PARROT - A versatile platform for AI-driven image segmentation and dose prediction

Wei Zhao[1,✉], Margerie Huet[2], Sylvain Deffet[3], Edmond Sterpin[4], Ana Barragan Montero[2], Sébastien Jodogne[1], and John Lee[1,2]

[1] Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), UCLouvain, Belgium. ✉ wei.zhao@uclouvain.be
[2] Molecular Imaging, Radiotherapy & Oncology (MIRO), UCLouvain, Belgium
[3] Ion Beam Applications SA, Louvain-La-Neuve, Belgium
[4] KU Leuven, Department of Oncology, Laboratory of Experimental Radiatiotherapy, Leuven, Belgium

Abstract. This paper details the design and architecture of PARROT (Platform for ARtificial intelligence guided Radiation Oncology Treatment), a free and open-source, Web-based platform that aims to improve therapeutic decision-making through advanced artificial intelligence (AI) techniques. The primary goals of PARROT are to facilitate the visualization of the outputs of AI models, to provide tools for annotating and modifying AI segmentation results, and to assist in comparing treatment plans and selecting the most suitable option. PARROT addresses common challenges in AI research, such as simplifying the execution and management of AI experiments and facilitating the comparison of results between published and institution-trained models. It enables users to interact with AI models for radiation oncology in real time, allowing them to understand and refine model outputs by visually inspecting and correcting biases. PARROT provides an end-to-end workflow with an easy-to-use graphical interface that promotes the implementation of AI models in clinical settings.

Keywords: medical imaging, radiation oncology, artificial intelligence, image segmentation, open-source

1 Introduction

Artificial intelligence (AI) plays an important role in radiation oncology and is currently undergoing intensive research. Indeed, AI can assist doctors by analyzing medical images, such as CT scans, MRIs, and X-rays, faster and more accurately than humans [10]. This enables better detection, diagnosis, and monitoring of cancer. AI can also aid in planning radiation treatments by accurately mapping the location of the tumor, ensuring that the radiation beams effectively target the cancer cells while sparing healthy tissues [11].

The development of AI models that are suitable for use in the everyday clinical practice is a major challenge. In a lengthy process, researchers create,

evaluate, and compare new models, refining promising candidates step by step. The opinion of a medical expert is often requested to assess the quality of a model. Obtaining this expert's opinion requires not only high-quality models but also effective tools for visualizing and analyzing results.

In this paper, we introduce PARROT (Platform for ARtificial intelligence guided Radiation Oncology Treatment) as a Web-based, free and open-source platform designed for visualizing imaging data as well as evaluating AI models for segmentation and dose inference. PARROT consists of multiple components that enable users to manage and execute various AI models, as well as assess their predictive results. AI models can perform two tasks in PARROT: segmentation of regions of interest (target and organs at risk) and dose prediction for different treatment modalities [6]. PARROT offers a rich set of features, including a manual segmentation editing tool that allows clinical experts to annotate results, as well as a clinical decision support system for choosing between PT (proton therapy) and XT (X-ray therapy) treatments. Users can create their own models, edit existing models, or access a library of pre-trained models. PARROT is a Web-based application that can be either self-hosted on a local computer or hosted on a remote server. For inference, the AI models are executed in Python virtual environments, which provides a consistent and isolated execution context.

2 Related Work

Given the complexity of the AI models and algorithms for radiation oncology, an open platform is needed to help researchers save time and enhance efficiency. We compared various software that supports the DICOM standard, which is essential for facilitating data exchange. Firstly, RayStation is a commercial desktop software with extensive customization options [2]. Its proprietary nature makes it difficult for researchers to access. Secondly, CERR (Computing Environment for Radiotherapy Research) is an open-source desktop application that provides dosimetry, imaging, and comparison tools [1]. For interoperability, CERR stores modified regions of interest (ROIs) as new DICOM RT-STRUCT instances. Finally, 3D Slicer is an advanced open-source desktop software with multiple extensions [3], including a plugin for radiotherapy (SlicerRT). 3D Slicer provides features for labeling and segmentation, and recently added segmentation models such as TotalSegmentator to its workflow.

Unlike PARROT, none of these three well-known applications provide a Web interface, which would enhance collaboration among researchers. Furthermore, they do not support audit logs, which are essential for tracking modifications, including timestamps, object names, and reasons. Instead of tracking modifications, they generate new image data in various formats (such as DICOM-SEG, NIfTI, MRRD, etc.), increasing data volume and management complexity. In addition, 3D Slicer is complex and requires significant time to master because of its generic nature, while PARROT offers a simpler interface that is specifically focused on radiotherapy. This paper provides a detailed description of the

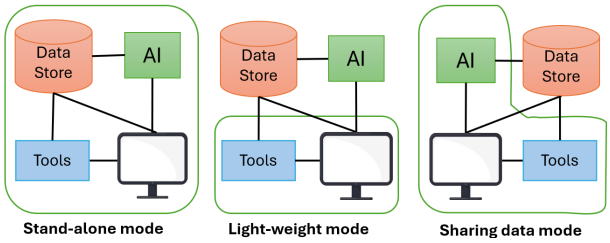

Fig. 1. The three deployment modes of PARROT. The rounded rectangles highlight the application components that run locally on the end user's computer. The AI Marketplace, not depicted in this figure, is hosted on remote servers in all configurations.

technical design and software architecture of PARROT, while an overview of its advanced features can be found Huet Dastarac et al. [7].

3 Methodology

This section provides a high-level overview of PARROT's architecture and design choices, followed by a detailed review of its components.

3.1 Overview

The main objective of PARROT is to improve the decision-making process for radiation treatment plans by using AI models for segmentation and dose inference, and by providing tools to evaluate the outputs of such AI models. Accurate evaluation is important to ensure that the radiation treatment plan targets the intended area while minimizing exposure to surrounding healthy tissues. To this end, the platform incorporates interactive functionality to process image data, to run AI models, and to visualize the results. Potential users of PARROT include physicians, medical physicists, and AI researchers.

The architecture of PARROT can be dissected into four abstract functional blocks. Firstly, the *data store* is responsible for storing persistent data, which notably includes the imaging studies and the outputs of the AI models. Secondly, the *AI* functional block hosts the different AI models that can be executed. Thirdly, the *evaluation tools* implement various types of tools for statistical data analysis and comparison. Finally, the *user interface* offers the graphical environment for image visualization, data management, and segmentation editing.

As depicted in Figure 1, PARROT can be deployed in different ways. In the *standalone mode*, all application components run locally on the personal computer of the user. This mode may lead to slow AI model inference, as many users only have modest laptops without a high-end GPU (Graphics Processing Unit). However, the advantage of the standalone mode lies in its ease of deployment and setup. Additionally, no data is stored on remote servers, which is important when privacy policies prohibit the transfer of patient data over the network. In

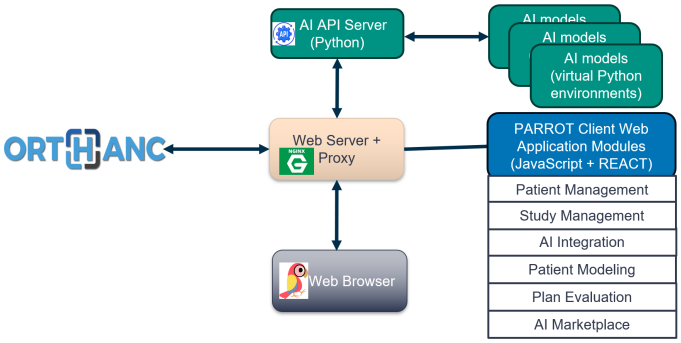

Fig. 2. Internal architecture of PARROT.

this standalone mode, the user is responsible for importing data received from others and exporting any results they wish to share.

A second possible deployment consists in installing PARROT on a dedicated server. This mode is referred to as the *lightweight mode*, where only the front-end and the evaluation tools run on the local computer. This mode of operation enables transparent data sharing between multiple users. Furthermore, since the AI model can run on server hardware shared among multiple users, it becomes a cost-effective way to provide PARROT with a high-end GPU, thereby improving AI model inference times. As a final possible configuration, the *sharing data mode* offers data sharing between users by using a dedicated data store server, while all other components run locally.

The concrete implementation of the above functional blocks is shown in Figure 2. The PARROT user interface is implemented as a Web application that is made of a set of static resources (i.e., HTML, JavaScript, and CSS files). These resources are delivered by a Web server and are executed by the Web browser of the client computer. The evaluation tools are integrated into the Web user interface and are therefore also executed by the Web browser. This explains why the lightweight mode keeps both the front-end and evaluation tools on the local computer. PARROT uses the free and open-source Orthanc server [9] as its data store. Orthanc is a lightweight PACS (Picture Archiving and Communication System) specifically designed for storing imaging data according to the DICOM specification. Finally, the AI functional block takes the form of a Web server written in Python using the Flask framework that orchestrates a set of virtual Python environments responsible for running the inference of AI models. The RESTful API for the data store and the AI functional block are routed through the same Web server using a reverse proxy implemented with the `nginx` Web server. The `nginx` software is also responsible for serving the static resources of the user interface and the evaluation tools. The following sections review these different functional blocks in more detail.

3.2 User interface and evaluation tools

The Web-based user interface consists of six modules. The two first modules are related to the management of patients and imaging studies. More precisely, the *Patient Management* module provides a user interface to browse and manage the patient data stored on the Orthanc server. With the *Study Management* module, the user can look for the available imaging studies on the Orthanc server and select and load a study for further processing in the other modules.

The next two modules are central to the user experience, as they provide an environment for displaying and editing radiation oncology information. Firstly, the *Patient Modeling* module provides a Multi-Planar Reconstruction (MPR) Web viewer to visualize DICOM imaging studies (currently, CT and MRI modalities are supported, but other modalities can be added if needed), segmentation data, and dose data. Segmentation data can be imported from existing structure sets in the DICOM RT-STRUCT format or be generated by AI models. Dose data in the DICOM RT-DOSE format can be imported as well. This module also provides tools to manually edit and annotate segmentation data. The history of the segmentation modifications is stored onto the Orthanc server and can be browsed and replayed step by step. Secondly, the *Plan Evaluation* module implements the rendering and comparison of radiation treatment plans. It can display existing dose data that are either encoded as DICOM RT-DOSE instances or predicted by AI models. For the evaluation and comparison of different plans, this module computes Dose-Volume Histograms (DVH), statistics to compare two doses, as well as the Normal Tissue Complication Probability (NTCP). Furthermore, the user can define clinical goals against which the compared plans are checked.

The last two modules are related to the management of AI models. The *AI Integration* module allows the user to configure different models, edit the necessary Python code for executing their inference process, and run them on the loaded imaging study. The *AI Marketplace* module provides access to a library of AI models hosted on the Hugging Face AI Community server, allowing users to download pre-trained models.

The user interface and associated evaluation tools are implemented in the Javascript language using the React.js framework, and are delivered to the client Web browser by the `nginx` server. The MPR viewer is built on the top of the VTK.js library written in JavaScript.

3.3 Data storage

As explained in Section 3.1, PARROT uses Orthanc, a free and open-source DICOM server [9], as its data repository. Orthanc is capable of storing several dozen terabytes of data, so for all practical purposes, its role as data store in PARROT is limited only by the available local (in stand-alone mode) or remote (in lightweight and sharing data mode) storage space. Note that a common setup for the well-known OHIF viewer for oncology [12] involves using Orthanc as its data store. Orthanc offers a built-in RESTful API for accessing, writing, and

searching DICOM resources in any programming language. PARROT leverages this API to manage its image data.

Users can import and export patient studies in the DICOM format directly through the PARROT user interface without interacting with the Orthanc server. PARROT accepts imaging series associated with the following DICOM modalities: CT, MRI, REG (image registration), RT-DOSE (dose distribution), RT-PLAN (treatment plans for radiation oncology), and RT-STRUCT (segmentation structure sets). The AI models can retrieve imaging studies from the Orthanc server and store their prediction back onto the same server, encoded as DICOM RT-STRUCT series (for segmentation predictions) or DICOM RT-DOSE series (for dose predictions).

Furthermore, PARROT stores manually edited and annotated ROI segmentations on the Orthanc server. The modification history of segmentation data is stored as a so-called *attachment* to the original structure set in Orthanc. Attachments are a feature of Orthanc that allows the storage of arbitrary user-defined data associated with a DICOM resource. In our case, the attachment belongs to the unmodified segmentation data, as uploaded by the user or by an AI model. To reduce the size of the modification history, when the contour of a ROI is edited, PARROT calculates for that ROI the difference between the modified version and the original version of the segmentation represented as a 3D bit field and compresses it using run-length encoding. The result is stored in JSON format in the attachment together with metadata (e.g. date of the modification, comment, etc). In the user interface, the user can navigate between the steps of the modification history and view and delete individual steps.

3.4 AI models

The AI server is part of the back-end of PARROT. Its role is to run the inference of AI models. Through its RESTful API, the server can be instructed by the front-end to start or to stop the execution of an AI model. It is also responsible for passing configuration parameters to the models. The actual models are implemented as independent and standalone virtual Python environments containing the script (Python code, binary executables, etc.) and data (model weights, configuration options, etc.) for the inference. Python virtual environments offer the advantage of isolating the execution of AI models, relieving users from the task of installing additional Python packages, and preventing version conflicts between the packages that are required by the different models. The models can be manually added to the server or downloaded from the AI Marketplace[5], using the corresponding user interface modules described in Section 3.2.

AI model inference begins with the user uploading image data through the interface. Next, a model configuration is created, including mappings that customize the model without changing its code. These mappings can rename segmentation sets to match the expectations of the AI model. Once configured, the

[5] The AI Marketplace is hosted on the Hugging Face repository of MIRO UCLouvain at: https://huggingface.co/AI4MIRO

user selects the AI model and initiates the image segmentation or dose inference process. A built-in editor allows modifications to the Python code of the inference script.

Running an AI model in PARROT involves three steps: Firstly, in the preprocessing step, the imaging data is downloaded from the Orthanc server and converted into a model-specific format, often NIfTI. For segmentation inference, a DICOM CT study is used, while dose inference requires both the CT volume and RT-STRUCT sets. Secondly, the inference step runs the AI model and caches the output. Finally, the post-processing step converts the output into DICOM format (RT-STRUCT for segmentation or RT-DOSE for dose), which is sent to Orthanc. Users can review and refine the results in the Patient Modeling module.

Note that PARROT does not impose particular restrictions on the internal operation or structure of a model. Consequently, the effort to integrate existing models such as nnU-Net [8] or SwinUNETR [5] mostly consists in writing code for the pre- and post-processing.

4 Results

The source code of the PARROT platform is released as free and open-source software under the MIT license[6]. The binaries of the latest standalone version of PARROT are also freely available for download[7]. No special server setup is required to deploy PARROT. All necessary components are included in the release package. The PARROT currently runs on Microsoft Windows, with GNU/Linux support planned for the next release. A demonstration video is also available[8].

Figure 3 illustrates how PARROT would be used by a researcher or physician looking for the highest quality AI models for segmentation and treatment planning. With the functions provided by the user interface and the AI server, the user can quickly iterate by running a model, reviewing the results, and adjusting the model parameters and code. Previous results are stored on Orthanc and are never overwritten, allowing the user to compare them and track improvements.

As a second example, let us consider a scenario where a medical expert compares AI-predicted segmentation with a clinical baseline using PARROT. Figure 4 shows the user interface of the Patient Modeling module after loading an imaging study that contains a CT volume and baseline structure sets. In this scenario, the user has selected several structures to be displayed as overlays on the top of the CT image. As shown in Figure 5, the code editor then opens, allowing the user to review and modify the model script before starting inference. Once the AI model is configured, the user can run it to get a segmentation prediction. The outcome is finally displayed, allowing the user to compare it against the baseline and assess its performance in a radiation treatment plan (cf. Figure 6).

In this scenario, the user is not satisfied with the output of the AI model, as they have noticed that the model incorrectly attributed shadows of the CT bed

[6] The source code of PARROT is available at: https://gitlab.com/ai4miro/parrot
[7] The homepage of PARROT is: https://huggingface.co/spaces/AI4MIRO/parrot.ai
[8] The location of the video is: https://youtu.be/1YO3PYhbvzs

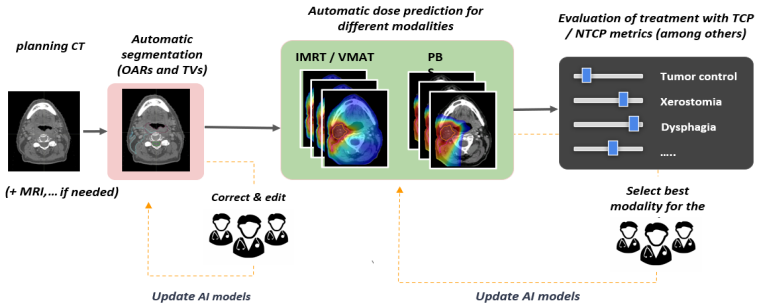

Fig. 3. Imaging data workflow in PARROT.

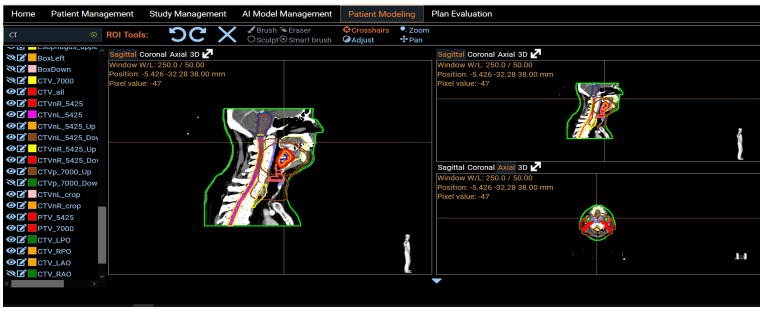

Fig. 4. Patient Modeling module rendering the patient imaging data.

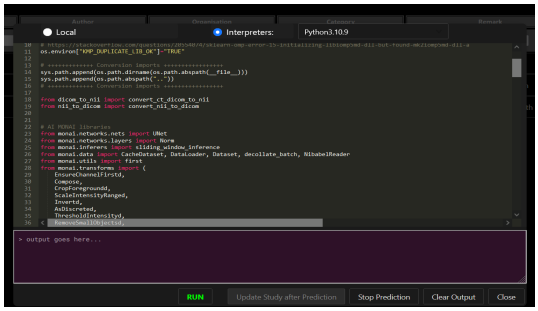

Fig. 5. Configuration of an AI model before running it.

to the body segmentation, which is visible to the left of the head in the sagittal plane view in Figure 6. The user can subsequently choose to manually correct the segmentation (left side of Figure 7) and immediately check how this adjustment would affect the quality of the treatment plan (Figure 8). By observing the updated dose distribution and assessing the accuracy of the modified segmentation, this interactive feedback loop enables continuous improvement, increasing accuracy and user satisfaction. The modifications applied to the segmentation

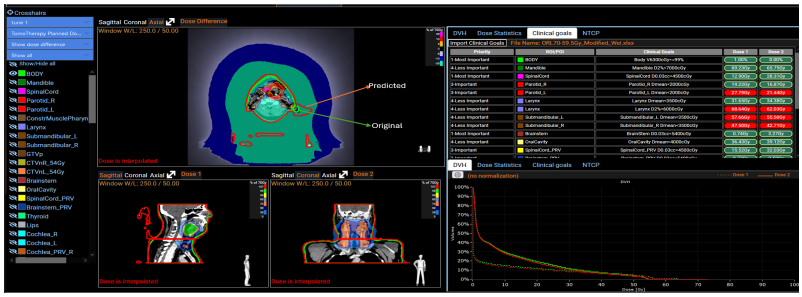

Fig. 6. Patient Modeling module rendering a predicted segmentation.

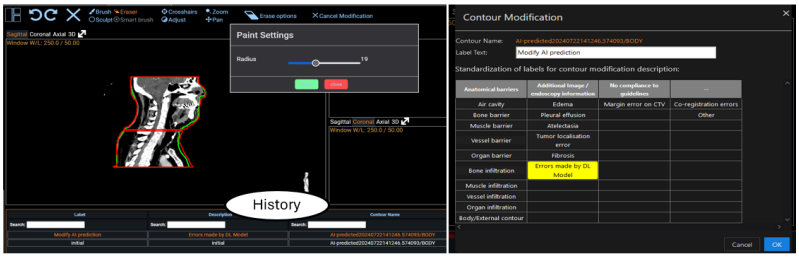

Fig. 7. Manual modification of an AI-generated segmentation. The right side of the image illustrates how audit logs are available to track the history of the segmentation.

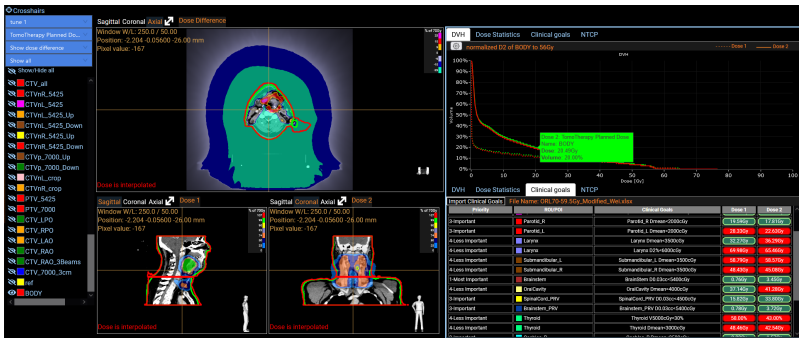

Fig. 8. Plan Evaluation module assessing the quality of a revised treatment plan, after a modification to the segmentation.

are tracked in the history of the dataset (right side of Figure 7), enabling the author of the segmentation to use this information for further refinements.

5 Conclusion

This paper introduces PARROT, a platform for AI-guided radiation oncology treatment. PARROT integrates image segmentation and dose prediction mod-

els, enabling the seamless execution, modification, and evaluation of AI model predictions within a single system. A tight interactive feedback loop allows the impact of model adjustments on the quality of the treatment plan to be assessed. The architecture of PARROT supports flexible deployment, utilizing both local and remote resources efficiently. Future development will focus on incorporating other image data types such as MRI with registration, on implementing the import of segmentation masks from various sources (notably 3D Slicer), on adding visualisation of uncertainty estimation of AI models [4], and on integrating PARROT into a pedagogical environment aimed at physicians and engineers.

References

1. Apte, A., Iyer, A., Crispin-Ortuzar, M., Pandya, R., Van Dijk, L., Spezi, E., Thor, M., Um, H., Veeraraghavan, H., Oh, J., et al.: Extension of CERR for computational radiomics: A comprehensive MATLAB platform for reproducible radiomics research. Medical physics **45**(8), 3713–3720 (2018)
2. Bodensteiner, D.: RayStation: External beam treatment planning system. Medical Dosimetry **43**(2), 168–176 (2018)
3. Choueib, S., Pinter, C., Lasso, A., Fillion-Robin, J., Vimort, J., Martin, K., Fichtinger, G.: Evaluation of 3D Slicer as a medical virtual reality visualization platform. In: Medical imaging 2019: image-guided procedures, robotic interventions, and modeling, vol. 10951, pp. 279–286. SPIE (2019)
4. Gillmann, C., Saur, D., Scheuermann, G.: How to deal with uncertainty in machine learning for medical imaging? In: IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX), pp. 52–58 (2021)
5. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., Xu, D.: Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. In: International MICCAI brainlesion workshop, pp. 272–284. Springer (2021)
6. Huet-Dastarac, M., Michiels, S., Rivas, S., Ozan, H., Sterpin, E., Lee, J., Barragan-Montero, A.: Patient selection for proton therapy using normal tissue complication probability with deep learning dose prediction for oropharyngeal cancer. Medical Physics **50**(10), 6201–6214 (2023)
7. Huet Dastarac, M., Zhao, W., Deffet, S., Longton, E., Cvilic, S., Sterpin, E., Lee, J., Barragan-Montero, A.: PARROT: A Web platform for AI-driven radiation therapy planning. In: ICCR (2024)
8. Isensee, F., Jaeger, P., Kohl, S., Petersen, J., Maier-Hein, K.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)
9. Jodogne, S.: The Orthanc ecosystem for medical imaging. Journal of Digital Imaging **31**(3), 341–352 (2018). DOI 10.1007/s10278-018-0082-y
10. Pinto-Coelho, L.: How artificial intelligence is shaping medical imaging technology: A survey of innovations and applications. Bioengineering **10**(12), 1435 (2023)
11. Wang, C., Zhu, X., Hong, J., Zheng, D.: Artificial intelligence in radiotherapy treatment planning: present and future. Technology in cancer research & treatment **18** (2019)
12. Ziegler, E., Urban, T., Brown, D., Petts, J., Pieper, S., Lewis, R., Hafey, C., Harris, G.: Open Health Imaging Foundation viewer: An extensible open-source framework for building web-based imaging applications to support cancer research. JCO Clinical Cancer Informatics **4**, 336–345 (2020). DOI 10.1200/cci.19.00131