User guide

OpenTPS

WORK IN PROGRESS

version 3.0.0

19 February 2026

Questions or feedback?

If you don’t find the answer to your question in this guide, feel free to ask the community or the development team on our **Discourse forum**

🔗 Other OpenTPS resources

🌐 **Website** – Project overview, downloads, and news
📺 **YouTube** – Tutorials, demos, and recorded talks
📰 **Newsletter** – Stay informed about releases and events
💻 **GitLab** – Source code, issues, feature requests, branches, and forks

Index

1. General information

1.1. About

OpenTPS is an **open-source treatment planning system** (TPS) developed to support research and innovation in **radiation therapy**, with a strong focus on **proton therapy**. The project originated in 2020 within the MIRO and PiLab laboratories at UCLouvain and has since grown into an international research initiative, benefiting from contributions by academic and industrial partners such as IBA s.a., Massachusetts General Hospital from Harvard Medical School, University of Ljubljana, Enger Lab at McGill University, or the RAPTOR consortium, among others, as well as independent contributors from the medical physics community.

The software is built around a modular architecture, with a **core library** handling data structures, dose calculation, optimization, and image processing algorithms, and a **graphical user interface (GUI)** for visualization and interaction (see Fig. X). OpenTPS supports diverse functionalities (see Section X) including DICOM import and export, image registration (rigid and deformable), Monte Carlo dose calculation with MCsquare and analytical dose calculation for photons, robust treatment plan optimization and evaluation. In addition, new functionalities can be added through dedicated plug-ins (see Section X).

1.2. Disclaimer

OpenTPS is provided as a **research and educational platform** and has **not been cleared or approved for clinical use by any regulatory authority** (e.g., CE, FDA).

Any use of OpenTPS or any of its components within a **clinical workflow** is undertaken **at the user's own risk**. The user or institution assumes full responsibility for ensuring compliance with all applicable **local, national, and international regulatory requirements**.

The developers and contributors of OpenTPS make **no warranties**, express or implied, regarding the accuracy, reliability, or clinical suitability of the software or its outputs, and shall not be held liable for any consequences arising from its use in clinical or quasi-clinical settings.

1.3. License

OpenTPS is distributed under a **modular, mixed open-source licensing model**. The **core libraries** are licensed under the **Apache License 2.0**, a permissive open-source license that allows broad academic and industrial use, including commercial prototyping and derivative works. The

graphical user interface is distributed under the **GNU General Public License (GPL v3 or later)** due to its dependency on PyQt under GPL. Some components integrated into the main repository, such as the **photon dose module (WiscPlan)**, are released under **GPL v2 (or later)**. Optional functionality is provided through separate **plug-ins**, which currently include **Flash TPS** and **PRBIO**, both licensed under **Apache-2.0**. Third-party dependencies such as VTK (BSD-style) and PyQtGraph (MIT) are permissively licensed and do not impose additional copyleft constraints beyond their own terms. Users and integrators are responsible for ensuring compliance with all applicable licenses when using, modifying, or redistributing OpenTPS and its components.

OpenTPS license overview

Component	Description	Repository	License	Notes
Core libraries	Core TPS logic, optimization, scripting API	Main OpenTPS repo	**Apache License 2.0**	Permissive; allows commercial use
GUI	Graphical interface (widgets, menus, interaction)	Main OpenTPS repo	**GPL v3 (or later)**	Due to **PyQt** under GPL
Photon dose module (WiscPlan)	Photon dose calculation integrated in main code	Main OpenTPS repo	**GPL v2 (or later)**	Own license; applies only to this module
Visualization	Image rendering via **VTK**	Third-party	**BSD-style**	Permissive; no copyleft
Plotting / DVH	Profiles, DVH plots via **PyQtGraph**	Third-party	**MIT**	Permissive
Plug-ins:	Extensions not in core repo	Separate repos	Varies	See below
— **Flash TPS**	FLASH proton therapy extensions	gitlab.com/flash-tps/flash-tps	**Apache License 2.0**	Same permissive terms as core (about.gitlab.com)
— **PRBIO**	Probabilistic & radiobiological optimization	github.com/Eliot-P/PRBIO	**Apache License 2.0**	Permissive research license (GitHub)

1.4. Versioning

1.5. Installation

We refer to the Quickstart section in our website for the step-by-step installation of OpenTPS. You will find two options: 1) **Easy Quickstart**, suitable for most users, willing to use OpenTPS features without developing new plug-ins or modifying the source code; and 2) **Advanced Quickstart**, for users wanting to actively contribute to the project.

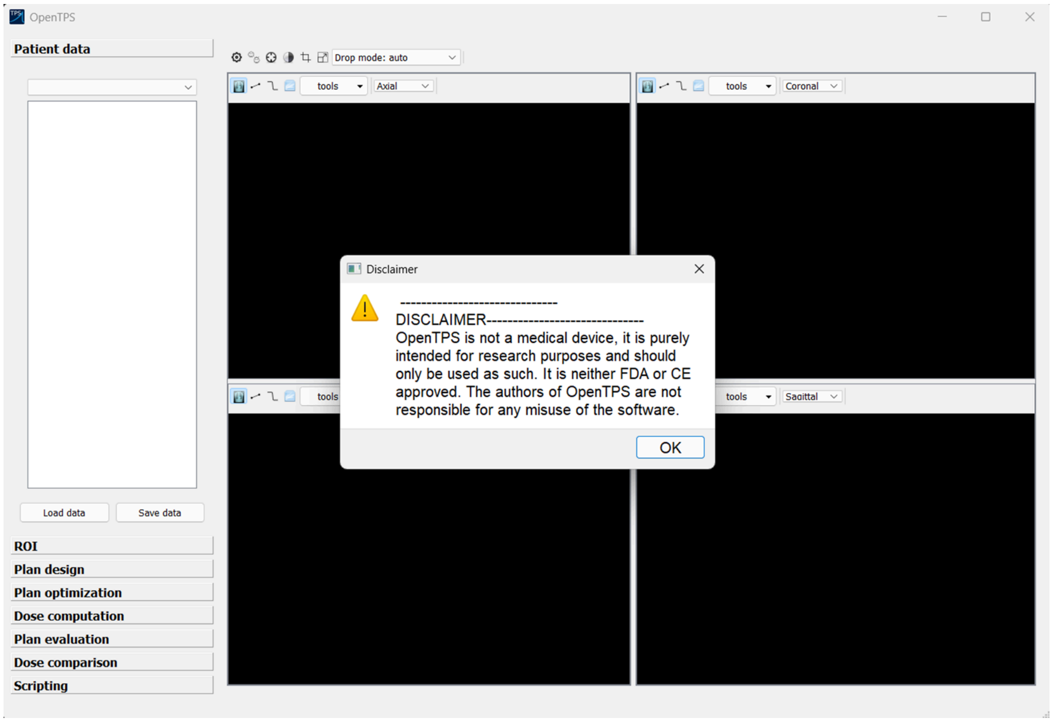
2. GUI features and user guide

2.1. Starting the GUI

Once the OpenTPS environment has been correctly created and activated, the GUI can be launched in two ways:

1. From the command line by executing: `start_opentps_anaconda_windows.bat`
2. From a Python script, for example:

```
import opentps.gui as opentps_gui
opentps_gui.run()
```



DISCLAIMER-------------------------------
OpenTPS is not a medical device, it is purely intended for research purposes and should only be used as such. It is neither FDA or CE approved. The authors of OpenTPS are not responsible for any misuse of the software.

When the GUI starts, a disclaimer dialog is displayed. The user must acknowledge this disclaimer to access the main interface.

2.2. Patient data structure and display

1. Patient Data Panel (left side)

The Patient data panel is the entry point for any workflow. At this stage, the interface is empty and no images are displayed.

- The Load data button allows importing patient datasets.
- The Save data button is used to store the current session.

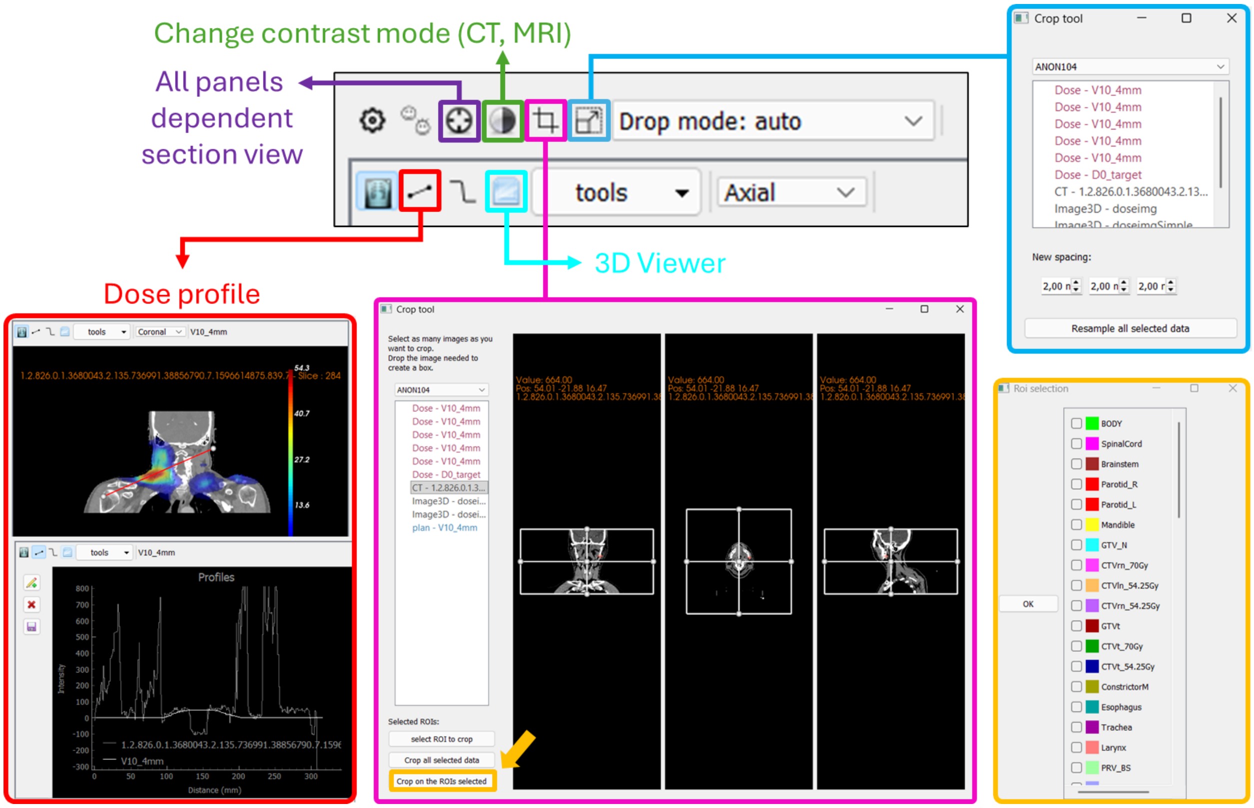
Supported data configurations include:

- CT images (typically used when creating a treatment plan from scratch and performing optimization.)
- RT plan + dose images (used when reviewing an existing plan, analyzing dose distributions, or recreating/modifying a plan.)
- MR Images

No further actions in OpenTPS are possible until data is loaded.

2. Visualization Views (center and right)

The central area is reserved for medical image visualization. Once data is loaded, it displays three orthogonal views: axial, coronal, and sagittal.



At the top of the interface, a global visualization toolbar provides a set of tools that control how data is displayed, independently of the current workflow step. First, a parameter button opens a panel where you can select a specific workspace, scanner folder of BDL file as well as remove/add some sections/panels in the visual interface for the content to be more adapted to your goals. Next to this button, the Independent Views button toggles whether all viewports share the same displayed dataset or can be controlled separately. When OFF (default), the application uses a single global Main Image (and optional overlays such as secondary image/dose), which is shown consistently across the axial/coronal/sagittal viewers. When ON, each viewer panel can display a different dataset independently, enabling side-by-side comparisons (e.g., CT vs MR, different phases of a 4DCT, or multiple reconstructions) without changing the content of the other viewports. To have an independent image view, you have to click on one specific image in the Patient Data panel and put it in the image viewer you want. Just next to this, there is the contrast mode button that enables an adjustment of the contrast and window level for the primary image (Typically CT or MRI). The Crop tool allows spatially restricting one or more images either on a manually defined box or based on ROIs that you can select via the “Select ROI to crop” button. The Resample tool changes the spatial resolution of selected images where you can choose the new voxel spacing to apply and resample multiple images simultaneously. The Crosshair tool enables interactive navigation across views. By moving the pointer in one view updates slice positions in all other views. This allows fast anatomical navigation and a precise localization of dose or structures in 3D space thanks to the displayed 3D coordinates.

Each view has:

- A toolbar at the top for navigation and interaction tools.
    - The Dose profile tool enables quantitative inspection along a line. By clicking on it, a white point will appear in another image viewer and you can construct from it a line on a selected slice. A profile plot is generated showing dose (or intensity) vs distance along the line. This is useful for analyzing dose gradients and verifying dose fall-off across structures (to see that, at least one ROI must be selected in the ROI panel)
    - A DVH visualiser (to see curves, at least one ROI must be selected in the ROI panel and one dose image must be applied)
    - The 3D viewer opens a three-dimensional visualization and enables an Interactive rotation and zoom.
- A drop-down menu to control image visualisation, colorbar and content. For both primary and secondary images, the user can adjust visualization settings via the tools menu:
    - Colormap selection
    - Colorbar visibility
    - Window / level (for CT)
    - Value range (for dose or comparison images)
- A drop-down menu to select the slice orientation.

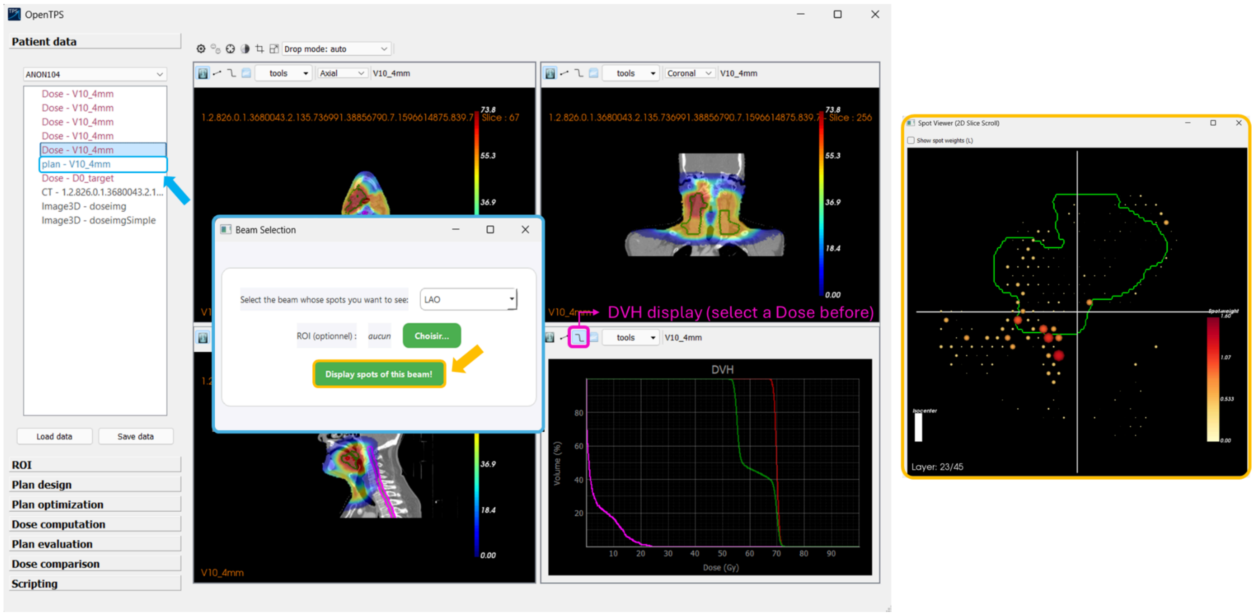
At startup, these views appear black because no imaging data has been loaded yet.

3. Workflow Panels (lower left)

Below the Patient data panel, a set of workflow modules is listed:

- ROI
- Plan design
- Plan optimization
- Dose computation
- Plan evaluation
- Dose comparison
- Scripting

These modules remain inactive or unusable until valid patient data is loaded.



Select the beam whose spots you want to see: LAO

ROI (optionnel) : aucun

DVH display (select a Dose before)

Once patient data is loaded, the CT or the MRI image is displayed automatically in the axial, coronal, and sagittal views.

- Slice indices and intensity color bars are shown on each view.
- Navigation tools are available from the toolbar above each viewport.

At this stage, only anatomical information is visible.

4. Displaying other images

Dose distributions can be displayed by double click on a dose object in the *Patient data* list (left panel).

- Clicking on a dose immediately overlays it on all active views.
- The active dose name is shown in each viewport header.

To display the CT image, you have to click on it and drop down to one image viewer panel.

Each visualization panel supports two image layers:

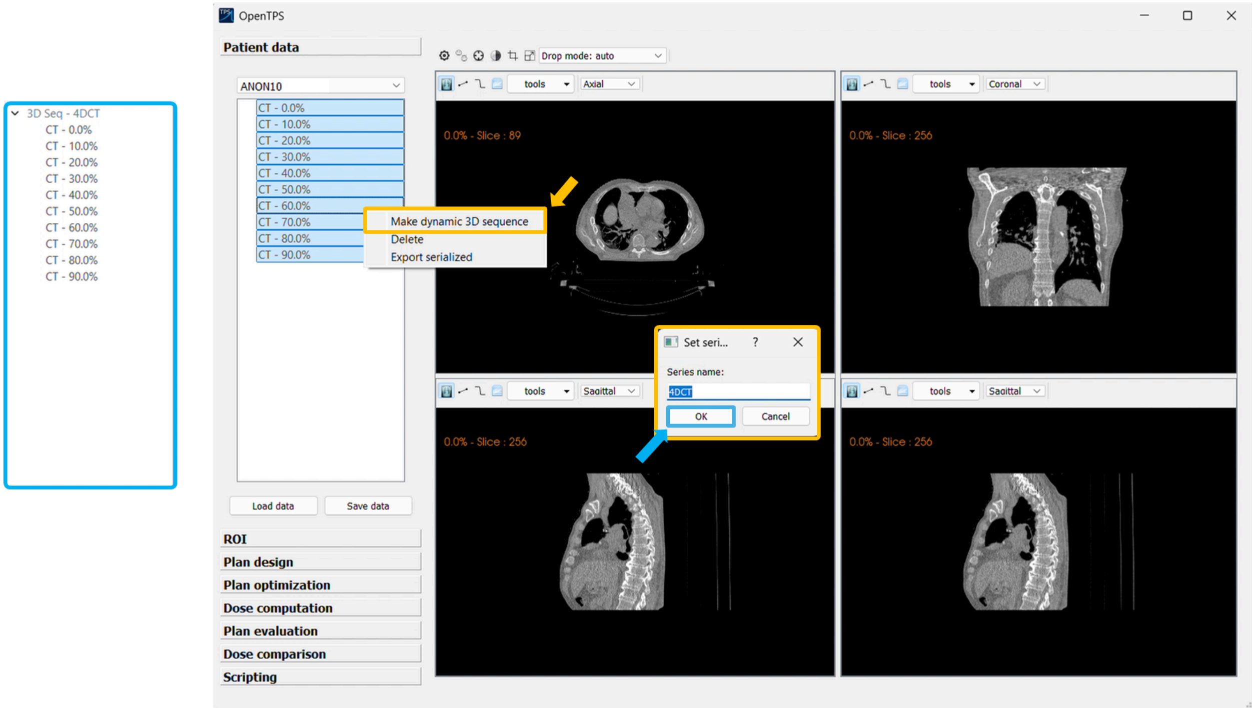
- Primary image (Typically the CT image, defining anatomical context.)
- Secondary image (Typically a dose image or comparison map, displayed as an overlay on the primary image.)

When a dose is selected, the DVH of specific volumes can be plotted:

- Dose-volume curves are shown for the available and checked ROIs.
- If no dose is selected, the DVH panel remains empty.

Each ROI is associated with a checkbox that controls its visibility across the interface. Unchecked ROI is hidden everywhere while checked ROI is displayed consistently in all relevant panels (contours in image view and its DVH curve in the DVH panel). Multiple ROIs can be displayed simultaneously, each with its own color. If the ROI does not intersect the current slice, no contour is shown in that view.

5. Dynamic Mode



Series name: 4DCT

OpenTPS allows the creation of dynamic 3D sequences from a set of selected image datasets (e.g. multiple CT phases). By selecting several images in the Patient data panel, clicking right and choosing Make dynamic 3D sequence, the selected datasets are grouped into a single named series that you can modify (e.g. 4DCT). Once created, the sequence appears as a new entry in the Patient data panel while the original images remain accessible and can still be visualized individually in static mode. When the dynamic series is selected, the image viewers display the sequence as a continuous loop across all images in the series and on the four panels, enabling temporal visualization of anatomical changes.

6. Displaying a Treatment Plan, Beams and Spots

When a treatment plan is selected in the Patient data list:

- The associated beam geometries are displayed in the image views.
- Beam directions and isocenter information become visible.
- A Beam Selection panel opens automatically.

The Beam Selection panel provides access to spot-level information:

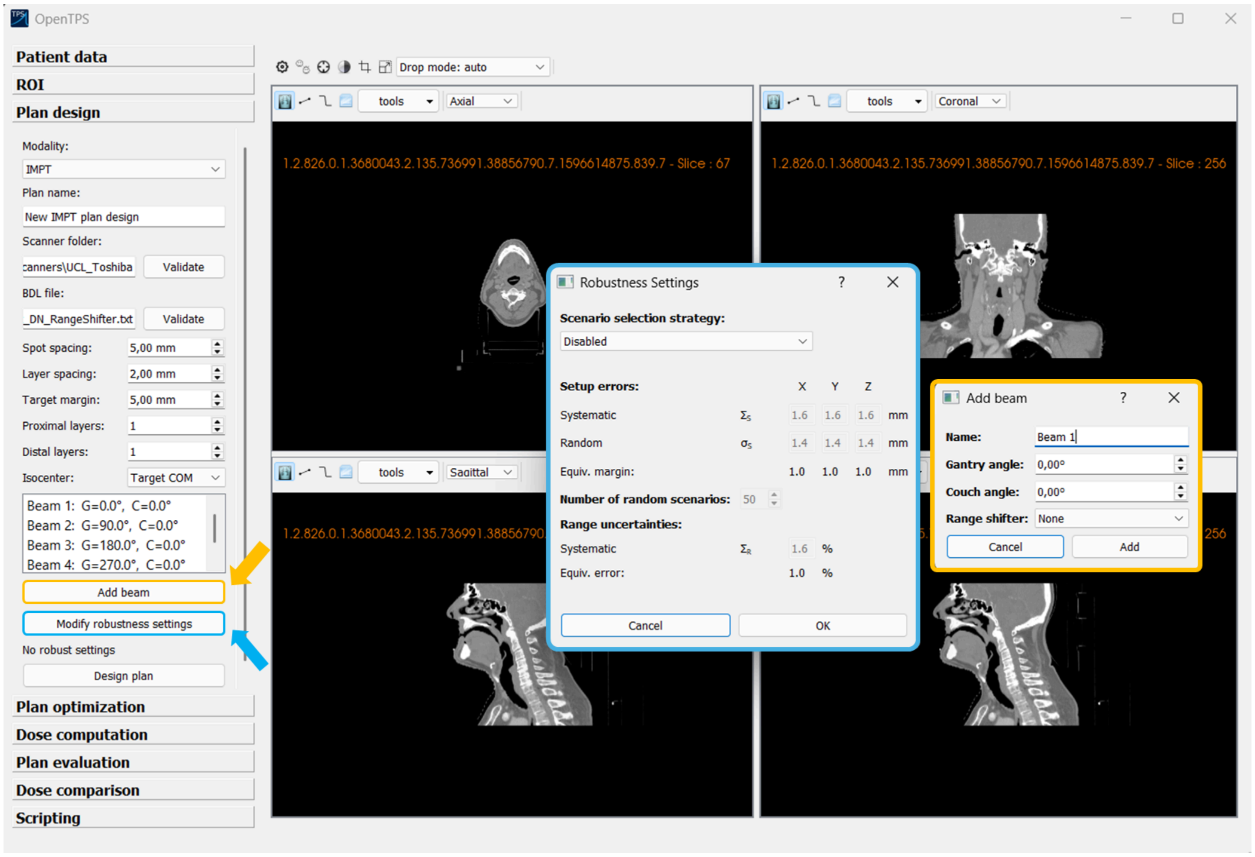
- A specific beam can be selected from the drop-down menu.
- Optionally, a ROI can be chosen to visualize spots on top of the specific ROI contour.

- Clicking “Display spots of this beam” opens a dedicated Spot Viewer window where proton spots are displayed layer by layer. The user can scroll through energy layers. Spot weights are visualized using a color scale but also the precise value can be accessible by checking the button (top left).

The following interactions are available in all image views:

- Slice navigation: Scroll the mouse wheel to move through slices in the current orientation.
- Zoom and pan: Right-click and move.

These interactions apply independently to each view but remain spatially consistent.



Modality: IMPT

Plan name: New IMPT plan design

Scanner folder: canners\UCL_Toshiba

BDL file: _DN_RangeShifter.txt

Spot spacing: 5,00 mm

Layer spacing: 2,00 mm

Target margin: 5,00 mm

Proximal layers: 1

Distal layers: 1

Isocenter: Target COM

Beam 1: G=0.0°, C=0.0°
Beam 2: G=90.0°, C=0.0°
Beam 3: G=180.0°, C=0.0°
Beam 4: G=270.0°, C=0.0°

No robust settings

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 67

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 256

Robustness Settings

Scenario selection strategy: Disabled

Setup errors:		X	Y	Z	
Systematic	Σ_S	1.6	1.6	1.6	mm
Random	σ_S	1.4	1.4	1.4	mm
Equiv. margin:		1.0	1.0	1.0	mm

Number of random scenarios: 50

Range uncertainties:			
Systematic	Σ_R	1.6	%
Equiv. error:		1.0	%

Add beam

Name: Beam 1

Gantry angle: 0,00°

Couch angle: 0,00°

Range shifter: None

2.3. Plan Design Panel

The Plan design panel is used to define the structure of a treatment plan before optimization. At this stage, no dose is computed and no optimization is performed.

The user specifies:

- The plan identity and modality
- The beam model

- The beam geometry
- (Optionally) robustness settings

This information is required before running any plan optimization. The following parameters must be defined:

- Modality: Select the planning modality (e.g. IMPT).
- Plan name: User-defined identifier for the plan. This name is used throughout the workflow.
- Scanner folder / BDL file: The beam model directory defining machine characteristics.

These elements define the physical and numerical framework used during optimization and dose computation. Beams must be added manually using the Add beam button. For each beam, the user specifies:

- Beam name
- Gantry angle
- Couch angle
- Range shifter (if applicable)

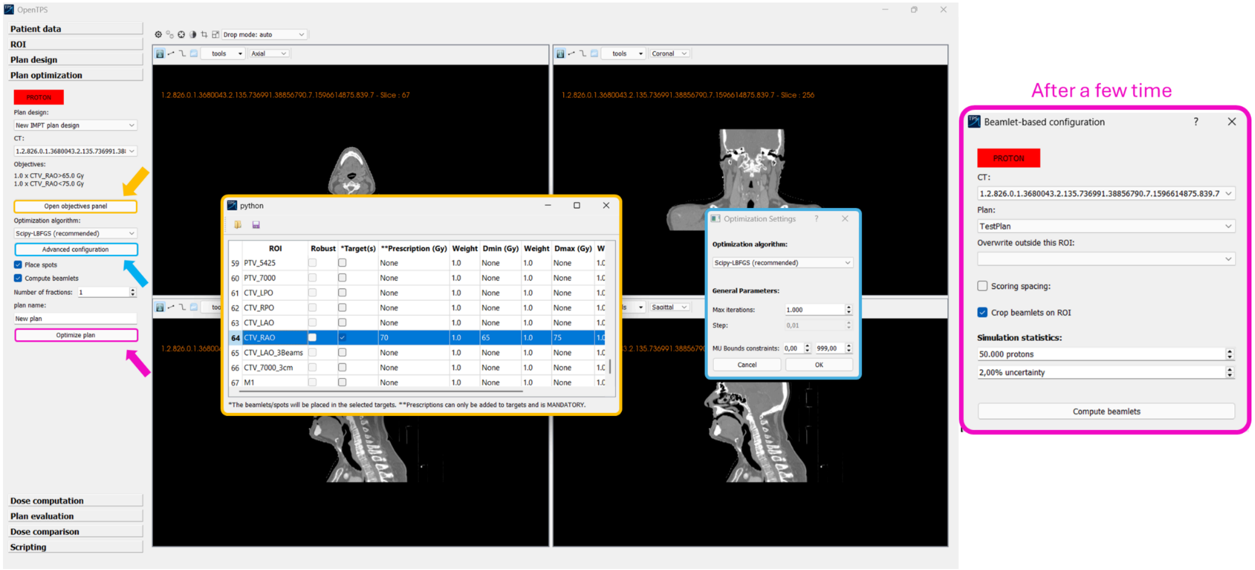
There is no fixed limit on the number of beams. Once added, beams appear in the beam list. They become available for optimization.

Robustness Settings (Optional)

Robustness parameters can be configured using “Modify robustness settings”. This opens a dedicated dialog where the user can define:

- Scenario strategy
- Setup uncertainties (systematic and random, per axis)
- Range uncertainties
- Number of random scenarios

If robustness is disabled, optimization is performed on the nominal scenario only. Once all required elements are defined, you can click on “Design plan” to create the plan object. The plan becomes available in the Patient data list and the workflow can then proceed to Plan optimization.



Plan design: New IMPT plan design

CT: 1.2.826.0.1.3680043.2.135.736991.38

Objectives:
1.0 x CTV_RAO>65.0 Gy
1.0 x CTV_RAO<75.0 Gy

Optimization algorithm: Scipy-LBFGS (recommended)

☑ Place spots

☑ Compute beamlets

Number of fractions: 1

plan name: New plan

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 67

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 256

	ROI	Robust	*Target(s)	**Prescription (Gy)	Weight	Dmin (Gy)	Weight	Dmax (Gy)	W
59	PTV_5425	□	□	None	1.0	None	1.0	None	1.0
60	PTV_7000	□	□	None	1.0	None	1.0	None	1.0
61	CTV_LPO	□	□	None	1.0	None	1.0	None	1.0
62	CTV_RPO	□	□	None	1.0	None	1.0	None	1.0
63	CTV_LAO	□	□	None	1.0	None	1.0	None	1.0
64	CTV_RAO	□	☑	70	1.0	65	1.0	75	1.0
65	CTV_LAO_3Beams	□	□	None	1.0	None	1.0	None	1.0
66	CTV_7000_3cm	□	□	None	1.0	None	1.0	None	1.0
67	M1	□	□	None	1.0	None	1.0	None	1.0

*The beamlets/spots will be placed in the selected targets. **Prescriptions can only be added to targets and is MANDATORY.

Optimization Settings

Optimization algorithm: Scipy-LBFGS (recommended)

General Parameters:

Max iterations: 1.000

Step: 0,01

MU Bounds constraints: 0,00 999,00

After a few time

Beamlet-based configuration

PROTON

CT: 1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7

Plan: TestPlan

Overwrite outside this ROI:

□ Scoring spacing:

☑ Crop beamlets on ROI

Simulation statistics:

50.000 protons

2,00% uncertainty

2.4. Plan Optimization Panel

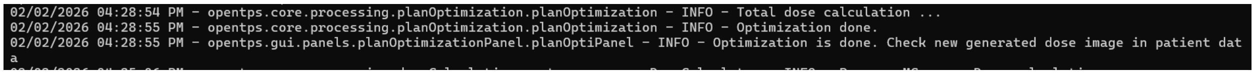
The Plan optimization panel is used to compute spot weights for an already designed plan. At this stage, the beam geometry is fixed, the CT and at least one ROI must already exist and the plan structure must have been created in *Plan design*. In the "Objectives panel", you have to check at least one ROI target among the listed ones to receive a prescribed dose. For each target selected, the user can define:

- Prescription dose
- Minimum and maximum dose constraints
- Associated weights

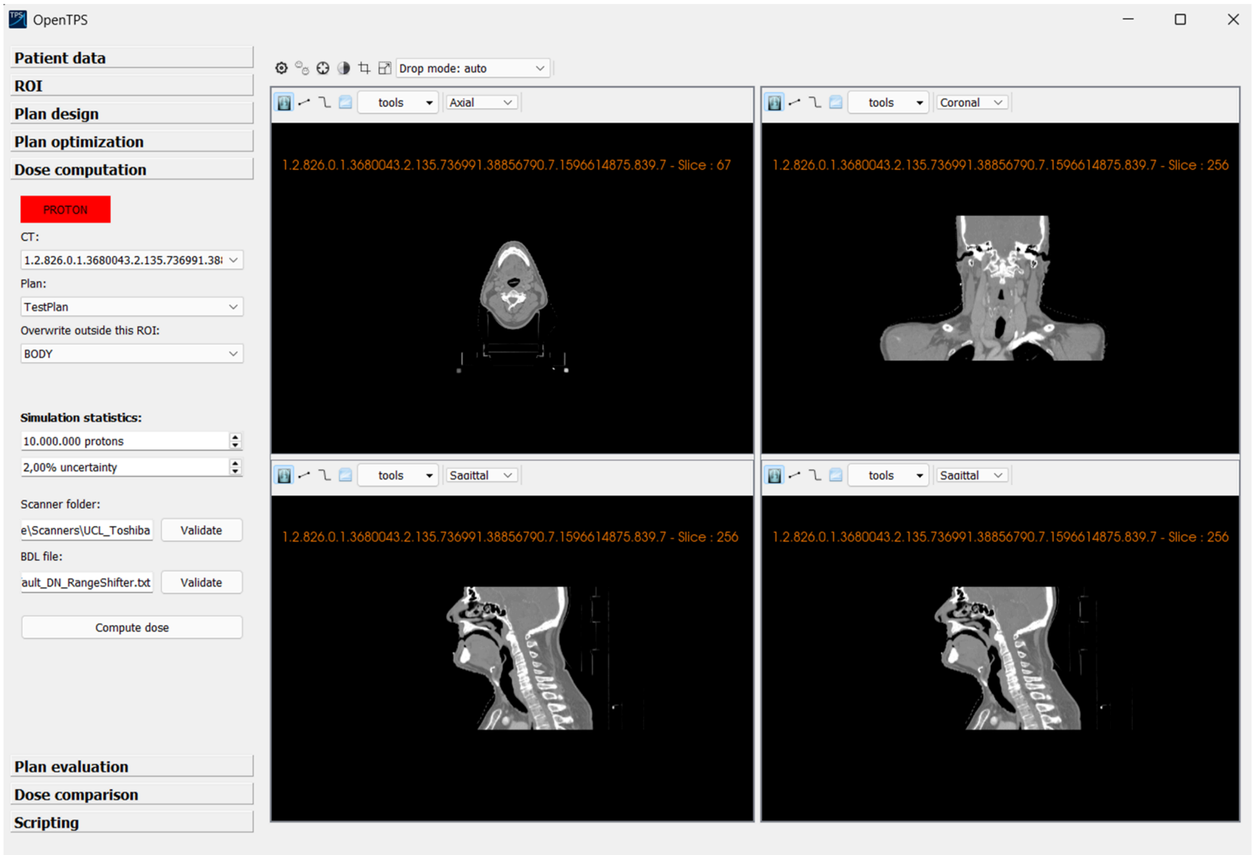
Non-target ROIs may be used for constraints but do not require prescriptions. The optimization algorithm "Scipy-LBFGS" is the recommended default but the user can choose among the accessible ones. The "Advanced configuration" panel is used to modify the optimization parameters, including the maximum number of iterations, the step size, the MU bounds and convergence-related parameters. Click Optimize plan to start the optimization. It will run iteratively and after a moment, a panel about the beamlets-based configuration will appear. This step allows to choose several options such as:

- Cropping beamlets to a selected ROI
- Scoring spacing
- Simulation statistics (number of protons, uncertainty level)

Click Compute beamlets to generate beamlet dose kernels. The beamlet computation step is time-consuming but performed once per configuration. After completion, the optimized spot weights are stored in the plan. The new plan becomes available in the Patient data panel and can be used for dose computation and evaluation. A message appears in the command prompt.



```
02/02/2026 04:28:54 PM - opentps.core.processing.planOptimization.planOptimization - INFO - Total dose calculation ...
02/02/2026 04:28:55 PM - opentps.core.processing.planOptimization.planOptimization - INFO - Optimization done.
02/02/2026 04:28:55 PM - opentps.gui.panels.planOptimizationPanel.planOptiPanel - INFO - Optimization is done. Check new generated dose image in patient dat
a
```



Drop mode: auto

CT: 1.2.826.0.1.3680043.2.135.736991.38

Plan: TestPlan

Overwrite outside this ROI: BODY

Simulation statistics:

10.000.000 protons

2,00% uncertainty

Scanner folder: e\Scanners\UCL_Toshiba

BDL file: ault_DN_RangeShifter.txt

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 67

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 256

1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 256

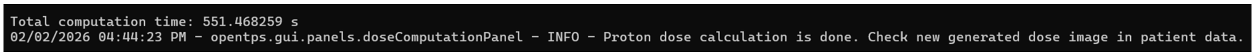
1.2.826.0.1.3680043.2.135.736991.38856790.7.1596614875.839.7 - Slice : 256

2.5. Dose computation Panel

The Dose computation panel is used to compute the final 3D dose distribution from an optimized plan. Dose computation is independent from optimization and can be repeated with different settings. Dose calculation is based on Monte Carlo simulation parameters:

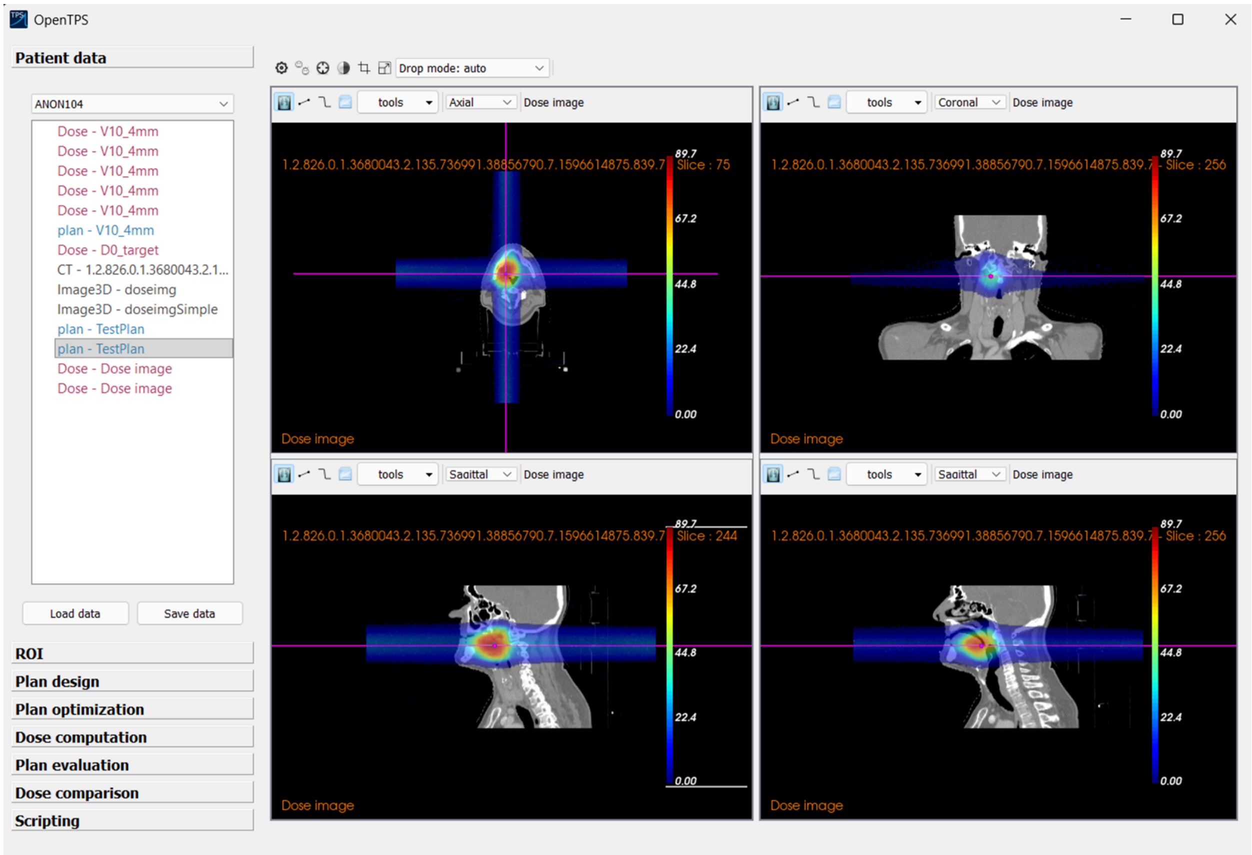
- Number of protons parameters to choose the total number of simulated protons per beamlet.
- Statistical uncertainty defines the target uncertainty level for the dose calculation.

Click Compute dose to start the calculation and the dose is computed on the full image or a restricted ROI, depending on user choice. After completion, a new dose object is added to the Patient data list and a message appears in the command prompt.



```
Total computation time: 551.468259 s
02/02/2026 04:44:23 PM - opentps.gui.panels.doseComputationPanel - INFO - Proton dose calculation is done. Check new generated dose image in patient data.
```

⚠️ **Warnings about the two following pictures : Those are just examples used for an illustration purpose and not realistic or clinically valid plans! They must not be interpreted as a realistic, optimized, or clinically acceptable treatment plan.**

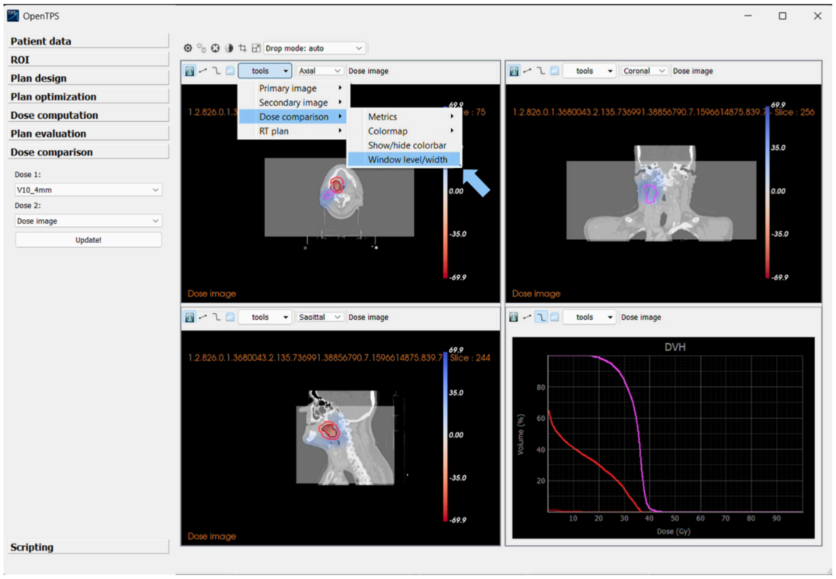


The new dose image can now be visualised and can for example be compared to the initial plan thanks to the Dose comparison panel.

2.6. Plan evaluation

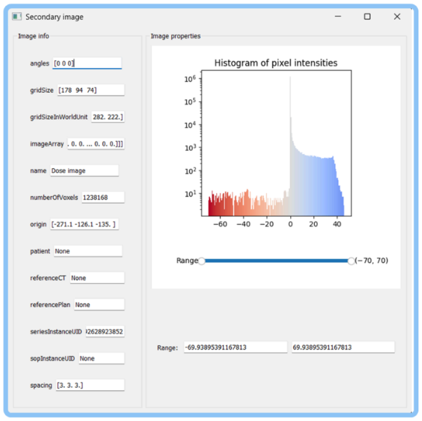
This panel is used to make a robustness evaluation of a selected plan but needs to be updated in the current version and is for the moment not supported in the GUI.

2.7. Dose comparison Panel



Dose 1: V10_4mm

Dose 2: Dose image



angles [0 0 0]

gridSize [178 94 74]

gridSizeInWorldUnit 282. 222.]

imageArray . 0. 0. ... 0. 0. 0.]]]

name Dose image

numberOfVoxels 1238168

origin [-271.1 -126.1 -135.]

patient None

referenceCT None

referencePlan None

seriesInstanceUID)2628923852

sopInstanceUID None

spacing [3. 3. 3.]

Range: -69.93895391167813 69.93895391167813

The Dose comparison panel allows direct comparison between two dose images. The user selects Dose 1 (reference) and Dose 2 (comparison) and once selected, the comparison updates immediately. Both spatial and DVH-based differences can be analyzed. Dose comparison is displayed as:

- A secondary image overlay representing dose differences or metrics
- A DVH comparison when ROIs are enabled

2.8. Scripting

The Scripting panel allows execution of standalone Python code snippets within the OpenTPS GUI. More information will be available as soon as possible on the way to access patient data from this part.

3. Core features

⚠️Note that all the described features in this section are included in the core libraries, but some of them might not be available in the GUI. For a complete description of features in the GUI, see Section 2.

⚠️When explaining each feature, some key classes and methods are pointed to as illustrative purposes, but this guide is not meant to be a scripting API. For a complete description of the core functions please visit https://opentps.org/docs_html/index.html

3.1. Data loading and exporting

OpenTPS enables users to work with different data files in **DICOM[1] (.dcm)** format, the international standard for medical image and related data (see table below). In addition to DICOM, OpenTPS supports research-oriented data formats such as **MetaImage (MHD/RAW)** and serialized Python objects. MHD files are commonly used for volumetric data in research. They contain a text header (**.mhd**) describing the image (size, spacing, data type, orientation, etc.), and the actual voxel data is stored in a companion **.raw** file. Serialized formats store internal OpenTPS data structures using Python pickle format (.p, .pkl, .pickle) or compressed pickle (.pbz2).

Data type	Import	Export	Comments / Limitations
DICOM CT	✅	✅	Export functionality can be useful when generating a phantom with basic shapes through scripting, and then wanting to export as DICOM Tested for vendors XXXX
DICOM MR	✅	❌	Currently not allowing right display of MRs with non standard ImageOrientation
DICOM PET	✅	❌	
DICOM RTSTRUCT	✅	✅	
DICOM RTDOSE	✅	✅	

[1] If you are new to DICOM, we encourage you to visit the Innolitics open-source initiative to get to know all about the different file types and information that can be stored - https://dicom.innolitics.com/ciods

DICOM REG	✅	❌	
DICOM RTPLAN	✅	✅	
MHD	✅	✅	No standardized clinical metadata. Not suitable for clinical workflows on its own

```
fileLists = {
        "Dicom": [],
        "MHD": [],
        "Serialized": [],
        "txt": []
```

point to functions `readData`

3.2. Image processing

3.2.1. Segmentation

OpenTPS provides a limited but practical set of tools to perform segmentation on CT images and to generate **ROIMask** objects. These tools are primarily designed for simple, intensity-based segmentation workflows.

Segmentation in OpenTPS focuses on threshold-based extraction of anatomical regions from CT images. Users can isolate structures by applying intensity ranges, and optionally compute bounding boxes around the resulting regions to support cropping or targeted processing. The output is a **ROIMask** that can be used by downstream modules (e.g., registration, dose computation, or visualization).

In practice, the workflow starts by selecting an intensity range that corresponds to the structure of interest. The image is then converted into a binary mask, where voxels inside the selected range are kept and voxels outside are discarded. The resulting mask can be used directly or refined further by spatial constraints such as bounding boxes in scanner coordinates.

3.2.1.1. SegmentationCT class

OpenTPS also provides the **SegmentationCT** class, which offers predefined CT segmentation presets based on HU. These presets are designed for common anatomical targets such as body, bones, and lungs, and they return **ROIMask** objects compatible with the rest of the framework. This provides a quick and consistent way to obtain standard anatomical masks without manual threshold tuning.

3.2.2. Registration

The registration workflow includes image resampling, optional region-of-interest constraints, and smooth deformation modeling. These capabilities enable robust alignment of 3D images while preserving anatomical consistency.

In practice, a fixed image is selected as the reference grid. The moving image is resampled to this grid, and the alignment quality is assessed by comparing intensities between the two images. When a region of interest is defined, computations are restricted to that anatomical area. Deformation fields are regularized to enforce smooth, physically plausible motion, which improves stability and reduces unrealistic distortions.

3.2.2.1. midPosition

OpenTPS also supports mid-position (MidP) image computation for 4D sequences. This generates a representative reference image by combining motion information across all phases.

The process registers each phase of a 4D sequence to a reference phase, averages the resulting motion fields, and then applies these fields to deform all phases toward a common mid-position. The mid-position image is then computed by combining the deformed phases.

3.2.3. Others

3.2.3.1. Apply baseline shift

3.2.3.2. Cupy image processing

3.2.3.3. Sitk image processing

3.2.3.4. ROI mask processing

3.2.3.5. Resample

3.3.

3.4. Dose calculation

The following dose engines are currently interfaced with OpenTPS.

WiscPlan

MCsquare

Mcsquare[2,3] is a **Monte Carlo** dose engine developed specifically for **pencil-beam scanning proton** therapy. The code was optimized for modern **multi-core CPU architectures**, exploiting parallel processing to achieve high performance without requiring specialized hardware such as GPUs. In a typical desktop, a MCsquare simulation with 10^6 primaries takes less than 1 minute.

MCsquare has been benchmarked against general-purpose Monte Carlo codes such as GATE/Geant4, showing excellent agreement within about 2% dose and 1 mm spatial accuracy across both homogeneous and heterogeneous media.

⚠️Note that OpenTPS uses the compiled binaries from MCsquare to call a dose calculation, and the source code is hosted in an independent GitLab repository from OpenTPS: https://gitlab.com/openmcsquare/MCsquare. For more information, visit also MCsquare website.

Inputs for MCsquare include a voxelized CT image with material assignment (via Hounsfield units), proton spot parameters (energies, positions, weights), and machine-specific beam model data. **Outputs** are three-dimensional dose distributions.

3.5. Plan design

3.5.1. targetMargin

Defines the margin (in millimeters) added around the target volume within which spots will be placed.

[2] http://www.openmcsquare.org/

[3] Souris **et al.**, *Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures Med Phys* **43(4)** (2016) https://doi.org/10.1118/1.4943377

3.5.2. defineTargetMaskAndPrescription

Defines the region where spots are placed, based on the provided target mask and the targetMargin. The prescription is used only for initializing spot weights; therefore, multiple prescriptions are not required.

3.6. Optimization

3.6.1. Multiple targets

Regarding multiple-target optimization, this functionality is not fully implemented in OpenTPS yet. As a workaround, we recommend creating a union of the different target masks and using this combined ROI with the defineTargetMaskAndPrescription method.
In practice, you can proceed as follows:

1. Create a single ROI mask corresponding to the union of all target volumes.
2. Pass this ROI mask to defineTargetMaskAndPrescription.
3. Define the optimization objectives separately using the original individual target masks, allowing different objectives for each target.

3.6.2. Robust

3.6.3. 4D

3.6.4. Beamlet-free

3.7. Plan evaluation

3.8. Plug-ins

For feature development, we strongly encourage adopting a “plug-in” approach. The OpenTPS core package is designed to remain as stable and robust as possible. To achieve this, we recommend using plug-ins to add or extend functionalities.

A plug-in is an independent module created to integrate seamlessly with OpenTPS without being part of its core system. OpenTPS itself does not depend on any plug-ins to function, ensuring its stability and reliability. Instead, plug-ins serve as optional add-ons that allow users to implement new features or customize their workflows.

To illustrate this concept, here are two examples of such plug-ins:

3.8.1. Flash

FLASH proton therapy is an emerging technique in radiation oncology that involves delivering ultra-high doses of radiation to a tumor in a very short period—typically less than one second.

This innovative approach uses protons as the radiation source and has gained attention due to its potential to significantly reduce damage to surrounding healthy tissues while effectively targeting cancer cells. A FLASH plugin for OpenTPS v1.1.2 is available here

3.8.2. PRBIO

RBIO is a plug-in developed for probabilistic and radiobiological optimizations. It provides functionality for creating simple Clinical Target Distributions (CTDs) and introduces new optimization capabilities based on metrics such as Tumor Control Probability (TCP), Normal Tissue Complication Probability (NTCP), and CTD based TCP (named PTCP). More details about the PRBIO plug-in, along with access to the tool, can be found here