EUCAIM
CANCER IMAGE EUROPE

Project title: European Federation for Cancer Images

Project acronym: EUCAIM

Grant Agreement: 101100633

Call identifier: DIGITAL-2022-CLOUD-AI-02

D7.4. Definition of a benchmarking test set to be used for comparing tools and technologies

Responsible partner:	BSC
Author(s):	Carles Hernandez-Ferrer (BSC), Laura G. Antiga (BSC), Josep Lluis Gelpí (UB), Salvador Capella (BSC)
Reviewers (WP)	Dimitris Filos (AUTH)
Date of delivery:	30.06.2025
Version:	V1
Due date:	Month 30
Type:	Report
Dissemination level:	Public

Table of contents

1. Introduction

1.1 Document Purpose

The purpose of this document is to define the benchmarking infrastructure and processes that will be used within the EUCAIM project to evaluate and compare tools and technologies. This includes detailing the architecture involved, the components used to manage federated execution, and how benchmarking events are designed, deployed, and analysed.

1.2 Document Scope

This deliverable outlines the current status and intended use of the benchmarking test sets in EUCAIM. It includes the technical setup, the role of the Federated Execution Manager (FEM), the integration with EUCAIM components, and the next steps for implementation and execution. It also presents the structure and workflow of ongoing benchmarking events.

2. The Architecture

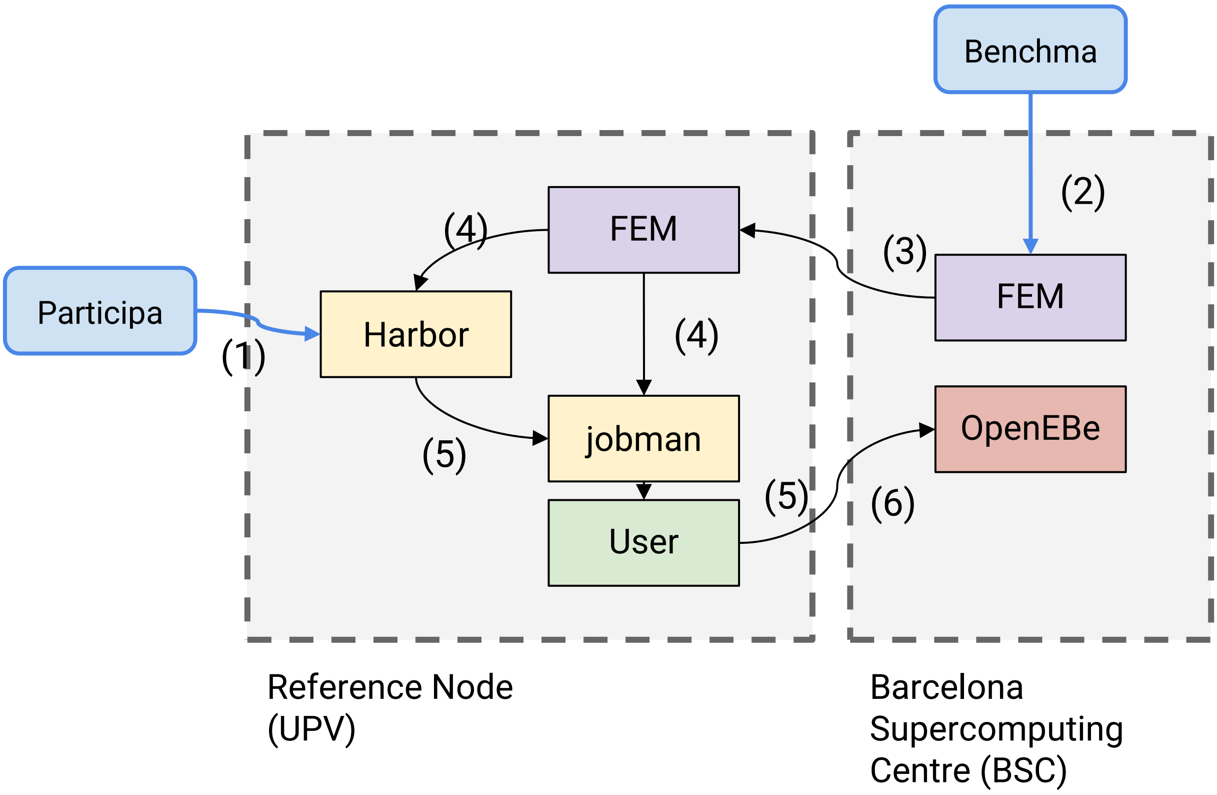

Figure 1. Global description of the structure for benchmarking in EUCAIM. (1) The participants onboard their software and, in the last step, push the docker image into Reference Node' Harbor. (2)The Organization committee of the benchmarking event connects to FEM Orchestrator to trigger each participant on the selected dataset. (3) FEM Orchestrator prepares the job to be triggered at the Reference Node and FEM Client pulls it. (4) FEM client checks if the requested image is available in the Harbor and, if so, requests to the local job manager, jobman, to trigger that specific

image on the selected dataset. (5) jobman pulls the image from Harbor and connects it with the selected dataset, saving the results on the User Space. (6) FEM Orchestrator pulls the results and, after a hand curation process, they are submitted into OpenEBench.

2.1 EUCAIM Federated Architecture

The infrastructure of EUCAIM is based on a federation of nodes providing computation and data resources. For partners who cannot provide such resources, two reference nodes serve as central points where data can be deposited and computations can be run.

To leverage this federated architecture, the Federated Execution Manager (FEM) plays a key role in orchestrating the execution of analytical jobs across the network. FEM enables the distribution of analytical workflows by sending computational tasks to each node—whether reference or federated—and collecting the resulting outputs.

2.2 Federated Execution Manager (FEM)

The Federated Execution Manager (FEM) is a software solution originally developed for a previous EU-funded project (https://www.datatools4heart.eu/) and is now being extended and adapted by UB and BSC for use in EUCAIM.

FEM supports the federated execution of analytical jobs across the network and consists of two primary components: the **orchestrator** and the **client**.

- **The Orchestrator**, currently hosted at BSC (and planned to be migrated to a reference node at UPV), offers a graphical interface through OpenVRE and an API to manage and configure analytical workflows. Integration is ongoing with EUCAIM's Authentication and Authorization Infrastructure (LSAAI) and with the EUCAIM Negotiator, which governs dataset access based on user project memberships.
- **The Client** is a lightweight daemon that runs on each participating node. It pulls jobs from the orchestrator using a message broker (RabbitMQ), reads local infrastructure configuration, executes the analytical tasks, and reports their status (pending, running, success, failure) back to the orchestrator.

2.3 Benchmarking Infrastructure in EUCAIM

The benchmarking infrastructure leverages the developments from WP4 and WP6 and relies on FEM to configure and execute benchmarking tasks.

Periodically, WP7 will define a set of benchmarking goals and launch a call for participation to all EUCAIM partners. This will trigger the setup of a new benchmarking event, which also corresponds to a new section in OpenEBench.

> *OpenEBench is a European infrastructure for the benchmarking and evaluation of bioinformatics tools and services. It supports community-driven assessments by providing datasets, metrics, and platforms for fair and reproducible comparisons.*

> *Developed under ELIXIR, it promotes transparency, interoperability, and performance tracking across biomedical software. OpenEBench also contributes to the FAIRification of benchmarking data and workflows.*

Upon joining the event, partners submit their software for benchmarking. Submissions follow WP4 containerization guidelines and are uploaded to the Harbor registry of the reference node.

Concurrently, WP7 — together with the coordination team and other relevant WPs — selects appropriate datasets to be used as ground truth. These datasets must be:

- Properly annotated, in terms of minimal metadata for EUCAIM's data catalogue and expert curated segmentation.
- Not previously used for training by any of the submitted tools (a special flag "benchmarking only" was created in the data catalogue so the data access committee was instructed to forbid access to the datasets with those flag unless a benchmarking event is ongoing and is the requested of the dataset).

The benchmarking process proceeds as follows:

1. The dataset is made available to each containerized tool.
2. Each tool is executed on the dataset using FEM.
3. The output is compared to the ground truth annotations, and performance metrics are extracted (e.g., segmentation accuracy, precision, recall).
4. Metrics from all tools are aggregated into a unified result file.
5. Aggregated results are submitted to OpenEBench for dissemination and comparison.

3. Next Steps

An open benchmarking event focused on segmentation is currently ongoing in EUCAIM. Full details are available in the shared live document here (static versión can be found as an annex). Summary:

- Three datasets from the ChAImeleon project were identified, uploaded to the EUCAIM catalogue as "benchmarking only" to preserve them for evaluation and avoid future training use.
- A public call for participation was issued to all EUCAIM partners, alongside personal invitations based on tools listed in the EUCAIM Toolbox.
- The benchmarking committee is actively working with partners to finalize containerization and test the infrastructure at the reference node hosted at UPV.

Key Dates

- Event Preparation: May 2 – May 30
- Software Submission: June 2 – August 1
- Benchmarking Execution: August 1 – August 11

- Metrics Extraction & Publication: August 11 – August 30

4. Conclusions

This document has presented the definition, components, and procedures involved in establishing a benchmarking test set for EUCAIM. The infrastructure — anchored by the Federated Execution Manager and supported by WP4 and WP6 developments — offers a robust and scalable foundation to evaluate tools and technologies in a transparent and reproducible manner.

Ongoing efforts, such as the segmentation benchmarking event, will serve as pilot initiatives to refine the process and ensure seamless integration with EUCAIM's federated architecture. Future benchmarking rounds will continue to contribute towards the project's objective of improving analytical tool quality and reproducibility across Europe.

Annex

Description of the Open Benchmarking Event on Segmentation

Event Committee

Add yourselves here if you want to help organizing the event (tasks will be distributed according to participation):

- Carles Hernandez-Ferrer (BSC)
- Miriam Groeneveld
- José Almeida
- Philip Seebök

Timeline

- Event Preparation: May 2 – May 30
- Software Submission: June 2 – August 1
- Benchmarking Execution: August 1 – August 11
- Metrics Extraction & Publication: August 11 – August 30

Extra tasks to be considered

During the “Software submission”:

- Revisit the metrics extraction python package
- Decide if we want to push, now, an automated way to publish metrics into GC and/or OEB

Challenges

Ch1: Tumor segmentation

Ch2: Organ and Tumor segmentation

Datasets

Ch1 + Ch2

- Prostate Cancer (MRI) cases - Catalogue Link [here]
    - Studies/Subjects count: 40/40
    - Age range: Between 52 years and 84 years
    - Year of diagnosis range: Between 2015 and 2023
    - Sex: Male (40)
    - Modality: MR (40)
    - Manufacturer: General Electric (21), Siemens (17), Philips (2)
    - Body part:
        - PROSTATE (16)

    - ARM (4)
    - PELVIS (2)
    - HEAD (1)
    - Unknown (22)

- Colon Cancer (CT) cases - Catalogue Link [here]

  - Studies/Subjects count: 40/40
  - Age range: Between 54 years and 86 years
  - Year of diagnosis range: Between 2011 and 2022
  - Sex: Male (23), Female (17)
  - Modality: CT (40)
  - Manufacturer: Philips (21), General Electric (11), Toshiba (6), Siemens (2)
  - Body part:
    - ABDOMEN (7)
    - CHEST (3)
    - COLOSCANNER (2)
    - LUNG (1)
    - SPINE (1)
    - TAP (1)
    - Unknown (31)

- Lung Cancer (CT) cases - Catalogue Link [here]

  - Studies/Subjects count: 40/40
  - Age range: Between 41 years and 81 years
  - Year of diagnosis range: Between 2015 and 2022
  - Sex: Male (30), Female (10)
  - Modality: CT (39), DX (1)
  - Manufacturer: Philips (15), General Electric (14), Siemens (8), Toshiba (3)
  - Body part:
    - CHEST (13)
    - ABDOMEN (4)
    - THORAX / ABDOMEN (3)
    - HEAD (2)
    - LUNG (2)
    - TAP (1)
    - CRANE (1)
    - WHOLEBODY (1)
    - Unknown (25)

Task specific metrics for Segmentation

Dice score

It is a coefficient that measures the similarity between two sets. The coefficient ranges from 0 to 1, where 1 indicates that the two sets are identical, and 0 indicates that the two sets have no overlap.

Dice coefficient = 2 * |A ∩ B| / (|A| + |B|)

Where |A| represents the number of elements in set A, and |B| represents the number of elements in set B. |A ∩ B| represents the number of elements that are present in both sets.

Intersection over union

Provides a ratio of the intersection to the union of the ground truth and prediction masks. The coefficient ranges from 0 to 1, where 1 indicates that the two areas fully overlap, and 0 indicates no overlap between the two areas.

Intersection Over Union = Area of Intersection / Area of Union

Hausdorff Distance

It is a distance that measures how far two subsets of a metric space are from each other by identifying the greatest distance one must travel from a point in one set to reach the closest point in the other set.

$d_H(A,B) = \max\{ \sup_{a\in A} \inf_{b\in B} d(a,b), \sup_{b\in B} \inf_{a\in A} d(b,a) \}$

Here, d(a,b) denotes the distance between points a and b. The first term finds the point in A that is farthest from any point in B, and vice versa for the second term. The Hausdorff Distance is the larger of these two values, capturing the worst-case mismatch between the sets.

Normalized Surface Distance

It is a distance that quantifies the average error between the predicted and ground truth surfaces, normalized by a tolerance threshold.

$NSD = 1 / |S_{gt}| \ SUM_{x\in Sgt}(F(d(x, S_{pred}) < \tau)$

Here the S_{gt} is the surface defined by the ground truth while S_{pred} is the surface of the prediction. Therefore the $d(x, S_{pred})$ is the shortest distance from a point x to the predicted surface. F is the indicator function (1 for true and 0 otherwise). Finally,τ is the predefined distance tolerance

Performance metrics

During the benchmarking event, we aim to collect a set of performance metrics to evaluate the resource usage of each submitted segmentation software. The metrics to be gathered include:

- Execution time

- Memory usage
- CPU usage
- GPU usage

To extract these metrics, we will use the following tools and methods:

- Execution time will be measured using jobman, which tracks the start and end times of each job.
- CPU and memory usage will be monitored using docker stats, which provides resource usage information including CPU (%) and memory (%) for containers.
- GPU usage will be collected using nvidia-smi, which allows us to monitor GPU utilization, memory consumption, and active processes.

Alternatively, we will leverage the JobStatus information from the Kubernetes infrastructure to complement and validate our performance tracking, particularly for GPU-based workloads.