



HAL
open science

MP3: Medical software for Processing multi-Parametric images Pipelines

Clément Brossard, Olivier Montigon, Fabien Boux, Aurélien Delphin, Thomas Christen, Emmanuel L Barbier, Benjamin Lemasson

► **To cite this version:**

Clément Brossard, Olivier Montigon, Fabien Boux, Aurélien Delphin, Thomas Christen, et al.. MP3: Medical software for Processing multi-Parametric images Pipelines. 2020. hal-02937866v1

HAL Id: hal-02937866

<https://hal.science/hal-02937866v1>

Preprint submitted on 14 Sep 2020 (v1), last revised 16 Nov 2020 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MP3: Medical software for Processing multi-Parametric images Pipelines

Clément Brossard^{1,2}, Olivier Montigon¹, Fabien Boux^{1,3},
Aurélien Delphin¹, Thomas Christen¹, Emmanuel L. Barbier¹,
and Benjamin Lemasson^{1,2,*}

September 2020

¹ Univ. Grenoble Alpes, Inserm, U1216, Grenoble Institut Neurosciences, Grenoble, France

² MoGlimaging network, HTE program of the French Cancer Plan

³ Univ. Grenoble Alpes, Inria, CNRS, G-INP, Grenoble, France

* Corresponding author: Benjamin Lemasson - benjamin.lemasson@univ-grenoble-alpes.fr

Abstract

This article presents an open source software able to convert, display and process medical images. It differentiates itself from the existing software by its ability to design complex processing pipelines and to wisely execute them on a large database. An MP3 pipeline can contain unlimited homemade or ready-made processes and can be carried out with a parallel execution system. As a viewer, MP3 allows to display up to 4 images together and to draw Regions Of Interest (ROI). Two applications showing the strengths of the software are here exposed: a preclinical study involving Magnetic Resonance Imaging (MRI) data and a clinical one involving Computed Tomography (CT) images. MP3 is downloadable at <https://github.com/nifm-gin/MP3>.

Keywords

MP3 ; Medical Images ; Software ; Pipelines ; Database ; MRI ; CT

1 Introduction

Researchers in medical imaging now have access to large amounts of data via open source databases (The Cancer Imaging Archive (TCIA) [1], Center Trau-

matic Brain Injury (CTBI) [2]). In parallel of the quantity of images available, the complexity of the medical images post-processing is increasing. Where data scientists applied a single process to their images to obtain valuable results [3], they have now to design and execute very complex interdependent processes called *pipelines* involving the execution of many operations referred as *modules* on large databases [4]. Among the modules used (e.g. bias removal, brain extraction or image registration) some of them are part of toolboxes well recognized by the community (such as SPM [5] or FSL [6] for the neuroimaging community) whereas others are home-made. How to reconcile flexibility, adaptive, speed, performance and reproducibility of a post-processing? Several software have been recently developed to solve this issue. One can divide them in two classes: specific and generic software. Specific one, are built to process a specific type of data, for instance a modality (e.g. Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), ...), an organ imaged (e.g. liver, brain, ...), or even a format (e.g. Digital imaging and communications in medicine (DICOM), Neuroimaging Informatics Technology Initiative (NIFTI), ...) using a predefined pipeline. On the other hand, generic software, aimed to apply different processes on several types of data. Most of the available software can be gathered in the first category. This includes BrainCAT [7] which generates diffusion tensor images (DTI) from MRI diffusion weighted images, MRtrix [8], which allows to visualize and specifically process diffusion MRI data, Pydipper [9] which offers registration algorithms, or FuNP [10] which gathers some of the most famous functions available for MRI data in a single post-processing pipeline. In those software, either the type of input data or the processing applied is unalterable. The second category contains software that can apply many different processes on a large type of data. To our knowledge, only few software available in free access belong to this category. One can quote Vaa3D [11], able to handle large dataset of images converted in Tagged Image File Format (TIFF) in a few seconds, to display them in 3D, and to define and execute homemade pipelines. One can also refer to GraphMIC [12], which provides a node based interface for famous image processing libraries and allows to link the modules of these libraries to create complex image processing pipelines without programming, simply by connecting existing modules. This powerful software makes the design of complex pipeline user-friendly as well as the integration of home-made modules. Those powerful software lack of a functionalities that we believe will become essential to develop new imaging biomarkers in order to handle and process large heterogeneous databases. Instead of processing one file or subject at a time, one will need a tool able to handle large cohorts, containing heterogeneous (multi-format) and multi-parametric data, and to process them with specific complex pipelines and with respect to the intra-subject time dependencies. To answer these limitations, we propose an open source software called "*Medical software for Processing multi-Parametric images Pipelines (MP3)*" [13]. This software therefore aims to facilitate the design of complex image analysis pipelines using existing or home-made modules as well as their execution on large heterogeneous databases.

2 Method

2.1 Overview

MP3, a MATLAB [14] toolbox, intends to assist an end-to-end research study, from the loading and the converting of raw images to the statistical analysis through the creation of a database containing metadata and the design and execution of complex analysis pipelines. MP3 is composed of three linked graphical user interfaces (GUI) that stands as its backbone: the converter, the viewer and the pipeline manager (Figure 1A). Briefly, those GUI allow to convert, display and process different medical image formats and architectures (Bruker, DICOM, PAR/REC, NIFTI, BIDS[15]). The imported data is summarized in a database able to be homogenized, filtered, or improved with metadata such as the name of the patient or the day of the acquisition. The viewer allows to display up to four 5D images simultaneously and to draw ROI. Eventually, a graphical interface called Pipeline Manager allows to create, manage, and execute complex pipelines thanks to editable modules. A pipeline execution system named PSOM (v2.2.2) [16] allows to judiciously parallelize the execution of the modules. In this section, we detail the aims and functionalities specific to each of these GUI and present the concept of a MP3 project and its architecture. Details information as well as videos presenting latest developments of MP3 are available online (Github: <https://github.com/nifm-gin/MP3>, Youtube: <https://www.youtube.com/playlist?list=PL-Tj6Wc9aE9x7i6s-RLetvNE0isnEsFm7>).

2.2 Data import and conversion

MP3 is built on the NIFTI format, which contains the imaged volumes as a matrix (up to 7 dimensions), and a short header that stores information about the image as its type or its size. This header also contains a transform matrix that describes the position of the volume in a conventional space which is very useful to co-register two NIFTI volumes without modifying the image. In order to store metadata of each image acquisition that may be essential for some post-processing (echo time, repetition time. . .), a consortium led by France Life Imaging defined conventional metadata and a way to write them in a json file [17]. A pair of NIFTI/json files can then describe entirely a medical image. To allow the use of classical formats of medical images, we developed a converter able to transform Bruker, PAR/REC, DICOM, and NIFTI files to a NIFTI plus json format. It is also possible to import data organized in the Brain Imaging Data Structure (BIDS) convention. Each user can also customize the metadata to be collected from the raw data (Bruker, DICOM ...) that will be stored in the json file by editing the provided yaml file. Full details about the customization of the image conversion is available online (https://populse.github.io/mri_conv/Home/index.html). The converter interface (Figure 1B) can be launched

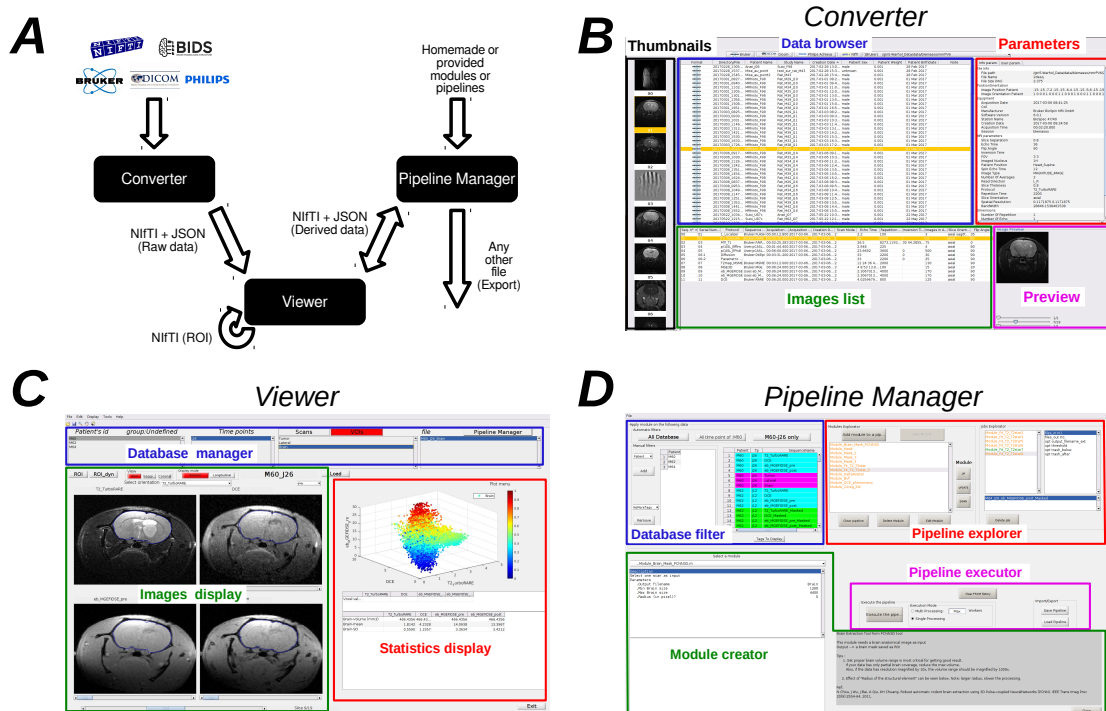


Figure 1: (A): Architecture overview and data flow, (B): Converter, (C): Viewer, (D): Pipeline Manager.

from the main MP3 window. To start a new project, one defines a folder to store the project. Then, thanks to the different parts of the converter (data browser, parameters, images list, etc...), one can easily navigate through complex and often unreadable medical image architecture. When the desired images are found, they will be automatically converted to NiftTi/json and metadata such as timepoints and subjects names may be modified. Then, back to the viewer GUI, a database has been created and will be used to manage your project data (Video 1: <https://www.youtube.com/watch?v=ebofxMSquFs&list=PL-Tj6Wc9aE9x7i6s-RLetvNE0isnEsFm7>).

2.3 Project Architecture

On the hard drive, the main directory of each project, sorted in multiple sub-folders, contains all the data needed to open the project via MP3. We defined several types of files, among which *Scans*, classical medical images, described

as a NIfTI and a json file, and *ROI*, described as a NIfTI file (containing a binary matrix). Each imported scan file is stored in the "Raw_data" sub-folder, while the processed files, written after a pipeline execution, are saved in the "Derived_data" sub-folder and the ROI in "ROI.data". Other sub-folders can be part of your project, such as for instance "Tmp", where lies temporary files, or "Saved_Pipelines", that stores the .mat files that summarize your designed pipelines (see the Pipeline Manager section). To manage these sub-folders, files, and associated metadata, MP3 relies on a database containing, for each entry, several information referred as *tags*, as its Type-Tag (Scan or ROI), Subject-Tag, Timepoint-Tag or Group-Tag associated, as well as its Path-Tag or Filename-Tag. Since MP3 is developed on MATLAB, we decided not to use a real database, as we could have developed in SQL language, but to use a MATLAB *table* object. The power of these variables relies on their ability to be quickly and easily filtered, which is a key operation in MP3. Each entry is linked to the corresponding files through this table. It is therefore this database that undergoes operations as renaming, sorting, or filtering to select part of our available data or to homogenize the database to make it more consistent. Since a project is completely described by its database, MP3 offers the possibility of easily transfer a whole project to another user. To save space and to easily share any project, MP3 handles the compressed NIfTI format *.nii.gz*.

2.4 Viewer

The Viewer is a MATLAB GUI that is divided into 3 parts (Figure 1C): a *database manager*, to display and manage the database, the *image display*, that displays up to four 5D images from the database and supports the drawing of ROIs, and the *statistics display*, which allows to display scatter plots, curves, histograms, or first order statistics, like mean or standard deviation within a ROI (Video 2: <https://www.youtube.com/watch?v=X26RV7VmXTA&list=PL-Tj6Wc9aE9x7i6s-RLetvNE0isnEsFm7>).

2.4.1 Database manager

The database (section Project Architecture) is presented as four listboxes each displaying a tag (Subject-Tag, Timepoint-Tag, Sequence_Name-Tag and Filename-Tag) and working as filters that reduce the database. Thereby, when the Subject-Tag list displays all the different subjects of the database, selecting one of them filters the database and reduces the files displayed in the other lists. This way of presenting the database is inspired by the BIDS architecture and is a simple and efficient way to quickly access a specific scan. One can switch from the Scans list to the ROIs list by just a click on the specific button above the Sequence_Name-Tag list. These lists allow to rename the value of a tag, copy or delete an entry in the database thanks to specific right-click menus associated to each listbox. To compute these operations on a large amount of entries, a sub-menu of the Edit menu offers some "Delete from database" or

”Rename from database” features.

2.4.2 Image display

One can load and display up to four 5D scans simultaneously. One of the main advantages of the NIfTI files is their transform matrix stored in the header (section Data import and conversion). This matrix locates the center of the scanner and its geometry. It is therefore possible, thanks to a light interpolation provided by functions of the SPM toolbox [5], to display different orientations and different resolutions. This ability is used to open the selected scans in a selected referential. One can change the selected referential thanks to a popup menu displayed on the top of the figures. Open different scans in the same referential is particularly useful when displaying two scans acquired with a different field of view or a different voxel size. Three push buttons above the images allow to change the referential orientation among axial, sagittal or coronal. On a displayed image, one can use some classic features such as zoom and pan the image or vary its contrast to see hidden pattern (right-click on the mouse). One can also set the same contrast to all displayed images, and then quantitatively compare several images values (middle-click on the mouse). The viewer also offers a graphical tool able to create contour to define ROI. This contour can be manually or automatically drawn thanks to an algorithm based on an active contour growth [18]. ROIs are automatically stored as NIfTI files and can then be displayed on any other scan, no matter its orientation or geometry.

2.4.3 Statistics display

This part of the viewer is aimed to display quantitative information about the scans or ROIs loaded. It incorporates a sub-window to display graphics and a table for numeric values. The first metric is the values of the voxels. When the mouse pointer flies over an image, the value of the current voxel on each loaded scan is displayed on the table, which gives a quick overview of the quantitative values of each image. Another MP3 feature is its ability to study the temporal evolution of a four dimensional scan. Indeed, click on a 4D voxel displays its values along the fourth dimension, which is particularly useful to study MRI perfusion or functional MRI. When a ROI is loaded, mean and standard deviation within the ROI for each loaded scan are displayed in a table, in addition to its volume in mm^3 . When only one scan is loaded, the figure above that table hosts an histogram and when several scans are displayed, this histogram is replaced by a scatter plot, which represents the pixel values of one scan versus the others. Since the MP3 project is open-source, one can easily display any simple feature or statistics by editing some of the main functions.

2.4.4 Longitudinal follow-up view mode

Another way to analyze images from the database is to switch the display mode of the viewer from *session* to *longitudinal*. This mode allows to compare a

Scan within the different time-points of a patient. One can then easily check a coregistration, and conduct volumetric analyses.

2.5 Pipeline Manager

Since the basic features (section Statistics display) are not enough to process complex analyses and are limited to the analysis of one subject we developed a third graphical user interface called the *Pipeline Manager* (Figure 1D). The pipeline manager allows to create, edit, save and execute complex pipelines on any file of a project. Moreover, it integrates fast and reproducible computation processes (mutli-CPU) and history handling. This section exposes the philosophy of pipeline manager and the main functionalities of this GUI (Video 3: <https://www.youtube.com/watch?v=QOULzRsrzzg&list=PL-Tj6Wc9aE9x7i6s-RLetvNE0isnEsFm7>).

2.5.1 Principal of the pipeline manager

Thanks to our structured project - data conversion (section Data import and conversion) and database (section Project Architecture), we have at hand a strong tool to manipulate our data. The pipeline manager allows to iterate a given process over the whole database (or a sub-part of the database). For instance, we are able to create a process that computes an image B from one specific image A and to apply it on each image A of the database. This process is called *pipeline*, because it can be a simple computation or a complex sequence of independent steps.

2.5.2 Pipeline definition

As illustrated on Figure 2A, a pipeline is composed of one or several *modules*. A module is a function, more or less complex, that is designed to compute one or more NIfTI files thanks to one or more other NIfTI files, and some parameters. For instance, a module called *Module_Smooth* takes as input a Sequence_Name-Tag and 3 parameters (The dimension of the filter - 2D or 3D Gaussian, the size of the filter - the variance of the Gaussian, and the Extension string of the output files), and generates a smoothed image. A module shall contain a basic operation rather than a whole process, although nothing prevent to do it. A module applied to a database creates a certain number of *jobs*, i.e. a certain number of occurrences of that module. Therefore, since a pipeline is simply a sequence of modules linked or not together, applying a pipeline to a database creates a certain number of jobs for each modules, so potentially a large number of jobs.

2.5.3 Procedure to create a pipeline

Now that the concept of pipeline is introduced, we now expose the procedure to create one (Figure 2B). First, one needs to select the part of the data on which to execute the pipeline. Indeed, MP3 allows the user to apply a pipeline to the entire database or only on a sub-part of it (i.e. the 10 first subjects, or on a specific timepoint). This operation is managed by the "database filter" on the upper left corner (Figure 1D). This section contains manual or predefined filters as well as a table that shows the filtered database. After the data selection, one can go to the module creator (Figure 1D) and choose a module among all the ones currently available (section Provided modules). Selecting a module displays all its required parameters. For instance, selecting the *Module.Smooth* shows a line aimed to select a Sequence_Name-Tag value, another aimed to set a filter size, etc... After the modification of all parameters (which can also be left to their default value), one can add the module to the pipeline by clicking on the button *add module to a pipeline*. When one adds a module to a pipeline, the module automatically creates as much jobs as necessary (it depends of the database). Each output Scan or ROI of each job are then added to the filtered database displayed on the upper left. Since the pipeline has not been executed yet, all those files do not exist yet but are still accessible for the user in the displayed database. This offers the possibility to parametrize another module with the tags of those *virtual entries*, and therefore link some modules together. The operation of filtering the database, selecting a module, parametrizing it, and adding it to the pipeline, which creates jobs (Figure 2B), can be reproduced an unlimited number of times. The *Pipeline explorer*, on the upper right corner (Figure 1D), allows to explore any part of the pipeline, from the different modules to each related job, to each job parameter in the job. Any operation (editing a module, deleting a module or a job...) is entirely under the control of the user.

2.5.4 Saving and sharing

In order to share a pipeline between several users, computers or projects, MP3 offers the possibility to save and load pipelines. Since a pipeline is a list of modules, we just store in a .mat file the sequence of modules with their parameters, but we do not store the jobs, since they are specific to a database. Load a pipeline consists in applying each modules to the new database. If the tag values of the pipeline to load are not consistent with the database, a module cannot generate jobs. The module name is then displayed in red in the pipeline explorer and one just need to edit the affected module, select the adapted tag value and save it to make it green again. This color convention intends to make clear the content of a pipeline as well as the consequences of its execution. If some modules are displayed in orange, it means that some of their jobs (in orange too) have already been executed (during a previous execution of the pipeline). In that case, one can decide to overwrite these files or to delete the related jobs when executing the pipeline.

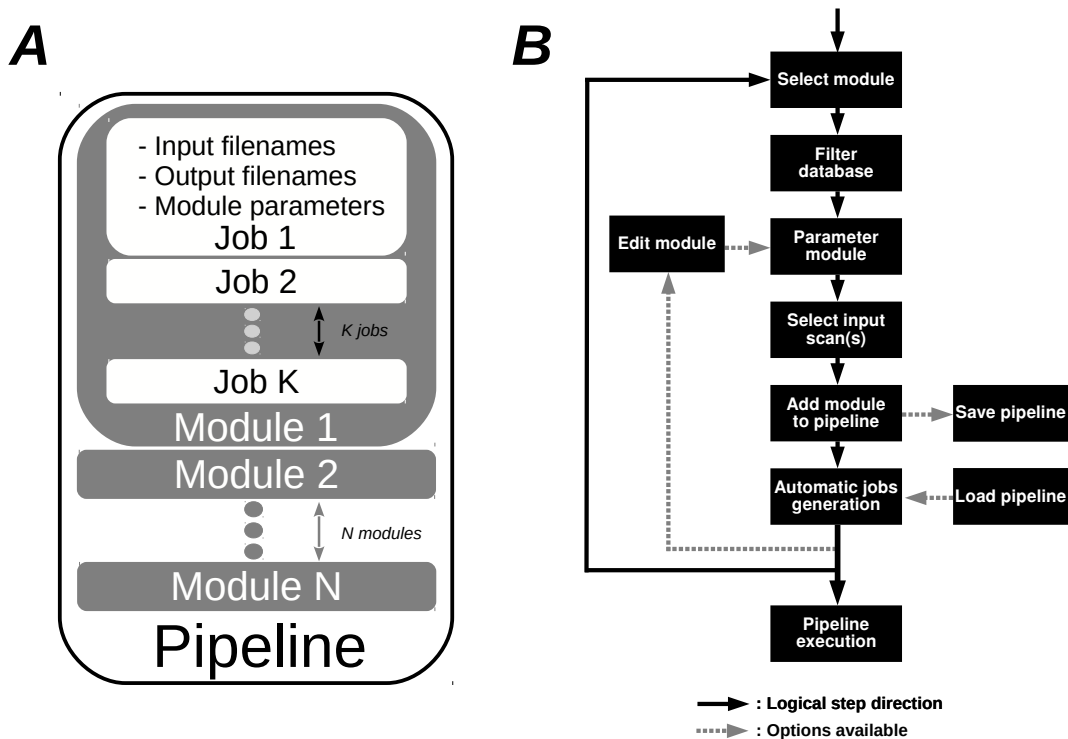


Figure 2: (A): Pipeline architecture, (B): Pipeline creation procedure.

2.5.5 Execution of a pipeline

Once a pipeline is well designed, there are two execution ways in MP3. The first way is the *Single Processing*. Each job is executed in the module order: all the jobs of the first module, then all the jobs of the second one, etc... This quick execution way, without strong dependencies between the modules and a low computing time, is especially useful when testing or developing a module. The second way is to use a pipeline execution system called "PSOM" (v2.2.2) [16] by selecting the option *Multi Processing*. This powerful system allows to execute a pipeline on several cores or upon a cluster. The dependencies between jobs are taken into account and the jobs are then distributed upon each core. This system also offers a garbage collector in order to save Random Access Memory and a way to monitor the number of jobs launched in parallel. As it has an influence on the behavior of its computer during the execution of the pipeline, one can set this number of *Workers* before the execution. During the execution,

all output files are written in a temporary folder. At the end, each output file of a successful job is saved in its data folder according to its type (Derived_data or ROI_data). In order to help navigating in the database (cf. *Database filter* section), we defined a color code based on each scan type. The blue represents raw data, derived data are in green, while the pink means ROI data, and the yellow colorizes the virtual files (output files of not executed pipeline yet).

2.6 Modules

The power of the pipeline manager lies in its modules. They are the basic operations needed to design complex pipelines. We adapted the way to define modules exposed in PSOM (v2.2.2) [16]. A module is then a MATLAB function, stored in a .m file whose name begin by the string *Module_*. All those functions lie in a folder called *Modules* of the MP3 source code and are sorted by area, with all the modules of an area in a folder. On launch, the pipeline manager reads the *Module* folder and its sub folders to list every available module. Thereby, add or delete a file in the subfolders of this repository updates the module list during the launching of the pipeline manager.

2.6.1 Provided modules

MP3 is provided along with 12 modules, performing basic operations, as smooth, threshold, or mask an image with a ROI, compute a brain extraction module [19], export a .csv file containing first order statistics or an HTML report of images and ROI. We also provide modules used in the preclinical pipeline exposed section Preclinical application, which are MRI oriented, and a module able to delete files of the project's database. Finally, a module interfacing the famous toolbox [5] allows to reslice images to match the referential of a reference image.

2.6.2 Template

Other users must be able to create their own modules. To facilitate the development of a new module, a module named *Module_Template* is available. It is extensively commented and explains how does a module work. One has then to slightly complete and adapt it to the wanted behaviour and to save the new .m file in the *Module* folder to rapidly apply a new module on a database.

2.6.3 Traceability

At each job execution, the module parameters and all the input/output file-names are stored in the json file associated with each output scan. Thereby, a scan obtained through the pipeline manager is linked to the raw data through the history of each module applied to the raw scan, and each module parameters. A basic GUI, called 'File History', launchable from the viewer when a scan is loaded, is able to display all of this history. For each scan, we can go back to

the past and know exactly how this scan was obtained and when each module has been executed.

2.6.4 Module repository

The homemade modules are aimed to gather the laboratory knowledge and know how. It is also a way to avoid loss of skill due to the departure of a team member. That is why we already wrote more than 60 modules that match our needs of image processing. All those modules are available at https://github.com/nifm-gin/MP3_User_Modules_Repository. One can find MRI oriented modules, interfaces to others toolboxes functions, as coregistration modules from [5] or [6], or more advanced processing, such as clustering of texture analysis, MR fingerprinting analysis, quantitative MR relaxometry, MR perfusion analysis (cerebral blood flow, cerebral blood volume, vessels size, vessels permeability,....)

3 Results

MP3 has been tested and validated on several MRI and CT data studies, but nothing prevents its use on any file that can be converted in NIFTI.

3.1 Preclinical application

A research study published in 2017 compared the blood-barrier permeability changes induced by synchrotron microbeam or uniform radiation therapy. 18 rats bearing intracranial tumors were treated and imaged by multi-modal MRI using the Grenoble MRI facility IRMaGE [20]. All procedures related to animal care conformed to the Guidelines of the French Government with licenses 380325 and 380321 (authorized lab A3818510002 and A3851610004). Part of these data are available online as an example of a MP3 project [21]. This project contains data of two rats among the 18 ones each imaged at three timepoints. Each timepoint contains 10 scans corresponding to the data described in [20], a ROI delineating the brain, and some masked scans obtained thanks to the former ROI. The data processing, composed of 16 modules, was designed as an MP3 pipeline and applied on the database of the study. As shown on Figure 3, it created several occurrences of the pipeline aimed to process the data of each timepoint of each subject in the same way.

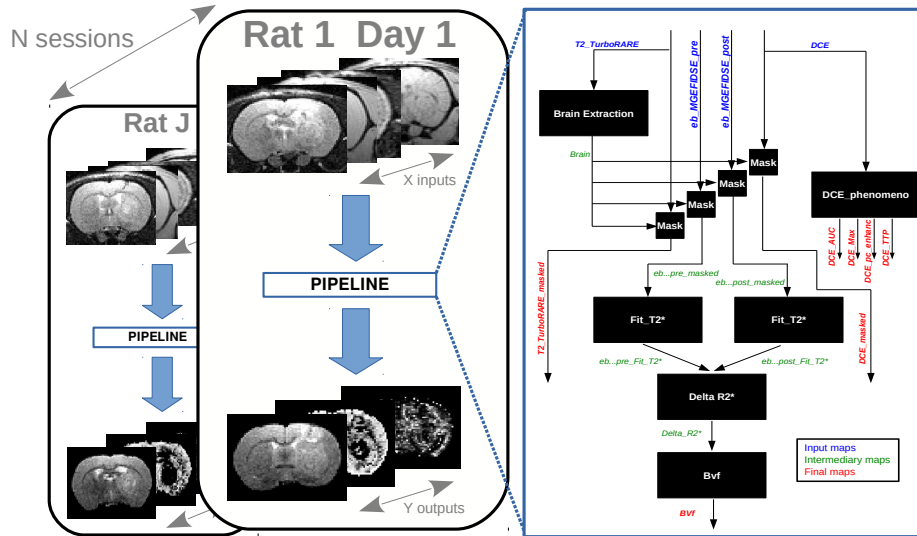


Figure 3: Preclinical study: Occurrences of a pipeline on a database.

3.2 Clinical application

As previously said, any image able to be converted in Bruker, DICOM, PAR/REC or NIfTI format can be processed in MP3, no matter its nature. For instance, MP3 is used in a non-published study aimed to predict the evolution of brain injuries following a traumatic brain injury (TBI) thanks to CT-scans acquired at the admission in the hospital, and at respectively one and three days after. A pre-processing pipeline for CT-scans, defined and described by [22] was integrated as part of our own post-processing pipeline. This process resizes, clips, filters and extracts the brain from CT-scans. For our study, we add a module of coregistration (between the timepoints of a patient) from the SPM toolbox [5] and a homemade module able to process a local entropy map, as well as some other texture images. All those modules are available at https://github.com/nifm-gin/MP3_User_Modules_Repository. The pipeline applied to a patient and its 3 timepoints is described on Figure 4A. This patient is part of a cohort whose data acquisition was allowed by the french institution "Comité de protection des personnes" and respects the patients written inform consent obligation. The CT-scans and entropy maps, respectively inputs and outputs of the pipeline, are displayed on Figure 4B. One can see that entropy scans are co-registered and skulls were well removed. Such pipeline may be easily apply to a large cohort of patient in order to follow the evolution of the brain entropy, used as a potential prognostic biomarker of neurological outcome.

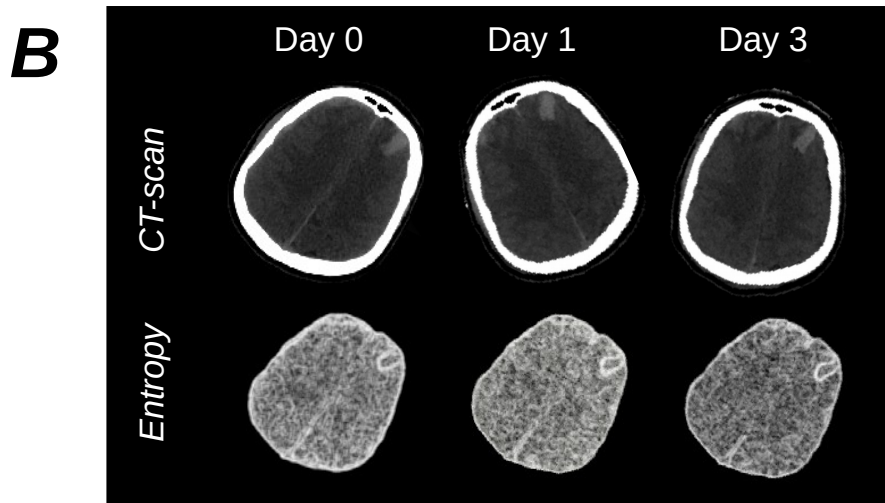
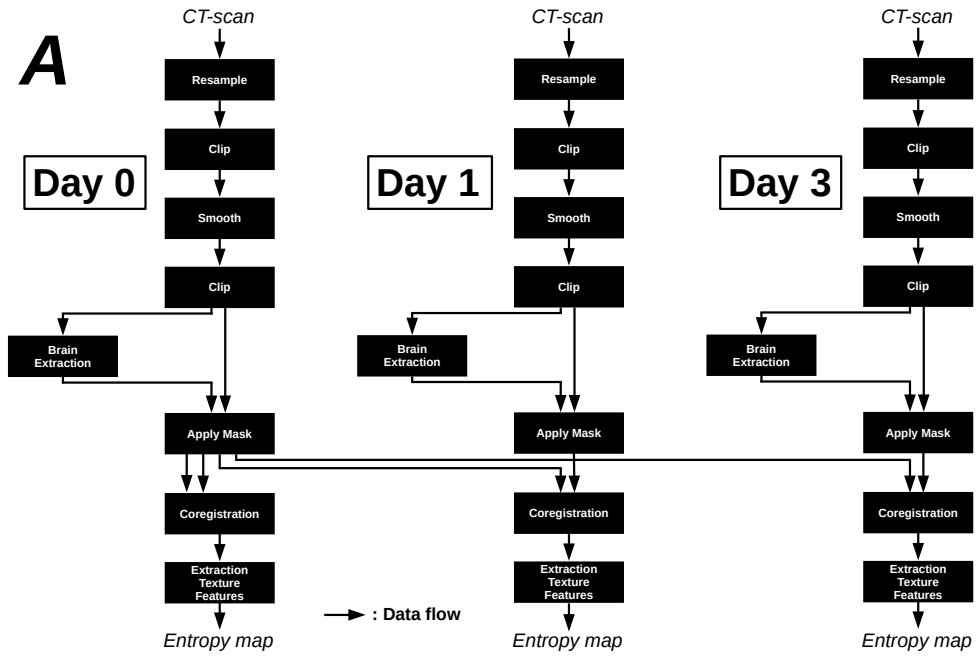


Figure 4: (A): Processing pipeline applied on the images acquired at day 0, day 1 and day 3, (B): Input maps: CT-scans, and output maps: Entropy maps.

4 Conclusion

This article exposed a new open source software able to support an end-to-end research study on a large amount of data. Thanks to 3 graphical interfaces, MP3 offers to convert and visualize medical images and to interact with them by comparing or analyzing them. Based on a pipeline execution system called PSOM, MP3 also offers to create, manage and execute complex pipelines on heterogeneous cohorts described in databases, which handle time dependencies and multiparametric data. MP3 can be used either by end-users (non-developers) or by developers which can improve it or develop their own modules. MP3 has been tested on several studies, from preclinical to clinical, from MRI to CT data, and allow us to process a preclinical cohort of more than 450 animals. MP3 can be downloaded on github: <https://github.com/nifm-gin/MP3> and modules developed by our lab (more than 60) are available on: https://github.com/nifm-gin/MP3_User_Modules_Repository.

5 Requirements

MP3 can be run on MATLAB 2017b and higher and need the Image Processing Toolbox. In order to fully enjoy the software, we recommend to get the Statistics and Machine Learning Toolbox and the Parallel Computing Toolbox. Since the converter GUI is developed in Java, Java 8 is also required. Any of the three main Operating System (OS) can handle this software. MP3 is open source, open development and available on github [13]. To develop new modules, one need to know MATLAB programming.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Funding

This work has been supported by the MoGImaging network funded by the Heterogeneity of Tumors and Ecosystem program within the framework of the french Cancer Plan.

Acknowledgments

Grenoble MRI facility IRMaGe was partly funded by the French program “Investissement d’Avenir” run by the ‘Agence Nationale pour la Recherche’ ; grant ‘Infrastructure d’avenir en Biologie Santé’ - ANR-11-INBS-0006.

Data Availability Statement

The software is available at <https://github.com/nifm-gin/MP3>. Example dataset and documentation can be found on the wiki page of the github repository. Video documentation is provided on the software repository wiki page and on youtube: <https://www.youtube.com/playlist?list=PL-Tj6Wc9aE9x7i6s-RLetvNE0isnEsFm7>.

References

- [1] Dataset. Welcome to The Cancer Imaging Archive, 2020.
- [2] Dataset. Home // CENTER-TBI.eu, 2020.
- [3] Benjamin Lemasson, Hanxiao Wang, Stefanie Galbán, Yinghua Li, Yuan Zhu, Kevin A. Heist, Christina Tsein, Thomas L. Chenevert, Alnawaz Rehemtulla, Craig J. Galbán, Eric C. Holland, and Brian D. Ross. Evaluation of Concurrent Radiation, Temozolomide and ABT-888 Treatment Followed by Maintenance Therapy with Temozolomide and ABT-888 in a Genetically Engineered Glioblastoma Mouse Model. *Neoplasia*, 18(2):82–89, February 2016.
- [4] Thomas Funck, Kevin Larcher, Paule-Joanne Toussaint, Alan C. Evans, and Alexander Thiel. APPIAN: Automated Pipeline for PET Image Analysis. *Front. Neuroinform.*, 12, 2018.
- [5] Karl Friston, John Ashburner, Jon Heather, Andrew Holmes, and Jean-Baptiste Poline. SPM - Statistical Parametric Mapping, 2020.
- [6] Mark Jenkinson, Christian F. Beckmann, Timothy E. J. Behrens, Mark W. Woolrich, and Stephen M. Smith. FSL. *NeuroImage*, 62(2):782–790, August 2012.
- [7] Paulo César Gonçalves Marques, Jose Miguel Soares, Victor Alves, and Nuno Sousa. BrainCAT - a tool for automated and combined functional Magnetic Resonance Imaging and Diffusion Tensor Imaging brain connectivity analysis. *Front. Hum. Neurosci.*, 7, 2013.
- [8] J-Donald Tournier, Robert Smith, David Raffelt, Rami Tabbara, Thijs Dhollander, Maximilian Pietsch, Daan Christiaens, Ben Jeurissen, Chun-Hung Yeh, and Alan Connelly. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202:116137, November 2019.
- [9] Miriam Friedel, Matthijs C. van Eede, Jon Pipitone, M. Mallar Chakravarty, and Jason P. Lerch. Pydpiper: a flexible toolkit for constructing novel registration pipelines. *Front. Neuroinform.*, 8, 2014.

- [10] Bo-yong Park, Kyoungseob Byeon, and Hyunjin Park. FuNP (Fusion of Neuroimaging Preprocessing) Pipelines: A Fully Automated Preprocessing Software for Functional Magnetic Resonance Imaging. *Front. Neuroinform.*, 13, 2019.
- [11] Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H. Simpson, and Eugene W. Myers. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat Biotechnol*, 28(4):348–353, April 2010.
- [12] Alexander Zehner, Alexander Eduard Szalo, and Christoph Palm. GraphMIC: Easy Prototyping of Medical Image Computing Applications, 2015.
- [13] Software. nifm-gin/MP3, December 2019. original-date: 2019-03-05T10:35:38Z.
- [14] Software. *MATLAB version 9.3.0.713579 (R2017b)*. The Mathworks, Inc., Natick, Massachusetts, 2017.
- [15] Krzysztof J. Gorgolewski, Tibor Auer, Vince D. Calhoun, R. Cameron Craddock, Samir Das, Eugene P. Duff, Guillaume Flandin, Satrajit S. Ghosh, Tristan Glatard, Yaroslav O. Halchenko, Daniel A. Handwerker, Michael Hanke, David Keator, Xiangrui Li, Zachary Michael, Camille Maumet, B. Nolan Nichols, Thomas E. Nichols, John Pellman, Jean-Baptiste Poline, Ariel Rokem, Gunnar Schaefer, Vanessa Sochat, William Triplett, Jessica A. Turner, Gaël Varoquaux, and Russell A. Poldrack. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci Data*, 3(1):1–9, June 2016.
- [16] Pierre Bellec, Sébastien Lavoie-Courchesne, Phil Dickinson, Jason Lerch, Alex Zijdenbos, and Alan C. Evans. The pipeline system for Octave and Matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows. *Front. Neuroinform.*, 6, 2012.
- [17] M Kain, M Bodin, S Loury, Y Chi, J Louis, M Simon, J Lamy, C Barillot, and M Dojat. SHANOIR – Sharing NeuroImaging Resources, 2020.
- [18] Li Wang, Lei He, Arabinda Mishra, and Chunming Li. Active contours driven by local Gaussian distribution fitting energy. *Signal Processing*, 89(12):2435–2447, December 2009.
- [19] Nigel Chou, Jiarong Wu, Jordan Bai Bingren, Anqi Qiu, and Kai-Hsiang Chuang. Robust automatic rodent brain extraction using 3-D pulse-coupled neural networks (PCNN). *IEEE Trans Image Process*, 20(9):2554–2564, September 2011.
- [20] Audrey Bouchet, Marine Potez, Nicolas Coquery, Claire Rome, Benjamin Lemasson, Elke Bräuer-Krisch, Chantal Rémy, Jean Laissue, Emmanuel L. Barbier, Valentin Djonov, and Raphael Serduc. Permeability of Brain

Tumor Vessels Induced by Uniform or Spatially Microfractionated Synchrotron Radiation Therapies. *International Journal of Radiation Oncology • Biology • Physics*, 98(5):1174–1182, August 2017.

- [21] Dataset. nifm-gin/Data_test_mp3, July 2019. original-date: 2019-06-28T14:46:06Z.
- [22] John Muschelli, Natalie L. Ullman, W. Andrew Mould, Paul Vespa, Daniel F. Hanley, and Ciprian M. Crainiceanu. Validated automatic brain extraction of head CT images. *NeuroImage*, 114:379–385, July 2015.