the user at what stage the pipeline is and how long the processing will approximately take.

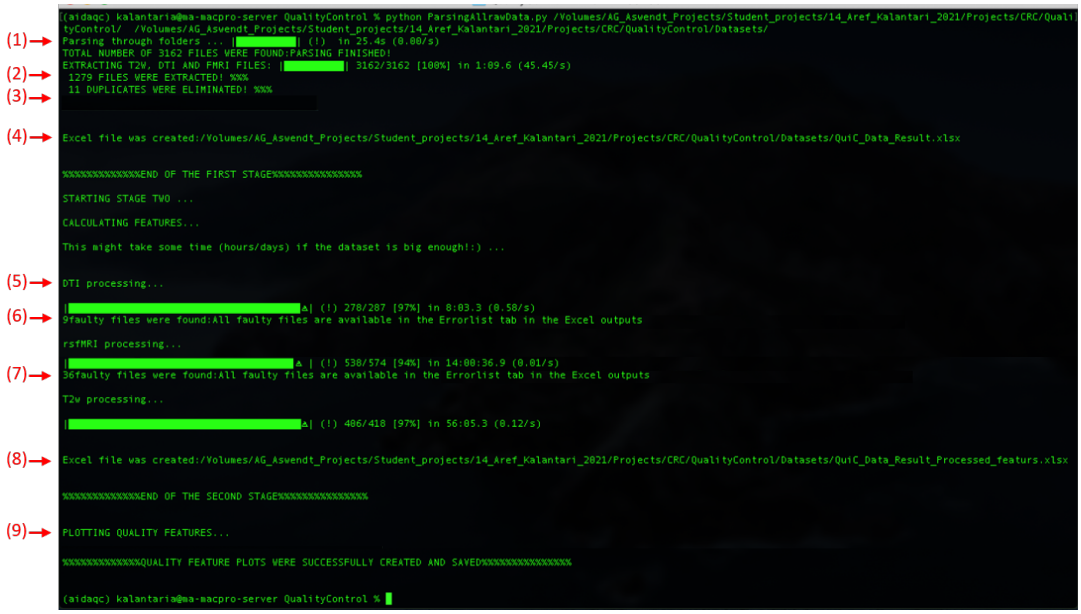

Figure 8: Structural overview and summary of the pipeline. 1) Searching for all MR files available 2) Extracting the sequences related to T2w, DTI, and fMRI measurements from the parsed files 3) Duplicate MR files will be only considered once and any file address of copies is eliminated. 4) Final Excel file of stage (I) containing all addresses is created at the defined location. 5) In this part all of the file addresses of part 4 are processed sequence-based. 6) Some files which can contain faulty data or faulty structures with incomplete data won't cause any problems and will also be saved as an Error_data tab in the corresponding Excel table. 7) Same as in 6 faulty fMRI data will be filtered out. 8) Final Excel file in stage (II) containing all of the calculated QC features is saved in this stage. 9) Finally, statistical plots are created and all of them are saved in a pdf file.

5 FAQ

Q1) How is the SNR of the T2w and DTI sequences calculated?

The SNR is calculated based on the chang method. Simply said this method calculated the SNR without needing to define regions in the image. As for this pipeline, the input T2w dataset usually consists of more the one slice of the brain. For example, we have an image set with a dimension of 128×128×20, the pipeline extracts the 5 best subsequent slices with the highest average value indicating a good image of the brain. Usually, the best slices are the middle ones, meaning slices 8 to 12. If the first or the last slices are the highest slices a warning like in figure 9 will be shown in the terminal and this can indicate a faulty dataset. The reason for this happening can be two things, either the dataset has just one slice which makes no problem and the error can be ignored or the dataset has more slices and it was a faulty measurement where the position of the slice was selected wrongly.

Figure 9: Warning for a dataset in which the first or the last slices have the most signal.

Q2) How is the tSNR calculated?

To calculate the temporal signal to noise ratio multiple approaches are available in the literature, most of them are using normal SNR measurements but add some calculation steps to make it legitimate to be called tSNR. To understand how it is calculated in this pipeline, let's assume a simple example again. Consider an fMRI dataset with dimensions of 128 × 128 × 20 × 500 with 20 being the slices and 500 being the temporal time points. Similar to the T2w approach, the

4 best subsequent slices are chosen based on average image intensity. After this step, SNR is calculated again based on the chang method for every time point of each slice. Based on this tSNR method, the tSNR for one slice and 500-time points is then defined as the mean SNR divided by the standard deviation.

$$tSNR = \frac{\mu}{\sigma} = \frac{\mu}{\sqrt{1/N \sum_{i=1}^{N} (x_i - \mu)^2}} \tag{1}$$

This is done for the 4 chosen slices and the final tSNR is the averaged value of those.

Q3) How is the Movement severity calculated?

Mutual information (MI) was used to calculate the movement severity in the resting state MR measurements. Check out Mutual information as an image matching metric to better understand the concept of *Mutual information* in image analysis. To explain how MI was used in this pipeline, it's easier to use our example dataset with a dimension of $128 \times 128 \times 20 \times 500$. The question is how much movement has happened between the image at $T = t$ and $T = t + 1$. The four best slices are chosen based on the approach already explained in Q1 and Q2. The image of the first time point is then used as the reference image and all of the following 499 images are compared to the first one by calculating the MI between them. If the MI is high, it means the movement was small. If it's low then the movement was relatively big. The standard deviation of all the MI values is then used as a metric for the overall movement severity. So the final output observable in the Excel file are standard deviation values.

Q4) How does the parsing work in detail?

The parsing technique of this pipeline can be difficult to understand. With the help of figure 10 it gets clearer how the parsing works.

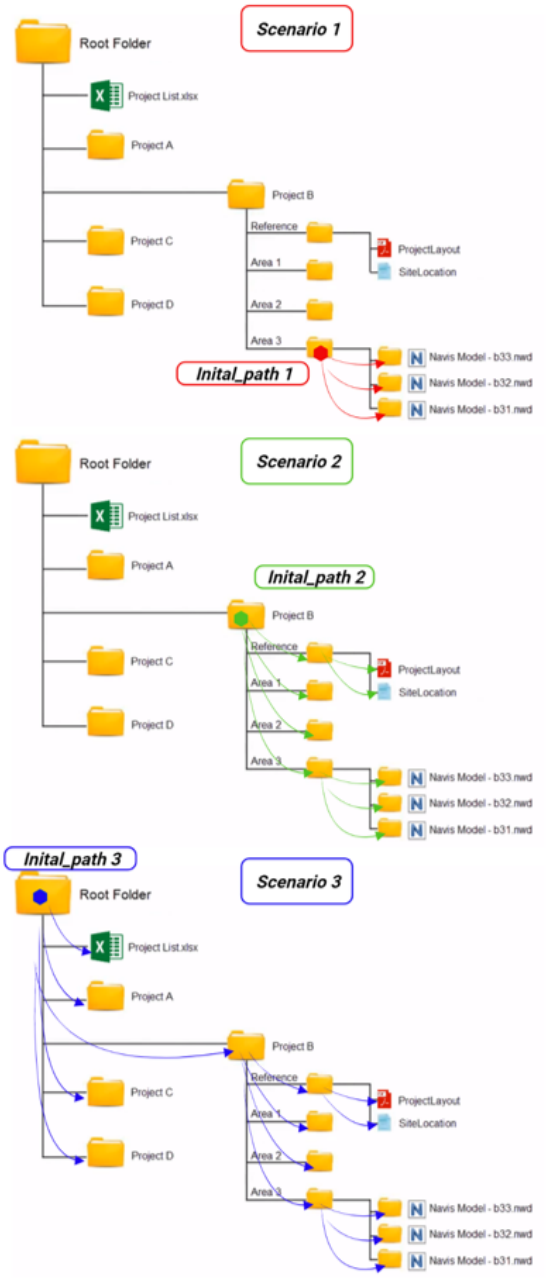

Figure 10: In this illustration, three scenarios can be seen. In each scenario the `initial_path` *as used in figure 6 is set to different folders. The colored arrows is the exact way how the pipeline searches the folders for MR files.*

Q5) Can the pipeline process other sequences as well?

For now, the pipeline can only parse T2w, DTI, and fMRI sequences. This is done by using the *sequence types* extracted from the header information of each measurement. In this pipeline, they are defined as: `['Dti*','EPI','RARE']` which are only relating to the sequence type used for the measurement and the names are predefined default names of Bruker's ParaVision Software. So it might be possible that other kinds of measurements use these sequences as well but are not compatible to be used with this pipeline[2]. Another problem that might accrue is that if one of the permitted sequences (T2w, DTI, fMRI) is measured with multiple echo times, repetition times, separate receiver channels, repetitions for averaging purposes, or any kind of additional dimension except Slices, Time and Diffusion directions, the pipeline will not work. It is planned in the near future to add more features.

Q6) How are the SNR of the T2w and DTI sequences calculated?
The only difference that we have in the DTI scans compared to the T2w scans, is the dimension of the diffusion directions. Here a small portion of the images in the diffusion direction is used to calculate the SNR similar to the T2w approach.

Q7) Why is the pipeline divided into separate stages?

The reason behind this is to increase the stability of the program and to prevent the need to run the pipeline multiple times from the beginning. As explained above, separate excel tables are created in each stage. Stage (I) is relatively fast and it won't cause any long waiting periods to rerun that part if necessary. Stage(II) on the other hand can take more time, sometimes up to hours to finish, therefore it is possible that if an error accrues in the plotting part of the code, to simply correct the error and to read the excel file created from the second stage and to do the plotting without the need to wait again for the whole pipeline to run again from the beginning.

Q8) Why are some of the SNR values empty in the excel table?

This can happen when a value with infinite SNR is calculated because of a division by zero which happens when the noise that is calculated

[2]For example Arterial Spin Labeling (ASL) sequences, DCE and DSC sequences, etc.

by the Chang method is equal to zero. Experience shows that usually these files are images with good quality and do not need to be discarded. However, it's best to also check them manually with an image viewer to see if they are not damaged in any way. It is planned to resolve this issue in a logical way in the future.

Q9) What is meant by the warning: Some faulty files were found: All faulty files are available in the Errorlist tab in the Excel outputs ?

With this warning, the pipeline just informs the user that some key files of the main image file could not be read. These can be one of the `method, viso_parameters, ...` files from the Bruker sequence folder. However, all these files are also listed and saved in a separate tab in the Excel output.

Q9) After running the pipeline, where can I find the data that I should not use?

In the final Excel sheet named `QuiC_Data_Result_Processed_featurs.xlsx` a tab can be found named `"Final Result"` where all of the unusable data, whether from faulty data or from bad quality features are listed with the reason of failure and some other useful information to compare it with.

Author:

I have designed this pipeline to help me validate all the MR data that I use for further processing in my project. I was searching for a kind of standardization to dichotomize MR data into good and bad data. Over time I found out that this can't be done as easily as thought. The key point of this standardization tool is that one realizes good and bad can not be set with fixed global values of any kind of features like the SNR and as everything in life is, it should be looked at *relatively*. If you have any questions regarding this pipeline, feel free to contact me at:

aref.kalantari-sarcheshmeh@uk-koeln.de

The end