1 Introduction

It is possible to access the AIDAmri pipeline via Docker. Note, that the AIDAmri GUI is not supported and only the CLI is accessible. The general CLI usage is described in the manual (click here). Before installing the Docker image, it is necessary to download and install the Docker application (click here). It is highly recommended to read the Docker documentation beforehand to understand the usage.
In this manual, some instructions will require custom entries like local directory paths. In those cases the paths or file names are written in CAPSLOCK and should be substituted by appropriate terms.

2 Installation

After setting up Docker and cloning the AIDAmri git or the `bin/` and `lib/` folders like in the manual, open a terminal and switch to the AIDAmri directory or the directory that contains said folders, respectively:

```
cd PATH/TO/AIDAmri
```

Here, the following command should be run. Be aware that the installation process takes a few minutes, requires a stable internet connection and approximately 12 GB of memory.

```
docker build -t aidamri:latest -f Dockerfile .
```

Note that the dot at the end is part of the command and refers to the current directory. The build command creates a so-called image, that is a template for a usable interface, the container. Once built, containers can be launched and stopped at will. The `aidamri:latest` is the ID and the tag of the image, respectively and has to follow the `-t` command. It is recommended to use the proposed ID and tag names. The `-f` flag invokes the Docker file (here: `Dockerfile`) for building the image.

3 Running Container and Mounting Data

After the image is built, a container can be run:

```
docker run \
-dit \
--rm \
--name aidamri \
--mount type=bind,source=PATH/TO/DATAFOLDER,\
target=/aida/DATAFOLDER \
aidamri:dev
```

Note that the backslashes are line breaks in Unix shells (Linux and macOS by default). PowerShell, i.e. the Windows shell, uses backticks (‘) instead, so if you use Windows, substitute the backslashes by backticks. Note further that there is no space between the comma after the `PATH/TO/DATAFOLDER` and the backslash as the arguments for the `--mount` flag must be written together and without any spaces. Docker run creates a usable interface, the container. It should be run in an interactive mode and will be detached from the current terminal, hence the `-dit` flag (`-d` flag and `it` flag combined, for detached and interactive mode, respectively). The `--rm` flag deletes the container after it is stopped to avoid unnecessary memory stacking. The container is named via the `--name` flag, in this case it is called `aidamri`. As separate containers could be run, please note that additional containers have to be named differently. The `--mount` flag is important to include the source data for processing. It mounts the data onto the container by reference, hence it does not occupy additional memory. The source directory has to be passed to the `source` tag (use absolute path) and it is then put into the `target` directory inside the container. Attach the name of your data folder to the `/aida/` path, thus the data can be found at this path within the container. Lastly, put ID and tag of your built image (`aidamri:dev`).

4 Usage

Here the usage of the container is described. Keep in mind that this is only one method to use the container. There are further methods like using volumes instead of binding mounts, so feel free to get comfortable with Docker structure and usage by using the Docker documentations.
After the container is up and running you can enter the container environment by typing the command:

```
docker attach aidamri
```

This will open an isolated command line interface, with the working directory being the `/aida/` directory. Typing in the `ls` command should give you the following output:

```
NiftyReg DATAFOLDER bin dsi_studio_ubuntu_1804 fslinstaller_mod.py
lib requirements.txt
```

From here on, the pipeline can be used as described in the manual. After processing close the environment by entering CRTL+P and CRTL+Q. Do not exit the container by typing in exit as this will stop and remove the container and the the processed data will be lost. You can enter the environment again by typing in the attach command.

To extract the data from the environment back to your host sytem, enter the following command:

```
docker cp aidamri:/aida/DATAFOLDER /PATH/TO/TARGETDIR
```

The TARGETDIR will be created if not existing already and save the data from the DATAFOLDER in the target directory. This may overwrite directories and files with the same names, so be careful with namespaces. Note, that the container aidamri has to be given before the colon.