EUCAIM
CANCER IMAGE EUROPE

Project title: European Federation for Cancer Images
Project acronym: EUCAIM
Grant Agreement: 101100633
Call identifier: DIGITAL-2022-CLOUD-AI-02

D5.4: Data Preprocessing Tools and Services

Responsible partner(s): Quibim

Author(s): Celia Martín (Quibim), Jose Munuera (Quibim), Alejandro Vergara (Quibim), Xavier Rafael (Quibim), Laure Saint-Aubert (MEDEX, group member of BC Platforms), David Rodríguez (CSIC-IFCA), Valia Kalokyri (FORTH), Stelios Sfakianakis (FORTH), Katerina Dovrou (FORTH), Katerina Nikiforaki (FORTH), Ioannis Karatzanis (FORTH), Nikolaos Tachos (FORTH), Dimitrios Zaridis (FORTH), Vasileios Pezoulas (FORTH), Olga Giraldo (DKFZ), Wahyu Wijaya Hadiwikarta (DKFZ), Mirna El Gosh (LIMICS), Carina Soler (HULAFE), Pedro Miguel Martínez (HULAFE), Andrián Galiana (HULAFE), Mario Verdicchio (SYNLAB), Marco Aiello (SYNLAB), Diego Silveira (ITI), Silvia Ruiz (ITI), Christian Salvatore (DeepTrace), Elia Schiavon (DeepTrace), Alexandra Kosvyra (AUTH), Dimitris Fotopoulos (AUTH), Dimitris Filos (AUTH), Ioanna Chouvarda (AUTH), Ioannis Iakovou (AUTH), Stefanos Finitsis (AUTH)

Reviewer (s): Alberto Labarga (BSC), Carles Hernández-Ferrer (BSC), Ana Jimenez (Quibim), Irene Marín (HULAFE), María Gonzalez (SAS), Pedro Malloll (HULAFE), Gianna Tsakou (MAG)

Date of delivery: 28 June 2024
Version: 1.0

Link to demonstrable: https://www.youtube.com/watch?v=prcyL7hmUYc

Table of contents

A. Introduction

Aim and scope of the deliverable

This report completes the information presented in the demonstration video and jointly contributes to Deliverable D5.4: Data Preprocessing Tools and Services. The EUropean Federation for CAncer IMaging (EUCAIM) preprocessing tools and services support the data preprocessing pipeline, which supports data from the infrastructure to become GDPR compliant, annotated, harmonised, quality checked and ready for sharing within the EUCAIM repository for their subsequent reuse in data-driven applications. Additionally, it supports data compliance with the FAIR (Findable, Accessible, Interoperable, Reusable) principles. In a federated infrastructure, data originates from multiple heterogeneous sources with different formats, standards, and potential inconsistencies. Effective and consistent preprocessing minimises these disparities. Well-preprocessed data forms a solid foundation for robust, efficient, and insightful data-driven applications such as the development and validation of Artificial Intelligence (AI) algorithms.

The EUCAIM project originates from five Artificial Intelligence for Health Imaging (AI4HI) projects: ProCancer-I, ChAImeleon, EUCANImage, Incisive, and Primage. Many tools presented in this document were developed under the umbrella of these projects. In contrast, others are being developed within EUCAIM to fulfil some functional requirements of the data preparation process. At the moment, over 40 tools are part of the EUCAIM catalogue (Figure 1), from which 12 are shown in the demonstrator video accompanying this document. In the future, new tools that cover gaps in the functionalities or new use cases will be incorporated into the tools catalogue. To ensure that data processing respects data and infrastructure privacy, and data integrity, a validation process for tool incorporation is being defined together with WP6 and WP7.

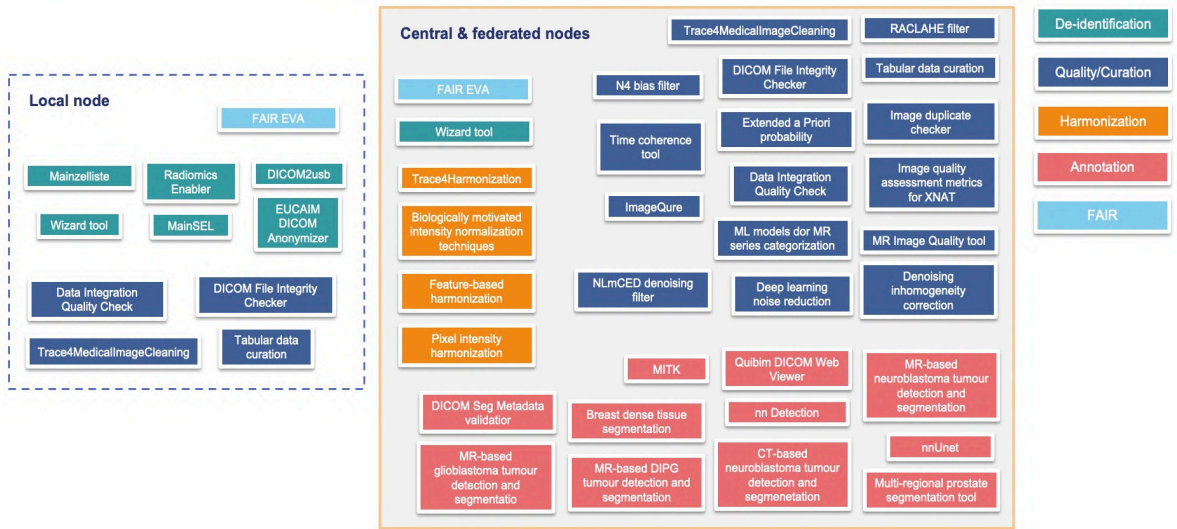


Figure 1. Overview of preprocessing tools in EUCAIM and their location within the EUCAIM infrastructure (local or central/federated nodes). The colour code indicates the task they belong to.

In this first deliverable of the preprocessing tools and services, we present a proof of concept of how data is curated for its further reuse in the platform. The demonstrator video shows three scenarios: the preprocessing of MRI Prostate T2 weighted (T2W) images, the

preprocessing of mammography images, and the evaluation of a FAIR data point. The first two scenarios show the whole pipeline of the data preparation process, from the anonymization and quality checks performed by the Data Holder (DH) to the data preparation prior to the AI model training/validation by the Data User/Researcher (DU/R) in the central node. This deliverable focuses on the preprocessing in the central node while in the next deliverables, we will move further towards preprocessing distributed data. Similarly, we follow a staged approach with the data type, focusing on this deliverable on imaging data while, in the following deliverables, we will move further to numerical data. This approach allows us to thoroughly validate and refine our tools for each type of data, ensuring robust and comprehensive solutions for the entire EUCAIM infrastructure.

This document aims to add context to understand the demonstration. It is structured as follows: in section A, an overview of the locations of the EUCAIM preprocessing tools is provided, along with a description of data preparation for DHs depending on the Tier level. In section B, the document delves into data preprocessing on-premises, focusing on de-identification, minimum quality assessment, and fairness assessment. Section C details the preprocessing of data in the central node, including further quality assessments, annotation tasks and harmonisation of data. Following this, in section D, the incorporation process of new tools to EUCAIM is described. Section E complements the demonstration video by introducing the test environment and presenting the results obtained. Finally, section F outlines future work.

Preprocessing tools workflow

Preprocessing tools are distributed across many locations within EUCAIM and cover several critical scenarios, assisting with the data preparation for different tasks and to different EUCAIM Stakeholders (Figure 2). In this section, we give an overview of the varied scenarios and describe the particularities of each one of them.

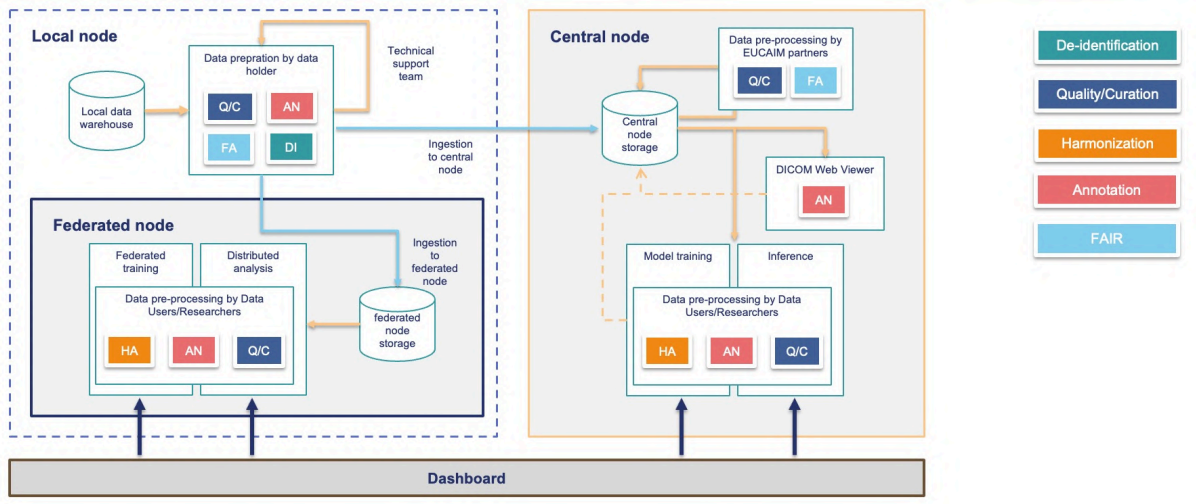


Figure 2. High-level workflow of the preprocessing tools within EUCAIM.

Data preparation by Data Holders in the local node

DHs can contribute with data either by (a) becoming a federated node or (b) transferring de-identified data directly to the Central Repository. Both scenarios require that the DHs perform some preparation of the data. The necessary preprocessing varies depending on the Tier level. For example, for Tier 1, DHs are only required to de-identify the data if they

transfer it to the central node or share it under the corresponding Data Sharing Agreement (DSA). On the other hand, the most demanding process is required to reach Tier 3 data. The data preprocessing steps followed by the DH, depending on the Tier, are described in the next section.

To help DHs prepare their data before its ingestion to EUCAIM, a set of downloadable tools is provided. Additionally, the Technical Support Team can assist DHs in this process.

Data preprocessing by EUCAIM partners
After data is ingested into the central node, it can be placed in a quarantine state, during which it is not visible to EUCAIM users. The quarantine state allows EUCAIM partners to conduct additional checks and validations on the data before it is published and made accessible. During this phase, a comprehensive set of tools is available for EUCAIM partners to ensure the integrity and quality of the data. These tools enable the assessment of critical aspects such as proper anonymization, data quality, and the fairness of datasets. By performing these checks, EUCAIM ensures that the data meets the necessary standards and is ready for use in various applications, thus maintaining the reliability and trustworthiness of the repository.

Central node viewer
On the one hand, the UPV central node features the **Quibim DICOM Web Viewer**. This viewer offers comprehensive image interaction capabilities, including zoom, pan, scroll, and window/level adjustments. It also supports the annotation process, with tools to extract ROI measurements or to generate segmentation masks. Annotations can be created manually from scratch or semi-automatically. In the latter case, AI tools can be executed from the viewer, producing annotations that clinicians can further refine and correct, thereby speeding up the annotation process. Finally, segmentation masks are stored in DICOM SEG format. The persistence of annotations is possible under specific conditions, leveraging the use of the central node viewer by expert radiologists to annotate cancer imaging data.

On the other hand, the Erasmus MC node uses the **XNAT OHIF Viewer**, which also includes general interaction capabilities. It supports the contouring of ROIs and ROI-level statistics extraction. Additionally, it provides tools for manual, semi-automatic, and automatic segmentation.

Data preprocessing by Data Users or Researchers
DU/R after being granted access, can utilise the data located in the central and/or federated node(s) to train or validate AI algorithms. Here, the preprocessing tools assist them in preparing the data, including harmonisation to reduce the bias introduced by different scanner models or acquisition protocols, filters that reduce common artefacts in images such as the bias field artefact, or automatic segmentation algorithms that can be used as a first step in an analysis pipeline. Researchers can access multiple state-of-the-art tools in the marketplace to prepare the data for training.

In the central node, it is possible to save the results of the preprocessing tools under certain conditions, enabling their further reuse. This functionality ensures that prepared datasets can be efficiently accessed for future projects, enhancing the overall utility and efficiency of the EUCAIM infrastructure.

Data preprocessing depending on the Data Tiers

Data preprocessing in healthcare requires special attention. EUCAIM deals with a considerable amount of diverse data from different repositories/sources. Therefore, it requires defining standards and structures for how the data is modelled and stored to avoid any misconception, allow data-driven tools to deal with the data and metadata consistently, and prevent any false negative/positive result.

To ensure a smooth incorporation of data and to facilitate the participation of new DHs, EUCAIM defines three Tiers. The Tiers have different data preprocessing requirements, and DHs are encouraged to curate their data to reach higher Tiers. This section outlines the Tiers (described thoroughly in D4.3), the preprocessing requirements for each of them, and the tools that EUCAIM provides to curate data within each data Tier (Figure 3).

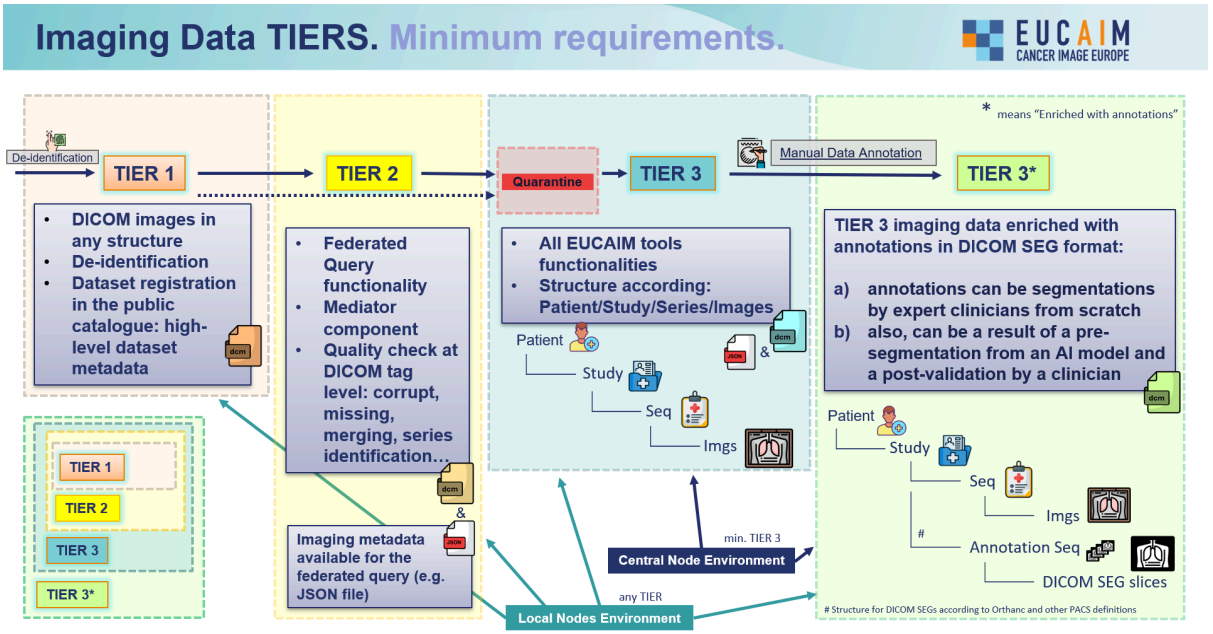


Figure 3. Minimum requirements for Imaging data within EUCAIM Data Tiers.

Tier 1

Data within Tier 1 is accepted by the Federation with no additional technical requirements in terms of compliance with the common data formats (EUCAIM's hyper-ontology), hence no federated/distributed processing capabilities nor a homogeneous framework for research is available. In other words, only the publication and visualisation of the dataset in the public metadata catalogue is possible. This dataset can be made accessible under the defined data request process.

In Tier 1, high-level dataset metadata is registered in the public catalogue of EUCAIM, and data preprocessing is not required if data is kept on the premises of DHs. In case the data is moved to the central node(s) or shared after the corresponding DSA, the data assigned to Tier 1 must adhere to some minimal requirements to facilitate its incorporation. The mandatory requirements for Tier 1 data focus on data de-identification to ensure that EUCAIM datasets do not contain personal information about patients and minimise the risks of patient re-identification. Imaging data must be in DICOM format. To ensure the correct anonymization process of imaging data, EUCAIM provides two tools to data holders:

- **EUCAIM DICOM Anonymizer**: The tool, developed by FORTH, performs a de-identification of imaging data DICOM tags. It takes as input a folder with one or multiple patients/cases, and de-identifies the DICOM tags according to a predefined de-identification profile which specifies how to process each of them.
- **Wizard tool**: This tool is under development during the project. Its goals are to support the identification of risks and propose ways to mitigate them, to raise awareness of the weak points of each process, to foster a secure-by-design anonymization planning, and to facilitate compliance with EUCAIM requirements and accountability obligations. Clinical data in Tier 1 is not compliant with the Common Data Model (CDM) thus the functionalities for the clinical data of the tool are deactivated.

More information about these tools and the process of de-identification can be found in section B. The anonymization of imaging data is straightforward since DICOM is the standard. However, the process is more complex with clinical data since in Tier 1 data does not comply with the CDM and is not standardised. Thus, DHs cannot take advantage of the **Wizard tool** for the clinical data and must ensure that no sensitive information is present in the dataset before providing it to the EUCAIM ecosystem. As a minimum data quality assessment, EUCAIM also provides DHs with a tool for the detection of corrupted files and the assessment of the correct number of files per sequence: the **DICOM File Integrity Checker tool**. This tool developed by HULAFE can, among other features, generate a report containing information about the selected sequences, corrupted files, missing files, etc. This tool, part of the data quality tool catalogue, is part of the demo video recording for D5.4 and is further described in section B. Additionally, Tier 1 data has to comply with a minimal set of Research Data Alliance[1] (RDA) FAIR indicators, particularly the ones related to the presence of globally unique identifiers for data and metadata, and the harvesting of metadata.

Tier 2

Data within Tier 2 is compliant with Tier 1 requirements and with EUCAIM's Federated Query service. This requires greater effort and involvement from a DHs perspective but allows for improved visibility and usability of the data. The successful execution of federated queries requires providing metadata according to EUCAIM's Hyper-ontology, and/or operating a local "mediator" service to execute the federated queries and report back the aggregated results.
The EUCAIM Federated Query component allows querying DICOM metadata, which plays a key role in medical imaging and AI-related applications.

Data within Tier 2, as is compliant with Tier 1 requirements, needs to be de-identified to ensure that the data does not contain personal information. The **EUCAIM Dicom Anonymizer and the Wizard tool**, presented in the previous subsection, are also provided to DHs to assist them with that task, in addition to the **DICOM File Integrity Checker tool** for the detection of missing or corrupted files. Further compliance with FAIR indicators is to be expected in Tier 2. On top of this, DHs should provide through the EUCAIM Hyper-ontology/metadata catalogue, information that allows users to localise datasets with data that would be relevant to their research questions using the attributes that are mandatory for the EUCAIM metadata catalogue.

[1]DOI: 10.15497/RDA0050

Tier 3

Tier 3 represents the highest level of data compliance within EUCAIM, enabling full access to its functionalities. Data within Tier 3 can be used to train or validate algorithms in a federated manner. This, however, implies that data meet specific minimal standards to mitigate the effects produced by the heterogeneous sources of data on model output and maximise the power of EUCAIM as a repository of cancer images.

Firstly, data within Tier 3 needs to comply with Tier 1 and Tier 2 requirements. Secondly, data must adhere to standardised formatting guidelines:

- Image data in DICOM format has to be structured following the Dataset>Patient>Study>Series hierarchy. The **DICOM File integrity checker tool**, mentioned in the previous sections, also allows restructuring of the data to comply with this requirement.
- Clinical data should comply with the EUCAIM CDM, which is handled by a dedicated Extract Transform Load (ETL) process (see section D “Compliance to EUCAIM CDM”).

In addition, to adhere to a standardised format, the data within Tier 3 should comply with minimal data quality standards. The requirements and tools necessary to reach the minimum data quality to become Tier 3 data are further described in Section B.

As mentioned for Tier 2, FAIR indicators can be classified by different priorities (essential, important and useful). So while full compliance of all indicators can not be expected for Tier 3, higher expectations are placed for the indicators that are considered essential by the RDA, in particular by adding more data, while in the previous Tiers the indicator on metadata dominated. A detailed description of the requirements for each of the Tiers can be found in D4.3.

Tier 3*

Tier 3* is an extension of Tier 3, designed to incorporate datasets enriched with annotations. These annotations should be either manually created or (semi-)automatically generated and validated by expert clinicians. The segmentations can be produced outside the EUCAIM infrastructure before data ingestion into the central node or by utilising the EUCAIM central node viewer and segmentation algorithms to generate new annotations for new datasets. Annotations must be in the standard DICOM SEG format. However, since segmentation annotations might be in other formats, such as NIfTI or DICOM RT Struct, a **non-standard converter tool** is provided within EUCAIM to facilitate compliance with this requirement. Further information can be found in Section C.

B. Preprocessing tools on premises

De-identification

De-identification pipeline in EUCAIM

EUCAIM is designed to store extensive amounts of medical imaging data alongside corresponding clinical information. Given the sensitive nature of this data, the project must adhere to stringent policies to ensure the protection of patients' personal information. A crucial component of this effort is the effective de-identification of data before it is integrated into the EUCAIM ecosystem. EUCAIM employs a mixed federated approach, allowing data to be stored either at federated nodes within providers' facilities or at a central node. Regardless of the storage location, all data must undergo a thorough de-identification process. This process needs to be tailored based on various factors, including the type of data (imaging or clinical).

Imaging Data

The primary format for imaging data within EUCAIM is DICOM (Digital Imaging and Communications in Medicine). DICOM files include metadata in the form of tags, many of which are standardised and structured to enhance data interoperability. To de-identify sensitive information in these metadata tags, EUCAIM has developed a series of de-identification profiles based on different imaging modalities. These profiles outline specific actions for each DICOM tag, indicating whether the tag should be retained (if it does not contain personal or identifying information) or erased/modified (if it does).

Utilising these profiles, we have created a de-identification tool, which is part of the EUCAIM tool catalogue. This tool can be downloaded and used by data providers to ensure the proper de-identification of DICOM images, thereby safeguarding patient privacy while maintaining the utility and interoperability of the medical imaging data. Moreover, certain imaging modalities may necessitate additional steps in the de-identification process, such as removing burned-in text or de-facing brain-related images. To address these specific requirements, we are integrating tools designed to assist with these tasks, including the DICOM2usb de-facing tool (D5.4 Supplementary Material).

Clinical data

Standardising clinical data across various providers' databases presents significant challenges. While EUCAIM is developing a CDM to ensure data interoperability and facilitate Federated Search (see deliverable D5.1), it also aims to support Data Holders by accommodating data at different levels (Tiers) of compliance with the Data Federation Framework (see deliverable D4.3). The de-identification process for clinical data varies depending on whether the data conforms to the EUCAIM CDM.

Non-Compliant Data

For clinical data that does not adhere to the CDM, Data Holders must provide a detailed description of each clinical variable in the dataset, ensuring that none contain sensitive information that could identify patients. The access committee will establish validation steps

to confirm proper de-identification. This may involve techniques like K-anonymity, which helps identify and manage outliers that could potentially reveal patient identities.

Compliant Data
When clinical data is compliant with the CDM, EUCAIM can offer tools such as the **Wizard tool** (see section below). This tool is designed to validate the de-identification applied to datasets, identify potential risks, and propose mitigation strategies. By providing these mechanisms and tools, EUCAIM aims to streamline the participation of Data Holders and ensure robust data de-identification practices across varying levels of compliance, enhancing the overall security and privacy of patient information within the ecosystem.

Wizard tool

W1. Goals of the Wizard tool

The rationale for the development of an EUCAIM **Wizard tool** stems from the need to optimally integrate a plethora of datasets from providers with variable degrees of experience and different established workflows regarding data sharing. The overall effort is based on maintaining data **clinical value** and **data security** which often pose contradicting requirements. The whole process needs to be **adaptive** to different use cases since the endpoints of future data contributions are not yet defined. Moreover, the workflow needs to consider the degree of **homogeneity** which is an important factor for data usability.
The aim of the Wizard, as discussed in the initial meetings among the EUCAIM participants, can be summarised in:

- Support the identification of risks - propose ways to mitigate them
- Raise awareness of the weak points of each process
- Foster a secure-by-design anonymization planning
- Facilitate compliance with EUCAIM requirements and accountability obligations

W2. Functionalities

W2.1 Blacklisting per modality (Imaging data, cohort/cancer type agnostic)
Starting from the different de-identification profiles of the five AI4HI projects, a working sheet has been built with the different views of each project regarding the handling of imaging metadata, i.e. DICOM header tags. A large degree of variability has been noticed which was the starting point to formulate the rationale of the EUCAIM **Wizard tool**.

W2.2 Whitelisting per modality (Imaging data, cohort/cancer type agnostic)
Whitelisting may precede blacklisting or may be performed as a separate task. The rationale for whitelisting-as opposed to blacklisting- is to keep only the necessary tags for further processing rather than removing the dangerous tags. This results in a more concise and homogeneous list of tags among partners. Both black and whitelisting tags will be modality-specific in order to avoid unrelated or obsolete tags that contaminate the list and make processing more cumbersome. Moreover, one basic aspect of the whitelisting process is the definition of the potential levels for the tags that have been characterised as conveying information of variable degrees of accuracy. Such information will be inspected by contributors to list and define the potential intervals or broader categories at which information can be modified when this specific attribute is considered at high risk for identification, either for a single individual or for the whole cohort. The specific circumstances that may entail such information modification will be the output of the risk assessment

process and will be proposed for individuals at risk (with a unique combination of quasi-identifiers) or for cohorts with underpopulated attributes. The different levels of accuracy will be used to propose to the user the potential solutions of minimised risk during the iterative process where the data provider interacts with the system to finalise the schema of the provided cohort information.

W2.3 Risk assessment (clinical and imaging dataset)

Starting from the fact that anonymization is not a binary concept, as stated by the Spanish protection agency, the Wizard aims to analyse, measure, and optimise the degree of anonymization. Characterising data as anonymous is very important, as when this characterization is gained, the information falls outside the restrictions of the GDPR. However, the question about when data is anonymous can hardly be answered. A feasible goal is to take measures against re-identification in order to minimise the risk of data being re-identified under means reasonably likely to be used in terms of time, cost, and available technology. The factors that have to be taken into account for increasing the security regarding a specific cohort are –but are not limited to– the size and variable distribution. However, the distributions may be subjected to changes in order to optimise the distributions, especially for underpopulated categories. That has to be performed with caution so as not to obscure the specific identity of each participant and thus compromise data value. This entails a double-faced attention to both data security and usability.

To calculate the total risk of a specific dataset, it is important to integrate the total amount of information that is provided for any individual. Thus, it is necessary to aggregate all available information in an appropriate format that can be effectively inspected. Inspection aims to detect missing information, overlapping or duplicated entries, as well as problematic entries or entry types. The most efficient type of file format to associate imaging data and clinical data integration is a tabular working sheet, where available clinical variables, as well as DICOM header extracted tags, will be registered. One of the aims of the **Wizard tool** is to inspect and label tags automatically, as indicatively shown in Figure 4. Each working sheet will be tailored to be specific for each different cancer type.

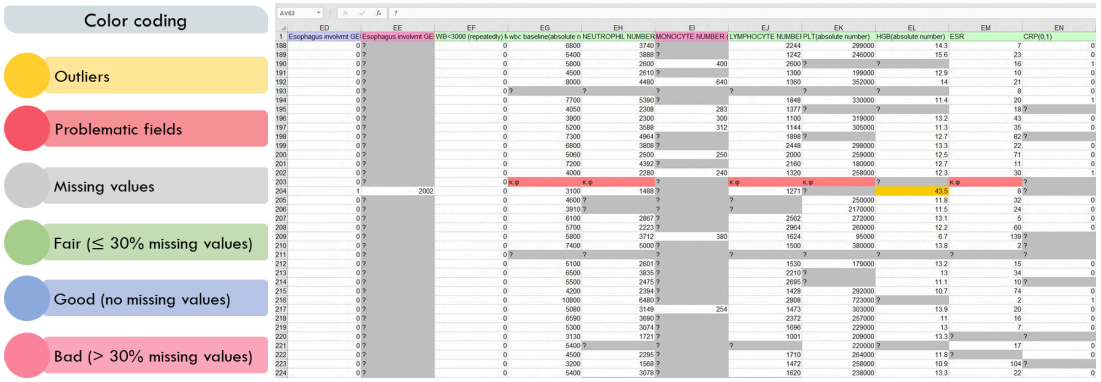


Figure 4. Example working sheet showing the results of the risk assessment and the detected risks associated with a colour code.

W2.4 Risk Optimization - Iterative process

Re-identification risk, as risks derived from population uniqueness, can be estimated with different statistical methods. The **Wizard tool** aims to initiate an iterative process where possible re-identification scenarios are considered by the user and will be tailored according to both needs of security and data integrity.

The solution that best fits, given the individual characteristics of each cohort, will be the final output of the iterative risk estimation process, requiring the feedback of the data provider to conclude the final plan. Firstly, for each quasi-identifier, a graph showing the initial distribution will be presented to identify the underpopulated categories. Moreover, combinations of attributes will be formed to analyse the degree to which different attribute combinations of variables separate the records from each other, which is an indication of the number of individuals -and attributes- at higher risk for re-identification. Population uniqueness in a sample are the individuals with a unique combination of attributes. It is considered that cohort uniques are also unique within the underlying population from which the data has been sampled. Different models are considered to return the estimates of population unique. Their results will be compared, as each model conveys different assumptions. During this process, some user-defined inputs will be required regarding the sample size, the prevalence of the specific pathology in the general population, etc. Each specific anonymization schema will be graded differently according to its security and usability under different sample sizes or representations in the cohort.

As mentioned above, the whitelisting process will include different levels of accuracy for some specific tags from the imaging data that are either numeric or categorical. Considering that a similar process will be performed for the clinical entries per cancer type, a hierarchy of quasi-identifiers for the total amount of information will result. The accuracy at which each attribute will be accepted in combination with other attributes and sample characteristics will be the field of investigation of the risk minimisation process.

Many possible solutions will be the outcome of the user's interaction with the **Wizard tool**, where the user will inspect different ratings of data security and usability under proposed de-identification schemas. The user will be invited to decide on the best-tailored list of tags with specific accuracy and specific care taken for individuals at risk. The whole process aims to make the user aware of the danger of escalation, to exhaust the possibility of protecting data while ensuring usability and maintaining a reasonable balance among these two contradicting needs. Specific algorithms will provide qualitative or quantitative metrics for the data security and usability before and after the iterative Wizard-guided process. This information will be part of the dataset identity.

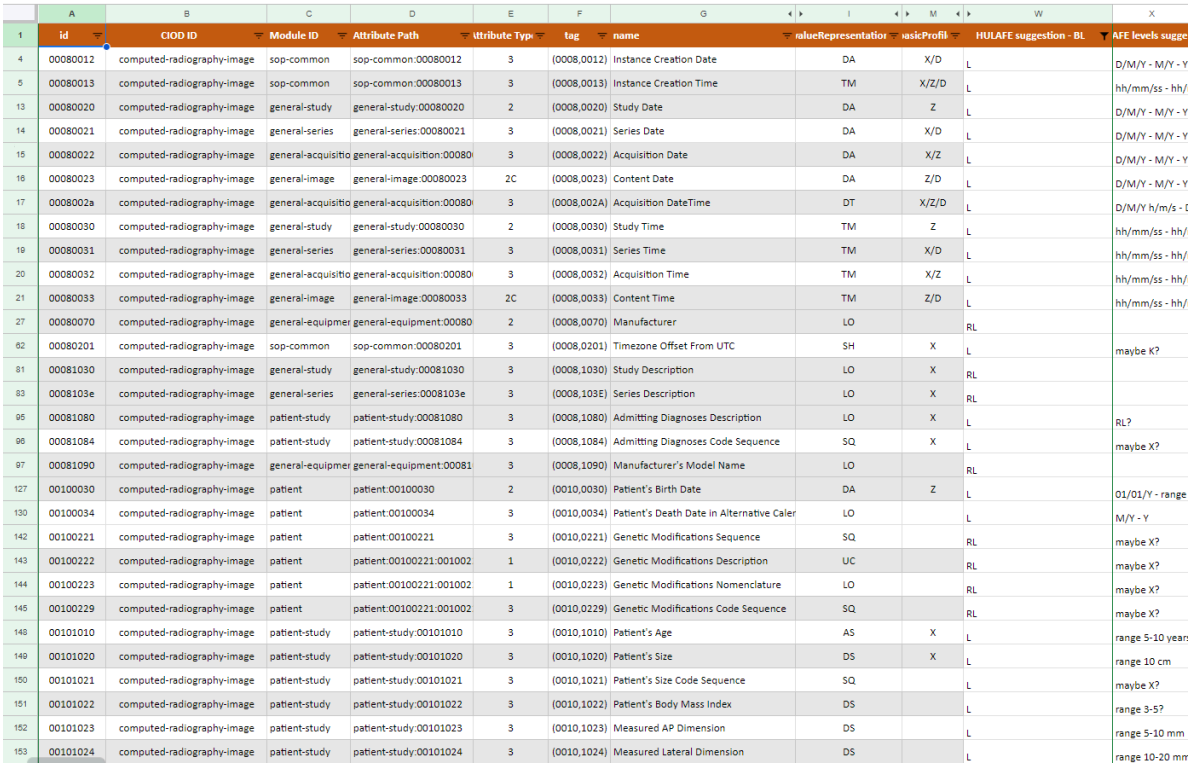
• Descriptors of input data (i.e., number of records, number of attributes)	• Quasi-identifying attributes, type of entries (i.e., string, controlled text)	• Hierarchies, number of levels per attribute (where relevant)
• Number of solutions proposed (i.e. search space size)	• Privacy models used (i.e., k-Anonymity)	• Solutions materialized for the final outcome
Final Risk outcome		

Figure 5. Indicative information produced during the risk minimization process and presented in the final report

W2.5 Report

The final report will contain a descriptive list of proposed and completed actions regarding the optimization process of data security and data usability that was guided within the EUCAIM **Wizard tool**. The estimated level of data security as well as the level of data usability will be stated in metrics and units that will be defined by the Wizard team. The report will be issued after the whole cohort set is known as it will describe the level of data

security for the specific population with the given distribution of each quasi-identifier. Indicatively, the number of individuals at relatively higher danger for re-identification will be stated, at different stages of the Wizard workflow, most importantly at the beginning and the end of the process. The individuals at risk for re-identification will be identified based on unique combinations of quasi-identifiers within the specific cohort, given the levels of tags and corresponding attribute intervals of the final de-identification plan.



	A	B	C	D	E	F	G	I	M	W	X
1	id	CIOD ID	Module ID	Attribute Path	ttribute Typ	tag	name	alueRepresentatio	asicProfil	HULAFE suggestion - BL	AFE levels sugge
4	00080012	computed-radiography-image	sop-common	sop-common:00080012	3	(0008,0012)	Instance Creation Date	DA	X/D	L	D/M/Y - M/Y - Y
5	00080013	computed-radiography-image	sop-common	sop-common:00080013	3	(0008,0013)	Instance Creation Time	TM	X/Z/D	L	hh/mm/ss - hh/
13	00080020	computed-radiography-image	general-study	general-study:00080020	2	(0008,0020)	Study Date	DA	Z	L	D/M/Y - M/Y - Y
14	00080021	computed-radiography-image	general-series	general-series:00080021	3	(0008,0021)	Series Date	DA	X/D	L	D/M/Y - M/Y - Y
15	00080022	computed-radiography-image	general-acquisitio	general-acquisition:00080	3	(0008,0022)	Acquisition Date	DA	X/Z	L	D/M/Y - M/Y - Y
16	00080023	computed-radiography-image	general-image	general-image:00080023	2C	(0008,0023)	Content Date	DA	Z/D	L	D/M/Y - M/Y - Y
17	0008002a	computed-radiography-image	general-acquisitio	general-acquisition:00080	3	(0008,002A)	Acquisition DateTime	DT	X/Z/D	L	D/M/Y h/m/s - [
18	00080030	computed-radiography-image	general-study	general-study:00080030	2	(0008,0030)	Study Time	TM	Z	L	hh/mm/ss - hh/
19	00080031	computed-radiography-image	general-series	general-series:00080031	3	(0008,0031)	Series Time	TM	X/D	L	hh/mm/ss - hh/
20	00080032	computed-radiography-image	general-acquisitio	general-acquisition:00080	3	(0008,0032)	Acquisition Time	TM	X/Z	L	hh/mm/ss - hh/
21	00080033	computed-radiography-image	general-image	general-image:00080033	2C	(0008,0033)	Content Time	TM	Z/D	L	hh/mm/ss - hh/
27	00080070	computed-radiography-image	general-equipmer	general-equipment:00080	2	(0008,0070)	Manufacturer	LO		RL	
62	00080201	computed-radiography-image	sop-common	sop-common:00080201	3	(0008,0201)	Timezone Offset From UTC	SH	X	L	maybe K?
81	00081030	computed-radiography-image	general-study	general-study:00081030	3	(0008,1030)	Study Description	LO	X	RL	
83	0008103e	computed-radiography-image	general-series	general-series:0008103e	3	(0008,103E)	Series Description	LO	X	RL	
95	00081080	computed-radiography-image	patient-study	patient-study:00081080	3	(0008,1080)	Admitting Diagnoses Description	LO	X	L	RL?
96	00081084	computed-radiography-image	patient-study	patient-study:00081084	3	(0008,1084)	Admitting Diagnoses Code Sequence	SQ	X	L	maybe X?
97	00081090	computed-radiography-image	general-equipmer	general-equipment:00081	3	(0008,1090)	Manufacturer's Model Name	LO		RL	
127	00100030	computed-radiography-image	patient	patient:00100030	2	(0010,0030)	Patient's Birth Date	DA	Z	L	01/01/Y - range
130	00100034	computed-radiography-image	patient	patient:00100034	3	(0010,0034)	Patient's Death Date in Alternative Caler	LO		L	M/Y - Y
142	00100221	computed-radiography-image	patient	patient:00100221	3	(0010,0221)	Genetic Modifications Sequence	SQ		RL	maybe X?
143	00100222	computed-radiography-image	patient	patient:00100221:001002	1	(0010,0222)	Genetic Modifications Description	UC		RL	maybe X?
144	00100223	computed-radiography-image	patient	patient:00100221:001002	1	(0010,0223)	Genetic Modifications Nomenclature	LO		RL	maybe X?
145	00100229	computed-radiography-image	patient	patient:00100221:001002	3	(0010,0229)	Genetic Modifications Code Sequence	SQ		RL	maybe X?
148	00101010	computed-radiography-image	patient-study	patient-study:00101010	3	(0010,1010)	Patient's Age	AS	X	L	range 5-10 year
149	00101020	computed-radiography-image	patient-study	patient-study:00101020	3	(0010,1020)	Patient's Size	DS	X	L	range 10 cm
150	00101021	computed-radiography-image	patient-study	patient-study:00101021	3	(0010,1021)	Patient's Size Code Sequence	SQ		L	maybe X?
151	00101022	computed-radiography-image	patient-study	patient-study:00101022	3	(0010,1022)	Patient's Body Mass Index	DS		L	range 3-5?
152	00101023	computed-radiography-image	patient-study	patient-study:00101023	3	(0010,1023)	Measured AP Dimension	DS		L	range 5-10 mm
153	00101024	computed-radiography-image	patient-study	patient-study:00101024	3	(0010,1024)	Measured Lateral Dimension	DS		L	range 10-20 mm

Figure 6. Working Excel file presenting DICOM tags, entry types, and tag hierarchies prepared during the Wizard working groups to define the blacklist.

W3. Completed Work

Blacklisting per modality: A collaborative effort has been made to define the Blacklisting protocol for each modality specifically. The total set of possible modalities has been divided into 8 broad categories, i.e CT, MRI, X-rays and mammography, US, Nuclear medicine, PET, etc. The full list of DICOM tags possibly present in each category has been listed and reviewed by each partner to be examined regarding its value for use in possible scenarios and regarding its ability to jeopardise patients anonymity.

A multi-centric team has been set to assign different labels for each DICOM tag regarding the proposed action in order to be admitted into the EUCAIM repository. Each team overtook specific modality groups and defined a proposed action. As a second step, each team presented for discussion the proposed actions to the Wizard partners and put the decision forward for discussion and consent. Four meetings were held for this purpose during the EUCAIM de-identification working group, with broad participation. A modality specific worksheet comprised the DICOM tag number, DICOM tag descriptions-as derived from Innolitics[2], the value representation of each tag, the type, as well as the position of each tag in a broader category of tags, when nested. One column was created to host the full

[2] https://dicom.innolitics.com/ciods

description of the tag. The possible proposed actions were marked as U: Update, C: Clean, X: delete, Z: Zero, to match the DICOM 144 committee document for data de-identification. Other labels were also added to meet the needs of the EUCAIM project, mainly L, standing for “Levelled”, implying the need to define the possible levels of accuracy that this tag will obtain in the whitelisting process. For the Blacklisting process, tags that are likely not related to the clinical value of the dataset in a possible future use case have been identified by the whole Wizard team and have been labelled accordingly. The orientation of blacklisting is to discard the possibly dangerous tags for re-identification and thus contributes more to data usability than data security and homogeneity.

ralueRepresentatior	asicProfil	UCAIM Profil	ralueMultiplicity	etire	Description
UT	X	K	1	N	Issuing Agency. See Section 10.29.1.
LO	X	K	1	N	Further description in free form text describing the device. This can be used to distinguish between Items when multiple UDIs are recorded in a Sequence.
TM	X	K	1-n	N	Time when the image acquisition device calibration was last changed in any way. Multiple entries may be used. See Section C.7.5.1.1.1 for further explanation.
LT	X	C	1	N	User-defined comments about the image.
SQ	X		1	N	This icon image is representative of the Image. Only a single Item is permitted in this Sequence.
ST	X	C	1	N	A text description of how this image was derived. See Section C.12.4.1.1 for further explanation.
PN	X	X	1-n	N	Name of the physician(s) administering the Series.
SQ	X	X	1	N	Identification of the physician(s) administering the Series. One or more Items are permitted in this Sequence. If more than one Item, the number and order shall correspond to the value of Performing Physician's Name (0008,1050), if present.
PN	X/Z/D	X	1-n	N	Name(s) of the operator(s) supporting the Series.
SQ	X/D	X	1	N	Identification of the operator(s) supporting the Series. One or more Items are permitted in this Sequence. If more than one Item, the number and order shall correspond to the value of Operators' Name (0008,1070), if present.
LO	X/D	C	1	N	User-defined description of the conditions under which the Series was performed. Note This Attribute conveys Series-specific protocol identification and may or may not be identical to the protocol described in the Performed Protocol Code Sequence (0040,0260) in the Performed Protocol Code Sequence (0040,0260) in Table 10-16 “Performed Procedure Step Summary Macro Attributes”.
SH	X		1	N	User or equipment generated identifier of that part of a Procedure that has been carried out within this step.
LO	X	C	1	N	Institution-generated description or classification of the Procedure Step that was performed.
SQ	X	C	1	N	Sequence that contains Attributes from the Imaging Service Request. One or more Items are permitted in this Sequence.
LO	X/Z	C	1	N	Institution-generated administrative description or classification of Requested Procedure.
SQ			1	N	A Sequence that conveys the Procedure Type of the requested procedure. Only a single Item is permitted in this Sequence.
LO	X	C	1	N	Institution-generated description or classification of the Scheduled Procedure Step to be performed.
SQ			1	N	Sequence describing the Scheduled Protocol following a specific coding scheme. One or more Items are permitted in this Sequence.

Figure 7. Working Excel sheet prepared during the blacklisting per modality definitions. Wizard group consensus on proposed actions and tag descriptions as well as tag types and multiplicities for consultation during discussions is shown on the dedicated columns for every tag appearing in each row.

De-identification tools

For the current demo, the tool that is shown is the **EUCAIM DICOM Anonymizer**. It is a standalone (desktop) application for Microsoft Windows and MacOS 64 bit machines, and its function is to anonymize a set of DICOM files. It supports both anonymization on a case-by-case scenario, meaning one DICOM Study at a time (single-mode) or on multiple cases concurrently (batch mode). Once the tool is opened, the user can drop the folder where the DICOM images are present to de-identify them according to a preconfigured de-identification profile.

Figure 8. EUCAIM DICOM Anonymizer interface.

In order to perform the DICOM anonymization, the EUCAIM DICOM Anonymizer integrates the RSNA DICOM Anonymizer[3]. After the DICOM files are anonymized, the user can inspect both the images and the associated DICOM tags using an integrated DICOM viewer. The resulting anonymized files will be stored locally in a results folder.

Together with this tool and the **Wizard tool**, we have been working on the incorporation of additional de-identification tools:

- **Radiomics Enabler:** Commercial tool to extract, de-identify (using CTP Anonymizer, pixel mask, and dateshift), and export data from hospital facilities to external repositories.
- **Mainzelliste:** Open-Source web-based pseudonymization and record linkage solution in use and co-developed at many institutions.
- **MainSEL:** short for "Mainzelliste Secure EpiLinker"; an extension to Mainzelliste to perform Record Linkage using Secure Multi-Party Computation, thus without revealing input data.
- **DICOM2usb DICOM exporter and anonymizer tool:** a tool aimed to fully anonymize any DICOM data except data containing burned-in text in images (thus, not Ultrasound). Defacing of DICOM images is included.

Further information about the tools can be found in the D5.4 Supplementary Material.

Minimum data quality assessment

All data holders must comply with a standard for data quality. The EUCAIM standard for data quality is not finalised yet, but it will require, at minimum, that each DH provides data that comply with basic principles of data quality, in full alignment with EU Regulations[4], such as:

1. No personally identifying information is available in the data;

[3] https://mircwiki.rsna.org/index.php?title=The_DicomAnonymizerTool
[4] https://www.edps.europa.eu/data-protection/data-protection/glossary/d_en#data_quality

2. Transparency of the data: any label, variable, or wording associated with the data should be self-explanatory;
3. Integrity of the data: the data holder certifies that, to the best of their knowledge, data are not corrupted/damaged;
4. Minimisation of the data: the amount of data provided is limited to what is considered necessary by the data holder to pursue research on the data;
5. Accuracy of the data: the data holders certify that, to the best of their knowledge, all information contained in the data is accurate.

This should also pave the way to more FAIRness in source data. A more mature version of the data quality standard will be provided in the next deliverable on preprocessing tools. To address this standard, data holders will be provided with a set of local tools for data quality assessment and data cleaning classified as “highly recommended”. Because they may not apply to all data types, and/or may not always be relevant, we will not impose their use on data holders, but they will be encouraged to use them whenever possible.

Tier 1

For Tier 1 data, since the data are not compliant with EUCAIM CDM, no further processing is required after de-identification. As Tier 1 clinical data are not standardised, no further tool can easily be applied, so if transferred to the central node, they may be kept in the quarantine environment of the central platform after de-identification until some level of standardisation can be achieved. As for imaging data, a certain standardisation is already followed with the DICOM format, which allows the further use of processing tools. Data holders will nonetheless be encouraged to apply to their imaging data the highly recommended local tools - if applicable (Table 1). They will be provided with instructions on how to download and use them (see D5.4 Supplementary material).

- **Dicom File Integrity Checker**: as briefly exposed above (see the “Data preprocessing depending on the Data Tiers” section), the tool performs, among other features, a quality check in terms of the correct number of files per sequence and generates a report. In addition, the tool may be used to detect information in DICOM metadata to allow federated searches, allowing conversion to Tier 2 level for imaging data (Figure 1). This tool is showcased in the demo related to this deliverable.

- **Trace4MedicalImageCleaning:** This tool, developed by DeepTrace, aims at detecting and removing encapsulated text in 2D ultrasounds and mammographies, as they may contain potentially identifying information. This tool is under development.

If the imaging data are sent to the central node, the data holder will have to provide, along with the data, any report resulting from the use of EUCAIM local tools (both for de-identification and data quality assessment) as a certificate of data quality. If the EUCAIM tools were not used, the data holder should explain why and provide additional proof of minimal data quality.

Tier 2

Tier 2 imaging data have associated metadata that allow federated querying and thus further preprocessing and quality checks. Once data have been successfully de-identified,

additional considerations are required in order to enhance their quality. Currently, the same tools are available for Tier 2 data as for Tier 1 data. In addition to the detection of undesirable files, the **DICOM File Integrity Checker** tool offers the possibility to specify the directory hierarchy and organise all DICOM files following the structuring Dataset>Patient>Study>Series, allowing conversion of imaging data to Tier 3. Note: it will be suggested to DHs to apply the **Dicom File Integrity Checker** tool to the data before running the **Trace4MedicalIMagesCleaning** tool, in order to be able to select the relevant series and discard possible corrupted files.

In addition to the preprocessing tools described, a tool allowing for imaging metadata extraction and compilation in a structured JSON file will be provided to DHs for Tier 2, in order for the federated query component to run on the data. This will be further documented in the next deliverable (D5.5) dedicated to data and metadata ingestion and transformation workflow.

Tier 3

Since data within Tier 3 needs to comply with Tier 1 and Tier 2 requirements, the same tools as in other Tiers are provided to DHs. The structuration of Tier 3 imaging data allows the use of most processing tools and AI methods. Furthermore, clinical data in Tier 3 are fully compliant with the EUCAIM CDM; as a consequence, in terms of minimal local data quality assessment, additional tools are highly recommended to DHs for clinical data (Table 1):

- **Data Integration Quality Check Tool (DIQCT)**: This tool developed by AUTH applies to both clinical and imaging data. The tool checks the clinical metadata quality (validity, completeness), the integrity between images and clinical metadata provided, the de-identification protocol applied, the imaging analysis requirements, and the existence of annotation, and informs the user on corrective actions prior to data upload.
- **Tabular data curator**: This tool, developed by FORTH, applies to tabular data such as clinical data. It identifies duplicated fields (lexically similar and/or highly correlated features), outliers, and data inconsistencies, and provides options to deal with missing values. This tool will only apply to clinical data provided in tabular format.

In alignment with the **Tabular data curator** functions, Tier 3 clinical data will be further processed using an ETL to allow for structured tabular datasets to be 1) ingested and harmonised, 2) transformed to EUCAIM CDM, and 3) shared with authorised users (see below section D on “Compliance to EUCAIM CDM”). The ETL is currently under development and will build on the existing tools at hand.

Level	Data quality category	Data eligibility	Data type	Tool deployment	Tier 1	Tier 2	Tier 3
Highly recommended	· Data integrity	Dicom File Integrity Checker	Imaging data	Local (and central)	✓	✓	✓
	· De-identification	Trace4MedicalImageCleaning	Imaging data	Local (and central)	✓	✓	✓
	· Data integrity · De-identification	Data Integration Quality Check Tool	Imaging and Clinical Data	Local (and central)			✓

	• Data completeness • Data minimization	Tabular data curator	Clinical data	Local (and central)			✓

Table 1: List of highly recommended local tools for data quality assessment and data cleaning.

Data fairness

EUCAIM's approach to checking data FAIRness is based on the RDA (Research Data Alliance) Data Maturity Model (FAIR Data Maturity Model. Specification and Guidelines[5] (zenodo.org)), that defines 41 indicators for different aspects of findability, accessibility, interoperability and reusability for both data and metadata.

There are some limitations to the FAIR compliance possible with medical data, and medical imaging in particular, due to the sensitive nature of the same. Due to this, no check should be done at the level of the individual subject, but on datasets.

EUCAIM uses the FAIR:EVA (Evaluator, Validator & Advisor) developed in the EOSC Synergy projects. Its adaptation to EUCAIM's requirements has already started. In particular, the adapted tool is able to query FAIR Data Points, like the one to be available in the EUCAIM Catalogue. It will also check the presence of the mandatory attributes (defined by EUCAIM) in the datasets that will support distributed queries.

[5] https://zenodo.org/records/3909563#.ZFn7NHZBybh

C. Preprocessing tools on the central node

Data quality tools

As no local preprocessing tool is made mandatory to use at this stage of the project, all the highly recommended local tools described above will also be available centrally to the end users, and apply to the same Tiers.

Highly recommended central tools

For imaging and clinical data (any Tier), additional tools for data quality checks may be available to the end user in the marketplace. At this stage, one tool is highly recommended to use for imaging data (Table 2):

- **Image Duplicate Checker**: This tool developed by AUTH allows for checking that data from a selected dataset do not contain duplicates. This is particularly relevant in the case of a dataset of selected data from various sources that are at risk of containing the same data.
  Example 1: a DH provides a large dataset of medical images from patients with prostate cancer; that same data holder has also contributed to one AI4HI project on prostate cancer, with possible overlap between this dataset and the one he/she provided as DH.
  Example 2: a patient has a medical examination in a hospital, and moves to another city where he/she undergoes follow-up medical examinations. Both hospitals may provide EUCAIM data from the same patient, as two different ones.

  This tool may apply to any imaging data already structured.

For Tier 3 clinical data, we highly recommend the use of the following tool:

- **Time coherence tool**: this tool developed by HULAFE's team aims to validate the chronological order and logical consistency of dates associated with a patient's medical history, from tabular datasets (Table 2).

Level	Data quality category	Data eligibility	Data type	Tool deployment	Tier 1	Tier 2	Tier 3
Highly recommended	· De-identification	Trace4MedicalImageCleaning	Imaging data	Central (and local)	✓	✓	✓
	· Data integrity · De-identification	Data Integration Quality Check Tool	Imaging and Clinical Data	Central (and local)			✓
	· Data integrity	Dicom File Integrity Checker	Imaging data	Central (and local)	✓	✓	✓
	· Data completeness · Data minimization	Tabular data curator	Clinical data	Central (and local)			✓

	· Redundancy	Image duplicate checker	Imaging data	Central		✓	✓
	· Data coherence	Time coherence tool	Clinical data	Central			✓

Table 2: List of highly recommended central tools for data quality assessment and data cleaning

Optional tools

A large set of additional tools are available from the central marketplace to run on Tier 2 and/or Tier 3 data (Table 3). Nine of them apply to imaging data only, and tackle mostly noise reduction, bias reduction, and other types of filtering. One final tool addresses clinical datasets and bias reduction. While they are all made optional to use, they are dedicated to improving data quality and potentialize their use for higher-level processing (ML, AI-based tools, etc.).

Level	Data quality category	Data eligibility	Data type	Tool deployment	Tier 1	Tier 2	Tier 3
Optional	· Data minimization · Other features	ML models for MR series categorization	Imaging data	Central		✓	✓
	· Bias assessment and reduction	N4 Bias filter	Imaging data	Central			✓
	· Noise assessment & reduction · Other features	ImageQure	Imaging data	Central			✓
	· Noise assessment & reduction	NLmCED denoising filter	Imaging data	Central			✓
	· Noise assessment & reduction	Denoising inhomogeneity correction	Imaging data	Central		✓	✓
	· Noise assessment & reduction	Deep Learning Noise Reduction	Imaging data	Central			✓
	· Noise assessment & reduction · Other features	Image quality assessment metrics for XNAT	Imaging data	Central			✓
	· Noise assessment & reduction · Other features	MR image quality tool	Imaging data	Central			✓
	· Other filtering	RACLAHE filter	Imaging data	Central			✓
	· Bias assessment and reduction	Extended a Priori Probability	Clinical data	Central			✓

Table 3: List of optional tools in the central node for data quality assessment and data cleaning

- **ML models for MR series categorization**: a tool developed by HULAFE, that categorises MRI series by using standardised DICOM tags. The categorisation includes the type of sequence, the weighting, the presence of fat suppression and the detection of non-relevant / junk series.
- **N4 Bias filter**: a tool developed by FORTH to apply bias field correction to T2W MR prostate images. This tool is showcased in the demo related to this deliverable.

- **ImageQure**: a tool developed by Medex, providing several metrics of signal and noise, artefacts, and overall quality from T2W MR images.
- **NLmCED denoising filter**: the Local mean Coherence Enhancing Diffusion (NLmCED) denoising filter is a denoising tool developed by CNR/Eurobioimaging that aims to reduce Rician Noise in MR images.
- **Denoising inhomogeneity correction**: a customisable image preprocessing tool developed by HULAFE, allowing to apply five of the most common denoising filters as well as a N4 bias field correction filter. The parameter configuration of this tool has been optimised for T1W, T2W, Diffusion Weighted Image (DWI) and Dynamic Contrast Enhanced (DCE) sequences in neuroblastoma and paediatric brain tumours but it can also be configured for other types of studies as well.
- **Deep Learning Noise Reduction**: a tool developed by FORTH, meant to reduce the noise on already noisy MR prostate images.
- **Image quality assessment metrics for XNAT**: a tool developed by CNR-IBB to assess the quality of medical image datasets within an XNAT platform, both at subject level and at project (dataset) level.
- **MR image quality tool**: a comprehensive UI tool developed by FORTH to report image quality scores and types of artefacts from conventional or dynamic MR series.
- **RACLAHE filter**: a tool developed by FORTH, applicable to prostate axial T2W MR images, to enhance the prostate gland area. This is of particular interest for subsequent segmentation tasks.
- **Extended a Priori Probability** (**EAPP**): a tool developed by ITI providing a semi-supervised metric that evaluates the ease or complexity of a binary classification task, highlighting potential biases in the dataset.

Annotation tools

Within annotation, different tasks such as segmentation, detection, and classification are differentiated. The first task to be addressed is segmentation because of the following reasons:

- **Clinical time consumption.** Segmentation is a time-consuming task for clinicians. The use of automatic segmentation tools facilitates this task by avoiding the need to annotate from scratch, enabling the clinician to solely make corrections, which significantly reduces the time required.
- **Relevance in analysis pipelines.** Segmentation is often a preliminary step in analysis pipelines. It can be useful to focus the analysis on a specific organ or lesion, allowing the extraction of characteristics from a specific region. These features can then feed models to predict clinical endpoints.
- **Standard format**. Segmentation involves the definition of a standard format and its corresponding tags and values. In addition, this standard must consider the hyper-ontology and comply with the central viewer requirements to allow the reading and writing of segmentations on it.

To facilitate the standardisation of radiological image formats collected in EUCAIM, a "DICOM in – DICOM out" strategy is followed. DICOM format is adopted as the standard for images in the EUCAIM project. To extend it to the annotations, the DICOM SEG format is proposed as the standard format for segmentations.

Conversion tool - From non-standard formats to DICOM SEG

Since segmentation annotations may be found in other formats, such as NIfTI or DICOM RT Struct, work is being done on developing and testing a conversion tool that allows these non-standard formats to be transformed into DICOM SEG.

Scenarios

Currently, the tool allows bidirectional conversion between NIfTI and DICOM SEG, ensuring both direct conversion to the standard format and the possibility of reverting to NIfTI. This facilitates the use of tools that are not yet adapted to DICOM SEG, both in their inputs and outputs, adding flexibility to the data preparation step. Because of the bidirectional nature and the possibility of having overlapping segments in the DICOM SEG, four scenarios are considered:

1. **One NIfTI to one SEG**. In the general case, when converting a segmentation from NIfTI to SEG, there is no overlapping in the SEG file since NIfTI does not allow assigning multiple labels to one pixel. Then, one SEG file is generated, given one NIfTI file.
2. **Several NIfTIs to Several SEGs**. When handling multiple NIfTI files, the contents can vary widely, ranging from multiple regions of interest (ROIs) in a volume (with a risk of overlap) to the same ROI annotated by different clinicians (multi-annotator case). Therefore, the most appropriate approach is to generate one SEG file for each NIfTI file, as the contents of the NIfTI files may vary significantly.
3. **Non-overlapping SEG to one NIfTI**. If we confirm through logical inference that the SEG segments that correspond to a volume do not overlap, then a single NIfTI file can be generated.
4. **Overlapping SEG to Several NIfTI**. In cases where overlapping segments are identified, each segment produces its own NIfTI file.

Tool implementation and execution

To perform the conversion we have tested two tools. First, we evaluated the function `itkimage2segimage` from the *dcmqi*[6] (DICOM for quantitative imaging) library developed by the Quantitative Imaging Informatics for Cancer Research[7] project. The input arguments for this function are explained in the library documentation[8]. In scenarios 1 and 2, the NIfTI segmentation needs to be accompanied by a JSON file containing segmentation metadata. This metadata describes various aspects of the segmentation, such as its origin and characteristics.

The second approach consists of a custom Python-based application that extends the *dcmqi*-based function capabilities to check for overlaps between labels of different segments, and to verify and set the proper orientation, ensuring that the segmentation data is aligned with the imaging data. Additionally, this tool automatically generates the metadata JSON file, allowing fully automated conversion processes. The inputs of this function are:

- *input path*: path to the DICOM SEG file.
- *output path*: path to the folder where the segments are stored.

[6] https://qiicr.gitbook.io/dcmqi-guide
[7] https://qiicr.org/
[8] https://qiicr.gitbook.io/dcmqi-guide/opening/cmd_tools/seg/itkimage2segimage

- *orientation* (optional): Use this option if a different orientation is required. A path of a NIfTI file and a string with the orientation that follows a R/L, A/P, or S/I schema can be passed.

In scenarios 1 and 2, both tools require the reference DICOM series over which the segmentation was performed, otherwise the conversion will not be possible. These source DICOM files provide crucial context, including information about the geometrical space of the segmentation, the patient, and the imaging study, which is inherited by the SEG file.

In scenarios 3 and 4, both tools provide not only the NIfTI file but also a JSON file containing all the necessary information from the DICOM metadata to convert it back to DICOM SEG, ensuring a consistent and comprehensive conversion process.

These functions are containerized and pushed to DockerHub[9]. The following are execution examples:

- From NIfTI to DICOM SEG: *docker run -v DicomsegTest:/tmp dicomseg:latest dicomseg --inputImageList /tmp/nifti.nii.gz --inputMetadata /tmp/metadata.json --inputDICOMDirectory /tmp/DICOM_folder --outputDICOM /tmp/dicomsegfile.dcm* where nifti.nii.gz is the segmentation in NIfTI format, metadata.json contains the imaging metadata, DICOM_folder contains DICOM files corresponding to the reference DICOM series, and dicomsegfile.dcm is the resulting DICOM SEG file.

- From DICOM SEG to NIfTI
    - *dcmqi*-based function: *docker run -v DicomsegTest:/tmp dicomseg:latest itkimage --inputDICOM /tmp/dicomsegfile.dcm -p nifti_file -t nifti --outputDirectory /tmp* where dicomsegfile.dcm is the DICOM SEG segmentation, “nifti” is the output file format, “nifti_file” is the name of the output file, and “/tmp” is the output directory where the output NIfTI file is stored.

    - Custom function: *docker run -v DicomsegTest:/tmp dicomseg:latest itkimage2 /tmp/dicomsegfile.dcm /tmp /tmp/nifti.nii.gz* where nifti.nii.gz is the NIfTI file created from the corresponding DICOM image and dicomsegfile.dcm is the DICOM SEG segmentation.

Testing

The tool is currently being tested on two datasets:

- NSCLC-Radiomics dataset[10]. This collection contains images from 422 non-small cell lung cancer (NSCLC) patients. For these patients pretreatment CT scans, manual delineation by a radiation oncologist of the 3D volume of the gross tumour volume and clinical outcome data are available. The ROIs segmented are oesophagus, neoplasms, heart, spinal cord, left lung, and right lung. These ROIs are provided in a single DICOM SEG file. The challenge proposed with this dataset is the satisfactory conversion of a multi-label dataset.
- Advanced MRI breast lesions dataset[11]. This dataset is a single-institutional, retrospective collection of 632 breast-MRI imaging sessions. The lesions were

[9] https://hub.docker.com/r/mariov687/dicomseg
[10] https://www.cancerimagingarchive.net/collection/nsclc-radiomics/
[11] https://www.cancerimagingarchive.net/collection/advanced-mri-breast-lesions/

segmented by multiple annotators. A single DICOM SEG file with all the annotations overlapped is provided. The aim of this dataset is to cover the overlapping scenario.

Next steps

Future work involves both further validation of the tool on new datasets and covering the scenario of converting the DICOM RT Struct format, widely used for radiotherapy planning.

EUCAIM Hyper-ontology for annotation

Identification of DICOM SEG tags required by users

In the context of DICOM SEG files, semantic annotation requires the use of specific tags (or attributes) to provide detailed and meaningful descriptions of segmentation data. These tags represented in the EUCAIM hyper-ontology will ensure that the segmentation can be accurately interpreted, shared, and utilised across different medical imaging systems. The first step towards semantic annotation of DICOM SEG files is the identification of attributes required by users.

A list of 132 DICOM tags was compiled from 3 sources: (1) ProCancer-I project, (2) bibliography[12], and (3) partners with experience in DICOM SEG management. Finally, 26 DICOM tags were selected for segmentation purposes and are summarised in Table 4.

Table 4. A total of 26 DICOM SEG tags were compiled from three AI4HI projects, a relevant bibliographic source, and partners experienced in DICOM SEG management

Attribute name	ID	Type	ProCancer -I	Aiello, M., et al. 2021	Partners
Study Instance UID	(0020,000D)	Conditionally Required (1C)	Y	N	Y
Series Instance UID	(0020,000E)	Conditionally Required (1C)	Y	N	Y
Segmentation fractional type	(0062,0010)	Conditionally Required (1C)	N	Y	N
Segment algorithm name	(0062,0009)	Conditionally Required (1C)	N	Y	N
Series Description	(0008,103E)	Optional (3)	Y	N	Y
Image Type	(0008,0008)	Optional (3)	Y	Y	Y
Segments overlap	(0062,0013)	Optional (3)	N	Y	N
Segment description	(0062,0006)	Optional (3)	N	Y	N
Segmented property category Code Sequence	(0062,0003)	Optional (3)	N	Y	Y
Definition source sequence	(0008,1156)	Optional (3)	N	Y	N
Segmentation algorithm identification attribute	(0062,0007)	Optional (3)	N	Y	N
Segmented Property Type Modifier Code Sequence	(0062,0011)	Optional (3)	N	N	Y
Segmentation type	(0062,0001)	Required (1)	Y	Y	Y
Segment sequence	(0062,0002)	Required (1)	Y	Y	Y
Segment number	(0062,0004)	Required (1)	Y	Y	Y
Segment label	(0062,0005)	Required (1)	Y	Y	Y
Segment algorithm type	(0062,0008)	Required (1)	Y	Y	Y

[12] Aiello, M., Esposito, G., Pagliari, G. ***et al.*** How does DICOM support big data management? Investigating its use in medical imaging community. ***Insights Imaging*** **12**, 164 (2021). https://doi.org/10.1186/s13244-021-01081-8

Attribute name	ID	Type	ProCancer -I	Aiello, M., et al, 2021	Partners
Segmented property type Code Sequence	(0062,000F)	Required (1)	Y	Y	Y
Segment Identification Sequence	(0062,000a)	Required (1)	N	N	Y
Slice Thickness	(0018,0050)	Required, Empty if Unknown (2)	Y	N	Y
Pixel Spacing	(0028,0030)	Required, Empty if Unknown (2)	Y	N	Y
Spacing Between Slices	(0018,0088)	Required, Empty if Unknown (2)	Y	N	Y
Instance Number	(0020,0013)	Required, Empty if Unknown (2)	Y	Y	Y
Laterality	(0020,0060)	Conditionally Required, Empty if Unknown (2)	Y	N	N
Manufacturer	(0008,0070)	Optional (3)	Y	N	Y
Modality	(0008,0060)	Required (1)	Y	N	Y

Due to the need of representing these DICOM SEG attributes in a standardised way, a controlled vocabulary is required. Thus, the EUCAIM hyper-ontology needs to be extended for annotation.

Hyper-ontology for annotation and DICOM SEG format

The EUCAIM hyper-ontology is a domain ontology that explicitly and semantically defines the structured and controlled vocabulary reflecting the fundamentals of the oncology domain. Besides, hyper-ontology is an application-based ontology intended for federated querying and processing and image segmentation/annotation. The main content of the hyper-ontology is developed based on the clinical and imaging knowledge provided by the AI4HI projects. For clinical knowledge, various categories are defined considering the standard concepts provided by the OMOP/FHIR data collections, such as cancer types, morphology, surgical/therapeutic procedures, cancer staging/grading, etc. For the imaging knowledge, the required data/metadata include some DICOM tags (e.g., *SeriesDescription* (0008,103E), *BodyPartExamined* (0018,0015)), standard concepts from RadLex (e.g., *Patient Position* (Radlex:RID10420), *Patient Orientation* (Radlex:RID10461)), and values required for annotations (e.g., *PZ* (peripheral zone of prostate), *TZ* (transitional zone of prostate), *CZ* (central zone of prostate)). For more details regarding the provided clinical/imaging data, an Ontology Requirements and Specification Document (ORSD)[13] is produced and available (version 0.2 beta with v1.0 in progress). We note that to annotate a cancer image, *labels/values* that target the essential characteristics of the oncology domain are required.

- **RadLex standard concepts:** standard imaging concepts, such as image modalities, annotated region, and patient position, are defined from RadLex (ProCancer-i) as follows: *Imaging Modality* (RadLex:RID10311), *Anatomic Region* (Radlex:RID13390), *Patient Position* (RadLex:RID10420)). They are defined in the hyper-ontology and aligned to other standard terminologies/ontologies (e.g., SNOMEDCT) using exact match mapping and annotated using DICOM name/tags. Figure 9 depicts an example of mapping *Imaging Modality* to *Modality* (0008,0060) and *Imaging modality* (SNOMEDCT:360037004)).

[13] https://doi.org/10.5281/zenodo.11109765

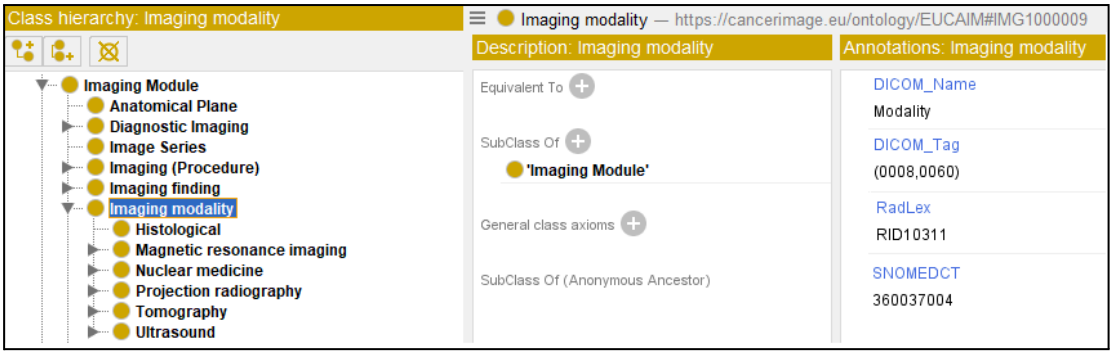


Figure 9. Part of the hyper-ontology around Imaging modality (v1.0) represented in Protege

Besides, the annotation *labels* that represent the *values* of the DICOM tags and are required for the image annotation are represented in the hyper-ontology. For instance, “MRI”/”MR” is an imaging modality label defined as a specific class of the *Imaging Modality* category (see Figure 10).

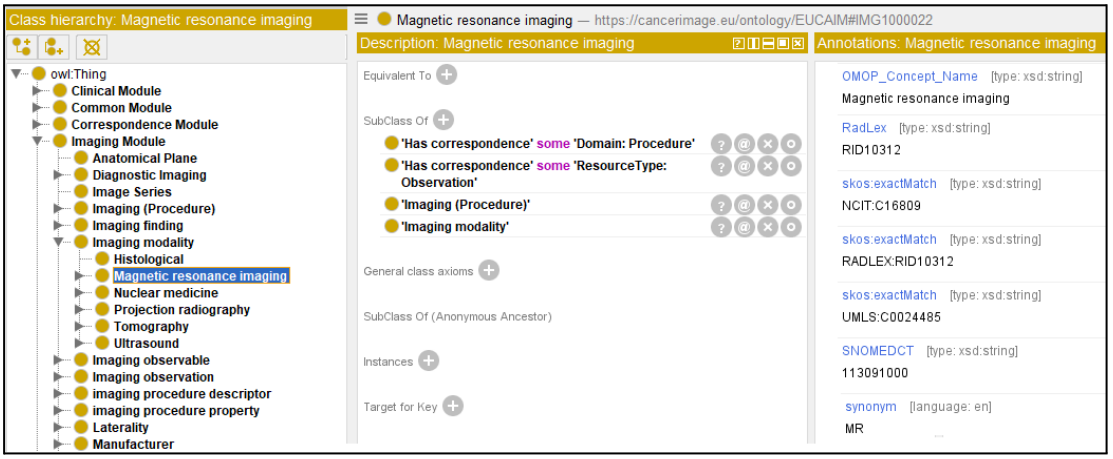


Figure 10. Part of the hyper-ontology around *Magnetic resonance imaging* (v1.0) represented in Protege

- **Annotation/Segmentation labels:** The projects provide these as labels for image querying/segmentation purposes. For instance, *PZ* (peripheral zone of prostate) and *CZ* (central zone of the prostate) are given as values associated with the DICOM tag *segment label* (0062,0005) (ProCancer-i). These labels are represented in the hyper-ontology as standard concepts from RadLex, such as *Peripheral zone of prostate* (RadLex:RID347) and *Central zone of prostate* (RadLex:RID348), and mapped to SNOMEDCT and NCIT. The standard concepts are specified in the *Body Structure* category (see Figure 11).

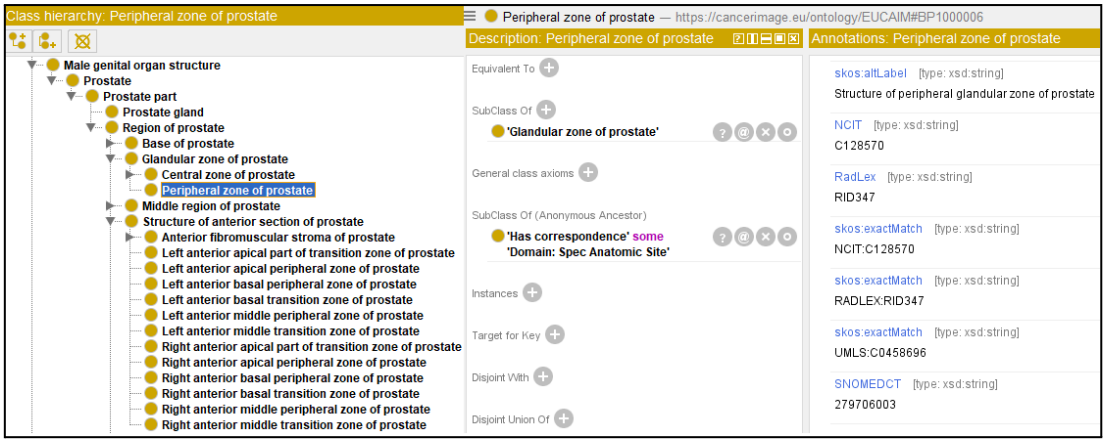


Figure 11. Part of the hyper-ontology around *the Peripheral zone of the prostate* (v1.0) represented in Protege

Besides, labels related to the segmentation method (or algorithm) are also provided (ProCancer-i), such as *Manual*, *Semiautomatic*, and *Automatic*. These labels are represented as standard concepts from SNOMEDCT and specified as *qualifier values* in the *common Module* (see Figure 12).

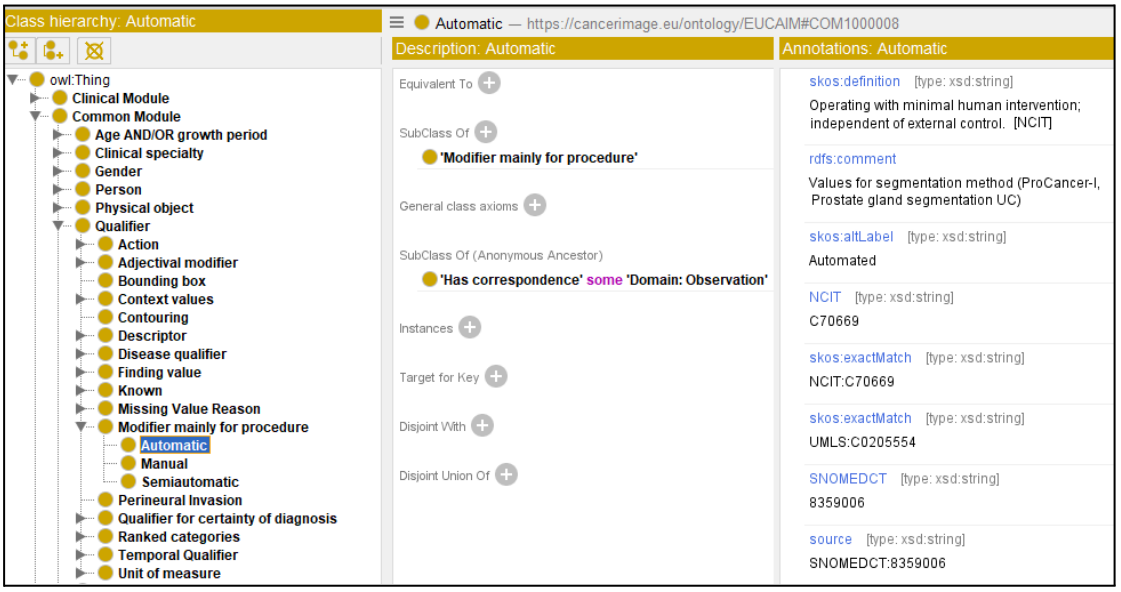


Figure 12. Part of the hyper-ontology around *Automatic* (v1.0) represented in Protege

- **DICOM tags**: The DICOM tags represented in standard terminologies/ontologies, such as RadLex, are considered in the hyper-ontology. We previously mentioned the example of *imaging modality* (Figure 9 and Table 5). However, for the tags not found in standard resources, such as *segment label* (0062,0005) and *SeriesDescription* (0008,103E), their explicit representation in the hyper-ontology depends on the decision of the *required imaging metadata*. Nevertheless, the values associated with some of these DICOM tags and needed in the image annotation/segmentation tasks are specified in the hyper-ontology (see *annotation labels/values* Figures 11 and 12, and Table 6).

The DICOM attributes available in the EUCAIM Hyper-ontology (version 1.0 *in progress*) are summarised in Table 5. In Table 6, we present the DICOM attributes, which are not explicitly defined in the hyper-ontology, but their values are specified.

Table 5. DICOM tags represented in the EUCAIM hyper-ontology (version 1.0)

Attribute name	ID	Type	Vocabulary source ID	EUCAIM Concept ID
Patient Position	(0018,5100)	Optional (3)	RADLEX: RID10420	IMG1016605
Body Part Examined	(0018,0015)	Optional (3)	SNOMEDCT:52530000	BP1000024
Manufacturer	(0008,0070)	Optional (3)	NCIT:C25392	IMG1000010
Modality	(0008,0060)	Required (1)	SNOMEDCT:360037004	IMG1000009
Laterality	(0020,0060)	Conditionally Required, Empty if Unknown (2)	RADLEX:RID5821	IMG1016305
Patient Orientation	(0020,0020)	Conditionally Required, Empty if Unknown (2C)	RADLEX: RID10461	IMG1016610
Slice thickness	(0018,0050)	Required, Empty if Unknown (2)	RADLEX:RID28669	IMG1016306
Echo time	(0018,0081)	Required, Empty if Unknown (2)	RADLEX:RID12463	IMG1016641

Table 6. DICOM tags whose values are represented in the EUCAIM hyper-ontology (version 1.0)

Attribute name	ID	Type	Examples of Values	Vocabulary source ID	EUCAIM Concept ID
Segment label	(0062,0005)	Required (1)	TZ (Transition Zone of prostate), CZ (Central Zone of prostate), PZ (Peripheral Zone of Prostate)	RADLEX:RID351, RADLEX:RID348, RADLEX:RID347	BP1000100, BP1000168, BP1000006
Segment method/algorithm type	(0062,0008)	Required (1)	Automatic, Semi-automatic, Manual	SNOMEDCT:8359006, NCIT:C172484, SNOMEDCT:87982008	COM1000008, COM1000005, COM1000003
Segmentation Type	(0062,0001)	Required (1)	Binary	NCIT:C45969	COM1000023
Image Type	(0008,0008)	Optional (3)	Primary, Axial	SNOMEDCT:63161005, SNOMEDCT:24422004	COM1000017, COM1000018

Annotation pathways

After data preparation by Data Holders at the local level, data can be ingested into the central node. At this point, different annotation scenarios emerge.

Scenario I. Manual annotation.
The annotation can be performed manually using interactive applications within the EUCAIM central node. One of these tools is the **Quibim DICOM Web Viewer**, which offers image interaction and segmentation capabilities. After loading the DICOM imaging series into the viewer, manual annotations can be performed from scratch, and the resulting annotations are stored in DICOM SEG format.

Scenario II. Semi-automatic annotation.
In this case, to avoid segmentations from scratch, an automatic segmentation tool is executed from the viewer. The intention is that the segmentation tools collected by the

EUCAIM partners are integrated into the viewer and can be run from it to provide the clinicians with a pre-segmentation. This way, the time cost is drastically reduced since the medical professional only needs to correct and refine the automatic annotation. It is worth noting that the automatic segmentation is executed on a case-by-case basis.

Scenario III. Automatic annotation.
To perform automatic segmentations in batch, it is necessary to execute the annotation tool in a dedicated environment rather than through the viewer. Two scenarios can be distinguished:

1. **Preprocessing Step**: In this scenario, the automatic annotation serves as a preprocessing step within an analysis pipeline. Here, the annotation itself is not the primary objective but rather a preliminary task required as input for subsequent tools or models. Annotations obtained in this scenario are not persistent.
2. **Batch Inference**: The final aim here is to generate annotations in batch by making inferences on a dataset. These annotations can be reviewed and refined by a medical professional, after which they could become persistent.

Annotation tools

The annotation tools presented in the demo are summarised in this Section.

Multi-regional prostate segmentation tool (Quibim)[14]
The tool performs an automatic multi-regional segmentation of the prostate into central-transition zone (CZ+TZ), peripheral zone (PZ), and seminal vesicle (SV) using a T2W MRI image, as shown in Figure 13. A heterogeneous database of 243 T2W prostate studies was used to train a U-Net-based model with deep supervision. Further information can be found in D5.4 Supplementary Material.

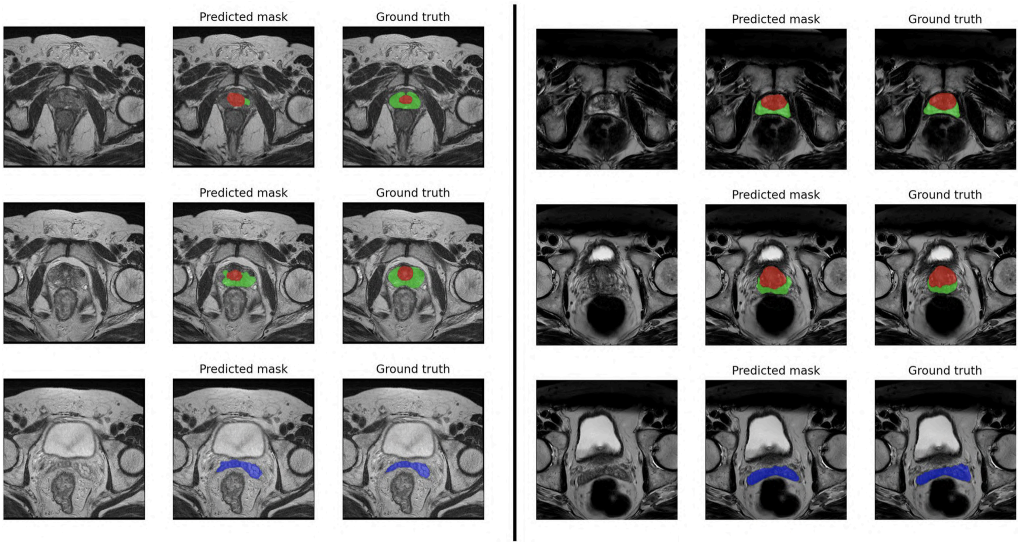


Figure 13. Multi-regional prostate segmentation results in two different subjects (left and right blocks) at three different anatomical levels (rows). For each case the original image (left), the predicted mask (middle) and the ground truth (right) are shown.

[14] Jimenez-Pastor, A., Lopez-Gonzalez, R., Fos-Guarinos, B., Garcia-Castro, F., Wittenberg, M., Torregrosa-Andrés, A., Marti-Bonmati, L., Garcia-Fontes, M., Duarte, P., Gambini, J. P., Bittencourt, L. K., Kitamura, F. C., Venugopal, V. K., Mahajan, V., Ros, P., Soria-Olivas, E., & Alberich-Bayarri, A. (2023). Automated prostate multi-regional segmentation in magnetic resonance using fully convolutional neural networks. *European radiology*, *33*(7), 5087–5096. https://doi.org/10.1007/s00330-023-09410-9

Breast dense tissue segmentation (ITI)[15]

The tool performs automatic segmentation of the breast area and the dense tissue in digital mammograms, as shown in Figure 14. The model was trained with a heterogeneous database of 2496 mammograms obtained from 11 different centres. The trained model (CM-YNet) returns both the dense tissue mask and the breast tissue mask. Further information can be found in D5.4 Supplementary Material.

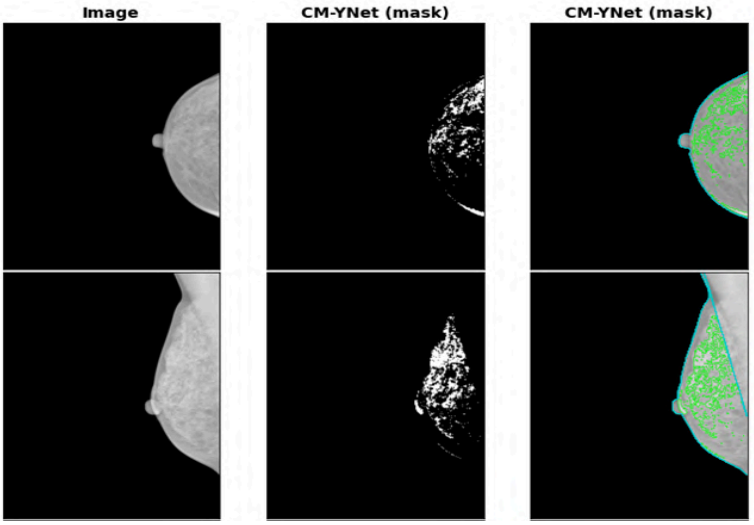


Figure 14. Breast dense tissue segmentation in digital mammograms in two different subjects (top and bottom). For each case the original image (left), the predicted mask (middle), and the predicted mask overlapped on the original image (right) are shown.

In addition, although not shown in the demo, the following tools have been compiled and will be integrated into the platform in the future. They extend the segmentation use cases and include object detection capabilities.

- **MR-based neuroblastoma tumour detection and segmentation**[16]. The tool performs an automatic segmentation of the possible neuroblastoma tumours on T2W or T2W Fat Sat MRI images in DICOM. The output can be provided in DICOM SEG format or NIfTI.
- **MR-based DIPG tumour detection and segmentation**. The tool performs an automatic segmentation of the possible DIPG tumours on T1W and/or T2W/FLAIR MRI images in DICOM. The output can be provided in NIfTI.
- **MR-based glioblastoma tumour detection and segmentation**[17]**.** The tool performs an automatic segmentation of the possible glioblastoma tumours on contrast-enhanced T1W, T2W, and FLAIR MRI images and its subregions. The inputs and outputs are in NIfTI.

---

[15] Larroza, A.; Pérez-Benito, F.J.; Perez-Cortes, J.-C.; Román, M.; Pollán, M.; Pérez-Gómez, B.; Salas-Trejo, D.; Casals, M.; Llobet, R. Breast Dense Tissue Segmentation with Noisy Labels: A Hybrid Threshold-Based and Mask-Based Approach. *Diagnostics* **2022**, *12*, 1822. https://doi.org/10.3390/diagnostics12081822

[16] Veiga-Canuto, D., Cerdà-Alberich, L., Sangüesa Nebot, C., Martínez de Las Heras, B., Pötschger, U., Gabelloni, M., Carot Sierra, J. M., Taschner-Mandl, S., Düster, V., Cañete, A., Ladenstein, R., Neri, E., & Martí-Bonmatí, L. (2022). Comparative Multicentric Evaluation of Inter-Observer Variability in Manual and Automatic Segmentation of Neuroblastic Tumors in Magnetic Resonance Images. *Cancers*, *14*(15), 3648. https://doi.org/10.3390/cancers14153648

[17] Beser-Robles, M., Castellá-Malonda, J., Martínez-Gironés, P. M., Galiana-Bordera, A., Ferrer-Lozano, J., Ribas-Despuig, G., Teruel-Coll, R., Cerdá-Alberich, L., & Martí-Bonmatí, L. (2024). Deep learning automatic semantic segmentation of glioblastoma multiforme regions on multimodal magnetic resonance images. International journal of computer assisted radiology and surgery, 10.1007/s11548-024-03205-z. Advance online publication. https://doi.org/10.1007/s11548-024-03205-z

- **CT-based neuroblastoma tumour detection and segmentation.** The tool performs an automatic segmentation of the possible neuroblastoma tumours on contrast-enhanced CT images. Outputs can be provided in DICOM SEG format or NIfTI.
- **nnUnet**[18]. The tool performs semantic segmentation allowing both training and inferencing. Includes many pre-trained models such as cardiac MRI, abdominal MRI, CT thorax, and CT Liver. Output segmentations are provided in DICOM SEG.
- **nnDetection**[19]. The tool automates the configuration process for medical object detection. The resulting self-configuring method adapts itself without any manual intervention to arbitrary medical detection problems.
- **MITK**[20]. This application features an interactive user interface for image analysis, incorporating segmentation tools ranging from traditional manual options to advanced methods like GrowCut, TotalSegmentator, SegmentAnything, and nnUnet. It also includes a suite of command-line tools for automating basic image processing tasks such as file conversion, registration, stitching, and resampling. It handles different formats such as DICOM, NIfTI or NRRD.

Data Harmonisation

Medical data harmonisation involves preprocessing steps that are essential for standardising and integrating diverse datasets, ensuring their compatibility and comparability. This process is critical for both imaging data (e.g., MRI, CT scans) and numerical data (e.g., lab results, clinical measurements, imaging features) to enable reliable and reproducible analyses across different sources and studies.

Several applications can benefit from data harmonisation such as multi-center studies, longitudinal research, machine learning analysis and/or personalised medicine.

- In multi-center studies, harmonisation enables the combination of data from different institutions, increasing sample size and statistical power[21].
- Longitudinal research benefits from consistent data collection over time, which is crucial for tracking disease progression and treatment outcomes[22].
- High-quality, standardised datasets are indispensable for training and validating machine learning models[23].
- In personalised medicine, harmonisation facilitates the integration of diverse data types (e.g., genomic, imaging, clinical), enabling tailored treatments for individual patients[24].

---

[18] Isensee, F., Jaeger, P.F., Kohl, S.A.A. *et al.* nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods* 18, 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z

[19] Baumgartner, M., Jäger, P. F., Isensee, F., & Maier-Hein, K. H. (2021). nnDetection: a self-configuring method for medical object detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24* (pp. 530-539). Springer International Publishing.

[20] https://www.mitk.org/wiki/The_Medical_Imaging_Interaction_Toolkit_(MITK)

[21] Ducharme, S., et al. (2017). Addressing Heterogeneity in Multicenter MRI Studies: Data Harmonization Strategies. *FronTiers in Neuroscience, 11*, 632.

[22] Fortin, J.-P., et al. (2017). Harmonization of Multi-Site Diffusion Tensor Imaging Data. *NeuroImage, 161*, 149-170.

[23] Johnson, K. A., et al. (2017). The Role of Data Harmonization in Big Data Analytics for Precision Medicine. *Journal of Biomedical Informatics, 70*, 37-50.

[24] Pomponio, R., et al. (2020). Harmonization of Large MRI Datasets for the Study of Brain Aging: A Comparison of Methods. *NeuroImage, 208*, 116450.

Goals

Data harmonisation is a broad term that is sometimes used in different domains with several meanings, such as protocol standardisation, data normalisation, artefact correction, data transformation, and integration techniques.

In this specific task, we have focused our efforts on providing advanced statistical and machine learning techniques to harmonise datasets that potentially originated from different populations, acquisition protocols and/or devices. This ensures their comparability and enables more accurate downstream analysis. To achieve this, recent contributions and research conducted by the partners about data harmonisation, such as those from the AI4HI EU projects, were proposed, filtered, evaluated, and integrated into the EUCAIM project.

Requirements

We have provided two different categories of data harmonisation tools according to their input data. Thus, we divided them into:

- Methods based on medical images (e.g. MRI, CT)
- Methods based on tabular data (i.e., radiomic features, clinical data)

To this end, the accepted input format for such methods was DICOM for raw images and CSV files for numerical features. These methods were containerized in docker images in order to be accessible, reusable and interoperable within the EUCAIM architecture/platform. The containerized tools are expected to have at least two parameters: one specifying where to find the input data, and another specifying where to retrieve the results. All the provided tools were expected to meet and comply with the Tier-3 level specifications outlined by EUCAIM.

Subtasks

Along with the development and integration of the data harmonisation tools into the EUCAIM platform, three complementary and transversal subtasks were addressed in the scope of this task.

NIfTI to DICOM. A conversion procedure between common medical image formats such as NIfTI and DICOM was discussed and resolved. Originally, some tools were designed for functioning with NIfTI format and therefore an additional pre/post process was required to be included in the tools for expanding their compatibility with the EUCAIM default imaging format.

Traceability for data harmonisation. To enable study reproducibility, all processes should be traced and this involves data preprocessing methods as well. Few alternatives were discussed in this subtask.

- One option was editing the resulting harmonised DICOM metadata. However, DICOM itself does not have specific tags for image processing parameters or algorithms but it does allow the use of private tags to store additional information that may not be covered by the standard. A private tag is identified by having an odd group number, e.g., (0051,xxxx) and they can be created and used within a specific context or application. Hence, the tag headers could describe the order of application

and the specific parameters used for each process. However this approach may lack standardisation.

- Alternatively, this annotation process could be better performed using DICOM Structured Reporting (SR), or also by means of using some DICOM viewers to support the addition of annotations or comments.
  The feasibility and usability of these and other options will be discussed and considered with other work package tasks in order to reach some consensus.

Tools evaluation risks. Due to the need of containerizing and incorporating these tools into a central platform, we anticipated some potential sources of errors to be addressed by such tools. Hence, we distinguished between tool errors (e.g., image /dimension mismatch, data folder nonexistent) and docker specific errors (e.g., out of memory, denied permission). Further efforts in testing and evaluating the tools will be performed in the next stages of this task.

Tools

In this section, we describe the harmonisation tools that are presented in the demo. Further conceptual, technical and usage details of these tools can be found in the D5.4 Supplementary Material.

Biologically motivated intensity normalisation techniques (FORTH). The tool is designed to perform normalisation at the image level. This normalisation method aims to reduce the variability in the intensity values of MRI prostate images due to different scanners, acquisition protocols and conditions, based on the intensity values of specific tissues. This tool implements three biologically-motivated intensity normalisation techniques: (1) The fat-based normalisation method, (2) The muscle-based normalisation method, and (3) The single tissue (fat or muscle) piecewise normalisation method. This tool uses as input T2W prostate MRI data. Further information can be found in D5.4 Supplementary Material.

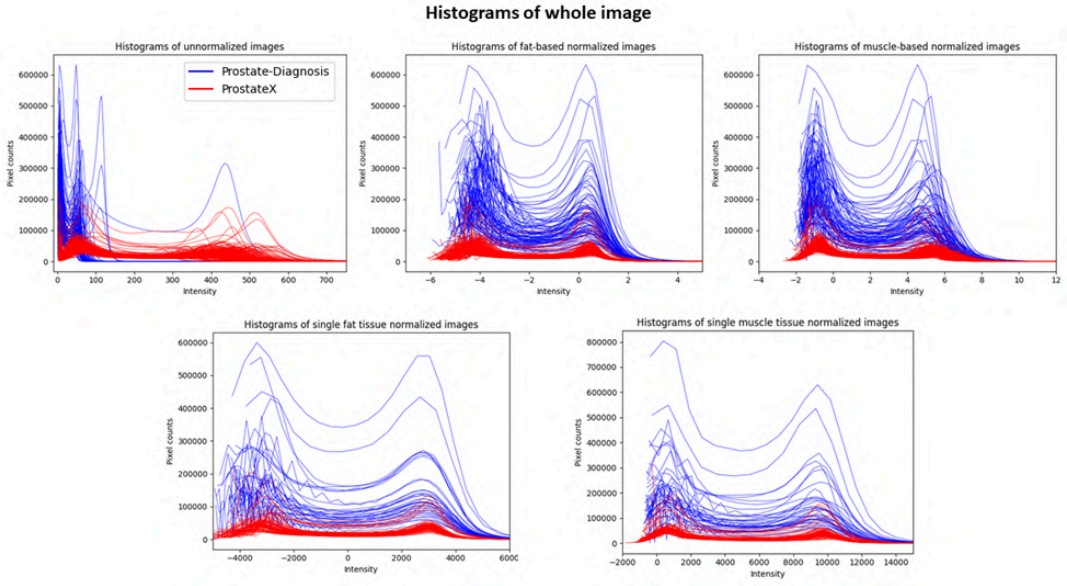


Figure 15. Example results of the Biologically motivated intensity normalisation tool showing the histograms of pixel intensities at diagnosis (blue) and after normalisation (red).

Image Intensity harmonisation (QUIBIM). The tool is effective in the harmonisation of intensity dynamic ranges in MRI and is backed by AI. The variability in intensity that results from different acquisition protocols and scanners difficults the performance of AI tools and global information computation. The methodology can be applied to MRI samples and it's based on self-supervised learning that leverages the use of MRI frequency domain to synthetically generate contrast variations in reference images by making subtle changes in specific frequencies and then transforming the image back to the spatial domain where the images have its original content with altered intensities. These synthetically generated paired images are finally used to train an autoencoder based model with harmonisation purposes on prostate MRI. This tool uses T2W prostate MRI. Further information can be found in D5.4 Supplementary Material.

Figure 16. Example results of the Image Intensity harmonisation tool showing the images before harmonisation (centre) and after harmonisation (right). The images on the left are the reference images used for the normalisation.

Feature-based harmonisation (FORTH). The tool is designed to perform harmonisation at the feature level. Feature-based harmonisation method aims to reduce the variability in the radiomics features due to different scanners, acquisition protocols and conditions by using empirical Bayesian methods to estimate differences in radiomics values and then expressing them in a common space (location/scale adjustment). The tool offers two methods: (1) ComBat method, which shifts the radiomics features to the overall mean and pooled variance of all centres, and (2) M-ComBat method, which shifts the radiomics features to the mean and variance of the chosen reference centre with the most samples. This tool uses numeric variables (e.g. radiomic features). Further information can be found in D5.4 Supplementary Material.

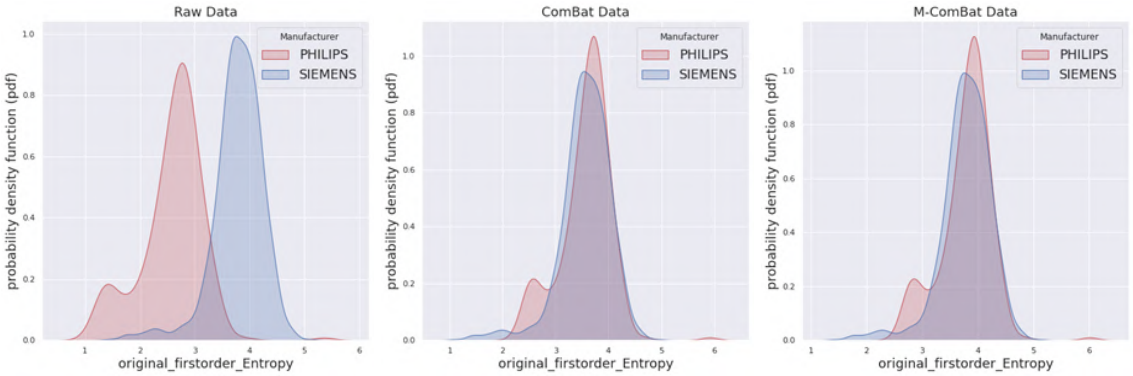


Figure 17. Example results of the Featured-based harmonisation tool showing the probability density function of radiomic features from different providers before (left) and after normalisation using the ComBat method (centre) and M-ComBat method (right)

***Trace4Harmonization*™** (**DeepTrace Technologies**, subcontractor of IRCCS Policlinico San Donato). This tool aimed at harmonising numerical features, including (but not limited to) potential imaging biomarkers such as features extracted from medical images that were acquired under different conditions (e.g. different acquisition systems or different acquisition protocols). More specifically, the aim is twofold: the first is related to calibrating a harmonisation model based on a dataset of unharmonized samples; the second is the application of the calibrated model to new samples. The tool has as input data numeric variables (e.g., radiomic features extracted from medical images, with no limitation on the data modality from which the features are extracted). Further information can be found in D5.4 Supplementary Material.

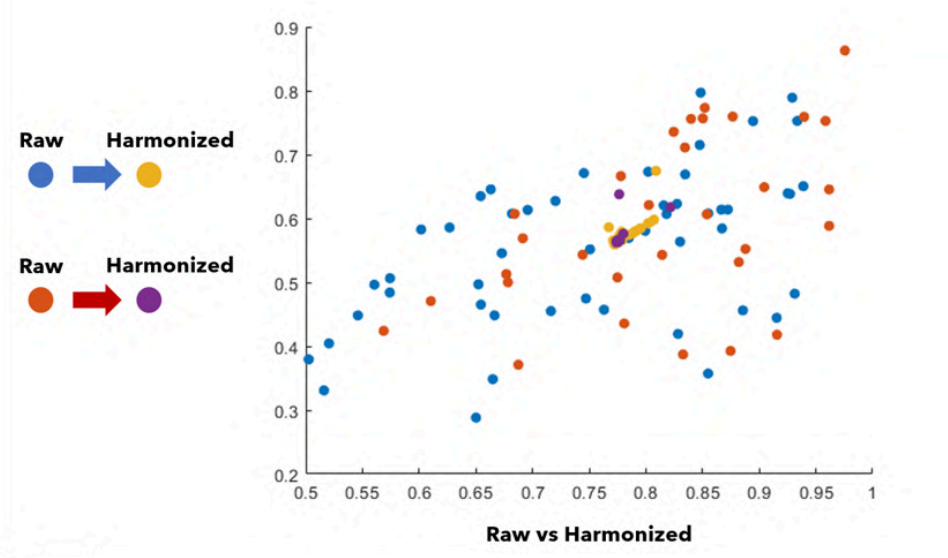


Figure 18. Example results of the Trace4Harmonization tool, showing raw vs harmonised data.

D. Tools incorporation

Minimum requirements

General requirements

Imaging DICOM format

EUCAIM's standard format for imaging data is DICOM. As mentioned earlier, this standardised format provides useful information on the acquisition, that can further be queried as well as used for advanced analyses.

That said, some preprocessing tools may only apply to NIfTI (see above; some harmonisation tools), a format that provides less metadata to sift through and further analyse but is often found easier to handle by researchers. To avoid excluding any valuable tool from the catalogue, while keeping DICOM as the mandatory input format, the following recommendations are made to tool developers:

1. The tool should accept DICOM files as input, and generate (when applicable) DICOMs as output.
2. If the tool does not support the DICOM format, the tool providers should evaluate whether it is possible to adapt their tool to comply with this requirement.
3. If the tool does not support DICOM format and no adjustment can be made, for example, due to limited project resources or technical difficulties, a DICOM to NIfTI converter may be added as a module to the tool. Alternatively, a DICOM to NIfTI converter available in the marketplace can be used. This converter may also be used to convert NIfTI outputs back to DICOM.

In addition, as detailed in the section above on annotations, some DHs may only have available annotations in NIfTI format. In such a case, a tool will allow conversion from NIfTI and other non-standard formats to DICOM SEG format.

Compliance with EUCAIM CDM

Full compliance with EUCAIM CDM is required for Tier 3 data and is addressed at the preprocessing stage. In the context of federated processing/analysis, and enabling the ambition of accepting various input formats for clinical datasets provided by data providers, an ETL process will be deployed on-site, to allow for structured clinical datasets to be 1) ingested and harmonised, 2) transformed to EUCAIM CDM, and 3) shared with authorised users.

The first preprocessing step normalises and unifies these formats into a tabular format that will be aligned with the expected templates for each type of dataset. This normalisation is necessary because experience shows that, particularly with highly flexible and open formats like XLS, it is often essential to correct issues where data do not conform to their column definitions. This can be resolved, broadly speaking, in two ways:

- Using different automatic preprocessing methods, which include techniques with varying levels of risk. Here, "risk" refers to the potential for introducing bias or noise into the dataset, which may differ from the original intent of the experts.
- Combining automatic preprocessing with manual review. Therefore, a tool with a user interface will be available to present the data points that require such review, possibly offering suggestions from the automatic preprocessing.

In a way that imposes the least possible workload on both data providers and the hypothetical team adapting the ETL for a specific case, a harmonised output is produced without issues for processing and transformation into the CDM. This output will be annotated with additional metadata regarding the scope, size, type and risk of corrections made.

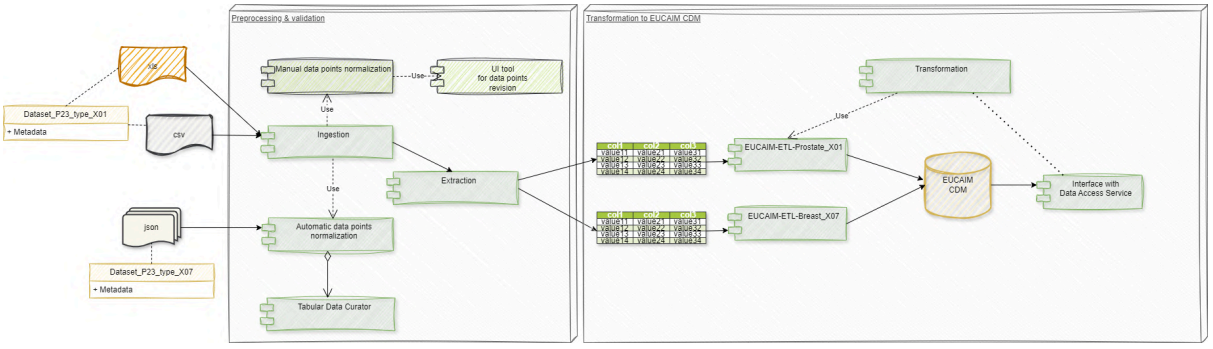


Figure 19. High level workflow of ETL process within EUCAIM

With this information, the extraction node will be able to deploy on demand and execute a specific component for that type of dataset (already normalised) and transform it into the CDM based on the EUCAIM hyper-ontology, also storing the metadata.

For greater extensibility of the ETL, there will be a generic component for its transformation to the EUCAIM CDM and an extensible library of specific components for extraction and pre-transformation to tabular format according to dataset typology and other characteristics. Finally, a small component acts as an external interface in this node. A detailed description of the clinical data ingestion and transformation workflow with the ETL will be provided in the next deliverable (D5.5) at month 24.

Validated & Approved by EUCAIM partners

Before being included in the final catalogue, each tool must undergo a thorough approval process by the EUCAIM Technical Committee. This committee rigorously evaluates the tool's usefulness, functionality, and security. The evaluation is based on detailed information provided during the tool's validation phase. Only tools that meet the committee's criteria are approved and added to the final catalogue, ensuring high standards of quality and reliability.

Requirements for tools running in the central/federated nodes

Several analysis platforms that support the execution of containerised tools for federated learning and distributed analysis were proposed, including *jobman*, OpenVRE, and VIP. At the moment, each platform has its own containerisation specifications, however, work is being done to standardise this process across all of them.

For demonstration purposes, the ChAImeleon environment was used, as the EUCAIM central platform will be inspired by ChAImeleon. In this setup, jobman is the tool executor. To include a tool in the platform's internal repository so it can be visible and used by all the users, it needs to be dockerized according to specific guidelines[25]. The ChAImeleon platform also allows interactive tools that require a GUI or web UI. For these tools, a desktop environment needs to be installed, which may include a web browser if necessary. Additionally, a Helm chart must be created. Once this chart is uploaded to the charts repository, a new application will be available to deploy a remote desktop.

Validation framework

The validation framework created within WP5 is the first approach to validate internally the selection, and integration of the preprocessing tools included within Task 5.3: preprocessing tools in the EUCAIM platform. Adhering to this proposed validation introduces an additional layer of validation, enhancing our ability to monitor, anticipate, and assess the correct delivery of the tools for subsequent tasks. The framework will be extended together with WP6 and WP7 to further ensure that the processing of data respects data privacy, data integrity, infrastructure privacy, and infrastructure integrity. Additionally, this process will be extended to new tools that are incorporated into EUCAIM in the future.

The validation framework comprises three main stages: conceptual validation, technical validation, and integration validation. Each stage involves documentation, along with a presentation or demonstration conducted for experts in the topic or, in this case, members of the corresponding Focus Group (FG). The role of the validator is introduced; one or more partners that oversee the progression of the tool through each validation step. The validation process is iterative, emphasising a cyclic and incremental approach to ensure thorough validation of the tools. This iterative nature allows for continuous refinement, adjustment, and feedback incorporation, fostering an ongoing improvement cycle.

Additionally, before their inclusion in the final catalogue of tools, each tool undergoes approval by the EUCAIM Technical Committee, which will ensure their compliance with the quality and security requirements. Once the tool is finally approved to be part of the EUCAIM platform, it will be registered in ELIXIR bio.tools.

Table 7. Overview of the different phases of the validation process

Validation	Goal	Documentation	Demo
Conceptual	Tool aligns with EUCAIM purpose	Goal, task, data, thorough definition of input/output…	Presentation to the FG
Technical	Tool is technically well-prepared for EUCAIM platform	Methodology description, hardware requirements, installation instructions….	Demo/video of tool running on the platform
Integration	Tool is correctly integrated into the EUCAIM infrastructure	Communication channel, common errors, FAQs, tool usage	

[25]https://github.com/chaimeleon-eu/workstation-images#how-to-design-a-workstation-image-for-the-chaimeleon-platform

Validator

The validator is a role assigned to one or more partners who will be responsible for reviewing the documentation and presentation/demonstration of a given tool. This partner(s) will make sure that all the requirements defined in each of the steps of the validation process included in this framework are followed and accomplished. As validators, their feedback will also help tool owners make the appropriate changes to deliver a tool that tailors the EUCAIM standards.

Selection criteria

The selection of the validator/s per tool will be defined under the corresponding Focused Group regarding each topic. The distribution of efforts by the partners should be equilibrated. The selection will be a volunteer process, in which any other partner different from the tool owner can be eligible. In the case multiple partners or none want to validate a tool, the task leader will suggest the eventual validator according to the background and affinity criteria of the candidate.

Validator tasks

The role of the validator in the different validation stages is to enssure that both artefacts (documentation and demo) are done by the tool provider, as well as to assure their quality and completeness. Each time a validation stage is accomplished, the validator will oversee adding the corresponding outputs generated by that process in a shared repository. FG leaders will create these folders for each of the tools.

Validation stages

The validation may include an initial pre-selection of tools depending on the aim of the validation. For example, a pre-selection for a given deliverable or according to the maturity level of the tool. For instance, some criteria could be if they are already dockerized, if they have been used by other users or in other projects, if they have some peer reviewed publication (e.g. in some journal or international conference) and/or if the code is available in some open repository.

Conceptual Validation

The conceptual validation aim is to ensure that a tool is aligned in its purpose and functionality with the preprocessing tools specified in T5.3. For this, each tool provider will perform two different artefacts: documentation and an oral presentation of the tool. The first **clearly explains the necessity of the tool, and what and how it does it**. The oral presentation helps the rest of the FG partners to understand the tool better as well as to provide feedback. The main content of the documentation that will be required by each tool includes:

- **Name:** name of the tool.
- **Contributor:** partner providing the tool.
- **Area**: annotation/de-identification/quality/harmonisation/FAIRification
- **Tool description:** a concise overview of what the tool does, its primary purpose, and specific task(s) that the tool is designed to perform, such as segmentation, harmonisation, classification, etc.
- **Data:** description of data the tool is aimed at; image modality, cancer type, series...

- **Methodology/performance:** brief description of the methodology employed for the development of the tool; methods and architecture, preprocessing…. (in depth description will be performed in the technical validation).
- **Use:** brief description of the tool's functioning (if it applies).
- **Input/output formats:** description of the input/output of the tool at the validation stage(will help if some adaptation is needed).
- **Quantitative results**: performance obtained during training of the tool (if it applies)
- **Qualitative results:** provide some visual results (if available) of applying such tools.
- **Additional information:** successful use cases, external resources (open code, papers...) licence, certification… (if they apply).

An oral presentation is performed by the provider of the tool within the corresponding FG. The content of this presentation will be mainly a summary of the above points. The result of this presentation will be a video recording together with the slides of the presentation.

Technical Validation

The technical validation aim is to collect the technical specifications of a given tool and prepare the tool for its integration into the EUCAIM test environment. This may include, among others, some modification in the input/output, or the inclusion of monitoring mechanisms. In this stage, the tool providers should provide the following documentation:

- **Data**: in depth description of the data used to train and validate the tool.
- **Methods:** in depth description of the methodology used for its development including all data preprocessing.
- **Specific Technical information:**
    - CPU/GPU
    - Programming language
    - Expected RAM usage.
    - Running mode (interactive/batch-based/case-based…)
    - Software version
    - Libraries
    - Security measures: does the tool require administrator privileges?

- **Traceability** and **monitoring** mechanism.
- **Unitary tests**: description of the tests implemented to verify the correct functioning of the tool.
- **Access restriction:** Do you have any access restrictions to the source code or to the binaries of your tool?
- **Containerization:** Is the tool already dockerized?
- **Additional information** for tool integration.

Integration Validation

In this stage, further documentation is required by the tool providers. In particular, the following important points are suggested to be described about the tools:

- **Communication channel** for the helpdesk: please provide an email address for a contact person who can be reached with any questions regarding your tool.
- **Most common errors**
- **FAQs**
- **User Manual**:
    - **Installation/configuration instructions** (only for downloadable tools)
    - **Usage instructions**

        - **Additional considerations**: Input/output description, if any preprocessing is needed, mandatory/optional data, and cases in which the tool should not be used.
    - **Integration tests**: Description of tests for assessing the correct integration of the tool
    - **Results of non-functional tests**

Finally, in this validation stage, a demonstration will be performed. For this, the tool should be integrated into the platform and a dataset should be available to test the tool. The idea of the demo consists of the developer running the tool on the platform with a test dataset.

Bio.tools registration

As a final step, the preprocessing tools will be registered in bio.tools (https://bio.tools/).

E. First demonstrator of the preprocessing tools

Demonstrator test environment

In this first demonstrator, the execution of the preprocessing tools is shown in two different environments. On the one hand, the current ChAImeleon platform and applications are used to simulate the central node, where developers are allowed to deploy Ubuntu remote desktops or Jupyter Notebooks (Figure 20) and have an environment to execute their tools, with the possibility of using datasets available in this project. The EUCAIM Central node processing environment will highly resemble the one from ChAImeleon. Thus, this was the selected environment for demonstrating the preprocessing tools in the central node. On the other hand, the execution of some tools is shown at each developer's premises, which resembles a local environment independent of the EUCAIM infrastructure. This simulates the preparation that Data Holders would need to perform locally before the data is ingested to EUCAIM.

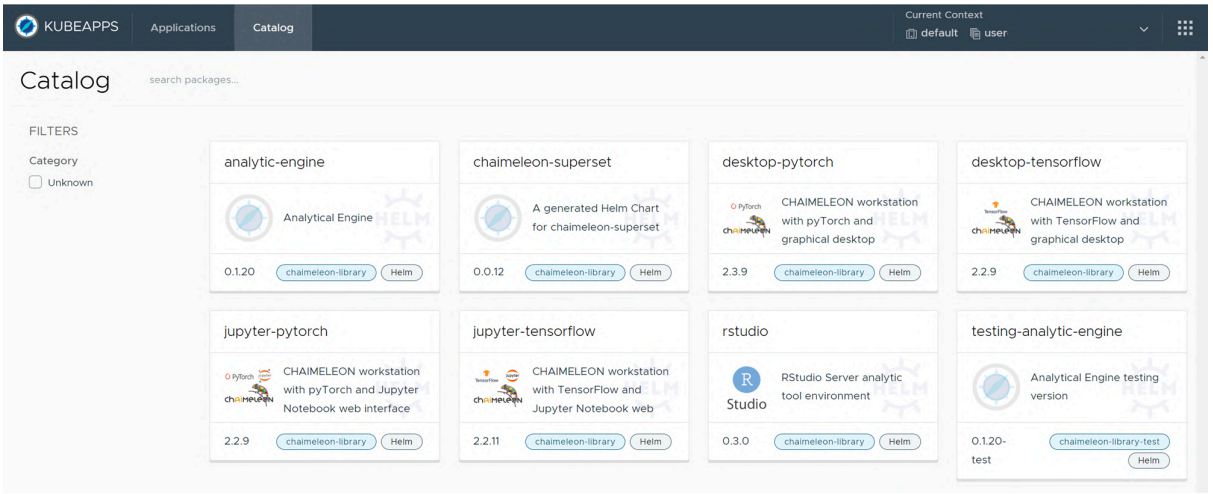


Figure 20. Overview of the applications available on the ChAImeleon platform[26]

The dockerization process of the tools has been different in each case. Although common guidelines were followed to create Docker images that could later be executed with Jobman tasks on the ChAImeleon platform[27]. This approach involved uploading the code to a public repository and using the project's Docker image registry, which has generated controversy among the developers who need to keep the source code private. Some alternatives to compile and obfuscate the source code in order to protect it were studied, but for the context of this demonstrator, other temporary alternatives with easier implementation were offered. Firstly, developers had the option to import Docker images into their workspaces (previously built and saved locally) and run uDocker containers on the central node, with user permissions controlled. Secondly, an alternative option was provided to share a private repository only with the ChAImeleon platform developers, allowing them to create the Docker

[26] https://chaimeleon-eu.i3m.upv.es/dataset-service (*)
[27] https://github.com/chaimeleon-eu/workstation-images#how-to-design-a-workstation-image-for-the-chaimeleon-platform
* Authentication required to access the environment

image and store it in a separate registry. Thus, the Jobman configuration was adapted so that only specific users were able to launch this Docker image on the central node.

Regarding the data used in this demonstrator, in the cases of the local environment, either data downloaded from public repositories or proprietary data were used, as only developers showing the tool have access to them. For the tools on the central node, it was possible to use datasets available in the ChAImeleon project (Figure 21) when the developers were already part of this project, whereas access needed to be manually granted for new users on the platform. Additionally, ChAImeleon does not contain numerical data. Consequently, the developers showcasing numerical-based tools had to upload small datasets to their workspace on the remote desktops deployed for the demonstration.

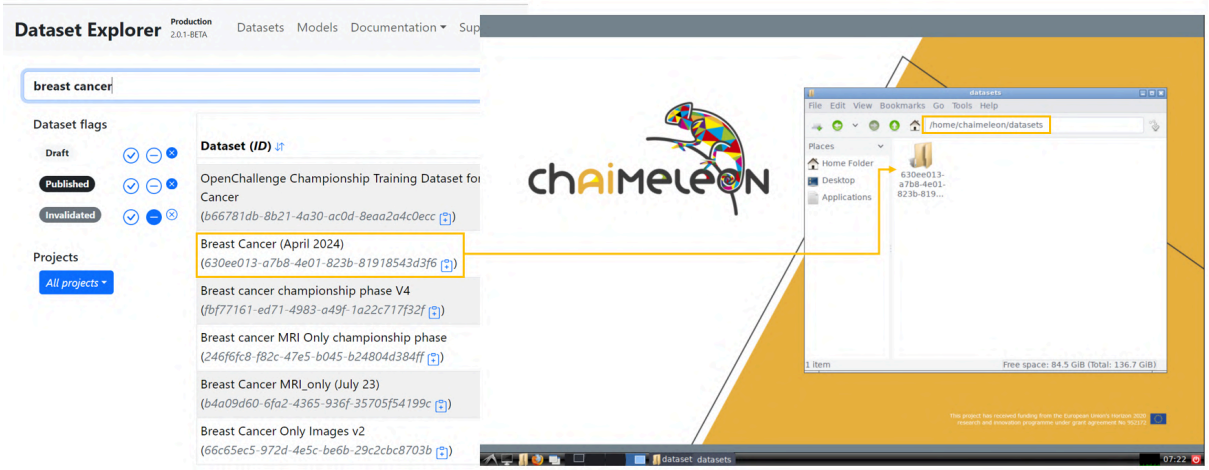


Figure 21. Example of the dataset used for the 2nd scenario of this demonstrator[28]

All these security measures have been meticulously considered to facilitate the demonstration of each preprocessing tool within controlled environments, ensuring that developers maintain the necessary level of control over their respective tools. These measures align seamlessly with the scenarios outlined below, where the corresponding videos were recorded.

Results

In this demo, we showcased three scenarios that resemble the preprocessing of data within EUCAIM. Within the first scenario, we show how T2W prostate MRI data is prepared, first by the data holder in a local environment and later in the central node, by a researcher before it is used for the development of an AI algorithm. In a similar way, the second scenario shows the preprocessing of mammography data. Finally, the third scenario shows how Fairness of data will be assessed within EUCAIM.

Scenario 1: Preprocessing of prostate MRI data

The first scenario illustrates the pre-processing of prostate T2-weighted MRI. Seven tools are demonstrated: two at a local node, simulating how data holders curate data locally, and

[28]https://chaimeleon-eu.i3m.upv.es/dataset-service/datasets/630ee013-a7b8-4e01-823b-81918543d3f6/details) and its availability within the remote desktop.

five at the test EUCAIM central node, showing how a data user prepares the data for training or validating an AI algorithm.

At the local node, the data holder initially assesses the data quality using the **DICOM File Integrity Checker**. This tool can be employed prior to anonymization to detect corrupted files. Subsequently, the data is anonymized using the **EUCAIM DICOM Anonymizer** before being transferred to the central node.

At the central node, the data user/researcher (DU/R) applies five tools to preprocess the data for AI algorithm training. First, the DU/R uses the **N4 bias filter tool** to reduce MRI inhomogeneities. Then, two normalization tools are applied to minimize variability introduced by different scanners and acquisition protocols: the **Biologically motivated normalisation** technique and the **Image intensity harmonisation**. In this example, the DU/R extract radiomic features from the prostate lesions for the training of an AI model. To segment the lesions, he/she employs the **MRI-based prostate segmentation** method. Finally, the Trace4harmonization tools is used to further harmonize the radiomic features.

Table 8. Overview of the tools used in the 1st scenario of the demonstrable.

Data quality	DICOM file integrity checker	HULAFE	Local
De-identification	EUCAIM DICOM Anonymizer	FORTH	Local
Data export from local environment to central node			
Data quality	N4 bias filter	FORTH	Central
Harmonisation [image-based]	Biological motivated normalisation technique	FORTH	Central
	Image intensity harmonisation	Quibim	Central
Annotation	MRI-based prostate segmentation	Quibim	Central
Harmonisation [numeric-based]	Trace4harmonisation	DeepTrace	Central

DICOM file integrity checker

The video demonstrates the procedure for running the **DICOM File Integrity Checker** tool in a local environment. In this case, the input consists of several patients with prostate MRIs, with completely unstructured folders where DICOM files are not separated by series or studies. The tool is executed by running a Docker container and mounting the volumes needed for input, output, and configuration. Two use cases are demonstrated: passing all MRI sequences or selecting only T2 sequences based on a list of known names used by different manufacturers that are specified in a catalogue. The results include a structured and clean dataset, as well as reports generated in various formats, where there is information about the included and excluded sequences for each case, the presence of some corrupted files, the concordance between the images per sequence expected and the

present ones i.e., detecting missing files, the merging of certain dynamic sequences and a summary of the present timepoints.

EUCAIM DICOM Anonymizer

The video demonstrates the process of anonymizing prostate cancer-related DICOM images using a patient case from the publicly available ProstateX[29] dataset. The anonymization procedure is performed locally by running the **EUCAIM DICOM Anonymizer** desktop application. In this scenario, the user provides a single patient folder containing one prostate MR DICOM Study and six different MR DICOM Series within the study. Upon execution, the tool prompts the user to select the patient's DICOM folder and then performs the anonymization based on the EUCAIM anonymization profile for the MR modality. Once the process is successfully completed, the user can inspect the anonymized results using the embedded MicroDicom DICOM Viewer to approve the outcome, or open the output folder containing the anonymized DICOM files and review them with their preferred software tool.

N4 bias filter

This video demonstrates the process of running the **N4 bias filter** tool on prostate MRI T2W cases using a remote desktop application in the ChAImeleon environment. Initially, the input path containing the dataset to be used is presented. The tool is executed by running the appropriate udocker command and mounting the necessary volumes for both the input and output of the tool. The execution takes approximately 8 minutes to complete for one T2W sequence. Upon completion, the derived N4 filtered images can be found in the specified output path. A slice of the original and the N4 filtered images are displayed side by side for visual comparison (Figure 22), along with a subtraction image (Figure 23) that highlights the differences in intensity values between the original and the N4 filtered images.

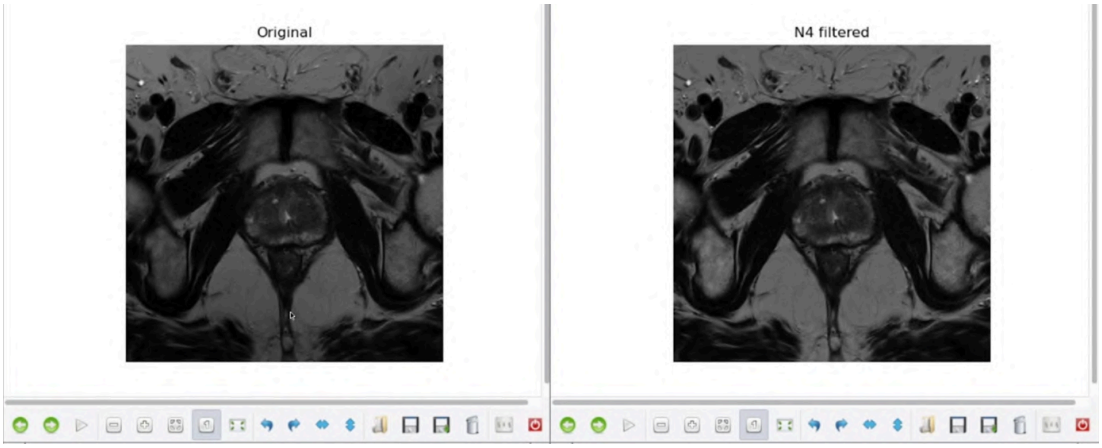


Figure 22: Output file displaying a slice of the original image (left) and the same axial slice of the N4 filtered image (right) side by side for visual comparison.

[29] Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., & Huisman, H. (2017). SPIE-AAPM PROSTATEx Challenge Data (Version 2) . The Cancer Imaging Archive. https://doi.org/10.7937/K9TCIA.2017.MURS5CL

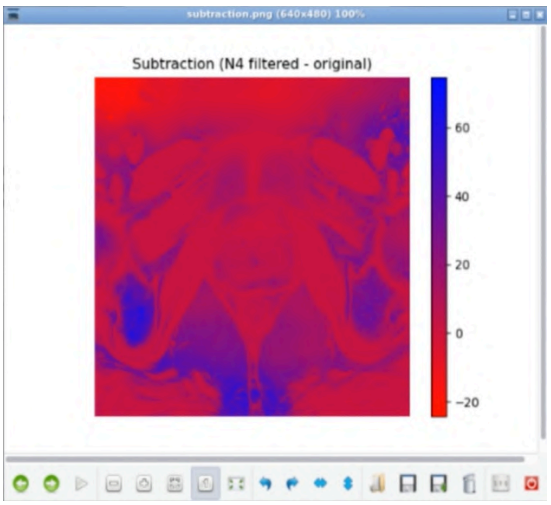


Figure 23: Subtraction image highlighting on an axial slice the differences between the original and the N4 filtered images, using coloured intensity values.

Biologically motivated normalisation technique

This video demonstrates the process of running the **Biologically motivated normalisation** tool on prostate MRI T2W cases using a remote desktop application in the ChAImeleon environment. In this scenario, the fat-based normalisation technique is selected for demonstration, as it is representative of the other available techniques. Initially, the input path containing the dataset to be used is presented. The tool is executed by running the appropriate udocker command and mounting the necessary volumes for both the input and output of the tool. The execution takes approximately 8 minutes to complete for one T2W sequence. Upon completion, the user can inspect the derived normalised images stored in the specified output path. A slice of the original and the normalised images are displayed side by side for visual comparison, along with histograms that highlight their overlap after applying the normalisation techniques.

Image intensity harmonisation

This video demonstrates the process of running the **Image intensity harmonisation** tool on a prostate MRI T2W case. An interactive notebook interface is used in this demonstration. First, the command for the tool is typed, including three parameters: the input path where the folder containing the DICOM files is located, the output path where the tool will write the harmonised DICOM files, and the mode (which, for this tool, is "Prostate"). Once the job is launched and completed, a qualitative comparison is shown between an input DICOM slice and its corresponding slice from the harmonised images.

MRI-based prostate segmentation

The video demonstrates the execution of the **multi-regional prostate segmentation** tool on a prostate T2W MRI using an interactive notebook. The process begins by loading a sample case from the prostate dataset and visualising it to display the tool's input. The tool is then run using *jobman*, which requires specifying the directory containing the DICOM files of the sample case and the output directory for storing the results. After the tool completes its execution, the resulting multi-label mask is loaded and overlapped on the input T2W image to evaluate the segmentation. The video shows three segmented regions: the central/transitional zone in red, the peripheral zone in green, and the seminal vesicles in

blue. Finally, the structure of the output DICOM SEG is reviewed, verifying the various tags associated with the segmentation.

Trace4harmonisation

This video demonstrates the process of running the **Trace4harmonisation** tool on using a remote desktop application in the ChAImeleon environment. Initially, the required input data, consisting of a CSV file, is used to execute the tool in calibration mode by running it with the appropriate udocker command and mounting the necessary volumes for both the input and output of the tool. Upon completion of the calibration process, the user can inspect the results such as a logs file, parameters from the calibration, and the harmonised calibration dataset. Then, the file containing the data to harmonise is used with the tool, this time in classification mode and having as input the results of the calibration phase. As a result, we obtain in the corresponding output folder the data already harmonised.

Scenario 2: Preprocessing of mammography data

The second scenario illustrates the preprocessing of mammography data using four tools. One tool operates at a local node, simulating how DH anonymize data before sending it to the central node. The other three tools operate at the test EUCAIM central node, demonstrating how a data user prepares data for training or validating an AI algorithm.

In this scenario, the data holder (DH) uses the **EUCAIM DICOM Anonymizer** to de-identify the data before uploading the data to the central node. At the central node, the data user/researcher (DU/R) employs the **DICOM File Integrity Checker** to select the relevant mammography series. Next, the **Breast Dense Tissue Segmentation Tool** is used segment the breast area and dense tissue. Radiomics are extracted from the dense tissue for trainin an AI Algorithm. As a last preprocessing step, the derived radiomic features are harmonized using the **Feature-Based Harmonization Tool**.

Table 9. Overview of the tools used in the 2nd scenario of the demonstrable.

De-id	EUCAIM DICOM Anonymizer	FORTH	Local repository
Export from local to central			
Data quality	DICOM file integrity checker	HULAFE	Central
Annotation	Breast dense tissue segmentation	ITI	Central
Harmonisation [numeric-based]	Feature-based harmonisation	FORTH	Central

EUCAIM DICOM Anonymizer

The video demonstrates the process of anonymizing breast cancer-related DICOM images using a patient case from the publicly available CBIS-DDSM dataset[30]. The anonymization procedure is performed locally by running the EUCAIM DICOM Anonymizer desktop

[30] Sawyer-Lee, R., Gimenez, F., Hoogi, A., & Rubin, D. (2016). Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM) [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2016.7O02S9CY

application. In this scenario, the user provides multiple patient folders containing multiple mammography DICOM images. Upon execution, the tool prompts the user to select the folder that contains all the patient DICOM files to be anonymized in a batch mode, and then performs the anonymization based on the EUCAIM anonymization profile for the mammography modality. Once the process is successfully completed, the user can inspect the anonymized results using the embedded MicroDicom DICOM Viewer to approve the outcome, or open the output folder containing the anonymized DICOM files and review them with their preferred software tool.

DICOM File integrity checker

The video demonstrates the procedure for running the DICOM File integrity Checker tool on the central node, specifically on one of the applications available in the ChAImeleon project platform. Initially, users deploy a remote desktop and select a breast cancer dataset as input. Then, the tool is configured and executed for two specific use cases: performing a quality check on all sequences and selecting only mammographies. The execution is carried out as a Jobman task, with parameters set for input and output directories and configuration files. The output folders are organised hierarchically by patient, study, and series, allowing users to see the difference between including all modalities and sequences (i.e. MRI series) and selecting only mammographies. Additionally, the results include detailed information on included and excluded sequences, identification of corrupted files, and reports generated in various formats (xlsx, json and html). The execution process and DICOM file preparation are meticulously monitored through logs.

Breast dense tissue segmentation

The video demonstrates the procedure for automatically segmenting the breast area and dense tissue in digital mammograms using the tool on the central node. First, a remote desktop in the ChAImeleon application is deployed. Once the desktop is loaded, the series and images corresponding to the study of interest are identified in the breast cancer dataset. The mammography image to be segmented is then selected and can be visualised before running the tool. The output directory where the segmentation will be saved is also defined, and then the tool can be run.

The tool execution is automated by creating a job with Jobman. This job is executed with two parameters that have been previously defined in the tool's entry point: one indicating the input path of the DICOM image and another one specifying the output directory where the segmentation will be stored. In about thirty seconds, the process is successfully completed, generating a DICOM SEG file in the output directory that includes both a dense tissue mask and a breast tissue mask combined into one image. These results can be inspected by visualising the masks over the original image.

Feature-based harmonisation

This video demonstrates the process of running the feature-based harmonisation tool on radiomic features using a remote desktop application in the ChAImeleon environment. Initially, the required input for the tool, consisting of a CSV file with the radiomic features and a CSV file with the corresponding metadata, is presented. In this scenario, the user selects the ComBat method, using the manufacturer as 'centre-effect'. The tool is then executed by running the appropriate udocker command and mounting the necessary volumes for both the

input and output of the tool. Upon completion of the harmonisation process, the user can inspect the results, including the harmonised radiomic features and the harmonisation parameters, in the specified output path. Furthermore, histograms of an indicative radiomic feature under different scanner settings (Philips and Siemens) are displayed, showing a significantly better overlap of the harmonised data compared to the raw data after applying the harmonisation methods.

Scenario 3: FAIR EVA tool

The video first shows how the version of FAIR EVA adapted for EUCAIM can be downloaded and deployed using docker. For this demo we have used a previously existing Fair Data Point and we chose a dataset present in it to show the capabilities of FAIR EVA. Once the tool is running the video shows a series of queries to FAIR EVA using curl as client. A total of 22 tests for RDA indicators are run, showing both cases where the dataset passes the test and others where not. The list of RDA indicators tested in the video is the following:

- RDA-F1-01M
- RDA-F1-01D
- RDA-F1-02M
- RDA-F1-02D
- RDA-F2-01M
- RDA-F3-01M
- RDA-F4-01M
- RDA-A1-01M
- RDA-A1-02M
- RDA-A1-02D
- RDA-A1-03M
- RDA-A1-03D
- RDA-A1-04M
- RDA-A1-04D
- RDA-A1-05D
- RDA-A2-01M
- RDA-I1-01M
- RDA-I1-01D
- RDA-I1-02M
- RDA-I1-02D
- RDA-I2-01M
- RDA-I2-01D

F. Future work

Federated processing

EUCAIM is a federated infrastructure designed to address the current fragmentation of cancer data, however, it provides two central nodes equipped with storage and advanced processing capabilities. These central nodes are crucial for data providers who lack the resources to set up a federated node and serve as a repository for data that needs to be transferred after the completion of a project. As a first proof of concept, the preprocessing tools were showcased in a test demonstration environment, while the forthcoming work in EUCAIM will expand the data preprocessing capabilities to the entire federated infrastructure to create a robust, preprocessing workflow across EUCAIM. This work will be reflected in future deliverables.

Clinical data preprocessing and ETL

In the next months we will also further develop the preprocessing workflow of clinical data. We focus first on providing a robust pipeline for images since the EUCAIM infrastructure primarily focuses on cancer imaging data. However, these images are often enriched with critical clinical information, which provides instrumental clinical context to the images, and is essential for many research applications. As a counterpart, clinical data require thorough preprocessing to ensure its utility and accuracy.

New and uncovered use cases

Despite the numerous tools already part of the EUCAIM preprocessing tool catalogue (over 40), some use cases remain uncovered. Therefore, future efforts will focus on gathering tools that enhance the preprocessing capabilities within EUCAIM, allowing the preprocessing of more organs, modalities, and tasks. To this end, EUCAIM internal partners are encouraged to contribute additional tools to the consortium, and external members are invited to apply to EUCAIM calls to further develop and validate their tools. Furthermore, the inclusion of external state-of-the-art tools such as TotalSegmentator[31] and MONAI pretrained models[32] is being considered to cover a broader range of use cases.

DICOM Web Viewer in the central node

In parallel, work is underway on the Quibim DICOM Web viewer, the central node viewer. Once a robust version of the viewer is integrated into the central node, efforts will shift to incorporating various annotation tools within the viewer. This integration will enable direct execution of these tools from the viewer, facilitating a semi-automatic segmentation approach for clinicians.

Improved harmonisation capabilities

Future directions for data harmonisation will include the development of further automated harmonisation tools facilitating real-time data harmonisation and improving efficiency. Furthermore, integrating and standardising diverse data types for comprehensive multi-modal analyses or harmonisation for longitudinal data would be another relevant areas to address in the future.

De-identification

Regarding data de-identification, a key focus is on developing the **Wizard tool**. As mentioned in Section B, we have already completed the definition of the blacklisting per modality. The next steps include defining the whitelisting, developing the risk assessment module, and preparing the tool's output report. Additionally, we will concentrate on the de-identification of clinical data. This will involve strategies for both scenarios: when the clinical data complies with the common data model and when it does not. Moreover, we plan to integrate new tools into the de-identification pipeline, such as de-facing solutions and tools for removing burned-in text in images.

[31] https://github.com/wasserth/TotalSegmentator

[32] https://monai.io/model-zoo.html

Project title: European Federation for Cancer Images
Project acronym: EUCAIM
Grant Agreement: 101100633
Call identifier: DIGITAL-2022-CLOUD-AI-02

D5.4: Supplementary material

Responsible partner(s): Quibim

Date of delivery: 28 June 2024
Version: 1.0

This document complements the material presented in the document **D5.4: Data Preprocessing Tools and Services** and the video that can be found here:
https://www.youtube.com/watch?v=prcyL7hmUYc

1- Annotation tools validation documentation

DA1. Multi-regional prostate segmentation tool

Partner: Quibim
Validator: HULAFE
Tool state: On development/Developed/**Containerized**
Registered in bio.tools: Yes/**No**
Project source: -
Document version: 0.0
Validated: Yes/No

DA1.1. Conceptual validation

Tool description
The tool performs an automatic multi-regional segmentation of the prostate into central-transition zone (CZ+TZ), peripheral zone (PZ), and seminal vesicle (SV) using a T2-weighted MRI image. A heterogeneous database of 243 T2-weighted prostate studies was used to train a U-Net based model with deep supervision.

Data
The segmentation algorithm expects as input a prostate-centered T2-weighted MRI.
For the training of the model, the total number of MRI studies gathered was 243. The age of the subject cohort was within the range of 25 to 92 years old. A balance (50/50%) between non-clinically significant (PIRADS < 3) and clinically (PIRADS ≥ 3) significant prostate cancer patients was found in the dataset (121 non-clinically significant vs. 122 clinically significant, respectively). The studies were acquired from 7 countries, with 10 different MRI scanner models from 3 vendors using a wide variety of acquisition parameters.

Methods
The methods followed to train the model are the following. First, a preprocessing was applied to the images to resize them to a common size and normalise them to [0,1] interval. Then, a U-Net 2D with deep supervision was trained with batches of 2D slices of 256x256, together with DSC as cost function, a cyclical learning rate, and Adam as optimizer. Additionally, to produce slightly different batches, data augmentation transformations were used, consisting in rotations, flips, and noise addition. Finally, mislabelled pixels were reassigned using morphological operators to refine the segmentation.

Use
The process does not need any interaction from the user since it is fully automatic.

Input/Output formats

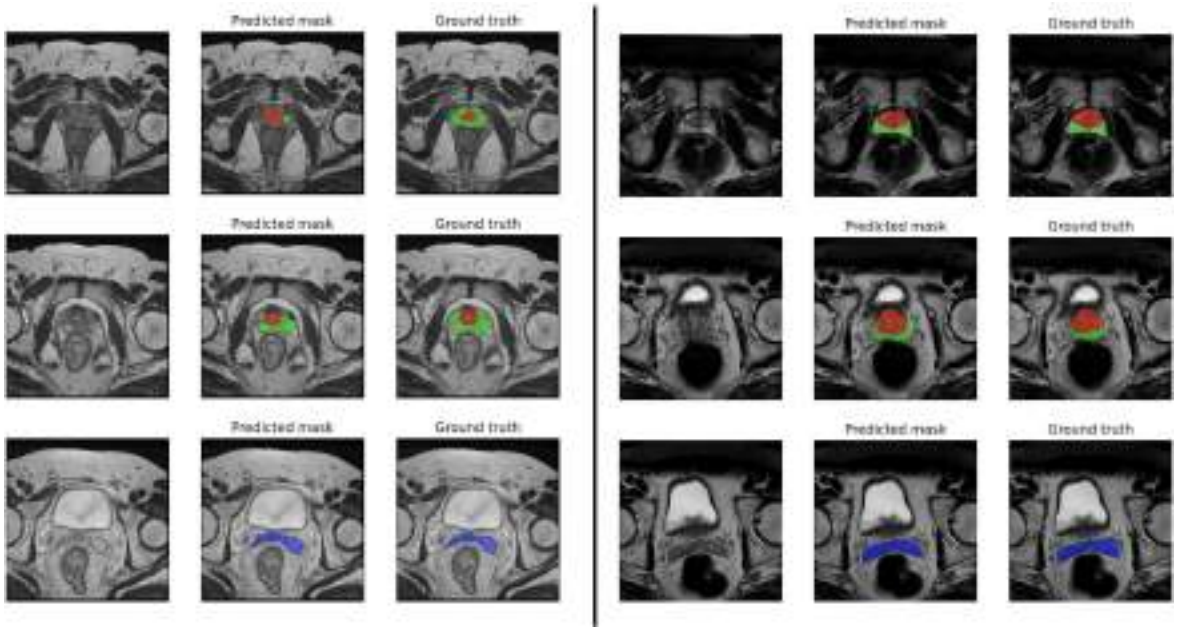
The tool receives a T2-weighted prostate-centered PIRADS-compliant acquisition in DICOM format and returns a multi-label segmentation (CZ-TZ, PZ, and SV) in DICOM SEG format.

Quantitative results

The results are scored using Dice Score Coefficient (DSC). We obtained a DSC of 0.88 ± 0.01 in prostate gland segmentation, 0.85 ± 0.02 in CZ-TZ, 0.72 ± 0.02 in PZ, and 0.72 ± 0.02 in SV. Because of the limitations of the DSC, segmentation results with a DSC above 0.7 were considered accurate.

Qualitative results

Qualitative results are depicted in the following image.



Additional information

- The algorithm holds FDA 510k and MDR certifications.
- **Use case I**. In the ProCancer-I project, its application serves a dual purpose: (a) integrating the algorithm into the ProCancer-I viewer to execute automatic segmentation, thereby eliminating the need for radiologists to segment new cases entirely, enabling them to adjust the automatic segmentation instead; and (b) retraining the algorithm using the manual corrections made by clinicians.
- **Use case II**. The algorithm is integrated into QP-Prostate® solution, allowing to streamline radiologists' workflows by assisting in PSA and volume calculation, and biomarkers extraction per region.
- Publication can be found here.

DA1.2.Technical validation

Data

The total number of MRI studies gathered was 243. The dataset was collected retrospectively through an observational study approved by the Ethics Committee at every institution and waived from informed consent collection. The age of the subject cohort was within the range of 25 to 92 years old. A balance (50/50%) between non-clinically significant (PIRADS < 3) and clinically (PIRADS ≥ 3) significant prostate cancer patients was found in the dataset (121 non-clinically significant vs. 122 clinically significant, respectively). The latter was built by 18% PIRADS 3 (22 cases), 25% PIRADS 4 (30 cases), and 57% PIRADS 5 (70 cases). The studies were acquired from 7 countries, with 10 different MRI scanner models from 3 vendors using a wide variety of acquisition parameters (Table 1). The cases were split into training set, validation set, and testing set.

Manual delineations of the CZ-TZ, PZ, and SV were performed by a team of two radiologists, each with at least 20 years' experience on pelvic MR, from the same center (University and Polytechnic Hospital La Fe in Valencia, Spain). When there were significant differences between the delineations, both radiologists agreed the consensus prostate segmentation mask considered the ground truth to train the model. All annotations were performed on a slice-by-slice basis using the ITK-SNAP tool.

Table 1. Ranges of the technical parameters of the T2-weighted sequence.

SEQUENCE	Fast-spin echo (FSE)/turbospin-echo (TSE)
MAGNETIC FIELD	1.5–3 T
PIXEL SIZE	[0.28–0.94] mm
ACQUISITION MATRIX	[220–512] × [160×512] mm
SPACING BETWEEN SLICES	[2.1–6.5] mm
GAP	[0–2] mm
SLICE THICKNESS	[2.1–6] mm
TR	[2020–13,634] ms
TE	[80–170] ms
FLIP ANGLE	[90–160]◦

Methods:

Several steps were taken to perform prostate segmentations into CZ+TZ, PZ, and SV.

First, to achieve a balance between resolution and computational load, all the 2D images were resized to a common size of 256 x 256. Then, all the images were intensity-normalised to the interval [0, 1] to reduce intensity range.

Regarding the network architecture, a U-Net-based one was used. To improve the performance of the model, a deep supervision stage was added which applied a 1 x 1 convolutional filter to the outputs of the decoder blocks to combine them with the output of the network to force earlier layers to provide activation maps closer to the desired segmentation. To train this network, the inverse of the Dice score coefficient (DSC) was used as cost function and Adam as optimization algorithm along 300 epochs dividing the data in batches of 100 2D input slices of 256 x 256 and their respective segmentation masks of 256 x 256 x 4 (one channel per class).

During training, random data augmentation transformations were used to produce slightly different batches of data to improve the model's performance and generalisation. The transformations applied to the images consisted of Gaussian noise addition of standard deviation varying in the interval [0, 0.08], rotation in the range of [−15, 15] degrees and left–right flip. The probability that a specific transformation was applied to a given image was 30% in each data augmentation iteration. Also, DSC was used as loss function, Adam as optimizer, and a cyclical learning rate was implemented.

Finally, to avoid randomly mislabeled pixels and background-labeled pixels, morphological operators were applied, consisting of a per-class opening and a black hat to the combination of the CZ-TZ and PZ masks, that was used to find the background-labeled pixels, which were assigned as CZ-TZ pixels afterwards.

Specific technical information

- **GPU/CPU:** CPU
- **Programming language:** Python
- **Expected RAM usage**: 4 GB
- **Software version**: 1.0.0
- **Libraries**: Numpy, Tensorflow
- **Minimal security measures (containers):**
    - Writing to host data restricted to a non-root user: Yes/No
    - Container require to be executed in a privileged mode: Yes/**No**

Traceability and monitoring

The tool incorporates logs to keep the user informed about the process status during runtime, which are stored in a file accessible upon completion of the segmentation process. Moreover, it generates a 'success.txt' file upon successful execution. In case of an error, the tool will display the corresponding error message to inform the user about the source of the issue.

Unitary tests

We conducted tests to verify the tool's functionality, ensuring thorough coverage of all potential functionalities.

DA1.3.Integration validation

Communication channel for the helpdesk, technical support channel
For any inquiries contact: alejandrovergara@quibim.com, celiamartin@quibim.com.

Most common errors

- Unsupported Image Orientation: segmentation is only compatible with axial slices.
- Invalid input format: only DICOM is accepted.

FAQs

Which is the purpose of the tool?
The tool performs an automatic multi-regional segmentation of the prostate into central-transition zone (CZ+TZ), peripheral zone (PZ), and seminal vesicle (SV) using a T2-weighted MRI image. Then, it can be used as an automatic annotation tool to avoid performing a manual segmentation from scratch or as a preprocessing step included in an analysis pipeline.

Which imaging format is accepted?
The tool expects DICOM imaging.

What output does the tool generate?
The output consists of a DICOM SEG file with the segmentation generated by the tool, a file including the logs of the tool execution process, and a file indicating if the tool was executed without errors.

How long does it take to perform a segmentation?
About 2 minutes at most, depending on the computation capacities.

User Manual

Installation/configuration instructions (only for downloadable tools)
N/A

Usage instructions
The tool needs two arguments:

- **-i** → input directory to the T2w sequence containing .dcm files.
- **-o** → output directory to store the results.

Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.

- Input: T2w MRI imaging in DICOM format.
- Output: DICOM SEG with the segmentation, *_output.log* file with the logs generated during the execution, and *_success.txt* file generated if the tool was executed without errors.
- Preprocessing: the preprocessing required for the images is already included in the tool, consisting in an image normalization and resizing.
- Mandatory/optional data: the T2w MRI is mandatory. No optional data is accepted.

- Cases in which the tool should not be used. The performance may decrease if the T2w MRI acquisition parameters are not within the ranges described in Table 1. Additionally, it is recommended not to use the tool when the acquisition protocol includes endorectal coil, since the T2w will be artefacted.

DA2. Dense Tissue Segmentation

Partner: ITI
Validator: QUIBIM
Tool state: On development/Developed/**Containerized**
Registered in bio.tools: Yes/**No**
Project source: -
Document version: 0.0
Validated: Yes/No

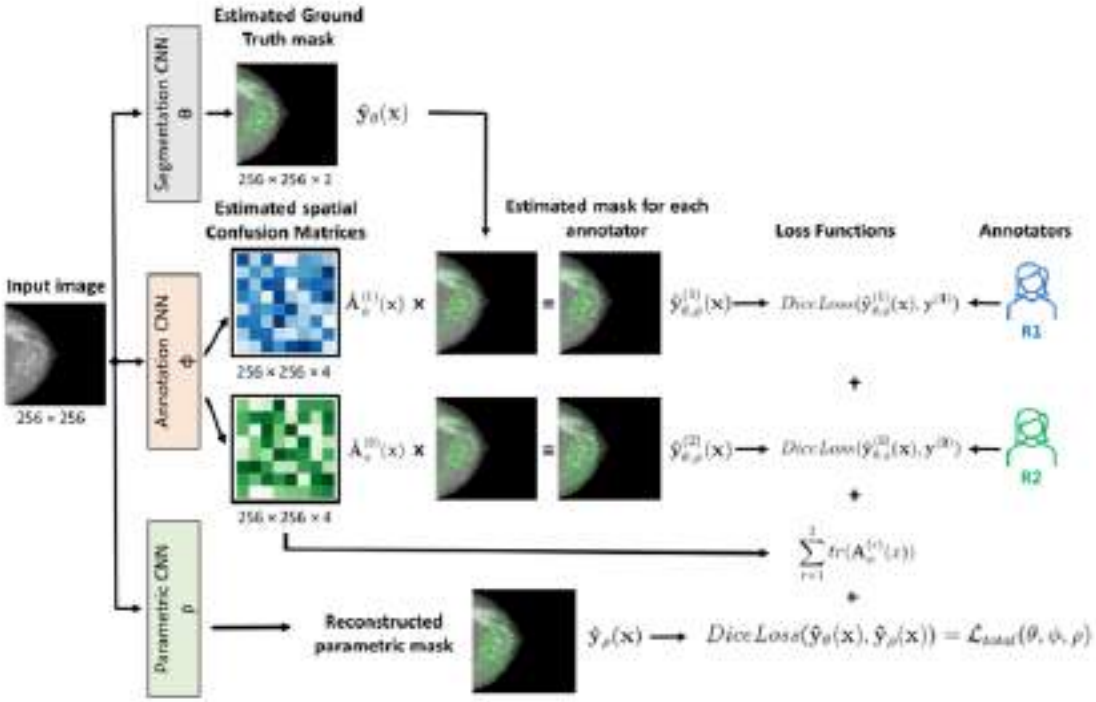
DA2.1.Conceptual validation

Tool description
The tool performs automatic segmentation of the breast area and the dense tissue in digital mammograms. The model was trained with a heterogeneous database of 2496 mammograms obtained from 11 different centers. The trained model (CM-YNet) returns both the dense tissue mask and the breast tissue mask.

Data
The segmentation model expects as input a 2D digital FFDM in “for presentation” format.

A multi-center study covered women from 11 medical centers of the Generalitat Valenciana (GVA) as part of the Spanish breast cancer screening network. It included 1785 women with ages from 45 to 70. The cranio-caudal (CC) and mediolateral-oblique (MLO) views were available for 10 out of 11 of the centers, while one center only collected the CC view. The dataset was randomly partitioned into 75% (2496 mammograms) for training and validation (10%), and 25% for testing (844 mammograms).

Methods
We proposed a CM-YNet architecture which models the ground truth labels provided by two radiologists. The model estimates the dense-tissue mask and segmentation parameters for compatibility with threshold-based tools. The structure of the proposed CM-YNet model is shown in the following figure:



The algorithm was implemented in Pytorch and trained for a maximum of 500 epochs. The epoch with the lowest validation loss was saved and used for test predictions. All the models were trained on an NVIDIA Tesla V100 using a batch size of 8, a learning rate of 0.0001, and the AdamW optimizer with default parameters. The DICOM input images were resized to 256 × 256 pixels. Data augmentation was performed during training with random vertical flips assuming that all the images were left-oriented.

Use

The tool will show the segmentations (breast and dense tissue) for the provided input images.

Input/Output formats

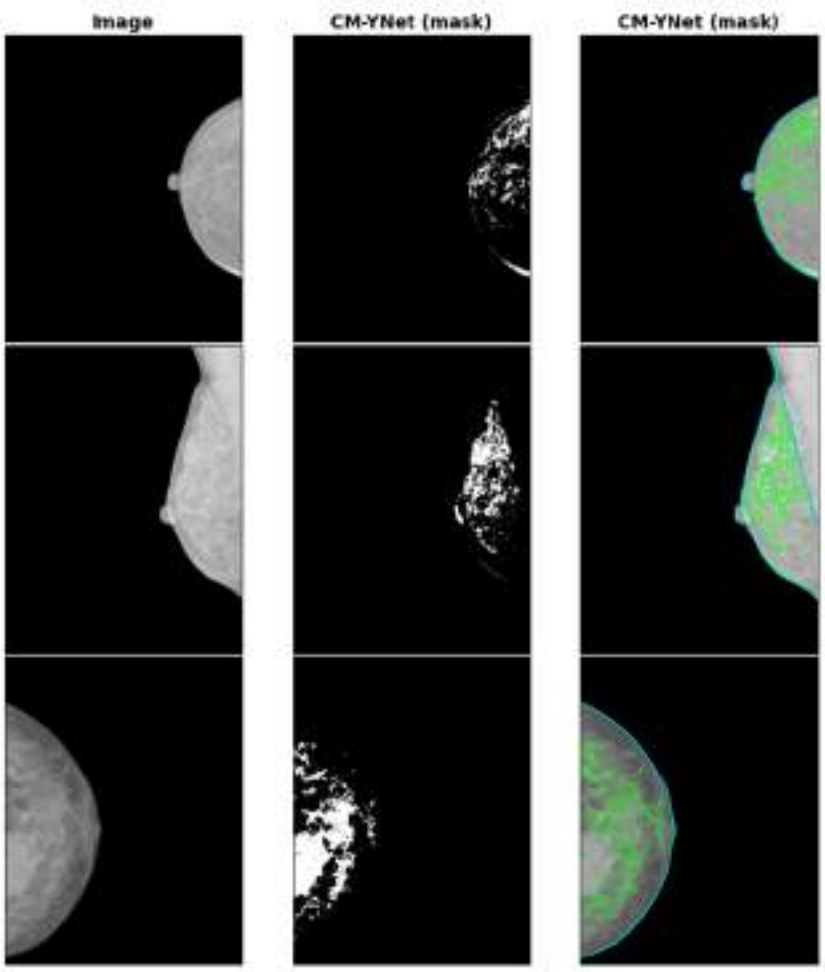
The tool receives a 2D digital FFDM in 'for presentation' format and writes the results in DICOM-SEG format, including the resulting dense tissue mask and breast tissue mask.

Quantitative results

The results are scored using Dice Score Coefficient (DSC). We obtained a DSC of 0.84 ± 0.10 for the dense tissue segmentation. For the breast area segmentation the obtained DSC was 0.99 ± 0.01 for the same test set.

Qualitative results

Qualitative results are depicted in the following image.



Additional information

- **Use case I**. The ITI Breast calculate automatic segmentation pipeline was implemented at the Hospital del Mar d'Investigacions Mèdiques (IMIM), as it was used to segment and automatically calculate the percent density in a prospective research study including approximately 750.000 mammograms.
- **Use case II**. There is an online version of the tool aimed for demonstration at request.
- Publication can be found here.

DA2.2.Technical validation

Data

A multi-center study covered women from 11 medical centers of the Generalitat Valenciana (GVA) as part of the Spanish breast cancer screening network. It included 1785 women with ages from 45 to 70. The cranio-caudal (CC) and medio lateral-oblique (MLO) views were available for 10 out of 11 of the centers, while one center only collected the CC view. This dataset was used for training, validation, and testing. The dataset was randomly partitioned into

75% (2496 mammograms) for training and validation (10%), and 25% for testing (844 mammograms). The mammograms of the same patient were always included in the same set.

Additionally, an independent dataset composed of 381 images obtained at the Institut Hospital del Mar d'Investigacions Mèdiques (IMIM) was included only for testing to obtain a better evaluation of the generalization performance of the models. Because the researchers at IMIM had a particular interest in testing the fully automated tool in various types of images, 283 out of the 381 images at IMIM were obtained from old acquisition devices with lower image quality, making the segmentation task more challenging. Only CC views were provided for this dataset.

Since in Spain "raw" mammograms are not routinely stored, all the mammograms are of the type "for presentation". All mammograms were segmented independently by two expert radiologists using a proprietary thresholding-based tool.

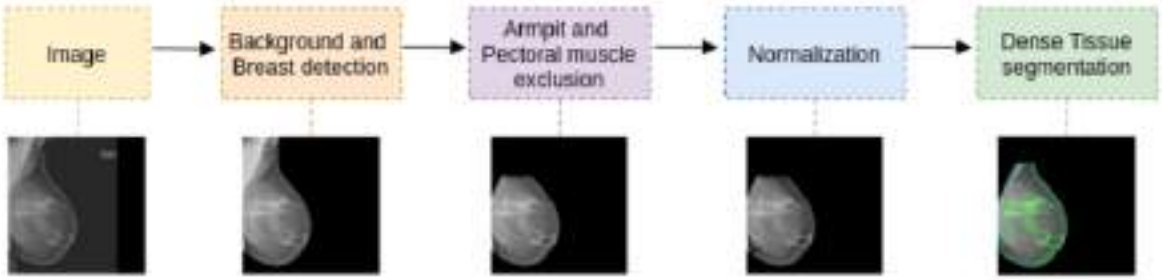
A summary of the data used is presented in the following table:

Id	Center	Device	#Women	#Images
01	Castellón	FUJIFILM	191	382
02	Fuente de San Luis	FUJIFILM	190	380
04	Alcoi	IMS s.r.l./Giotto IRE (*)	66	132
05	Xàtiva	FUJIFILM	159	318
07	Requena	HOLOGIC/Giotto IRE (*)	28	56
10	Elda	SIEMENS/Giotto IRE (*)	311	622
11	Elche	FUJIFILM	278	556
13	Orihuela	FUJIFILM	117	234
18	Denia	IMS s.r.l./Giotto IRE (*)	38	76
20	Serrería	(**)	177	354
99	Burjassot	Senography 2000D	230	230
21	IMIM-1	FUJIFILM	98	98
22	IMIM-2	Lorad/Hologic Selenia	283	283
		Total	2166	3721

(*) Implies the use of a new device [Gioto IRE] since 2015. (**) The device is unknown.

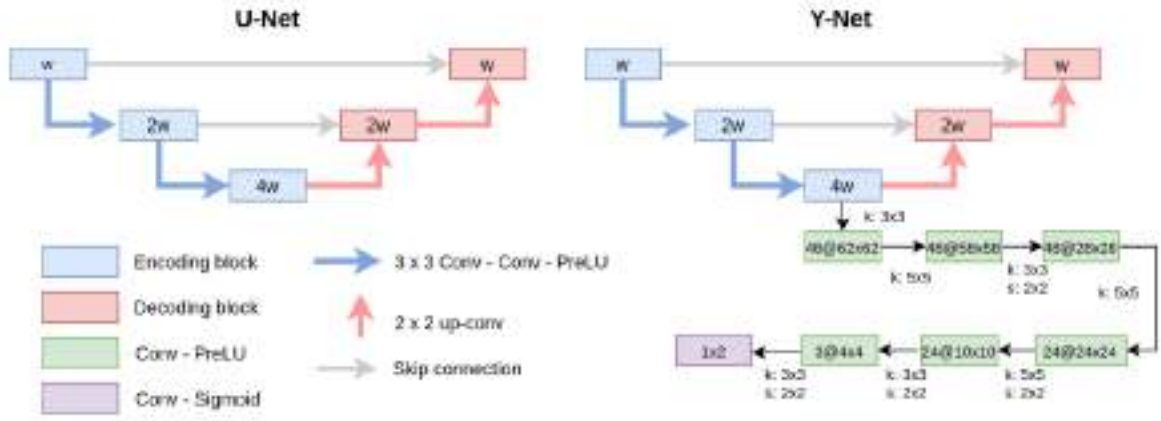
Methods:

The segmentation pipeline consists of the following steps: a first step covering breast detection and pectoral muscle exclusion, a second step to exclude armpit and pectoral muscle, a third step to normalize the histogram variability between acquisition devices, and finally, a Deep Learning model carrying out the dense-tissue segmentation task.



We implemented the CM-YNet architecture, which is a generalization of the U-Net architecture by adding a parallel branch that outputs a classification label. The results demonstrated that the joint learning implemented within Y-Net improved diagnostic accuracy. In the Y-Net implementation, the parallel branch predicts the segmentation parameters α and th. Therefore, with the Y-Net model, we can simultaneously estimate the dense-tissue mask and the segmentation parameters that would allow manual modifications with a threshold-based tool.

The parameter estimation branch consists of several convolution layers added from the last encoder layer of the U-Net. Similarly, as in ECNN, the convolutions reduce the inputs until the segmentation parameters are extracted.



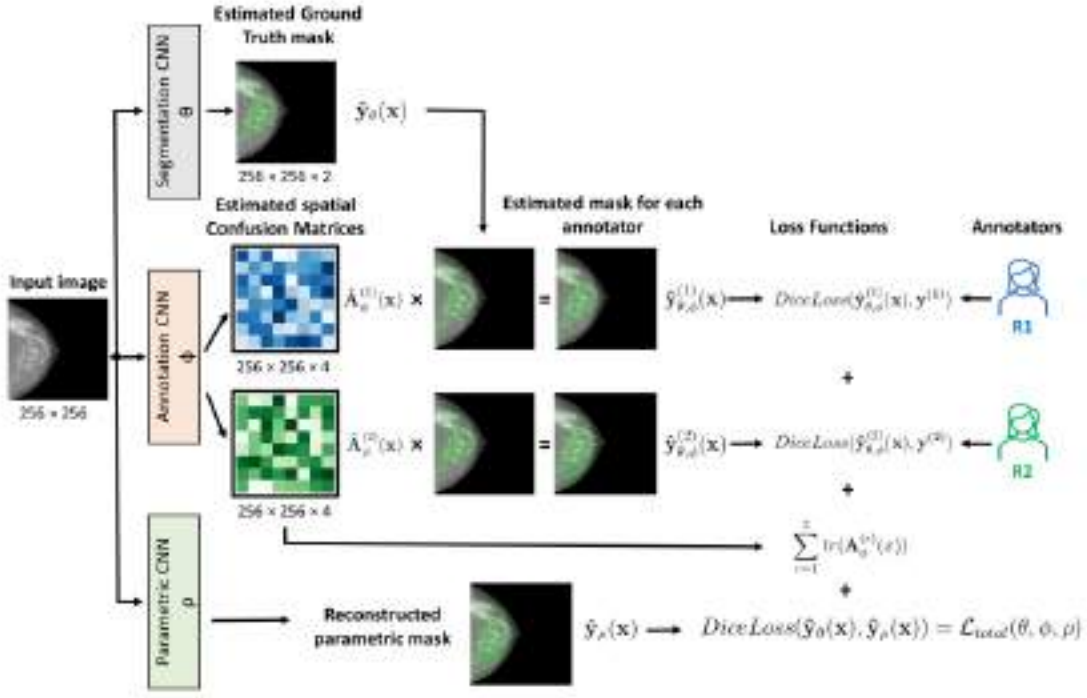
The lack of a unique ground truth due to inter-reader variability means that it is normally necessary to train the Deep Learning models using the segmentation of both radiologists as independent annotations, or by fusioning both labels. Either approach can yield a high degree of concordance between the model and the annotators. However, the available expert labels are noisy approximations of the unknown ground-truth segmentation mask. The implemented architecture consists of two coupled convolutional neural networks (CNNs):

1. The first is the segmentation network that estimates the true segmentation.

2. The second is the annotation network that models the characteristics of individual experts by estimating the pixel-wise confusion matrices (CM).

The annotation network shares the same parameters as the segmentation network apart from the last layers. It estimates the CMs at each spatial location, thus yielding a c × c output, where c is the number of channels, which is two for a binary segmentation, as in our case. The implemented segmentation network is the YNet described above.

The CM-YNet architecture, shown in the following figure, models each radiologist's label to estimate the dense-tissue mask and segmentation parameters for compatibility with threshold-based tools.



Specific technical information

- **GPU/CPU:** GPU
- **Programming language:** Python
- **Expected RAM usage**: 16 GB
- **Software version**: 0.6
- **Libraries**: Numpy, Pytorch, pydicom, and ITI-IA (a library developed by ITI with AI methods)
- **Minimal security measures (containers):**
    - Writing to host data restricted to a non-root user: Yes/**No**
    - Container require to be executed in a privileged mode: Yes/**No**

Traceability and monitoring

Informative messages are displayed on the screen when running the Docker container. If an error occurs, the corresponding error message informs the user. A completion message is displayed at the end of the execution.

Unitary tests

Yes, there are unitary tests for all the main functions of the tool.

DA2.3.Integration validation

Communication channel for the helpdesk, technical support channel

For any inquiries contact: dsilveira@iti.es, silviaruiz@iti.es.

Most common errors

- Invalid input format: only DICOM is accepted.
- Invalid input dimensions: the input should be a 2D image.
- Name of the image not provided.

FAQs

Q: What is the purpose of the tool?
A: The tool performs an automatic segmentation of the breast area and the dense tissue using digital mammograms. Therefore, it can be used as an automatic tool to avoid manual segmentations.

Q: Which imaging format is accepted?
A: The tool expects a 2D DICOM image.

Q: What output does the tool generate?
A: The tool generates an output that includes both a dense tissue mask and a breast tissue mask combined into one image. This combined image is provided in DICOM-SEG format.

Q: How long does it take to perform a segmentation?
A: About 30 seconds using GPU, depending on the computation capabilities.

User Manual

Installation/configuration instructions (only for downloadable tools)

N/A

Usage instructions

The tool is dockerized and needs two arguments:

- The input path to the dicom image: input_dcm_path
- The output directory to store the results: output_directory

Run the following command:

- jobman submit -i mammography-density-segmenter -r small-gpu -- -p input_dcm_path -o output_directory

Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.

- Input: A 2D digital FFDM in DICOM format.
- Output: DICOM-SEG file with the segmentation.
- Preprocessing: Image preprocessing is not necessary.
- Mandatory/optional data: The 2D FFDM image is mandatory to run the tool; no optional data is accepted.
- Cases in which the tool should not be used: The tool should not be used in structures besides the breast, and it should not be used with any format other than 2D DICOM FFDM.

DA3. MR Neuroblastoma Tumor Segmentation

Partner: HULAFE
Validator: DKFZ
Tool state: On development/Developed/**Containerized**
Registered in bio.tools: Yes/**No**
Project source: -
Document version: 0.0
Validated: Yes/No

DA3.1.Conceptual validation

Tool description
This code implements automatic segmentation of Neuroblastic tumor in T2-weighted magnetic resonance images using a trained nnUnet architecture within the PRIMAGE project framework. The model outputs the segmentation masks of the input images.

Data
This dataset includes MR data from 132 pediatric patients with neuroblastic tumors diagnosed between 2002 and 2021. Patients from Spain, Austria, and Italy were included. Tumor types varied, with abdominopelvic and cervicothoracic locations. Images were obtained using 1.5 T or 3 T scanners from different manufacturers. MR protocols included T1-weighted, T2-weighted, diffusion-weighted, and dynamic contrast-enhanced sequences. Images were pseudonymized using the EUPID system and stored for analysis in the PRIMAGE project.

Methods
The automatic segmentation model was developed using the state-of-the-art, self configuring framework for medical segmentation, nnU-Net. All the images were resampled with a new voxel spacing: [z, x, y] = [8, 0.695, 0.695], corresponding to the average values within the training data

set. The model training was performed along 1000 epochs with 250 iterations each and a batch size of 2. The loss function to optimize each iteration was based on the Dice Similarity Coefficient (DSC). A z-score normalization was applied to the images.

Use

The tool runs through a Docker image. To build the image, it's necessary to mount a /data volume containing the Database (BBDD) folder with input images.

docker run -v /project_path:/data nnunet_segmentation

Replacing /project_path/ by the actual path to the database.

Input/Output formats

The segmentation model expects Neuroblastic tumor T2-weighted MR images as input. In the corresponding database root folder:

- Patient 1
  - Study
    - SequenceName1_0000.nii.gz
- Patient 2
  - Study 1
    - SequenceName2_0000.nii.gz
  - Study 2
    - SequenceName3_0000.nii.gz
- ...
- Patient N
  - Study 1
    - SequenceNameN_0000.nii.gz

Images should have names ending with _0000.nii.gz.

The segmentations output will be saved in the BBDD_result directory with the following structure:

/project_path/BBDD_result

- Patient 1
  - Study
    - SequenceName1.nii.gz
- Patient 2
  - Study 1
    - SequenceName2.nii.gz
  - Study 2
    - SequenceName3.nii.gz
- ...

- Patient N
    - Study 1
        - SequenceNameN.nii.gz

Quantitative results

The median Dice Similarity Coefficient (DSC) for the trained model is 0.965 (±0.018 IQR (interquartile range)). The automatic segmentation model achieved a better performance regarding the False Positive rates (FPRm). MR images segmentation variability is similar between radiologists and nnU-Net. Time leverage when using the automatic model with posterior visual validation and manual adjustment corresponds to 92.8%

Qualitative results

Three sample cases are presented in the following image, including transversal and coronal MR images, the automatic and manual segmentations, and the manual and predicted mask comparison.

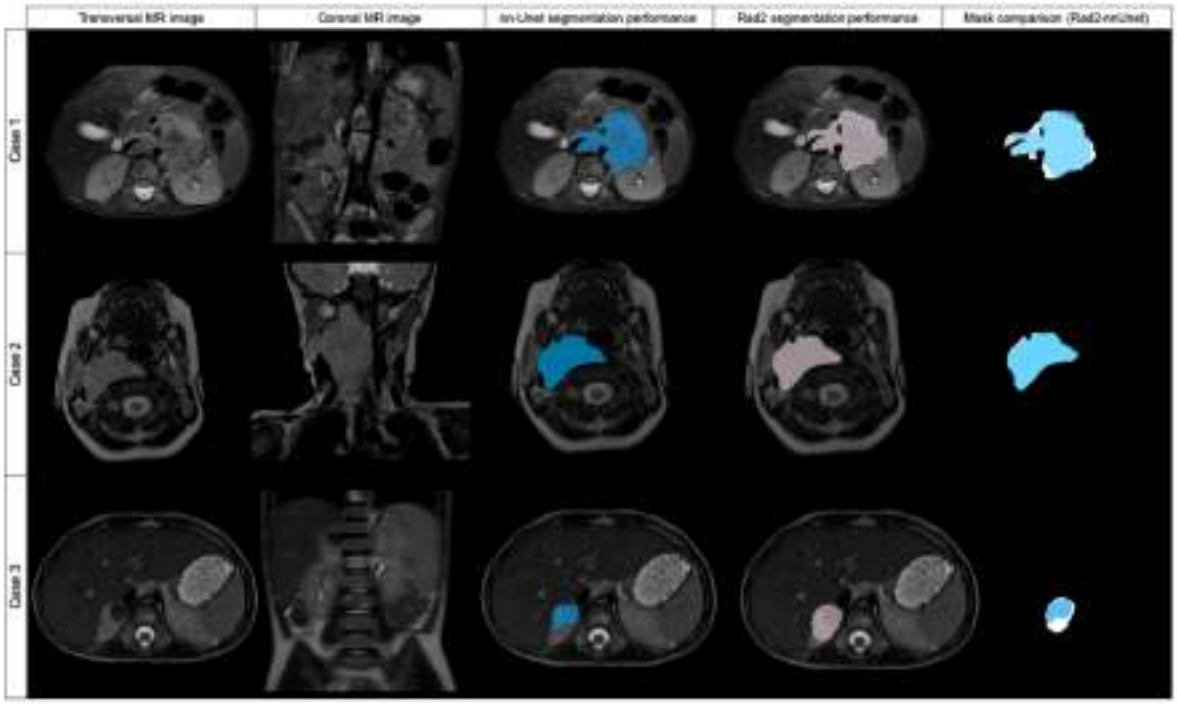


Figure 4. Original transversal and coronal MR images and examples of three cases automatically segmented by nnU-Net (blue labeled) and Radiologist 2 (pink labeled), with mask superposition for comparison. Case 1 was segmented in T2w fat-sat with a DSC of 0.869. Case 2 was segmented on T2w and the DSC obtained was 0.954. Case 3 was segmented with a DSC of 0.617.

Additional information

The code of the tool was developed within the PRIMAGE project framework by Leonor Cerda Alberich, Diana Veiga Canuto and Matías Fernandez Paton of Biomedical Imaging Research Group (GIBI230) at Hospital Politécnico La Fe.

The tool is under MIT license and can be found here, and its paper here.

DA4. nnU-Net segmentation tool

Partner: DKFZ
Validator: Quibim
Tool state: On development/Developed/**Containerized**
Registered in bio.tools: Yes/No
Project source:- https://github.com/MIC-DKFZ/nnUNet
Document version: 1.0
Validated: Yes/No

DA4.1.Conceptual validation

Tool description
nnU-Net is a self-configuring method for deep learning-based biomedical image segmentation, developed by the Applied Computer Vision Lab (ACVL) of Helmholtz Imaging and the Division of Medical Image Computing at the German Cancer Research Center (DKFZ). It is designed to automatically adapt to a given dataset, analyzing the provided training cases to configure a matching U-Net-based segmentation pipeline without requiring expertise from the user. This makes nnU-Net particularly useful for the EUCAIM WP5 project, which aims to create a federated European infrastructure for cancer images data, by providing a powerful tool for processing and analyzing the vast amount of DICOM data involved in cancer research and treatment. The nnUNet is available as an open source tool. For further reading please refer to [1].

Data
nnU-Net is capable of handling 2D and 3D images with various input modalities/channels, including RGB images, CT scans, MRI scans, and microscopy images. It can process images with different voxel spacings, anisotropies, and is robust even when classes are highly imbalanced. The nnUNet is not limited to a specific organ or a region. It covers all datasets part of the Medical Segmentation Decathlon [2] and more without any change in the network topology. Refer [3] to see all supported organ models available.

This versatility makes nnU-Net suitable for processing DICOM images from the EUCAIM project, which encompasses a wide range of cancer types and imaging modalities.

Methods

The nnU-Net pipeline uses a heuristic rule to determine data-dependent hyperparameters, known as the "data fingerprint," to ingest the training data. This process generates pipeline fingerprints that produce network training for 2D, 3D, and 3D-Cascade U-Net configurations. The ensemble of different network configurations, along with post-processing, determines the best average Dice coefficient for the training data, which is then used to produce predictions while inferencing. This approach allows nnU-Net to efficiently process and segment the DICOM images in the EUCAIM project, adapting to the specific characteristics of each dataset.

Use

In the context of the EUCAIM WP5 project, nnU-Net can be used to process and analyze the DICOM images collected as part of the project.
The inference workflow can be invoked headlessly by calling the command:
nnUNet_predict -i INPUT_FOLDER -o OUTPUT_FOLDER -t TASK_NAME_OR_ID -m CONFIGURATION
The process does not need any more interaction from the user since it is fully automatic.

Input/Output formats

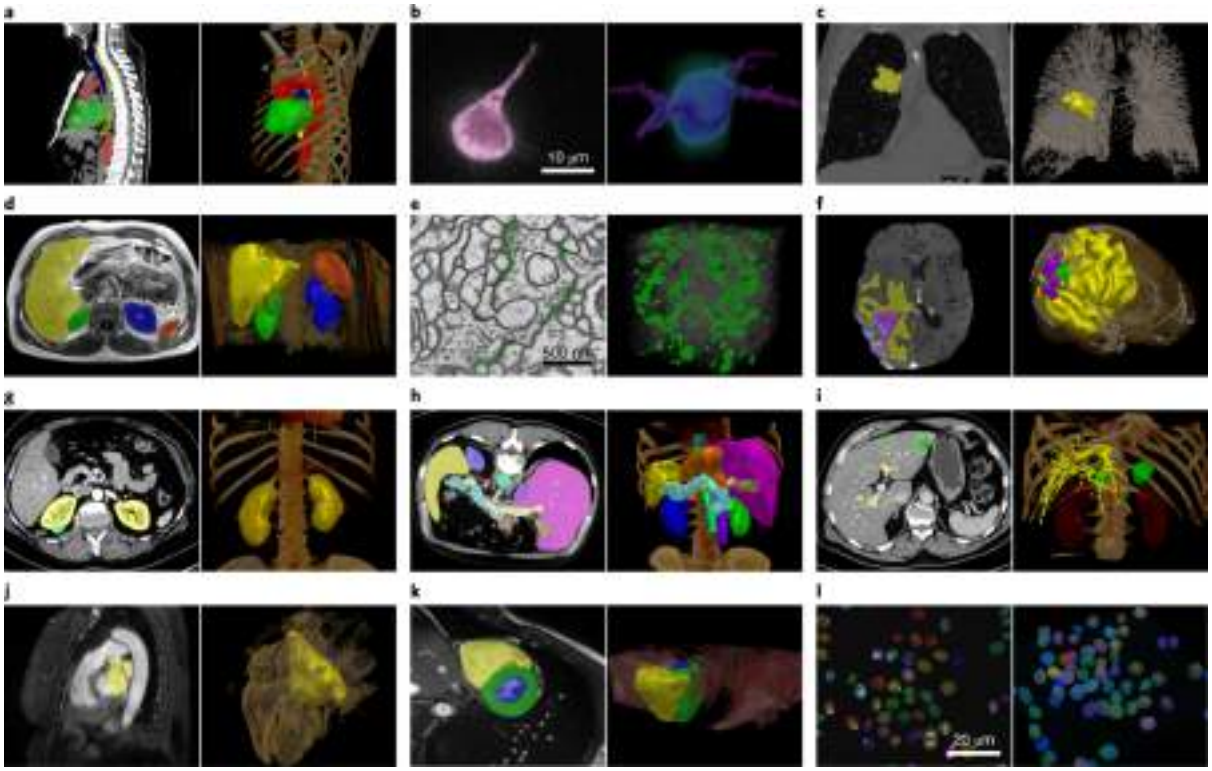
Depending on the task at hand, the tool can receive DICOM data across any modalities.
Eg.

1. Brain: mp-MRI data nativeT1-weighted (T1), post-Gadolinium (Gd) contrast T1-weighted (T1-Gd), native T2-weighted(T2), and T2 Fluid Attenuated Inversion Recovery (FLAIR)
2. Liver: CT image data.

Quantitative results

nnU-Net has demonstrated its effectiveness across a range of biomedical segmentation competitions, surpassing most existing approaches on 23 public datasets. It has been used as a baseline and method development framework by 9 out of 10 challenge winners at MICCAI 2020 and 5 out of 7 in MICCAI 2021, and won the AMOS2022 challenge. More recently, nnU-Net won the *Top Cow 2023* challenge and also was used as baseline for top performers of the *Ps-Hf-Aop 2023* & *TDSC-ABUS* 2023 challenges. These results underscore nnU-Net's capability to handle the complex segmentation tasks involved in the EUCAIM project, making it a valuable tool for processing and analyzing the DICOM images collected as part of the project.

Qualitative results



Some exemplary segmentation results produced by nnU-Net and its corresponding 3D visualization created using MITK Workbench [4].

Additional information
License: Apache-2.0 license
Code: https://github.com/MIC-DKFZ/nnUNet
Publication in [1].

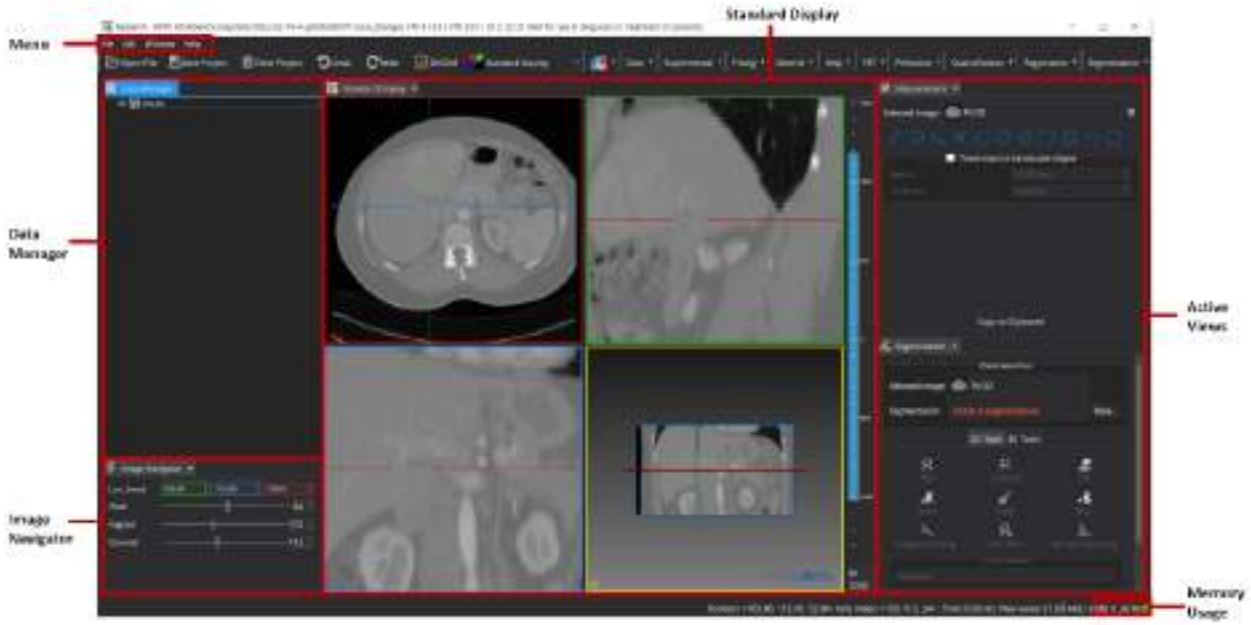
References:
[1] https://www.nature.com/articles/s41592-020-01008-z
[2] https://www.nature.com/articles/s41467-022-30695-9
[3] https://zenodo.org/records/4485926
[4] https://www.nature.com/articles/s41592-020-01008-z/figures/1

DA5. Medical Imaging Interaction Toolkit (MITK)

Partner: DKFZ
Validator: Quibim
Tool state: On development/Developed/**Containerized**
Registered in bio.tools: Yes/No
Project source: -
Document version: 1.0
Validated: Yes/No

DA5.1.Conceptual validation:

Tool description
The Medical Imaging Interaction Toolkit (MITK) is a free, open-source software designed for the development of interactive medical image processing applications. It could be a crucial component in the EUCAIM WP5 project, aiming to facilitate the processing and analysis of DICOM images for cancer research and patient care. MITK**[1]** provides a powerful platform with an easy-to-use interface allowing for the application of various image viewing and processing techniques such as segmentation and registration medical images, making it an ideal “tool-of-tools” for handling the vast amounts of data generated in cancer research and clinical practice.



Data
MITK is designed to work with a wide range of medical image data, including DICOM images, which are commonly used in cancer research and diagnostics. The tool is particularly suited for processing up to 4D images from various modalities, such as MRI, CT, and PET scans. This versatility makes MITK suitable for segmenting & registering DICOM images from the EUCAIM project, which encompasses a wide range of cancer types and imaging modalities.

Methods
MITK employs a modular architecture, allowing for the integration of various image processing and analysis methods. It supports a variety of filter operations for image preprocessing and enhancement, including morphological operations, image measurements and statistics. The plugins provide much of the extended functionality of MITK. Each encapsulates a solution to a problem and associated features.

The Segmentation plugin allows you to create segmentations of anatomical and pathological structures in medical images of the human body. Various 2D & 3D tools are bundled in the segmentation plugin, from basic manual contouring tools to fully automatic deep learning-based tools such as TotalSegmentator. All featured tools are further explained in [2].

MITK also supports image registration via the MatchPoint framework[3]. MatchPoint is a translational image registration framework written in C++. It offers a standardized interface to utilize several registration algorithm resources (like ITK, plastimatch, elastix) easily in MITK. Different algorithms and processing facilities are described in [4].

Use
In the context of the EUCAIM WP5 project, MITK can be used to open & process the DICOM images collected as part of the project. The inference workflow can be invoked headlessly by calling the command:
cd <path MITK folder>
./MitkWorkbench

This opens an interactive MITK workbench standalone application window. The workbench is completely offline. However, to access tools like TotalSegmentator or Segment Anything an internet connection is required.

To use MITK Workbench for processing DICOM images in the EUCAIM project, users can load an image into the MITK Workbench and then activate the desired plugin/tool for the purpose.

Quantitative results
MITK has been the tool of choice for quite a few scientific publications since 2004. The list of citations received for MITK are included in [5].

Additional information

Licence: BSD-3-Clause licence
Code: https://github.com/MITK/MITK

References:
[1] https://mitk.org
[2] https://docs.mitk.org/2023.12/org_mitk_views_segmentation.html
[3] https://github.com/MIC-DKFZ/MatchPoint
[4] https://docs.mitk.org/2023.12/org_mitk_views_matchpoint_algorithm_control.html
[5] https://www.mitk.org/wiki/Publications

2 - De-identification tools validation documentation

DI1. EUCAIM DICOM Anonymizer

Contributor: FORTH

Area: De-identification

Tool state: Completed

Project source: Cardiocare

Licence: LGPLv3

Additional resources:

- https://mircwiki.rsna.org/index.php?title=The_DicomAnonymizerTool

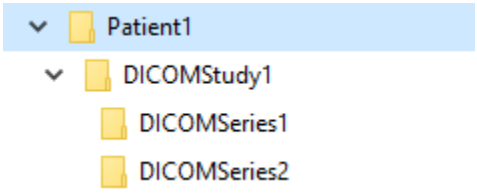
DI1.1. Conceptual description

Tool description
The EUCAIM DICOM Anonymizer is a standalone (desktop) application for Microsoft Windows and MacOS 64bit machines and its function is to anonymize a set of DICOM files. It supports both anonymization on a case-by-case scenario, meaning one DICOM Study at a time (single-mode) or on multiple cases concurrently (batch mode).

Data
The tool supports imaging data in DICOM format, of any modality and cancer type, as long as it is fed with the proper anonymization script (specific to the modality).

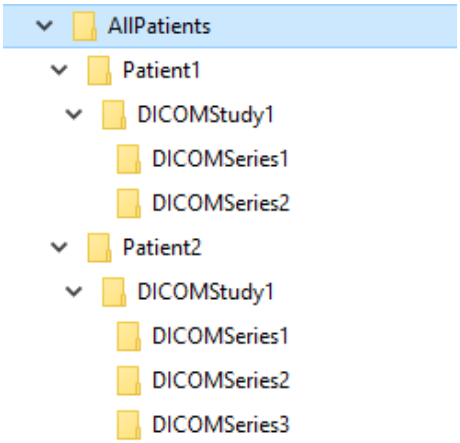
Input/output
It takes as input a DICOM folder and an input config file (anonymization script).
In case the input is a single folder (single-mode):

- A case folder should be created for the patient, containing all the DICOM images for anonymization. If a certain patient has multiple studies, the user should create a unique folder for each DICOM study acquired at different time points. The PatientID of those DICOM studies must be the same and should not be empty.



In case the input is a folder with multiple patient-associated folders (batch-mode):

- A root folder should be created containing all the patient cases, each in a separate folder. Then, each patient case should follow the same structure as in the single-mode.



The output is a DICOM folder where the de-identified patients are saved in the same structure as they are given

Methodology/performance

The tool is implemented in C++ using the Qt Framework[1] version 5.15 and the code was compiled with Microsoft Visual Studio 2015.

In order to perform the DICOM anonymization the EUCAIM DICOM Anonymizer integrates the RSNA DICOM Anonymizer[2] which is a Java command line tool. Therefore, the tool requires a recent Java Developer Kit (JDK) or Java Runtime Environment (JRE). Prebuilt OpenJDK binaries can be downloaded from the Eclipse Temurin web site[3] or from https://jdk.java.net/.

Use

The tool can be launched by double clicking on the *dcm-upload.exe* executable. The first screen/landing screen prompts the user to initiate the process of anonymization by clicking on the “Open” button, to select the patient cases to be anonymized:

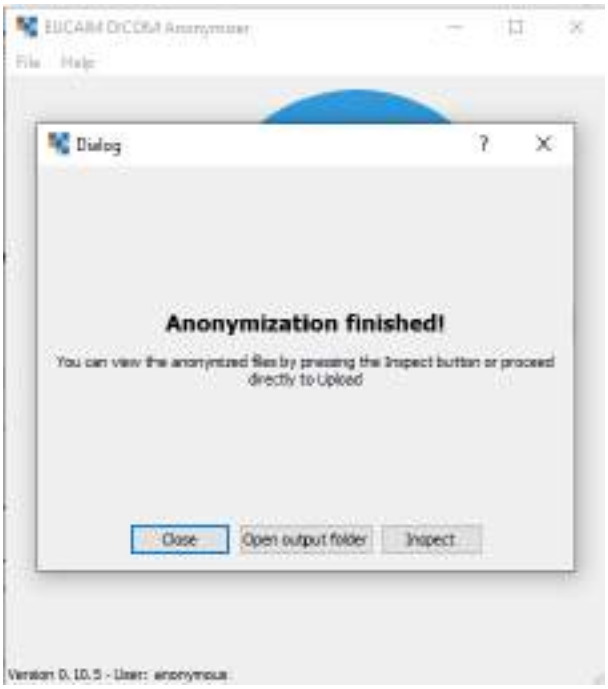
[1] https://www.qt.io/product/framework
[2] https://mircwiki.rsna.org/index.php?title=The_DicomAnonymizerTool
[3] https://adoptium.net/temurin/releases/

After clicking the “Open” button, a drop area is shown that allows the user to "drag and drop" a directory containing DICOM files[4]:

After the user "drops" a folder containing the DICOM images to the tool's area, the tool will load all the DICOM files found in this folder and start the anonymization process. After the tool anonymizes all the files, it creates new DICOM files[5]:

[4] Users can also use the File menu to open the folder containing the DICOM images.
[5] The new anonymized DICOM files are stored in a folder called *anon-out* in the same folder that the tool's executable file resides. These anonymized files are not automatically deleted, the user can delete them at any time.

As shown below, when the anonymization finishes the user is presented with the following dialog where s/he is able to either select the "Open Output Folder" or the "Inspect" button:



If the user chooses the "Inspect" button the MicroDicom viewer (if installed) will appear to present the new DICOM files (i.e., the ones created after anonymization). If the user chooses the “Open Output Folder”, s/he will be able to inspect the result of the anonymization by using a DICOM viewer of their choice. Please note the following:

☞ We advise the users to always inspect the results of the anonymization. Of course, the users could use the viewer to see the DICOM images before the anonymization, but it is especially

important to have a visual inspection before the data are used to be uploaded to any external platform so as to make sure that the correct image files were used and that there is no "burned in" information, like personal patient information, embedded on the image itself (on the "pixel data") and evaded the anonymization. If there's still some patient information on the image, the users should contact the EUCAIM technical team to address this issue.

DI1.2. Technical description

CPU or GPU use: CPU
Main programming language: C++, Java
Expected RAM usage: ~8GB (actual RAM requirements depend on the size of the dataset)
Running mode (interactive/batch-based/case-based…): All are applied.
Software version: x202 (https://github.com/johnperry/CTP/releases/tag/x202)
Libraries: CTP anonymizer tool with version x202 (https://github.com/johnperry/CTP/releases/tag/x202) for the anonymization process, and the C++ Qt framework (https://qt.io/) for the cross-platform GUI.
Does the tool require administrator privileges?: No

Traceability and monitoring mechanism
All actions are logged in a SQLite database on the local (end user's) workstation for observability and problem tracing purposes.

Unitary tests
The tool is a "thin" wrapper around the RSNA CTP Anonymizer so no tests for the actual anonymization mechanisms have been done. Nevertheless, we have done extensive testing on the whole package in the EU project CardioCare[6] where it is used in production mode for the anonymization of a large number of DICOM cases of the prospective study.

Access restriction
There are no restrictions on its use and the source code will be provided under the GPLv3 licence.

Containerization
The tool itself offers a Graphical User Interface (GUI) with a "drag and drop" facility for anonymizing DICOM folders and therefore due to its tight integration with the "host" operating system there's no point in "containerizing" it. Nevertheless, we also offer a Docker image with the RSNA CTP tool that can be used in scripts for automating the anonymization of DICOM imaging data when proper parameters are given to its command line.

[6] https://cardiocare-project.eu/

DI2. Radiomics Enabler

Contributor: MEDEXPRIM

Area: de-identification

Tool state: Completed

Project source: ChAImeleon

Licence: Commercial

Additional resources: the deployment of the tool requires specific technical prerequisites : a virtual machine with Ubuntu 18 as OS, 16Gb of RAM, and two disk partitions (one with approx 100Gb and one with ideally 2Tb.

DI2.1. Conceptual description

Tool description
Commercial tool to extract, de-identify (using CTP Anonymizer, pixel mask, dateshift), and export data from hospital facilities to external repositories.

Data
Any radiological DICOM image, of most modalities (CR, CT, MR, PET, US, MG, SM). Other modalities may be supported (to evaluate on a case-by-case basis)

Input/output
The input and outputs are in DICOM format. Some specific DICOM formats such as DICOM-RT and DICOM-SM are also compatible.

Methodology/performance
Radiomics Enabler® is a tool deployed at hospital facilities, that uses the CTP Anonymizer pipeline for de-identification of medical images, as well as a DicomFilter script to exclude specific series/modalities, and a PixelAnonymizer script to mask pixels of the images that may contain identifying information. A DICOM viewer is embedded to allow series control check and their possible exclusion. Export of images can be configured using specific exporter settings such as DicomWeb API.

Use
Radiomics Enabler® is a web application that automates the extraction, de-identification, curation and export of batches of images from a PACS. The application is functionally structured around research projects, with user rights attached to each project. For a dedicated project, a customised protocol of de-identification is defined. The user selects via a user interface all DICOM studies and/or series that must be extracted in batches from the PACS to which the tool is connected. Once a batch of DICOM files is extracted, the files are pseudonymized or

anonymized according to the protocol of de-identification. A quarantine process is also available to flag and exclude images as potentially identifying. Images can be visually checked using a DICOM viewer embedded in the tool, and all DICOM tags can be explored. Once a visual check has been performed by the user, imaging exams can be exported towards a specific destination that has been configured for the project.

DI2.2. Technical description

CPU or GPU use: CPU
Main programming language: python
Expected RAM usage: 16Gb approx
Running mode (interactive/batch-based/case-based…): interactive
Libraries: docker
Software version: 2.10.0
Does the tool require administrator privileges?: No

Traceability and monitoring mechanism
The log of all user actions and running tasks is available in a log file for each container, but only accessible by the support team (MEDEX).

Unitary tests
[Description of the tests implemented to verify the correct functioning of the tool].
Unitary tests are only launched from a test environment at MEDEX. They are not made to be used in production at local sites.
Some basic checks are to be done by user before starting extraction of data such as :
- Visual check from the home page that connection to PACS is up
- Visual check the configuration of anonymization scripts on a dedicated project
- Testing of images data extraction from PACS using a “TEST” project

Access restriction
[Are there any access restrictions to the source code or to the binaries]
The source code in python is accessible on the local container on the virtual machine. The access and use of the tool is however restricted by a signed licence agreement between the site and MEDEX.

Containerization
The tool is dockerized, but the whole deployment is handled by MEDEX support team.

Additional information
Several pre-requisites are needed to be implemented by the facility where the tool should be deployed.

Radiomics Enabler® is a web application part of the Medexprim Suite™, running in Docker containers. Authorized users can access the application web interface from one of the following web browsers : Mozilla Firefox, Google Chrome, Microsoft Edge.

MEDEX will install Radiomics Enabler® and Indexa™ modules by default on two virtual machines (VM) either:

- on virtual machines provided by the institution.
- providing the virtual machines, ready to use, that the facility will install on its servers.

1. Hardware of provided server

Provide two virtual machines allowing the deployment of Medexprim Suite™software

VM (1) : Radiomics Enabler®	
OS	Ubuntu 20
CPUs	4
RAM	16 Gb
Disk Partitions	Three disk partitions with: **/**: partition of approximately **100 GB** for application and treatment pipelines[1]. **/processing/**: processing partition with enough disk space to process the images. Proposal: **1 TB** scalable. Physically attached partition. Must be expandable later on as needed. **/data/**: partition with enough disk space to store images. Proposal: **1 TB** scalable. NFS mount possible. Must be expandable later on as needed.

[1]In case the VM is configured with LVM; take care that the disk space of /var/lib is not saturated. This may cause incompatibility with Docker.

VM(2) : Indexa™	
OS	Ubuntu 20
CPUs	4
RAM	16 Gb

Disk Partitions	Two disk partitions with: **/** : partition of approximately **300 GB** for application and treatment pipelines **/data/** : partition with enough disk space to store various data. Proposal: **100 GB** scalable. NFS mount possible. Must be expandable later on as needed.

2. Remote access

Provide a nominative remote access (e.g. VPN) to enable MEDEXPRIM to provide support. Also provide the information required to download, install and use this remote access.

Provide a system user with administrative rights to the VM. It can be the root user, or a user named "medexadmin".

3. Network configuration

For the installation and application updates, the following accesses must be open:

- **Package repositories** for the operating system
- Docker repository: **https://download.docker.com**
- Docker and Docker-compose repository: **https://github.com/docker**
- Medexprim Docker image registry: **https://repo.medexprim.net**
- Ansible repository: **https://launchpad.net/~ansible**

By default, Docker uses the default network range 172.20.0.0. In the case this conflicts with your internal network, please specify another network range for Docker to use.
The application's flow matrix can be seen below.

Flow matrix of Medexprim Suite™				
From	**To**	**Port**	**Protocol**	**Description**
VM(1): Radiomics Enabler® (and eZCRF™if needed)				
SSH access from VPN access	VM(1)	22	SSH	SSH access for support
VM(1)	Hospital PACS	As specified	DICOM	Requests c-echo, c-find, c-move to the PACS

Hospital PACS	VM(1)	11112	DICOM	Medexprim Suite receive PACS information
Web browser from hospital network	VM(1)	8000	HTTPS	Access to application Radiomics Enabler® web interface
Web browser from VPN access	VM(1)	8000	HTTPS	
Web browser from hospital network	VM(1)	8080	HTTPS	Access to RSNA CTP web interface
Web browser from VPN access	VM(1)	8080	HTTPS	
Web browser from hospital network	VM(1)	6901	HTTPS	Access to Guacamole interface to use the RSNA CTP Launcher.jar program
Web browser from VPN access	VM(1)	6901	HTTPS	
VM(1)	VM(2)	8500	HTTPS	API access from Radiomics Enabler® to Indexa™
Web browser from hospital network	VM(1)	9050	HTTPS	Access to application eZCRF™"on premise" web interface
Web browser from VPN access	VM(1)	9050	HTTPS	
VM(1)	VM(1)	9050	HTTPS	API access from Radiomics Enabler® to eZCRF™"on premise"

VM(2): Indexa™(and Orchestra™if needed)				
SSH access from VPN access	VM(2)	22	SSH	SSH access for support
VM(2)	Hospital PACS	As specified	DICOM	Requests c-echo, c-find to the PACS
Hospital PACS	VM(1)	11112	DICOM	Medexprim Suite receive PACS information

Web browser from hospital network	VM(2)	8500	HTTPS	Access to Indexa™web interface
Web browser from VPN access	VM(2)	8500	HTTPS	
VM(2)	VM(1)	8000	HTTPS	API access from Indexa™to Radiomics Enabler®
VM(2)	VM(1)	9050	HTTPS	API access from Indexa™to eZCRF™"on premise"
Web browser from hospital network	VM(2)	8500	HTTPS	Access to Orchestra™admi n panel and web interface
Web browser from VPN access	VM(2)	8500	HTTPS	
VM(2)	VM(2)	5432	TCP	PostgreSQL access from Orchestra™to Indexa™
VM(2)	VM(2)	8500	HTTPS	API access from Orchestra™to Indexa™
	VM(2)	9450-9500	MLLP	Ports used by Orchestra™pipeli nes
PACS or RIS	VM(2)	6666	MLLP	Messages HL7 v2
PACS or RIS	VM(2)	6667	HTTPS	Messages HL7 v3
Statistics				
Web browser from hospital network	stats.medexprim.com	443	HTTPS	Gathering non-personal generic data for statistical purposes
VM(1) + VM(2)	stats.medexprim.com	443	HTTPS	
For data intermediation projects supported by Medexprim				
Web browser from hospital network	https://chaimeleon-eu.i3m.upv.es	443	HTTPS	Export for project Chaimeleon
Web browser from hospital network	rwide-ezcrf.medexprim.cloud	443	HTTPS	
VM(1)+VM(2)	rwide-ezcrf.medexprim.cloud	443	HTTPS	
VM(1)+VM(2)	rwide-imaging.medexprim.cloud	22	SFTP	S3 Object Storage

VM(1)+VM(2)	sftpgo.medexprim.net	22	SFTP	Storage for de-identified data regarding the project (S3 Storage)

4. PACS configuration

Provide the PACS **DICOM c-echo, c-find, c-move SCP** parameters to Medexprim to configure Medexprim Suite™.

Declare on the PACS Medexprim Suite™as **SCU c-echo**, **c-find**, **c-move** and **SCP c-store** with the following parameters:

- **AETitle: RADENAB**
- **IP: IP address of VM(1) server**
- **Port: 11112**

- **AETitle: INDEXA**
- **IP: IP address of VM(2) server**
- **Port: 11112**

A « test patient » exam must be provided for support purposes.

Flow matrix of Medexprim Suite™
The PACS viewer parameters can be provided to Medexprim to allow Medexprim Suite™to use it during a search • URL to call the PACS viewer • Parameters of PACS viewer to include in URL for study and series level)

DI3. *MainSEL*

Name of the tool: MainSEL

Contributor: DKFZ

Area: de-identification

Tool state: Completed

Additional resources:

- https://github.com/medicalinformatics/mainsel
- https://github.com/medicalinformatics/SecureEpilinker

DI3.1. Conceptual description

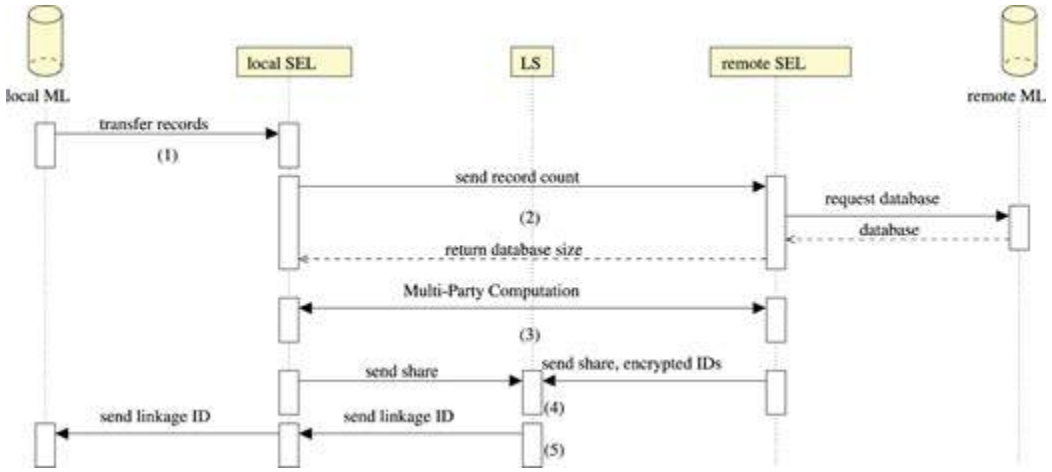
Tool description
Short for "Mainzelliste Secure EpiLinker"; an extension to Mainzelliste to perform Record Linkage using Secure Multi-Party Computation, thus without revealing input data.

Data
Patients' identifying attributes

Input/output
Input: Linkable metadata (e.g. identifiers, pseudonyms, identifying data).
Output: Linkable Pseudonyms, linkage weights.

Methodology/performance
MainSEL consists of the Mainzelliste as the data source and and Secure EpiLinker as the secure multi-party computation unit. Because secure multi-party computation is used there is no need for a trusted third party.

Use
MainSEL uses the same algorithm for determining the similarity score between two records as Mainzelliste. For field comparison either simple equality or Dice-coefficients of Bloom filter are used. Because it had been shown that using different protocols for different kinds of calculations with intermediate conversions may be more efficient than staying in the same protocol, even if this incurs additional conversion costs, four variations of the circuit are implemented, choosing different sMPC protocols for Boolean/logic and arithmetic components of the circuit, with possible conversions in between where necessary.



DI3.2 Technical description

CPU or GPU use: CPU
Main programming language: C++
Running mode (interactive/batch-based/case-based…): interactive
Software version: v1.0.1
Does the tool require administrator privileges?: No

Unitary tests
https://github.com/medicalinformatics/SecureEpilinker/tree/master/test
https://github.com/medicalinformatics/SecureEpilinker/tree/master/test_scripts

Access restriction
The tool doesn't have any access restriction.

Containerization
The tool is dockerized

Additional information
https://academic.oup.com/bioinformatics/article/38/6/1657/5900257

DI4. *Mainzelliste*

Contributor: DKFZ

Area: de-identification

Tool state: Completed

Licence: AGPLv3

Additional resources:

- https://bitbucket.org/medicalinformatics/mainzelliste/src

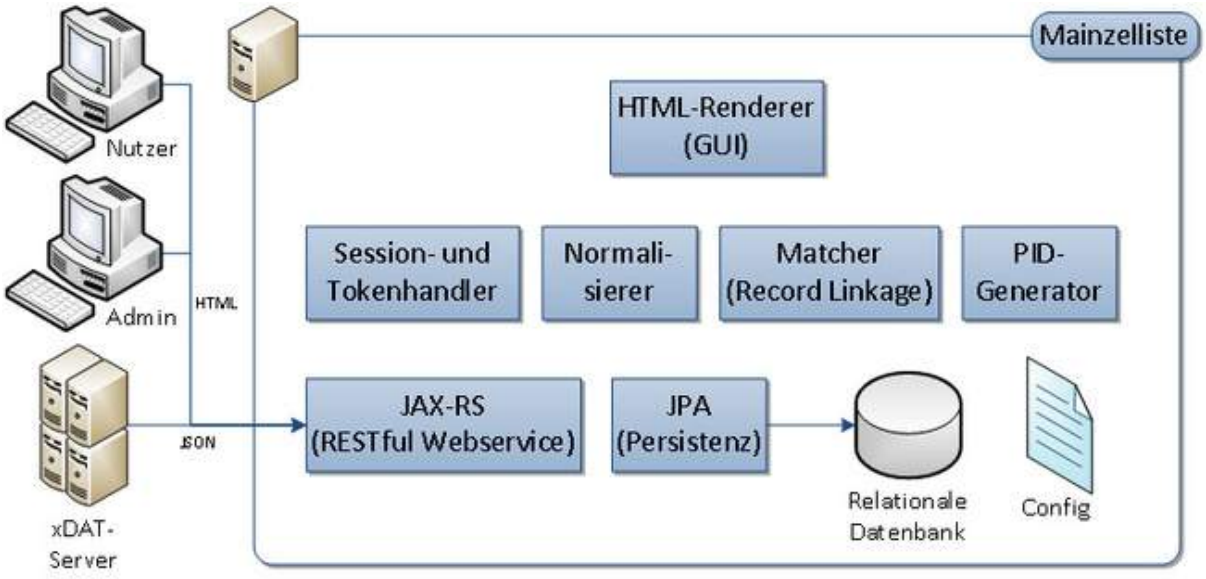
DI4.1. Conceptual description

Tool description
Open-Source web-based pseudonymization and record linkage solution in use and co-developed at many institutions; see code repo.

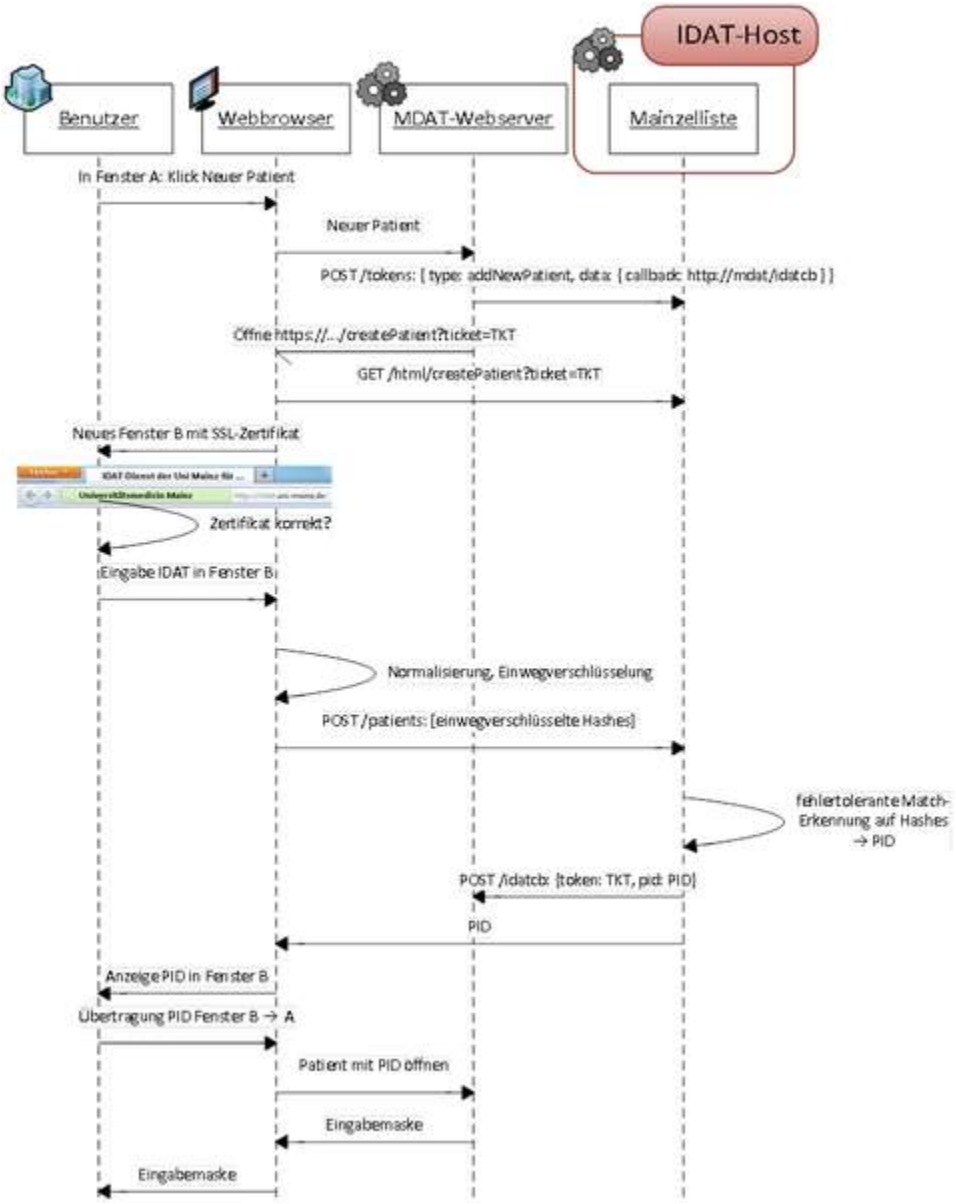
Data
Patients' identifying attributes

Input/output
Input: Linkable metadata (e.g. identifiers, pseudonyms, identifying data).
Output: Linkable Pseudonyms, linkage weights.

Methodology/performance
Mainzelliste is a web-based first-level pseudonymisation service. It allows for the creation of personal identifiers (PID) from identifying attributes (IDAT), and thanks to the record linkage functionality, this is even possible with poor quality identifying data.



Mainzelliste methods are accessed through can be accessed through the REST interface, which allows an easy integration within other tools.



Use

Main functionality of Mainzelliste:

- Pseudonym generation – substitution of identifying personal data with the string not related to the original data

- Record Linkage – for each patient there is only one pseudonym of a certain type which is generated, even if there are small errors in the identifying data. Near the record linkage based on the raw identifying data, Mainzelliste also allows the privacy preserved record linkage based on the irreversibly transformed identifying fields and the secure

multiparty computation where the identifying data never leaves the storage.

- REST Interface – allows the connection of Mainzelliste to different systems, such as registrys, biobanks, EDC and study management systems.
- Graphical interface – allows easy use of Mainzelliste functions without showing the complex workflow behind them.
- Backward compatibility
- Audit Trail

For each patient entered, Mainzelliste creates one or more non-speaking so-called personal identifiers (PID) that are compatible with the identifiers of the original PID generator. These deterministically created 8-digit strings are appropriate for use on the Web as well as for manual data transfer, as they can identify up to two typos.
For each patient there should only be exactly one pseudonym created, even for multiple entries of the patient. Therefore, when a patient is added the database is examined to see if this patient already exists. Thanks to a modular record linkage system that can be adapted to the demands of specific uses through a configuration file, this is possible even in the event of typos or alternate spellings. Particularly new compared to the PID generator is the possibility of using in-house phonetic codes and string comparisons, thereby allowing names from other linguistic backgrounds to be fault-tolerantly compared. Currently, weight-based record linkage is supported, but the modular concept allows for retrofitting an in-house algorithm. The possibility to manually re-work uncertain assignments further supports the automatic matching process.

DI4.2. Technical description

CPU or GPU use: CPU
Main programming language: Java
Running mode (interactive/batch-based/case-based…): interactive
Software version: 1.12.1
Does the tool require administrator privileges?: No

Unitary tests
https://bitbucket.org/medicalinformatics/mainzelliste/src/master/test/de/pseudonymisierung/mainzelliste/

Integration tests
https://bitbucket.org/medicalinformatics/mainzelliste/src/master/ci/newman_tests/

Access restriction

The tool doesn't have access restrictions

Containerization
The tool is dockerized
Additional information
https://www.youtube.com/watch?v=csiHrXsshPc
https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-014-0123-5

DI5. Dicom2usb DICOM Defacing, DICOM Exporter and Anonymizer tool

Contributor: Dicom2usb

Area: On-site export, de-identification, and defacing of DICOM images

Tool state: In development/Developed/Containerized

Additional resources:

- https://www.dicom2usb.com/
- https://docs.google.com/presentation/d/1-1G2r7hWLUfNgZMzO_s98nKcqycnILgY/edit?usp=sharing&ouid=101278778279669990486&rtpof=true&sd=true

DI5.1. Conceptual description

Tool description
Dicom2usb has a plug-in architecture. This allows the user to write own scripts that are automatically executed after a series has been received and anonymized. These scripts run outside Docker, allowing easy development and deployment.

Additionally, the plug-in feature could conceivably be used also on a central site, if data is via VPN or DICOM TLS from the hospital on-site dicom2usb.

The defacing is an example of such a plug-in script.

Using Dicom2usb at the hospital has the advantage that it negotiates file type with the sender (multiple options: for instance asking for non-JPEG DICOM compression, and Explicit Transfer Syntax). Both these features make further processing much more robust, making sure that other scientist's code can open the images. This also makes it robust to unknown DICOM elements

Data

Dicom2usb is aimed to fully anonymize any DICOM data except data containing burned-in text in images (thus, not Ultrasound). Defacing of DICOM images is included, and under continuous development.

Input/output

Input alternative 1) DICOM data can be transferred directly on-site to the Dicom2usb anonymize using DICOM sending. Dicom2usb allows complete DICOM anonymization of both header elements and face features in the image.

Input alternative 2) DICOM data in a folder can be manually anonymized as in alternative 1) above

Output: Anonymized DICOM data is stored in an output folder, with a configurable folder structure*. If defacing is used, defaced DICOM files are stored in a separate folder. After manual validation of defaced DICOM files, the validated files are stored in a third folder.

Output to remote: Anonymized data can be automatically / manually transferred to a remote server using DICOM transfer or SFTP. In the case of manual validation, the transfer to a remote server is initiated manually.

*NOTE***:** Dicom2usb negotiates transfer syntax and explicit VR with the sender. allowing for a consistent data set regardless of source. This means that for instance odd JPEG-compressions may be disallowed. Also allows for more general anonymization of private tags.

** Any combination of DICOM elements post anonymization may be used to build the output folder structure. For instance: "/DEFACED_DICOMFILES/ANON-0003 (2015-OCT-15) - 739702/[MR] MT OFF_ON - serie601/". Obviously the date seen here is not the true date.*

Methodology/performance

Anonymization on a normal performance Linux server runs at about 600 image files per second, including DICOM transfer and anonymization. Using Defacing, a series can be defaced between anything from a few seconds to close to a minute.

The tool is intended to be deployed on premise, so that all transfer of non-anonymized data stays behind the firewall, and under full control of the local IT department. The tool could optionally be used at EUCAIM-image central repository, for the defacing step.

The tool is deployed using Docker, containing one container for anonymization, one for web interface, and one for handling queues. The anonymization software is written in Java. Defacing is performed with the ITK framework. The software is recommended to be deployed on a Linux server with Docker and Docker-compose installed. Software is self-contained in the containers, but with startup scripts external. Defacing is an additional Docker container with its own startup scripts. The Docker containers are created for Intel 64-bit architecture, but it is possible to create them also for Arm architecture.

The anonymization, defacing and imaging are handled in a consistent expeirence, but internally handled by different parts of the software :

- The anonymization work flow is set up once and for all, and requires no further interaction than to add designated study codes for each individual patient.
- Study codes are added through a web interface, before pushing the DICOM images from the PACS, workstation etc
- DICOM images are received through a network transfer. The DICOM headers are anonymized and new DICOM files are written to a folder 'DICOMFILES'. When network transfer is used, no non-anonymized images are saved.
- If input is DICOM images stored on disk, they are internally transferred by a DICOM transfer, and lands in a folder 'REPROCESSED_DICOMFILES'.
- The defacing plugin (if enabled) is automatically executed when new images are transferred, and performs defacing by matching to template head images, and removing a predefined mask from the original DICOM image. The defaced new files are written to a new folder 'DEFACED_DICOMFILES'. At the same time, preview images in axial, sagittal and coronal planes are created centered at the mask position. These preview images show both the defaced image, and a second set of preview images show the original image with a transparent yellow mask ontop.
- A web server (another container) is used for user interactaction with the data on the Dicom2usb. The results from above mentioned folders can be browsed, so that the anonymized DICOM headers can be viewed, and the 'DEFACED_DICOMFILES' folders can be viewed with preview images. Hovering a preview switches between the defaced preview and the preview with yellow transparent mask. Pressing a 'Validate' button moves the series to a folder 'VALIDATED_DEFACED_DICOMFILES'
- The 'VALIDATED_DEFACED_DICOMFILES' folder would be the result data for a defacing workflow. Without defacing, the 'DICOMFILES' folder will be the result. Data can be further pushed to a destination with sFTP or DICOM transfer.
- The following DICOM application profiles are used and documented by populating the '*De-identification Method Code Sequence (0012,0064)'* :
  Basic Application Confidentiality Profile,
  Clean Graphics Option,
  Retain Patient Characteristics (weight for PET, can be turned off),
  Remove unspecified tags,
  Retain Safe Private Option,
  Retain Longitudinal Temporal Information With Modified Dates,
  Clean Recognizable Visual Features Option
- Starting and stopping Dicom2usb and its internal containers is performed with a single command.
- Multiple projects can have their own Dicom2usb nodes with their own AE titles, and ports, and can run simultaneously listening for data. This means that Dicom2usb is just a set-up and forget infra-structure.

Use

The Dicom2usb is a containerized dicom node for the hospital network, to which images can be sent using DICOM network protocol from PACS, workstations etc.

Image files are received and anonymized in a flow before, written to disk. Thus no personal information is stored on disk.

A combination of VR-based rules, and explicitly defined anonymization rules cover all standard DICOM elements. Also, VR-based rules may be applied to anonymize private DICOM elements

Dicom2usb allows for multiple projects to be set up at the same hospital server, with separately configured anonymization, post-processing, and data-sending rules

Qualitative results
Defacing is validated through the web interface of the tool (on the local network). A 3-view preview in axial, coronal and sagittal directions show the defaced part of the image. The user validates each series quickly from this previous by pressing a button to move the validated series to the results folder.

DI5.2. Technical description

CPU or GPU use: CPU
Main programming language: Java
Expected RAM usage: 1GB
Running mode (interactive/batch-based/case-based…): batch-based/case-based (interactive for validation)
Does the tool require administrator privileges?: Yes, as is

Unitary tests
Integration tests are performed covering sending data all the way to proper DICOM anonymization of header elements. Manually validated test cases has been used to automatically check for changes in behavior

Access restriction
At this time the source code is closed source, but we are open to discussion of different models.

Containerization
Yes. Three Docker containers are spun up with a startup script for each project. Started containers are automatically restarted after server reboot.

As stated previously, Dicom2usb allows for multiple projects to be set up at the same hospital server, with separately configured anonymization, post-processing, and data-sending rules. Each project starts three containers, and we have not seen any detrimental effects on servers with tens of projects.

Dicom2usb has a plug-in architecture. This allows the user to write own scripts that are automatically executed after a series has been received and anonymized. These scripts run outside Docker, allowing easy development and deployment.

Additionally, the plug-in feature could conceivably be used also on a central site, if data is via VPN or DICOM TLS from the hospital on-site dicom2usb.

The defacing is an example of such a plug-in script.

Using Dicom2usb at the hospital has the advantage that it negotiates file type with the sender (multiple options: for instance asking for non-JPEG DICOM compression, and Explicit Transfer Syntax). Both these features makes further processing much more robust, making sure that other scientist's code can open the images. This also makes it robust to unknown DICOM elements

3 - Data quality and curation tools validation documentation

DQ1- DICOM_file_integrity_checker

Contributor: HULAFE

Area: Imaging data quality assessment and imaging data curation

Status : Containerized

Purpose : Medical imaging data organisation, preprocessing and quality assurance at DICOM tags level, generating a report containing information about the selected sequences, corrupted and missing files and other relevant information about the process performed.

DQ1.1. Conceptual validation

Tool description:
This is a Python script that selects DICOM files of specific sequences from an input directory based on a configuration file and copies them into a new directory, performing several data preprocessing and quality assurance. The input directory can be at any organisation level (structured, unstructured, with several subdirectories or not…), as the tool goes recursively through the entire contents of the input folder until it reaches the .dcm files.
The main functionalities are:

- Selection of sequences according to a previous list of sequences by manufacturer and sequence type (T2, DWI, DCE, T1w, etc.) compiled in a catalogue.
    - You can adapt the catalogue file (there is an example of catalogue.xlsx for Prostate Cancer imaging provided by HULAFE, but it can have another name if specified in catalogue_file in the configuration parameters) to the sequence names you know in your specific project/machine/institution. It can be provided either in .xlsx or .json format.
- Identification of missing and corrupted files. In the case of missing files, the number of theoretical images (that can be consulted in the DICOM headers) is compared with the actual number of images in the directory. In the case of corrupted files, an attempt is made to decompress the DICOM files and do a python preload, without checking any specific DICOM tag.
- Merging sequences that are separated into several ones and should not be, i.e., multivolume sequences (especially dynamic ones such as DCE) that are sometimes separated by temporal instances.
- Identification of diffusion sequences by different B values in the same sequence (low and high, commonly 0 & 1000 or 0 & 1500).
- Reporting useful information from the dataset and the QA process performed.
- Parallelization of the process to speed up the execution.

Optional functionalities are:

- Only report generation by analysing the input dataset but without copying any dcm file as output (faster).
- Extraction of the patient code / pseudo-anonymized code from the input folder for cases in which patientID and patientName DICOM tags are empty.
- Adjustment of the minimum number of sequences needed for understanding as a multivolume one separated that must be merged.
- Matching of the SeriesDescription in DICOM file with the one in the list of the catalogue file. By default only try to find a string between the other one.
- Change of the SeriesDescription tag of the files to a generic one if needed (e.g., change "GE_axial_t2W_con_contraste" to "T2").
- Change of the Series folder name of the files to a generic one if needed (e.g., change "GE_axial_t2W_con_contraste" to "T2").

- Introduction of the B values of a diffusion sequence into the sequence name (in seriesDescription tag or series folder name depending on the option chosen with the other arguments).
- Introduction of preferences about the desired diffusion sequence depending on the B values. Example: [0-1000, 0-800...] you sort the b-value list according to your preferences. It tries to find the b-value_max of the first option. If not found, it is searched for the next option, and so on. In case there is no exact match, the closest to the first option is kept.
- Selection of how choosing the possible dwi: from catalogue names, from b_values preferences indicated or addressing both cases. Example: in the catalogue you can have several options for the prostate acquired in several B fields but you do not want sequences centered in ganglia (so they are not in te catalogue) and you prefer a b_value preference selected in the list given in the json of optional_parameters.

What is the invention? What is its objective?
It involves the selection of sequences and quality control of medical image repositories. Its objective is to prepare folders before processing image studies or sharing their content, offering the possibility to select desired sequences for each project and ensuring that their content is correct.

What unmet need or relevant problem does it solve?
Ability to perform automatic quality control of image studies by accessing DICOM header information and obtaining the necessary information to identify sequences, merge them if necessary, and detect corrupted files.

Innovations and Advantages:
The current development provides great flexibility and the possibility of incorporating it into Medical Imaging workflows in a lightweight and robust manner (dockerization).

Data: DICOM MR, DICOM US, DICOM RTDOSE, DICOM RTSTRUCT, DICOM CT, etc.

Methodology/performance:
The development process, conducted in Python, utilized libraries such as NumPy, Pandas, PyDicom, and incorporated the MPire library for parallelization. The exhaustive process involved analyzing DICOMs, identifying common issues for different use cases, and automating codes. Given an anonymized raw dataset, the tool verifies its correctness for AI analysis, ensuring the correct DICOM format. Visual quality at the pixel matrix level will be addressed in future iterations or related tools. The final production result has been Dockerized, facilitating integration into any system by specifying the input directory and configuration parameters. The output includes an analysis report and an optional copy of the processed directory.
Information on the performance of this tool can be consulted through the messages that are printed during its execution or in the log file that is generated at the end (depending on the

debug option chosen). In both cases, both the start time, end time and duration of the execution are detailed, as well as details on the number of DICOM files processed and their speed, with a progress bar.

Use: brief description of the tool's functioning

The **select_scope_sequences** function performs the following steps:

1. Loads the configuration file and extracts the selected sequences based on the user's input.
2. Initializes several worksheets in an Excel file using the xlsxwriter package.
3. Loops over all DICOM files in the input directory, reads their metadata using the pydicom package, and organizes them based on the patient, study, and series.
4. Filters the series based on the selected sequences and copies the DICOM files for the selected series to the output directory.
5. Detect diffusion sequences by different B values in the same sequence (low and high, commonly 0 & 1000 or 0 & 1500) and include them.
6. Merges sequences that are separated into several ones and should not be, i.e., multivolume sequences (especially dynamic ones such as DCE) that are sometimes separated by temporal instances.
7. Generates an Excel report containing information about the selected sequences, corrupted files, missing files, and timepoints.

Input/output formats:

Input: DICOM files of specific sequences

Output: processed dataset and an Excel report containing information about the selected sequences, corrupted files, missing files, timepoints (studies) found, and merged sequences if applicable.

Qualitative results: Provide some visual results (if available) of applying such tools.

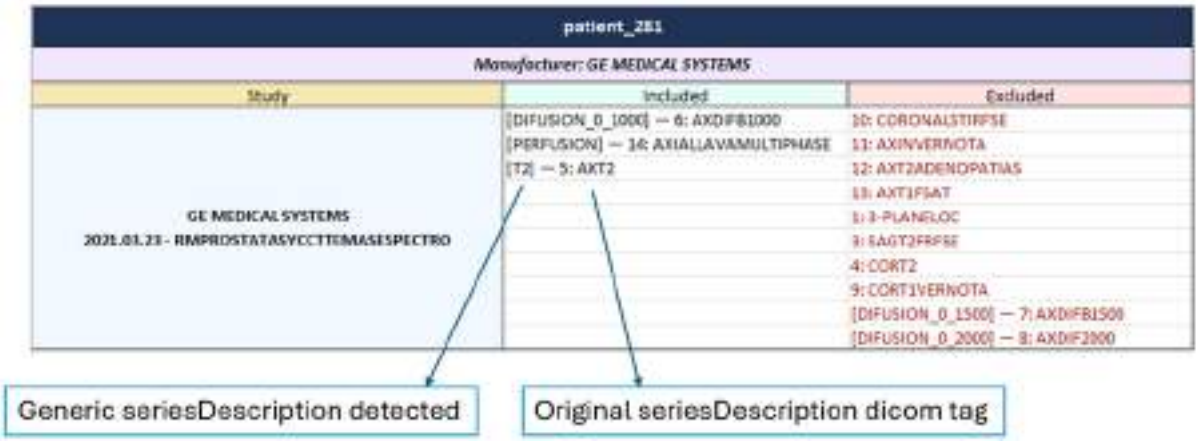


Figure 1. Included/excluded sequences description in report.

Patient	Date	Manufacturer	Study	Series	Series Number	# theoretical images	# Real images found	# Missing
patient_407	2015.12.15	GE MEDICAL SYSTEMS	RZ515C-RMESTADIAJEGENITALESMASCULINOSCTE	AXT2PROSTATA	7	28	13	15
patient_417	2015.12.15	GE MEDICAL SYSTEMS	RZ515C-RMESTADIAJEGENITALESMASCULINOSCTE	PERFUSION+C	13	1080	72	1080
patient_258	2019.01.28	GE MEDICAL SYSTEMS	RMPROSTATAIVCETTEMASSESPECTRO	AXDIFB1000	13	84	42	23

Figure 2. Missing images description in report.

Patient	Date	Manufacturer	Study	Series	Series Number	file_name	file_size
patient_335	2016.01.04	SIEMENS	RZ515C-RMESTADIAJEGENITALESMASCULINOSCTE	PHOENIXZIPREPORT	99	******.dcm	8 KB
patient_335	2016.01.04	SIEMENS	RZ515C-RMESTADIAJEGENITALESMASCULINOSCTE	PHOENIXZIPREPORT	99	******.dcm	8 KB

Figure 3. Corrupt images description in report.

#	ID	TIME_POINT	DATE	T2	DIFUSION	PERFUSION
0	patient_341	1	06/09/2018	1	1	1
	patient_341	2	28/06/2022	1	1	1
1	patient_335	1	04/01/2016	1	0	1
2	patient_282	1	11/12/2018	1	1	1

Figure 4. Desired timepoints detected description in report.

patient_335		
Manufacturer: SIEMENS		
Study	Sub-series to merge	Output merged series name
SIEMENS 2016.01.04 - RZ515C-RMESTADIAJEGENITALESMASCULINOSCTE_20160104	[37]-T1_VIBE_FSTRA_DYN	T1_VIBE_FSTRA_DYN_MERGED
	[36]-T1_VIBE_FSTRA_DYN	
	[27]-T1_VIBE_FSTRA_DYN	
	[23]-T1_VIBE_FSTRA_DYN	
	[32]-T1_VIBE_FSTRA_DYN	
	[24]-T1_VIBE_FSTRA_DYN	
	...	
	...	
	[18]-T1_VIBE_FSTRA_DYN	

Figure 5. Merged series example in report.

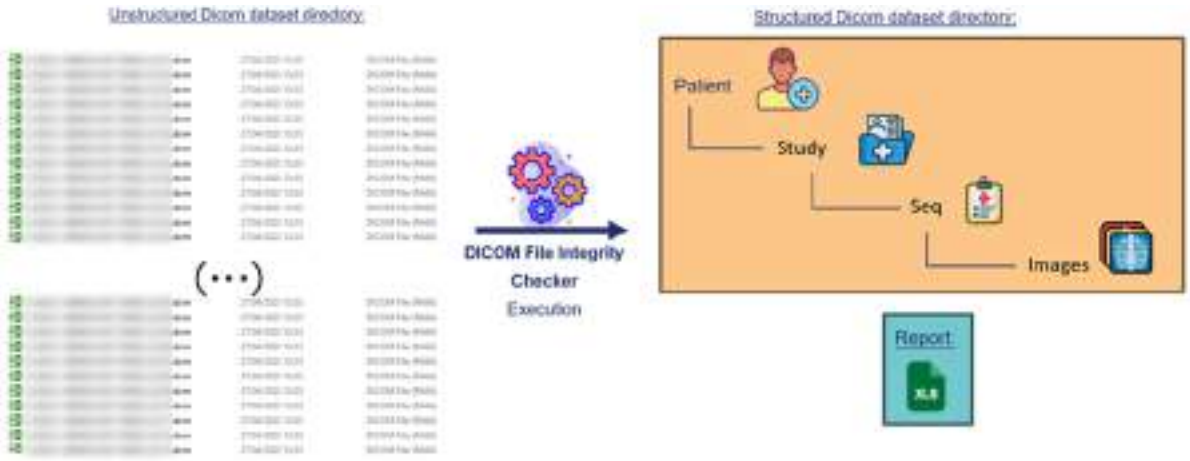


Figure 6. Comparative between input & output of the tool.

Additional information:

- **Successful use cases:** HULAFE cases from PROCANCER-I, merged Perfusion (DCE-Dynamic Contrast Enhancement MRI) sequences of PRIMAGE, all PROCANAID cases (Project PLEC2021-007709 ProCanAid funded by Spanish Science & Innovation Agency).
- **Licence :** This project is under licence at [IISLAFE]
- **Acknowledgements:** This script was developed and continuously improved with new features by [Pedro Miguel Martinez & Adrian Galiana]. Reviewed and adapted to Airflow pipelines by [Carina Soler]. All done at [GIBI230] Research Group Environment, at [La Fe Health Research Institute - LA FE HOSPITAL VALENCIA].
- **Keywords for searching in databases:** sequences, DICOM, quality control, headers, QA, QC, imaging integrity, data completeness.

DQ1.2. Technical specifications

Data: In depth description of the data used to train the tool.
This tool is not a model trained; nevertheless, the data used to design, test, and utilize the tool in production research environments were standard 2.0 DICOM files extracted from several imaging equipments such as CT, MR, and PET/MR from GE Healthcare, Siemens Healthineers, and Philips manufacturers.
Methods: in depth description of the methodology used for its development including all data preprocessing.

- **Installation**

1. Clone the repository

2. Build the Docker image. Assuming you have your main.py file and Dockerfile in the same directory, you can build the Docker image with the following command: *docker build -t my_image:version .*

This will build the Docker image with the name my_image and the tag version from the current directory (.) that contains your Dockerfile.

- **Usage**

1. Set up a folder where your project is (<input_path>), with a folder for input images and, if desired, a different one for output. Specify the path names in the parameters file if they are different from the default ones.
2. Create a directory for the configurations (<config_path>) and inside it fill the file with the configuration parameters file (by default, parameter_configuration.json as this example) and the catalogue file editing it if necessary (e.g., catalogue.xlsx). You can also use a json file for the catalogue (e.g., catalogue.json).
   2.1. Introduce new sequence names into the list or edit. You can add rows maintaining the format, also columns for new sequence types.
   2.2. Change the titles of the sequence columns if you desire another, like DIFFUSION or DW or DWI. Change accordingly to the sequence_selection in parameter_configuration.json.
   2.3. Choose "sequence_selection": ["ALL"] if you want to pass every sequence from the input to the output folder. Or choose only the types you want (so exclude the others).
3. Be sure that parameter_configuration file is edited as you desire, if it is empty it will take the default values. Maintain the name of the file as it is defined in the environment file (.env) or change it in the environment file (not recommended).
4. Run the Docker container with the built image:

docker run -it --rm --name my-container --env-file=.env -v "<input_path>:/Proyecto" -v "<config_path>:/Parameters_config" my_image:latest

This will run your main.py script within the Docker container creating the specified volumes.

Note that in the above example, the -it option is used for an interactive session, and the --rm option is used to remove the container after the script has been executed. The configuration parameters (all are optional arguments) will be defined in a JSON file (e.g., parameter_configuration.json) in the config_path and you can adjust them according to your needs. In addition, with the --env-file option you specify the name of the file with the environment variables needed for the code execution, which are the directories of the docker containers where the volumes are created and the name of the file with the configuration parameters. These parameters can be modified from the environment file itself (.env) or changed when running the container, with the -e option and a new parameter value (e.g. -e CONFIG_FILE=new_configuration.json), but it is not necessary.

- **Optional parameters in configuration JSON file:**

  - catalogue_file: (optional) name of the file with the sequence catalogue. It can be an .xlsx or a .json. By default it is catalogue.xlsx. (string)
  - sequence_selection: (optional) name of the sequences to be selected, according to the columns of the catalogue in excel format or to the sections of the catalogue in

json format. If you want to select all of them, you must write: "ALL". By default: ["ALL"]. (list of strings)

- input_directory: (optional) directory name within the project folder where the patient images you want to use as input are located. By default, if nothing is specified, it will be the "INPUT" folder. (string)
- output_directory: (optional) directory name within the project folder where you want to save the selected sequences. By default, if nothing is specified, it will be overwritten in the "OUTPUT" folder. (string)
- only_report: (optional) if set to True, only a report will be generated instead of the output directory with the dicom files included and the report. By default: False. (boolean)
- new_report_name: (optional) parameter that allows to specify a different name for the report. The default name will be report_dicom_studies_qa_{now}.xlsx, but can be changed to {new_report_name}_{now}.xlsx.
- id_from_folder: (optional) if True, take Patient ID from folder (for cases where PatientID dicom tag is generic or anonymized)(files must be by hierarchy 'patient/study/series/.dcm. By default: False. (boolean)
- num_min_dyn_seq: (optional) The minimum number of dynamic sequences that must be found to consider an image series as relevant. The default value is 3. (int)
- force_sequence_name: (optional) If set to True, sequences will be forced to have specific names instead of generic names. By default: True. (boolean)
- change_header_names: (optional) If set to True, generic names will be used for sequences. By default: True. (boolean)
- change_folder_names: (optional) If set to True, generic names will be used for the folders containing the sequences. By default: True. (boolean)
- b_values_in_name: (optional) If set to True, b values will be included in sequence names. By default: True. (boolean)
- b_value_choices: (optional) A list of b values to search for in sequence names. (list of int)
- dwi_selection_source: (optional) 3 options to select which is the source for the possible diffusion series: "catalogue", "b_value_choices" or "both". (string)
- modalities_to_avoid_exclusion_corrupt_files: (optional) Name all modalities which are not able to open with pydicom library, but you know they are not corrupt files. Ex.: "RTSTRUCT", "RTDOSE" or "RTPLAN". (list of strings)

- modalities_to_include_everything: (optional) Name all modalities you want to include complete because series_descriptions do not follow a pattern (list of strings)
- dwi_quantity: (optional) Number of diffusion sequences to be selected. Options 1 to any number or "all" for letting pass anyone. By default: 1. (int)

All arguments have default values if not provided.

Specific Technical information:

1. CPU
2. Programming language : python
3. Expected RAM usage : depends on the data, the tool is designed to function across various specification levels. However, processing time may increase with lower specifications. It has been tested on both low and high specification machines*.
4. Running mode : batch-based
5. Software version : v1.2.9
6. Libraries : docker
7. Security measures: writing to host data restricted to a non-root user ; container does not require it to be executed in a privileged mode; any file/folder is moved or removed, it only has copying or reading capacities.

*The execution times of the tool were compared under the same configuration and launched on the same number of cases (10 patients with 1 study each, including all sequences) but across different execution environments, in terms of performance levels. Specifically, the main technical characteristics of the devices used for running the tool are detailed in the following table, being the last one a NVIDIA DGX™ A100 node with 8 Graphic Processing Units (GPUs) installed in a high-performance computing (HPC) infrastructure of a biomedical imaging research group.

Device Type	Operating System	RAM	CPU	CPU speed	Cores
Developer Laptop	Windows 10 Pro	32 GB	11th Gen Intel® Core™ i7-1165G7	2.80 GHz	4
Hospital Server	Windows 10 Pro	32 GB	Intel® Core™ i5-4590	3.30 GHz	4
HPC Node	Ubuntu 20.04.6	2 TB	AMD EPYC 7742 64-Core Processor	3.94 GHz	128

The results of this comparison were as follows:

Server	Number of cases	Configuration	Execution time
Developer Laptop	10	All sequences	4 min 57 s

Hospital Server	10	All sequences	14 min 8 s
HPC Node	10	All sequences	2 min 18 s

Traceability and monitoring mechanism.
The tool currently has some formal traceability or monitoring mechanisms. One of the outputs is an Excel file containing detailed information from the analyzed directory. This file provides a comprehensive overview of what the tool has encountered and observed in the directory. Additionally, the tool registers events and errors on a log file. Future iterations may consider incorporating other expected mechanisms based on CPU and memory usage, etc.

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
Functional and Boundary Tests were implemented and Input Validation Tests and Error Handling Tests were partially performed. The functionality of individual components or modules has been tested to ensure they operate as expected. The behavior of the tool at boundary conditions, such as maximum/minimum input values or edge cases, has been assessed. However, Verification of how gracefully the tool handles various input types and provides appropriate error messages for invalid inputs should be improved as well as more extensive validation of the tool's response to unexpected errors or exceptional conditions, ensuring it fails gracefully and provides informative error messages.
Access restriction : do you have any access restriction to the source code or to the binaries of the tool?
Considering that our tool is licensed by our institution and is currently undergoing scientific publication and patentability study, this tool can be operated under the EUCAIM guidelines; however, access to the source code or the binaries is not yet permitted.
Additional information for tool integration.

- Compatibility: The tool has been designed to ensure compatibility with other software components, frameworks, or platforms it interacts with as it is dockerized.
- Documentation: comprehensive documentation covering installation instructions, usage guidelines has been provided to facilitate integration and usage by EUCAIM platform users.
- Scalability: the tool has been designed to scale efficiently to accommodate varying workloads and datasets, considering factors such as parallel processing and distributed computing where applicable.
- Maintenance: no plans are currently in place for establishing a maintenance plan for regular updates, bug fixes, and support channels to address user inquiries or issues. However, future versions may be released to address identified flaws or to introduce new and improved features. It is anticipated and desirable, though not guaranteed, to inform and share such updates with EUCAIM platform.

DQ1.3 Integration specifications

Communication channel for the helpdesk, technical support channel

Via e-mail to pedromiguel_martinez@iislafe.es ; adrian_galiana@iislafe.es ; carina_soler@iislafe.es ; pedro_mallol@iislafe.es or leonor_cerda@iislafe.es

Most common errors

- **Incorrect Configuration File Format:** Errors may arise if the configuration file is not correctly formatted or if mandatory fields are missing. Ensure the file adheres to the specified JSON or Excel format and includes all necessary parameters.
- **Missing Catalogue File:** The tool requires a catalogue file to identify and process sequences. If this file is missing or incorrectly named, the tool will not function correctly.
- **DICOM File Corruption:** Corrupted DICOM files cannot be processed. These files should be identified and removed or replaced with intact versions.
- **Unknown DICOM Modalities:** Certain DICOM files, especially from newer or less common imaging modalities, may not be detected if you do not identify the Series Description tags before and add to the catalogue.
- **Insufficient Permissions:** Lack of file or directory access permissions can prevent the tool from reading input data or writing outputs. Ensure the running environment has the necessary permissions.

Specific errors with the integration to the ChAImeleon platform

Several considerations have been encountered during the integration process with the ChAImeleon platform, and the following adjustments were made:

a) **Error**: The input and output folders on the ChAImeleon platform are predefined and located in separate directories.

Solution: The code was adjusted to define two different volumes and the paths can be specified as environment variables (*INPUT_PATH* and *OUTPUT_PATH*) .

b) **Error**: The original implementation of 'pathlib' library was inadequate as it did not handle symbolic links used to mount ChAImeleon datasets correctly and input DICOM files were not found.

Solution: 'pathlib' was replaced with 'os.walk 'to list directories and subdirectories.

c) **Error**: The tool failed to read certain DICOM files due to insufficient permissions in the input dataset.

Solution: Permission changes were requested to the ChAImeleon platform developers (UPV).

d) **Error**: There was no LibreOffice viewer available to open xlsx files (for the catalogue and the reports generated).

Solution: An HTML format report was generated as an alternative. The format of the reports can be selected in the configuration parameter *report_type*.

e) **Error**: Excessive CPU thread usage could overload the server.

Solution: Adjustments were made to allow the number of CPU threads to be configured via a new parameter specifying the percentage of thread usage (*threads_percentage*).

f) **Error**: It was not possible to select modalities, only sequences.

Solution: A new functionality was added to select modalities and specify them in the configuration parameter *modality_selection*.

g) **Error**: Implementing the tool on the ChAImeleon platform could compromise the privacy of the source code.

Solution: Possible strategies for protection were analysed, and it was decided to use *Jobman* instead of *Udocker* to run the tool. Thus, the image is executed without the possibility of accessing the container's content, where the code is located.

FAQs

- **Can I process files from any DICOM-compatible imaging modality?** The tool is designed to process standard 2.0 DICOM files from common imaging equipment such as CT, MR, and PET/MR from manufacturers like GE Healthcare, Siemens Healthineers, and Philips. Compatibility with other modalities is mainly supported, provided they adhere to the DICOM standard compliance.
- **How do I add new sequence names to the catalogue?** You can add new sequence names by editing the catalogue file (either catalogue.xlsx or catalogue.json). Ensure to maintain the format and structure of the file.
- **What should I do if I encounter an unsupported file type?** If you identify a DICOM file type that the tool cannot process, please contact the helpdesk with the file type and any error messages received. Our team will investigate and provide guidance.
- **Can the tool be integrated with other software?** Yes, the tool is designed to be lightweight and compatible with various software frameworks, as it is containerized for ease of integration.

User Manual

This is the guide to follow in order to run the tool on the ChAImeleon platform, which simulates the EUCAIM central node in the first demonstrator of the pre-processing tools.

a) Installation/configuration instructions (only for downloadable tools)

Not-downloadable tool. The Docker image will be already built and available in the registry used by Jobman.

b) Usage instructions

i. Select a remote desktop application from the ones available on the Chaimeleon platform and choose the dataset you want to load in this environment (it will be available in the **~/datasets** directory and will be used as input). Once deployed, you can access it through Guacamole.

ii. Create a directory for the configurations in your **~/persistent-home** and inside it fill the file with the configuration parameters file and the catalogue file. You can create the json files directly in this environment or upload them from your local with a simple drag

and drop. Here you can find examples of these configuration files, although you should adapt them according to your needs:

1. Introduce new sequence names into the list or edit. In the excel file, you can add rows maintaining the format, also columns for new sequence types.
2. Change the titles of the sequence columns if you desire another, like DIFUSION or DW or DWI. Change accordingly to the sequence_selection in config.json.
3. Choose "sequence_selection": ["ALL"] if you want to pass every sequence from the input to the output folder. Or choose only the types you want (so exclude the others).

iii. Be sure that configuration file is edited as you desire, if it is empty it will take the default values. You can change the name of the file and specify it in the environment variable *CONFIG_FILE.*

iv. Run the tool:

jobman submit -i dicom-file-integrity-checker -- param1 -- INPUT_PATH=~/datasets OUTPUT_PATH=~/persistent-home/output CONFIG_PATH=~/persistent-home/config CONFIG_FILE=config.json

Note: the input and output paths must point to *~/datasets* and *~/persistent-home* respectively, as these are the default volumes used in the ChAImeleon platform containers. However, you can rename the persistent-home folders for output and configuration as you wish, depending on how you have created them.

c) Additional considerations

- Input/Output Description:
  - **Input:** The tool accepts standard 2.0 DICOM files from a variety of imaging modalities, including CT, MR, PET/MR, US and others. The input files can be organized or not therefore it is applicable to any TIER. Both .xlsx and .json formats are supported for the sequence catalogue and configuration parameters file.
    - Note: with the configuration parameter i**nput_directory** you can specify the name of the folder within the INPUT_PATH where the dataset is located (corresponds to the ID of the dataset).
  - **Output:** Outputs include a processed dataset where selected sequences are copied into a new directory, and a report summarizing the selected sequences, corrupted files, missing files, and any other relevant QA information.
    - Note: with the configuration parameter **output_directory** you can specify the name of the folder within the OUTPUT_PATH where you want to save the results, as well as the report_type and report_name for the generated reports.

- Preprocessing Requirements. Before using the DICOM file integrity checker, ensure that:
    - A valid catalogue file and a configuration parameters file are available and correctly formatted.
    - The system running the tool has adequate permissions to access the input data and write to the output location.
    - The number of cpu threads has been adjusted for your execution in case there are usage limitations.

- Mandatory and Optional Data:
    - **Mandatory:** Any DICOM files to read and analyze as INPUT.
    - **Optional:** Users can modify the configuration parameters file to adjust the tool's operation, such as selecting specific sequences or changing the output directory name, as well as many other additional parameters.

- Cases in Which the Tool Should Not Be Used.
    - When working with non-DICOM or proprietary-formatted files that do not comply with the standard DICOM 2.0 specifications.
    - When detailed pixel-level image analysis is required. The tool focuses on DICOM file integrity and sequence selection rather than image content analysis. In that cases, the tool can be used but these image-based analysis must be done after.

Integration tests : Description of tests for assessing the correct integration of the tool

Integration tests are conducted to ensure that the DICOM file integrity checker seamlessly integrates with other components of the medical imaging workflow and the EUCAIM platform. These tests include:

- Environment Compatibility: Verifying that the tool operates correctly across different computing environments, including various operating systems and hardware configurations.
- Workflow Integration: Testing the tool's ability to integrate with existing medical imaging workflows, including data ingestion, processing, and analysis stages.
- Performance Metrics: Assessing the impact of the tool on processing times and resource utilization in integrated scenarios to ensure it meets performance benchmarks.
- Error Handling: Evaluating the tool's behavior in integrated workflows when encountering errors, such as missing data or corrupt DICOM files, to ensure robust error handling and reporting mechanisms.

Results of non-functional tests

Non-functional tests focus on evaluating the tool's performance, scalability, and reliability. Results indicate that the DICOM file integrity checker:

- Performance: Demonstrates efficient processing of large datasets, with optimizations for multi-threaded environments to expedite data handling.
- Scalability: Scales effectively with increasing data volumes, maintaining performance levels due to its containerized architecture and parallel processing capabilities.
- Reliability: Shows high reliability in identifying and handling corrupt or missing DICOM files across various test datasets, with minimal false positives.
- Security: Adheres to security best practices by only reading and copying files, never removing or modifying the original data. The tool processes input information as provided; thus, if the input data is pseudo-anonymized or anonymized, the output will maintain this privacy level. However, it's important to note that the tool reads DICOM tags for patient ID and name. If the input data is not properly anonymized, this may be detectable in the output, aside from the folder or file names of the input, which could reveal whether the data is correctly anonymized or not.

DQ2. N4 Bias Filter

Contributor: FORTH

Area: Imaging data curation**Status** : Containerized

Purpose: The N4 bias filter tool aims to correct the bias field using the N4 filter and identify the optimal configuration of the N4 filter on prostate MR T2W images.

DQ2.1. Conceptual validation

Tool description: The tool is designed to perform image pre-processing to reduce the bias field effect, improving the quality of the image. Two main functionalities are offered: (1) Apply N4 filter to image/images (either with the default parameters values or with parameters values defined by the user) (2) Find the optimal configuration of the N4 filter for specific image/images by measuring the Full Width at Half Maximum (FWHM) of the periprostatic fat distribution.

Data: Magnetic Resonance Imaging (MRI) T2 weighted (T2W) prostate images.

Methodology/performance: The bias field correction is performed using the state-of-the-art method for bias field correction, N4ITK filter. The values of the parameters of the N4ITK filter can be either the default or can be specified by the user. The default values of the parameters of the N4 filter are the values that were identified as optimal through our exploratory analysis performed on 4 publicly available prostate datasets. More specifically, the default values of the parameters are: convergence threshold 0.001, shrink factor 2, fitting level 5, number of iterations 25 and the use of default mask (i.e., the non-zero voxels of the image). For the identification of the optimal configuration, an exploratory analysis examining 240 different configurations is performed for each image. For each examined configuration, the periprostatic fat distribution is identified and the Full Width at Half Maximum (FWHM) of this distribution is calculated. The filter configuration that results in the minimum FWHM for each patient is considered as optimum, indicating homogeneous tissue representation.

Use: brief description of the tool's functioning (if it applies).
The tool can be used for bias field correction of MR T2W prostate images. It aims to improve the quality of the image by reducing the intensity inhomogeneities caused by the low-frequency variation. Furthermore, it can be used to identify the optimal configuration (i.e. optimal values for each parameter) of the N4 filter for an T2W MR image or a batch of images. Thus, it can be used as a preprocessing step to reduce the bias field or identify the optimal configuration of N4 filter for MR T2W prostate images.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : The input of the tool is a path with the raw images, which should be in DICOM format.

Output : For the first functionality (apply N4 filter), the output is the N4 filtered images, which are saved to a path specified by the user with the same format. For each patient, a folder will be created and the N4 filtered image in DICOM format will be stored within the folder. For the second functionality (identify optimal configuration), a text or excel file containing the optimal configuration(s) is saved to a path specified by the user.

Quantitative results: performance obtained during training of the tool (if applies)

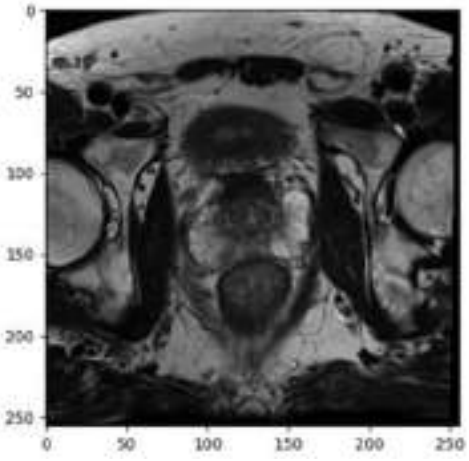
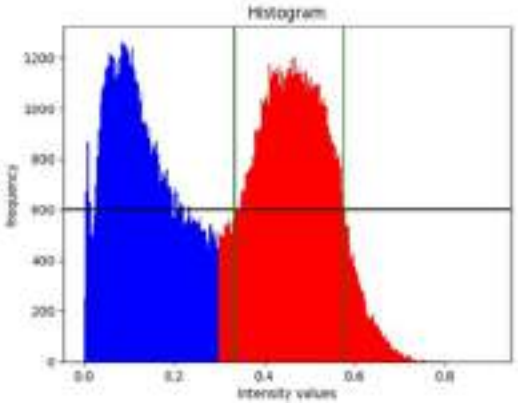
The results show that the set of N4 parameters, converging to optimal representations of fat in the image, were : convergence threshold 0.001, shrink factor 2, fitting level 6, number of iterations 100 and the use of default mask for prostate images acquired by a combined surface and endorectal coil at both 1.5T and 3T.

The corresponding optimal N4 configuration for MR prostate images acquired by a surface coil at 1.5T or 3T was: convergence threshold 0.001, shrink factor 2, fitting level 5, number of iterations 25 and the use of default mask.

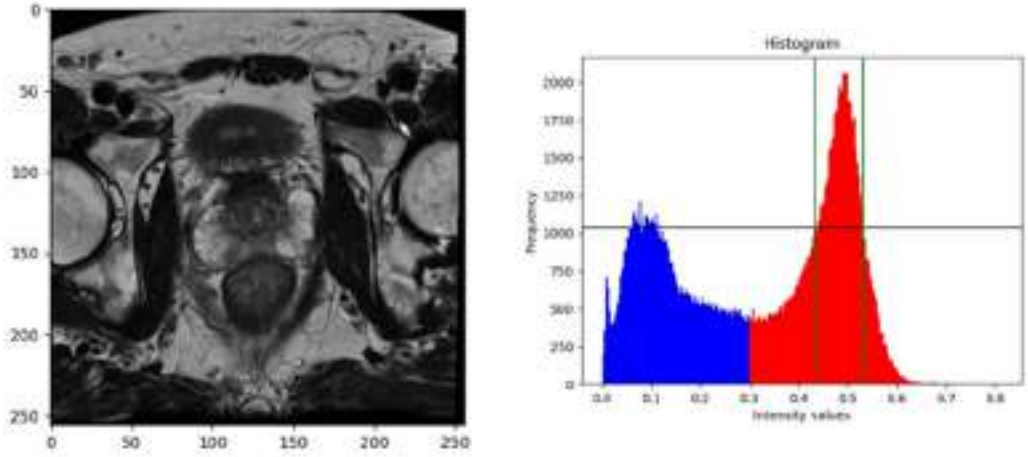
Even for the patients that their minimum FWHM was obtained with different setting, the difference in the FWHM values between the derived optimal configuration and the setting with the minimum FWHM was small (< 20%).

Qualitative results: Provide some visual results (if available) of applying such tools.

The original and the N4 filtered image are presented, indicating that the image becomes brighter after applying the N4ITK filter as the low-frequency variation has been reduced. The corresponding histograms of the original and the N4 filtered image, respectively, show that the periprostatic fat distribution (indicated with red color) becomes narrower after applying the N4ITK filter. Thus, the FWHM of the periprostatic fat distribution becomes smaller after the N4ITK filter, indicating the reduction of the intensity inhomogeneities in the image.

	Image	**Histogram**
Original		

N4 filtered



Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

- **Successful use cases:** The tool was successfully applied to MR T2W series of 4 publicly available prostate datasets: PROSTATEx, PROSTATE-DIAGNOSIS, Prostate-MRI and PI-CAI dataset. Furthermore, the tool was successfully applied to T2W series from the ProCAncer-I project as a pre-processing step.
- **Publication link :**

https://www.sciencedirect.com/science/article/pii/S0730725X23000589?via%3Dihub#bb0075

- **Keywords for searching in databases:** N4, N4ITK, bias field correction, bias field, T2W, MRI, preprocessing, prostate, prostate imaging

DQ2. 2. Technical specifications

Data: In depth description of the data used to train the tool.

Magnetic Resonance Imaging (MRI) T2weighted (T2W) prostate images. Four publicly available datasets were used to train and validate the tools. The PROSTATE-DIAGNOSIS dataset contains 89 prostate cancer T2W MR images acquired on a 1.5T Philips Achieva scanner using combined surface and endorectal coil. The PROSTATEx contains 204 prostate T2W MR images that were acquired on two different types of Siemens 3T MR scanners with surface coil, the MAGNETOM Trio and Skyra. Two more datasets, the Prostate-MRI and the PI-CAI dataset, were used to validate the results. The Prostate-MRI dataset consists of 26 prostate T2W MR images acquired on a 3T Philips Achieva scanner using combined phased-array surface and endorectal coil. The PI-CAI dataset contains prostate T2W MR images acquired on Siemens Healthineers or Philips Medical Systems-based MR scanners at 1.5 T or 3 T using surface coils. Two subsets of T2W images from the PI-CAI dataset were selected based on the two different magnetic field strengths, i.e., 1.5T (59 patients) and 3T (80 patients).

Methods: in depth description of the methodology used for its development including all data preprocessing.

To apply the N4ITK filter, only the values of its parameters are required to be used as input to the filter's algorithm.
The parameters of the N4ITK filter are :

- the convergence threshold
- the shrink factor
- the fitting level
- the number of iterations
- the use of mask

To identify the optimal configuration of the filter, 240 different configurations are being examined. For each configuration, the original and the N4 filtered image are cropped to include only the periprostatic region by keeping the middle part of the image and removing the heterogeneous prostate gland and the area distant from the prostate. The mask of the prostate gland was derived automatically by using a deep learning model that extracts a cubic box around the estimated position of the whole prostate gland. The K-means algorithm (with K=2) is used to identify the periprostatic fat distribution with the high intensity signal. Then, the Full Width at Half Maximum (FWHM) of the periprostatic fat distribution is calculated. The filter configuration that results in the minimum FWHM for each patient is considered as optimum.

Specific Technical information:

1. CPU
2. Programming language : python 3.8.13
3. Expected RAM usage : at least 8GB (depends on the amount of input data)
4. Running mode : batch-based docker
5. Software version : 1.7
6. Libraries : docker
7. Security measures: The tool does not require admin rights to execute and does not require privileged execution mode.

Traceability and monitoring mechanism.

Informative messages are displayed on screen when running the docker in interactive mode (specifying -it). If an error occurs, the corresponding error message will be displayed to inform the user what the source of the error was. Corresponding messages for the start and the completion of the tool's execution are also displayed on screen.

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.

Tests were performed in order to ensure the proper functionality of the tool. All possible functionalities were tested, i.e. applying the N4 bias filter with the default values, applying the N4 bias filter with values defined by the user, applying the pipeline to identify the optimal configuration for 1 image, and applying the pipeline to identify the optimal configuration for a batch of images.

Access restriction: Do you have any access restriction to the source code or to the binaries of your tool?

No.

Additional information for tool integration.

The tool is compatible with any operating system as it is dockerized. A comprehensive documentation is provided for the users.

DQ2. 3. Integration specifications

Communication channel for the helpdesk, technical support channel

For any inquiries: dovrou@ics.forth.gr, nikiforakik@gmail.com

Most common errors

The most common errors are:

- No functionality selected (available functionalities: -N or -P)
- Empty directory
- Not valid input format (valid formats: DICOM)
- Lack of slice location information on the DICOM header
- Invalid image dimensions (the image should be represented by a 3D array of shape [SLICES, WIDTH, HEIGHT])
- Dimension mismatch error between image and corresponding mask

FAQs

Q: What is the purpose of the tool?

A: The tool can be used as an image pre-processing step to reduce the bias field effect on prostate images. It aims to improve the quality of the image by reducing the intensity inhomogeneities caused by the low-frequency variation. Furthermore, it can be used to identify the optimal configuration (i.e. optimal values for each parameter) of the N4 bias filter for an image or a batch of images.

Q: Which are the functionalities of the tool?

A: The tool offers two main functionalities:

1. Apply N4 filter to image/images (either with the default parameters values or with parameters values defined by the user) to reduce the bias field.
2. Find the optimal configuration of the N4 filter for specific image/images by performing the exploratory analysis using 240 different configurations and measuring the Full Width at Half Maximum (FWHM) of the periprostatic fat distribution for each configuration. The smaller the FWHM value, the better the bias field correction.

Q: In which data can the tool be applied?

A: The tool can be applied only for Magnetic Resonance (MR) T2weighted (T2W) prostate images.

Q: Are there any specific requirements for the execution of the tool?

A: The tool can run in any operating system, as it is containerized in order to ensure compatibility. However, the execution of the tool is time-consuming due to the iterative process of the N4 bias field correction algorithm. The process for the identification of the optimal configuration of the N4 filter requires much time due to the experimentation on 240 different configurations for each image. For example, it requires approximately 25 seconds to apply the N4 bias filter with the default parameter values to one image and 10 minutes to identify the optimal configuration of the N4 filter for one image.

User Manual

Usage instructions

The tool is dockerized.

1. Download the latest version of the docker image (image_batch_n4filter_v1.7.tar)
2. Load the docker image by running the following command (assuming you are in the same directory as the tar file):
    - udocker load -i image_batch_n4filter_v1.7.tar
3. Run the following command in order to create a container and instantiate the docker image:
    - udocker run --rm
      -v "your_input_path:/home/chaimeleon/datasets"
      -v "your_output_path:/home/chaimeleon/persistent-home"
      image_batch_n4filter:1.7
      [-h] [-i] [-N] [-t T] [-s S] [-f F] [-I I] [-m] [-c] [-P]

where *your_input_path* is the path that contains the folders with the original DICOM image of each patient

your_output_path is the path where the N4 filtered image of each patient will be saved

available arguments:
-h, --help: show this help message and exit
-i, --interactive: Execution in interactive mode
-N, --n4filter: Performs N4 bias field correction method
-t T, --threshold T: Convergence threshold
-s S, --shrinkFactor S: Shrink factor
-f F, --fittingLevel F: Fitting level
-I I, --Iterations I: Number of iterations
-m, --masks: If specified, the N4 algorithm uses masks
-c, --custom-masks: the path that contains the corresponding DICOM mask of the prostate gland of each patient. If specified, custom masks are used. If not specified, the masks will be extracted automatically by the algorithm
-P, --pipeline: Apply the whole pipeline to identify the optimal N4 configuration

The two main options are:
1) Apply N4 filter to image/images, by specifying -N

2) Find the optimal N4 configuration, by specifying -P

Input/Output description:
Input : The input of the tool is a path with the dataset. The dataset should contain raw unnormalized images, which should be in DICOM format. In each dataset directory, an index.json file should be contained in order to walk through the contents of the dataset.
Examples of the required format of the input path:
-- patient_level (folder)
    -- study_level (folder)
    -- series level (folder)
    -- DICOM files (files)
…
-- index.json (file)

Output : For the first functionality (apply N4 filter), the output is the N4 filtered images, which are saved to a path specified by the user with the same format. For each series, a folder will be created and the N4 filtered image in DICOM format will be stored within the folder. For the second functionality (identify optimal configuration), a text or excel file containing the optimal configuration(s) is saved to a path specified by the user.

Mandatory/optional data: The original T2W MR image for each patient is required to run the tool.
Cases in which the tool should not be used: The tool should not be used in any organ/anatomy other than prostate. Furthermore, the tool should not be used in any sequence other than T2W. The tool is developed only for T2W MR axial prostate images.

DQ3. Trace4MEdicalImageCleaning

Contributor: DeepTrace Technologies (subcontractor of IRCCS Policlinico San Donato)
Area : Imaging data quality assessment and imaging data curation
Status : Under development
Purpose : The aim of the tool is to detect and remove text in medical images

DQ3.1 Conceptual Validation

Tool description:
Trace4MedicalImageCleaning™ is a tool aimed at automatically detecting and removing text in medical images, with a specific focus on 2D ultrasound and mammography studies.

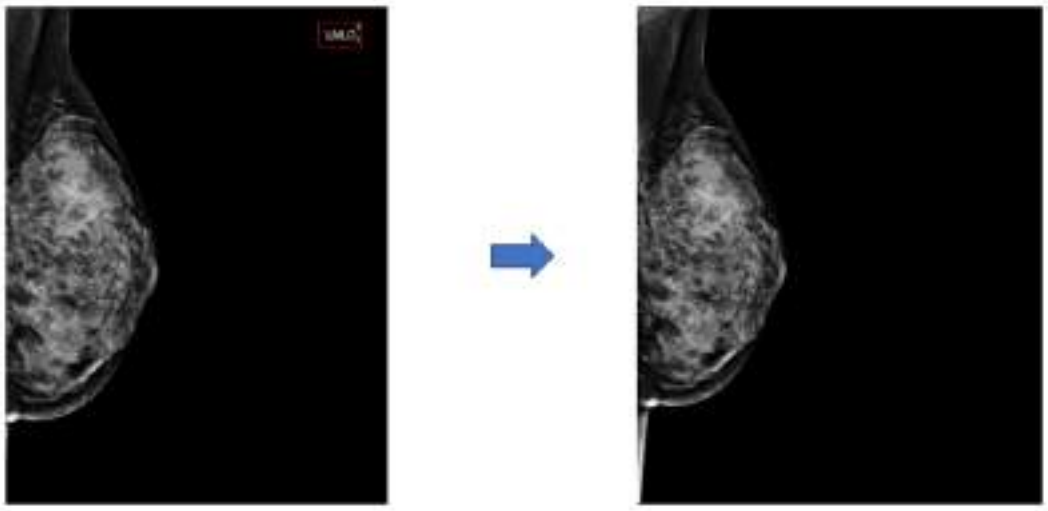
Data:
2D UltraSound (US) imaging studies; 2D Mammographic (MG) imaging studies

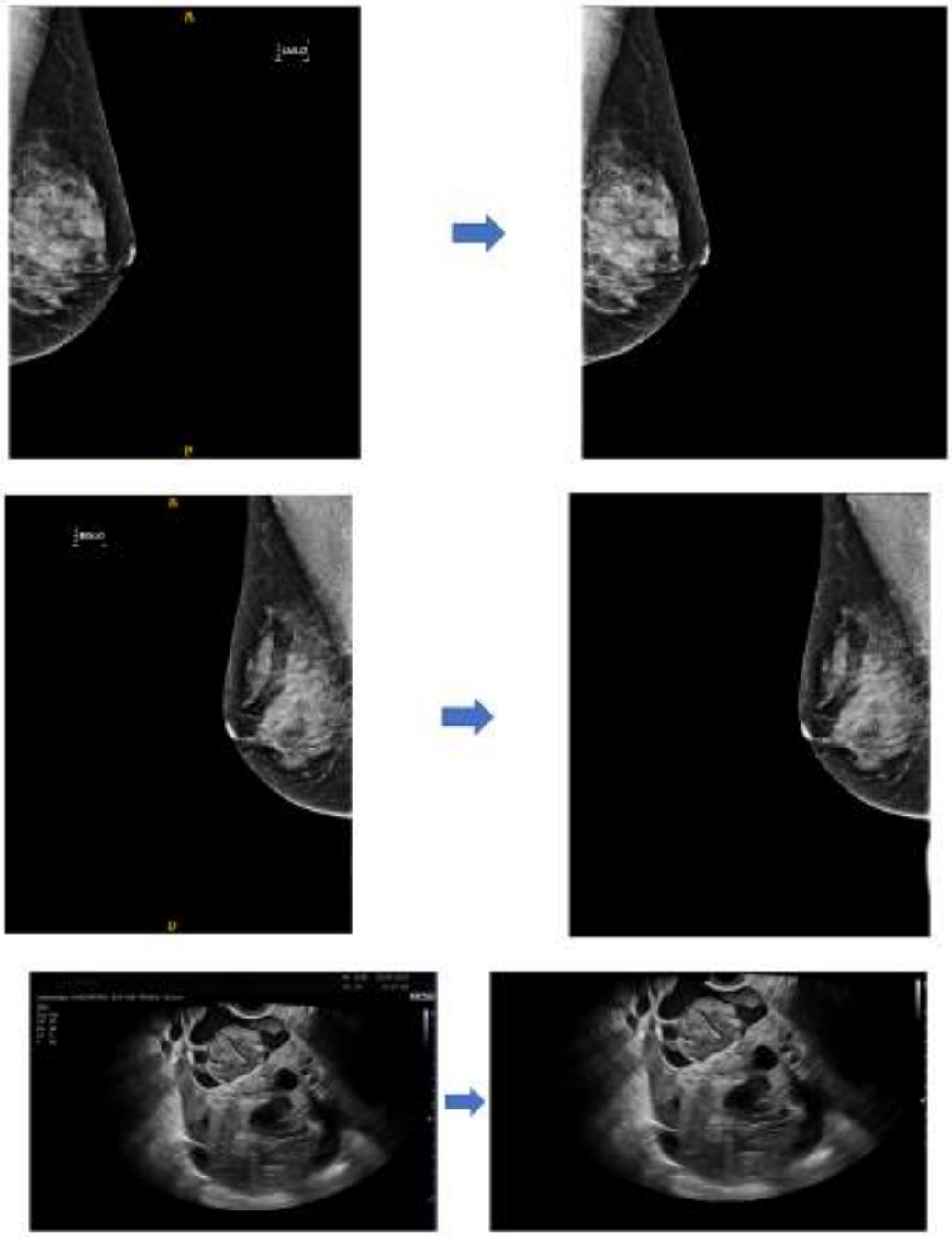
Methodology/performance:
Trace4MedicalImageCleaning™ is based on a CRAFT deep learning model (https://arxiv.org/abs/1904.01941) able to detect texts in an image coupled to information retrieved from the DICOM-study metadata. This technique is used to:

- detect text within the image frame (e.g. annotations) and to replace the text with black frames;
- detect the presence of frames in the 2D image, which may potentially contain personal identifiable information, and remove these frames from the 2D image.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input :
The input of Trace4MedicalImageCleaning™ is the path to the raw 2D US/MG image, which should be in DICOM format. More precisely, the input is a string that encompasses the absolute path to the image.
Example of the required format of the input path: “path/to/the/dicomstudy.dcm”
Output :
-- path to the processed image (.dcm file) as described in the “Methodology” section
-- path to the logfile (.out file)
-- flag (0/1 as unsuccessful/successful)

Qualitative results: The raw (left) and processed (right) 2D images of representative examples (both MG and US) are reported below.





DQ3.2. Technical specifications

This part of the documentation is dedicated to providing all relevant technical specifications to prepare for the tool's integration into the EUCAIM test environment.

It is possible that the tool's integration will require, among others, some modifications in the input/output, or the inclusion of monitoring mechanisms.

Data: In depth description of the data used to train the tool.

Trace4MedicalImageCleaning™ is based both on 1) a CRAFT deep learning model (https://arxiv.org/abs/1904.01941) able to detect text in an image, and 2) information extracted from the DICOM metadata. The CRAFT (Character Region Awareness For Text detection) deep learning model was previously trained on multiple-domain images containing multiple-language text, in order to detect texts in such images in 9 different languages, encompassing English, Italian, French, German, Arabic, Chinese, Japanese, Korean, and Indian.

Methods: in depth description of the methodology used for its development including all data preprocessing.

Trace4MedicalImageCleaning™ is based on 1) a CRAFT deep learning model (https://arxiv.org/abs/1904.01941) able to detect text in an image, and 2) the extraction of information from the DICOM-study metadata in order to identify and locate the presence of frames which potentially contain text or personal information.

The application of Trace4MedicalImageCleaning™ to a 2D US or MG DICOM study will thus result in:

1) the detection of text within the image frame (e.g. annotations);
2) the detection of frames that may potentially contain personal identifiable information.

In both cases, the identified text or frame will be removed from the 2D image: in the first case, the box containing the identified text will be darkened; in the second case, the frame will be cut off from the image.

No further preprocessing is applied to the input image before performing the described procedure.

The clean image will finally be saved as a DICOM file and returned as output of the tool.

Specific Technical information:

a. GPU is used if available, otherwise CPU is used

Minimum 3 GB of free HDD space required for installation and running

Operating System: Windows 10 (version 1709 or higher)

Processors (minimum): any Intel or AMD x86-64

b. Programming language: Matlab, Python

c. Expected RAM usage: depending on the input data, minimum 8 GB of free RAM

d. Running mode: case based, without interaction with a user interface on the screen

e. Software version: Trace4MedicalImageCleaning™ v1.0.00

f. Libraries : Additionally, Matlab runtime is installed along with the containerized software

g. Security measures: does the tool require administrator privileges? No admin privileges are necessary to run the tool

Traceability and monitoring mechanism.

Trace4MedicalImageCleaning ™ does not store any information into system files nor communicates any information through internet connection.

The results of the analysis are stored locally in the output files (DICOM image) that contains the original image with the identified text darkened.

In addition, for traceability and monitoring, a logfile is generated for each analysis, which reports non-personal information such as the version of the tool, the outcome of the analysis (successful/unsuccessful), details on any occurred error during analysis, the path to the results of the analysis and the path to the logfile itself. The logfile is structured as follows:
Trace4MedicalImageCleaning™ 2023-2024
08-Apr-2024 09:00:00
Processing...
====== !! Using GPU !! ======
====== !! Image processed !! ======
output_path = 'path\to\clean_image.dicom'
output_logpath = 'path\to\log.out'
flag = 1

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
Unitary tests were conducted to validate, assess and evaluate all potential functionalities in use, specifically applying the proposed method to a testing dataset of selected 2D ultrasound and mammography images.

Access restriction: Do you have any access restriction to the source code or to the binaries of your tool?

Additional information for tool integration.
For running the dockerized version of Trace4MedicalImageCleaning™, the following command must be used (after having downloaded the latest docker image of the tool)

docker run -ti --rm -v local/path/to/Trace4MedicalImageCleaning/app:/app/files --env-file local/path/to/envfile --env-file 'local/path/to/.aws/credentials' python3 /app/startup.py

where the env-file must have the following structure

STORAGE_SOURCE_FOLDERPATH=dt-trace4medicalimagecleaning-source-test
STORAGE_RESULTS_FOLDERPATH=dt-trace4medicalimagecleaning-dest-test
OPERATION=apply
REMOTE_RESULTS_FOLDERNAME=cleaning_11072023_151136/

Including information such as the path of the folder containing the files on which the operation will be applied.

DQ4. Time Coherence Tool

Contributor: HULAFE
Status : Developped
Area: Clinical data quality assessment and curation

Purpose :The aim of the tool is to validate the chronological order and logical consistency of dates associated with a patient's medical history.

DQ4.1 Conceptual Validation

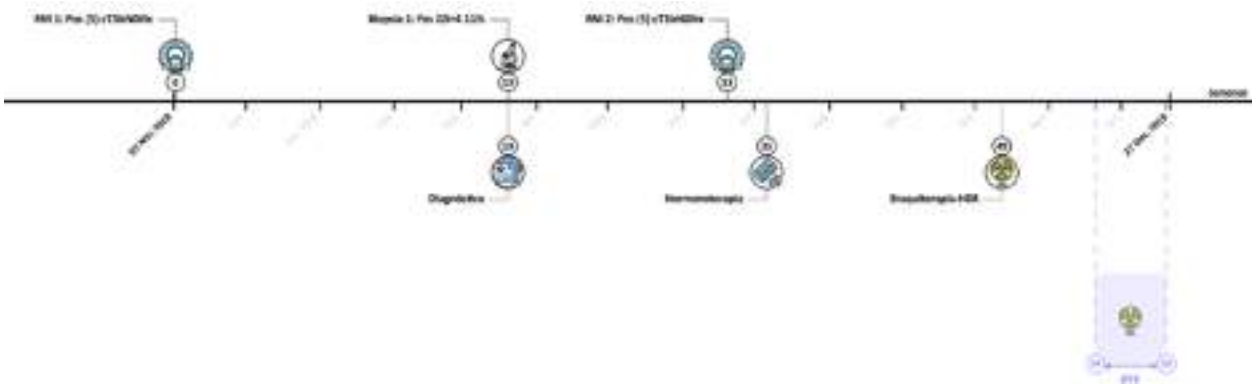
Tool description:
Simple tool that aims to validate visually the chronological order and logical consistency of dates associated with a patient's medical history, to do so it generates a timeline visualization for each patient of a database file.

Data:
Data from Excel or databases containing relevant dates such as MRI scans, biopsies, consultations, diagnoses, and lab tests, etc.

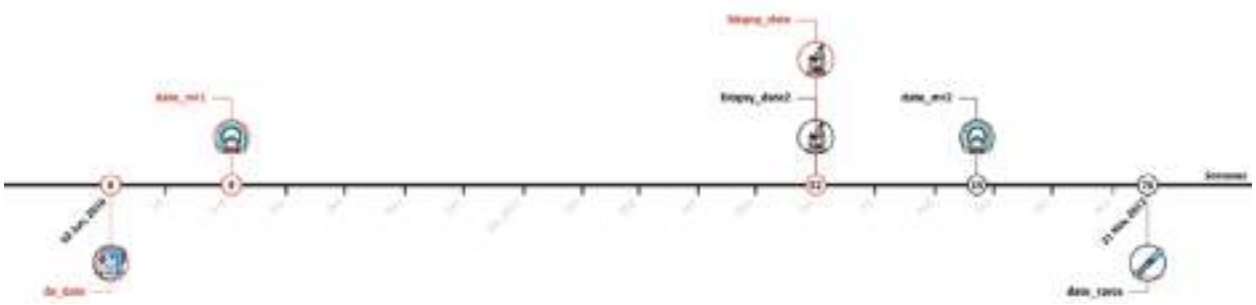
Methodology/performance:
By using pre-defined timeline rules and the Pytimeline python library, the script reads an Excel type database file and checks for time coherence using the defined rules. It finally outputs and displays an .svg file for the timeline and rule violations visualization.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : xls, xlsx, xlsm, xlsb, odf, ods and odt.
Output : svg image file.

Quantitative results: performance obtained during training of the tool (if applies)
Example output 1:



Example output 2 with timeline rule violations:



Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).
Script written by Adrian Galiana based on the Pytimeline github library.
Licence
Under internal license by IISLAFE (HULAFE) institution

DQ4.2 Technical specifications

Data: In depth description of the data used to train the tool.
Excel database with medical patients information.

Methods: in depth description of the methodology used for its development including all data preprocessing.

1. Requirement Analysis: Gathered insights from medical professionals to understand timeline validation needs.
2. Design: Conceptualized tool functionalities, input formats, and validation rules.
3. Testing: Conducted testing with real medical data.
4. Integration: Integrated Pytimeline library for efficient timeline visualization.

Specific Technical information:

a. CPU
b. Programming language : python 3.11
c. Expected RAM usage :
d. Running mode : batch-based
e. Software version :
f. Libraries : tkinter, pendulum, svgwrite, loguru, pandas, tqdm, datetime
g. Security measures: writing to host data restricted to a non-root user ; container does not require it to be executed in a privileged mode

DQ5. Tabular Data Curator

Contributor: FORTH
Area: Clinical data curation
Status : Developed
Summary: A fully automated service which can be applied on any kind of tabular data (e.g. clinical) to automatically identify duplicated fields (lexically similar and/or highly correlated features), outliers, data inconsistencies. It can also deal with missing values through the application of smart data imputers.

DQ5.1 Tool description for its conceptual validation

The conceptual validation aims to ensure that a tool is aligned in its purpose and functionality with the pre-processing tools specified in EUCAIM T5.3. For this, the following documentation will detail the necessity of the tool and its functioning.

Tool description: The proposed tool follows a three-layer architecture and serves as a data diagnostics framework for reporting outliers, missing values, duplicated features and other data inconsistencies in tabular datasets.

Data: Any kind of tabular dataset, where the number of rows correspond to patient records and the number of columns corresponds to features which can be related to different categories, like demographic, laboratory examinations, medical conditions, therapies, etc.

Methodology/performance: The tool was developed following a waterfall architecture. The main function of the data curator executes the following phases:

(i) the data annotation which involves the estimation of the quality of each feature, the number of features (columns), the number of instances (rows), the number of missing values, the number of features with good, fair, bad quality state, the number of detected outliers

(ii) the quality assessment phase which involves the application of univariate methods for outlier detection (supported options include: the z-score, the interquartile range, and the Grubb's test) and for imputation (supported methods include: Average/most frequent, Random, None)

(iii) the generation of the data quality evaluation report phase which aims to generate a concise report in a JSON format that summarizes the findings from the first phase in a standardized format, (iv) the generation of the curated dataset phase which refers to the original dataset where the problematic fields (missing values, outliers, features with good/fair/bad quality state, incompatible features) are highlighted using proper color coding, (v) the generation of the clean dataset phase where the bad features are removed from the analysis (this is optional). Each phase was developed in a separate function which is called within the main function.

Use: brief description of the tool's functioning (if it applies).
The TDC tool is a flask-based application which has developed in Python 3.9 and has been distributed as a dockerized application. It takes as input a tabular dataset in either .csv or .xlsx or a specified .JSON format and provides five user-friendly reports (in .xlsx format) summarising metadata and feature-level diagnostics, problematic fields, as well as, highly correlated and lexically similar pairs of features. The TDC tool also provides a .JSON file which includes all the previous information.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input: A tabular dataset in .csv or .xlsx or .JSON format.
Output: The TDC tool generates the following reports: (i) a data quality evaluation report which summarises useful metadata and feature-level diagnostics, (ii) a curated dataset which is the original dataset where the problematic fields (e.g., outliers, missing values, data inconsistencies) are highlighted using colour coding, (iii) a clean curated dataset which is the curated dataset where the features with bad quality (i.e. with more than 30% missing values) are automatically removed, (iv) a similarity report which summarises the highly correlated pairs of features (if any), (v) another similarity report which summarises the lexically similar pairs of features (if any), and (vi) a structured .json file which includes all the information from the previous reports in a structured way to support the work of programmers towards the development of customised front-end interfaces.

Quantitative results: performance obtained during training of the tool (if applies)
Note: The tool is not AI-oriented and thus there is no need for training.
Cohort specific execution times are reported in the study of Pezoulas et al., 2019 .For the University of Athens (UoA) cohort, which included data from 200 patients, the average execution time of the web service was 3.79 seconds. For the Harokopio University of Athens (HUA) cohort, which comprised data from 100 patients, the execution time was shorter, averaging at 1.9 seconds. The time required for fetching data was almost negligible for both cohorts, taking less than 1 second, which underscores the efficiency of data retrieval operations. The execution time for applying the service, which includes data annotation and evaluation, outlier detection, similarity detection was around 1.1 seconds for the UoA cohort and 1.3 seconds for the HUA cohort. This indicates that the bulk of computational efforts are dedicated to these critical operations, with minimal differences observed between the two cohorts despite their size difference. Constructing the data evaluation reports, along with generating the curated dataset, took approximately 9 seconds for the UoA cohort and 4 seconds for the HUA cohort. This variation in time can be attributed to the number of patients in each cohort, with the larger UoA cohort requiring more time to process and compile the reports and curated dataset. Manual attempts at data curation would require significantly more time, highlighting the substantial time-saving benefits offered by the tabular data curation tool.

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

An indicative list of publications related to successful use cases is presented next.

- The core publication of the tool (with successful use cases):

Pezoulas, Vasileios C., et al. "Medical data quality assessment: On the development of an automated framework for medical data curation." Computers in biology and medicine 107 (2019): 270-283.

- Additional publications with successful use cases:

- Pezoulas, Vasileios C., et al. "Enhancing medical data quality through data curation: A case study in primary Sjögren's syndrome." Clin. Exp. Rheumatology 37.3 (2019): 90-96.
- Pezoulas, Vasileios C., et al. "Distilling knowledge from high quality biobank data towards the discovery of risk factors for patients with cardiovascular diseases and depression." 2023 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI). IEEE, 2023.
- Pezoulas, Vasileios C., et al. "A computational pipeline for data augmentation towards the improvement of disease classification and risk stratification models: A case study in two clinical domains." Computers in Biology and Medicine 134 (2021): 104520.

Keywords for searching in databases: tabular data, data curation, data quality assessment, data quality control.

DQ5.2. Technical specifications

Data: In depth description of the data used to train the tool.
The tool is not an AI tool. It has been tested on clinical data across various diseases including cardiovascular diseases, autoimmune diseases, mental disorders.

Methods: in depth description of the methodology used for its development including all data preprocessing.
Implementation following python development best practices to ensure the quality and reliability of data curation. The methodology includes various stages, varying from data preprocessing and quality assessment to the implementation of a framework for data curation. Here is an in-depth description of the methodology:

- Objectives and framework design: First we defined the objectives and designed a comprehensive framework to address the challenges of data quality assessment in the medical domain. The framework aimed at automating the process of data curation, focusing on key data quality measures such as accuracy, completeness, consistency, and relevance.
- Design of a three-Layer architecture: The framework was structured around a three-layer architecture, which facilitated a systematic approach to data curation. This architecture included data evaluation, data quality control modules, each designed to perform specific functions in the data curation process.
    - Data evaluation module: This module provides a preliminary analysis of the raw data, extracting metadata, and computing descriptive statistics to understand the data's

structure and quality. It is responsible for identifying the data types, range of values, and categorizing the attributes based on their quality and relevance.

- Data quality control module: This module employs various techniques for anomaly detection (outlier detection using univariate and multivariate methods), missing value detection, and similarity detection to identify duplicate fields and highly correlated distributions. It plays a critical role in ensuring data accuracy, completeness, and reliability.

- Automated curation and case study application: The methodology emphasized the automation of the data curation process, leveraging computational techniques to handle large datasets efficiently.

Specific Technical information:

a. CPU: Monolithic algorithmic implementation / no need for GPU.
b. Programming language: Python 3.9.
c. Expected RAM usage: 16GB (actual RAM requirements depend on the size of the dataset).
d. Running mode: case-based.
e. Software version: v1.0.0.
f. Libraries:
    - Flask==3.0.3,
    - numpy==1.26.4,
    - Orange3==3.36.2,
    - outlier-utils==0.0.5,
    - pandas==2.2.1,
    - python-Levenshtein==0.25.1,
    - scikit-learn==1.3.0, scipy==1.13.0,
    - xlutils==2.0.0.
g. Security measures: does the tool require administrator privileges?

The tool is dockerized (no privileges). It only requires temporary storage allocation for temporary files in the local non-privileged user.

Traceability and monitoring mechanism.
Currently no mechanism exposed externally from the docker.

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
Only debugging tests were applied and the results were reported in published papers. The tool was tested on extensive datasets with valid outcomes.

Access restriction : do you have any access restriction to the source code or to the binaries of the tool?
No. The tool is dockerized, therefore if it is called from a web API the user does not have access to the source code.

Additional information for tool integration.
Continuous optimization of the provided algorithm and toolset is performed.

DQ5.3. Integration Validation

In this stage, further documentation is required by the tool providers. In particular, the following important points are suggested to be described about the tools:

Communication channel for the helpdesk, technical support channel:
We have a team of developers who can support any errors regarding the tool.

Most common errors:
Uploading invalid input dataset format. Some algorithms (i.e. for outlier detection, for similarity detection) are sensitive to the type of the input data.

FAQs:
Currently there is not a list of FAQs since the tool has been exploited internally.

User Manual:

a. Installation/configuration instructions:

Please follow these steps:

- Clone the repository from https://github.com/vpz4/TDC.
- It is recommended that you store it in “C:/TDC” (e.g. in Windows).
- Install docker and docker compose utility as prerequisites.
- Build the docker by executing “docker build -t tdc-app .”.
- Run “docker run -d -p 5000:5000 -v C:/TDC/results:/app/results --name tdc-app tdc-app”.
- The tool is accessible through a browser at: http://127.0.0.1:5000/main.

This is how the TDC tool appears:

This is the structure of the TDC parent folder:

- hosted: a folder where the server temporarily stores the input dataset (it is removed after the execution).
- results: a folder where the results are stored.
- sample_datsets: includes a test input dataset (for demonstration purposes) in .xlsx, .csv and .json formats.
- static: includes a subfolder “images” which contains the HTML logos (from EUCAIM and from FORTH).
- .gitignore: this is needed when pushing unnecessary updates to github.
- templates: a folder which includes the HTML script (index.html) along with the logos (from EUCAIM and FORTH).
- app.py: the main script of the TDC tool.
- DockerFile: the file for building the dockerized application.
- README.md: a readme document which is needed for the GitHub repository.
- requirements.txt: the versions of the required python libraries that are necessary when building the docker.

b. Usage instructions:

- Select your dataset (supported formats: .csv/.json/.xlsx).
- Select a method for outlier detection (mandatory).
    - z-score: A measure of how many standard deviations a data point is from the mean of its distribution.
    - z-score (mod.): A robust measure of standard deviations for data points in a sample, typically using the median and median absolute deviation instead of the mean and standard deviation, which improves resilience to outliers. Recommended option.
    - interquartile range (IQR): A measure of statistical dispersion based on the difference between the 75th and 25th percentiles.
    - Grubb’s test: A statistical test which is used to identify outliers by comparing the extreme values to the expected values under a normal distribution.
    - Local outlier factor (LOF): An algorithm that detects outliers by measuring the local deviation of a given data point with respect to its neighbours. Note: This option is sensitive to data type errors and is currently ignored.
    - Isolation Forests: An ensemble method for anomaly detection that isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
    - Isolation Forests (mod.): A modified feature wise application of the Isolation Forests algorithm.
- Select a method for similarity detection (optional).
    - Spearman rank-order correlation coefficient: A non-parametric measure that assesses how well the relationship between two variables can be described using a monotonic function, based on their ranked values. Recommended option.
    - Perason’s correlation coefficient: A measure of the linear correlation between two variables, giving a value between -1 and 1 inclusive, where 1 is total positive

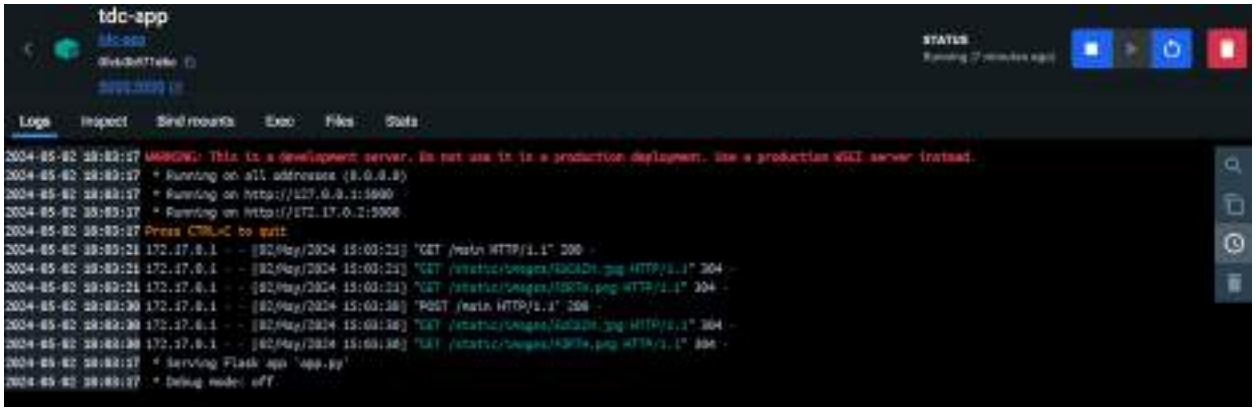
linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation.
    - Kendall's tau: A non-parametric statistic used to measure the ordinal association between two measured quantities, reflecting the similarity of the orderings of the data when ranked by each of the quantities.
    - Covariance: A measure indicating the extent to which two random variables change in tandem, showing whether increases in one variable tend to be accompanied by increases (positive covariance) or decreases (negative covariance) in the other variable.
    - None: No similarity detection is applied
      Note: The TDC tool has a built-in functionality that calculates the Jaro distance between each pair of feature labels to estimate the lexical similarity and thus input is not requested by the user.
- Select a method for data imputation (optional).
    - Average/median: Imputes missing values using the average (mean) or median of the observed data points in the same variable, which is useful for maintaining the central tendency of the dataset.
    - Random: Fills missing entries with zeros, which can be appropriate for datasets where a zero can represent the absence of an attribute or a neutral baseline, but may skew data distributions if zero is not a meaningful value for the variable.
    - Zeros: Fills missing entries with zeros, which can be appropriate for datasets where a zero can represent the absence of an attribute or a neutral baseline, but may skew data distributions if zero is not a meaningful value for the variable.
    - None: No imputation method is applied. Recommended option.
- Sample input format (in .xlsx):

num_feat1	num_feat2	text_cat_feat	text_feat	bool_feat	num_missing_feat	num_missing_cat_feat	text_missing_cat_feat	text_missing_feat	bool_missing_feat	mixed_data_types	bad_feat
0.1	0.2 severe	greece	"true"		0.1	1			"false"		
2.5	3 mild	usa	"false"		0.1	4 alpha	left			+	
21.1	21.1 none	france	"false"		1	1	left				
0.21	0.21 severe	greece	"true"		2	2 beta			"true"		
100	32.5 mild	usa	"false"				middle			2022-10-20T11:44:23Z	
10	10 none	italy	"true"		5.1	3 delta	middle				
5.1	5.1 severe	greece	"true"						"true"	2	
2.5	2.5 mild	japan	"true"		1.1	beta	right				
7.8	7.8 none	germany	"false"		2		right		"false"	2019-11-20T11:44:23Z	
3.6	3.6 severe	greece	"true"			alpha					

- Sample input format (in .csv):

- Select the parameters and then press the Apply button to start the curation process. In this example we select:
    - the test.csv file which is located in the sample_datasets folder,
    - the z-score (mod.) method for outlier detection,
    - the Spearman rank-order correlation coefficient for similarity detection,
    - the imputation method set to None.

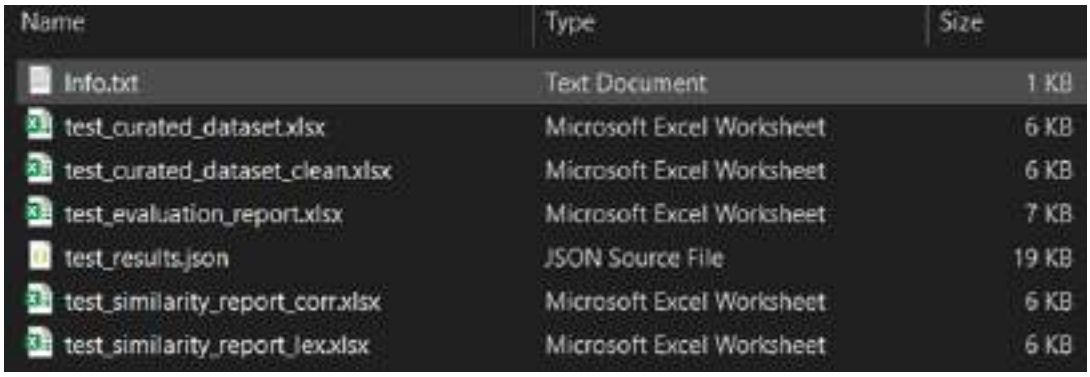
- This is an example of how the docker looks during the execution phase.



- This is an example of the docker's response in Visual Studio.

- A message appears upon the successful execution of the tool (in green colour) and the parameters in the page are redirected to their original state.

- The reports are stored in the results folder as follows:
    - Info.txt: Indicates that the results folder should not be deleted.
    - test_curated_dataset.xlsx: the curated dataset.
    - test_curated_dataset_clean.xlsx: the clean curated dataset.
    - test_evaluation_report.xlsx: the quality evaluation report.
    - test_similarity_report_corr.xlsx: the similarity report with the identified highly correlated pairs of features.
    - test_similarity_report_lex.xlsx: the similarity report with the identified lexically similar pairs of features.



Name	Type	Size
Info.txt	Text Document	1 KB
test_curated_dataset.xlsx	Microsoft Excel Worksheet	6 KB
test_curated_dataset_clean.xlsx	Microsoft Excel Worksheet	6 KB
test_evaluation_report.xlsx	Microsoft Excel Worksheet	7 KB
test_results.json	JSON Source File	19 KB
test_similarity_report_corr.xlsx	Microsoft Excel Worksheet	6 KB
test_similarity_report_lex.xlsx	Microsoft Excel Worksheet	6 KB

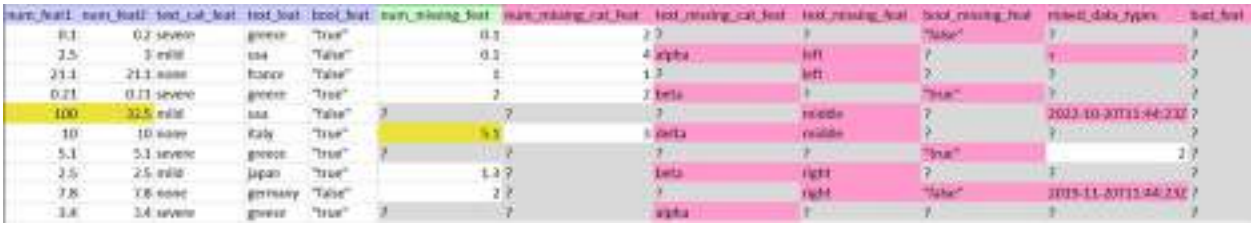
An instance of the data quality evaluation report is presented next. The report consists of two panels, the “Metadata” panel and the “Quality evaluation” panel.

- The “Metadata” panel includes information regarding:
    - the number of features,
    - the number of instances,
    - the number of discrete features (i.e. those with categorical variable type),
    - the number of continuous features (i.e. those with numeric data type),
    - the number of unknown features (i.e. those with mixed data type),
    - the percentage of missing values,
    - the percentage of good (no missing values), fair (less than 30% missing values) and bad features (more than 30% missing values).
- The “Quality assessment” panel provides feature-level information regarding:
    - the value range (minimum and maximum intervals),
    - the data type (numeric or categorical or unknown),
    - the variable type (int, float, string, unknown),
    - the quality state (good, fair, bad),
    - whether outliers were detected or not, and
    - a summary on the detected incompatibilities.

An instance of the curated dataset is presented next. This is in fact the input dataset, where:

- the outliers are highlighted with orange colour,
- the missing values are highlighted with grey colour followed by “?” for easier tracking,
- the inconsistencies are highlighted with red colour,
- the feature names are also highlighted based on their quality state as follows:
    - good quality state with blue colour,
    - fair quality state with green colour,

- bad quality state with red colour.



An instance of the clean curated dataset is presented next. This is in fact the curated dataset, where the features with bad quality have been automatically removed.

num_feat1	num_feat2	text_cat_feat	text_feat	bool_feat	num_missing_feat
0.1	0.2	severe	greece	"true"	0.1
2.5	3	mild	usa	"false"	0.1
21.1	21.1	none	france	"false"	1
0.21	0.21	severe	greece	"true"	2
100	32.5	mild	usa	"false"	?
10	10	none	italy	"true"	5.1
5.1	5.1	severe	greece	"true"	?
2.5	2.5	mild	japan	"true"	1.3
7.8	7.8	none	germany	"false"	2
3.4	3.4	severe	greece	"true"	?

An instance of the similarity report is presented next which highlights the highly-correlated pairs of features along with the respective correlation value.

f1	f2	value
num_feat2	num_feat1	0.996965092

An instance of the similarity report is presented next which highlights the lexically similar pairs of features along with the respective Jaro distance value.

f1	f2	value
num_feat2	num_feat1	0.925925926

Additional considerations:

- Input description (the format is mandatory):
  - The dimension of the input tabular dataset must be MxN, where M is the number of rows (i.e. samples) and N is the number of columns (i.e. features).
  - For the input .xlsx and .csv files, the first row must include the feature names. The samples must start from the first row. The tool looks only at the first sheet.
  - Recommended options:
    - z-score (mod.) for outlier detection,
    - Spearman rank-order correlation coefficient for similarity detection,
    - None for data imputation.
  - The input .json file must have the following two fields (feature names and data values), in a structured way, as follows:

```
{
      "features": [
      "num_feat1",
      "num_feat2",
      "text_cat_feat",
      "text_feat",
      "bool_feat",
      "num_missing_feat",
      "num_missing_cat_feat",
      "text_missing_cat_feat",
      "text_missing_feat",
      "bool_missing_feat",
      "mixed_data_types",
      "bad_feat"
  ],
  "data": [
      [
      0.1,
      0.2,
      "severe",
      "greece",
      "true",
      0.1,
      2,
      null,
      null,
      "false",
      null,
      null
      ],
   ...
  ]
}
```


Output description: The five generated reports (in .xlsx format) which were shown in the previous section are human-readable. On the other hand, the results.json file offers a lower level structure which can help the programmers to develop customised front-end interfaces for the reports. More specifically, the fields in the .json file represent the parameters which are included in the five generated reports along with the colour codes that are used to highlight all the fields including the problematic ones and the quality state of the features. These colour codes are returned as lists similar to those with the values.

Integration tests: Integration test shall be conducted in the EUCAIM marketplace.

Results of non-functional tests: These are reported in published papers (scalability, usability, applicability, generalizability).

DQ6. RACLAHE Filter

Contributor: FORTH
Area: Imaging data curation
Summary: The RACLAHE filter's purpose is to locate the prostate's whole gland in T2 MR axial images and enhance that area by applying CLAHE algorithm. The filter proved to be effective on segmentation tasks as it improves the segmentation performance on 5 DL models.
Status : Containerized
Purpose : The Notion of the tool relies on the image enhancement of T2W MR images in order to increase the performance of Deep Learning Segmentation models to segment Prostate's Whole Gland

DQ6.1. Tool description for its conceptual validation

Tool description: A bounding box model was trained to identify the region of interest (prostate) in prostate MR T2W images, and afterwards the CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm was applied within that region to enhance it. The experiments indicated a tendency to further improve the prostate and zonal segmentation performance on several DL models when RACLAHE is applied as an image preprocessing technique prior to model training.

Data: Prostatic T2w MRI

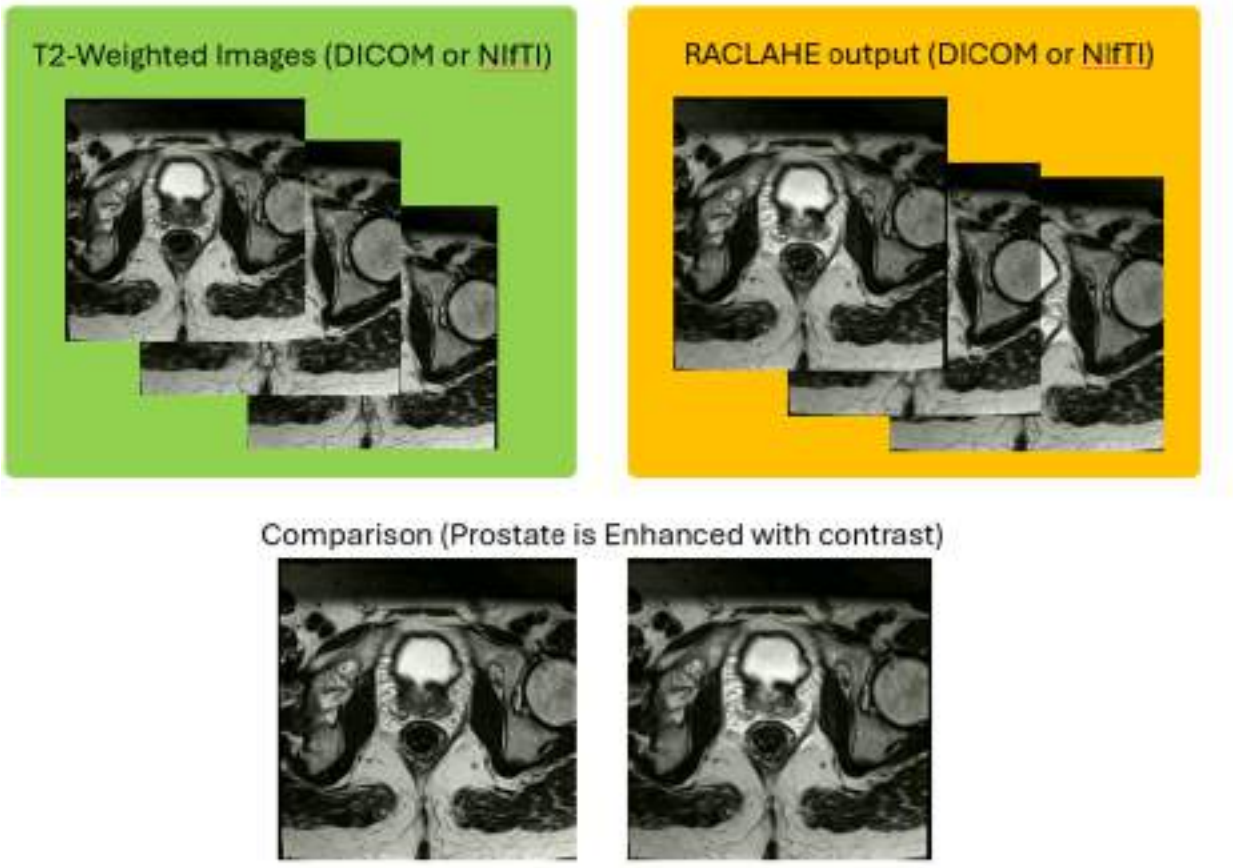
Methodology/performance: Deep Learning Model + Histogram Processing Algorithm. In particular, a Deep Learning U-Net model(intended for image segmentation tasks) is employed to identify the region of interest we would like to apple the CLAHE algorithm and enhance that area. Regarding the Use Case, RACLAHE is meant to be used for enhancing prostate area in T2W MR prostate images
Enhancement of the performance of segmenting the Prostate on T2W MR images from 3 to 9% Dice Score.

Use: brief description of the tool's functioning (if it applies): RACLAHE performs histogram processing within a bounding box proposed by a U-Net Deep Learning model, capable of identifying the prostate in T2W MR images.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : NIFTI or DICOM images
Output : DICOM (prioritised) or NIfTI files with the processed imaging exam of the patient.

Quantitative results: performance obtained during training of the tool (if applies)
Difference between RACLAHE and the other tested image filters: Dice Score from 3 to 9 %.

Qualitative results: T2W is processed in a way that the prostate region within the image is enhanced and it appears brighter and more contrasted compared to other non relative areas within the area of interest (bounding box).



Additional information:

- Use Cases: Prostate's Whole Gland, Prostate's Peripheral zone, Prostate's transition zone
- Licence: MIT

open access code with examples and instructions to install the python package: https://github.com/dzaridis/RACLAHE_Image_Enhancement_for_CNN_model_segmentation
Python Package: raclahe==0.1.2, https://pypi.org/project/raclahe/

- Publication : https://www.nature.com/articles/s41598-023-27671-8

Keywords for searching in databases: MRI, Prostate, T2-Weighted, Image enhancement, Image Processing, Image Filtering

DQ6.2. Technical specifications

This part of the documentation is dedicated to providing all relevant technical specifications to prepare for the tool's integration into the EUCAIM test environment.

It is possible that the tool's integration will require, among others, some modifications in the input/output, or the inclusion of monitoring mechanisms.

Data: In depth description of the data used to train the tool.
T2W Prostate MR images. More specifically, the tool has been trained on 204 cases from the Prostate-X2 dataset and validated on 50 cases from the prostate 3T Dataset and 1000 cases from the PICAI dataset.

Methods: in depth description of the methodology used for its development including all data preprocessing.
A U-Net bounding box model is utilized to isolate the whole gland from the outer parts of the examination and the CLAHE algorithm is applied in the isolated region whole gland region to enhance further that area.

a. The algorithm reads NIfTI or DICOM patient inputs (either a folder of DICOM images with the name of the folder as the patient name or a NIfTI file)
b. Resize of 256 x 256 pixels is applied
c. Minimum-Maximum Normalization technique applied to set the pixel ranges to [0, 1] value ranges
d. The Bounding Box model has been trained and validated on 204 T2-weighted MR images from Prostate-X2 dataset, consisting of 3206 frames of 3T magnetic field strength and Siemens vendor (TrioTim, Skyra models).
e. The Bounding Box model has been externally tested on 30 T2-weighted MR images from Prostate-3T dataset, consisting of 421 frames of 3T magnetic field strength and Siemens vendor (Skyra model).
f. For the inference part the trained bounding box model is applied on the inferred 2D images
g. The output of the model is a NIfTI file for a RACLAHE processed patient

Specific Technical information:

a. CPU and GPU (Works on Both, Does not require Both)
b. Programming language : python/tensorflow
c. Expected RAM usage : depend on the data
d. Running mode : batch-based docker
e. Software version : 3.0
f. Libraries : docker
g. Security measures: No administrator privilege required

Unitary tests: description of the tests implemented to verify the correct functioning of the tool. Only debugging tests were applied and the results were reported in published papers. The tool was tested on external datasets with valid outcomes.

6. Access restriction: Do you have any access restriction to the source code or to the binaries of your tool?
No, the code is open source while the dockerized tool is closed access

DQ6.3. Integration validation

Communication channel for the helpdesk:
dimzaridis@gmail.com

Most common errors :
The region may not be identified correctly

FAQs
Where do i find the tool: request from the author (dimzaridis@gmail.com)

User Manual

a. Installation/configuration instructions

https://github.com/dzaridis/RACLAHE_Image_Enhancement_for_CNN_model_segmentation

b. Usage instructions

Please refer here for the library intended use:
https://github.com/dzaridis/RACLAHE_Image_Enhancement_for_CNN_model_segmentation
Docker Instructions:

Volume #1: folder containing subfolders with NIfTI file or DICOM images.
Host Volume: /dir/input
Volume #2: folder with Dicom outcomes if feasible. If the DICOM metadata on the original T2-Weighted sequence are corrupted the outcome will be provided in NIfTI format
Host Volume: /dir/output

c. Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.

Please use it only on T2W Prostate MR images

Integration tests: Description of tests for assessing the correct integration of the tool

- The Tool has been tested on a variety of datasets
- Since it is dockerized the volumes should be configured correctly

The host volumes should be the following

Host Volume: /dir/input : Refers to the Input Folder the user will provide containing DICOM or NIfTI files
Host Volume: /dir/output : Refers to the Output Folder the user will provide to store the DICOM or NIfTI RACLAHE processed Images

- The tool may produce a boundary box that is wrong and therefore the processed area may not fully include the prostate region

If the tool cannot handle the input images (corrupted) will not stop working but will terminate without producing an output

DQ7. NLmCED denoising filter

Contributor: CNR-IBB and EuroBioImaging
Area: Imaging data quality assessment
Status : Containerized
Purpose: To develop a denoising method that can reduce the noise and increase the quality of MR images.

DQ7.1. Tool description for its conceptual validation

Tool description: An hybrid denoising approach based on the combination of the nonlocal mean filter and the anisotropic diffusion tensor method, dockerized and pulled into the XNAT platform.

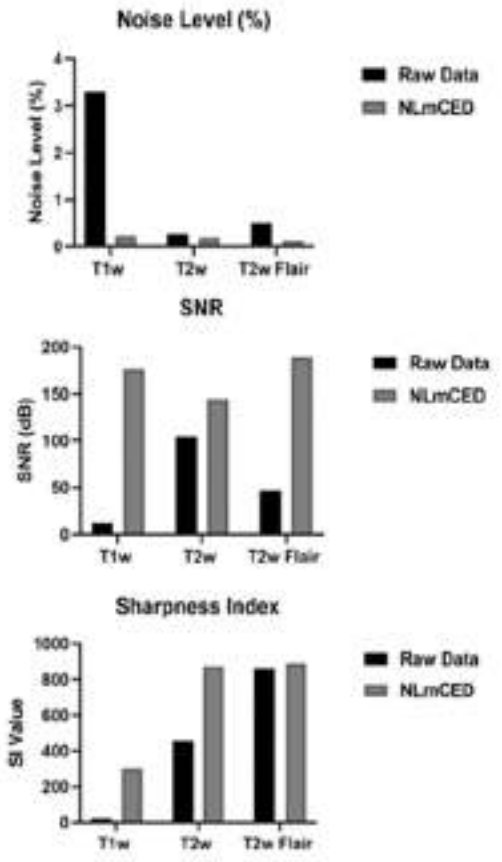
Data: MR (T1w, T1 + Gadolunium (contrast enhanced), T2w Flair and CEST-APT) DICOM images.

Methodology/performance: Docker containers with MATLAB and Python script were deployed for the XNAT Platform.

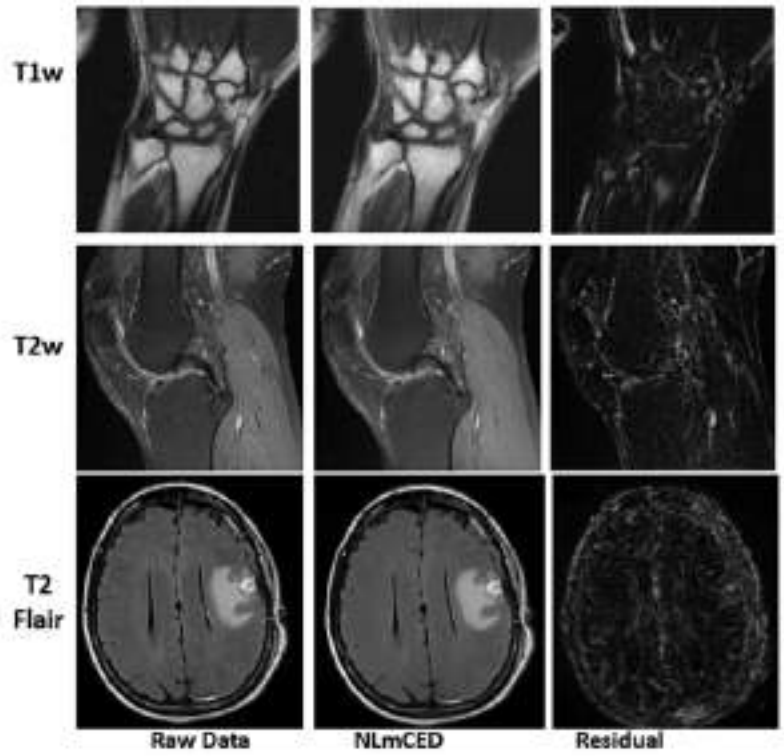
Use: brief description of the tool's functioning (if it applies).
The filter NLmCED is a combination between two powerful denoising methods such as the Non Local Mean filter and the Anisotropic Diffusion Tensor filter. The filter can provide improve the quality for T1w, T1 + Gadolunium (contrast enhanced), T2w and Flair MR images. It has proven its efficiency also in reducing noise in the CEST-MRI images improving the APTw CEST contrast detection in clinical Brain Tumor at 3T.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : DICOM images
Output : DICOM images

Quantitative results: performance obtained during training of the tool



Qualitative results: Provide some visual results (if available) of applying such tools.



Additional information:

- Communications

- Improving clinical 3T amide proton transfer weighted contrast in brain tumors by using a novel spatial denoising method. Romdhane F., Longo D.L., Papageorgakis C., Firippi E., Mancini L., Bisdas S., Zaiss M., Casagranda S. ISMRM 2022.
- Implementing a new spatial filter for improving clinical brain tumor APTw contrast at 3T, Romdhane F., et al., CEST2022, Atlanta, USA, August 2022.

- Publications

- Evaluation of a similarity anisotropic diffusion denoising approach for improving in vivo CEST-MRI tumor pH imaging. Romdhane F., Villano D., Irrera P., Consolino L., Longo D.L. Magn Reson Med. 2021 Jun;85(6):3479-3496. doi: 10.1002/mrm.28676. Epub 2021 Jan 26. PMID: 33496986.
- A new method for three-dimensional magnetic resonance images denoising. Romdhane F., Faouzi B., Amiri H. International Journal of Computational Vision and Robotics. 2018 Jan; 8(1):1 DOI:10.1504/IJCVR.2018.10008239.

Keywords for searching in databases: Noise reduction, Denoising, NLM, Anisotropic Diffusion Tensor, XNAT

DQ7.2. Technical specifications

This part of the documentation is dedicated to providing all relevant technical specifications to prepare for the tool's integration into the EUCAIM test environment.

It is possible that the tool's integration will require, among others, some modifications in the input/output, or the inclusion of monitoring mechanisms.

Methods: in depth description of the methodology used for its development including all data preprocessing.
In the developed denoising tool, we normalise the input DICOM data to the maximum signal intensity in the whole volume before starting the denoising step in which:
- First We estimate the noise by using the Adaptive MAD (Median Absolute Deviation) estimator for Rician noise
- Then we calculate our proposed diffusion function from the noisy data based on The hyperbolic tangent function and to guarantee that it respects the contours, we have parameterized it by a edge indicator index which is a measure of confidence in order to discriminate structures located in the volume data.
- Then we combine the two Euclidean distances from the noisy and denoised data by our proposed diffusion function in order to get the final denoised data, by implementing Non local mean with the new proposed weight average.

Specific Technical information:

a. CPU
b. Programming language :Python and MATLAB
c. Expected RAM usage : depend on the data
d. Running mode (interactive/batch-based/case-based…): interactive
e. Software version : MATLAB R2023b,
f. Libraries : OS module in Python
g. Security measures: Writing to host data volume is restricted to a (predefined) non-root user. The tool does not require privileged container execution mode.

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
Simulated T1w MRI dataset from online database BrainWeb have been used to test the performance of the denoising tools. PSNR which quantifies the quality of the reconstruction of the denoised image compared to the ground image and the structural similarity index SSIM which describe how much similar the reconstructed denoised image to the ground truth have been used to evaluate the denoising results.

Access restriction: No access restriction to the source code or to the binaries of your tool.

DQ7.3. Integration validation

Communication channel for the helpdesk: email to feriel.ramdhane@unito.it and dario.longo@unito.it

User Manual
Usage instructions

User need to set up the input parameters such as : iter: number of iteration, rho (ρ): A standard deviation of the Gaussian kernel for the creation of the structure tensor [0, 0.1] and alpha(α): A single value to control the diffusion tensor matrix [0, 0.1]. The default values for the different parameters are defined as : iter =1, ρ =0.01 and α=0.01.

DQ8. MR image quality tool

Contributor: FORTH
Area: Imaging data quality assessment and imaging data curation
Status : under development
Summary: The tool is a comprehensive UI profiling image quality app for conventional or dynamic MR series to report image quality score and types of artifacts. SNR and CNR related metrics as well as a number of objective metrics are measured to support and attract the user's attention to the most suspicious slices for image degradation.

DQ8.1. Tool description for its conceptual validation

Tool description: The tool is developed to perform image quality assessment by experts/radiologists, providing also objective metrics for conventional and dynamic series. The tool provides a questionnaire to the experts to capture the etiology of image degradation. Furthermore, useful metadata for the acquisition protocol and supported graphs of objective quality metrics across slice location or time points are derived. Moreover, local quality assessment metrics can be calculated with user interaction for ROI delineation.

Data: Magnetic Resonance Imaging (MRI) Conventional and Dynamic Contrast Enhanced (DCE) images

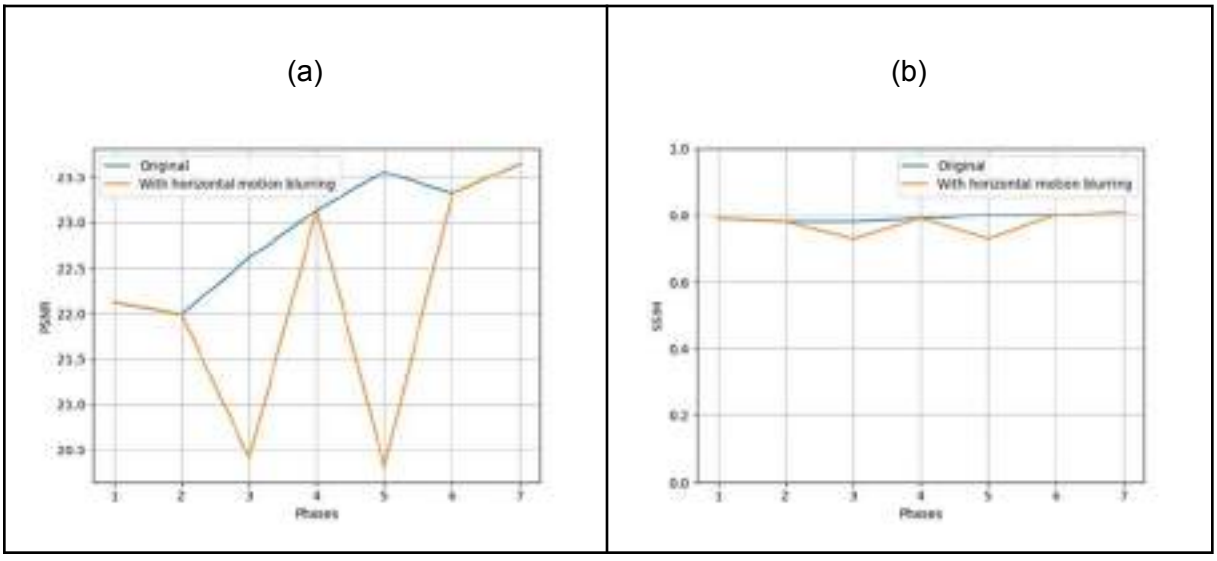
Methodology/performance:
The DICOM header of each image is used to derive useful metadata for the acquisition protocol. A questionnaire is presented to the user where he/she provides input by selecting the appropriate answer to 5-7 questions regarding different aspects of image quality. Furthermore, No Reference (NR) and Full Reference (FR) metrics are calculated on the raw images to provide objective quality metrics to the user. More functionalities are offered (ROI-based statistics related to quality) when the tool is used as a plug-in in a freely available DICOM viewer (Mango tool)

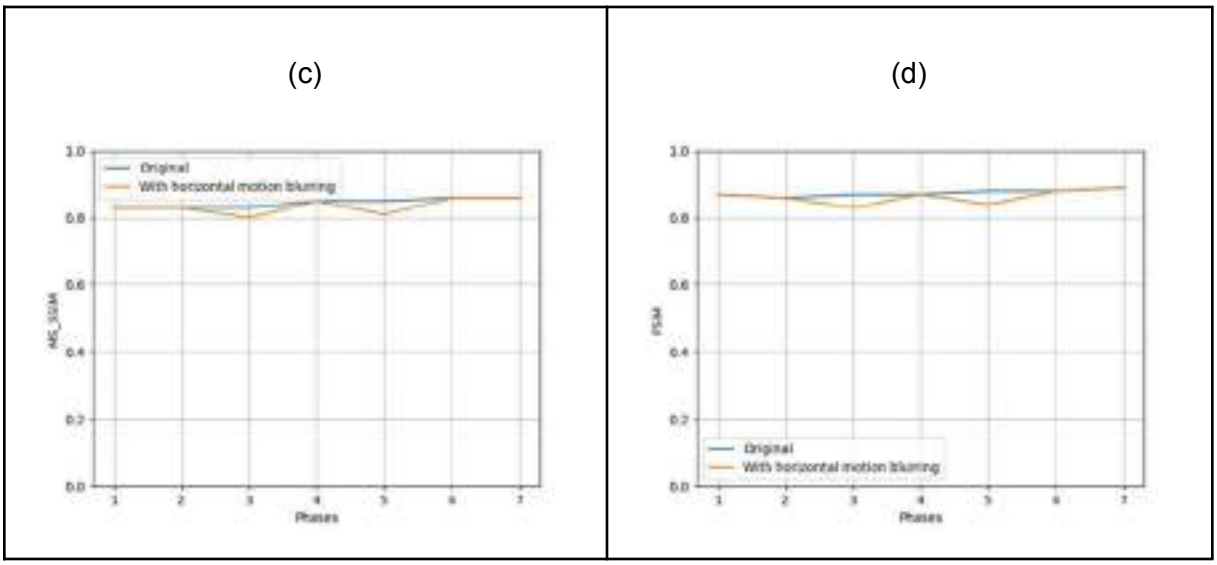
Use: brief description of the tool's functioning (if it applies).

The tool is used by human experts/ radiologists to assess the quality of an image. The experts should fill in the questionnaire and objective metrics are provided to guide them during the image evaluation where they identify major reasons for image quality degradation. To their support they consult objective metrics presented in graphs for intra sequence and inter-sequence (for dynamic acquisitions) evaluation.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : DICOM DCE or Conventional MR images
Output : a) a report in text file (answers to the questionnaire from the clinical experts regarding degrading factors), b) excel file with metadata, and c) jpeg graphs (showing the values of metrics).

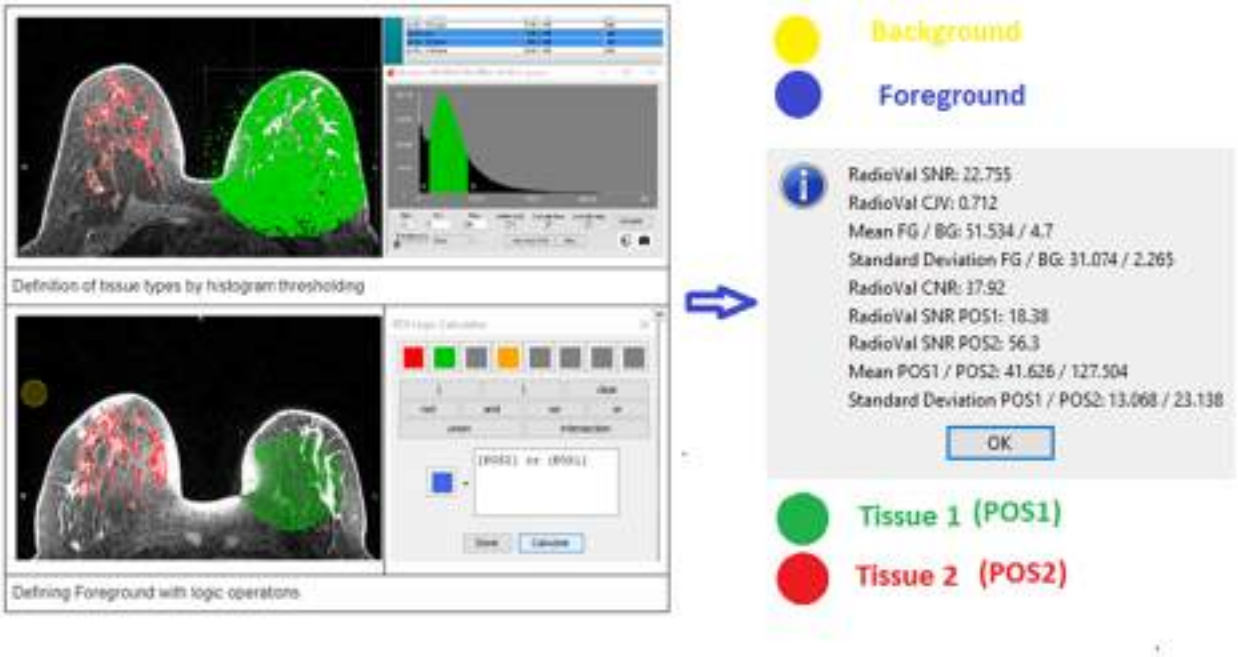
Quantitative results: performance obtained during training of the tool (if applies)
Feasibility testing of identifying degraded time points inside a time-series dynamic acquisition. Full-Reference metrics were calculated for all time phases for the original dataset, as well as the original dataset after applying blurring filters in time phases 3 and 5 out of 7. (a) Peak Signal-to-Noise Ratio (PSNR); (b) Structural Similarity Index Measure (SSIM); (c) Multi-Scale Structural Similarity Index Measure (MS-SSIM); (d) Feature Similarity Index Measure (FSIM).





Qualitative results: Provide some visual results (if available) of applying such tools.
Snapshot from the Windowed Application showing the distinct compartments: 1: Working Path, 2: DICOM tags extracted from the sequence headers, 3: Questionnaire to be addressed by the expert (only this part is mandatory for completing the IQA evaluation) 4`: Objective quality metrics for the total number of slices in the selected sequence. The windowed application does not support ROI-based SNR and CNR measurements.

Workflow for measurement of ROI-based metrics (segmentation based on histogram sub-sections, ROI logical operations for defining the foreground ROI) related to tissue-specific or local SNR and CNR.



Additional information:
Successful use cases: The tool was successfully applied to MR conventional images and MR DCE series of the publicly available DUKE breast dataset. Furthermore, the tool was successfully applied to conventional and DCE images from an indicative dataset of the RadioVal project.
Publication Link: https://www.mdpi.com/2313-433X/10/5/115

Keywords for searching in databases: Image Quality Assessment

DQ8.2. Technical specifications

Data: In depth description of the data used to train the tool.
Training of the tool is not required. Data from the RadioVal project and the publicly available DUKE dataset were used to develop the tool.

Methods: in depth description of the methodology used for its development including all data preprocessing.
The tool works as a window application or can be imported as a plugin into the Mango (freely-available software) DICOM viewer. The user interface is divided into three main compartments. The first shows metadata from the imaging sequence and the second shows a number of quality related questions to be answered by clicking on one of the provided options. The answers are saved in txt format and stored in the same folder as the evaluated image sequence, constituting the main output of the process. To extract the metadata for each image that describes the scanner and acquisition protocol used, the corresponding Dicom tags from the Dicom header were derived. For the reader's support a number of objective metrics are provided to improve intra and inter evaluator bias. The No-Reference (NR) metrics, BRISQUE

score and Total Variation, were calculated for each slice. Specifically for dynamic acquisitions, i.e. acquired sequentially in time with the same parameters, such as the Dynamic Contrast Enhanced (DCE) images, Full-Reference (FR) metrics (PSNR, SSIM, MS-SSIM, FSIM) across timepoints were calculated to capture motion-related artifacts or anatomy translocation. The use of objective metrics serves to provide measurable evidence for image quality and support the user in his/her task. There is the option for user interaction in the case of running the application inside Mango viewer of delineating ROIs and calculating statistical metrics related to regional image noise or contrast. This functionality is enabled from the Mango main plugin menu. It is initiated after importing a .jar file provided to the users, and added in the user's plugin library by the import option of the main menu.

Specific Technical information:

a. CPU
b. Programming language : Python 3.10.13
c. Expected RAM usage : Depending on the data size
d. Running mode (interactive/batch-based/case-based…)
e. Software version :
f. Libraries : tkinter ttkbootstrap pillow pydicom=2.4.3 pylibjpeg-libjpeg=1.3.4 numpy=1.26.0 pandas=2.1.3 piq=0.8.0 pytorch=2.1.1 torchvision=0.16.1 matplotlib=3.8.1 wxPython=4.2.0 pywin32=306 pywin32-ctypes=0.2.0

DQ9. Image Qure

Contributor: Medexprim
Area: Imaging data quality assessment and imaging data curation
Status : developped
Purpose : The main goal of Image Qure is to automatically analyze medical images and provide detailed quality metrics plus a global probability for this image to be classified as bad quality.

DQ9.1. Tool description for its conceptual validation

Keywords for searching in databases: image, imaging, noise, artefact, quality

DQ9.2. Technical specifications

Specific Technical information:

a. CPU
b. Programming language : python 3.8.10
c. Expected RAM usage : depend on the data, approx. 8Go
d. Running mode (interactive/batch-based/case-based…)
e. Software version :

f. Libraries : Pydicom: 2.1, Matplotlib: 3.4.2, Nibabel: 3.2.1, scikit-image: 0.18.2, pydicom: 2.1.2, pickle: 4.0, sklearn: 0.24.2, pandas: 0.23.0, pillow: 9.2, numpry: 1.23, scipy: 1.9.

DQ10 - Image Quality Assessment metrics for the XNAT platform

Contributor:CNR-IBB and EuroBioImaging
Area:Imaging data quality assessment
Status : Containerized
Purpose : Providing standardised and open source tools for assessing several image quality metrics for easy, robust and fast evaluation of clinical and preclinical image datasets.

DQ10.1. Tool description for its conceptual validation

Tool description: A set of five Image Quality Assessment (IQA) metrics that can assess perceptual image quality of clinical image datasets within the XNAT platform, both at subject level and at project (dataset) level.

Data: MRI (T1w, T1w contrast enhanced, T2w, FLAIR, DWI, DCE), CT and PET images in DICOM format.

Methodology/performance : Docker containers with python and Matlab scripts were deployed for the XNAT Platform.

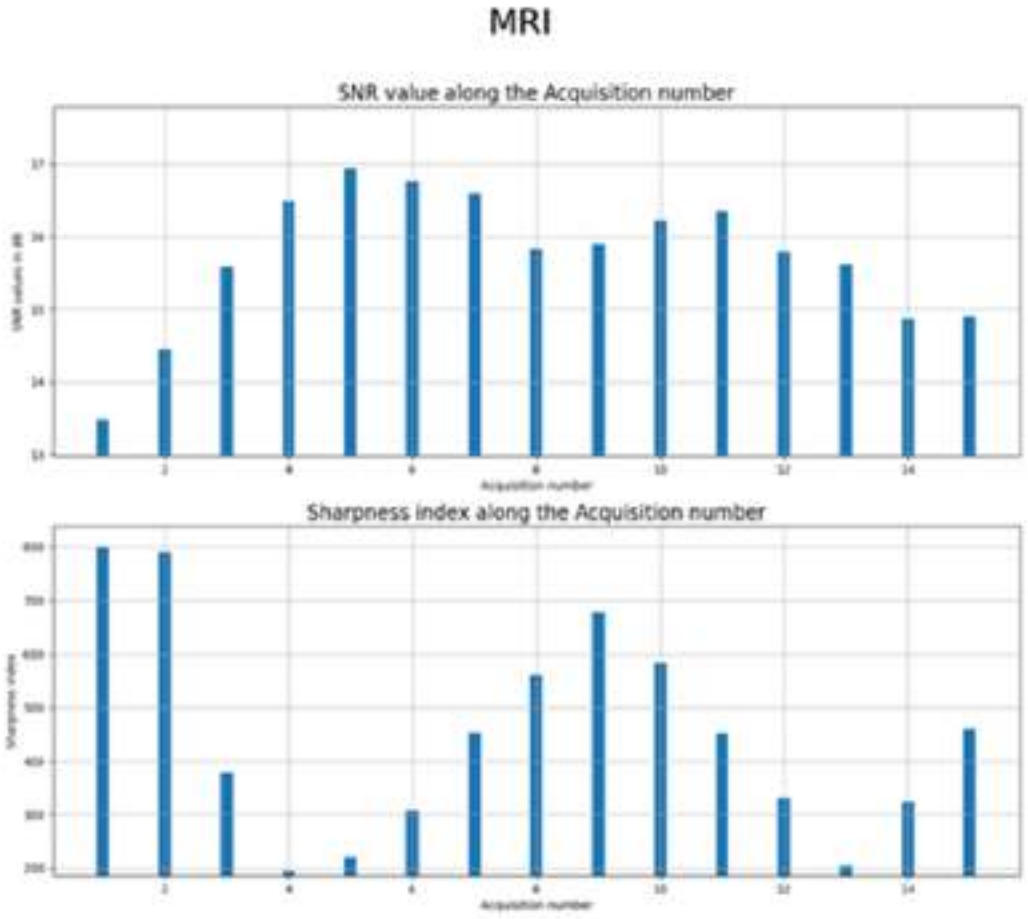
Use: brief description of the tool's functioning (if it applies).

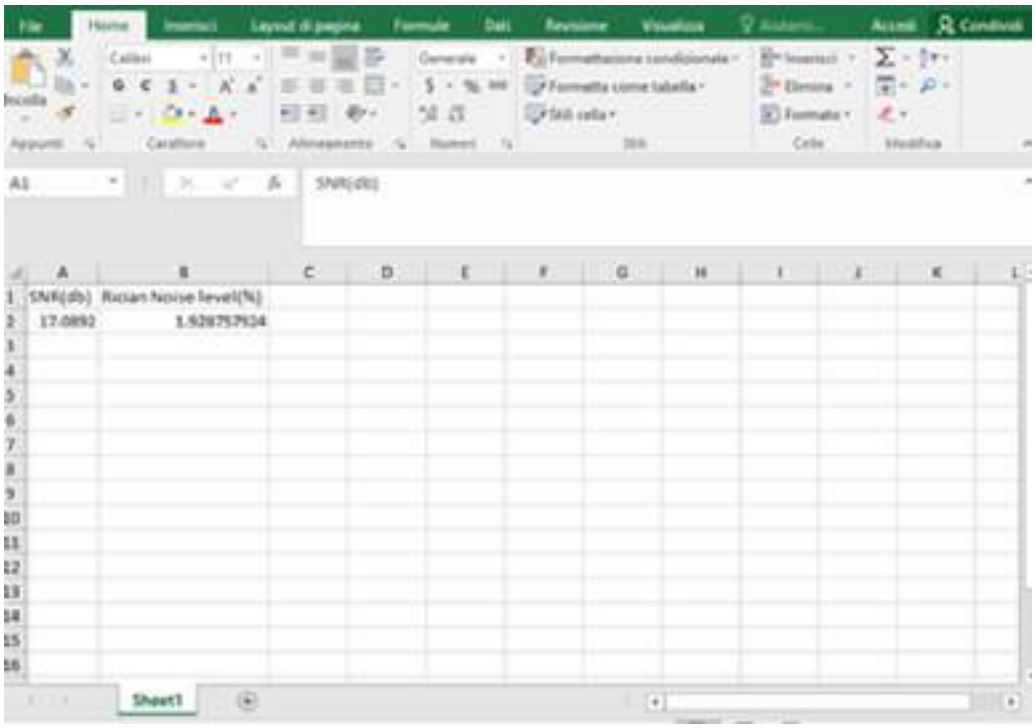
- Signal to Noise Ratio (SNR) is used to characterise image quality with higher values indicating better image quality
- Sharpness Index (SI) which refers to an image's overall clarity and detail with higher values indicating sharper images.
- Noise level estimation is based on the Mean Absolute Deviation to estimate the Rician noise in MRI.
- Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) : no-reference image quality score, uses Natural Scene Statistics (NSS) (It is concerned with the statistical regularities related to scenes) based model in spatial domain to perform quality estimation and distortion identification. The quality range is between 0 and infinite.
- Mutual Information (MI) calculates the mutual dependency with higher index corresponding to higher similarity between two consecutive acquisitions.

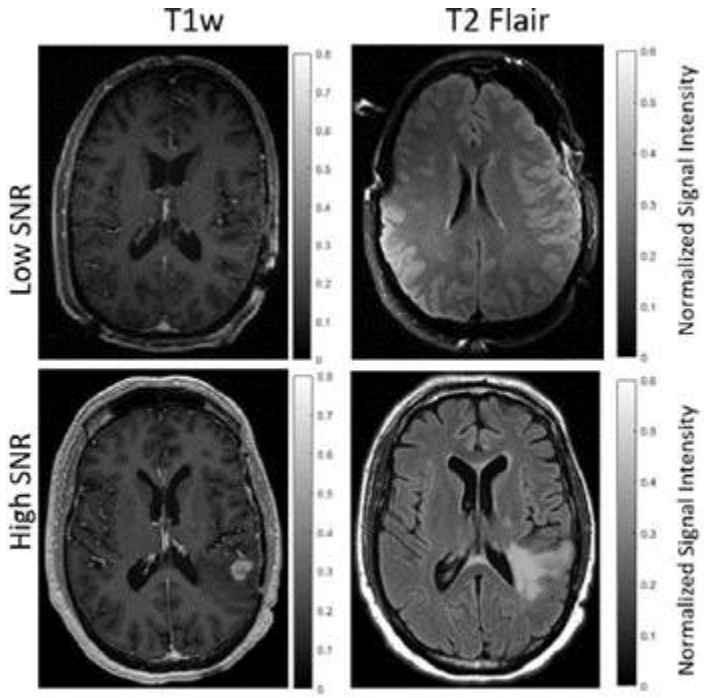
Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input: DICOM images
Output : PNG figure and xlsx/csv datasheet

Quantitative results: performance obtained during training of the tool (if applies)
For a public clinical MRI dataset including brain and breast images with overall 3000 MRI DICOM data (T1w, T2w FLAIR), we evaluated the quality of T1w dataset with SNR > 8.3 (dB) and SI > -1099.4 and of T2w FLAIR dataset with SNR > 4.11 (dB) and SI > -2357.59.

Qualitative results: Provide some visual results (if available) of applying such tools.







representative MRI images with High and Low SNR

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

Communications : Developing Efficient and Open-Source Metrics of Image Quality Assessment for Clinical and Preclinical Datasets. Romdhane F. et al. European Meeting of Molecular Imaging - EMIM 2024.

Keywords for searching in databases: Image Quality Assessment, SNR, SI, Noise level, BRISQUE, MI, XNAT

DQ10.2. Technical specifications

Data: In depth description of the data used to train the tool.

Methods: in depth description of the methodology used for its development including all data preprocessing.

Specific Technical information:

a. CPU
b. Programming language : Python and MATLAB
c. Expected RAM usage : depend on the data
d. Running mode (interactive/batch-based/case-based…) interactive
e. Software version : MATLAB R2023b,Libraries : pydicom, numpy, SimpleITK, opencv, matplotlib, pillow, Xlsxwrite, pandas, math, logerfc. Security measures: Writing to host data volume is restricted to a (predefined) non-root user. The tool does not require privileged container execution mode.

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
To validate the tools we tested one clinical MRI dataset with 76 subjects including brain and breast images coming from the public database [1, 2] and with overall 60 T1w and 124 T2 FLAIR volumes, and 3 volume of T1w, T2w and Flair from the Osirix database (https://www.osirix-viewer.com/resources/dicom-image-library/). We also used the interquartile range (IQR) and applied the outlier rule to categorise the results in enough image quality or not. IQR is a measure of spread and variability in a data, calculated by subtracting the third quartile Q3 (or the 75th percentile of the dataset) minus the first quartile Q1 (or 25th percent of the dataset).

1. Juvekar, P., Dorent, R., Kögl, F., Torio, E., Barr, C., Rigolo, L., Galvin, C., Jowkar, N., Kazi, A., Haouchine, N., Cheema, H., Navab, N.,Pieper, S., Wells, W. M., Bi, W. L., Golby, A., Frisken, S., & Kapur, T. (2023). The Brain Resection Multimodal Imaging Database (ReMIND) (Version 1) [dataset]. The Cancer Imaging Archive. https://doi.org/10.7937/3RAG-D070.
2. Newitt, D., & Hylton, N. (2016). Single site breast DCE-MRI data and segmentations from patients undergoing neoadjuvant chemotherapy (Version 3) [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2016.QHsyhJKy

Access restriction: Do you have any access restriction to the source code or to the binaries of your tool?
No

DQ11. Image Duplicates Checker

Contributor: Aristotle University of Thessaloniki (AUTH)
Area: Clinical and Imaging Data Quality assessment
Status : under development/testing
Summary: The tool is designed to search for DICOM images across one or more directories, compare the images retrieved, and identify any duplicates. It performs comparisons using a unique field in the DICOM metadata as well as image similarity metrics such as MSE, SSIM, and cosine similarity. The findings are then compiled and presented in a report

DQ11.1. Tool description for its conceptual validation

Tool description: The tool is designed to search for DICOM images across one or more directories, compare the images retrieved, and identify any duplicates. It performs comparisons using a unique field in the DICOM metadata as well as image similarity metrics such as MSE, SSIM, and cosine similarity. The findings are then compiled and presented in a report

Data: a dataset with DICOM series/images

Methodology/performance: The tool is containerized with Docker and will be distributed as a Docker image

Use: detailed documentation will accompany the tool

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : DICOM files (directory path with DICOM series/images)
Output : report with the results of each comparison

Qualitative results: Provide some visual results (if available) of applying such tools.
Images and figures will be added

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).
Licence: Custom Software License Agreement

Keywords for searching in databases: data curation, data quality control, data cleaning

DQ11. 2. Technical specifications

Data: data that are collected in the scope of the INCISIVE project or/and data from open repositories (e.g. TCIA). All data are DICOM images. Can work also with regular types of images e.g. “.jpeg”, “.png”
Specific Technical information:

a. CPU
b. Programming language : Python
c. Expected RAM usage : 16GB
d. Running mode (interactive/batch-based/case-based…)
e. Software version : 1.0
f. Libraries : pydicom, pandas
g. Security measures: requires sudo privileges (to run the docker image)

Traceability and monitoring mechanism: No such mechanism has been implemented, but a method for error logging can be integrated

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.

Access restriction : Yes, the source code should be protected and not accessible to other partners.

Additional information for tool integration: The tool is configured to run with DICOM image format. If a different image format is required e.g. jpeg slight modifications should be performed

DQ11.3. Integration validation

Communication channel for the helpdesk
difoto@auth.gr; loannach@auth.gr;

User Manual
A readme with instructions will be provided

DQ12. Extended a Priori Probability (EAPP) tool

Contributor: ITI
Area: Data quality assessment
Status : Containerized
Purpose: Method to calculate a more informative metric set than the simple a priori probability for estimating the difficulty or bias of the data in the context of a binary classification task.

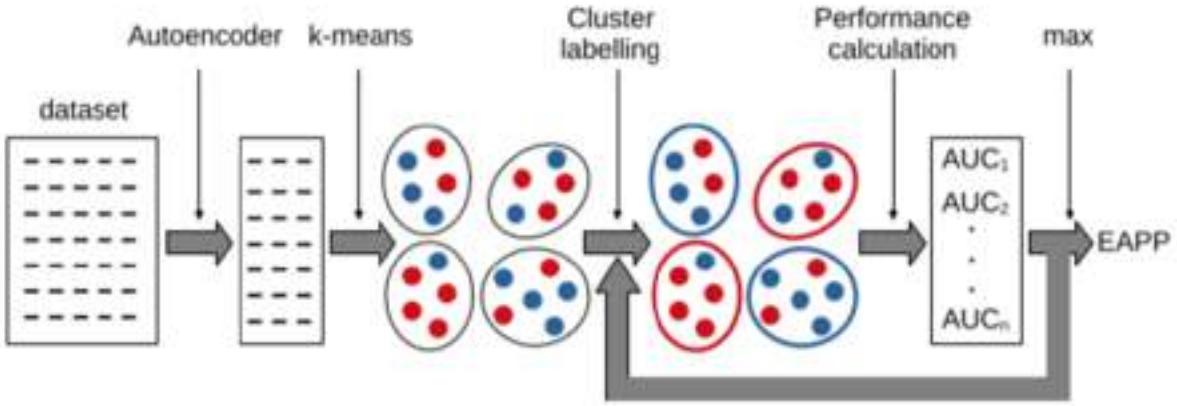
DQ12.1. Tool description for its conceptual validation

Tool description: The tool provides a semi-supervised metric for binary classification tasks that considers not only the a priori probability but also some possible bias present in the dataset, as well as other features that could provide a relatively trivial separability of the target classes. Therefore, it allows for evaluating the ease or complexity of the task or bias of the data beyond the well-established baseline for any binary classification.

Data: The EAPP algorithm accepts both tabular data and images as input.
To test the EAPP behaviour for diverse binary classification tasks, a wide range of scenarios was covered so different well-known image datasets were analyzed:

a. since the EAPP deals with binary classification tasks, a subset of the handwritten digits dataset MNIST [1] was extracted. The images of '1' and '7', which are relatively similar, were compared to those of '8'.
b. The ImageNet dataset [2] comprises 3.2 million images covering up to 20000 categories. Under the assumption that the categories mushroom and wedding may have different environmental elements, we selected the images of these two classes, looking at whether the background that does not contain the object could introduce bias to the dataset.
c. We also evaluated the metric using the BIMCV-PADCHEST [3] chest x-ray image dataset, a dataset with the presence of bias, and using (d) a subset of the nCOV2019 dataset [4] which contains potential bias sources.

Methodology/performance: The approach is based on the area under the ROC curve (AUC ROC), known to be quite insensitive to class imbalance. The procedure involves multiobjective feature extraction and a clustering stage in the input space with autoencoders and a subsequent combinatory weighted assignation from clusters to classes depending on the distance to nearest clusters for each class. Class labels are then assigned to establish the combination that maximizes AUC ROC for each number of clusters considered. To avoid overfit in the combined feature extraction and clustering method, a cross-validation scheme is performed in each case. EAPP is defined for different numbers of clusters, starting from the inverse of the minority class proportion, which is useful for a fair comparison among diversely imbalanced datasets. This metric represents a baseline beyond the a priori probability to assess the actual capabilities of binary classification models. The EAPP process is shown in the following figure:



Use: brief description of the tool's functioning (if it applies).
The process does not need any interaction from the user since it is fully automatic. The tool will provide the EAPP value for the provided input.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).

Input : As previously commented, EAPP deals with binary classification tasks, so the tool receives two files, one for class 0 and the other for class 1. Tabular data must be in CSV format, and images must be in jpeg or png format and packed in a zip file.
Output : The tool returns a numeric value corresponding to the EAPP metric.

Quantitative results: performance obtained during training of the tool (if applies)
Quantitative results are shown using the EAPP metric. A high EAPP usually relates to an easy binary classification task, but it also may be due to a significant coarse-grained bias in the dataset, when the task is previously known to be difficult.

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).
There is an online version of the tool aimed for demonstration at request.

Publications
Extended a Priori Probability (EAPP): A Data-Driven Approach for Machine Learning Binary Classification Tasks. Ortiz V., Perez-Benito FJ., Del Tejo Catala O., Igual IS., January 2022; IEEE Access PP(99):1-1 ; DOI:10.1109/ACCESS.2022.3221936.

Datasets used to test the EAPP behaviour:
See bibliography section below
Keywords for searching in databases: bias, clustering, autoencoder, combinatory, a priori probability, binary classification, semi-supervised, EAPP.

Bibliography

1. Y. LeCun, C. Cortes, and C. J. C. Burges. (Jan. 2022). The MNIST Database of Handwritten Digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/
2. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘‘ImageNet A large-scale hierarchical image database,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.
3. A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, ‘‘PadChest: A large chest X-ray image dataset with multi-label annotated reports,’’ Med. Image Anal., vol. 66, Dec. 2020, Art. no. 101797.
4. B. Xu, B. Gutierrez, S. Mekaru, K. Sewalk, L. Goodwin, A. Loskill, E. L. Cohn, Y. Hswen, S. C. Hill, and M. M. Cobo, ‘‘Epidemiological data from the COVID-19 outbreak, real-time case information,’’ Sci. Data, vol. 7, no. 1, pp. 1–6, 2020.

DQ12.2. Technical specifications

Data: In depth description of the data used to train the tool.
First of all, mention that the complete algorithm is semi-supervised, so it faces the problem of overfitting as do others of this kind. For this reason, a 10 cross-validation was established as a standard for all our experiments. To be precise, for each dataset, the whole process is split into two parts: the training process, in which 90% of the data is used to learn all clustering schemes

for any k (centroids) and a particular internal representation, and the testing process, in which the remaining 10% is processed with all these parameters obtained by training. This procedure is repeated 10 times for each experiment to allow a fairer comparison throughout all datasets.
Four well-known datasets were used in this process:

a. MNIST is a standard database of handwritten digits commonly used for training classifiers. It consists of the 10 different arabic numerals, but our approach is defined for binary classification. Therefore, we selected the set {‘1’, ‘7’} for the first class, and {‘8’} for the second class.
b. With MNIST experiments, a particular spatial distribution for bright pixels is easily noticeable. To overcome this, a slightly more complex dataset with more image richness is used: ImageNet. It is another well-established image dataset containing more than 20,000 categories. Still, again, we focused on two visually different categories (wedding and mushroom) to work with an appropriate subset for binary classification.
c. The BIMCV-PADCHEST chest x-ray image dataset, a dataset with the presence of bias.
d. The algorithm was also applied to the numerical, structured nCOV2019 dataset to evaluate the difficulty of a binary classification task that did not deal with image data and that, in addition, contains potential bias sources.
   Finally, mention that, for nCov2019 dataset, the cross-validation process was 50-fold because of its reduced size, as it was not appropriate to suppress 10% of the data for training in each fold. Thus, we trained with 98% of the data in each iteration.

Methods: in depth description of the methodology used for its development including all data preprocessing.

The main objective of this methodology is to obtain a simple semi-supervised metric that allows evaluation of the ease or complexity of the task beyond the well-established baseline for any binary classification (namely, the a priori probability, that is, the proportion of examples belonging to the majority class). If the task is previously known to be difficult but the EAPP is high, then the dataset is likely to have a heavy bias, as a simple algorithm may be able to tell the difference between examples from two classes without supervision, using only the locality of the observations in the representation space.

For this purpose, a preprocessing step was performed as the analyzed image datasets contained images of multiple shapes. As Neural Networks are used and scale invariance is not our focus, the task is simplified, resizing all images to the same shape to perform feature extraction. Therefore, they were cropped as a square, keeping the same image center, and resized to 128 × 128 pixels. Regarding numerical datasets, all raw features were separately normalized (mean 0 and standard deviation 1).

EAPP aims to assess how well a non-supervised feature extraction method automatically splits classes into different clusters. The process is based on assigning the same class to all the examples that fall into the same cluster. This is done iteratively for various numbers of clusters and combinations of class assignments. A probability of belonging to a class is assigned to each observation depending on its distance to the centroids of the nearest positive and negative classes’ clusters. The process is divided into three stages: feature extraction, clustering, and combinatory analysis.

Labels must not be used during the training phase to compute the EAPP metric. Therefore, feature extraction is performed using unsupervised learning methods only. Algorithms such as Convolutional AutoEncoders (CAEs) are valid candidates and are indeed used for this methodology.

Additionally, this algorithm should group observations that have similar features, as they are likely to be from the same class. Therefore, clustering algorithms such as K-means are helpful to find the inner clusters that group samples of the dataset. Even if only two classes are present within the data, we cannot assume that the latent representations of both classes are linearly separable. The optimal number of clusters is unknown, so the algorithm should explore multiple values up to a certain limit.

Once a particular clustering scheme has been computed for a given feature representation, the aim is to assess the ease or complexity of the classification task by searching a higher baseline above the a priori probability, dependent only on the number of instances of each class. The underlying idea is, once the clustering process converges for a range of different cluster cardinalities k in the training stage, to save the model information (namely, the centroids) and apply this clustering to new test data but for each cluster assignation (for each k considered), assigning all the possible different combinations of binary labels to the clusters, leaving out the trivial ones (the extreme all negative and all-positive correspondences, since they would lead to a strongly unbalanced, useless classification). Moreover, we sort the set of observations based on the distance to the inferred clusters. The assignation to a binary class for each example in each combination is then performed depending on the distance between that instance and the centroids of the nearest positive and negative class clusters to make a continuous score available. Hence, to sum up, for each k, we compute all the possible binary assignations, leaving out both extreme configurations, and we obtain a similarity indicator depending on the distance to the nearest positive and negative clusters. Then, using this sorting procedure, we compute a performance index, in this case, the Area Under the ROC, and select the assignation that leads to the best score.

By using the previously mentioned datasets, this methodology has proven beneficial to preliminarily assess the difficulty of a binary classification task and suggest a certain level of bias in cases where the task is perceived to be easy and a high EAPP is found. Thus, our metric represents a baseline beyond the a priori probability to assess the actual capabilities of binary classification models.

Specific Technical information:

a. CPU: 8 cores
b. Programming language : python
c. Expected RAM usage : at least 8 GB (depends on the amount of input data). 8GB allows for approximately 500 images (524 x 524).
d. Running mode (interactive/batch-based/case-based…): batch-based
e. Software version : v1.0.1
f. Libraries : docker
g. Security measures: writing to host data restricted to a non-root user ; container does not require it to be executed in a privileged mode

Traceability and monitoring mechanism.
Not applicable

Unitary tests: description of the tests implemented to verify the correct functioning of the tool.
Yes, there are unitary tests for all the main functions of the tool.

Access restriction : do you have any access restriction to the source code or to the binaries of the tool?
The tool is dockerized and it can be executed in this environment; however, access to the source code or the binaries is not permitted.

Additional information for tool integration.
The tool will be modified to accept images in DICOM format (focusing on 2D images)

DQ13. Data Integration Quality Check Tool (DIQCT)

Contributor: Aristotle University of Thessaloniki (AUTH)
Status : Developed
Area: Imaging data quality assessment and imaging data curation
Clinical and Imaging Data Quality assessment
Summary: A tool that checks the clinical metadata quality (validity, completeness), the integrity between images and clinical metadata provided, the de-identification protocol applied, imaging analysis requirements and existence of annotation and informs the user on corrective actions prior to data upload.

DQ13.1. Tool description for its conceptual validation

Tool description: This tool's aim is to ensure that the requirements for data quality are met and informs the user for corrective actions. This tool is a rule-based tool that compares the input data with a predefined rule set and produces a series of reports. It consists of a total of 9 components divided in 3 categories: (i) Clinical metadata, (ii) Connection between clinical metadata and images, (iii) DICOM images.

Data: A whole dataset including clinical and imaging data.

Methodology/performance:
The tool's components were developed in python and R programming languages, incorporated in an RShiny-based user interface and provided as a stand-alone docker image.

Use: A complete user guide is provided.

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).

Input: A dataset with the following format: a directory containing: (i) an excel file - template for the clinical data and, (ii) subdirectories for each one of the patients containing the imaging modalities.
Output: Reports in a visual format in the user interface and in csv files placed in a separate folder in the initial dataset's path.

Quantitative results: performance obtained during training of the tool (if applies): N/A

Qualitative results: Provided in detail in the user guide.

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).
Publications: Kosvyra, A., Filos, D., Fotopoulos, D., Tsave, O. and Chouvarda, I., 2022, July. Data Quality Check in Cancer Imaging Research: Deploying and Evaluating the DIQCT Tool. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 1053-1057). IEEE.]

Successful use cases: Already used in INCISIVE project

Licence : Custom Licence Software Agreement
Acknowledgements: INCISIVE's partners (Data Providers)

Keywords for searching in databases: data quality, imaging data, clinical data, completeness, validity, integrity, consistency, accuracy, uniqueness

DQ13.2. Technical specifications

Data: The data used to test and validate the tool was the data that populated the INCISIVE repository, including clinical and imaging data.

Methods: The methodology was: (i) set the requirements, (ii) define the rules that data should follow based on the requirements, (iii) develop each component that checks if the data comply with a rule and produces a report, (iv) incorporate the components in user interface, (v) dockerize the application.

Specific Technical information:

a. CPU
b. Programming language : R and Python
c. Expected RAM usage : 16Gb
d. Running mode (interactive/batch-based/case-based…)
e. Software version : v5
f. Libraries : docker

g. Security measures: requires sudo privileges (to run the docker image)

Traceability and monitoring mechanism.
Traceability and monitoring mechanism: Full documentation and recording of each version of the tool, with information on added components or improvements in visualization or functionality. Continuous monitoring of the tool's use from the data collectors for each version release to get feedback on possible errors or functionality improvements. Moreover, datasets are uploaded along with the reports produced by the tool. The reports are date marked so the reduction of errors can be monitored.

Unitary tests: In the first place, internal validation was conducted with the use of mock datasets with various induced errors and tested the components for possible bugs and oversights. In the second place, external validation was conducted in collaboration with a small number of data providers, to identify possible errors and test the performance of the tool. Feedback was used to improve the functionality of the tool. Finally, a user evaluation questionnaire was circulated to gain insights on the user experience.

Access restriction : do you have any access restriction to the source code or to the binaries of the tool? Yes, the source code should be protected and not accessible to other partners.

Additional information for tool integration. Possible adaptations: (i) Adapt the tool to read the clinical data in a different format, (ii) adapt the tool to the rules deriving from the requirements of EUCAIM

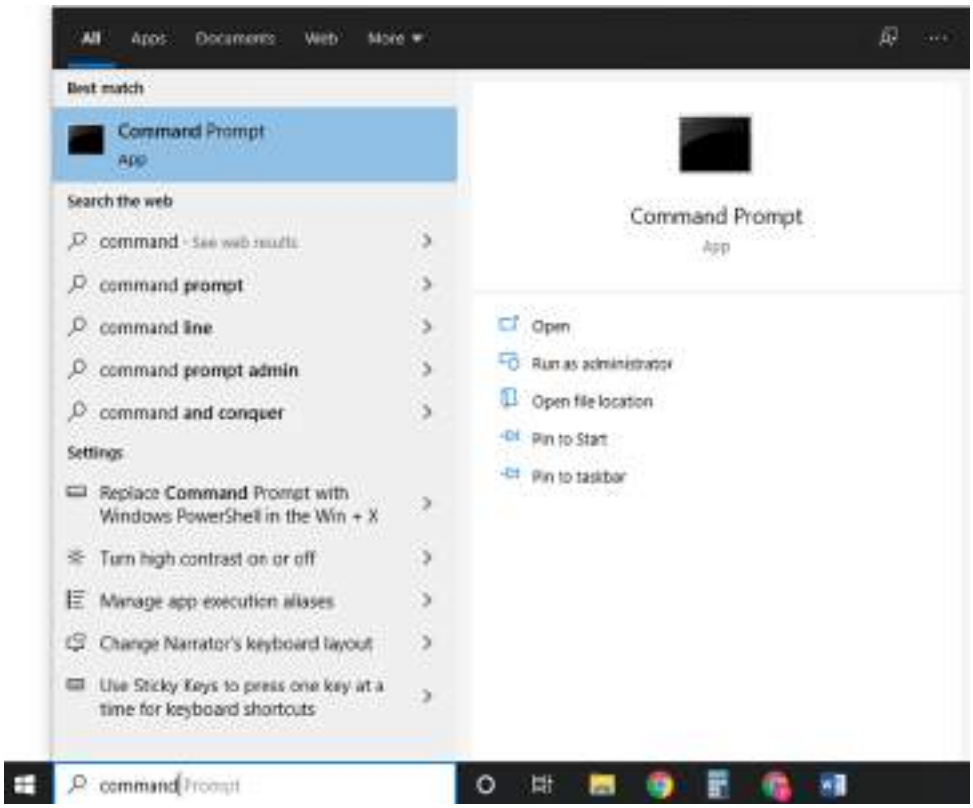
DQ13.3. Integration validation

Communication channel for the helpdesk: aekosvyra@auth.gr, dimfilos@auth.gr, ioannach@auth.gr

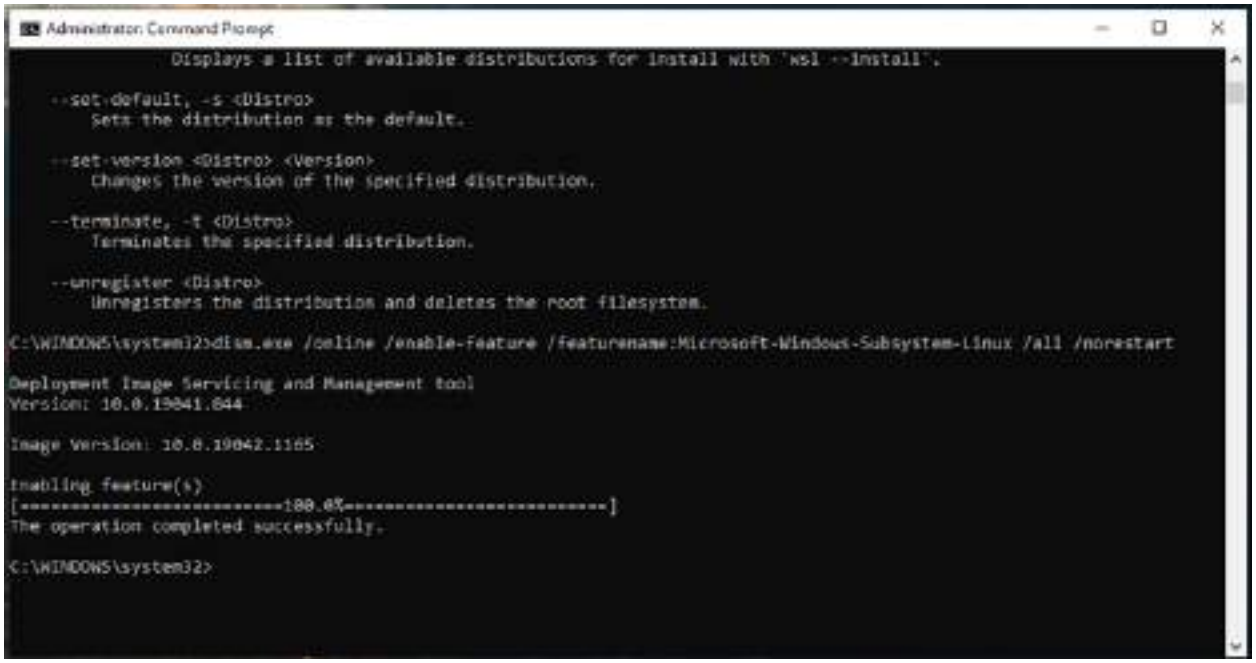
Most common errors
- Excel file with clinical data not found in directory
- Data structure is not right

User Manual
Installation Guidelines for the Data Integration Quality Check Tool

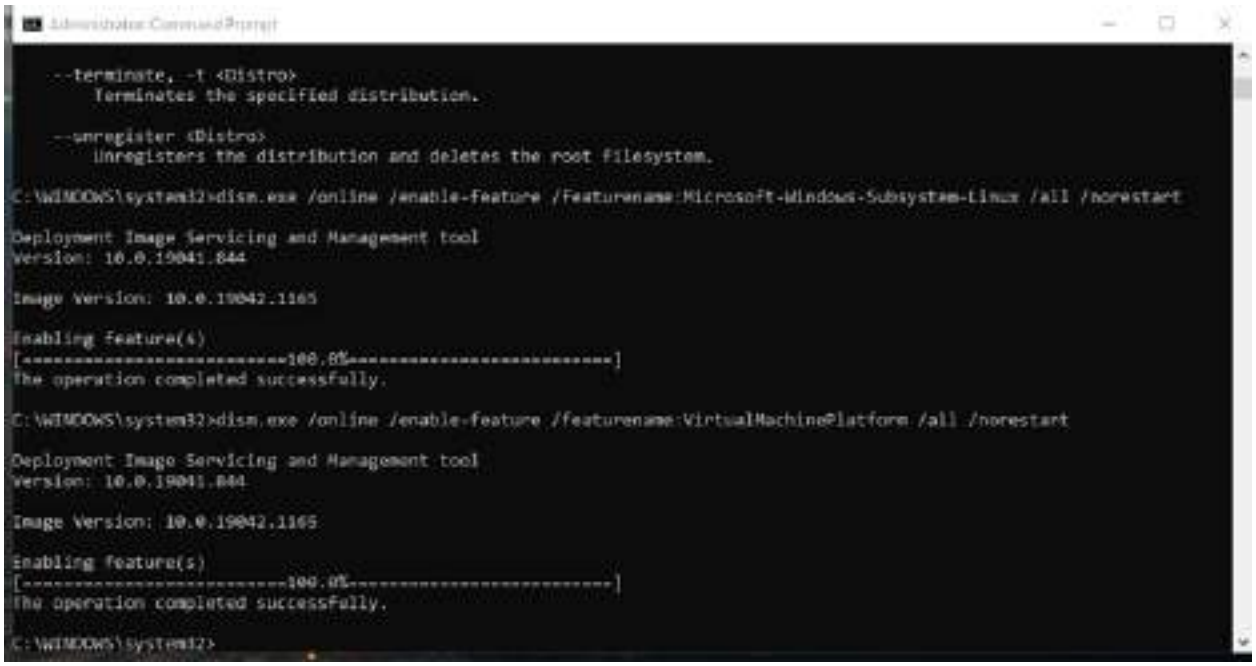
1. Windows
   1.1. Step 1: Install Docker Engine
   You first need to install Windows Subsystem for Linux
      a. Open PowerShell as Administrator:



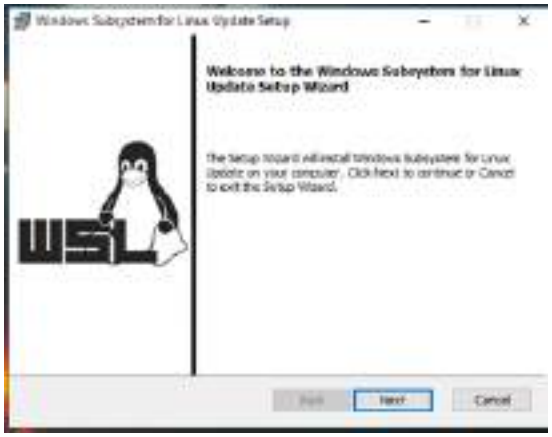
b. Copy and paste the following command to Enable the Windows Subsystem for Linux : “dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart”

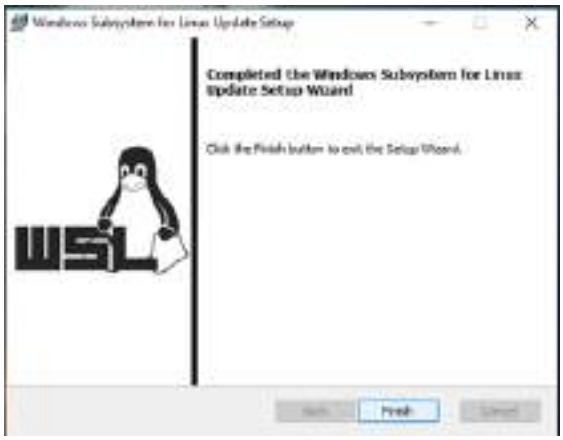


c. Copy and paste the following command to Enable Virtual Machine feature: “dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart”

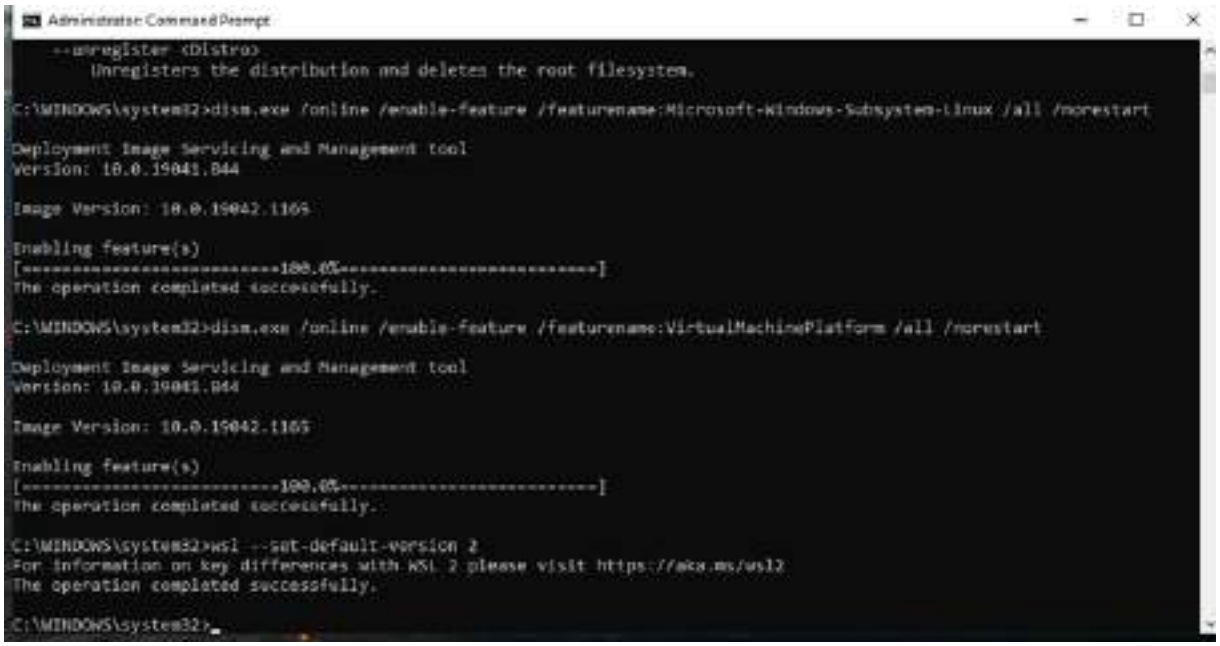


d. Restart your computer
e. Download the Linux kernel update package from here
f. Run the update package downloaded in the previous step. (Double-click to run - you will be prompted for elevated permissions, select ‘yes’ to approve this installation.)

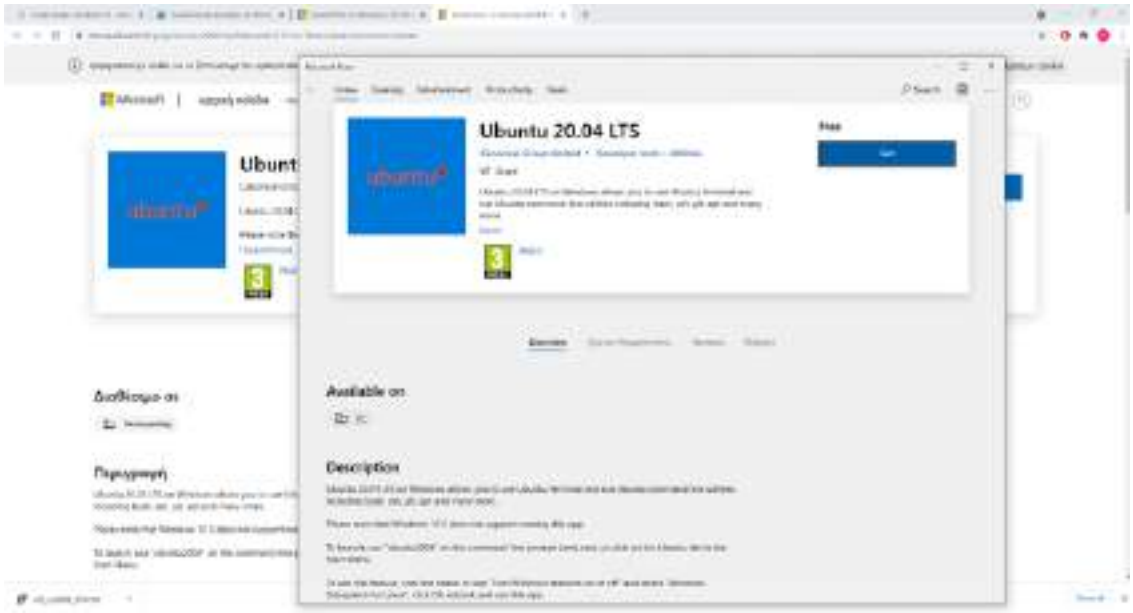


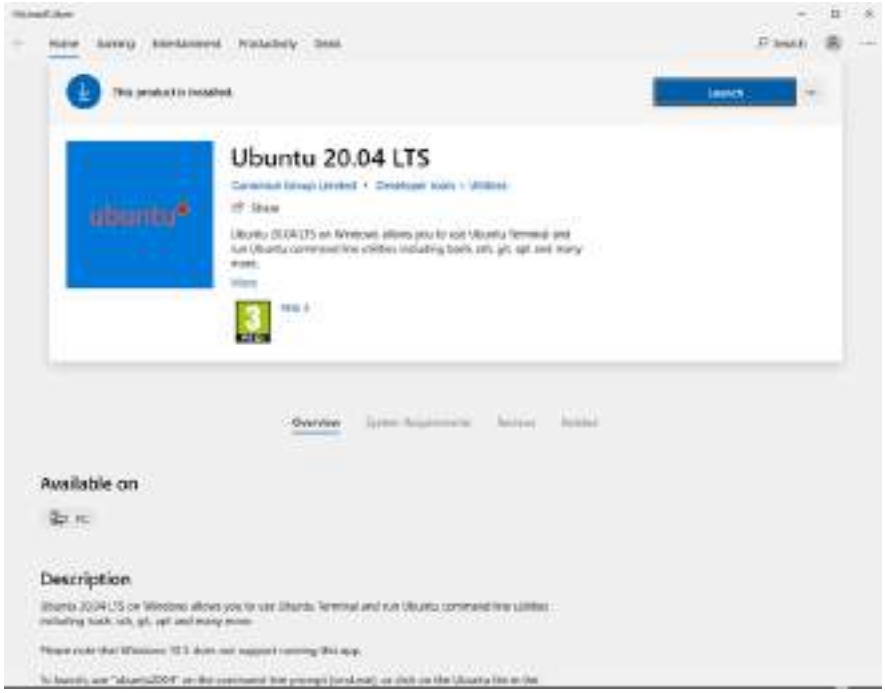


g. Copy and paste the following command to Set WSL 2 as your default version: “wsl --set-default-version 2”

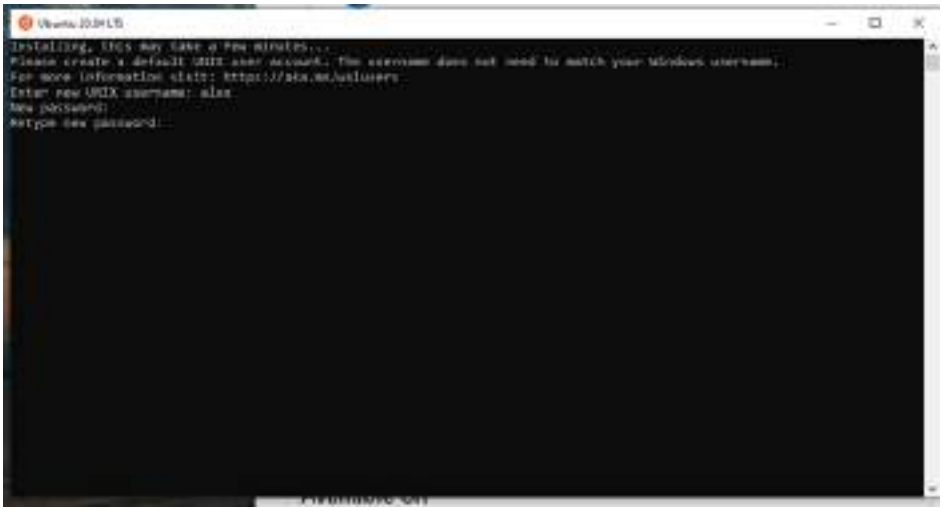


h. Install your Linux distribution of choice from here
i. Click on ‘Get’. Then, the Microsoft store opens.
j. Click on ‘Get’ again. The ubuntu 20.04 distribution is downloading. Wait until this procedure is over and then click on ‘Launch’





k. A terminal open and you will then need to create a user account and password for your new Linux distribution. Type a username and click enter. Type a password and click enter. retype your password and hit enter.



1.2. Step 2: Install Docker Desktop for Windows

1.2.a. Download docker desktop installer from here

1.2.b. Double-click **Docker Desktop Installer.exe** to run the installer

1.2.c. When prompted, ensure the **Enable Hyper-V Windows Features** or the **Install required Windows components for WSL 2** option is selected on the Configuration page

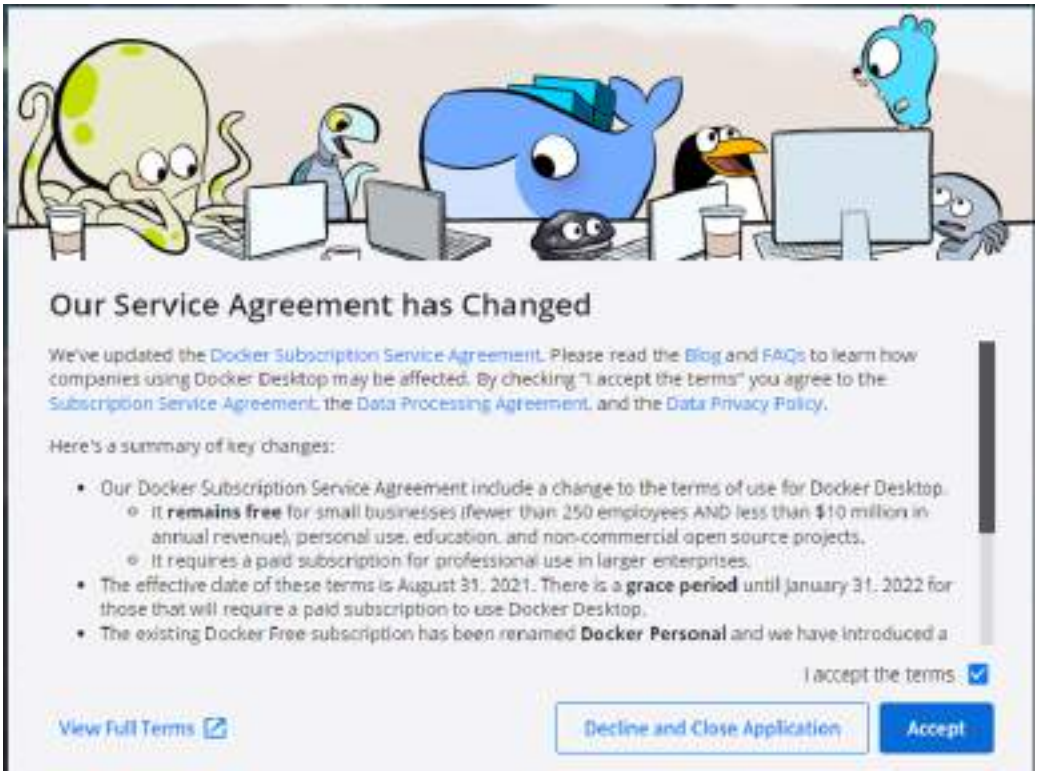
1.2.d. Follow the instructions on the installation wizard to authorize the installer and proceed with the install

1.2.e. When the installation is successful, click Close to complete the installation process.

1.2.f. ** If your admin account is different to your user account, you must add the user to the docker-users group. Run Computer Management as an administrator and navigate to Local Users and Groups > Groups > docker-users. Right-click to add the user to the group. Log out and log back in for the changes to take effect.

1.3. Step 3: Start Docker Desktop and Run the Quality tool

1.3.a. Double click on the whale icon on Desktop and accept the terms and conditions

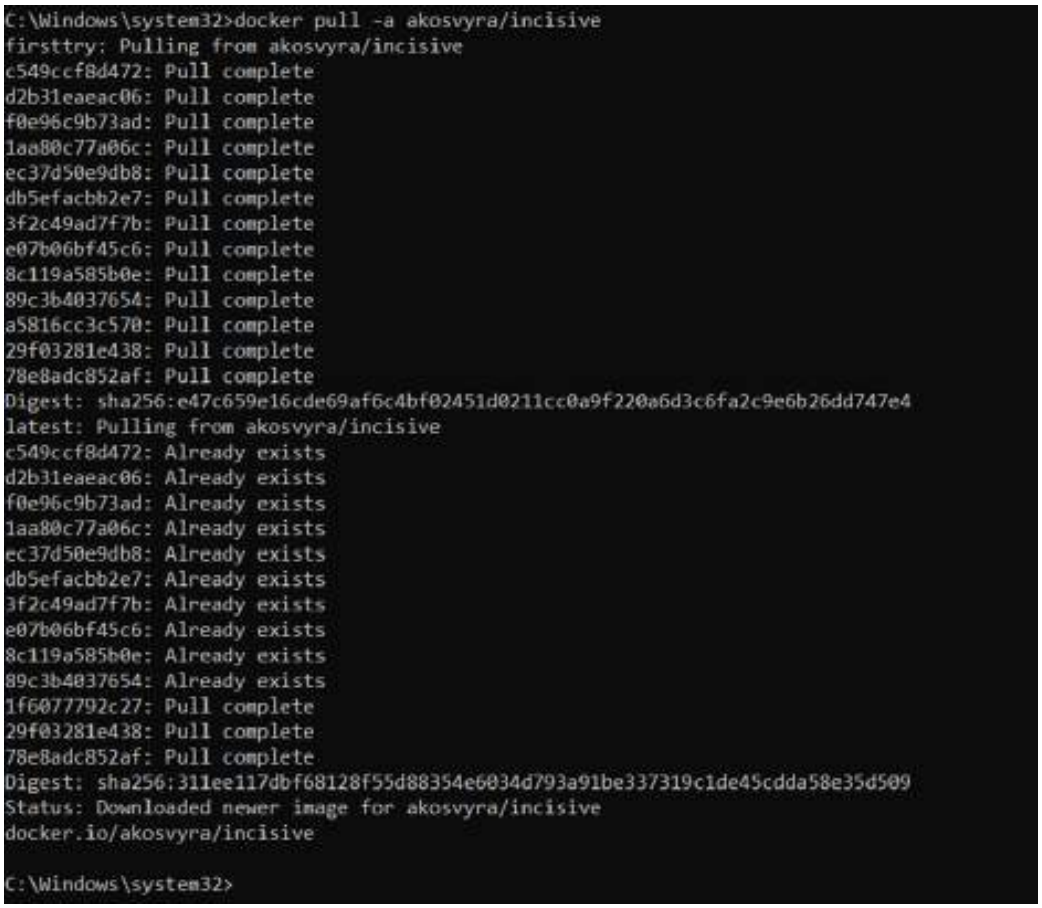


1.3.b. Restart your computer

1.3.c. Open a command prompt as administrator

1.3.d. Type: docker pull image_name

1.3.e. Wait until the download is complete



1.3.f. Type: sudo docker run -d --rm -p 3838:3838 -v {path-to-data}:/home image_name

Example: sudo docker run -d --rm -p 3838:3838 -v E:\INCISIVEdata:/home image_name

2. MAC

2.1. Step 1: Install docker engine

In this link you can find the instructions on how to install docker:
https://docs.docker.com/desktop/mac/install/

2.2. Step 2: Download image and Run the Quality Tool

In a terminal type: docker pull image_name
Wait until the download is complete

```
C:\Windows\system32>docker pull -a akosvyra/incisive
firsttry: Pulling from akosvyra/incisive
c549ccf8d472: Pull complete
d2b31eaeac06: Pull complete
f0e96c9b73ad: Pull complete
1aa80c77a06c: Pull complete
ec37d50e9db8: Pull complete
db5efacbb2e7: Pull complete
3f2c49ad7f7b: Pull complete
e07b06bf45c6: Pull complete
8c119a585b0e: Pull complete
89c3b4037654: Pull complete
a5816cc3c570: Pull complete
29f03281e438: Pull complete
78e8adc852af: Pull complete
Digest: sha256:e47c659e16cde69af6c4bf02451d0211cc0a9f220a6d3c6fa2c9e6b26dd747e4
latest: Pulling from akosvyra/incisive
c549ccf8d472: Already exists
d2b31eaeac06: Already exists
f0e96c9b73ad: Already exists
1aa80c77a06c: Already exists
ec37d50e9db8: Already exists
db5efacbb2e7: Already exists
3f2c49ad7f7b: Already exists
e07b06bf45c6: Already exists
8c119a585b0e: Already exists
89c3b4037654: Already exists
1f6077792c27: Pull complete
29f03281e438: Pull complete
78e8adc852af: Pull complete
Digest: sha256:311ee117dbf68128f55d88354e6034d793a91be337319c1de45cdda58e35d509
Status: Downloaded newer image for akosvyra/incisive
docker.io/akosvyra/incisive

C:\Windows\system32>
```

In a terminal type: sudo docker run -d --rm -p 3838:3838 -v {path-to-data}:/home image_name

3. Ubuntu
    3.1. Step 1: Set up the repository

In a terminal type:

1. sudo apt-get update
2. sudo apt-get install \

ca-certificates \
curl \
gnupg \
lsb-release

3. sudo mkdir -p /etc/apt/keyrings
4. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

3.2. Step 2

In a terminal type:

1. sudo apt-get update
2. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

3.3. Step 3

In a terminal type: sudo docker pull image_name
Wait until the download is complete

```
C:\Windows\system32>docker pull -a akosvyra/incisive
firsttry: Pulling from akosvyra/incisive
c549ccf8d472: Pull complete
d2b31eaeac06: Pull complete
f0e96c9b73ad: Pull complete
1aa80c77a06c: Pull complete
ec37d50e9db8: Pull complete
db5efacbb2e7: Pull complete
3f2c49ad7f7b: Pull complete
e07b06bf45c6: Pull complete
8c119a585b0e: Pull complete
89c3b4037654: Pull complete
a5816cc3c570: Pull complete
29f03281e438: Pull complete
78e8adc852af: Pull complete
Digest: sha256:e47c659e16cde69af6c4bf02451d0211cc0a9f220a6d3c6fa2c9e6b26dd747e4
latest: Pulling from akosvyra/incisive
c549ccf8d472: Already exists
d2b31eaeac06: Already exists
f0e96c9b73ad: Already exists
1aa80c77a06c: Already exists
ec37d50e9db8: Already exists
db5efacbb2e7: Already exists
3f2c49ad7f7b: Already exists
e07b06bf45c6: Already exists
8c119a585b0e: Already exists
89c3b4037654: Already exists
1f6077792c27: Pull complete
29f03281e438: Pull complete
78e8adc852af: Pull complete
Digest: sha256:311ee117dbf68128f55d88354e6034d793a91be337319c1de45cdda58e35d509
Status: Downloaded newer image for akosvyra/incisive
docker.io/akosvyra/incisive

C:\Windows\system32>
```

In a terminal type: sudo docker run -d --rm -p 3838:3838 -v {path-to-data}:/home image_name

Usage instructions

Full documentation is available here, here.

DQ14. Denoising-Inhomogeneity Correction Tool

Contributor: HULAFE

Area: Imaging data quality

Summary: A customisable image pre-processing tool that performs 5 of the most common denoising filters and ANTS N4 bias field correction filter. The tool has two independent steps, one for the denoising steps and another for the N4 filter, which can be configured as some of their parameters using a parameter configuration Json file. In particular, the parameter configuration of this tool has been optimised for TW1, T2W, DWI and DCE sequences in neuroblastoma and paediatric brain tumours.

Status : Containerized

Purpose : Denoising filter and N4 bias field correction filters to increase the quality of MR images.

DQ14.1. Tool description for its conceptual validation

Tool description: The tool is designed to perform a customisable image pre-processing to reduce noise and inhomogeneity field effect, thus improving image quality and reproducibility of radiomics features. This tool consists of two independent steps: one for denoising using one of the 5 integrated filters (Bilateral Filter, Anisotropic Diffusion Filter (ADF), Curvature Flow Filter (CFF), SUSAN and Non Local Means (NLM)), and another for the ANTs N4 and another for the ANT's N4 bias correction filter. The parameter configuration of this tool has been optimised for TW1, T2W, DWI and DCE sequences in neuroblastoma (NB) and paediatric brain tumours, but it can also be configured with some of their parameters using a JSON parameter configuration file.

Data: The tool is optimised for TW1, T2W, DWI and DCE sequences in neuroblastoma (NB) and paediatric brain tumours in Magnetic Resonance Imaging (MRI), but also can be used in other modalities such as Computed Tomography (CT) or Positron Emission Tomography (PET).

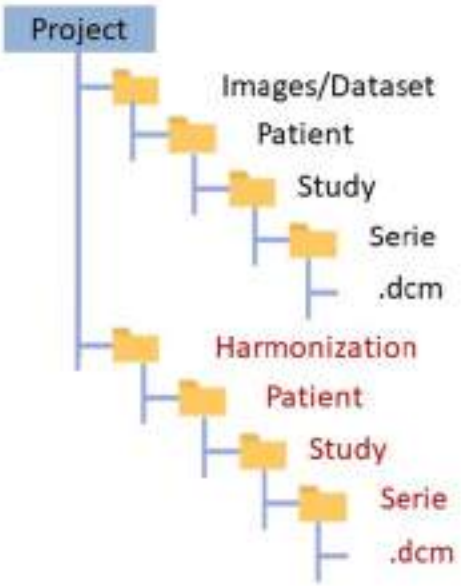
Methodology/performance:
The tool consists in two steps that can be run independently or together:

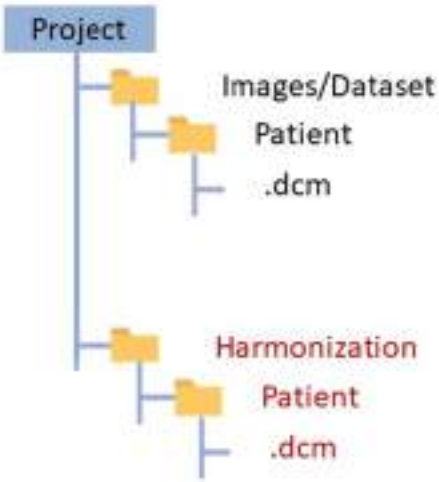
- Denoising: in this stage one of the following filters is applied:
    - Bilateral filter from SimpleITK library
    - Anisotropic Diffusion filter from SimpleITK library
    - Curvature Flow Filter from SimpleITK library
    - SUSAN from FSL library
    - Non-Local Means from DIPY library

- Bias field correction: in this stage N4 bias field correction filter of ANTs is applied.

Use: brief description of the tool's functioning (if it applies).
The tool is dockerized and needs to be mounted with two volumes one with the data and another with the configuration parameters Json file.

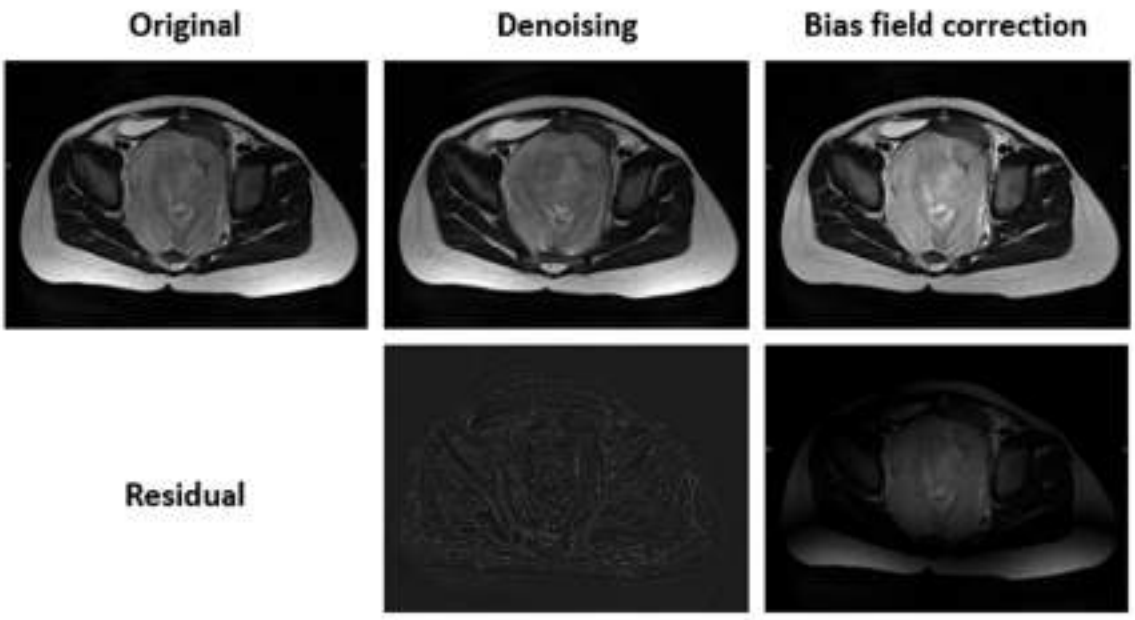
Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).
Input : The inputs of the tool are two volumes: a structured folder with files (/Project) and another folder with the configuration parameter folder named Parameters_configuration.json.
The first volume refers to the folder with data. This tool needs the path to each folder with the DICOM images.
Output : The output will be a replication of this structure up to the first hierarchical level (Images/Dataset) which in this case will be called Harmonization (red structure). Two examples of this are shown below:





Parameters_configuration.json with the name of paths and the configuration of filters

Qualitative results: Provide some visual results (if available) of applying such tools.



Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

- Datasets : This tool has implemented on the PRIMAGE platform.
- Publication link : https://pubmed.ncbi.nlm.nih.gov/34505958/

Keywords for searching in databases:

Denoising; Bias field correction; Image processing; Paediatric tumor, Oncologic imaging, MRI.

DQ14. 2. Technical specifications

Data: In depth description of the data used to train the tool.
Magnetic Resonance T2 weighted (T2W), T1 weighted (T1W), Diffusion weighted imaging (DWI) and Dynamic contrast enhanced (DCE) of neuroblastoma and diffuse intrinsic pontine glioma (DIPG) images from PRIMAGE project.

Methods: in depth description of the methodology used for its development including all data preprocessing.
The analysis can be run with the following command:
docker run -v path\of\Project:/Project -v \path\of\Parameter\folder:/Parameters_config -t harmonization:v01

The first volume refers to the folder with data. This tool needs the path to each folder with the DICOM images. The output will be a replication of this structure up to the first hierarchical level (Images/Dataset) which in this case will be called Harmonization (red structure).

The second volume refers to the parameter_config.json folder. This file contains a list of parameters for running the tools.

Following, an example of parameter_config.json:

```
{
"Paths":[
                    "/Project/Imagenes/Paciente_1/Estudio/PW_10_PERFUSION",
                    "/Project/Imagenes/Paciente_1/Estudio/DW_5_OAxDWI1000b",
                    "/Project/Imagenes/Paciente_1/Estudio/FLAIR_5_CORFLAIR",
                    "/Project/Imagenes/Paciente_1/Estudio/T1W_8_CORT1FSEPROPELLER",
                    "/Project/Imagenes//Estudio/T2W_8_AxialT2FS3",
                    "/Project/Imagenes/Paciente_2/T1W_40008_ORIGSagT1FSPGR"
"Denoising_adf": [{
                              "Conductance":0.5,
                              "Iterations":3,
                              "Time_step":0.0625
                    }
          ],

"N4": [{
                              "BSpline_size":50,
                              "Iterations":[50,30],
                              "Shrink_factor":2
                    }
          ]

}
```

Paths variable is a list of paths with each dicom folder and is a mandatory variable
The other two parameters (**N4** and **Denoising_adf**) are optional, you can apply one or both of them if you put in the Parameters_configuration.json. First the denoising step is applied if it exists and then the inhomogeneity correction if applicable. For the case of the first step you can substitute each of the following filters, each of them with its own parameters:

- Anisotropic Diffusion filter :
    - Conductance
    - Iterations
    - Time_step
- Curvature Flow Filter :
    - Iterations
    - Time_step
- Bilateral Filter:
    - DomainSigma
    - RangeSigma
- SUSAN:
    - Brightness_threshold: times the Otsu threshold
    - FWHM
- Non Local Means:
    - Sigma

    - Patch_radius
    - Block_radius
- N4 bias field correction filter
    - Bspline size
    - Iterations
    - Shrink_factor

The optimised parameter configuration for each tumour and sequence is shown in the table below:

		T1W	**T2W**	**DCE**	**DWI**
NB	**Denoising**	ADF Iterations=2 Conductance=1	ADF Iterations=2 Conductance=1	ADF Iterations=2 Conductance=1	SUSAN FWHM = 2 Threshold= 1.2 Otsu
	Bias Field Correction	N4 (Ants) Iterations =[50,30] BSpline =50 Shrink Factor=2	N4 (Ants) Iterations =[50,30] BSpline =50 Shrink Factor=2	N4 (Ants) Iterations =[50,30] BSpline =50 Shrink Factor=2	-
DIPG	**Denoising**	ADF Iterations=1 Conductance=0.5	Bilateral Domain sigma=0.5 Range sigma=60	Bilateral Domain sigma=0.5 Range sigma=60	SUSAN FWHM = 2 Threshold= 1.5 Otsu
	Bias Field Correction	N4 (Ants) Iterations =50 BSpline =50 Shrink Factor=2	N4 (Ants) Iterations =50 BSpline =50 Shrink Factor=2	N4 (Ants) Iterations =50 BSpline =50 Shrink Factor=2	-

Specific Technical information:

a. CPU

b. Programming language : Python
c. Expected RAM usage : depends on the size of file
d. Running mode : batch dockerr
e. Software version : 1.0.1
f. Libraries : docker, nipype, jsons, SimpleITK, nibabel, dipy, scikit-image
g. Security measures: No root privileges required, just permission to create and delete (nifti) files created in the input path

Traceability and monitoring mechanism: Just some messages printed on the screen

Access restriction: No restriction to the source code or to the binaries of your tool; as long as each time it is used it is referenced (https://pubmed.ncbi.nlm.nih.gov/34505958/).

Additional information for tool integration.
Some of the filters require the existence of niftis, they cannot work directly with DICOM. The tool adds the conversion from DICOM to nifti and from nifti to Dicom.

DQ14.3. Integration validations

Communication channel for the helpdesk: please provide an email address for a contact person who can be reached with any questions regarding your tool.
matias_fernandez@iislafe.es,pedro_mallol@iislafe.es or leonor_cerda@iislafe.es

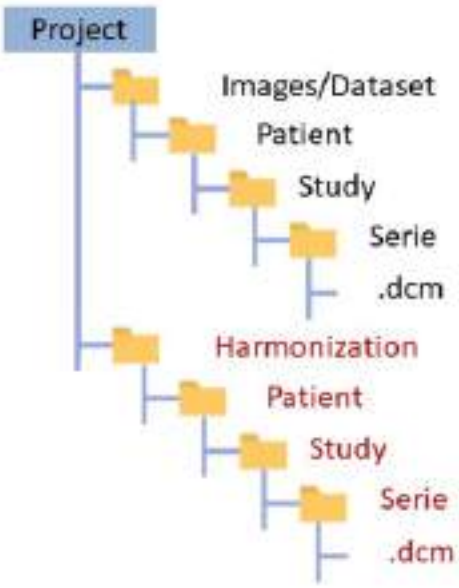
Most common errors

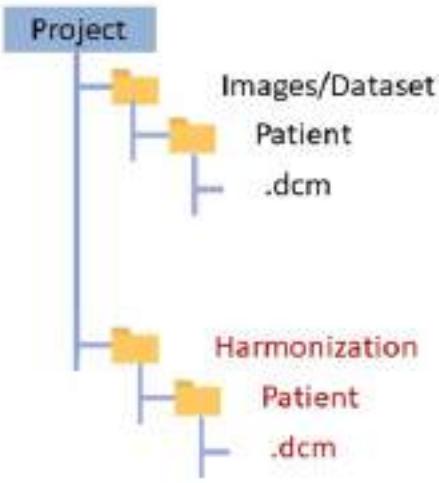
- Wrong path: the image paths in the parameters_configuration.json has to reference to the path in the volume of the docker container.
- Multivolume serie: The tool can give a corrupted DICOM if for other sequences different from the DWI and DCE has more than one volume.
- Incorrect Configuration File Format: Errors may arise if the configuration file is not correctly formatted or if mandatory fields are missing. Ensure the file adheres to the specified JSON includes all necessary parameters.
- DICOM File Corruption: Corrupted DICOM files cannot be processed. These files should be identified and removed or replaced with intact versions.
- Different DICOM series in the same folder: In each DICOM folder added to the variable path of the parameter configuration, there must be only one series.
- Insufficient Permissions: Lack of file or directory access permissions can prevent the tool from reading input data or writing outputs. Ensure the running environment has the necessary permissions.

FAQs

- How is the output structured?

The output is structured in the same way as the input, but in a different folder called Harmonisation as shown:





- Can I run only denoising or bias field correction?

Yes, if you remove the denoising parameters from the json file, only the bias field correction will apply. The same applies to removing bias field correction.

- Can I run the tool in other sequences or regions?

Yes, there are 5 filters that can be run for denoising and parameters to be configured for bias field correction (in case it is necessary) than can be tuned to optimise the results of each sequence.

- If I run in other regions, do I have to change the parameters?

The parameters given are optimised for paediatric brain and abdomen but not for other regions, you can use them but I recommended to change them.

- How do I choose the optimal parameters for other regions?

Here is the methodology we used to optimise the parameters for DIPG and NB (https://pubmed.ncbi.nlm.nih.gov/34505958/).

User Manual

a. Installation/configuration instructions (only for downloadable tools)
   The code along with all the documentation will be available here
   https://bitbucket.org/gibi230/harmonization_tool/src/master/
b. Usage instructions

1. Build docker image inside the code folder.
2. Fill the parameters configuration json with the paths referring to the volume to be mounted.
3. Fill the parameters denoising and/or bias field correction configuration json

Following, an example of parameter_config.json:

```
{
```

```
"Paths":[
                "/Project/Imagenes/Paciente_1/Estudio/PW_10_PERFUSION",
                "/Project/Imagenes/Paciente_1/Estudio/DW_5_OAxDWI1000b",
                "/Project/Imagenes/Paciente_1/Estudio/FLAIR_5_CORFLAIR",
                "/Project/Imagenes/Paciente_1/Estudio/T1W_8_CORT1FSEPROPELLER",
                "/Project/Imagenes//Estudio/T2W_8_AxialT2FS3",
                "/Project/Imagenes/Paciente_2/T1W_40008_ORIGSagT1FSPGR"
"Denoising_adf": [{
                        "Conductance":0.5,
                        "Iterations":3,
                        "Time_step":0.0625
                }
        ],

"N4": [{
                        "BSpline_size":50,
                        "Iterations":[50,30],
                        "Shrink_factor":2
                }
        ]

}
```

4. Execute the docker command:

docker run -v path\of\Project:/Project -v \path\of\Parameter\folder:/Parameters_config denoising_inhomogenity_tool:v01

The first volume refers to the folder with data. The second volume refers to the parameter_config.json folder.

Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.
The only input hierarchy required is

- /Project
    - Images/Dataset or whatever
        - Patient_1 or whatever
            …
            - .dcm

The output structure will be a replication of the input structure changing the Images/Dataset level to the Harmonisation level.

- /Project
    - Harmonization
        - Patient_1 or whatever
            …
            - .dcm

Integration tests: Description of tests for assessing the correct integration of the tool
Just check docker image is built.

DQ15. Deep Learning Noise Reduction

Contributor: FORTH
Area: Prostate MR imaging data curation
Status : Containerized
Purpose : A deep-learning-based noise reduction tool for prostate T2w examinations is used to enhance image quality of noisy images, improve diagnostic accuracy, reduce noise artifacts that are inherent to the MR imaging modality, improve preprocessing execution times and reduce complexity through an end-to-end noise reduction deep model.

DQ15.1. Tool description for its conceptual validation

Tool description: A fully convolutional (with no pooling layers) model was trained on a set of noisy images with the ground truth being the original image without the (synthetic) noise. Different levels of noise and types were incorporated into the training set. The experiments showed reduction in noise levels, but it can impact image quality when T2ws without noise is provided to the model.

Data: Prostate T2w MR images from ProstateX dataset and as an external validation set PI-CAI was used.

Methodology/performance: A protocol for selecting the highest quality of MRI examinations was established to ensure that the best slices are used for model convergence and evaluation. An experienced radiophysicist evaluated all the 346 T2-weighted scans of the ProstateX dataset. As a result, 20 scans (approximately 6% of the dataset) were rejected due to severe noise, motion, and other types of artifacts. Additionally, to mitigate the variation in spacing across the MRI examinations, an aspect ratio preserving reshaping with zero-padding and interpolation was applied to the original scans. This resulted in a pixel array of 384 by 384 pixels across all the slices. Prior to the analysis, the pixel intensities of the MRI slices were normalized. A Gaussian noise pattern was assumed for generating the synthetic noisy slices. Six noisy images for each real slice were generated with noise thresholds spanning from 4% to 14%. Therefore, approximately 38500 noisy slices were used for convergence and evaluation of the examined deep denoising models. Deep learning model reduced noise by 0.10±0.08 peak signal-to-noise ratio (PSNR), overall across different noise thresholds.

Use: brief description of the tool's functioning (if it applies).
DLNR reduces the noise of prostate T2w images via deep learning model. A docker image is used to easily pre-process the noisy images, as depicted in Figure 1.

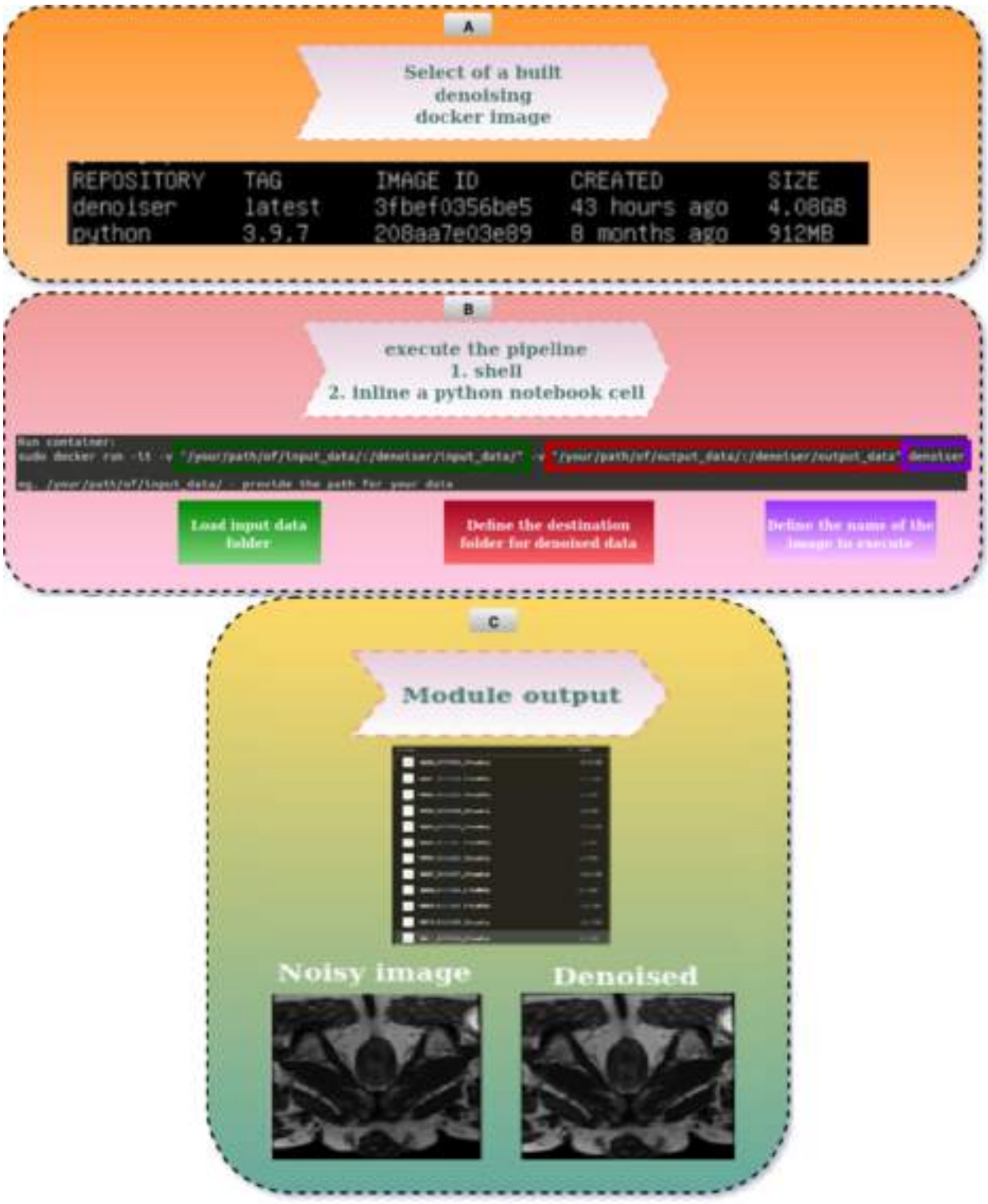


Figure 1. Usage of the DLNR tool

Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).

Input : Nifti (*.nia, *.nii, *.nii.gz, *.hdr, *.img, *.img.gz), nrrd (*.nrrd, *.nhdr), and meta-images (*.mha, *.mhd)
Output : The denoised images are exported while preserving the original meta-data, orientation, spacing, MRI intensities, filetypes, and filenames.

Quantitative results: performance obtained during training of the tool (if applies)
Evaluation metrics: The evaluation of the proposed methodology was conducted exclusively on the unseen testing set as a quantitative evaluation using the juxtaposition of noisy versus denoised images with metrics such as SSIM, and PSNR, as depicted in Figure 2.

Internal validation (ProstateX): The objective of this task was to identify the best performing model architecture out of several deep learning architectures for denoising in terms of image quality improvements. A modified version of the DrCNN did the best in terms of PSNR and SSIM on the unseen testing set. The findings suggest that denoised scans from a deep model have higher image quality than images processed by traditional image processing techniques. DL denoising delivered significant noise reduction with little visible blurring or loss of image quality. In particular, the delta between the PSNR of the noisy scan and the denoised scan shows significant improvements of up to 22% and up to 20% for SSIM compared to the ground truth image.

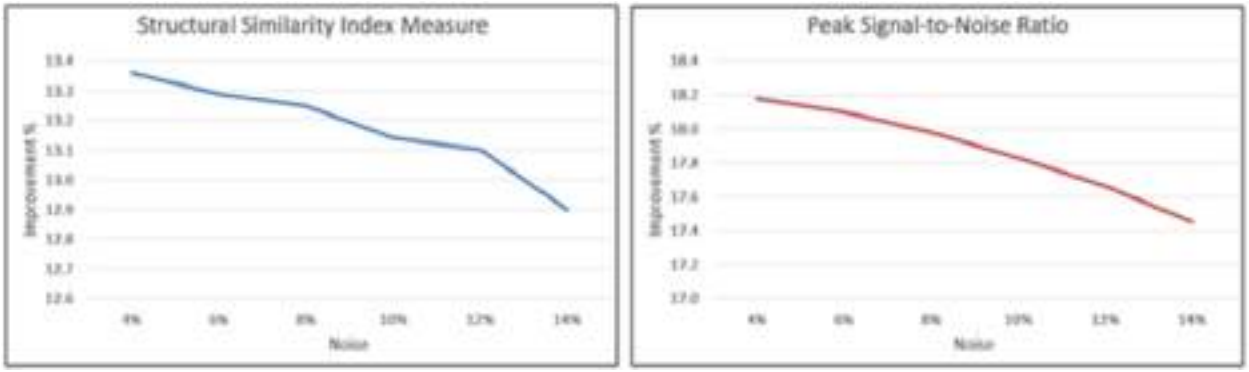


Figure 2. The improvement in image quality of the denoised versus the noisy image in different noise thresholds. The higher the noise the more difficult to denoise the T2w examination.

External validation (PI-CAI): The DL denoising module was tested on the previously unseen PI-CAI dataset to evaluate potential undesirable impacts of the DL model, radiomics stability, and quantify the texture differences. Since synthetic noise was not introduced in the dataset, a SSIM-based metric that estimates the differences between denoised and the original examinations was calculated. An overall difference mean of 0.02±0.04 (4e-4-0.35), which is to be expected because in such a dataset that has been curated for image analysis challenges, only a few examinations were found (qualitatively) to be noisy (less than 30 examinations with more than 0.1 SSIM difference).

Qualitative results: Provide some visual results (if available) of applying such tools.

An expert radiophysicist evaluated qualitatively the quality of the processed T2w. in particular, it was observed that the edges were preserved, and in many cases, enhanced, the original texture distribution was partially restored, noisy pixels were reduced, and pixel intensities were closer to the values of the original scan.Overall, the prostate T2w appears “cleaner” with reduced noisy macroblocking and smoother pixel intensity distribution while preserving texture quality. The model was also tested on the PI-CAI dataset (external testing set) and no artifacts or other distortions were observed after denoising.

Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).
The trained deep models are delivered as Docker containers to ensure that the packaging of the source code, binary model files, and dependencies across the required software (nvidia toolkit, tensorflow, python dependencies, etc.) can be deployed and transferred regardless of the underlying server infrastructure. The denoising module requires as input data the following meta-image formats: nifti (*.nia, *.nii, *.nii.gz, *.hdr, *.img, *.img.gz), nrrd (*.nrrd, *.nhdr), and meta-images (*.mha, *.mhd) with the original MRI intensities. A normalization on an examination-basis is performed prior to the DL denoising. Any examination shape is accepted, since the DL model is fully-convolutional and can be adapted to the input shape. A universal image orientation (RAI) is enforced across the input meta-images to ensure that the same orientation is applied throughout the denoising of the examinations. The denoised images are exported while preserving the original meta-data, orientation, spacing, MRI intensities, and filenames.

Keywords for searching in databases: prostate, deep learning, denoiser, noise reduction, T2w MRI

DQ15.2. Technical specifications

Data: In depth description of the data used to train the tool.
The T2-weighted images of the ProstateX dataset were used to train and evaluate the DL denoising models. The scans were produced using two Siemens 3T MRI scanners, the MAGNETOM Skyra and Trio. T2-weighted images with a resolution of roughly 0.5 mm in plane and a slice thickness of 3.6 mm were obtained using a turbo spin echo procedure. There are available two patient cohorts: a) 203 patients with their clinical data, gland and lesion annotations; and b) 143 with only the multi-parametric MRI available. In the context of this task, all the 346 scans were used since only the imaging data without annotations was required for the convergence of the denoising models.

Methods: in depth description of the methodology used for its development including all data preprocessing.
Data stratification: The examined dataset of 326 was split into three different sets on a patient-basis. The training set consisted of 276 patients, and it was used for fitting the deep learning models. A validation set of 25 patients was used for tuning the parameters of the deep learning models, early-stopping and assessing the status of overfitting during training. Finally, an

unseen testing set of 25 patient scans was utilized for the model evaluation protocol, providing a fair and robust assessment of the denoising effectiveness. The models were trained and evaluated on a slice-basis, yielding approximately: a) 5500 unique slices for the training set, b) 450 slices for the validation set, and c) 470 slices for the testing set.

Data augmentation: In the context of DL analysis, this step is essential for increasing the number of samples that are used during model fitting and also to minimize overfitting of the deep models. Aside from increasing the training sample count, data augmentation results in translation, perspective invariance, and artificially introduced variety in the examined dataset, which strengthens the generalization power of the deep models. Four types of transformations were performed: 1) pixel flipping from right to left, 2) pixel flipping from top to bottom, 3) 90-degree image rotation, and 4) 270-degree image rotation. The final training was comprised of approximately 132000 slices. A sample of augmented data is presented in Figure 3.

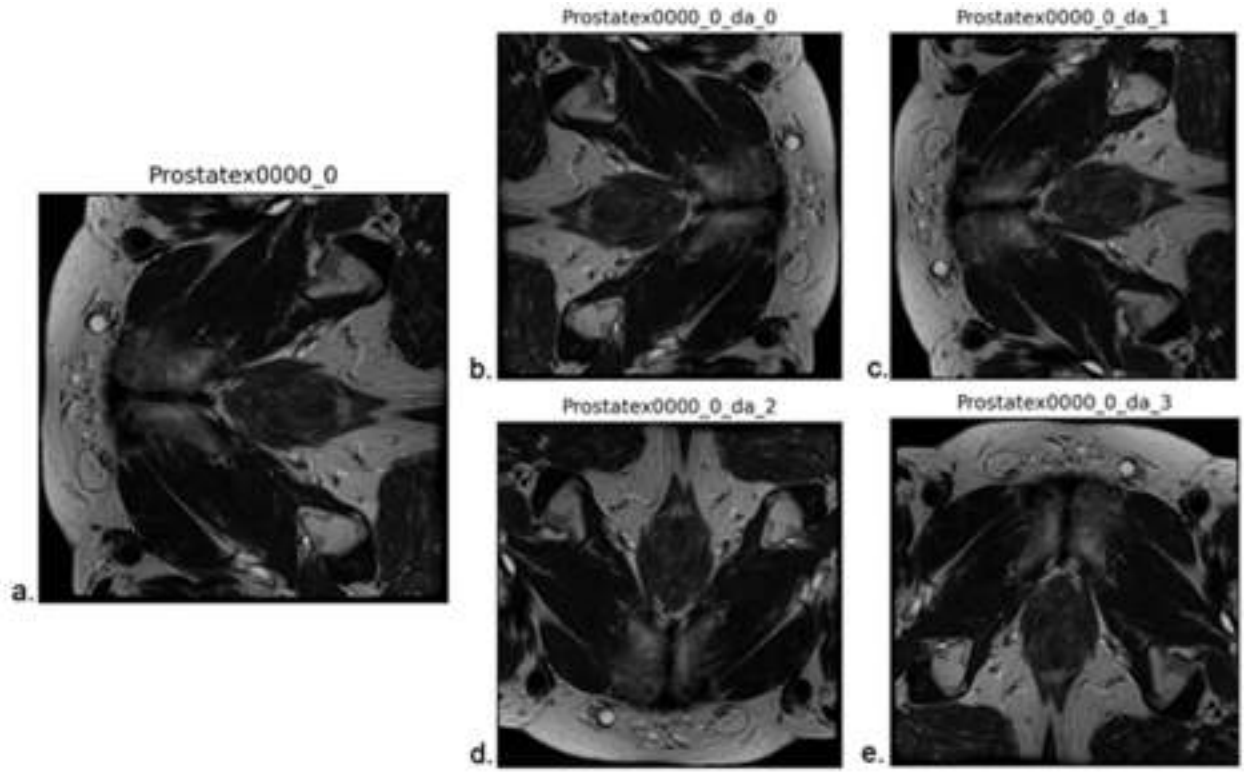


Figure 3. Data augmentation applied to a slice of the training cohort. This includes flipping the original image (a) from right to left (b) and top to bottom (c), rotating 90o (d) and 270o (e).

DL architectures for denoising: Four (4) fully-convolutional architectures were examined: a) a convolutional autoencoder with residual connections (CrAE), b) a denoising convolutional network (DnCNN[1]), c) a denoising convolutional network with residual connections (DrCNN), and d) a real image denoising network (RIDNet[2]). The fully-convolutional models were trained with a supervised learning strategy employing pairs of images; the high-quality ground truth image and the slice with synthetic noise. In most studies, mean squared error (MSE) is used as a loss function, despite the fact that this type of metric does not capture the statistical

distribution of image texture. During hyperparameter optimization, the structural similarity index measure[3] (SSIM) was identified as a better method to formulate the denoising task. SSIM encapsulates three key factors for comparing the aforementioned pair of slices: a) luminance (captures the pixel distortions for brighter regions), b) contrast (captures the pixel distortions of regions with high diversity), and c) structure (a sliding window calculates the statistical local dependencies of texture regions). Therefore, the proposed loss integrates these three factors, and it is formulated as an index that captures the structural differences (SDI) between two images. The adaptive moment estimation (ADAM) was to minimize the proposed SDI loss between the ground truth and the noisy slice. An L1 penalty was applied to the kernels of each layer, constraining the trainable weights of the model from taking outlier values and consequently preventing the model from learning noisy representations that can lead to overfitting. Residual connections were incorporated into the model's architecture to prevent the vanishing gradients[4] effect of the very deep convolutional networks. The integration of the soft-shrinkage activation function[5] was a key integration in the proposed fully-convolutional architecture because it allows the network to learn thresholds that are proportional to the noise power levels of the examined dataset.

Hyperparameter optimization: This process was very important to the success of the denoising task because hyperparameters are the least reported information in the published studies, and their value is dependent on the dataset that is used. Using how well the model performed, the optimization was done on the validation set to find the best model parameters. These parameters are comprised of learnable elements of the architecture (number of modules, kernels, and neurons) as well as other fundamental factors such as the learning rate, optimizer, activation functions, kernel initializers, and regularization penalties. To minimize model overtraining, obtain the most optimal model, and prevent redundant training iterations, early-stopping was implemented with a threshold of 20 epochs after minimizing the validation loss function. Furthermore, comparing the learning curves for loss can reveal information about the fitting state of the deep model. Therefore, to assess the denoising performance and model generalization ability, the learning curves were examined by juxtapositioning the minimum distance between the training and validation loss curves.

Evaluation metrics: The evaluation of the proposed methodology was conducted exclusively on the unseen testing set as a : a) qualitative score by expert radiophysicists or clinicians, and b) quantitative evaluation using the juxtaposition of noisy versus denoised images with metrics such as SSIM, and PSNR

---

[1] Zhang, Kai, et al. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." IEEE transactions on image processing 26.7 (2017): 3142-3155.
[2] Anwar, Saeed, and Nick Barnes. "Real image denoising with feature attention." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
[3] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.
[4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[5] Isogawa, Kenzo, et al. "Deep shrinkage convolutional neural network for adaptive noise reduction." IEEE Signal Processing Letters 25.2 (2017): 224-228.

Specific Technical information:

a. Both CPU and GPU
b. Programming language : python/keras
c. Expected RAM usage : Less than 8GBs for GPUs
d. Running mode : batch-based
e. Software version : v1.2
f. Libraries : docker
g. Security measures: not applicable

DQ15.3. Integration validation

Communication channel for the helpdesk: please provide an email address for a contact person who can be reached with any questions regarding your tool.
Information: trivizakis@ics.forth.gr, office hours, Greece holidays should be considered

Most common errors

a. Provide the correct paths for the input data and output preprocessed images.
b. Check if the input data filetypes are supported.
c. The user must have permissions for executing docker images.
d. The user must have permissions for writing the output data.

FAQs

a. Why is the docker image not running?

Docker runtime has to be installed. For more information check https://docs.docker.com/engine/install/.

b. Why am I getting a permission error?

Either execution permissions have to be granted by your institute's administrator, or use a path that your user has write permissions.

c. Why are the results of the downstream task worse with denoised data than without applying denoising?

Denoising should be applied only on noisy images. High image quality T2w examinations may suffer from quality loss if DLNR is applied.

d. Why does denoising require so much time to finish?

The execution time of DLNR is dependent on the noisy dataset size and the availability of a supported NVidia GPU. Refer to the official NVidia website https://developer.nvidia.com/cuda-gpus.

User Manual

a. Installation/configuration instructions (only for downloadable tools)

No installation is required beside Docker runtime itself (https://docs.docker.com/engine/install/).

b. Usage instructions

Check Figure 1.

c. Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.

The denoising module requires as input data the following meta-image formats: nifti (*.nia, *.nii, *.nii.gz, *.hdr, *.img, *.img.gz), nrrd (*.nrrd, *.nhdr), and meta-images (*.mha, *.mhd) with the original MRI intensities. No other preprocessing is required. Applied only on raw noisy datasets.

Results of non-functional tests : The results provided by DLNR are the difference between the noisy and the processed examination in terms mean PSNR and SSIM.

4- Harmonization tools validation documentation

DH1. *Biologically motivated intensity normalization techniques*

DH1.1 Conceptual description

Contributor: FORTH
Area: harmonization

Tool description: The tool is designed to perform normalization at the image-level. This normalization method aims to reduce the variability in the intensity values of the Magnetic Resonance (MR) prostate images due to different scanners, acquisition protocols and conditions, based on the intensity values of specific tissues. This tool implements three biologically-motivated intensity normalization techniques: (1) The fat-based normalization method, (2) The muscle-based normalization method, and (3) The single tissue (fat or muscle) piece-wise normalization method.

Data: Magnetic Resonance Imaging (MRI) T2weighted (T2W) prostate images.

Methodology/performance: Three pelvis-specific normalization methods were developed to be applied to the MR prostate images. A segmentation method was developed to automatically segment the fat and the muscle tissue in MR pelvic images. Initially, the N4ITK bias field correction method is applied to the image to create images free from artefacts. Then, the central part of the image was cropped to remove the heterogeneous prostate gland and the K-means algorithm (K=2) was applied to identify the muscle and the fat tissue. In the fat-based and the muscle-based normalization method, the mean value and the standard deviation of the voxels that correspond to the fat and the muscle tissue, respectively, were calculated to normalize the whole image according to statistics derived from the tissues' distribution. In the single tissue (fat or muscle) piece-wise normalization method, the distribution of the fat or the muscle tissue was used as input to the histogram matching algorithm proposed by Nyul and Udupa in order to

extract landmarks and learn the standard scale based on the values of the reference single tissue.

Use: The tool can be used for harmonizing the intensity values of MR T2W prostate images using as reference a representative biological tissue of this anatomy (fat or muscle tissue). It aims to bring into the same scale the distribution of the same tissue type in different images. Thus, it can be used as a preprocessing step to harmonize MR T2W prostate images.

Input/Output formats: The input of the tool is a path with the raw unnormalized MR T2W axial prostate images, which should be in DICOM format.

Only for the **piece-wise** normalization techniques, the input path should contain 2 subfolders, i.e. **train** and **test** folder. The **train** folder contains the folders of the patients that are used for the training of the algorithm to learn the standard scale, while the **test** folder contains the folders of the patients whose images will be normalized. If a train folder does not exist, a pretrained standard scale is used to normalize the images of the test folder.

The output is the normalized images, which are saved to a path specified by the user with the same format (DICOM).

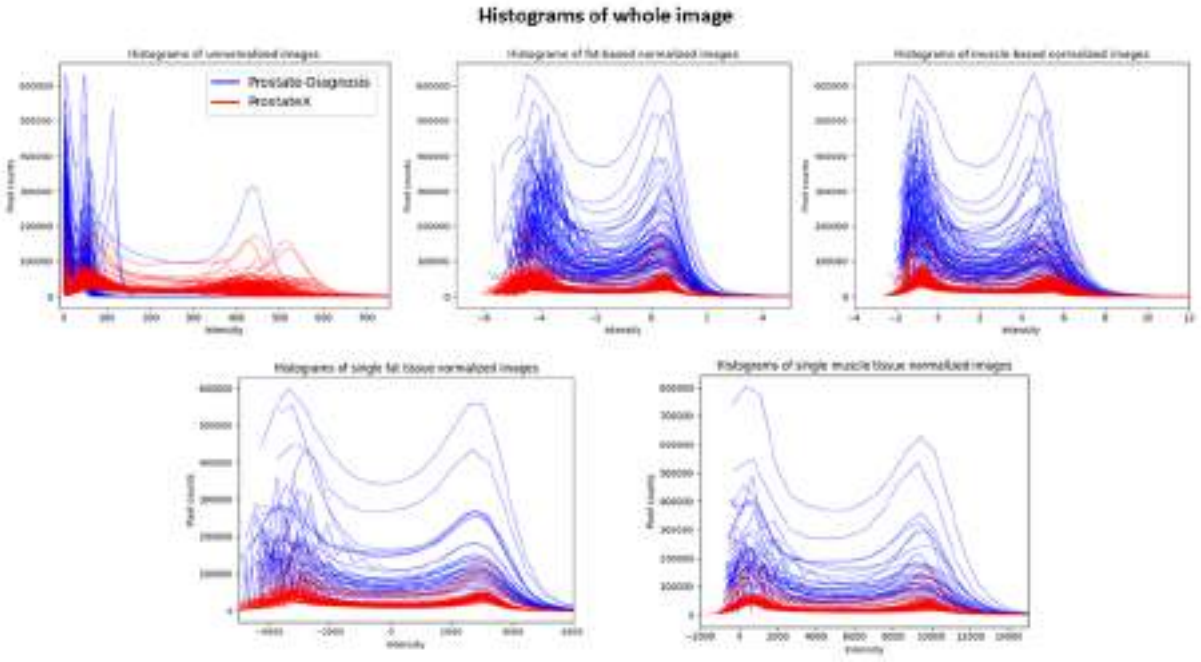
Quantitative results:
Two publicly available datasets (Prostate-Diagnosis and ProstateX) were used to evaluate the performance of the normalization methods of the tool. The measurements of the std of NMI show that the variation of the intensity values of the fat tissue and the muscle tissue is reduced after applying the normalization methods.

	STD OF NMI			
	Prostate-Diagnosis (1.5T)		**ProstateX (3T)**	
Normalization method	**Fat tissue**	**Muscle tissue**	**Fat tissue**	**Muscle tissue**
Unnormalized	0.092	0.055	0.09770	0.071
Muscle-based	0.063	$2.26e^{-16}$	0.068	$1.81e^{-16}$
Fat-based	$5.84e^{-16}$	0.066	$4.32e^{-16}$	0.073
Fat tissue piece-wise	0.00067	0.037	0.00049	0.042

Muscle tissue piece-wise	0.083	0.0085	0.09768	0.0099

Qualitative results:

The histograms of the images before and after the normalization methods were calculated for each dataset.



DH1.2 Technical description

Data: Magnetic Resonance Imaging (MRI) T2weighted (T2W) prostate images. Two publicly available datasets were used to train and validate the tools. The PROSTATE-DIAGNOSIS dataset contains 89 prostate cancer T2W MR images acquired on a 1.5T Philips Achieva scanner using combined surface and endorectal coil. The PROSTATEx dataset contains 204 prostate T2W MR images that were acquired on two different types of Siemens 3T MR scanners with surface coil, the MAGNETOM Trio and Skyra.

Methods: In order to apply the pelvis specific normalization methods, an approximation of the fat and the muscle tissue should be identified. Thus, an automated segmentation technique is developed to delineate fat and muscle tissues within MR pelvic images. Initially, the N4 bias field correction method is applied to generate bias-free images by eliminating artifacts. Subsequently, each image undergoes cropping, where the central 20% of columns are removed, excluding the heterogeneous prostate gland. The K-means algorithm with a value of K set to 2 is applied to identify two clusters: one representing low-intensity values (approximating muscle tissue) and the other including high-intensity values (approximating fat tissue). For precise muscle tissue segmentation, the 12th percentile of the distribution is computed to eliminate the lowest 12% of

values corresponding to background pixels like air and vessels. For the fat- and the muscle-based normalization technique, the mean value and the standard deviation of the fat and the muscle tissue, respectively, are calculated to standardize the image intensity values. In the single tissue piece-wise normalization method, the intensity values corresponding to the approximations of either fat or muscle tissue are used as input to the histogram normalization algorithm to learn and extract the standard scale.

Specific Technical information: CPU/GPU: CP

Programming language: python 3.8.13
Expected RAM usage: depending on the amount of input data (at least 8GB)
Running mode (interactive/batch-based/case-based…): batch-based docker
Software version: 1.6
Libraries: dockerized
Security measures: Not applicable

Traceability and **monitoring** mechanism: Informative messages are displayed on screen when running the docker. If an error occurs, the corresponding error message will be displayed to inform the user what the source of the error was. Corresponding messages for the start and the completion of the tool's execution are also displayed on screen.

Unitary tests: Tests were performed in order to ensure the proper functionality of the tool. All possible functionalities were tested, i.e. applying the fat-based normalization technique, applying the muscle-based normalization technique, applying the fat piece-wise normalization technique, and applying the muscle piece-wise normalization technique.

Additional information for tool integration: requires docker

DH1.3 Integration description

Usage instructions

The tool is dockerized.

1. Download the latest version of the docker image (image_batch_bio_intensity_norm_v1.6.tar)
2. Load the docker image by running the following command (assuming you are in the same directory as the tar file):

   udocker load -i image_batch_bio_intensity_norm_v1.6.tar

3. Run the following command in order to create a container and instantiate the docker image:

   udocker run --rm

   -v "your_input_path:/home/chaimeleon/datasets"

-v "your_output_path:/home/chaimeleon/persistent-home"

image_batch_bio_intensity_norm:1.6

[-h] [-f] [-m] [-p]

where *your_input_path* is the path that contains the folders with the original image of each patient

your_output_path is the path where the normalized image of each patient will be saved

available arguments:

-h, --help: show this help message and exit

-f, --fat-based: If specified, the fat-based normalization algorithm is applied

-m, --muscle-based: If specified, the muscle-based normalization algorithm is applied

-p, --piece-wise: If specified, the single tissue piece-wise normalization algorithm is applied, Default = fat tissue piece-wise

For instance, if the user wants to apply the single muscle tissue piece-wise normalization method, the arguments -p -m should be specified.

If the user specifies only the argument -p, then the fat tissue piece-wise normalization method will be applied by default.

Additional considerations : Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used

Input/output description: The input of the tool is a path with the dataset. The dataset should contain raw unnormalized images, which should be in DICOM format. In each dataset directory, an index.json file should be contained in order to walk through the contents of the dataset. Only for the piece-wise normalization techniques, the input path should contain 2 folders, i.e. train and test folder. The train folder contains the dataset that is used for the training of the algorithm to learn the standard scale, while the test folder contains the dataset whose images will be normalized. If a train folder does not exist, a pretrained standard scale is used to normalize the images of the test folder.

Examples of the required format of the input path:

For the fat- and muscle- based normalization techniques:

-- patient_level (folder)

-- study_level (folder)

-- series level (folder)

-- DICOM files (files)

…

-- index.json (file)

For the piece-wise (fat or muscle) normalization techniques:

-- train (folder)

-- patient_level (folder)

-- study_level (folder)

-- series level (folder)

-- DICOM files (files)

…

-- index.json (file)

-- test (folder)

-- patient_level (folder)

-- study_level (folder)

-- series level (folder)

-- DICOM files (files)

…

-- index.json (file)

The output is the normalized images, which are saved to a path specified by the user with the same format. For each patient, a folder will be created and the normalized image in DICOM format will be stored within the folder.

No preprocessing is needed. In the tool, the N4ITK bias field correction method is applied to the image to create images free from artefacts and identify the fat or the muscle tissue, before

applying each one of the available intensity normalization techniques.

Mandatory/optional data: The original T2W MR image for each patient is required to run the tool.

Cases in which the tool should not be used: The tool should not be used in any organ/anatomy other than prostate. Furthermore, the tool should not be used in any sequence other than T2W. The tool is developed only for T2W MR prostate images.

- **Integration tests**: Tests were performed in order to ensure the proper integration of the tool in the platform.
- **Results of non-functional tests:** Not applicable
- **Common errors**

The most common errors are:

- No functionality selected (available functionalities: -f, -m, -p)
- Empty directory
- Not valid input format
- Lack of slice location information on the DICOM header
- Invalid image dimensions (the image should be represented by a 3D array of shape [SLICES, WIDTH, HEIGHT])
- Required directory does not exist

FAQs

Q: What is the purpose of the tool?

A: The tool can be used as an image pre-processing step to harmonize Magnetic Resonance (MR) T2W prostate images. The normalization methods aim to reduce the variability in the intensity values of the MR prostate images due to different scanners, acquisition protocols and conditions, based on the intensity values of specific tissues. This tool harmonizes the intensity values of MR T2W prostate images using as reference a representative biological tissue (fat or muscle tissue) around the prostate gland. It aims to bring into the same scale the distribution of the same tissue type in different images.

Q: Which are the functionalities of the tool?

A: The tool offers three biologically motivated intensity normalization techniques:

a. Fat-based normalization method: the image is normalized using the mean and standard deviation of the periprostatic fat distribution.

b. Muscle-based normalization method: the image is normalized using the mean and standard deviation of the muscle distribution.

c. Single tissue (fat or muscle) piece-wise normalization method: the image is normalized using a standard scale based on the fat/muscle distribution.

Q: In which data can the tool be applied?

A: The tool can be applied only for Magnetic Resonance (MR) T2weighted (T2W) axial prostate images.

Q: Are there any specific requirements for the execution of the tool?

A: The tool can run in any operating system, as it is containerized in order to ensure compatibility.

Contact person for the helpdesk: dovrou@ics.forth.gr , nikiforakik@gmail.com

DH2. Image intensity harmonization

DH2.1 Conceptual description

Tool description: The tool is effective in the harmonization of intensity dynamic ranges in MRI and is backed by AI, this model is developed for Prostate T2W images. The variability in intensity that results from different acquisition protocols and scanners difficult the performance of AI tools and global information computation. The methodology can be applied to MRI samples and it's based on self-supervised learning that leverages the use of MRI frequency domain to synthetically generate contrast variations in reference images by making subtle changes in specific frequencies and then transforming the image back to the spatial domain where the images have its original content with altered intensities. These synthetically generated paired images are finally used to train an autoencoder based model with harmonization purposes on prostate MRI.

Data: Magnetic Resonance Imaging (MRI) T2weighted (T2W) prostate images.

Methodology/performance: A self-supervised learning approach was followed; therefore, the training data was created from the original dataset. For this purpose, alterations of the original images were performed using the MRI frequency domain simulating images acquired with different acquisition protocols. These alterations were generated following the process: The image was transformed using a technique called Fast Fourier Transformation, which gave us both the phase and magnitude information for each version of the image. Then, we adjusted the transformed image so that its values ranged between 0 and 1. We created a special guide, called a mask, to control the changes we wanted to make to the image. This mask changed in size and strength with each step of the process. It focused mainly on the central area of the image, which represents the lower frequencies. We combined the information from the previous steps to create a final mask that emphasized changes in the central part of the image. This final mask was adjusted by multiplying it with a value chosen within a specific range, which determined the intensity of the changes we wanted to make. Using the modified mask, we made additions and subtractions to the phase and magnitude of the image in the frequency domain. After each adjustment, we converted the image back to its original form using the Inverse Fourier Transformation. By following these detailed steps, empirically designed, the original image underwent a series of transformations resulting in an altered version with adjusted contrast. The full range of transformations were applied to each reference image, generating different representations of the original image.

Use: The tool can be used for harmonizing the intensity values of MR T2W prostate images, and can be used as a preprocessing step.

Input/Output formats: The only requirement for input format is a path to a DICOM directory of original T2 prostate samples. The output is the resulting DICOM directory saved in a chosen path.

Quantitative results: For the validation of the proposed harmonization solution, two different datasets were used. First, the dataset presented in[7] was used. It consisted of 120 T2w prostate MRI, collected at 7 different hospitals, comprising a total of 10 MRI scanners from 3 different manufacturers (Siemens, GE, and Philips). Manual delineation of the prostate gland by two experienced radiologists was available. Three different areas were delineated independently: CZ-TZ, PZ and SV.

The second validation dataset consisted of a subset of 392 prostate MRI studies from ProstateNet[8]. These cases were collected from 6 different hospitals, acquired with scanners from three different manufacturers (Siemens, GE, and Philips) and magnetic field strength (1.5T and 3T). The dataset included 55 MRI scans acquired with ERC. Prostate gland segmentation masks, including the same regions as the previous dataset, were available in the dataset.

Then, a quantitative assessment was conducted through three different experiments to compare the performance of an AI-based prostate segmentation solution. The three experiments were conducted as follows:

1. The CNN-based automatic segmentation solution presented in[33] was used to compare its performance over the test dataset (n=120) used in the same work, when using original or harmonized images as input. The algorithm was used to automatically segment the prostate gland in three different regions: CZ-TZ, PZ, and SV.
2. The CNN-based automatic segmentation solution presented in [25] was used to segment the prostate gland on a dataset of prostate cases acquired with ERC (n=55). The same anatomical areas as in the previous experiment were segmented using the original and harmonized images as input to the trained model.
3. Two completely new CNN-based segmentation solutions were trained using, on one hand, original images as inputs, and, on the other hand, the corresponding harmonized dataset. For this purpose, the same architecture as the one proposed in[33] was used to train the model. The ProstateNet dataset[34], excluding those cases with ERC (n=282 cases) was used to train the segmentation model, while the testing dataset presented in[33] (n=120) was used for testing. Both networks were trained during 200 epochs using Adam optimizer with a learning rate of 0.0001 and a multilabel DSC loss function.

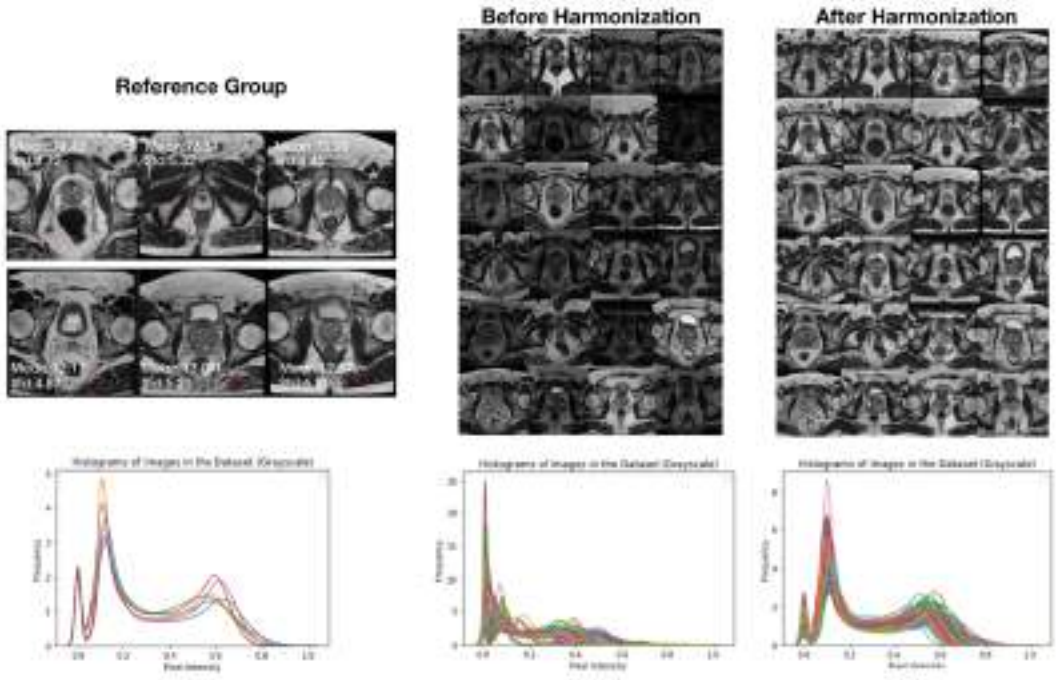
In all the experiments, the performance was compared by means of the DSC which was calculated on each region independently and in the whole PG, corresponding to both CZ-TZ and PZ zones.

	CZ-TZ	p	PZ	p	SV	p	PG	p

[7] Jimenez-Pastor A, et al. 2023. Automated prostate multi-regional segmentation in magnetic resonance using fully convolutional neural networks. Eur Radiol. 2023 Jul;33(7):5087-5096. doi: 10.1007/s00330-023-09410-9.
[8] Kondylakis, Haridimos & Sfakianakis, Stelios & Kalokyri, Varvara & Tachos, Nikolaos & Fotiadis, Dimitrios & Marias, Kostas & Tsiknakis, Manolis. (2022). Data Ingestion for AI in Prostate Cancer. 10.3233/SHTI220446.

Test #1								
Original Samples	0.842±0.08	0.23	0.698±0.138	0.86	0.704±0.139	0.90	0.863±0.05	0.18
Harmonized Samples	0.840±0.07		0.695 ± 0.128		0.704±0.135		0.858±0.05	
Test #2								
Original images	0.671±0.22	<0.01*	0.605±0.19	0.08	0.683±0.24	0.77	0.708±0.21	<0.01*
Harmonized images	0.823±0.13		0.667±0.13		0.763±0.10		0.831±0.09	
Test #3								
Original images	0.831±0.08	<0.01*	0.655±0.14	<0.01*	0.662±0.17	<0.01*	0.854±0.07	0.02*
Harmonized images	0.861±0.06		0.721±0.11		0.711±0.15		0.874±0.04	

Qualitative results: The histograms of the images before and after the normalization methods were calculated for each dataset, as well as the reference group from which the alterations were obtained in the training phase.



Data: Magnetic Resonance Imaging (MRI) T2weighted (T2W) prostate images. For the validation of the proposed harmonization solution, two different datasets were used. The first one consisted of 120 T2w prostate MRI, collected at 7 different hospitals, comprising a total of 10 MRI scanners from 3 different manufacturers (Siemens, GE, and Philips). Manual delineation of the prostate gland by two experienced radiologists was available. Three different areas were delineated independently: CZ-TZ, PZ and SV.

The second validation dataset consisted of a subset of 392 prostate MRI studies from ProstateNet. These cases were collected from 6 different hospitals, acquired with scanners from three different manufacturers (Siemens, GE, and Philips) and magnetic field strength (1.5T and 3T). The dataset included 55 MRI scans acquired with ERC. Prostate gland segmentation masks, including the same regions as the previous dataset, were available in the dataset.

Both datasets underwent segmentation tasks comparison between original and harmonized samples.

Methods: A subset of images with similar acquisition parameters was selected as reference cases based on signal quality and absence of artifacts. A self-supervised learning approach was used, creating training data by altering original images in the frequency domain to simulate images acquired with different protocols. The process involved transforming images into the frequency domain, normalizing them, generating a mask for alterations, and applying operations to the frequency domain image. The intention was to regulate the weight of low frequencies and create a variety of combinations. The final alteration mask was created by adding the normalized module in frequency domain with the low-frequencies mask together and normalizing them. This mask was then multiplied by a factor to intensify alterations specifically in the middle region (low frequencies), and operations were applied to the image in the frequency domain before reconstructing it back to the imaging domain.

Specific Technical information:

- CPU/GPU: CPU
- Programming language: python 3.7
- Expected RAM usage: depending on the amount of input data (at least 8GB)
- Running mode (interactive/batch-based/case-based…): case-based docker
- Software version:
- Libraries: dockerized
- Security measures: Not applicable

Traceability and **monitoring** mechanism: Informative messages are displayed on screen when running the docker in interactive mode (specifying -it). If an error occurs, the corresponding error message will be displayed to inform the user what the source of the error was. Corresponding messages for the start and the completion of the tool's execution are also displayed on screen.

Unitary tests: Tests were performed in order to ensure the proper functionality of the tool. All possible functionalities were tested, i.e. applying the fat-based normalization technique, applying the muscle-based normalization technique, applying the fat piece-wise normalization technique, and applying the muscle piecewise normalization technique.

Additional information for tool integration: requires Docker

DH2.3 Integration description

User Manual:

Usage instructions

The tool is dockerized.

1. The image is loaded and ready to execute using the jobman system.
2. Run the image by running the following command:

    !jobman submit -i mri_harmonization:latest -r ‘size’-gpu –
    /*your_input_path*/*your_output_path*/ *modality*

    where:

    size is the desired GPU capability (small, large).

    your_input_path is the path to a DICOM directory.

    your_output_path is the path where the harmonized image will be saved

    mode is the image modality you want to harmonize:

    - Prostate: prostate t2w.
    - Breast: breast t2w.
    - Rectum: rectum t2w.

Input/output description: The expected input of the tool is the path to a DICOM directory where a dataset image series of the organ/image modality executed is present. The tool will take the output path and make the pertinent directories and sub-directories where the harmonized DICOM series will be saved.

Examples of the required format of the input path:

-- patient_level (folder)

-- study_level (folder)

-- series level (folder)

-- DICOM files (files)

No preprocessing is required, the tool expects to receive original raw DICOM files as input.

Cases in which the tool should not be used: The tool should not be used in any organ/anatomy other than the modality specified in the command.

- **Integration tests**: Tests were performed in order to ensure the proper integration of the tool in the platform.
- **Results of non-functional tests:** Not applicable
- **Common errors**

The most common errors are:

- Typo error in command modality: ‘prostate’, ‘breast’, ‘rectum’.

FAQs

Q: What is the purpose of the tool?

A: The tool is used as image intensity harmonization for a MRI dataset of the three modalities included (prostate t2w, breast t2w, rectum t2w), the tool returns the exact same image with altered intensities which look clearer and brighter, and share the same tissue intensity balance characteristics among each harmonized image, which help reduce variability in image representations among different scanners and t2w protocols.

Q: Which are the functionalities of the tool?

A: The tool has been trained to bring any intensity representation image of the three offered modalities to a common and desirable intensity range.

Q: In which data can the tool be applied?

A: The tool can be applied only for Magnetic Resonance (MR) T2w prostate, T2w breast and T2w rectum, to be specified as modality input in the command.

Q: Are there any specific requirements for the execution of the tool?

A: The tool can run in any operating system, as it is containerized in order to ensure compatibility.

Contact person for the helpdesk:
For any inquiries: eduardoibor@quibim.com

DH3. Feature-based harmonization

DH3.1 Conceptual description

Tool description: The tool is designed to perform harmonization at the feature-level. Feature-based harmonization method aims to reduce the variability in the radiomics features due to different scanners, acquisition protocols and conditions by using empirical Bayesian methods to estimate differences in radiomics values and then expressing them in a common space (location/scale adjustment). The tool offers two methods: (1) ComBat method, which shifts the radiomics features to the overall mean and pooled variance of all centers, and (2) M-ComBat method, which shifts the radiomics features to the mean and variance of the chosen reference center with the most samples.

Data: Numeric variables (e.g. radiomic features)

Methodology/performance: The feature-based harmonization tool uses the Combining Batches method (ComBat), to adjust the data and express them in a common space. The input arguments required for harmonization are the multicenter radiomics with a sufficient size and the covariates, which express the ‘center-effect’. One covariate is chosen as a source of variation and the ComBat tries to shift the radiomic features to the overall mean and pooled variance of all the centers. Furthermore, the tool offers the option to apply the M-ComBat method, in which the radiomic features are shifted to the mean and variance of the chosen reference center (i.e. the center which provides the most data in absolute numbers) of the selected covariate. To evaluate the performance of the tool, boxplots and histograms of an example radiomic feature were calculated before and after applying the ComBat and the M-ComBat method. Furthermore, the “DiffFeatureRatio” introduced by Li Y et al. was calculated to quantify the performance of ComBat. This metric is the ratio of radiomic features with p-value < 0.05 to the overall radiomic features, representing the ratio of features that have significantly different feature distributions among different center-effects.

Use: This tool can be used for harmonizing radiomic features. It can be used as a preprocessing step to harmonize radiomic features.

Input/output formats: The input of the tool is a path that should contain two separate folders. The first folder must be named “radiomics” and contains a csv file with the radiomic features of each patient. The csv file should have a column named "PatientID" in which each row has the patient ID and the radiomics names columns in which each row contains the radiomic value per patient ID. The second folder must be named “metadata” and contains a csv file with the corresponding metadata of each patient. This csv file should have a column named "PatientID" in which each row has the patient ID, a column named “Manufacturer” or/and “ManufacturerModelName” in which each row contains the manufacturer’s name or/and the manufacturer’s model name that each patientID was scanned at.

Examples of the required format of the input path:

-- **radiomics** (folder)
  -- radiomics.csv (a csv file)

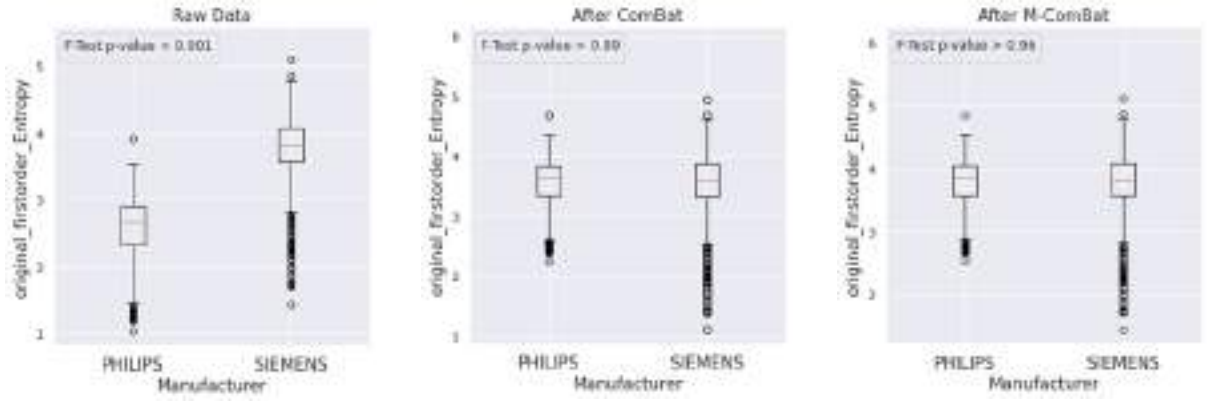
-- **metadata** (folder)
  -- metadata.csv (a csv file)

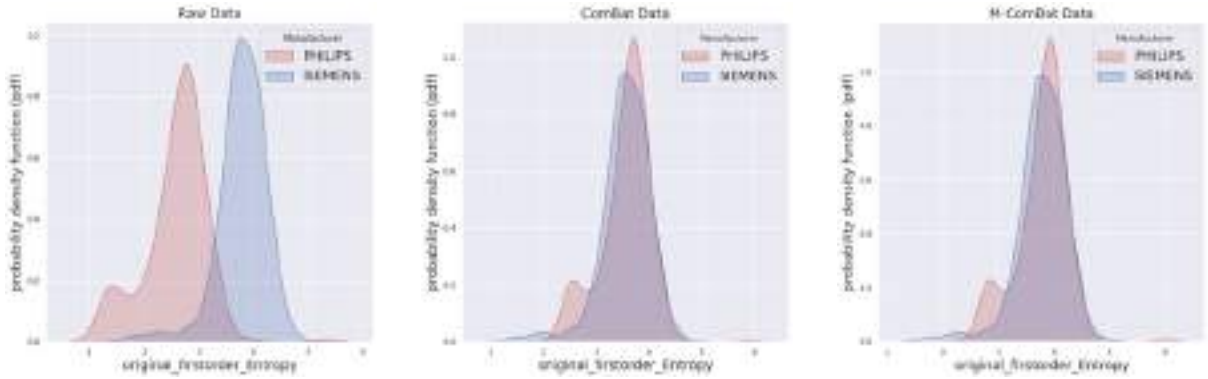
The output is the harmonized radiomic features for all patients, which are saved to a path specified by the user with the same format (.csv file). Also, harmonization estimates and other information are stored in additional pickle files.

Quantitative results: The publicly available PICAI dataset was used to evaluate the performance of the feature-based harmonization methods of this tool. The values of the "DiffFeatureRatio" show that the raw features without harmonization had a significantly different feature distribution. In contrast, the "DiffFeatureRatio" values of the harmonized data always tend to zero, which means that most radiomics features could not be detected to have significantly different feature distributions among the different scanner settings.

	DiffFeatureRatio		
Center-Effect	**Raw**	**ComBat**	**M-ComBat**
Manufacturer	0.93	0.00	0.00
Manufacturer model	0.91	0.00	0.00

Qualitative results: The boxplots and the histograms of the "1st order entropy" radiomic feature are calculated before and after ComBat and M-ComBat, showing that the distribution of the values of the radiomic feature are more similar after applying the harmonization. For the harmonization process, the manufacturer type (e.g. Philips or Siemens) is used as center-effect.





Data: Radiomic features extracted from MR T2W prostate images. The publicly available PI-CAI dataset was used to validate the tool. The PI-CAI dataset contains 1500 MRI prostate cases from two different scanners (Siemens and Philips) and seven different scanner models.

Methods: The feature-based harmonization tool is based on the Combining Batches method (ComBat), in which empirical Bayesian methods are used to estimate differences in values. These estimates are then used to adjust the data, expressing them in a common space. Thus, the input arguments required for harmonization are the multicenter radiomics with a sufficient size and the covariates, which are center-related parameters and express the 'center-effect'. These covariates refer to the manufacturers (e.g. Philips, Siemens etc.) or the manufacturer model (e.g. Skyra, TrioTim etc.) that are used from each center. One covariate is chosen as a source of variation and the ComBat tries to shift the radiomic features to the overall mean and pooled variance of all the centers. Furthermore, the tool offers the option to apply the M-ComBat method, in which the radiomic features are shifted to the mean and variance of the chosen reference center (i.e. the center which provides the most data in absolute numbers) of the selected covariate. Hence, the user can select the method (ComBat or M-ComBat) and the covariate (manufacturer or manufacturer model name).

Specific Technical information:

- CPU/GPU: CPU
- Programming language: python 3.8.13
- Expected RAM usage: depending on the amount of input data (less than 8GB)
- Running mode (interactive/batch-based/case-based…): batch-based docker
- Software version: 1.4
- Libraries: dockerized
- Security measures: Not applicable

Traceability and **monitoring** mechanism: Informative messages are displayed on screen when running the docker. If an error occurs, the corresponding error message will be displayed to inform the user what the source of the error was. Corresponding messages for the start and the completion of the tool's execution are also displayed on screen.

Unitary tests: Tests were performed in order to ensure the proper functionality of the tool. All possible functionalities were tested, i.e.applying the ComBat and the M-ComBat method using the manufacturer or the manufacturer model name as ‘center-effect’.

Additional information for tool integration: requires docker

DH3.3 Integration description

User Manual:

Usage instructions

The tool is dockerized.

1. Download the latest version of the docker image (harmonization_v1.4.tar)
2. Load the docker image by running the following command (assuming you are in the same directory as the tar file):

   udocker load -i harmonization_v1.4.tar

3. Run the following command in order to create a container and instantiate the docker image:

   udocker run --rm

   -v "your_input_path:/home/chaimeleon/datasets"

   -v "your_output_path/:/home/chaimeleon/persistent-home"

   harmonization:1.4

   [-h] [-m] [-M] [-c]

where *your_input_path* is the path that contains the folders with the radiomics and the metadata csv files.

your_output_path is the path where the harmonized radiomic features and the harmonization parameters files are stored.

available arguments:

-h, --help: show this help message and exit

-m, --manufacturer: If specified, the manufacturer variable will be used as center-effect.

-M, --manufacturerModelName: If specified, the manufacturer model variable will be used as center-effect.

-c, --combat: If specified, the ComBat method will be used. If not, the M-ComBat method will be used.

Additional considerations: Input/output description, if any preprocessing is needed, mandatory/optional data, cases in which the tool should not be used.

Input/output description: The input of the tool is a path that should contain two separate folders. The first folder must be named “radiomics” and contains a csv file with the radiomic features of each patient. The csv file should have a column named "PatientID" in which each row has the patient ID and the radiomics names columns in which each row contains the radiomic value per patient ID. The second folder must be named “metadata” and contains a csv file with the corresponding metadata of each patient. This csv file should have a column named "PatientID" in which each row has the patient ID, a column named “Manufacturer” or/and “ManufacturerModelName” in which each row contains the manufacturer’s name or/and the manufacturer’s model name that each patientID was scanned at.

Examples of the required format of the input path:

-- radiomics (folder)

    -- radiomics.csv (a csv file)

-- metadata (folder)

    -- metadata.csv (a csv file)

The output is the harmonized radiomic features for all patients, which are saved to a path specified by the user with the same format (.csv file). Also, harmonization estimates and other information are stored in additional pickle files.

No preprocessing is needed.

Mandatory/optional data: The csv files with the numeric features and the corresponding metadata (covariates), respectively, are mandatory to run the tool.

Cases in which the tool should not be used: Not applicable.

- **Integration tests**: Tests were performed in order to ensure the proper integration of the tool in the platform.
- **Results of non-functional tests:** Not Applicable
- **Common errors**

The most common errors are:

- No functionality selected (available functionalities: -m, -M, -c)
- Empty directory
- Required directory does not exist

FAQs

Q: What is the purpose of the tool?

A: The tool can be used as a pre-processing step to harmonize numeric features, such as radiomic features. This feature-based harmonization method aims to reduce the variability in the radiomic features due to different scanners, acquisition protocols and conditions by using empirical Bayesian methods to estimate differences in radiomics values and then expressing them in a common space (location/scale adjustment).

Q: Which are the functionalities of the tool?

A: The tool aims to harmonize multicenter numeric features with a sufficient size based on the covariates, which express the source of variation and the 'center-effect'. The tool offers two methods:

a. Combining Batches (ComBat) method, which shifts the radiomics features to the overall mean and pooled variance of all centers.

b. M-ComBat method, which shifts the radiomics features to the mean and variance of the chosen reference center with the most samples.

Q: In which data can the tool be applied?

A: The tool can be applied to numeric features originated from any anatomy. The values of the chosen covariate should also be provided.

Q: Are there any specific requirements for the execution of the tool?

A: The tool can run in any operating system, as it is containerized in order to ensure compatibility.

Contact person for the helpdesk: gmanikis@gmail.com , dovrou@ics.forth.gr

DH4. *Trace4Harmonization*

DH4.1 Conceptual description

Tool description: Trace4Harmonization™ is a tool aimed at harmonizing numerical features, including (but not limited to) potential imaging biomarkers such as features extracted from medical images that were acquired under different conditions (e.g. different acquisition system or different acquisition protocols).

More specifically, the aim is twofold: the first is related to calibrating a harmonization model based on a dataset of unharmonized samples; the second is the application of the calibrated model to new samples.

Data: Numeric variables (e.g., radiomic features extracted from medical images, with no limitation on the data modality from which the features are extracted)

Methodology/performance: Trace4Harmonization™ is based on ComBat (Combining Batches method), to harmonize features (numerical variables) in a common space. In order to be used, these techniques need a sufficient sample size as well as the corresponding information about group/batch and covariates for each sample (during the calibration of the harmonization model and the application of the calibrated model, the required input change – see sections below).

According to a standard categorization (see Hu et al., Image harmonization: A review of statistical and deep learning methods for removing batch effects and evaluation metrics for effective harmonization. Neuroimage 2023), our method is a retrospective (feature-level) method based on Principal Components Analysis (PCA) representation of the raw (unharmonized) features.

Use: Trace4Harmonization™ can be used for harmonizing numerical features such as radiomic features. The use of this tool is intended as a preprocessing step with respect to further analyses of numerical features and/or with respect to the use of these features to train machine-learning models.

Practical use through the dockerized version of Trace4Harmonization™ is detailed in the technical documentation.

Input/output formats:

<u>For the calibration of the harmonization model on a dataset of unharmonized samples</u>

INPUT

a) CSV file containing the following data:
   -- group or batch, intended as a 1D array of indexes showing how each sample belongs to a given group or batch of interest (this is the variable to be used for the harmonization of the dataset);
   -- class, intended as a 1D array of indexes showing how each sample belongs to a given class (this represents the covariate <u>not</u> to be used for the harmonization of the dataset);

-- numerical features, intended as a 2D array of features (columns = features, rows = samples).

OUTPUT

a) CSV file containing the harmonized features, intended as a 2D array of harmonized features (columns = harmonized features, rows = samples);

b) System file (.t4r) containing calibration parameters, intended as an array of parameters used to harmonize the dataset.

<u>For the application of the calibrated harmonization model to new samples</u>

INPUT

a) CSV file containing the following data:

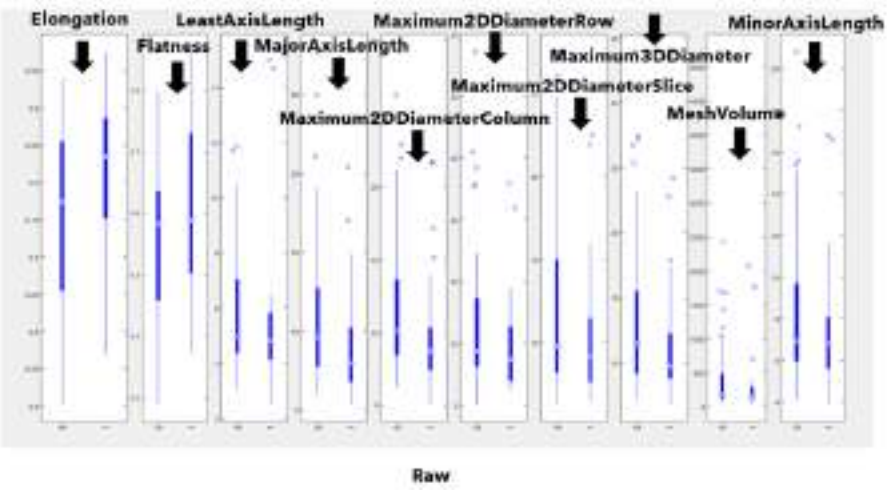
-- group or batch, intended as a numeric index showing how the current sample belongs to a given group or batch of interest (the group or batch must be present in the list of indexes used for the calibration of the harmonization model in the previous step);

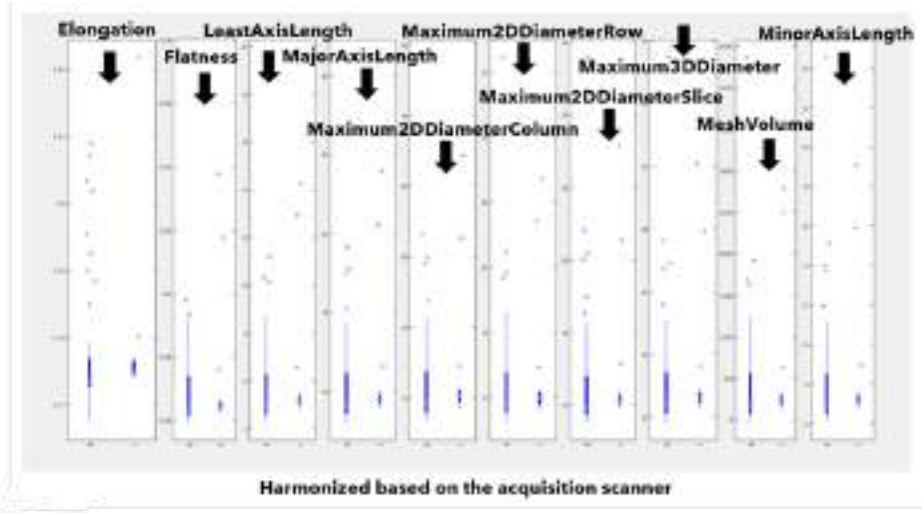
-- features, intended as a 2D array of features (columns = features, 1 row).

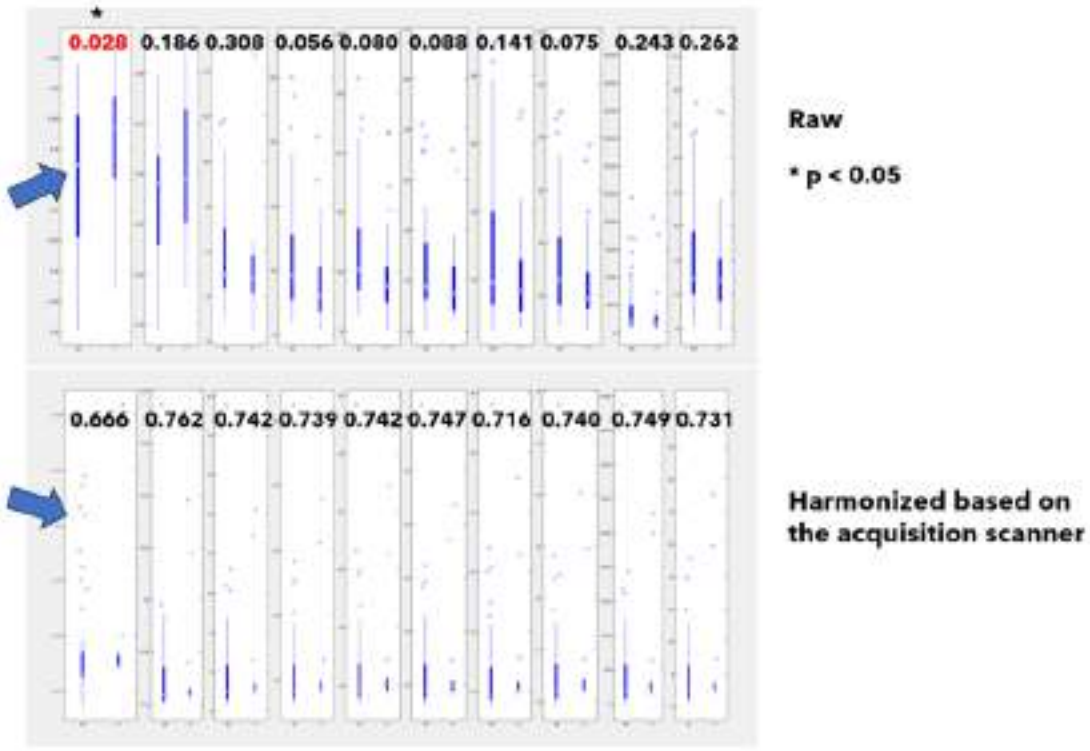
OUTPUT

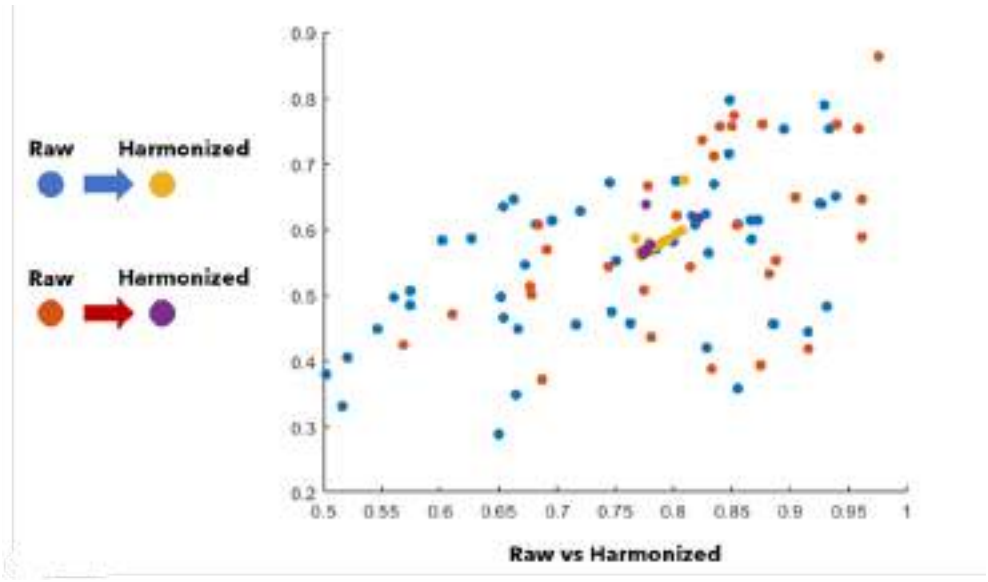
a) CSV file containing the harmonized features for the single sample, intended as a 1D array of harmonized features (columns = harmonized features, 1 row for the single sample).

Quantitative and qualitative results: The raw (unharmonized) and harmonized (based on the acquisition scanner) radiomic features extracted from representative examples (mammographic studies) are reported below.




Elongation
LeastAxisLength
Maximum2DDiameterRow
MinorAxisLength
Flatness
MajorAxisLength
Maximum3DDiameter
Maximum2DDiameterSlice
Maximum2DDiameterColumn
MeshVolume
Harmonized based on the acquisition scanner


0.028 0.186 0.308 0.056 0.080 0.088 0.141 0.075 0.243 0.262
Raw
* p < 0.05
0.666 0.762 0.742 0.739 0.742 0.747 0.716 0.740 0.749 0.731
Harmonized based on the acquisition scanner



Data: Trace4Harmonization™ is not based on ML techniques, but on the ComBat (Combining Batches) method to harmonize features (numerical variables) in a common space according to statistical measures. As such, no training was performed.

Methods. Trace4Harmonization™ is based on ComBat (Combining Batches method), to harmonize features (numerical variables) in a common space. In order to be used, these techniques need a sufficient sample size as well as the corresponding information about group/batch and covariates for each sample (during the calibration of the harmonization model and the application of the calibrated model, the required input change – see sections below).

According to a standard categorization (see Hu et al., Image harmonization: A review of statistical and deep learning methods for removing batch effects and evaluation metrics for effective harmonization. Neuroimage 2023), our method is a retrospective (feature-level) method based on Principal Components Analysis (PCA) representation of the raw (unharmonized) features.

Specifically, harmonization is applied with respect to mean and variance of the calibration dataset in the feature space; then, PCA transformation is applied and harmonization is applied in the PCA space; finally, the harmonized features are backprojected in the original feature space.
The same procedure is performed when the calibrated harmonization model is applied to new single-sample data.

No further preprocessing is applied to the input features before performing the harmonization procedure.

Specific Technical information of the current containerized version:

- CPU (no GPU required)

- Minimum 3 GB of free HDD space required for installation and running
- Operating System: Windows 10 (version 1709 or higher)
- Processors (minimum): any Intel or AMD x86-64
- Programming language: Matlab, Python
- Expected RAM usage: depending on the input data, minimum 4 GB of free RAM
- Running mode: case based, without interaction with a user interface on the screen
- Software version: Trace4Harmonization™ v1.0.00
- Additionally, Matlab runtime is installed along with the containerized software

Traceability and monitoring mechanism. Trace4Harmonization™ does not store any information into system files nor communicates any information through internet connection.
The results of the analysis are stored locally in the output files (.t4r, .csv) that can be used to read the calibration parameters and the harmonized data for further machine-learning analyses.
In addition, for traceability and monitoring, a logfile is generated for each analysis, which reports non-personal information such as the version of the tool, the outcome of the analysis (successful/unsuccessful), details on any occurred error during analysis, the path to the results of the analysis and the path to the logfile itself. The logfile is structured as follows:

```
Trace4Harmonization™ 2023-2024 v1.0.00

Warning: Trace4Harmonization™ is not a medical device. It is
not CE marked nor FDA cleared. Any use of the results of the
analysis performed using this library is intended for
research only.

03-Apr-2024 09:00:00

Expiring date: 30-Nov-2024 23:59:59

Analysis completed successfully

harm_csv_path = 'path\to\harmonized_ss_data.csv'

log_path = 'path\to\log.out'
```

Unitary tests. Unitary tests were conducted to validate, assess and evaluate all potential functionalities in use, specifically applying the proposed harmonization method to a testing dataset (generated ad-hoc and available for replication) based on radiomic features extracted from breast-imaging studies (ultrasound imaging).

Additional information for tool integration. For running the dockerized version of Trace4Harmonization, the following command must be used (after having downloaded the latest docker image of the tool)

```
docker run -ti --rm -v
local/path/to/Trace4Harmonization/app:/app/files --env-file
local/path/to/envfile --env-file
'local/path/to/.aws/credentials' python3 /app/startup.py
```

where the env-file must have the following structure

```
STORAGE_SOURCE_FOLDERPATH=dt-trace4harmonization-source-test
STORAGE_RESULTS_FOLDERPATH=dt-trace4harmonization-dest-test
OPERATION=apply
TRAINING_CALIBRATION_FILENAME=whole_data.csv
CLASSIFICATION_APPLY_FILENAME=ss_data.csv
REMOTE_RESULTS_FOLDERNAME=harmonization_11072023_151136/
```

Including information such as the operation to be performed (“calibration” or “apply”) and the name of the files on which harmonization will be applied (whole_data.csv for the calibration, ss_data.csv for the application of the calibrated harmonization model).

The csv (INPUT) files must be formatted as follows:

FOR "CALIBRATE"

- different records (patients/studies) must be put in different lines
- different features must be put in different columns
- the first column represents the batch variable to be harmonized (e.g. site/protocol/system)
- the second column represents the covariate to be considered while performing batch harmonization (e.g. diagnostic label)
- the following columns represent the features

FOR "APPLY"

- the single-record data must be formatted as a single row
- different features must be put in different columns of the row
- the first element (i.e., first column) represents the batch variable to be harmonized (e.g. site/protocol/system)
- covariate is not expressed in this case

- **User Manual**:

  Usage instructions
  Trace4Harmonization™ in its dockerized version can be accessed and used through the EUCAIM platform (currently, Chaimeleon platform is used as a support) following the instructions below:

1. Login to the Chaimeleon platform

2. Launch the Apps environment

3. Select a desktop machine (pytorch or tensorflow based). **Note**: There's no need to specify a dataset when launching the machine.

4. Access to the desktop machine you instantiated in point 3

5. Upload the data on which you want to perform the calibration phase

6. Move the data in the persistent home folder

7. From a terminal launch the following command:

   jobman submit -i trace4harmonization -- -- OPERATION=calibrate WORKDIR=~/persistent-home TRAINING_CALIBRATION_FILENAME=<file_to_be_used_for_harmonization_training>.csv

8. When the job has succeeded (you can see it by running the jobman list command), in the persistent home folder you will find the result of this first stage, that are:

   a. log.out : here you can find the logs of the run
   b. harmonized_< file_to_be_used_for_harmonization_training>.csv : the calibration file harmonized
   c. params.t4r : file containing the params for the application of the harmonization process to another file, based on the results from the first one

9. Upload another file that you want to be harmonized, given the result of the previous stage

10. Move the file to the persistent home folder

11. From a terminal launch the following command:

    jobman submit -i trace4harmonization -- -- OPERATION=apply WORKDIR=~/persistent-home CLASSIFICATION_APPLY_FILENAME=<file_to_be_harmonized>.csv

12. When the job has succeeded (you can see it by running the jobman list command), in the persistent home folder you will find the harmonized results of the input data as csv file

For the calibration of the harmonization model on a dataset of unharmonized samples

INPUT

a) CSV file containing the following data:

-- group or batch, intended as a 1D array of indexes showing how each sample
belongs to a given group or batch of interest (this is the variable to be used for
the harmonization of the dataset);
-- class, intended as a 1D array of indexes showing how each sample belongs to
a given class (this represents the covariate not to be used for the harmonization of the dataset);
-- numerical features, intended as a 2D array of features (columns = features,
rows = samples).

OUTPUT

a) CSV file containing the harmonized features, intended as a 2D array of harmonized features (columns = harmonized features, rows = samples);
b) System file (.t4r) containing calibration parameters, intended as an array of parameters used to harmonize the dataset.

For the application of the calibrated harmonization model to new samples

INPUT

a) CSV file containing the following data:

-- group or batch, intended as a numeric index showing how the current sample
belongs to a given group or batch of interest (the group or batch must be present in the list of indexes used for the calibration of the harmonization model in the previous step);
-- features, intended as a 2D array of features (columns = features, 1 row).

OUTPUT

a) CSV file containing the harmonized features for the single sample, intended as a 1D array of harmonized features (columns = harmonized features, 1 row for the single sample).

- **Integration tests**: Tests were performed in order to ensure the proper integration of the tool in the platform.
- **Results of non-functional tests:** Not applicable
- **Contact person** for the helpdesk: schiavon@deeptracetech.com, salvatore@deeptracetech.com

5- FAIRness tool validation documentation

DF1. *FAIR EVA for EUCAIM*

Partner: CSIC
Validator: David Rodríguez González
Tool state: **Developed (plugin under development)/Containerized**
Registered in bio.tools: No
Project source: EOSC Synergy
Document version: 0.1

DF1.1. Conceptual validation:

Tool description

FAIR EVA: Evaluator, Validator & Advisor is a tool developed in the EOSC Synergy project and maintained by CSIC, that checks the FAIRness level of digital objects from different repositories or data portals. FAIR EVA is a service that runs on the web. It can be deployed as a stand-alone application or as a docker container. The objective of its use in the context of EUCAIM is checking the FAIRness of the datasets included in the EUCAIM Catalogue.

Data

Dataset level metadata. Data and metadata unique IDs. It requires the object identifier (preferably persistent and unique identifier) and the repository to check.

Methods

FAIR evaluator implements a modular architecture to allow data services and repositories to develop new plugins to access its services. Also, some parameters can be configured like the metadata terms to check, controlled vocabularies, etc.
The service:

- Checks data and metadata from a digital object
- Based on some indicators, evaluates the FAIRness of the resource.
- Provides feedback to the users to improve

The config.ini file contains all the configuration parameters.

Use

To launch the application in an stand-alone mode, the steps are the following:

/FAIR_eva/fair.py &
/FAIR_eva/web.py &

The last step, running web.py is optional if you don't want to deploy the web visual interface. The ports to run the app are 9090 for the API and 5000 for the web interface. They can be configured if needed.

Input/Output formats
Input: DOI or handle PID.

Output: JSON with scores for the different RDA FAIR metrics and feedback.

Quantitative results
Scores for the RDA FAIR Metrics.

Qualitative results
Tests passed or not.

Additional information

Successful use case: Digital.CSIC https://fair.csic.es/es
Currently the tool is also under development/extension by the EPOS project

Licence: Apache License, version 2.0

Main repository: https://github.com/IFCA-Advanced-Computing/FAIR_eva

DF1.2.Technical validation:

Methods:
https://github.com/IFCA-Advanced-Computing/FAIR_eva/blob/main/docs/technical_implementation.md

Specific technical information

- **GPU/CPU:** CPU
- **Programming language:** Python

- **Expected RAM usage**: 0.5 GB
- **Software version**:
- **Libraries**: Numpy, pandas, FDP
- **Minimal security measures (containers):**
    - Writing to host data restricted to a non-root user: Yes/No
    - Container require to be executed in a privileged mode:No

Additional information for tool integration

The tool is dockerised.

DF1.3. Integration validation

Bio.tools registration: No

Most Common errors

Error	Description	Solution
Internal server error	The server encountered an internal error and is enable to complete the request.	If due to overload, retry. Check configuration and server logs.
No access information can be found in the metadata for: [terms]	Either the dataset doesn't contain the required metadata or the expected terms have not been correctly configured.	Double check the dataset ID and the configuration

Communication channel (helpdesk)

- Fernando Aguilar
- David Rodríguez

User manual

Documentation for installation and use is maintained in https://github.com/IFCA-Advanced-Computing/FAIR_eva/blob/main/docs/index.md